ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Maestría en Seguridad Informática Aplicada

"APLICAR HACKEO ÉTICO PARA DETECCIÓN DE VULNERABILIDADES MEDIANTE HERRAMIENTAS OPEN SOURCE EN LAS APLICACIONES WEB DE UNA INSTITUCIÓN DE EDUCACIÓN SUPERIOR"

EXAMEN DE GRADO (COMPLEXIVO)

Previa a la obtención del título de:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

IVÁN ALBERTO CORONEL SUÁREZ

GUAYAQUIL – ECUADOR

AÑO 2016

AGRADECIMIENTO

A Dios por darte la fortaleza y salud para cumplir cada una de mis metas propuestas. A mis padres por haberme inculcado ejemplo de superación y valores para convertirme en quien soy. A mi familia por creer en mí. A la Escuela Superior Politécnica del Litoral - ESPOL por brindarnos la oportunidad de formarnos en sus aulas y a los docentes y directores por la ayuda brindada.

DEDICATORIA

El presente trabajo se lo dedico a mi familia, a mis hermanas “Marjorie y Evelyn” y a mi pequeña hija “Jahely”, que son el motor que me impulsa a seguir adelante en mi preparación profesional.

TRIBUNAL DE SUSTENTACIÓN

Ing. Lenín Freire Cobo

DIRECTOR MSIA

Mgs. Rocky Barbosa Gilces

PROFESOR DELEGADO POR LA

UNIDAD ACADÉMICA

Mgs. Omar Maldonado Dañin

PROFESOR DELEGADO POR LA

UNIDAD ACADÉMICA

RESUMEN

En el presente trabajo se realiza un hacking ético a una institución de educación superior, la misma que cuenta con aplicaciones en línea para brindar una serie de servicios tanto a sus alumnos como a docentes, aplicaciones que van desde consultas de notas, ingreso de fichas, encuestas, ingreso de notas y planes de clases.

En el primer capítulo se da una breve introducción, se expone el problema y se plasma la propuesta de la solución. Luego se definen los procedimientos a realizar en la evaluación y se llevan a cabo paso a paso desde la recolección de información, que es donde se ponen en práctica técnicas de footprinting para obtener datos que nos pueden ayudar en el presente trabajo, en este capítulo se deja en evidencia que existe mucha información que puede ser útil para un delincuente informático y que podría obtener de una forma muy sencilla, como se lo expone aquí.

Se realizan trabajos de evaluación de vulnerabilidades y se enfoca en la más crítica para explotarla y así obtener la data de toda la institución, lo que

presenta un problema ya para la confidencialidad de la información y si el atacante así lo quisiera, poder llevar a cabo un perjuicio mayor afectando a la disponibilidad e integridad de la institución.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

La Institución de Educación Superior objeto de este estudio se encuentra legalmente constituida mediante su respectivo decreto de Ley y cuenta con su Registro Oficial, desde 1998 ha sido el alma mater de su región acogiendo a más de 4400 estudiantes en sus aulas.

La Universidad tiene como Visión liderar los procesos de la educación superior, investigación científica y tecnológica, de la región por lo que se encuentra en un constante crecimiento y tecnificación de todos sus servicios, maneja una gran cantidad de información en sus redes de comunicación y almacena información muy valiosa en sus bases de datos.

Con la conexión de todos sus usuarios a las redes y servicios de la Universidad se transmite un sin número de tráfico que contiene grandes cantidades de información de todo tipo lo que requiere se implementen sistemas de seguridad y barreras que mitiguen de alguna forma los riesgos que se presentan cuando equipos finales se encuentran comprometidos con algún tipo de virus, también

en caso de existir alguna vulnerabilidad en las aplicaciones que se desarrollan en la Institución.

Por todos los antecedentes expuestos es necesaria la identificación de los fallos de seguridad que presentan las aplicaciones web y que se identifiquen las debilidades en los servicios que presta la Institución a la comunidad universitaria.

CAPÍTULO 1

GENERALIDADES

1.1. DESCRIPCIÓN DE PROBLEMA

La información es el activo más importante de una institución pública o privada y la constante evolución de las Tecnologías de Información y Comunicación hacen necesario que estas se integren y evolucionen llegando así a manejar su información y ejecutar grandes procesos por medio de sus redes de comunicación, pero en los últimos tiempos el acelerado desarrollo de un sin número de ataques informáticos a estas instituciones ya sea a su infraestructura tecnológica como a sus usuarios hacen que exista una amenaza constante a los activos de las misma.

El trabajo diario de los usuarios de las instituciones está vinculado directamente al uso de las TICs en los cuales se exponen a tipos de ataques que provienen de virus, gusanos, troyanos, rootkits y spyware que en su forma más típica logran hacer más lentos los sistemas de comunicación, mismas amenazas que pueden llegar a extraer y borrar archivos confidenciales de la institución incluso a utilizar los equipos para actividades ilícitas formando parte de grandes botnets para realizar ataques más estructurados a terceros

En el desarrollo de las instituciones de formación de tercer nivel es necesaria la incorporación de las TICs pasando a ser un indicador importante en las evaluaciones de las mismas, siendo por esto que se debe dar una principal atención ya que si surge un problema por mínimo que sea puede ocasionar un mal funcionamiento en las comunicaciones y llegar a poner en riesgo la disponibilidad de la información, la privacidad e integridad del activo más importante.

La Institución cuenta con una población universitaria creciente de 3359 estudiantes matriculados con una planta docente y administrativa de 500 usuarios y la falta de controles en la seguridad de las aplicaciones es un problema que está en crecimiento.

Los ataques a las aplicaciones representan un gran problema para el correcto funcionamiento de los sistemas y servicios, ataques que han

logrado realizar alteraciones en las páginas de inicio (defacement) de sitios, intentos fallidos de hackeo a servidores, virus en máquinas, robo de Identidad de cuentas, etc., estos tipos de incidentes han ocasionado que se queden sin operar ciertos servicios que presta la Institución por pequeños periodos de tiempo.

Por lo que se crea la necesidad de un estudio de este tipo con aplicación directa de un Hackeo Ético para detectar las vulnerabilidades en las redes y sistemas con el objetivo de tratar de mejorar las seguridades y mitigar los posibles problemas que puedan presentarse por estos ataques.

1.2. PROPUESTA DE SOLUCIÓN

Elaboración de pruebas de penetración (Pentesting) basado en las metodologías del Hackeo Ético de las aplicaciones de la Institución para la determinación de vulnerabilidades y riesgos que puedan comprometer la confidencialidad, integridad y disponibilidad de la información.

El presente trabajo como solución tiene como función principal fortalecer todo el escenario de seguridad en cuanto a la estructura de aplicaciones demostrando por medio de pruebas y análisis con una serie de herramientas de distribuciones Linux como son Kali y herramientas de plataforma Windows con licencias libres las cuales dejarán al descubierto diversas vulnerabilidades.

Se planifican trabajos de simulación de comportamientos y técnicas usadas por hackers buscando analizar los niveles de seguridad de las aplicaciones, permitiendo identificar vulnerabilidades y realizar correcciones de forma inmediata, basados en estándares.

Realizado el análisis de las vulnerabilidades de seguridad que se han descubierto se evaluará el tipo y el impacto que podría tener si se las explota, la ventaja principal del proyecto trazado serán los informes, correcciones y recomendaciones que permitirán asegurar y garantizar un ámbito de trabajo aceptable para la Institución, recalcando que la seguridad informática no solo se trata de contar con antivirus en las máquinas o servidores, ni tener un firewall en nuestra red, la seguridad va mucho más allá con la implementación de normas, políticas de comportamiento y de desarrollo basadas en el estudio anterior que todos los que somos usuarios de esta red debemos conocer y aplicar.

CAPÍTULO 2

PLANIFICACIÓN Y METODOLOGÍA

2.1. OBTENIENDO INFORMACIÓN DE LA INSTITUCIÓN

Reconocimiento o Footprinting, esta fase es la más importante dentro de la realización del hacking ético que consiste en la recopilación de toda la información necesaria para fases posteriores del proceso.

Calles, J. (2011) nos expone sobre el proceso de Footprinting: “consiste en la búsqueda de toda la información pública, bien porque haya sido publicada a propósito o bien por que haya sido publicada por desconocimiento (abierta, por tanto no estaremos incurriendo en ningún delito, además la entidad ni debería detectarlo)” [1].

Astudillo, K. (2013) resalta lo siguiente sobre esta fase: “es muy importante que le dediquemos nuestro mejor esfuerzo y cabeza a esta fase y que invirtamos todo el tiempo necesario en realizar un buen levantamiento de información” [2].

La fase de reconocimiento puede ser activo o pasivo, depende si se tiene o no interacción con el objetivo a evaluar, un reconocimiento pasivo se realiza por medio de búsquedas en Google, Bind (conociendo a profundidad características avanzadas de búsquedas), visitas a la página web, aplicaciones de la institución o empresa a la que se está realizando la evaluación, así como en diarios o publicaciones de procesos de compras, etc. se puede obtener información como:

- Quien es la empresa u objetivo
- URL de la página web de la empresa
- IPs externas de los servidores
- Proveedor de los rangos de IPs
- Empleados importantes o relevantes para este tipo de pruebas
- Números de teléfonos
- Ruta de los servidores
- Mail de la compañía u objetivo
- Redes Sociales
- etc.

Esta recopilación de información puede ser obtenida por servicios como: Domain name lookup, Whois, Nslookup o por medio de herramientas automatizadas.

En cuanto a herramientas y Plataformas a utilizar Astudillo, K. (2013) nos dice: “La plataforma de sistema operativo puede ser Windows, Linux o Unix, según su preferencia” [2]

Primeramente procedemos a averiguar la IP del objetivo a evaluar mediante el comando nslookup como se aprecia en la figura 2.1 en este caso nos interesa solo saber la dirección del servidor que contiene las aplicaciones web a evaluar.

