UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Instalación y evaluación de vulnerabilidades del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas

Trabajo de graduación presentado por David Ytzen Hsieh Lo
para optar al grado académico de Licenciado en
Ingeniería en Ciencias de la Computación

Guatemala
2015

Instalación y evaluación de vulnerabilidades del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Instalación y evaluación de vulnerabilidades del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas

Trabajo de graduación presentado por David Ytzen Hsieh Lo
para optar al grado académico de Licenciado en
Ingeniería en Ciencias de la Computación

Guatemala
2015

Vo. Bo.:

(f) _____

Ing. Eddy López

Tribunal:

(f) _____

Ing. Douglas Barrios

(f) _____

Ing. Eddy López

(f) _____

Ing. Eduardo Castellanos

Fecha de aprobación: Guatemala, 05 de febrero de 2015.

PREFACIO

Este trabajo graduación surge de la necesidad de realizar la entrega a la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID) el Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas, el proyecto que fue desarrollado para facilitar la medición del cumplimiento de la Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas.

Apoyando la iniciativa y el objetivo por la cuál fue desarrollado el proyecto entregado, surge la idea de realizar una evaluación de seguridad para identificar la mayor cantidad de vulnerabilidades posibles que pueda tener la información sensible que se espera manejar en ella.

El trabajo es producto de la aplicación de conocimientos adquiridos a lo largo de la carrera, técnicas y conceptos de diversos cursos relacionados con la informática, y la investigación de vulnerabilidades, las causas y las técnicas para detectarlas y corregirlas.

ÍNDICE

Página

LISTA DE FIGURAS

Página

RESUMEN

El objetivo del presente trabajo graduación consiste en la instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas, el cual fue desarrollado como producto del Megraproyecto “Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas“ que dio inicio en el segundo ciclo del año 2011.

Con el sistema instalado en el servidor de la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID), se lleva cabo una evaluación de seguridad sobre la misma. Esta evaluación consiste en realizar un conjunto de pruebas sobre el sistema para identificar la mayor cantidad de fallas o debilidades posibles que pueda tener.

Todas las pruebas de la evaluación de seguridad fueron realizadas remotamente sobre el sistema que fue instalado sobre el servidor ubicado en las oficinas de SECCATID.

I. Introducción

El Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas es un sistema desarrollado para medir el cumplimiento de la Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas por medio de los proyectos que se llevan a cabo dentro de la esta entidad y las instituciones relacionadas con la misma.

Tras el desarrollo de dicho sistema de parte de un grupo de alumnos de la Universidad del Valle de Guatemala, se encuentra la necesidad de llevar a cabo la instalación de la misma en el servidor de las oficinas de la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID).

Siendo el proyecto un sistema que maneja información sensible, se ve la necesidad de llevar a cabo una evaluación de seguridad, para encontrar la mayor cantidad de vulnerabilidades posibles antes de que se encuentre en producción.

La evaluación de seguridad es llevada a cabo con la metodología que presenta el Proyecto de Pruebas OWASP versión 3. Esta metodología fue elegida debido a que el Proyecto de Pruebas OWASP es un proyecto abierto que fue elaborado en conjunto por un grupo de profesionales en el área de seguridad informática.

II. Objetivos

A. Objetivo general

Encontrar vulnerabilidades y evaluar del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas instalado en la infraestructura de la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas.

B. Objetivos específicos

1. Instalar cada uno de los módulos del sistema sobre la infraestructura proveída.
2. Encontrar vulnerabilidades mediante pruebas de seguridad.
3. Mitigar las vulnerabilidades encontradas.

III. Marco teórico

A. Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas

1. SECCATID. La Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID), por instrucciones de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas, CCATID, coordinó la formulación de la Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas.

SECCATID tiene como objetivo coordinar junto a otras instituciones públicas y privadas proyectos relacionados al consumo, la adicción y el tráfico ilícito de drogas.

2. Estructura. La Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, se encuentra estructurada a partir de principios por los que la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID). Por su objetivo, la política se clasifica en cinco ejes:

a. ***Económico social.*** Mejorar las condiciones de vida de la población, reduciendo los incentivos económicos que puede generar el tráfico ilícito de drogas.

b. ***Reducción de la oferta.*** Disminuir la producción, el tráfico y la distribución de drogas ilícitas, así también como la infraestructura que es utilizada para el este fin.

c. ***Reducción de la demanda.*** Reducir el consumo de drogas ilícitas y su uso indebido. Ofrecer alternativas de tratamiento, rehabilitación a farmacodependientes. Crear programas de prevención acorde a la realidad pluricultural del país.

d. ***Fortalecimiento jurídico e institucional.*** Crear más alianzas con instituciones para fortalecer los instrumentos jurídicos y operativos del Estado para combatir las organizaciones de narcotráfico.

e. ***Comunicación.*** Dar a conocer la importancia sobre la reducción de la oferta y la demanda de drogas ilícitas e lícitas a través de proyectos y programas, con mayor enfoque en la población infantil y juvenil.

B. Seguridad informática

1. Definición. Se puede definir seguridad como una necesidad de prevención de la vida y las posesiones. En el campo de la seguridad, el término es el mismo, pero aplicado específicamente a la información y datos manejados por sistemas informáticos.

2. Objetivo. El objetivo principal de la seguridad informática, consiste en mantener la integridad, disponibilidad, confidencialidad, y autenticidad de la información.

a. Integridad. Que la información o los datos no hayan sido alterados.

b. Disponibilidad. Que la información siempre se encuentra disponible cuando se necesite.

c. Confidencialidad. Que la información privada no sea vista por entidades no autorizadas.

d. Autenticidad. Que la información realmente sea originaria de la entidad que se dice ser.

C. CISSP

1. Definición. La certificación CISSP es un estándar reconocido globalmente que garantiza el conocimiento de un individuo en el área de seguridad informática.

2. Dominios.

a. Control de acceso. Consiste en aspectos de seguridad que controlan la interacción entre los usuarios y usuarios con otros sistemas y recursos. Protege la información, los recursos y el sistema de cualquier acceso no autorizado y pueden ser componentes que determinen el nivel de autorización que tiene un usuario luego de autenticarse.

El control de acceso es una de las primeras barreras de defensa que protege a un sistema de accesos no autorizados. El ingreso de usuario y contraseña por parte del usuario, y los permisos que tiene éste sobre recursos locales o en la red, según el grupo o rol al que pertenezca, forman parte del dominio de control de acceso.

Para que un usuario acceda a un recurso, se debe de comprobar que el

usuario es quien se dice ser (identificación)

b. ***Seguridad de red y telecomunicaciones.*** Cubre aspectos como la configuración de la red, los protocolos y servicios utilizados; los dispositivos de red; manejos de inoperatividad; instalación, configuración e interfaz de software y dispositivos de comunicación; y manejo de problemas de red.

c. ***Seguridad de la información y gestión de riesgos.*** En este dominio incluyen las políticas de seguridad, los procedimientos, los estándares, la clasificación de la información, la organización de seguridad, los análisis de riesgo, programas de seguridad y la educación de la seguridad.

Un análisis de riesgo identifica los bienes de la empresa, encuentra las amenazas que los podrían poner en riesgo, y estima la pérdida aceptable que la organización sería capaz de soportar.

d. ***Seguridad de aplicaciones.*** Este dominio abarca lo que son las aplicaciones. La seguridad, el diseño y la funcionalidad de las aplicaciones. Las vulnerabilidades que podrían tener en sus diferentes componentes: ingreso, procesamiento, procesos de comunicación, acceso, salida, interfaz a otros programas, etc.

e. ***Criptografía.*** Abarca todo relacionado con el almacenamiento y transmisión de manera de que los destinatarios sean los únicos que lo puedan leer o procesar. La criptografía es un método para proteger información almacenada o transmitida a través de un canal no confiable.

f. ***Arquitectura y diseño de seguridad.*** Se compone de dos conceptos fundamentales: La política de seguridad y el modelo de seguridad.

La política de seguridad es un documento que define cómo las entidades pueden acceder a la información, las operaciones que pueden ser realizadas, el nivel de protección requerida para el una aplicación y que acciones se deben de realizar en caso de que no se cumpla con alguno de los requerimientos.

El modelo de seguridad define los requerimientos necesarios para soportar e implementar una política de seguridad.

g. ***Seguridad de operaciones.*** Abarca todo lo relacionado a todo proceso realizado para mantener protegida la red, los sistemas de cómputo, las aplicaciones y los ambientes de trabajo. Incluye la auditoría, el monitoreo, los reportes, y el mantenimiento continuo.

h. ***Planes de continuidad del negocio y de recuperación frente a desastres.*** Consiste en métodos de minimización de efectos de desastres y definición de pasos necesarios para asegurar la disponibilidad de recursos, personal y procesos.

i. ***Legislación, regulaciones, cumplimiento de las mismas e investigación.*** Abarca toda la parte legal, que constituye las leyes y el crimen informático o las que pueden ser afectadas por la tecnología de la información.

j. ***Seguridad física.*** El robo, el fraude, el saboteo, el vandalismo y los accidentes son algunos de los riesgos que cubre la seguridad física. Estos riesgos pueden causar una pérdida considerable para una organización.

D. OWASP Testing Project

1. OWASP. El Proyecto Abierto de Seguridad de Aplicaciones Web (OWASP), es una comunidad que se dedica a ayudar a las organizaciones a desarrollar, comprar y mantener aplicaciones confiables. OWASP ofrece diversos recursos, tales como herramientas, documentos y foros comunitarios.

2. Definición. El Proyecto de Pruebas OWASP es una guía de pruebas de vulnerabilidades compilado por aportes de diferentes profesionales en el campo de la seguridad.

3. **Prueba de penetración de aplicación web.** Método para evaluar la seguridad de un sistema informático o red por medio de un ataque simulado. El proceso de una prueba involucra el análisis de la aplicación, identificando debilidades, defectos o vulnerabilidades.

4. **Vulnerabilidad.** Una vulnerabilidad es un defecto o debilidad en el diseño, implementación u operación y manejo de un sistema, el cual puede ser explotado para violar la política de seguridad del sistema. Una amenaza es ataque potencial, que al explotar una vulnerabilidad puede causar daño sobre los recursos encontrados en una aplicación. (OWASP, 2008).

5. **Categorías de pruebas.** El Proyecto de Pruebas OWASP se desglosa en nueve categorías de pruebas:

a. ***Obtención de información.*** Consiste en recolectar toda la información posible sobre la aplicación, utilizando herramientas públicas como motores de búsqueda.

b. ***Gestión de configuración.*** Consiste en el análisis de la infraestructura y la arquitectura de la topología para encontrar información de la aplicación. Código fuente, métodos HTTP permitidos, métodos de autenticación y configuraciones infraestructurales pueden ser obtenidos.

c. ***Autenticación.*** Consiste en establecer o confirmar que algo o alguien es auténtico, que es que o quien dice ser que es. En seguridad, la autenticación se lleva a cabo por el proceso de verificación de la identidad digital del emisor del a comunicación.

d. ***Gestión de sesiones.*** Consiste de todos los controles y funcionalidades sobre el usuario desde que se autentica hasta que salga de la aplicación.

e. ***Autorización.*** Consiste en dar derechos de acceso a recursos únicamente a los usuarios que tienen privilegios a utilizarlos. La autorización es el proceso que ocurre luego de una autenticación exitosa.

f. ***Validación de datos.*** Consiste en la validación del ingreso de los datos ingresados por el usuario. Por falta de validación de estos datos, puede existir la vulnerabilidad de que la aplicación llegue a interpretar los datos ingresados por el usuario como código de la aplicación.

g. ***Denegación de acceso.*** Es un tipo de ataque que consiste en hacer que un servidor inalcanzable o inaccesible a otros usuarios. Generalmente consiste en congestionar el tráfico de la red hacia la aplicación.

h. ***Servicios web.*** Consiste en servicios que son utilizados por aplicaciones cliente para obtener o escribir datos en la aplicación servidor. Al ser otra ruta de entrada y salida, existe la posibilidad de tener vulnerabilidades.

i. ***AJAX.*** Javascript Asíncrono y XML. Es una técnica de desarrollo utilizado para mejorar la usabilidad de aplicaciones web. Dado que el desarrollo se enfoca más en lo que se puede hacer que en lo que se debería de hacer, deja una mayor posibilidad de explotación de vulnerabilidades.

