Vulnerability Analysis Oriented to Business Logic of Web Applications. A Case Study

William-Rogelio Marchand-Niño[1]
william.marchand@unas.edu.pe

Ana Claudia Terrones Murga[1]
ana.terrones@unas.edu.pe

***Resumen*— El objetivo de este artículo es mostrar un proceso de análisis de vulnerabilidades de aplicaciones web orientada a la lógica de negocio, donde además de la identificación de vulnerabilidades tecnológicas, se debe prestar atención en aquellas relacionadas a las reglas de negocio que no son detectadas por herramientas automatizadas pero que generalmente son de mayor impacto, por lo que es importante la capacidad de análisis del pentester para identificarlas. Este tipo de vulnerabilidades están contempladas en la Guía de Pruebas OWASP v4 (*Open Web Application Security Project*) pero su aplicación es específica y diferenciada en cada situación particular. Para el caso de estudio de la investigación se consideraron algunas aplicaciones de una universidad con autorización previa.**

Palabras clave*— *pentesting, aplicaciones web, análisis de vulnerabilidades, cvss, owasp.

I. INTRODUCCIÓN

En un proceso de análisis de vulnerabilidades a través de ataques informáticos autorizados, busca verificar la existencia de vulnerabilidades a nivel de hardware, software y personas, para una posterior explotación para evaluar su impacto y determinar las acciones correctivas. El resultado ayuda a estimar un plan de mitigación para cerrar estas brechas en previsión de un ataque real, y así evitar pérdidas financieras y de información. [1]. También es importante resaltar que la información sobre las vulnerabilidades encontradas durante la prueba de penetración es confidencial y no debe divulgarse hasta que se resuelva por completo [2]. Abordar un análisis de vulnerabilidades y pruebas de penetración puedan darse en un contexto de caja negra (sin información del objetivo), caja blanca (con información completa del objetivo) o caja gris (con información parcial del objetivo) [3].

Asimismo, las organizaciones en estos tiempos no pueden sentirse seguros en el ciberespacio (Internet), puesto que no están libres de ataques contra sus infraestructuras, aplicaciones, usuarios y diversos sistemas digitales [4], por lo que es importante e imprescindible analizar el nivel de vulnerabilidades que puedan presentar los componentes de los sistemas de información (infraestructura tecnológica, aplicaciones) que soportan y procesan los activos de información, además de valorar el impacto que un ataque causaría en los mencionados sistemas [5].

Es importante señalar que existen bases metodológicas para aplicar un análisis de vulnerabilidades tales como:

- Open Web Application Security Project (OWASP)
- Open Source Security Testing Methodology Manual (OSSTMM)
- Vulnerability Assessment and Mitigation (VAM)

El trabajo de investigación se orienta principalmente al análisis de vulnerabilidades no tecnológicas, es decir las relacionadas a la lógica de negocio, que implica un conocimiento preciso de las funciones de las aplicaciones y sitios web en el sentido de responder a la pregunta ¿qué es lo que hace la aplicación y qué no debería hacer? Para responder a esta interrogante se realizaron pruebas de identificación y comprobación de las vulnerabilidades de forma manual con ayuda de algunas herramientas. En estos casos las herramientas automatizadas no son de gran ayuda para la detección de vulnerabilidades orientadas a la lógica de negocio.

Para llevar a cabo las pruebas se consideró la aplicación de los lineamientos de la Guía OWASP, OSSTMM y VAM.

En las siguientes secciones se describen con mayor detalle las metodologías, el sistema de puntuación CVSS para determinar el nivel de severidad de las vulnerabilidades, el detalle de las pruebas aplicadas, los resultados y las conclusiones.

II. METODOLOGÍAS PARA EL ANALISIS DE VULNERABILIDADES

En esta sección se describen brevemente las metodologías utilizadas en el trabajo de investigación.

A. Open Web Application Security Project (OWASP)

Es un proyecto abierto dedicado a identificar y combatir las causas que hacen que el software sea inseguro, y están plasmados en documentos como la Guía de Pruebas OWASP y el Top 10 de OWASP ampliamente adoptados *[6] [7]*. El método de prueba para aplicaciones web se basa en dos fases:

- Fase Pasiva: Es una prueba para comprender la lógica de la aplicación y verificar elementos que podrían significar una puerta abierta para un análisis detallado. Las herramientas se pueden utilizar para capturar información, por ejemplo, un proxy HTTP que observa todas las solicitudes y respuestas HTTP. Al final de esta fase, el *pentester* conocerá todos los puntos de acceso de la aplicación, tales como encabezados HTTP, parámetros, cookies, etc.
- Fase Activa: En esta fase, el *pentester* comienza a realizar las pruebas utilizando la metodología descrita en la Guía. El conjunto de pruebas activas se ha dividido en 11 categorías para un total de 91 controles.