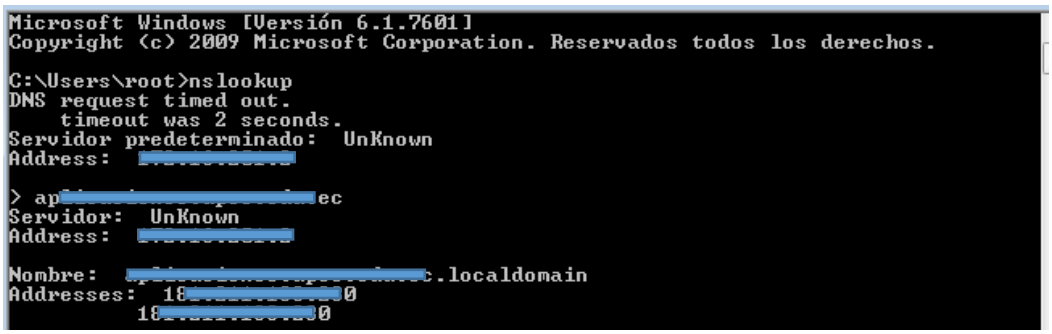


Figura 2.1 Resolución de la IP mediante comando nslookup

Teniendo el dominio y su IP vemos que información podemos obtener de herramientas como:

- Serversniff.net.ipaddress.com
- Domaintools.com

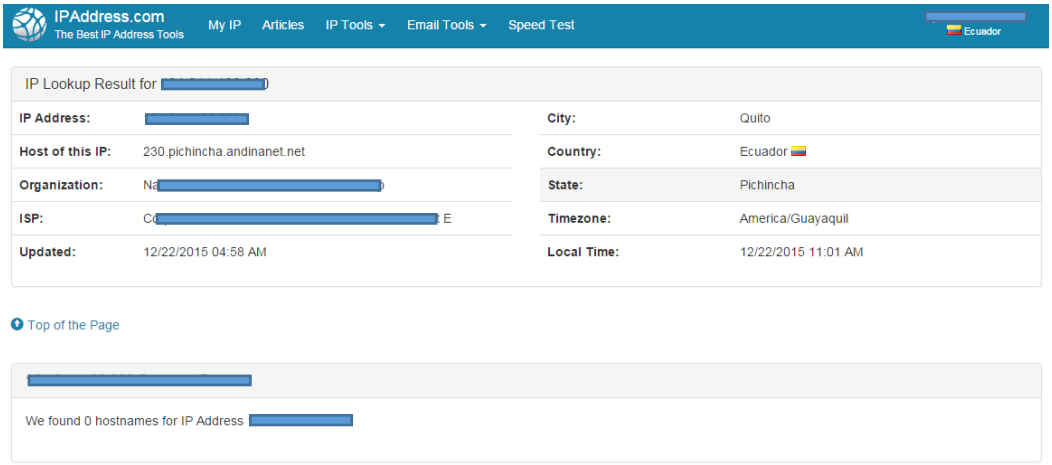


Figura 2.2 Información obtenida mediante Serversniff.net

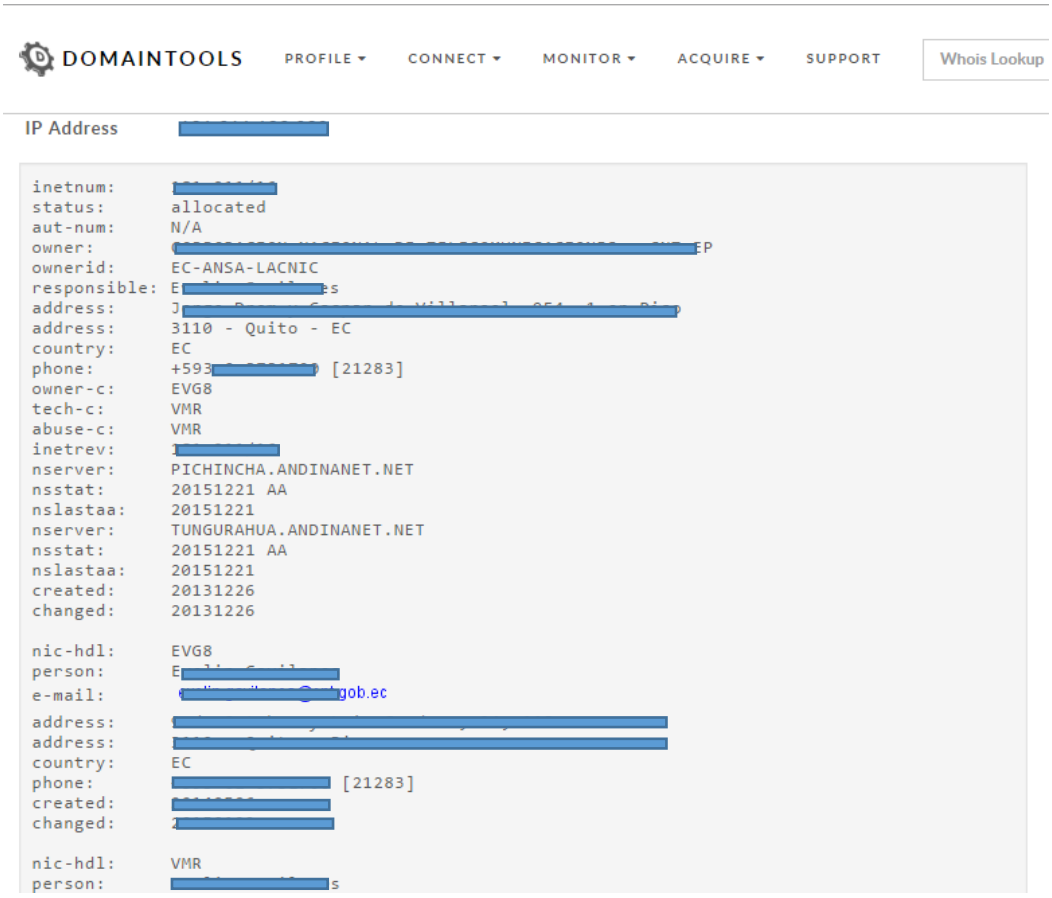


Figura 2.3 Información obtenida del servicio Domaintools

Siguiendo con la evaluación procedemos a consultar información a la base de datos Who-Is donde obtendremos información como propietario del dominio, la dirección, números de teléfonos, servidores DNS, etc.

Astudillo, K. (2013) nos dice: “El Who-Is es un protocolo que permite hacer consultas a un repositorio en Internet para recuperar información acerca de la propiedad de un nombre de dominio o una dirección IP” [2].

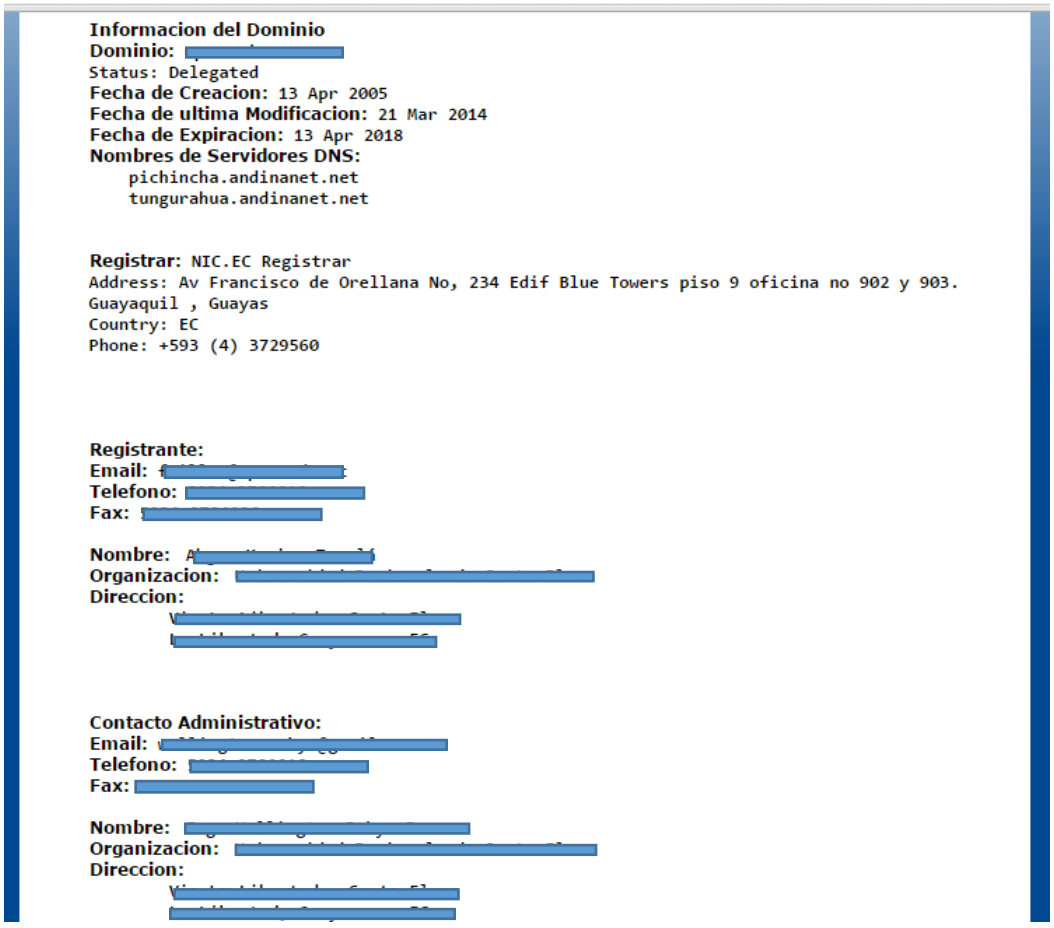


Figura 2.4 Información obtenida del nic.ec

González, P (2013) no expresa en su libro: “Whois es un protocolo TCP basado en petición/respuesta utilizado para efectuar consultas a una base de datos permitiendo determinar el propietario de un nombre de dominio o dirección IP pública” [3]

Vemos que información podemos obtener en el NIC.EC como nombre de la persona registrante, correo, números de teléfonos, la figura 2.4 nos muestra los resultados obtenidos.

Procedemos a la recolección de los correos electrónicos por medio de la herramienta theHarvester, los mismos que podrían servirnos más adelante si se considera necesario realizar un ataque de ingeniería social. Ver figura 2.5

Figura 2.5 Recolección de correos electrónicos institucionales

Realizamos footprinting por medio del navegador, primero visitando el dominio evaluado en donde encontramos una serie de enlaces a aplicaciones y a tutoriales de uso de las mismas.

Figura 2.6 Página con acceso a aplicaciones

Revisamos que información podemos encontrar en la web y en sus manuales sobre las aplicaciones.