E. Máquina virtual

1. Definición. Una máquina virtual es una implementación de una máquina que ejecuta aplicaciones como una máquina física.

Una máquina virtual provee una ambiente completo en el que un sistema operativo un varios procesos posiblemente pertenecientes a múltiples usuarios pueden coexistir. Utilizando una máquina virtual, una plataforma de hardware *anfitrión* puede soportar múltiples ambientes de sistemas operativos aislados simultáneamente. (Smith & Nair, 2005)

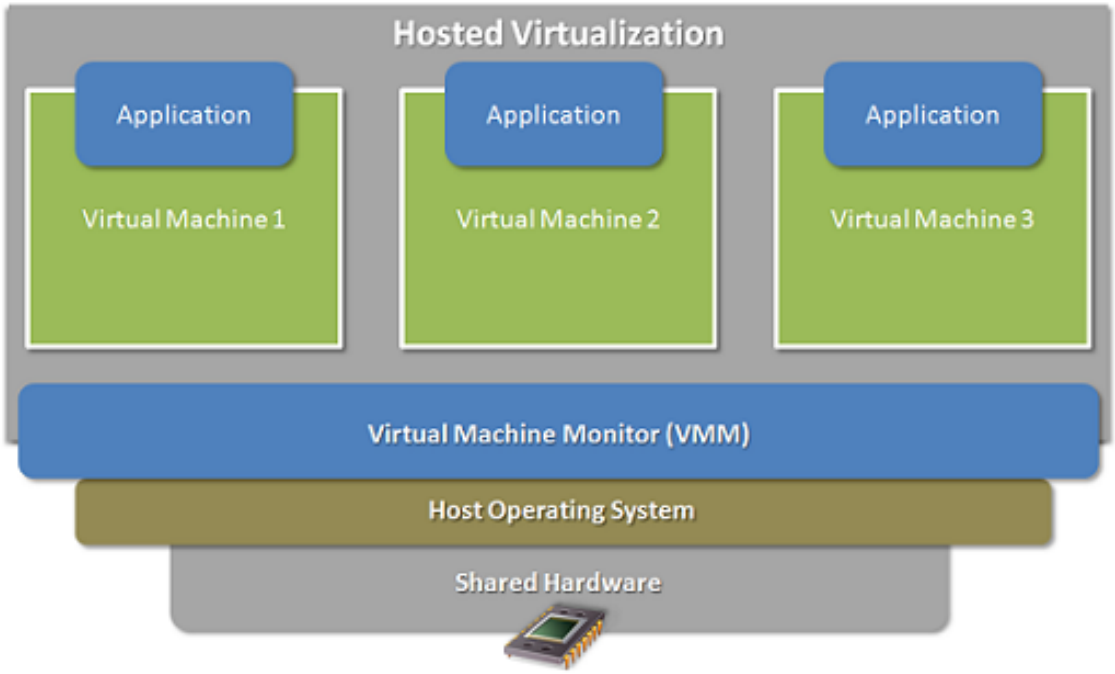
2. VirtualBox. Paquete de software de virtualización de arquitectura x86, creado por Innotek GmbH, posteriormente adquirido en el 2008 por Sun Microsystems y actualmente desarrollado por Oracle Corporation.

VirtualBox es instalado sobre un sistema operativo anfitrión como una aplicación, permitiendo la instalación de otros sistemas operativos bajo su propio

ambiente virtual.

Una propiedad de seguridad que poseen un monitor de máquina virtual como VirtualBox, es la capacidad de encapsular al sistema operativo invitado ejecutándolo en un ambiente virtual protegido, una máquina virtual, ejecutado como una proceso de usuario sobre el sistema operativo anfitrión. El invitado no puede comunicarse directamente con el hardware u otras computadoras, únicamente a través del monitor de máquina virtual. El monitor de máquina virtual provee recursos y dispositivos físicos emulados al invitado, los cuáles son accedidos por el sistema operativo invitado para ejecutar procesos. (Oracle, 2013). Esta propiedad puede ser apreciada en la Figura 1.

Figura 1: Esquema de virtualización



IV. Antecedentes

La Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas, SECCATID, es responsable de coordinar las políticas nacionales antidrogas en Guatemala. Esta entidad carece de un sistema que les permita tener un mejor manejo de los proyectos que realizan junto a otra instituciones gubernamentales y no gubernamentales.

Bajo un convenio entre la Universidad del Valle de Guatemala y SECCATID, se lleva a cabo el desarrollo del Megaproyecto del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas, el cual tiene como objetivo determinar el nivel de cumplimiento de la Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, brindando las herramientas necesarias y un ambiente colaborativo permitiendo la comunicación con las instituciones.

A finales del año 2012, se finaliza el desarrollo del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y Tráfico Ilícito de Drogas, y da por concluido el Megaproyecto.

V. Delimitación e impacto del tema

Este trabajo de graduación comprende de dos partes: La primera es la instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas sobre la infraestructura de SECCATID. La segunda parte consiste una evaluación de vulnerabilidades sobre el sistema.

La instalación del sistema consistió en la preparación y solicitud de los requerimientos necesarios sobre la infraestructura, y llevar a cabo la instalación del sistema sobre la misma. Debido al tiempo y trabajo que se requiere, el presente trabajo de graduación no incluye la implementación y control de calidad del proyecto.

La evaluación de vulnerabilidades se lleva a cabo utilizando el Proyecto de Pruebas OWASP, el cual se desglosa en nueve categorías de pruebas, con un total de 66 pruebas. Debido a la cantidad de pruebas y el trabajo requerido para llevarlas a cabo, la presente evaluación comprende de una cantidad de 28 pruebas, tomando en cuenta las más relevantes al sistema y ambiente en el que será utilizado.

Para cada una de las vulnerabilidades encontradas se puede encontrar descrito bajo las recomendaciones el proceso necesario para llevar a cabo la recomendación. Debido a que algunas de las recomendaciones pueden requerir la alteración de alguna funcionalidad del sistema, queda a discreción del cliente el método a utilizar para manejar la vulnerabilidad.

VI. Metodología

El trabajo de graduación se divide en dos fases: La instalación del sistema y la evaluación de vulnerabilidades.

Para llevar a cabo la primera fase, se evalúan los requerimientos de equipo necesarios para que el sistema pueda funcionar de una manera óptima sobre la infraestructura de SECCATID. Una vez realizada la evaluación, dichos requerimientos son dados a conocer a SECCATID para obtener el equipo necesario.

La instalación del sistema sobre la infraestructura es divido en cada una de los seis módulos que la componen: Tablero de Control y Reportes; Administración de Proyectos; Ambiente de Trabajo Colaborativo; Soporte y Documentación para Asistencia a Usuarios; Capacitación; y Seguridad.

Una vez al tener el sistema en ejecución, se prosigue la segunda fase, el cual consiste en la elaboración de una evaluación de seguridad del sistema. Para ello se utiliza de referencia el Proyecto de Pruebas OWASP versión 3 para llevar a cabo las pruebas de seguridad.

El Proyecto de Pruebas OWASP, consiste en una guía dividida en nueve categorías, cada diferentes cantidades de pruebas, teniendo un total de 66 pruebas. Debido a la cantidad de pruebas y la cantidad de trabajo que toma realizarlas, se elige un conjunto de pruebas que más aplicaban al sistema y contexto en el que se encuentra.

VII. Análisis del problema

A. Descripción del problema

El Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas fue desarrollado para permitir que SECCATID lleve el seguimiento del cumplimiento de la Política Nacional Contra Las Adicciones y el Tráfico Ilícito de Drogas, aprovechando los beneficios de la tecnología para almacenar la manejar esta información.

El planteamiento de este trabajo de graduación fue por dos razones: La primera consiste en la instalación del sistema en la infraestructura de SECCATID, luego de haber concluido el desarrollo del proyecto y estar listo para la siguiente fase, ejecución. La segunda consiste en realizar una evaluación de vulnerabilidades sobre el sistema, ya que como fue tomado en cuenta durante el desarrollo, SECCATID necesita asegurarse que la información crítica que se maneje en el sistema no sea diluida ni alterada por ninguna entidad externa.

B. Descripción de la solución

Para cubrir la necesidad del problema, se propone la solución de iniciar la instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, y llevar a cabo las solicitudes necesarias para tener todos los requerimientos para tener acceso remoto tanto al sistema como a la infraestructura para llevar a cabo las pruebas de instalación necesarias.

Se lleva a cabo una evaluación de vulnerabilidades sobre el sistema, utilizando como guía el Proyecto de Pruebas OWASP versión 3. Estas vulnerabilidades son identificadas para mitigar la posibilidad de ocurrencia de cualquier evento o acción dañina por parte de cualquier entidad con la intención de penetrar el sistema.

C. Justificación de la propuesta

La Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID) vinculada a la vicepresidencia de la República, funciona como oficina técnica central de la CCATID, autoridad nacional, responsable de coordinar las políticas nacionales antidrogas en Guatemala. Además de tener fundamento legal, es responsable de coordinar acciones en las siguientes áreas: reducción de la demanda, reducción de la oferta, programas de desarrollo relacionados con la prevención o la reducción de cultivos ilícitos, producción o tráfico de drogas, cooperación internacional y evaluación de programas.

A pesar de tener concluido el desarrollo del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, existía aún la necesidad de llevar a cabo la instalación sobre la infraestructura en la que se utilizaría, además de asegurarse ésta cumpla con los requerimientos necesarios para su uso.

Dado que se requiere realizar pruebas sobre el sistema, se utiliza la guía del Proyecto de Pruebas OWASP para realizar una evaluación de vulnerabilidades, el cuál consiste en pruebas sobre una aplicación web.

La evaluación de vulnerabilidades se enfoca en llevar a cabo intentos de penetración y otros métodos para obtener información sin autorización del sistema. Simulando intentos de acceso que un atacante podría realizar, lo cual es de gran importancia debido al alto nivel de delicadeza de la información manejada por esta entidad gubernamental.

Para la selección del Proyecto de Pruebas OWASP como guía para la evaluación de vulnerabilidades se realizó una investigación comparativa de diferentes guías y estándares de vulnerabilidades disponibles actualmente. A continuación en la Figura 2 se presenta una tabla que resume las características de cada una.

Figura 2: Tabla de comparación de guías de evaluación de seguridad

	OWASP (Open Web Application Security Project)	OSSTMM (Open Source Security Testing Methodology Manual)	PTES (Penetration Testing Execution Standard)
Enfoque	• Vulnerabilidades en aplicaciones web	• Seguridad física, interacciones humanas, comunicaciones alámbricas, inalámbricas, análogas y digitales.	• Ejecución de pruebas de penetración con herramienta automatizadas.
Contribuciones y documentación	• Comunidad abierta. • Contribuciones aprobadas bajo revisión de la organización. • Explicación del proceso de las pruebas.	• Comunidad abierta. • Contribuciones aprobadas bajo revisión de la organización. • Menciona las pruebas recomendadas a realizar para cada a evaluar.	• Comunidad abierta. • Contribuciones aprobadas bajo revisión de la organización. • Pruebas enfocadas en la utilización de herramientas automatizadas.
Licencia	• Material publicado y borradores gratuitos bajo licencia *Creative Commons Attribution 3.0 Unported*	• Material publicado gratuito bajo licencia *Creative Commons Attribution 3.0*. Borradores bajo suscripción pagada.	• Material publicado gratuito bajo licencia GPL.