[1]Grupo de Investigación en Redes, Seguridad y Gestión de TI
Universidad Nacional Agraria de la Selva - Tingo María, Perú

https://doi.org/10.12804/si9789587844337.15

B. *Open Source Security Testing Methodology Manual (OSSTMM)*

El Manual de Metodología de Pruebas de Seguridad de Código Abierto (OSSTMM) es una metodología para probar la seguridad operativa de ubicaciones físicas, flujo de trabajo, pruebas de seguridad humana, pruebas de seguridad física, pruebas de seguridad inalámbricas, pruebas de seguridad de telecomunicaciones, pruebas de seguridad de redes de datos y cumplimiento *[8]*.

El análisis de vulnerabilidades en aplicaciones web se pueden realizar de forma automatizada o manual con apoyo de algunas herramientas. Entre las herramientas más utilizadas para ejecutar pruebas automatizadas en el contexto de un escaneo y análisis de vulnerabilidades se incluyen a OWASP ZAP Proxy, Brup Suite, Acunetix, Nessus, entre otras [9] [10] [11].

La identificación de vulnerabilidades tecnológicas con el uso de herramientas automatizadas no es suficiente, por lo que se debe recurrir a un análisis manual que implica generalmente el conocimiento de la lógica de negocio en las aplicaciones web y que muchas veces pueden tener un mayor impacto y su solución puede ser más compleja [6] [9].

Pero ¿qué es una vulnerabilidad? Una vulnerabilidad es una falla en un sistema, que permite a un atacante comprometer su confidencialidad, integridad o disponibilidad. Las pruebas de penetración de aplicaciones web implican técnicas que conducen a la identificación de vulnerabilidades potenciales, que pueden comprometer las aplicaciones web [9]. Las vulnerabilidades de aplicaciones pueden catalogarse de acuerdo con la Guía OWASP, siendo las más comunes las siguientes [6] [12]:

- SQL Injection (SQLi)
- Cross site scripting (XSS), existen dos tipos básicos: XSS reflejado y XSS almacenado.
- Escalamiento de Privilegios, tanto horizontal como vertical.
- Secuestro de sesiones, generalmente por medio cookies.
- Insuficiente tiempo de expiración de sesiones.
- Directorios transversales. Acceso no autorizado a directorios y archivos no autorizados.
- Fallas en la validación de entradas.
- Referencia Directa Insegura a Objetos (IDOR)
- Clickjacking. Técnica maliciosa para engañar a un usuario para que haga clic en páginas web aparentemente inofensivas.

C. *Vulnerability Assessment and Mitigation (VAM)*

Para el análisis de vulnerabilidades también existe otra metodología como VAM (*Vulnerability Assessment and Mitigation*) que establece seis etapas generales [13]:

- Etapa 1. Identificar las funciones de información esenciales de la organización.
- Etapa 2. Identificar los sistemas de información esenciales para implementar las funciones esenciales identificadas en la etapa 1.
- Etapa 3. Identificar las vulnerabilidades de los sistemas esenciales identificados en la etapa 2.
- Etapa 4. Identificar las técnicas de seguridad pertinentes para mitigar las vulnerabilidades identificadas en la etapa 3.
- Etapa 5. Seleccionar y aplicar las técnicas de la etapa 4 basadas en restricciones, costos y beneficios.
- Etapa 6. Probar las técnicas aplicadas en la etapa 5 para determinar la robustez y las posibilidades reales bajo amenaza.

De las seis etapas planteadas, las que se alinean con el procedimiento empleado en el trabajo de investigación fueron las etapas 3, 4, 5 y 6, puesto que el objetivo del estudio fue evidenciar las vulnerabilidades de tipo no tecnológico, más allá de realizar un análisis de los activos y sistemas de información que bien puede ser parte de los trabajos futuros.

III. Medición de Severidad de Vulnerabilidades con CVSS

Para la medición de la severidad de las vulnerabilidades se hace uso del CVSS (Common Vulnerability Scoring System) que es un framework que incluye las características más significativas para evaluar las vulnerabilidades de software y hardware. Este framework ha sido desarrollado por First.org Inc. cuya finalidad es la de ayudar a los equipos de respuesta a incidentes de seguridad informática.

El CVSS consta de tres grupos de métricas, las cuales son, métricas base, métricas temporales y métricas ambientales. El grupo Base representa las cualidades intrínsecas de una vulnerabilidad que son constantes en el tiempo, el grupo Temporal refleja las características de una vulnerabilidad que cambian con el tiempo y el grupo Ambiental representa las características de una vulnerabilidad que son exclusivas en el entorno de una organización [14].