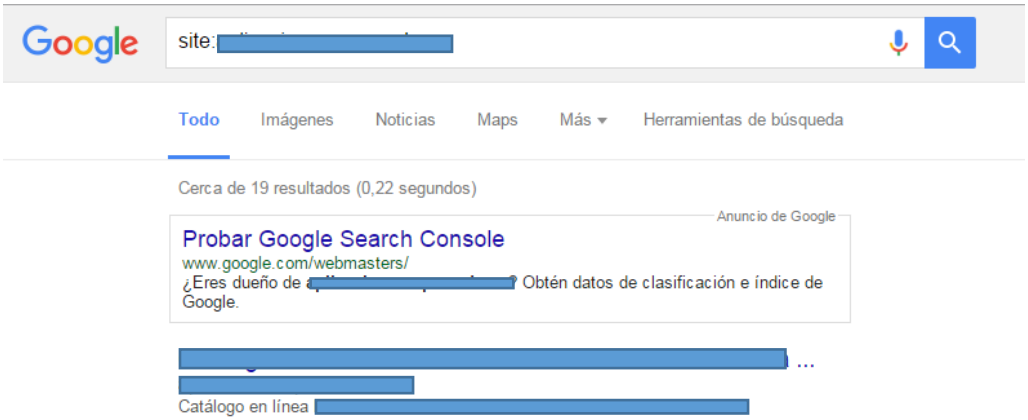


Figura 2.7 Búsqueda avanzada en google sobre objetivo

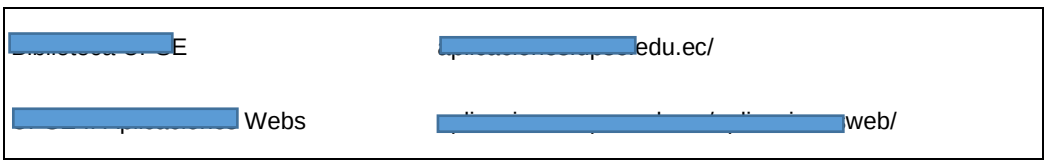
González, P (2014) sobre el uso de motores de búsqueda: “Los motores de búsqueda aportan gran cantidad de información a la humanidad, aunque en algunas ocasiones se pueden realizar búsquedas con un toque malicioso” [4].

Podemos recolectar mucha información sobre los subdominios que tiene la institución a auditar para proceder a identificar si están trabajando sobre un mismo servidor o en distintos, así como llegar a encontrar sus respectivos contactos a personal que nos puede ser de gran ayuda en fases posteriores, así mismo podemos encontrar paneles de ingreso a aplicaciones que podrían ser atractivas para atacantes maliciosos y una gran brecha de seguridad si es que no están debidamente protegidas.

Muchas veces los administradores de sitios y de servidores dejan directorios sensibles sin la debida protección en los cuales se pueden encontrar una serie archivos que nos podrían dar muy buena información del objetivo analizado.

En la tabla 2.1 presentamos todos los subdirectorios encontrados en el servidor evaluado.

Tabla 2.1 Directorios encontrados en el equipo auditado




Webs
cion.php
t.php
s/menu/inicio.php
hp
.php
.php
?cdn=objetivo
hp?cdn=vision
NQ==
==
==
Q==
e
Idiomas
==

En el footprinting realizado encontramos un manual para el uso de directores de la institución, en el cual se indica cual es el usuario y la clave para ingresar a los sistemas como se puede ver la figura 2.8 - 2.9

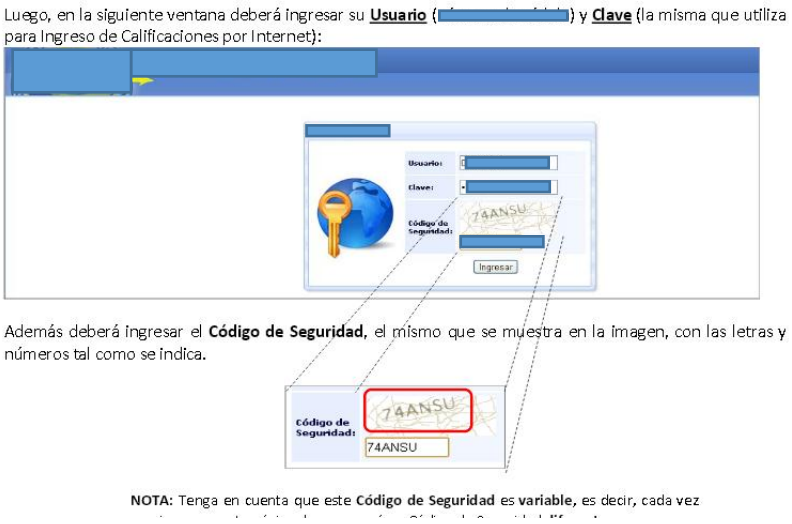


Figura 2.8 Manual de los sistemas en la búsqueda avanzada

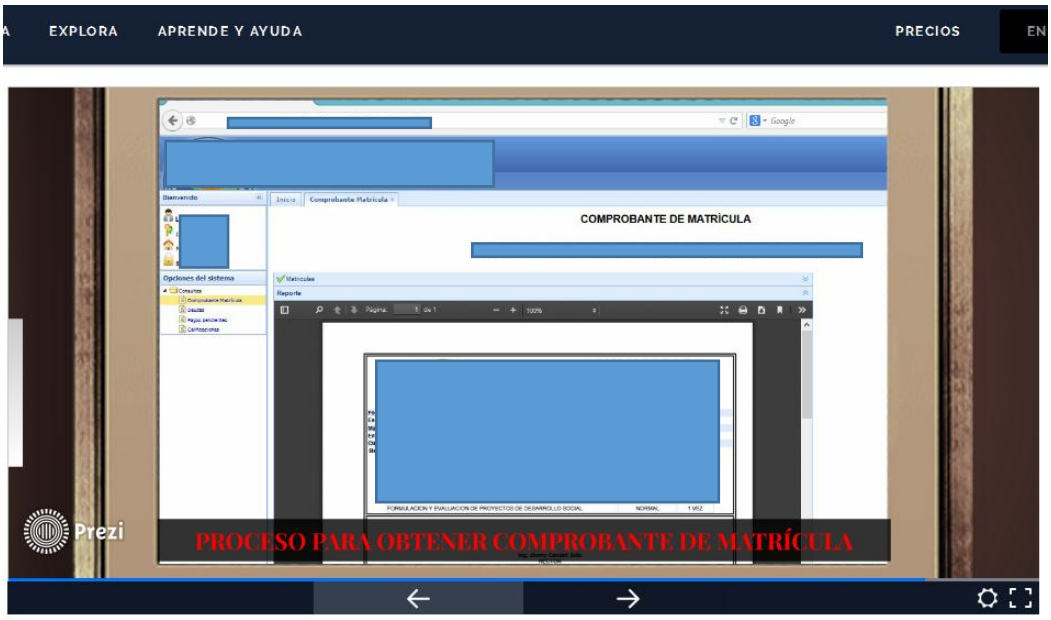


Figura 2.9 Manual en el que se deja en evidencia credenciales de acceso

Generando un código de error identificamos que la aplicación corre sobre un servidor Centos y en Apache 2.3.3. Ver figura 2.10

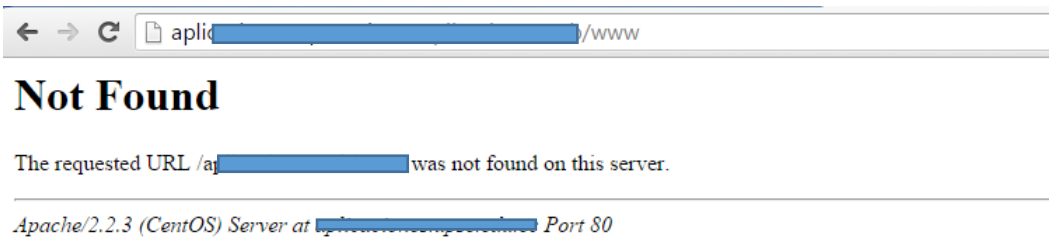


Figura 2.10 Error 404 en el server auditado

Por medio de Webscarab que es un framework escrito en Java por lo que podemos correrlo en varias plataformas procedemos a interactuar un poco con la aplicación web analizando los métodos de envío que emplea la aplicación.

Pauli, J (2013) acerca de spidering: “es la acción de indexación de todos los recursos de una aplicación web y catalogación para su uso futuro por medio de webscarab por toda la aplicación web” [5].

Habiendo corrido la herramienta como proxy entre nuestro navegador y la aplicación podemos ver los métodos de envío utilizados así como la estructura de directorios y archivos ya que corrimos el spidering a la url. Figura 2.11

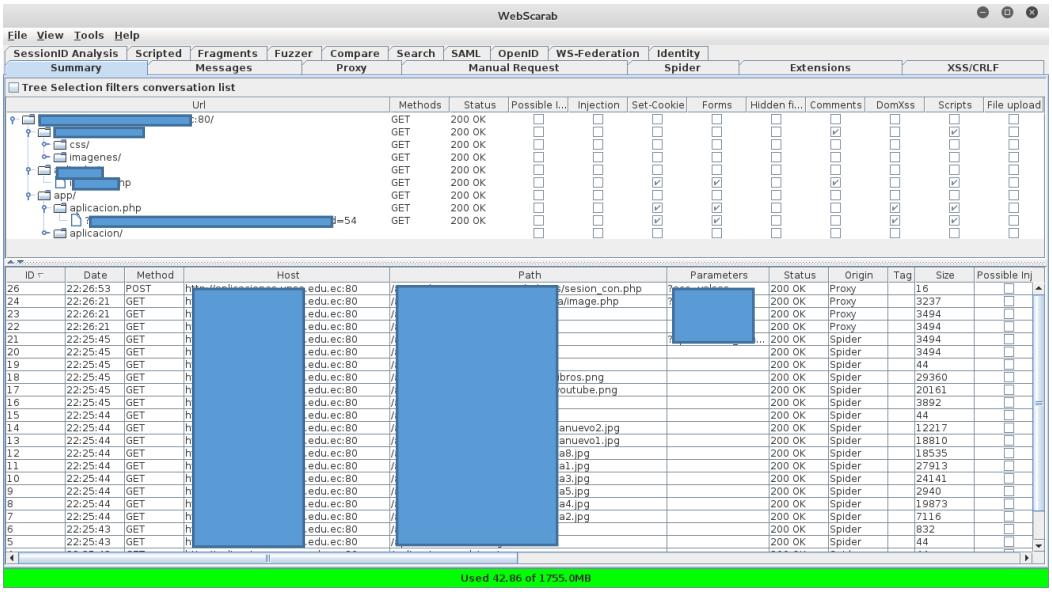


Figura 2.11 Análisis con herramienta Webscarab

Siguiendo con la revisión de las aplicaciones y analizando los códigos de error que podemos generar nos encontramos que la base de datos que se utiliza es Microsoft SQL Server

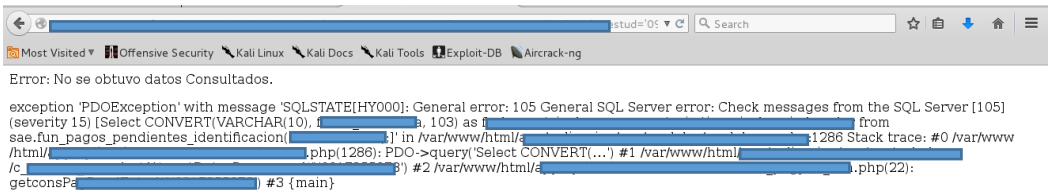


Figura 2.12 Error Microsoft SQL Server

Encontramos también el acceso a una intranet que no tiene códigos de verificación (Captcha) donde se podría más adelante aplicar un ataque de fuerza bruta.