Con base en la investigación, el Proyecto de Pruebas OWASP fue elegido por las siguientes razones:

- El enfoque exclusivo sobre las pruebas de penetración y las vulnerabilidades sobre las aplicaciones web.
- El acceso libre a borradores de nuevas versiones en desarrollo, lo cuál permite el acceso inmediato a todo conocimiento contribuido sin ningún costo de parte del cliente. Este aspecto es de gran importancia para la realización de futuras evaluaciones de vulnerabilidades periódicas.
- La explicación completa y concisa sobre el proceso de las pruebas descritas. Utilizando explicaciones a nivel de caja negra y caja gris.

D. Riesgos y limitaciones de la propuesta

El proceso de instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas consiste en tener el sistema sobre la infraestructura y en ejecución, y asegurar que se tenga acceso remoto al sistema por ser de ambiente web colaborativo. Después de dicha fase es necesario que se lleve a cabo una fase de implementación y control de calidad, para asegurarse que el sistema funcione correctamente y cumpla con los requerimientos.

La evaluación de vulnerabilidades es realizada a nivel de ataques realizados al sistema, por lo que se sugiere también realizar evaluaciones a nivel de sistema operati

VIII. Diseño

A. Instalación del sistema

Para la instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, se prepara una máquina virtual utilizando Oracle VirtualBox, con las siguientes características:

- 2GB de memoria RAM virtual
- 10 GB de espacio libre en disco duro expandible
- Adaptador de red virtual en modo puente

Las características de la máquina virtual fueron asignadas bajo los requerimientos mínimos establecidos por los desarrolladores. El modo puente del adaptador de red virtual es elegido para que dentro de la red la máquina virtual sea tratada y accedida como otro ordenador dentro de la red física. A este adaptador se le asigna la dirección IP pública fija.

El sistema operativo utilizado es Ubuntu Server 12.10, con las siguientes aplicaciones y servicios para ejecutar el sistema:

- Apache 2.2.22
- MySQL 5.5.27
- PHP 5.4.6

También es instalado phpMyAdmin para la administración de la base de datos y OpenSSH para el acceso remoto para facilitar el proceso de la instalación.

B. Evaluación de vulnerabilidades

La guía descrita en el Proyecto de Pruebas OWASP, está compuesta de pruebas clasificadas en nueve categorías:

1. Obtención de información
2. Gestión de configuración
3. Autenticación
4. Gestión de sesiones
5. Autorización
6. Validación de datos
7. Denegación de acceso
8. Servicios web
9. AJAX

Por la alta cantidad de pruebas, el tiempo disponible para realizarlas y la relevancia que tienen para el Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, se escogieron las más críticas e importantes que aplican para dicho sistema. Esta selección se realiza en base al conocimiento previo que se tiene sobre el sistema, como la plataforma, las funcionalidades que posee, la base de datos que utiliza, entre otros.

Debido a que el sistema no posee servicios web que permitan la comunicación hacia algún cliente, las pruebas de la categoría de Servicios web no son realizadas.

Las pruebas que pertenecen a la categoría de AJAX tampoco son realizadas, debido a que el sistema no posee de ninguna sección o módulo que haya sido desarrollado utilizando AJAX.

IX. Instalación del sistema

A. Requerimientos

Para llevar a cabo la instalación del Sistema de Monitoreo del Plan Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas, fue necesario contar con un sistema con los requerimientos mínimos:

- 2GB de memoria RAM
- 10 GB de espacio libre en disco duro
- Sistema operativo Linux
    - Servidor web Apache 2.x
    - Servidor de base de datos MySQL
- Acceso a internet con IP fija

Tras mandar la solicitud a la Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas (SECCATID), un servidor con las siguientes especificaciones fue ofrecido:

- 4GB de memoria RAM
- Disco duro de 110 GB
- Sistema operativo Windows Server 2008 R2 Standard
- Dirección IP pública fija

B. Proceso

Para llevar a cabo la instalación, se creó una máquina virtual con sistema operativo Ubuntu. Sobre la cual se llevó la instalación de todos los requerimientos de dependencias de software, servidores web y base de datos, y el sistema de monitoreo. El uso de una máquina virtual fue dado por las siguientes razones:

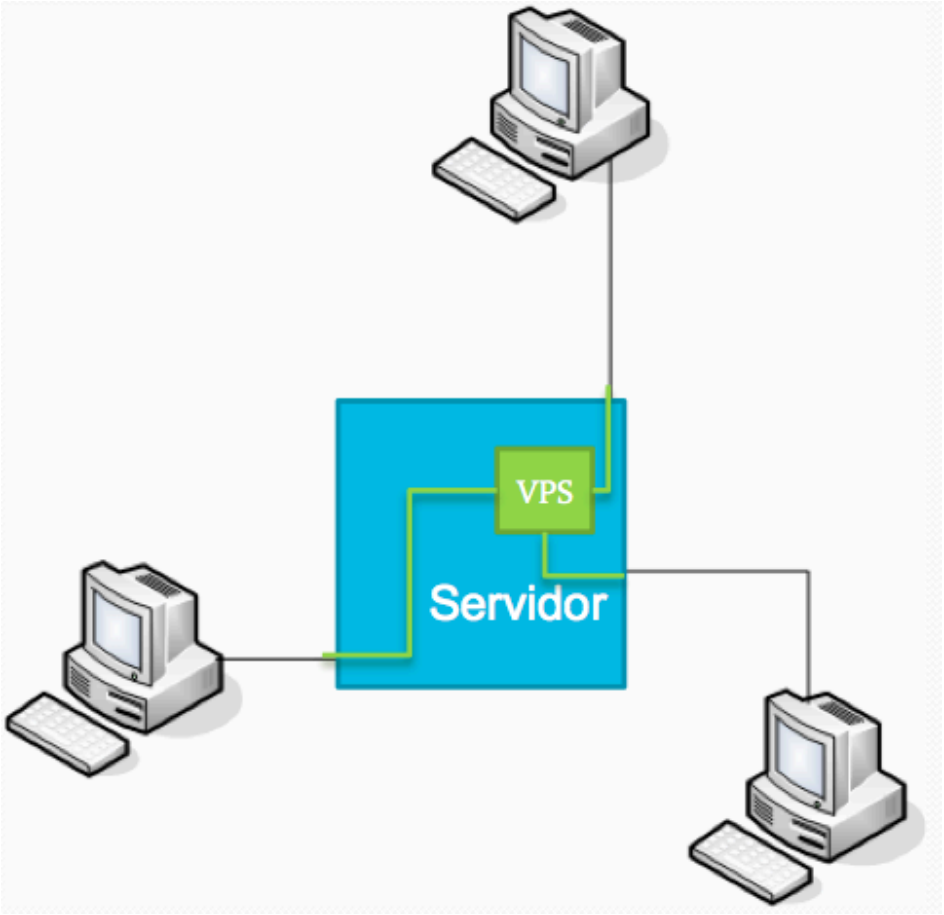
- El sistema operativo es diferente al requerido, un sistema operativo Linux era necesario para las dependencias sobre las cuáles se ejecuta el sistema.
- La instalación se lleva a cabo de una manera más práctica, ya que todo software es instalado sobre la máquina virtual. Una vez estando en las oficinas centrales de SECCATID, se instala únicamente dicha máquina virtual y se configura la red. De esta manera se redujo el tiempo requerido

para la instalación en las oficinas de SECCATID.

- La máquina virtual ofrece un aislamiento del sistema operativo en el cual se encuentra instalado el sistema de monitoreo, lo cual le da una capa de seguridad al sistema operativo del servidor y la información que se encuentra en él.

La dirección IP fija que fue proveída por el proveedor de servicio de internet de SECCATID, fue asignada sobre el adaptador de red virtual de la máquina virtual en la que fue instalada. De esta manera, al acceder al desde la dirección IP pública, se ingresa directamente a la máquina virtual, dejando al servidor físico aislado del acceso público.

Figura 3: Esquema de conexión clientes a servidor de aplicación en máquina virtual



C. Resultados

El sistema se encuentra ya en ejecución, funcionando sobre una máquina virtual con sistema operativo Ubuntu Server 12.10, accesible desde la IP pública proveída por SECCATID.

Figura 4: Portal de ingreso del sistema

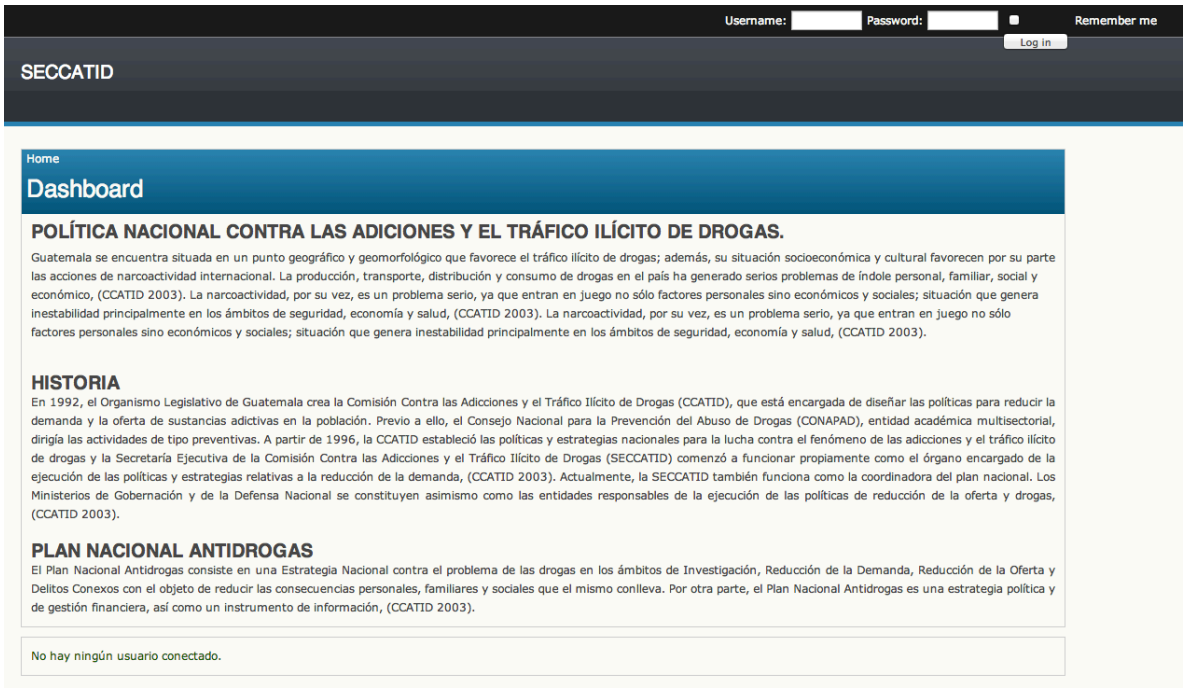
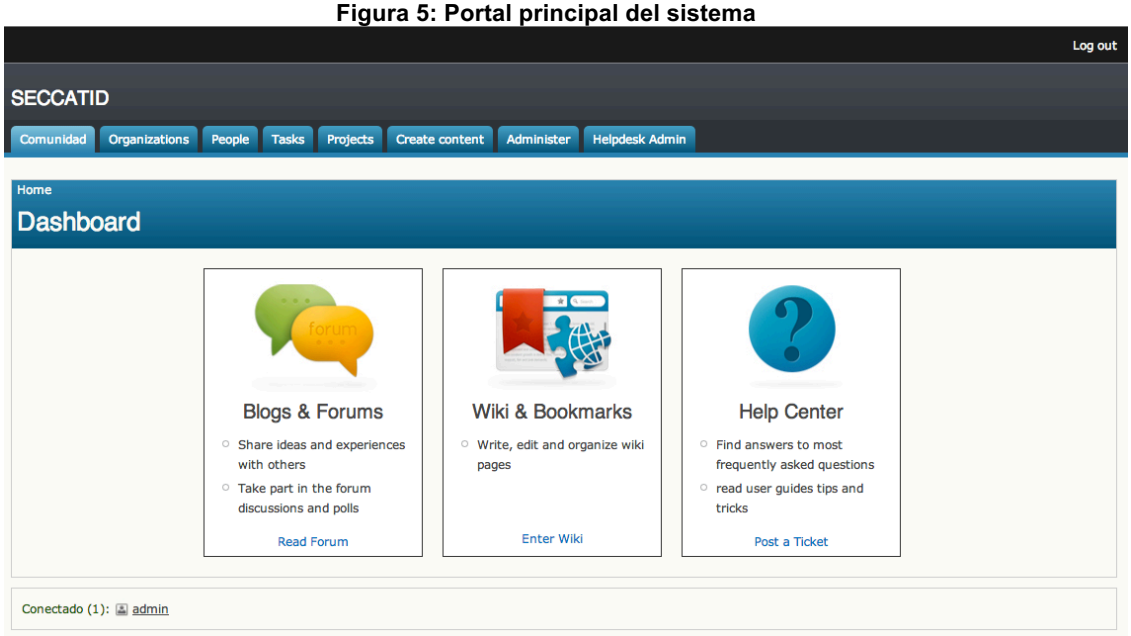