Para esta investigación, la determinación de la puntuación de severidad se aplicó el grupo de métricas base que se componen de las métricas de explotabilidad y las métricas de impacto (ver Fig. 1). El resultado final produce una puntuación que oscila entre 0 y 10 y la categorización se puede observar en la Tabla I.

TABLA I. Niveles de severidad CVSS

Nivel de severidad de vulnerabilidad	Puntuación CVSS
Ninguno	0.0
Bajo	0.1 – 3.9
Medio	4.0 – 6.9
Alto	7.0 – 8.9
Crítico	9.0 – 10.0

El componente de las métricas de explotabilidad refleja la facilidad de explotación de una vulnerabilidad y el componente de métricas de impacto expresan la consecuencia directa de la explotación exitosa de la vulnerabilidad.

A continuación, se explican brevemente las métricas utilizadas en la determinación del nivel de severidad de las vulnerabilidades identificadas.

- **Vector de ataque**. Esta métrica refleja el contexto por la que es posible la explotación de una vulnerabilidad (red, adyacente, local o físico). El valor será mayor

cuanto más remoto sea posible explotar la vulnerabilidad.

- **Complejidad de ataque**. Describe las condiciones que deben existir para explotar la vulnerabilidad. Puede ser baja si no existen condiciones de acceso especializadas, o alta si requiere que el atacante invierta una cantidad considerable de esfuerzo en la preparación y ejecución contra el componente vulnerable.
- **Privilegios requeridos**. Refleja el nivel de privilegios que debe tener un atacante antes de explotar con éxito la vulnerabilidad. Las alternativas para esta métrica son ninguno, bajo o alto.
- **Interacción con el usuario**. Esta métrica determina si la vulnerabilidad puede explotarse exitosamente a voluntad del atacante o requiere la interacción de un usuario de alguna manera como descargar algún archivo, hacer click en algún enlace, ejecutar o abrir algún documento, etc.
- **Alcance**. Esta métrica determina si una vulnerabilidad explotada impacta o no en recursos que están más allá del componente vulnerable y el alcance de su seguridad.
- **Confidencialidad**. Esta métrica mide el impacto en la confidencialidad de los activos de información debido a una vulnerabilidad explotada con éxito. El posible valor es alto, bajo o ninguno.
- **Integridad**. Esta métrica mide el impacto en la integridad de los activos de información producto de la explotación exitosa de una vulnerabilidad. El valor que puede tomar es alto, bajo o ninguno.
- **Disponibilidad**. Esta métrica mide el impacto en la disponibilidad de los activos de información producto de la explotación exitosa de una vulnerabilidad El valor que puede tomar es alto, bajo o ninguno.

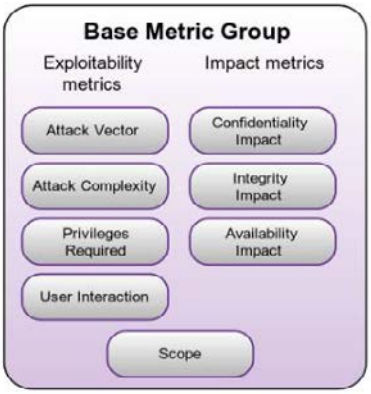


Fig. 1. Grupo de métricas base [14]

IV. Desarrollo del Análisis De Vulnerabilidades Orientado a la Logica de Negocio.

Las pruebas presentadas a continuación fueron realizadas según la guía de pruebas OWASP v4 y están orientadas a evaluar la lógica de negocio en las aplicaciones y sitios web.

La Tabla II muestra la lista de controles que se evaluaron en las aplicaciones web.

TABLA II. Pruebas Realizadas En Las Aplicaciones Web

ID de Prueba	Nombre de la Prueba
Recopilación de información	
OTG-INFO-005	Revisión de los comentarios de la página web y los metadatos para detectar fugas de información.
Pruebas de gestión de identidad	
OTG-IDENT-004	Prueba de enumeración de cuenta de usuario predecible
Pruebas de autenticación	
OTG-AUTHN-001	Prueba de credenciales transportadas a través de un canal encriptado
OTG-AUTHN-002	Prueba de credenciales por defecto
OTG-AUTHN-003	Prueba de mecanismo de bloqueo débil
OTG-AUTHN-004	Pruebas para eludir el esquema de autenticación
Pruebas de gestión de sesión	
OTG-SESS-002	Prueba de atributos de cookies
OTG-SESS-003	Pruebas para la fijación de sesión
OTG-SESS-006	Prueba de funcionalidad de cierre de sesión
OTG-SESS-007	Tiempo de espera de sesión de prueba