Figura 2.13 Intranet de la institución

2.2. ESCANEO

González, P. (2014) sobre escaneo podemos apreciar: “En los sistemas, cada puerto que se encuentra abierto da una vía de explotación al auditor, por lo que esta información es muy valorada en esta fase” [4]

En la fase de escaneo hemos identificado un host que aloja las aplicaciones web de la institución y realizamos la identificación de servicios que se encuentren corriendo a espera de peticiones en el server para la identificación de aplicaciones que estén corriendo en un puerto distinto al 80, vea figura 2.14

Lo realizamos mediante Nmap de la siguiente manera:

nmap -P0 -sT -sV -p1-65535 XXX.XXX.XXX.XXX

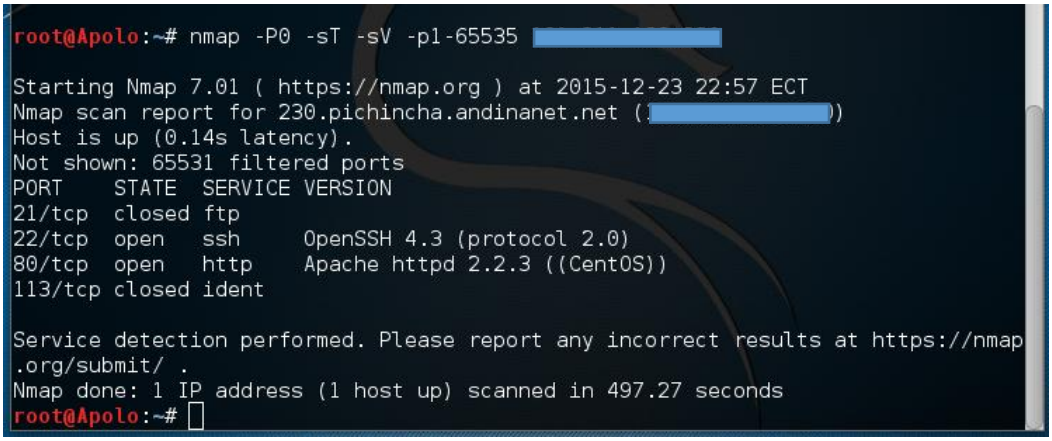


Figura 2.14 Análisis de servicios con Nmap

- Tenemos el Servidor Apache httpd 2.2.3 corriendo en el puerto 80
- Además del servicio OpenSSH 4.3 en el puerto por defecto
- Puerto 21 y 113 cerrados

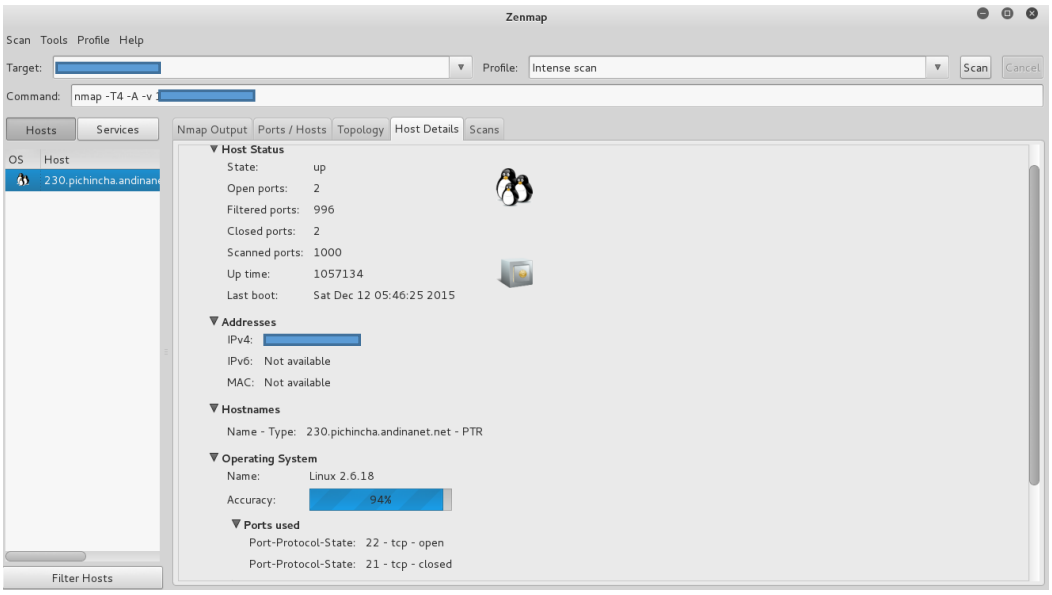


Figura 2.15 Análisis de servicios con Zenmap

Podemos apreciar que se ha realizado un trabajo tolerable en la seguridad del host como tal, lo que nos hace que avancemos a la siguiente etapa.

2.3. ENUMERACIÓN

La enumeración considerada también una subfase del escaneo nos permite la identificación mayor información sobre el objetivo.

Procedemos por medio de la herramienta DirBuster a enumerar los directorios que tiene el sitio de aplicaciones, pero para esto hemos generado un diccionario un poco más pequeño con la herramienta CEWL que nos permite personalizar en cierta forma como vamos a hacer uso de la herramienta DirBuster, ver figuras 2.16 – 2.17

Figura 2.16 Generación de diccionario Cewl

Por medio de la herramienta Dirbuster hemos encontrado la estructura del sitio de aplicaciones y podemos deducir que el mismo server es usado tanto para desarrollo como producción, por la forma en que se encuentran distribuidos los directorios.

No se han encontrado directorios con información sensible pero la simple estructura ya nos ayuda a entender la forma de programación que tienen los desarrolladores del sitio.

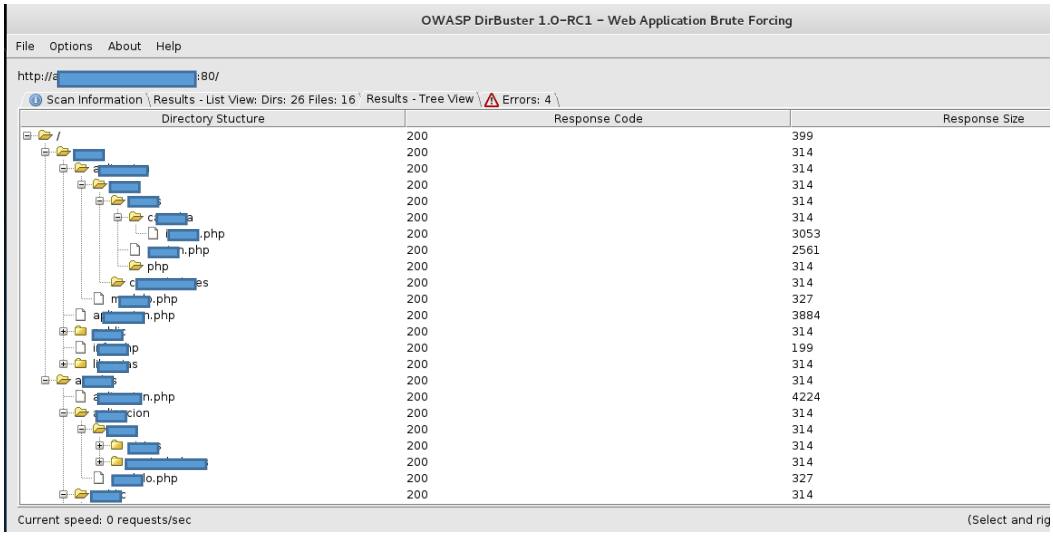


Figura 2.17 Análisis OWASP DirBuster

Automatizando ahora nuestro proceso hacemos uso de una herramienta que por su fácil manejo es utilizada muy a menudo en trabajos de auditoria y seguridad web, w3af, la misma que nos confirma la información encontrada anteriormente y nos arroja un detalle de las vulnerabilidades que posee el objetivo evaluado, figura 2.18

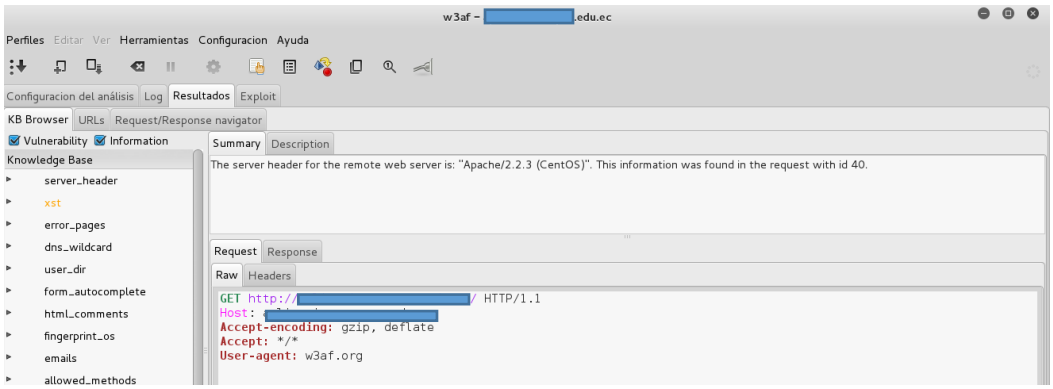


Figura 2.18 Identificación de vulnerabilidades con W3af

Resumiendo un poco los que nos presenta la herramienta en cuanto a vulnerabilidades encontradas son:

Figura 2.19 Vulnerabilidades encontradas con W3af

2.4. ANÁLISIS DE VULNERABILIDADES Y EXPLOTACIÓN

Una vez que hemos recopilado la información del objetivo auditado, como urls, directorios, usuarios, contraseñas y vulnerabilidades, analizamos cuál de ellas nos podría servir como un vector de ataque efectivo para conseguir comprometer el equipo o las aplicaciones.