Figura 5: Portal principal del sistema



X. Evaluación de vulnerabilidades

A continuación se presentan las pruebas de vulnerabilidades realizadas, organizadas por sus respectivas categorías según la guía de pruebas OWASP v3. Cada consiste de una descripción de la prueba y la vulnerabilidad de encontrar, el resultado obtenido y la recomendación en caso de que haya sido encontrado la vulnerabilidad.

Para la documentación del proceso realizado para cada una de las pruebas, consultar el anexo en la sección XIII.A.

A. Obtención de información

1. Spiders, robots y crawlers

a. Descripción. Esta prueba consiste en la obtención del archivo robots.txt, en el cual se definen las reglas sobre los archivos que se encuentran en el directorio y subdirectorios para los motores de búsqueda. De esta manera es posible definir los directorios o archivos que deben ser ignorados por el motor.

b. Resultado. Se encontró el archivo robots.txt en la raíz del sitio de la aplicación, con reglas de rechazo a diferentes archivos y carpetas del sistema.

c. Recomendación. Dado que cualquier usuario puede acceder a este archivo, se recomienda utilizar una única regla de rechazo sobre el directorio raíz del sistema.

2. Descubrimiento por motores de búsqueda

a. Descripción. Esta prueba consiste en encontrar copias almacenadas del sitio hechas por motores de búsqueda. Estas copias pueden ser de sitios o archivos ocultos o eliminados.

b. ***Resultado.*** Realizando la búsqueda utilizando la dirección IP del servidor como palabra clave no se encontró ningún resultado.

c. ***Recomendación.*** Ninguna.

3. Prueba de huella digital

a. ***Descripción.*** Consiste en conocer la huella digital del servidor web en el que la aplicación se encuentra ejecutando. De esta manera se puede conocer la versión y las vulnerabilidades que pueda tener el servidor.

b. ***Resultado***

Figura 6: Huella digital

```
HTTP/1.1 304 Not Modified
Server:        Apache/2.2.22
       (Ubuntu)
Connection: Keep-Alive
Keep-Alive:       timeout=5,
       max=100
ETag:             "43dc2-b1-
       4d8538fd4e327"
Vary: Accept-Encoding
```

Tal como se puede observar en la huella digital, se puede saber que el sistema se encuentra ejecutándose sobre el servidor Apache versión 2.2.22, y éste sobre el sistema operativo Ubuntu

c. ***Recomendación.*** Actualizar tanto el servidor web, como sus componentes y el sistema operativo es la mejor manera de impedir la explotación de vulnerabilidades identificadas en versiones anteriores.

4. Descubrimiento de aplicaciones

a. Descripción. Consiste en encontrar qué otras aplicaciones se encuentran instaladas y en ejecución en el servidor. Es importante conocer dichas aplicaciones debido a que éstas también pueden tener vulnerabilidades y el sistema puede llegar a ser afectado a causa de ello.

b. Resultado

Figura 7: Puertos abiertos

PORT	STATE	SERVICE
22/tcp	open	ssh
80/tcp	open	http
8080/tcp	open	http-proxy

a. Recomendación. Cambiar el número puerto para el servicio de conexión SSH a un número mayor, que no se encuentre en el rango de los puertos de servicios comunes. Se recomienda dejar únicamente el puerto 80 abierto, el cuál es utilizado por el sistema web.

Para cambiar el número de puerto, modificar el archivo */etc/ssh/sshd_config*, especificando el número de puerto en la línea que contenga *Port*.

5. Análisis de códigos de error

a. Descripción. La recolección y análisis de los diferentes códigos de error generados por el servidor web, permite tener información sobre aspectos técnicos, bases de datos o fallas que pueda tener el servidor.

b. Resultado

Figura 8: Mensajes de error

Error	Mensaje desplegado
404 Not Found	Not Found The requested URL /asd was not found on this server. *Apache/2.2.22 (Ubuntu) Server at 190.149.207.190 Port 80*

Continuación figura 8: Mensajes de error

403 Acces denied	Access denied You are not authorized to access this page.

c. ***Recomendación.*** Se recomienda crear mensajes de error personalizados para todos los errores posibles para evitar dar información sobre el sistema.

En el caso del error 403, Drupal despliega una página personalizada. Para el error 404, se puede instalar el módulo *Custom 404: https://drupal.org/project/custom404*

B. Pruebas de gestión de configuración

1. Prueba de receptor de base de datos

a. ***Descripción.*** Consiste en verificar si existe algún receptor de base de datos en ejecución. Este generalmente muestra información sobre la base de datos que se está utilizando.

b. ***Resultado.*** No se encontró ningún receptor de base de datos

c. ***Recomendación.*** Ninguna

2. Enumeración de métodos HTTP y XST

a. ***Descripción.*** La prueba consiste en encontrar sí existen métodos HTTP críticos que pueden dejar el sistema vulnerable. Estos métodos son:

- PUT: Permite la carga de archivos al servidor
- DELETE: Permite la eliminación de archivos en el servidor.
- TRACE: Permite la obtención de una respuesta de conexión satisfactoria de parte del servidor. Este método puede ser explotado para realizar Cross-Site Tracing.

b. ***Resultado.*** Utilizando el método HTTP OPTIONS se obtuvieron los métodos habilitados en el servidor web, estos fueron: GET, POST, HEAD y OPTIONS.

Ninguno de los métodos que presentan vulnerabilidades fueron enlistados. Para asegurarse de ello, se realizaron pruebas para utilizar los métodos TRACE, TRACK, PUT y DELETE. Ninguna llamada a estos métodos fue permitido.

c. ***Recomendación.*** Ninguna.

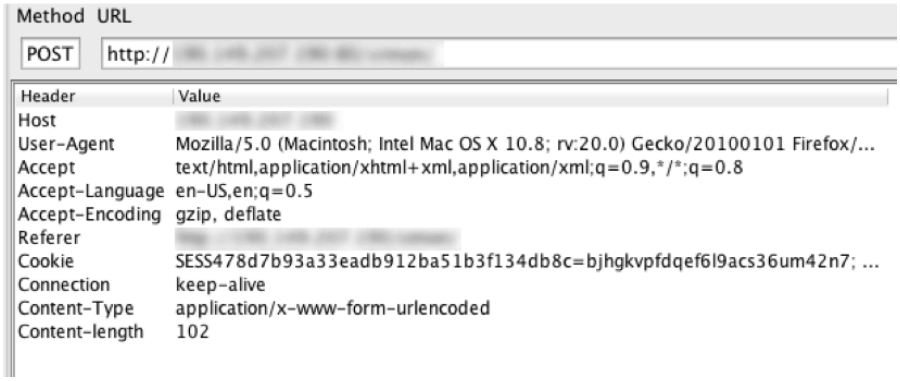
C. Pruebas de autenticación

1. Transferencia de credenciales sobre un medio encriptado

a. ***Descripción.*** Esta prueba consiste en verificar que las credenciales sean transferidas siempre sobre un medio encriptado, para que los datos no sean interceptados otras entidades.

b. ***Resultado.***

Figura 9: Petición de autenticación interceptada por WebScarab



El método que utiliza par mandar los datos del formulario web de autenticación de la aplicación es POST.

d. ***Recomendación.*** Se recomienda utilizar encriptación SSL para proteger el canal de transferencia de datos, para evitar que ésta sea interceptada fácilmente por cualquier usuario en la misma red por medio de un sniffer.

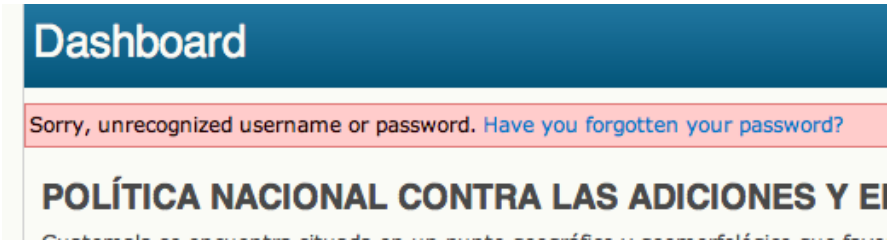
Para llevar a cabo la implementación de un medio encriptado SSL, se debe obtener un certificado válido de cualquiera de los proveedores[1] y seguir los pasos descriptos en la documentación de Drupal: https://drupal.org/https-information.

2. Enumeración de usuarios

a. ***Descripción.*** Consiste en encontrar si es posible encontrar un conjunto de usuarios válidos interactuando con el mecanismo de autenticación de la aplicación. Teniendo un conjunto válido de usuarios permite que un ataque de fuerza bruta sea más eficiente.

b. ***Resultado.*** Se obtuvo un mensaje de error de usuario o contraseña no reconocida al intentar ingresar ya sea con un usuario existente o inexsitente.

Figura 10: Mensaje de error de autenticación



Utilizando la herramienta WebScarab para interceptar las respuestas, no se obtuvo ningún mensaje que indique o implique que el usuario no existe o que la contraseña fuera incorrecta.

c. ***Recomendación.*** Ninguna

[1] Conulstar anexo para lista de proveedores de certificados SSL

3. Cuentas de usuario predeterminados

a. ***Descripción.*** Se puede decir que la mayoría de los sistemas luego de ser instalados poseen una sola cuenta de administrador, ésta usualmente está compuesto por una palabra común para el nombre de usuario y la contraseña. En muchos casos, estas cuentas de usuario son dejadas intactas por los administradores, permitiendo que cualquier usuario logre ingresar simplemente probando diferentes combinaciones comunes.

b. ***Resultado.*** Se encontró que la cuenta de usuario admin:admin se encuentra existente. También se encontró que el sistema tiene un máximo de cinco intentos fallidos.

c. ***Recomendación.*** Se recomienda eliminar la cuenta admin, y siempre evitar nombres de usuario genéricos, nombres que incluyan información de la persona, o palabras comunes que puedan ser encontradas en un diccionario.