XST Cross Site Tracing

Vulnerabilidad que viene del XSS la misma que es generada por el método HTTP TRACE, tener este método habilitado podría ser perjudicial en cierto modo, esto podría permitir el acceso a las cookies por medio del navegador, por lo general el método TRACE se emplea con fines de depuración.

TRACE tiene como función hacer debug del protocolo HTTP, la herramienta W3af nos da una alerta de que el método está habilitado. Figura 2.20

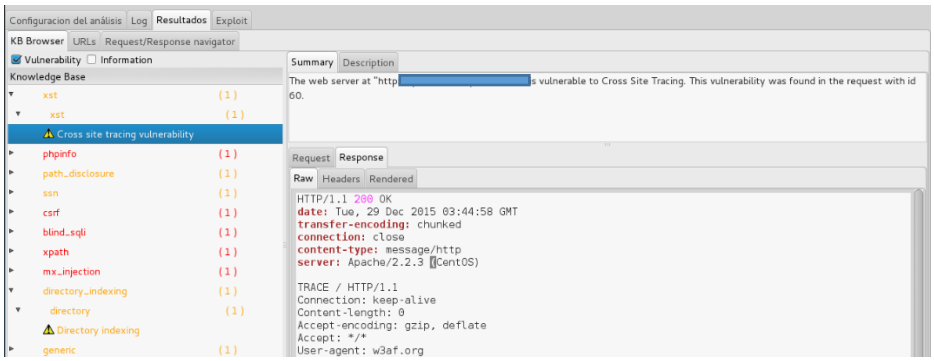


Figura 2.20 Vulnerabilidad XST

Comprobamos esto mediante el uso del comando #nc ip_auditada puerto y nos damos cuenta que efectivamente el método trace se encuentra habilitado

Figura 2.21 Método TRACE habilitado

CSRF – Cross Site Request Forgery

Este tipo de vulnerabilidad consiste en la falsificación de datos en sitios cruzados, permite que el atacante haga que la víctima ejecute código malicioso en su navegador sin que este se dé cuenta.

Veamos la figura 2.22 en el que se explica cómo funciona un ataque a este tipo de vulnerabilidad.

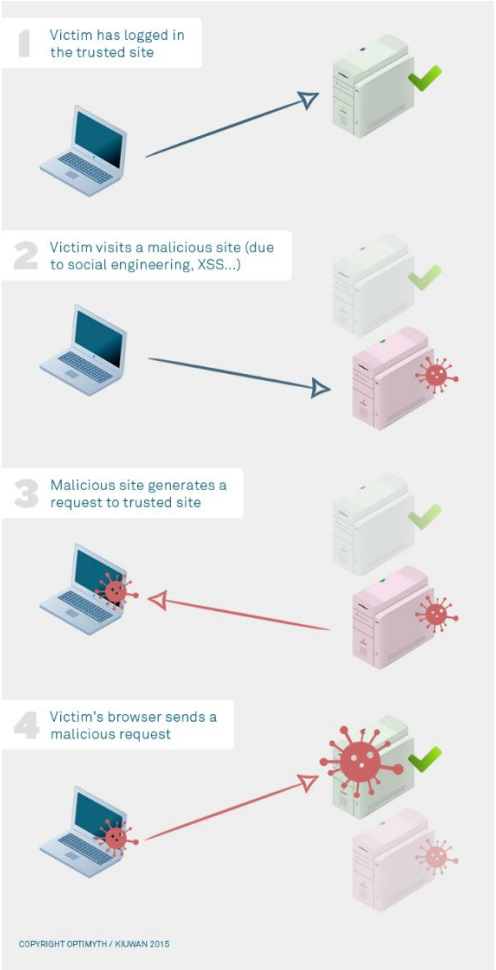


Figura 2.22 Vulnerabilidad CSRF

La aplicación es vulnerable a este tipo de ataques ya que no existe ningún mecanismo para confirmar que una petición o ejecución de alguna acción hecha por un usuario ha sido enviada intencionalmente por este, la aplicación cuenta con un formulario en el que se encuentra la vulnerabilidad, figura 2.23

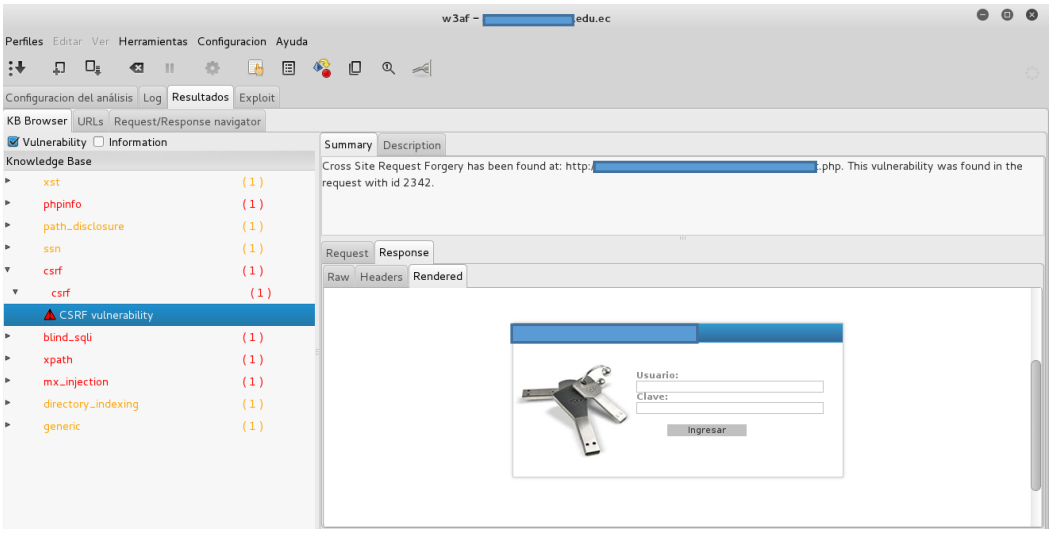


Figura 2.23 Vulnerabilidad CSRF

Blind SQL injection vulnerability

Este tipo de vulnerabilidad nos permite realizar un Blind SQLi o Ataque a ciegas por SQL, la diferencia entre el SQL injection y el SQLi es que en este caso el sitio vulnerable no muestra mensajes de error, por lo que es necesario realizar la comprobación por medio de verdaderos o falsos.

Las sentencias que siempre van a ser correctas “Or 1 = 1” o “having 1=1”.

Este tipo de vulnerabilidad es muy sencilla de explotar pero es algo tediosa, ya que es necesaria una gran cantidad de peticiones a la aplicación auditada.

La herramienta w3af nos muestra que la aplicación es vulnerable a este tipo de ataques. Figura 2.24

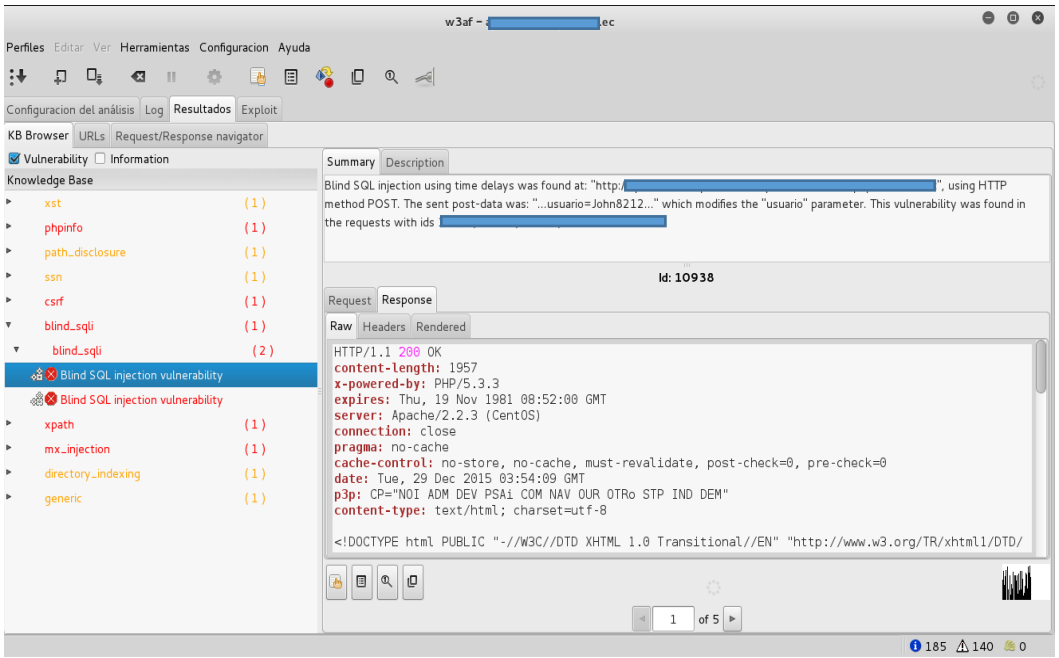


Figura 2.24 Vulnerabilidad Blind SQL

Directory indexing

Este tipo de vulnerabilidad se da cuando los servidores permiten listar directorios que podrían contener información sensible que no debería ser accedida por el público en general.

A menudo los Administradores se crean una falsa sensación de seguridad contra este tipo de vulnerabilidades argumentando que estos directorios no se encuentran enlazados por lo que no podrían ser accesibles, cosa que no es cierta ya que quedan en evidencia cuando se utilizan potentes escáneres de vulnerabilidades, figura 2.25.