Para eliminar las cuentas predeterminadas, acceder a la sección *Administer→Users* del sistema.

4. Prueba de fuerza bruta

a. ***Descripción.*** Esta prueba consiste en utilizar un conjunto de combinaciones de usuarios y contraseñas, que pueden ser obtenidos de diccionarios o utilizando cadenas de caracteres generadas, utilizando todas las combinaciones posibles para ingresar al sistema.

b. ***Resultado.*** Se obtuvieron falsos resultados positivos tras 20 intentos de combinaciones, debido a que después de 20 intentos, la dirección IP del cliente es bloqueado, mostrando una página diferente, el cuál no incluye el mensaje de autenticación fallida.

c. ***Recomendación.*** Se recomienda asignar preferiblemente los tiempos de esperas altos y los límites de reintentos bajos para máxima seguridad, pero tomando en cuenta el error humano.

También se sugiere eliminar el mensaje de autenticación fallida, lo cual es utilizado en la fuerza bruta para identificar los intentos de acceso fallidos. En este caso se notó que además del mensaje, los bordes de los campos de usuario y contraseña cambiaban a un color rojo, lo cuál puede ser retroalimentación suficiente para el usuario.

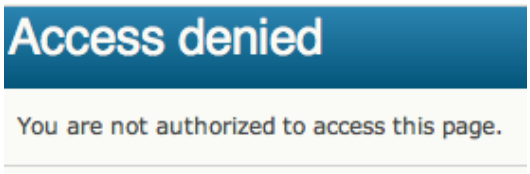
Los parámetros mencionados pueden ser configurados en la sección *Administer→Login Security* del sistema.

5. Salto de esquema de autenticación

a. ***Descripción.*** Esta prueba consiste en verificar la posibilidad de ingresar a sitios internos del sistema por medio de la dirección directa, si esta se conoce, sin pasar por el sistema autenticación. Estas direcciones generalmente pueden ser obtenidas por medio de un usuario que ya ha logrado autenticarse.

b. ***Resultado.*** Al intentar ingresando a la sección interna de administración de instituciones del sistema, desde varias direcciones a secciones internas, el sistema devolvió un mensaje de acceso denegado.

Figura 11: Mensaje de error de acceso denegado



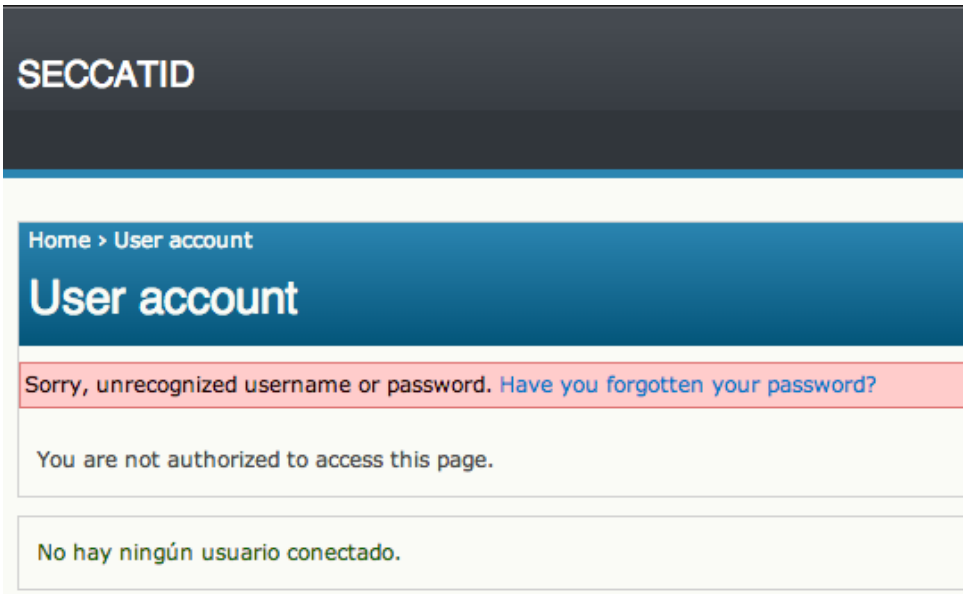
c. ***Recomendación.*** Ninguna.

6. Prueba de restablecimiento de contraseñas

a. ***Descripción.*** Consiste en identificar el mecanismo que utiliza el sistema para guardar la contraseña en el explorador y restablecer contraseñas. Dependiendo del mecanismo que se utilizó, éstos pueden tener alguna vulnerabilidad, como en el caso de las preguntas secretas, como mecanismo de restablecimiento de contraseña.

b. ***Resultado.*** Se encontró la opción *Forgotten password*, pero al ingresar se obtiene el mensaje de acceso denegado, por lo que se encuentra habilitado.

Figura 12: Mensaje de acceso no autorizado al ingresar a la opción de "Contraseña olvidada"



c. ***Recomendación.*** Se recomienda no utilizar dicho mecanismo, y eliminar los vínculos a la opción de restablecimiento de contraseña si se encuentran inhabilitada.

Este vínculo puede ser inhabilitado, desactivando el mensaje autenticación fallida, en la sección *Administer→Login Security.*

7. Prueba de cierre de sesión

a. ***Descripción.*** Consiste en verificar que el mecanismo del cierre de sesión no deje ningún mecanismo que permita reiniciar la sesión.

b. ***Resultado.*** Al intentar ingresar a una sección que requiere autenticación utilizando una cookie de una sesión "cerrada", se obtuvo el mismo mensaje de acceso denegado.

c. ***Recomendación.*** Ninguna.

8. Prueba de CAPTCHA

a. ***Descripción.*** Consiste en identificar si se utiliza algún mecanismo de CAPTCHA para la autenticación del usuario, y si posee alguna vulnerabilidad a ataques.

b. ***Resultado.*** No se encontró ningún sistema de CAPTCHA durante el proceso de autenticación.

c. ***Recomendación.*** En caso de implementar un sistema de CAPTCHA para la autenticación, es necesario verificar el mecanismo que se utiliza para generarla y la validación del ingreso de parte del usuario. Por lo general se recomienda utilizar sistemas CAPTCHA ya existentes como reCAPTCHA, que generalmente ya han sido puestos a prueba utilizando un OCR.

Figura 13: Ejemplo de una prueba de reCAPTCHA

Para implementar un mecanismo de CAPTCHA, basta con instalar el módulo de CAPTCHA encontrado en el sitio oficial de Drupal. https://drupal.org/project/captcha

D. Pruebas de gestión de sesiones

1. Prueba de fijación de sesión

a. ***Descripción.*** Cuando una aplicación de renueva el cookie después de una autenticación exitosa, es posible encontrar una vulnerabilidad y forzando al usuario utilizar un cookie conocido por el atacante, permitiendo al atacante entrar con la sesión del usuario.»(OWASP, 2008)

b. ***Resultado.*** Al iniciar sesión, terminarla y haber iniciado otra sesión, se puede observar que el sistema genera un nuevo cookie diferente.

c. ***Recomendación.*** Ninguna

2. Pruebas de Cross-Site Request Forgery (CSRF)

a. ***Descripción.*** CSRF es un tipo de ataque en el que el usuario sin la intención ejecuta acciones en una aplicación web en el que se encuentra autenticado. Este tipo de ataque se puede realizar con un poco de ingeniería social, mandando un link directo a la acción al usuario.

b. ***Resultado.*** Revisando las acciones posibles a realizar en el sistema, se pudo notar que todas se realizan por medio de POST, todos los cuales utilizan el form-build-id, generado por la página que permite ejecutar la acción. Por lo que el sistema no es vulnerable a este tipo de ataques.

c. ***Recomendación.*** Ninguna.

E. Pruebas de autorización

1. Ruta transversal

a. ***Descripción.*** Consiste en explotar funciones de obtención de usuarios para obtener archivos que se encuentran en directorios que normalmente no pueden ser accedidos, fuera del directorio web al que pertenece el archivo, los permisos que puedan tener. Utilizando la notación "../", que representa al directorio padre, es lo que hace posible realizar este ataque.

b. ***Resultado.*** No se encontró ninguna función que acceda a archivos locales al directorio de la aplicación.

c. ***Recomendación.*** Ninguna.

2. Prueba de salto de esquema de autorización

a. ***Descripción.*** La prueba consiste verificar la posibilidad de acceder a recursos a los cuáles el usuario no tiene autorización para accederlos.

b. ***Resultado.*** Se obtuvo mensaje de acceso denegado al intentar ejecutar acciones o acceder recursos no privilegiados al usuario.

c. ***Recomendación.*** Ninguna.

3. Escalamiento de privilegios

a. ***Descripción.*** El escalamiento de privilegios ocurre cuando un usuario obtiene acceso a más funcionalidades o recursos del que está permitido, obteniendo privilegio de manera vertical.

b. ***Resultado.*** No se encontraron posibles vulnerabilidades de escalamiento de privilegios durante la autenticación del usuario o durante la ejecución de acciones.

c. ***Recomendación.*** Ninguna.

F. Pruebas de validación de datos

1. Prueba de Cross Site Scripting

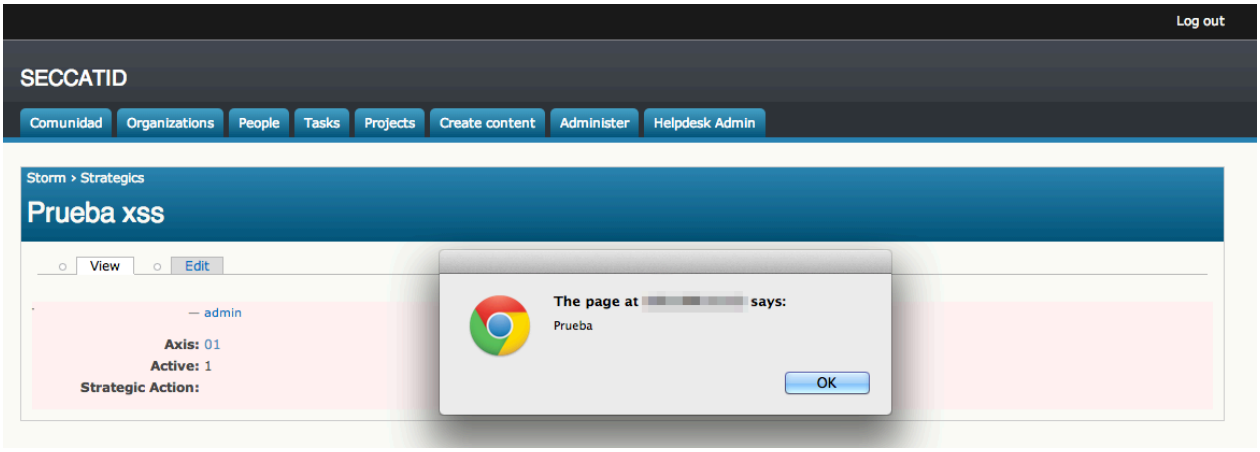
a. ***Descripción.*** Esta prueba consiste en identificar en campos de ingreso de texto la posibilidad de ejecutar scripts. Generalmente esto es posible en casos en el que el usuario le es permitido utilizar etiquetas de HTML.

b. ***Resultado.*** Se encontró una vulnerabilidad en la sección de “Crear acción estratégica”, el campo de texto para la descripción de la acción estratégica permite el ingreso de texto con etiquetas HTML. Se utilizó como prueba el siguiente script:

```
<script>alert('Prueba');</script>
```

La misma opción de ingreso de texto con etiquetas HTML fue encontrado para otras secciones como Organizaciones, Personas, Proyectos, pero en ninguno de ellos el script es ejecutado.

Figura 14: Resultado de prueba de XSS en el sistema



a. ***Recomendación.*** Se recomienda inhabilitar la opción de uso de etiquetas HTML en cualquier campo de ingreso o limitarlo únicamente a usuarios administradores.

Las etiquetas HTML pueden ser inhabilitadas en la sección *Administer→Modules* y desactivando el módulo *CKEditor*.

2. Inyección SQL

a. ***Descripción.*** Consiste en insertar consultas SQL válidas a través de campos de ingreso que se pueden encontrar en la aplicación, disponibles al usuario. Esta vulnerabilidad puede ser explotada para obtener datos críticos de la base de datos, autenticarse como otros usuarios sin conocer la contraseña o realizar acciones no privilegiadas al usuario.

b. ***Resultado.*** No se encontraron campos con vulnerabilidades de inyección SQL.

c. ***Recomendación.*** Ninguna.

3. Inyección ORM

a. ***Descripción.*** Al igual que la inyección SQL, consiste en insertar consultas a través de campos de ingreso. La diferencia se encuentra en la consulta que se envía, que en este caso son funciones del ORM utilizado, los cuales son utilizados para realizar operaciones CRUD en la base de datos.

b. ***Resultado.*** No se encontraron campos con vulnerabilidades de inyección ORM.

c. ***Recomendación.*** Ninguna.

4. Inyección XML

a. ***Descripción.*** Consiste en insertar datos estructurados con etiquetas XML a través de los campos de ingreso. Esta vulnerabilidad se puede encontrar en las aplicaciones que utilizan XML como medio de comunicación de datos.

b. ***Resultado.*** No se encontraron campos con vulnerabilidades de inyección XML.

c. ***Recomendación.*** Ninguna.

5. Inyección SSI

a. ***Descripción.*** Generalmente los servidores web tienen la capacidad de ejecutar código incrustado en páginas HTML para ejecutar funciones de parte del servidor. La inyección SSI consiste en insertar este tipo de comandos dentro de los campos de ingreso.

b. ***Resultado.*** No se encontraron campos con vulnerabilidades de inyección SSI.

c. ***Recomendación.*** Ninguna.

G. Pruebas de denegación de acceso

1. Ataques con comodines SQL

a. ***Descripción.*** Consiste en realizar búsquedas que cause al sistema realizar consultas SQL sobre la base de datos que requieran uso intensivo del procesador. Esta vulnerabilidad generalmente existe en toda aplicación que tenga la opción de búsqueda.

b. ***Resultado.*** No se encontró ninguna opción de búsqueda que permita al usuario ingresar texto al sistema. Las búsquedas se realizan mediante fechas y otros atributos seleccionados de una lista.

c. ***Recomendación.*** Ninguna.

2. Congelación de cuentas de usuario

a. ***Descripción.*** Esta prueba consiste en verificar si es posible congelar una cuenta por medio de una serie de intentos de acceso fallidos. Esta vulnerabilidad se encuentra en sistemas de autenticación que poseen un sistema de seguridad automatizado de bloqueo para evitar ataques de fuerza bruta.

b. ***Resultado.*** Se encontró que tras 20 intentos fallidos, la dirección IP del cliente fue bloqueada pero no el usuario, impidiendo el acceso desde la dirección IP del cliente de las pruebas.

c. ***Recomendación.*** Se recomienda utilizar un mecanismo de CAPTCHA para evitar cualquier bloqueo intencional de parte de un atacante y a la vez proteger el sistema de autenticación ante ataques de fuerza bruta.

Para implementar un mecanismo de CAPTCHA, seguir recomendación descrita para la prueba de CAPTCHA.

3. Prueba de desbordamiento de búfer

a. ***Descripción.*** Consiste en proveer una cantidad de datos que exceda el espacio de memoria reservado por la aplicación, lo cual causa un desbordamiento de búfer, causando que se cierre o congele la aplicación.

b. ***Resultado.*** Se realizaron pruebas de desbordamiento búfer en los diferentes campos de las secciones de creación de personas, proyectos, instituciones, y acciones estratégicas, y en la sección de autenticación. No ocurrió ningún desbordamiento de búfer.

c. ***Recomendación.*** Ninguna.

XI. Conclusiones y recomendaciones

- Se recomienda tratar las vulnerabilidades encontradas siguiendo las recomendaciones descritas en la evaluación antes de entrar a producción.

- Establecer un proceso de actualización de la plataforma, Drupal, y módulos, y asegurar que no se presenten vulnerabilidades conocidas

- Se notó una conexión sumamente inestable en la infraestructura en el cual se instaló el sistema. Se recomienda contar con redundancia de enlaces, de tal manera que la conexión del servidor no sea afectada por el uso de la conexión a internet de parte de las oficinas.

- Para una mayor seguridad, se recomienda adquirir servicios de certificados SSL lo cual permite una mejor autenticidad a los usuarios, y además provee un canal encriptado para la protección de los datos que son transferidos.

- Se recomienda realizar evaluaciones de vulnerabilidades periódicas, al menos una vez cada tres meses, con base en nuevas vulnerabilidades identificadas por la comunidad OWASP.

- Es recomendable realizar una fase posterior a esta en el que se realicen pruebas y se verifique que el sistema esté adaptada a las necesidades que tienen los usuarios.

- Realizar un proceso de “hardening” sobre el sistema operativo del servidor en el que se encuentra instalado el sistema.

XII. Bibliografía

Borghello, Cristian. (2001) Seguridad Informática: Sus implicancias e implementación.

Erickson, J. (2008). *Hacking: The Art of Exploitation* (2 ed.). San Francisco, Estados Unidos: No Starch Press.

British Standards Institution. (2005). International Standard ISO/IEC 27001. 1a Edición. BSI.

Erickson, Jon. (2008). Hacking: The Art of Exploitation. 2a Edición. No Starch Press.

Harris, Shon. (2007). CISSP Certification All-In-One Exam Guide. 4a Edición. McGraw Hill.

ISECOM. (2010). Open Source Security Testing Methodology Manual: Contemporary Security Testing and Analysis. 3a Edición. ISECOM.

Oraclle. (2013). Oracle VM VirtualBox: User Manual. Oracle Corporation

OWASP. (2008). The OWASP Testing Project. https://www.owasp.org/index.php/OWASP_Testing_Project

SECCATID. (2009). Política Nacional Contra las Adicciones y el Tráfico Ilícito de Drogas. Secretaría Ejecutiva de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas.

Smith, James; Nair, Ravi. (2005). The Architecture of Virtual Machines. IEEE Computer Society.

Tipton, Harold; Henry, Kevin. (2008). Official (ISC)2 Guide to the CISSP CBK. 2a Edición. Auerbach Publications.

XIII. ANEXO

A. Procedimientos realizados para las pruebas vulnerabilidades OWASP

1. Obtención de información

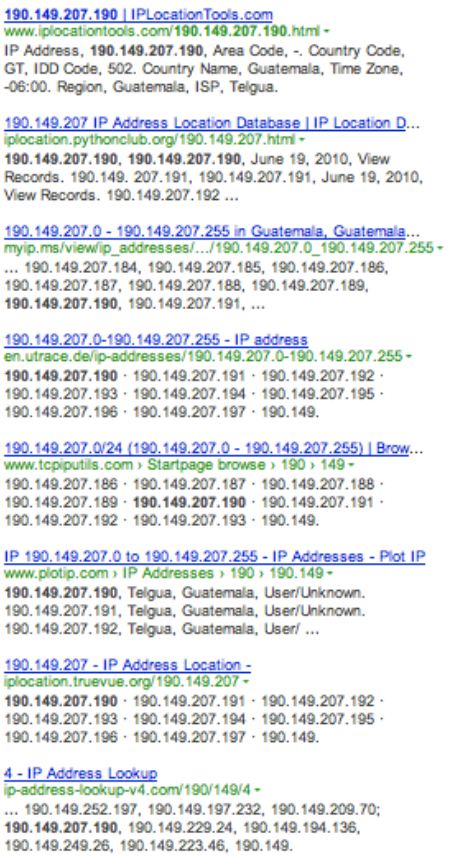
a. ***Spiders, robots y crawlers.*** Al buscar por el archivo robots.txt dentro de la raíz del sitio de la aplicación web se encontró el archivo con el siguiente contenido.

Figura 15: Reglas contenidas en robots.txt

```
User-agent: *
Crawl-delay: 10
# Directories
Disallow: /includes/
Disallow: /misc/
Disallow: /modules/
Disallow: /profiles/
Disallow: /scripts/
Disallow: /themes/
# Files
Disallow: /CHANGELOG.txt
Disallow: /cron.php
Disallow: /INSTALL.mysql.txt
Disallow: /INSTALL.pgsql.txt
Disallow: /install.php
Disallow: /INSTALL.txt
Disallow: /LICENSE.txt
Disallow: /MAINTAINERS.txt
Disallow: /update.php
Disallow: /UPGRADE.txt
Disallow: /xmlrpc.php
# Paths (clean URLs)
Disallow: /admin/
Disallow: /comment/reply/
Disallow: /filter/tips/
Disallow: /logout/
Disallow: /node/add/
Disallow: /search/
Disallow: /user/register/
Disallow: /user/password/
Disallow: /user/login/
# Paths (no clean URLs)
Disallow: /?q=admin/
Disallow: /?q=comment/reply/
Disallow: /?q=filter/tips/
Disallow: /?q=logout/
Disallow: /?q=node/add/
Disallow: /?q=search/
Disallow: /?q=user/password/
Disallow: /?q=user/register/
Disallow: /?q=user/login/
```

b. Descubrimiento por motores de búsqueda. Se ingresó la dirección IP sobre el motor de búsqueda Google, y se obtuvo únicamente resultados de sitios de localización de direcciones IP.

Figura 16: Resultados de búsqueda del sistema en Google



c. Huella digital. Para obtener la huella digital del servidor, se utilizó la herramienta NetCat para obtener el encabezado con el siguiente comando:

```
$ nc URLDELSISTEMA 80
HEAD / HTTP/1.0
Host: URLDELSISTEMA
```

d. ***Descubrimiento de aplicaciones.*** Para encontrar servicios y aplicaciones en ejecución en el servidor del sistema, se utilizó la herramienta NMAP para encontrar puertos abiertos en el servidor, con el comando

```
$ Nmap URLDELSISTEMA
```

e. ***Análisis de códigos de error.*** Se obtuvieron dos pruebas de error para analizar, obteniendo cada uno de la siguiente manera:

- 404 Not found: Buscando un archivo o una ruta inexistente en el sitio. En este caso se intentó ingresar a la dirección: http://URLDELSISTEMA/asd
- 403 Access denied: Intentando ingresar a una página o ruta no autorizada. En este caso se intentó ingresar a la dirección: http://DIRECCIÓNIPDELSISTEMA/?q=admin

2. Pruebas de gestión de configuración

a. ***Receptor de base de datos.*** Para encontrar si existe un receptor de base de datos en escucha, se utilizó la herramienta *netstat*, ejecutando el siguiente comando:

```
Netstat –lp
```

El parámetro –l es incluido para obtener los *sockets* que se encuentran en escucha, y el parámetro –p para obtener el nombre de la aplicación o servidor que está ocupando el *socket*.