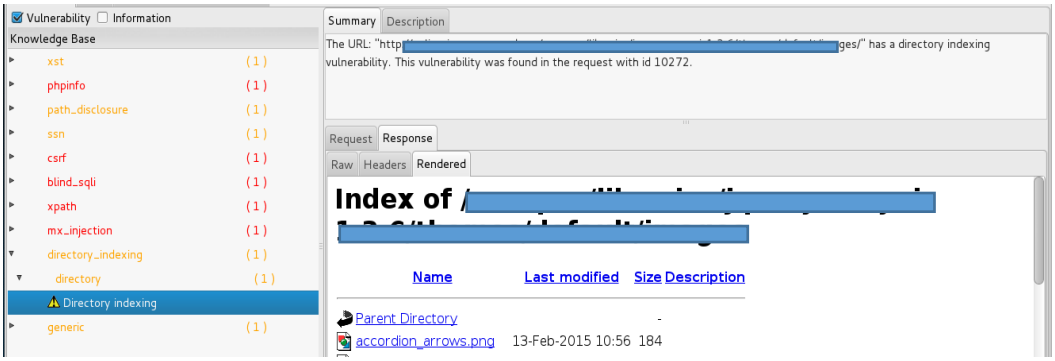


Figura 2.25 Vulnerabilidad directory indexing

Para nuestro Hackeo Ético vamos a centrarnos en la vulnerabilidad de Blind SQLi, haciendo uso de los resultados que nos dio nuestra herramienta W3af

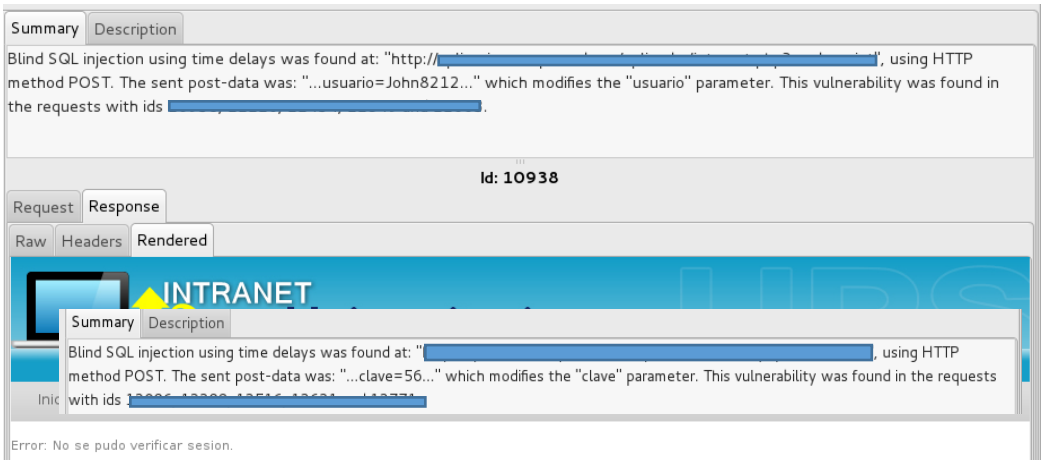


Figura 2.26 Vulnerabilidades Blind SQLi

Nos damos cuenta que el método de envío es POST por lo que será necesario el uso de herramientas como sqlmap, sabemos que es Microsoft SQL Server por el error obtenido en la recopilación de información.

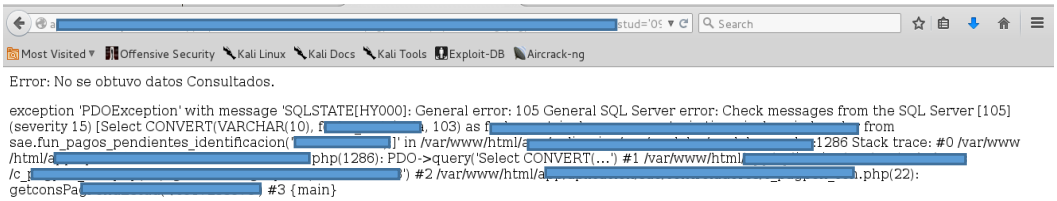


Figura 2.27 Error que indica motor de base de datos

Entonces los parámetros a enviar en la consulta serán:

sqlmap –url "[redacted]" --dbms="Microsoft SQL Server" --level 5 --risk 3 --dbs -p usuario --data "[redacted]"

Especificación de los parámetros utilizados:

- url: [redacted]
- dbms: La base de datos que utiliza la aplicación - Microsoft SQL Server
- level: nivel de dificultad, según otros escaneos realizados recomendado el valor de 5
- risk: riesgo de pruebas que se desea realizar – 3 máximos
- dbs: indicamos que extraiga las bases
- p : variable vulnerable

- data: cadena que se envía mediante método POST concatenada de la siguiente forma: “[redacted]”

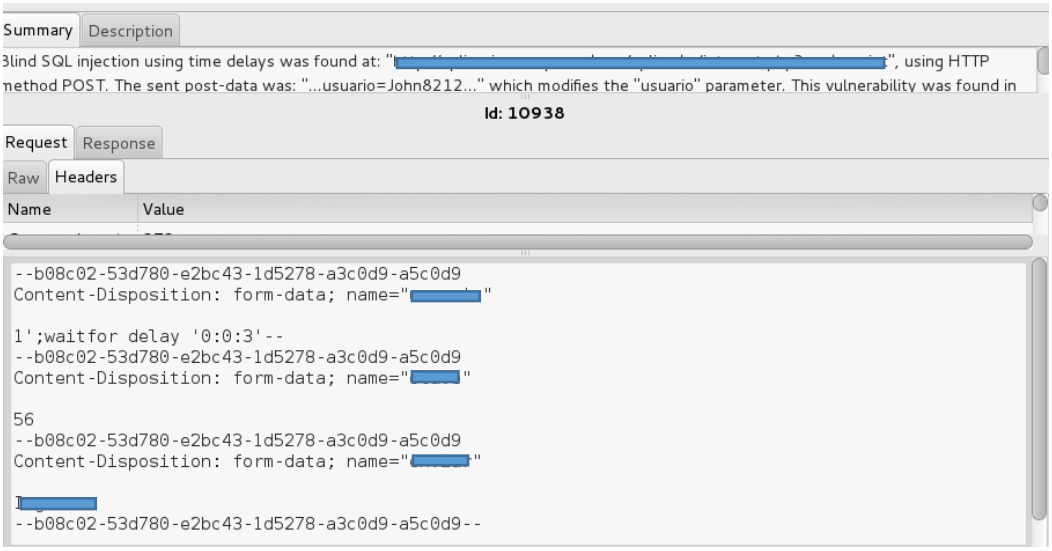


Figura 2.28 Cadena que se envía por método POST

La explotación de la vulnerabilidad ha sido positiva obtenemos las bases que se encuentran alojadas en el servidor, figura 2.29

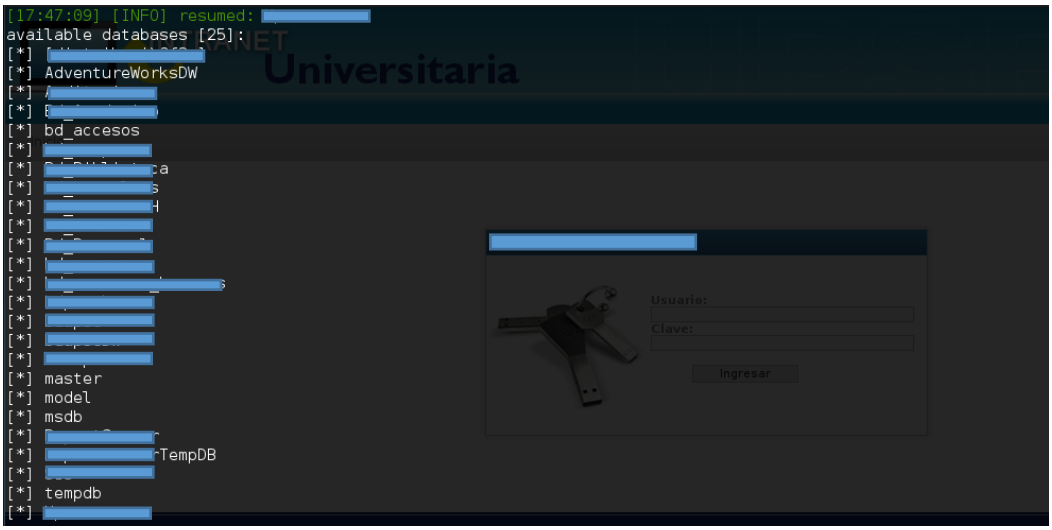


Figura 2.29 Bases de datos de la institución

De las bases que nos muestran nos llama la atención una que deducimos tiene usuarios y contraseñas que es la bd_accesos por lo que corremos el siguiente comando en sqlmap tratando de obtener las tablas

root@Apolo:~# sqlmap --url "[redacted]" --dbms="Microsoft SQL Server" --level 5 --risk 3 --dbs -p usuario --data "[redacted]" -D bd_accesos –table

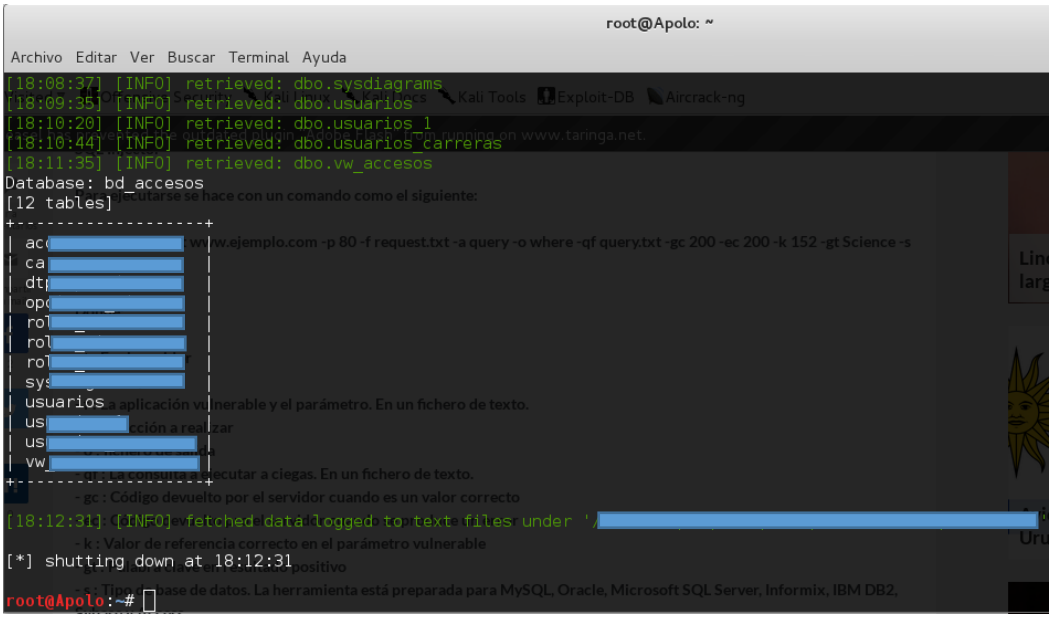


Figura 2.30 Tablas obtenidas con sqlmap

root@Apolo:~# sqlmap --url "[redacted]" --dbms="Microsoft SQL Server" --level 5 --risk 3 --dbs -p usuario --data "[redacted]" -D bd_accesos -T usuarios --columns

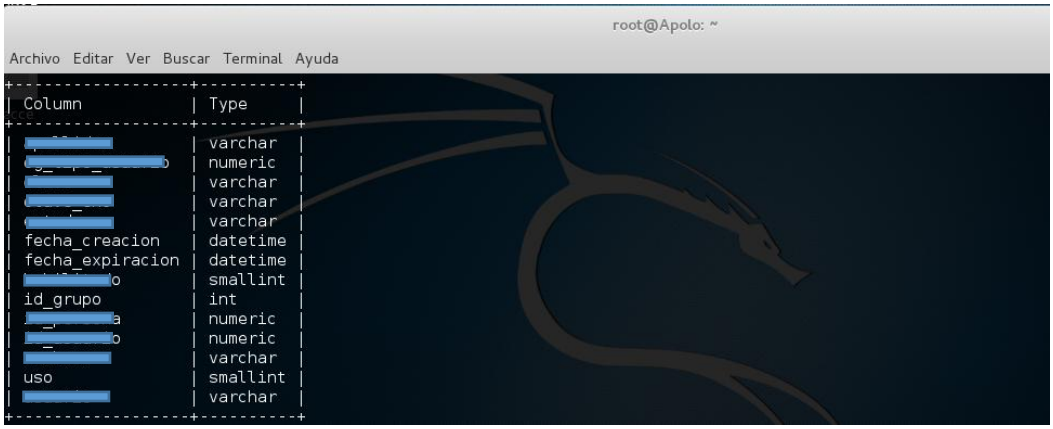


Figura 2.31 Columnas obtenidas con sqlmap

Una vez identificada la base que nos interesa, en este caso la que nos llamó la atención por el nombre peculiar fue bd_accesos, figura 2.30, hemos obtenido las tablas y las columnas de una de ellas procedemos a realizar la extracción de la data mediante el comando siguiente:

root@Apolo:~# sqlmap --url "[redacted]" --dbms="Microsoft SQL Server" --level 5 --risk 3 --dbs -p usuario --data "[redacted]" -D bd_accesos -T usuarios --columns –dump

Figura 2.32 Extracción de la data

De los datos obtenidos nos podemos percatar primero en los campos que tiene esta tabla, nos indica además de tener un id_usuario, también aparece un campo clave y otro clave_enc, figura 2.33.

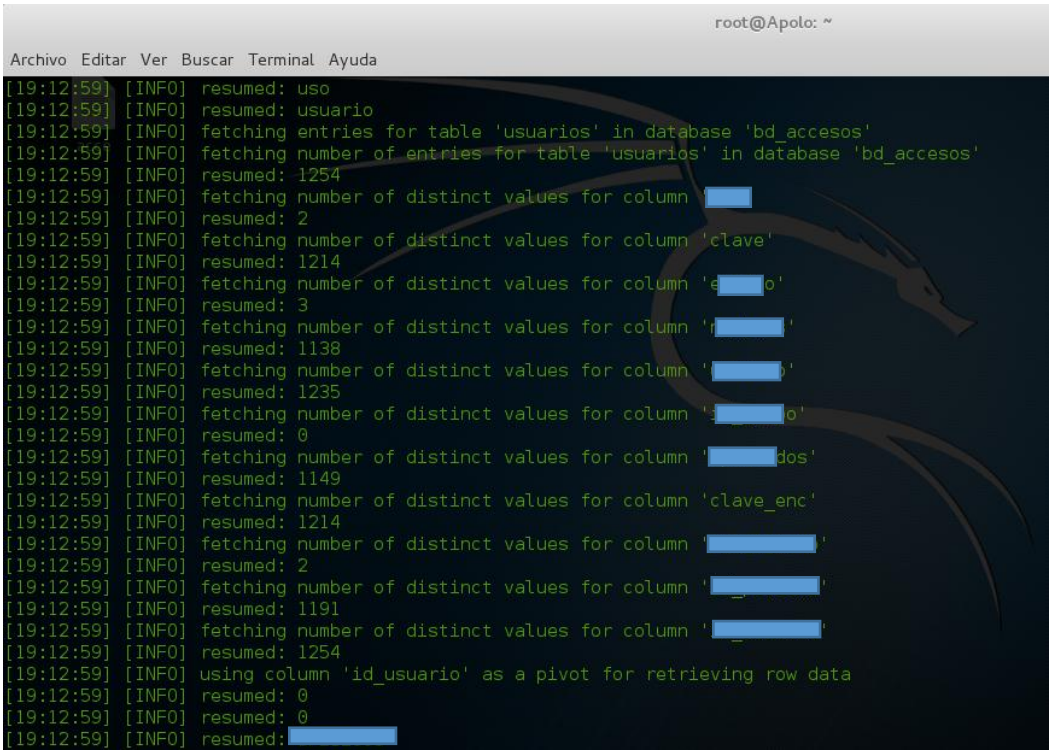


Figura 2.33 campos extraídos con sqlmap

Hemos podido conseguir datos sensibles de las bases, datos como usuarios y claves, por lo que un delincuente informático podría hacer mal uso de toda esta data.

De los datos obtenidos en la etapa inicial encontramos en los manuales la forma de acceso a las aplicaciones en donde nos explica el usuario y contraseña que debemos utilizar, ver figura 2.34.

- Número [redacted]
- Número de [redacted]

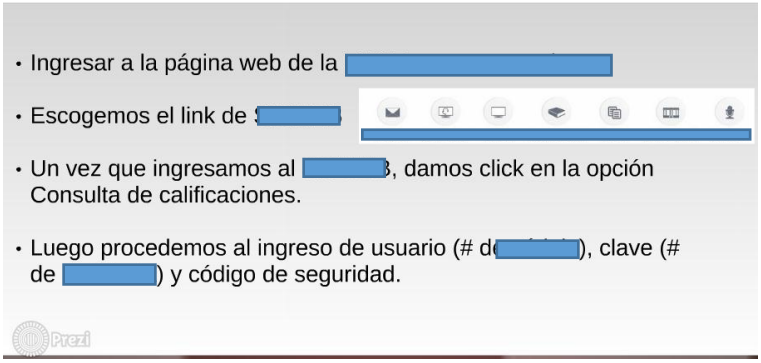


Figura 2.34 Datos de acceso encontrados en línea

En el mismo manual encontramos que en la pantalla se pueden distinguir estos dos datos que nos permiten el acceso

- Usuario1: [redacted]00
- Clave1: [redacted]3

Figura 2.35 Usuario y clave expuesto en línea

Permitiendo así el acceso a la aplicación, pero un acceso un poco limitado con solo dos opciones, un sistema de encuestas y un sistema de consultas

Figura 2.36 Ingreso al sistema

De los mismos manuales pudimos obtener el otro usuario, pero este nos pareció un poco más interesante ya que es un usuario con privilegios un poco más elevados, figura 2.37.

- Usuario2: W[redacted]o
- Dato2: [redacted]2
- Clave2:

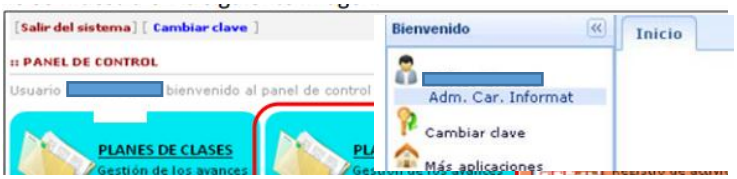


Figura 2.37 Datos de un usuario docente expuesto

Hasta aquí hemos conseguido el nombre del docente y el usuario, ahora del ataque de inyección de SQL pudimos notar tres cosas interesantes, primero que encontramos un usuario con el mismo nombre al del manual antes encontrado, segundo que la base nos dio un campo con la clave en un MD5 y que la misma tabla tenía un campo que se llamaba clave_enc

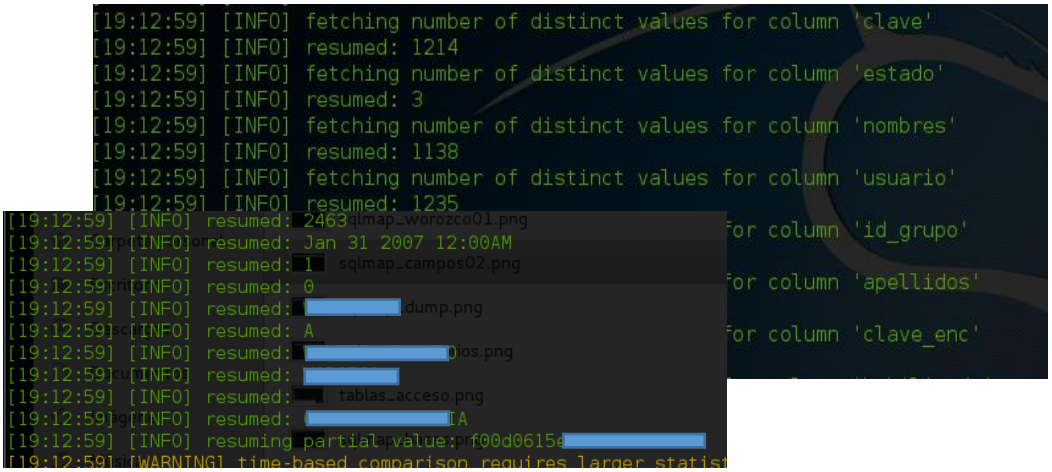


Figura 2.38 Extracción de datos con sqlmap

Decidimos hacer una prueba y generamos primero el MD5 de Administrador no encontramos coincidencia, luego el de [redacted] el cual nos coincidió con el MD5 arrojado por la base, por lo que hicimos la misma prueba con el dato [redacted] el mismo que coincidió, lo que nos deja en evidencia que se está guardando la clave sin encriptar en la tabla de acceso, figura 2.39.