Figura 17: Aplicaciones en escucha

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 *:ssh                   *:*                     LISTEN      -
tcp        0      0 *:www                   *:*                     LISTEN      -
```

Como se muestra en la Figura 17: Aplicaciones en escucha no se encontró ningún receptor en escucha de parte de la base de datos.

b. Enumeración de métodos HTTP y XST. Para encontrar los métodos HTTP permitidos, se utiliza la herramienta Netcat, para obtener la enumeración de las opciones habilitadas por el servidor web, con el siguiente comando.

```
$nc  URLDELSISTEMA 80
OPTIONS / HTTP/1.1
Host: URLDELSISTEMA
```

Resultado:

```
HTTP/1.1 501 Method Not Implemented
Date: Mon, 22 Apr 2013 13:20:09 GMT
Server: Apache/2.2.22 (Ubuntu)
Allow: POST,OPTIONS,GET,HEAD
Vary: Accept-Encoding
Content-Length: 299
Connection: close
```

Se puede observar en los resultados, que los métodos habilitados son: POST, OPTIONS, GET y HEAD. Para asegurar que los métodos que pueden ser explotados realmente no se encuentren habilitados, se lleva a cabo la ejecución de cada uno de ellos:

Prueba del método DELETE:

```
$nc URLDELSISTEMA 80
DELETE / HTTP/1.1
Host: URLDELSISTEMA
```

Resultado

```
HTTP/1.1 405 Method Not Allowed
Date: Thu, 20 Jun 2013 17:32:59 GMT
Server: Apache/2.2.22 (Ubuntu)
Allow: OPTIONS,GET,HEAD,POST
Vary: Accept-Encoding
Content-Length: 315
Content-Type: text/html; charset=iso-8859-1
```

Prueba del método PUT:

```
$nc URLDELSISTEMA 80
```

```
PUT / HTTP/1.1
Host: URLDELSISTEMA
```

Resultado

```
HTTP/1.1 405 Method Not Allowed
Date: Thu, 20 Jun 2013 17:34:28 GMT
Server: Apache/2.2.22 (Ubuntu)
Allow: OPTIONS,GET,HEAD,POST
Vary: Accept-Encoding
Content-Length: 312
Content-Type: text/html; charset=iso-8859-1
```

Prueba del método TRACE:

```
$nc URLDELSISTEMA 80
TRACE / HTTP/1.1
Host: URLDELSISTEMA
```

Resultado

```
HTTP/1.1 405 Method Not Allowed
Date: Thu, 20 Jun 2013 17:13:06 GMT
Server: Apache/2.2.16 (Debian)
Allow:
Vary: Accept-Encoding
Content-Length: 302
Content-Type: text/html; charset=iso-8859-1
```

3. Pruebas de autenticación

a. Transferencia de credenciales sobre un medio encriptado. Para realizar la prueba, se utiliza la herramienta WebScarab para interceptar la petición realizada por el navegador del usuario al intentar autenticarse.

Para que WebScarab intercepte las peticiones, es necesario utilizar la herramienta como un proxy en el navegador de internet. Para ellos se utilizó el navegador Mozilla Firefox, con la extensión FoxyProxy para manejar servidores proxy.

Una vez teniendo WebScarab para interceptar las peticiones, y el explorador utilizándolo como un servidor proxy, se prosiguió a autenticarse.

WebScarab intercepta la petición, mostrando los credenciales decodificados que se utilizaron para autenticarse.

b. ***Enumeración de usuarios.*** Para esta prueba se intentó realizar una autenticación con un usuario existente utilizando una contraseña incorrecta, y de esta manera verificar el mensaje de error que devuelve el sistema de autenticación.

c. ***Cuentas de usuario predeterminados.*** Para esta prueba se realizaron intentos de autenticación utilizando diferentes combinaciones de usuario y contraseña comunes que generalmente son las predeterminadas en varios sistemas.

d. ***Fuerza bruta.*** Para realizar la prueba de fuerza bruta se utilizó la herramienta Hydra, con el comando:

```
hydra -l usuario -P ArchivoConCadenasParaContraseña.txt -s 80
          URLDelSistema http-post-form
          "/node?destination=node:name=^USER^&pass=^PASS^&form_id
          =IdDelFormulario&form_build_id=IdBDelFormulario:Mensaje
          DeAutenticacionFallida."
```

En donde el parámetro -l es el nombre de usuario que se desea utilizar para realizar los intentos, cambiando el parámetro –l o –L, es posible utilizar un archivo .txt con todas las cadenas con las que se desea intentar.

El parámetro –P es el archivo con las cadenas a probar para la contraseña.

Los parámetros *form_id* y *form_build_id* son obtenidos del código fuente de la página, parte del formulario de autenticación.

Para el último parámetro, se utiliza el mensaje de autenticación fallida, el cuál es usado por la herramienta para detectar una autenticación fallida.

Para este caso se utilizó la siguiente línea de comando, con los parámetros específicos para realizar la prueba en el sistema.

```
hydra -l admin -P wordlist.txt URLDelSistema http-post-form
        "/seccatid_uvg/node?destination=node:name=^USER^&pass=^
        PASS^&form_id=user_login&form_build_id=form-
        717f2118605e72f82ee52e65b71fa061:Sorry, unrecognized
        username or password."
```

Los diccionarios utilizados para la contraseña se obtuvieron del sitio: http://download.openwall.net, el cual ofrece diccionarios de las palabras o cadenas más comunes utilizadas en contraseñas.

Iniciando la prueba con Hydra, se obtuvieron los siguientes falsos resultados positivos.

```
Hydra (http://www.thc.org/thc-hydra) starting at 2013-06-25 17:57:46
[DATA] 16 tasks, 1 server, 3546 login tries (l:1/p:3546), ~221 tries per task
[DATA] attacking service http-post-form on port 80
[80][www-form] host:              login: admin   password:
[80][www-form] host:              login: admin   password: 12345
[80][www-form] host:              login: admin   password: 123456
[80][www-form] host:              login: admin   password: password1
[80][www-form] host:              login: admin   password: password
[80][www-form] host:              login: admin   password: 123456789
[80][www-form] host:              login: admin   password: 1234567890
[80][www-form] host:              login: admin   password: 12345678
[80][www-form] host:              login: admin   password: abc123
[80][www-form] host:              login: admin   password: computer
[80][www-form] host:              login: admin   password: tigger
[80][www-form] host:              login: admin   password: qwerty
[80][www-form] host:              login: admin   password: 1234
[80][www-form] host:              login: admin   password: money
[80][www-form] host:              login: admin   password: carmen
[80][www-form] host:              login: admin   password: mickey
1 of 1 target successfully completed, 16 valid passwords found
```

Según los resultados, se obtuvieron 16 resultados válidos, esto es debido a que se encontraron las primeras autenticaciones válidas en los primeros 16 intentos simultáneos.

Ingresando al sitio del sistema, se obtuvo el siguiente mensaje:

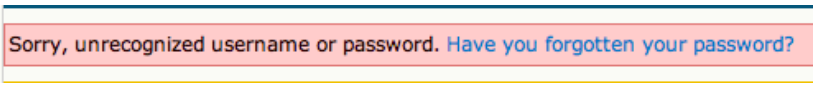
Con este mensaje se concluye que los resultados positivos fueron

falsos, debido a que la dirección IP del cliente en el que se realizó la prueba fue bloqueada tras los primeros intentos. Debido al despliegue de un mensaje diferente a la de una autenticación fallida, la herramienta los detecta como autenticaciones válidas.

e. ***Salto de esquema de autenticación.*** Para realizar esta prueba, se intenta ingresar a una sección o página que requiere autenticación de una manera directa ingresando la dirección URL conocida desde el explorador. En este caso se intentó ingresar a la página http://*URLDELSISTEMA* /?q=storm/projects

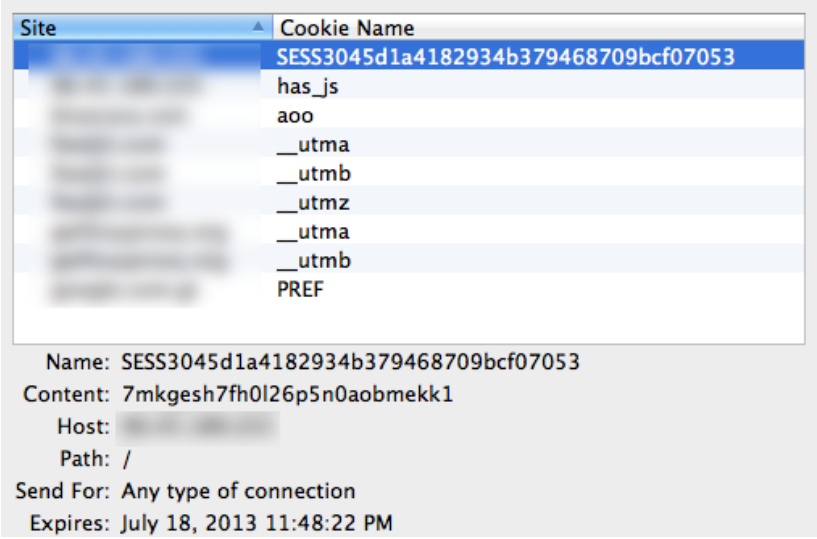
f. ***Restablecimiento de contraseñas.*** Para realizar esta prueba se intento restablecer la contraseña por medio del mecanismo que ofrece el sistema de autenticación.

Para acceder a este mecanismo, es necesario realizar una autenticación fallida previamente. El vínculo para restablecer la contraseña aparece junto al mensaje de autenticación fallida.



g. ***Cierre de sesión.*** Para la prueba de cierre de sesión, se inició sesión con un usuario existente y por medio del navegador, obtener el cookie de la sesión iniciada.

Figura 18: Cookie de sesión



El primer valor de los cookies pertenecientes al sitio del sistema, el primer valor es de la sesión actual, identificado con un nombre con el prefijo *SESS.*

Se cierra la sesión, y se utiliza Webscarab para interceptar las peticiones del navegador.

Se prosiguió a ingresar a un sitio que requiere autenticación, y se modificó la petición interceptada, para ingresar la sesión obtenida anteriormente.

El cookie de la sesión ya no es válida, por lo que devuelve un mensaje de acceso denegado.

h. ***CAPTCHA.***Esta prueba consistió en encontrar la existencia de algún mecanismo de CAPTCHA en el sistema.

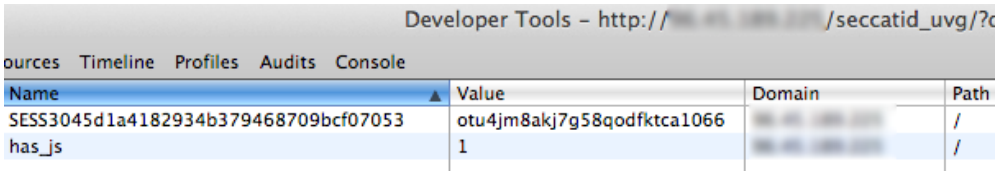
No se encontró ningún mecanismo de CAPTCHA en el proceso de autenticación.

4. Pruebas de gestión de sesiones

a. ***Fijación de sesión.*** Para llevar a cabo esta prueba, se inició sesión con un usuario en el sitio, y se obtuvo el valor del cookie del sitio desde el explorador.

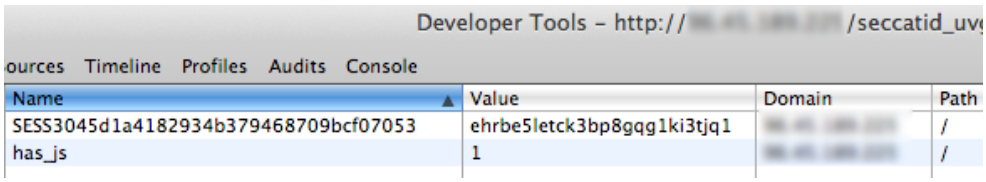
El valor del cookie de la sesión obtenida fue la siguiente:

Figura 19: Valor de cookie de sesión



Finalizando y volviendo a iniciar la sesión con el mismo usuario, se obtiene un valor del cookie de la sesión diferente.

Figura 20: Valor de cookie de sesión nueva



Dado que el sistema de sesiones genera un nuevo cookie para la sesión, no existe vulnerabilidad de fijación de sesión.

b. ***Cross-Site Request Forgery.*** Para llevar a cabo esta prueba, se inicia obteniendo el URL de una petición que requiera autenticación y/o privilegios. Se utilizó el URL para la eliminación de una acción estratégica.

En un navegador con la sesión iniciada en el sistema, se ingresó el URL obtenido, el sistema responde con la página de confirmación para la eliminación de la acción estratégica.

Dado que entre los parámetros enviados por POST para la eliminación de elementos se encuentra el *form-build-id*, el cuál es generado por el formulario que manda la petición para realizar la acción, se puede deducir que el sistema

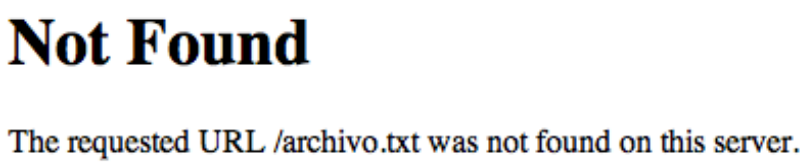
requiere que se envíe este parámetro válido para completar la acción.

5. Pruebas de autorización

a. ***Ruta transversal.*** En esta prueba se intentó acceder a archivos existentes en el servidor, utilizando la ruta transversal como parte del URL en el navegador. En este caso se utilizó el URL para acceder a un archivo existente en el directorio padre de la raíz del directorio en que se encuentra el sistema. http://URLDELSISTEMA/../archivo.txt

El sistema devuelve un error 404, de archivo no encontrado, y eliminando la ruta transversal, quedando el URL http://URLDELSISTEMA/archivo.txt .

Figura 21: Error de archivo no encontrado al utilizar ruta transversal



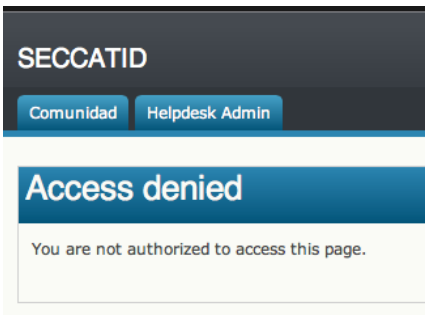
b. ***Salto de esquema de autorización.*** Para la prueba de salto de esquema de autorización, se intentó ingresar a una página que es visible exclusivamente para usuarios autenticados. En esta caso se intentó acceder a la dirección: http://URLDELSISTEMA/?q=storm/projects

El sistema devuelve un mensaje de error de acceso denegado.

c. ***Escalamiento de privilegios.*** Para el escalamiento de privilegios se inició sesión como un usuario sin privilegios en este caso el usuario *user, el cuál no posee privilegios para acceder la página administrativa*, y se intentó acceder a la sección de administración ubicado en http://URLDELSISTEMA/?q=admin

El sistema devuelve un mensaje de error de acceso denegado.

Figura 22: Acceso denegado



6. Pruebas de validación de datos

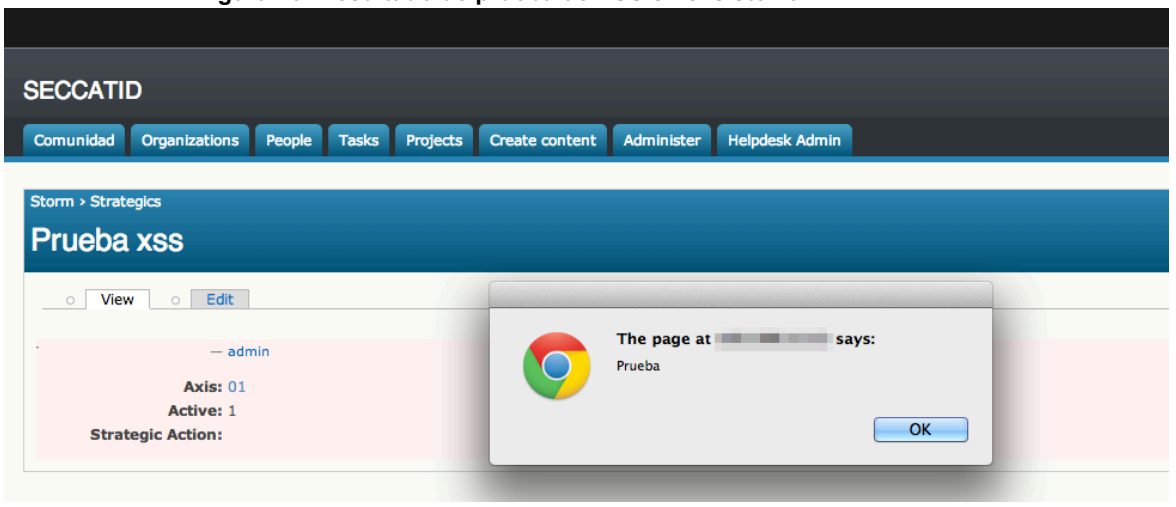
a. ***Cross site scripting.*** Para esta prueba, se identificaron los formularios con campos que permiten el ingreso de etiquetas HTML.

Para estos campos se utilizó como prueba el siguiente script:

```
<script>alert('Prueba');</script>
```

La prueba fue realizada los campos, y se encontró la vulnerabilidad en el campo de descripción, en la sección de creación de acción estratégica.

Figura 23: Resultado de prueba de XSS en el sistema



b. ***Inyección SQL.***En esta prueba se inició identificando los campos que pueden ser utilizados para realizar consultas SQL en el sistema. No se encontraron campos que puedan realizar consultas de búsqueda, pero sí se identificaron campos utilizados para insertar datos por medio de consultas SQL.

Se realizó la prueba de ingresar una comilla simple (‘) o comillas dobles (“) sobre cada uno de los campos de creación de contenido, y en los campos de autenticación.

No se obtuvo ningún error de consulta de SQL.

c. ***Inyección ORM.***Sobre los mismos campos identificados para la prueba de inyección SQL, se realizó la prueba ingresando como texto *OR 1-- .*

No se obtuvo ningún error.

d. ***Inyección XML.***Sobre los mismos campos identificados para la prueba de inyección SQL, se realizó la prueba ingresando como texto una comilla simple (‘).

No se obtuvo ningún error.

e. ***Inyección SSI.***Esta prueba se realizó en los mismos campos identificados para la prueba de *Cross-site scripting*, ingresando la siguiente línea como contenido:

```
<!--#echo var="DATE_LOCAL" -->
```

El cuál obtiene la fecha y hora local del servidor.

No se obtuvo ningún resultado de parte del código ingresado.

7. Pruebas de denegación de acceso

a. Ataques con comodines SQL. Esta prueba se inició identificando campos para realizar búsquedas en el sistema.

Entre los campos encontrados, no se encontró ningún campo que permite realizar búsquedas de palabras sobre la base de datos, únicamente campos para filtrar información a nivel de la aplicación.

Estos campos no son vulnerables a este tipo de ataques.

b. Congelación de cuentas de usuario. Para esta prueba se realizaron intentos de autenticación con un mismo nombre de usuario, tras 20 intentos, la dirección IP del cliente en el que fue realizada la prueba fue bloqueada.

La cuenta de usuario con la que se intentó ingresar no fue bloqueada.

c. Desbordamiento de búfer. Esta prueba consistió en ingresar cadenas largas de caracteres sobre diferentes campos de formularios encontrados en el sistema, incluyendo campos no visibles.

Para los campos no visibles, se utilizó WebScarab para interceptar las peticiones POST, y modificar el contenido que se envía en estos parámetros por cadenas más largas.

XIV. Glosario

- Apache: Servidor web desarrollado y mantenido por una comunidad abierta de desarrolladores.
- CAPTCHA: Mecanismo para verificar que el usuario es humano. Consiste en mostrar una imagen con letras distorsionadas y solicita que el usuario ingrese las letras que ve.
- CCATID: Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas.
- CISSP: Profesional de Seguridad de Sistemas de Información Certificado, es una certificación de seguridad informática independiente manejado por la ISC^2.
- Cookie: Pequeño fragmento de información que envía un sitio web, el cuál es almacenado por el navegador.
- Crawler: Bot que navega a través sitios web automáticamente.
- CRUD: Create, Read, Update, Delete. Operaciones básicas en una base de datos.
- CSRF: Falsificación de Petifición en Sitios Cruzados, ataque en que se fuerza al explorador web de la víctima a enviar una petición a una aplicación web, realizando una acción utilizando las credenciales de la víctima.
- Dirección IP: Número que identifica a un interfaz de dispositivo que se encuentra conectado a una red que utiliza el protocolo IP.
- HTML: Lenguaje de Marcado Hiper-Textual.
- HTTP: Protocolo de Transferencia de Hiper-Texto. Protocolo utilizado en las transacciones de la World Wide Web.
- ISC^2: Consorcio Internacional de Certificación de Seguridad de Sistemas de Información.
- ISECOM: Instituto de Seguridad y Metodologías Abiertas. Organización de seguridad responsable de publicar el manual de metodologías de pruebas de seguridad OSSTMM.
- Motor de búsqueda: Sistema automatizada que busca información y archivos en servidores web encontrados por un crawler.

- OCR: Reconocimiento Óptico de Caracteres: Proceso de digitalización de textos automatizado.
- ORM: Mapeo Objeto-Relacional. Técnica de programación para convertir datos de un sistema de tipos de lenguaje de programación orientado a objetos a una base de datos relacional.
- OSSTMM: Manual de Metodología de Pruebas de Seguridad Abierto. Guía de pruebas de seguridad publicado por la organización de seguridad informática ISECOM.
- OWASP: Proyecto Abierto de Seguridad de Aplicaciones Web. Proyecto de código abierto que tiene como objetivo encontrar las causas que hacen que el software sea inseguro.
- Proxy: Servidor que intercepta conexiones de red que un cliente realiza a un servidor destino.
- SECCATID: Secretaría de la Comisión Contra las Adicciones y el Tráfico Ilícito de Drogas
- SQL: Lenguaje de Consulta Estructurado. Es un lenguaje de acceso a bases de datos relacionales, permite especificar operaciones a realizar en ellas.
- SSL: Capa de Conexión Segura. Protocolos criptográficos que ofrecen comunicaciones seguras por una red.
- Ubuntu: Sistema operativo Linux basado en Debian.
- W3C: Consorcio de la World Wide Web. Consorcio internacional que da produce para la World Wide Web.
- WebScarab: Herramienta de pruebas de seguridad de aplicaciones web, funciona como un proxy para interceptar peticiones de un explorador y respuestas del servidor web.
- XML: Lenguaje de Marcas Extensible, es un lenguaje de marcas desarrollado por el W3C.
- XST: Secuencia de Comandos en Sitios Cruzados. Es un agujero de seguridad, que permite al atacante inyectar código en una aplicación web para ser ejecutado.