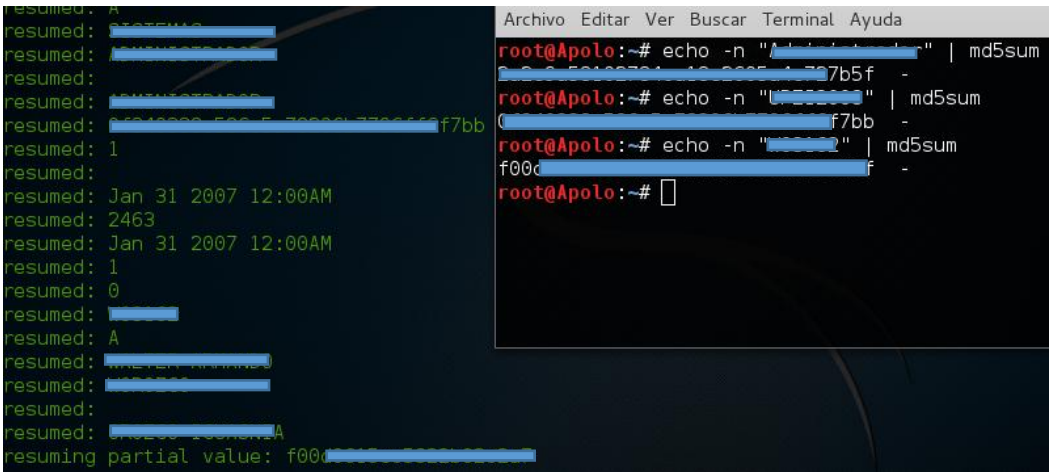


Figura 2.39 Generación de comprobación MD5

Procedemos a probar el ingreso con los datos encontrados y obtenemos acceso a una serie de aplicativos con un nivel de rol más elevado como podemos apreciar en la figura 2.40.

Figura 2.40 Acceso a sistemas con más privilegios

CAPÍTULO 3

ANÁLISIS DE RESULTADOS

3.1. REPORTE

Resumen Ejecutivo

La presente sección abarca los detalles de la evaluación realizada a la plataforma de aplicaciones de la institución por medio de un Hacking Ético el mismo que tiene como propósito la revisión de la seguridad frente a posibles ataques de forma externa.

Todas las pruebas fueron ejecutadas con consentimiento de la dirección y se simuló ser un delincuente informático en busca de puntos de acceso a la plataforma.

Se determinará:

- Debilidades en la configuración de la plataforma que aloja las aplicaciones web
- Puntos de acceso para delincuentes informáticos y el impacto en caso de conseguir dicho acceso
- Confidencialidad de los datos de la institución

Se prestó un enfoque más profundo a la evaluación y explotación de las vulnerabilidades que podrían permitir el acceso a un delincuente informático a la plataforma poniendo en riesgo la integridad de la institución

Línea de tiempo

Tabla 3.1 Tiempo de ejecución del Hacking Ético

Test de penetración	Fecha de inicio	Fecha de Finalización
Prueba No. 1	22/12/2015	02/01/2016

3.2. RESUMEN DE HALLAZGOS

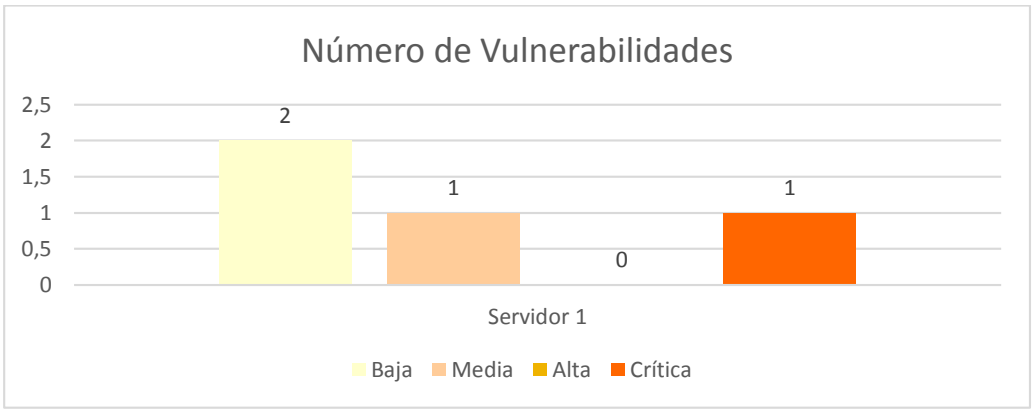


Figura 3.1 Vulnerabilidades encontradas

Tabla 3.2 Puertos encontrados en el servidor

Dirección IP	Tipo de sistema	Sistema Operativo	Puertos abiertos		
			Puerto#	Protocolo	Servicio
1 0	Servidor de aplicaciones	Centos Apache 2.2.3	21	TCP	FTP Closed
			22	TCP	SSH
			80	TCP	HTTP
			113	TCP	Auth Closed

Clasificación de los riesgos

Según el test de penetración realizado los riesgos que posee la institución evaluada se consideran de alto riego, los ataques a las aplicaciones tendrán un alto impacto, el atacante podrá extraer la data, modificarla e incluso borrarla o hacer uso de las mismas para redirigir tráfico a otros sitios web.

Tabla 3.3 Detalle vulnerabilidad XST

Cross site tracing XST
Clasificación: BAJO **Sistema afectado:** Aplicación **Descripción:** Se tienen el método TRACE habilitado, se podrían saltar la protección HTTPOnly **Impacto:** La data de los usuarios podría ser redirigida o robo de cuentas de autenticación **Remediación:** Los navegadores modernos ahora impiden se realicen peticiones al método TRACE, establece la directiva TraceEnable en “off” en el archivo de configuración

Tabla 3.4 Detalle vulnerabilidad CSRF

Falsificación de Peticiones de Sitios Cruzados – CSRF
Clasificación: MEDIO **Sistema afectado:** Intranet **Descripción:** Falsificación de peticiones en sitios cruzados, cuando no existe ningún mecanismo de comprobación de acciones del usuario en un sitio **Impacto:** El delincuente informático podrá ejecutar un proceso determinado a través de una sesión abierta legítimamente mediante otra web infectada. **Remediación:** Uso de mecanismos de verificación en las peticiones al servidor, permitiendo la validación de los valores asignados a un formulario en una sesión.

Tabla 3.5 Detalle vulnerabilidad BlindSQL

Blind SQL injection vulnerability
Clasificación: CRÍTICO **Sistema afectado:** Aplicaciones **Descripción:** Ataque a ciegas por inyección SQL **Impacto:** El delincuente informático podrá acceder a la data de la institución, modificarla e incluso borrarla **Remediación:** Mantener los datos no confiables de forma separada de los comandos y consultas

Tabla 3.6 Detalle vulnerabilidad Directory indexing

Directory indexing
Clasificación: BAJO **Sistema afectado:** Aplicación **Descripción:** La web permite listar directorios con información que debería estar de forma oculta para los usuarios **Impacto:** Permite la visualización de los delincuentes informáticos de archivos que podrían contener información sensible de la institución **Remediación:** Mecanismo que no permitan la visualización del contenido de los directorios, index en blanco.

DETALLES DE HALLAZGOS

Realizando los primero pasos del reconocimiento se pudieron determinar los diferentes puntos de entrada a las aplicaciones, también se determinó que en manuales de sistemas en línea, se encuentran dos usuarios para el acceso a la plataforma así como también un password.

Se determinó el mapa del sitio por medio de generación de diccionarios personalizados mediante la web de la institución y un ataque de fuerza bruta.

Un análisis más profundo permitió determinar vulnerabilidades de las cuales la más crítica es Blind SLQi la cual permitió ver y manipular la

información de la institución, permitiendo la explotación de esta, dándonos el acceso a las aplicaciones como un usuario de tipo docente permitiendo realizar diferentes cambios.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1.- No se cuenta con mecanismo que mitiguen en algo la recolección de información en un proceso de reconocimiento.

2.- Las actuales políticas de asignación y cambio de contraseñas no se encuentran correctamente elaboradas, por lo que crean un agujero de seguridad que ayuda a la explotación de las vulnerabilidades encontradas.

3.- El almacenamiento de las contraseñas debidamente encriptadas se está realizando de una manera errónea por lo que el atacante puede deducir fácilmente los accesos a los sistemas de la institución.

4.- Las políticas o reglas básicas para desarrollo seguro de software y sistemas necesita revisión ya que se están dejando brechas de seguridad directamente en las aplicaciones.

5.- La institución cuenta con vulnerabilidades críticas que si son explotadas por delincuentes informáticos pueden llegar a comprometer de forma drástica la confidencialidad, integridad y disponibilidad.

6.- El hackeo ético realizado con el consentimiento de la institución tuvo éxito pudiendo penetrar en los sistemas y en las bases, lo que dejaría a un delincuente la posibilidad de tomar control de los sistemas y si así lo quisiera de la infraestructura interna.

7.- Las pruebas realizadas en ningún momento afectaron el correcto funcionamiento ni paralización de los aplicativos en línea.

RECOMENDACIONES

1.- Se recomienda hacer uso de servicios de pagos para la no divulgación de información en cuanto a registros de dominios y mitigar en algo el acceso a la información en la fase de reconocimiento durante un ataque.

2.- El equipo que aloja las aplicaciones de producción y las que se encuentran en desarrollo deberían estar por separados y realizar la debida segmentación de la red para poder ponerlos detrás de una DMZ.

3.- Se debe de hacer las correcciones necesarias en las bases en cuanto al almacenamiento de las claves y lanzar una campaña masiva obligando al usuario realizar el cambio de sus credenciales de acceso.

4.- Efectuar el estudio de métodos y herramientas para el desarrollo seguro de aplicaciones e implementar otros métodos validación de acceso.

5.- Tomar acciones correctivas en las vulnerabilidades encontradas y monitorear constantemente la red en busca de comportamientos no comunes para evitar el escaneo no autorizado de vulnerabilidades.

6.- Realizar charlas y capacitación para que la comunidad de la institución tome conciencia de los peligros a que se encuentran expuestos en la red y el problema que acarrea la divulgación de información sensible.

BIBLIOGRAFÍA

[1] Calles J., La Biblia del Footprinting, 2011.

[2] Astudillo K., Hacking Ético 101, 2013.

[3] González P., Germán S. y Soriano J., Pentesting con Kali, Madrid: 0xWORD, 2013.

[4] González P., Ethical Hacking, Madrid: 0xWORD, 2014.

[5] Pauli J., The basics of web hacking, 2013.