TABLA III. Pruebas Realizadas En Los Sitios Web

ID de Prueba	Nombre de la Prueba
Configuración e implementación de pruebas de gestión	
OTG-CONFIG-005	Enumerar interfaces de administración de aplicaciones e infraestructura
Pruebas de autenticación	
OTG-AUTHN-003	Prueba de mecanismo de bloqueo débil

Realizadas las pruebas de acuerdo con la Tabla II y Tabla III, se identificaron como vulnerabilidades orientadas a la lógica de negocio a los siguientes hallazgos:

A. Credenciales débiles (OTG-AUTHN-002)

En muchas situaciones, cuando se crea una cuenta en una aplicación, se genera una contraseña predeterminada con algunas características estándar. Si esta contraseña es predecible y el usuario no la cambia en el primer acceso, esto puede hacer que un atacante obtenga acceso no autorizado a la aplicación. Este es el caso de una aplicación denominada Gestión Docente, cuya cuenta de administrador posee una contraseña débil (docente/password) tal como se muestra en la Fig. 2.

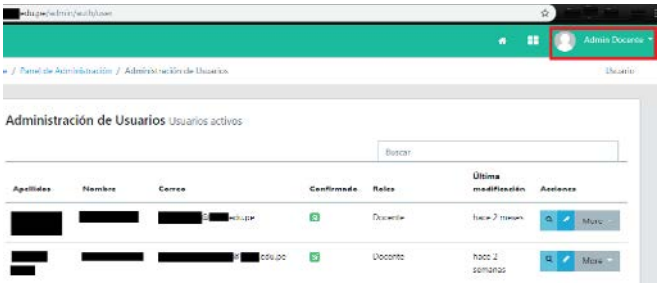


Fig. 2. Sesión con privilegios de administrador en la aplicación de Gestión Docente.

La puntuación determinada en el sistema CVSS es de 7.5 mostrada en la Tabla IV, lo cual indica un nivel de severidad alto, los factores (métricas) que contribuyen a una explotación exitosa de esta vulnerabilidad son: El vector de ataque es la red, es decir, es explotable de manera remota. No existen condiciones especializadas para su explotación, solo fue suficiente realizar una adecuada fase de reconocimiento donde se recolectaron correos electrónicos que sirvieron para elaborar un diccionario y encontrar alguna contraseña débil o por defecto. No se requirió ningún tipo de acceso ni privilegio

antes del ataque. Se puede realizar la explotación sin requerir la interacción de algún usuario. Hay una pérdida total de la confidencialidad, ya que con la explotación se logró el acceso como administrador quedando expuesta información confidencial de todos los usuarios de la aplicación.

TABLA IV. SEVERIDAD DE LA VULNERABILIDAD CREDENCIALES DÉBILES

Métrica	Valor
Vector de ataque	Red
Complejidad de ataque	Baja
Privilegios requeridos	Ninguno
Interacción con el usuario	Ninguno
Alcance	Sin cambios
Confidencialidad	Alto: Pérdida total de confidencialidad
Integridad	Ninguno
Disponibilidad	Ninguno
Puntuación Base CVSS v3.0	7.5

Para mitigar esta vulnerabilidad de contraseñas fáciles de predecir que facilitan el acceso no autorizado, existen dos soluciones: introducir controles de autenticación adicionales (es decir, autenticación de dos factores) o introducir una política de contraseña segura. El más simple y barato de estos es la introducción de una política de contraseñas sólida que garantice la longitud, complejidad, reutilización y antigüedad de las contraseñas, sin embargo, lo recomendable es agregar un factor adicional de autenticación (token, mensaje de texto, llamada por teléfono)

B. Mecanismo de bloqueo débil (OTG-AUTHN-003)

En esta prueba se verificó que el mecanismo de bloqueo no es efectivo, ya que presenta un temporizador aleatorio el cual no impide seguir realizando intentos, lo cual nos indica que la aplicación es susceptible a ataques de diccionario y fuerza bruta.

En la aplicación evaluada, las cuentas se bloquean después de 5 intentos fallidos de inicio de sesión y solo se pueden desbloquear después de un período de tiempo predeterminado o por intervención de un administrador. En los resultados de la prueba, que se muestran a continuación, se puede observar que en el sexto intento de inicio de sesión con un usuario válido y un password incorrecto, la aplicación envía un mensaje de bloqueo de la cuenta por 39 segundos (ver Fig. 3), luego, sin esperar el tiempo indicado se realizó un nuevo intento (séptimo intento) con los mismos datos y ahora se muestra un mensaje pero con un tiempo menor al anterior (Fig. 4), sucediendo lo mismo en el octavo intento (Fig. 5) y en el noveno intento ya no se mostró ningún mensaje de bloqueo (Fig. 6).

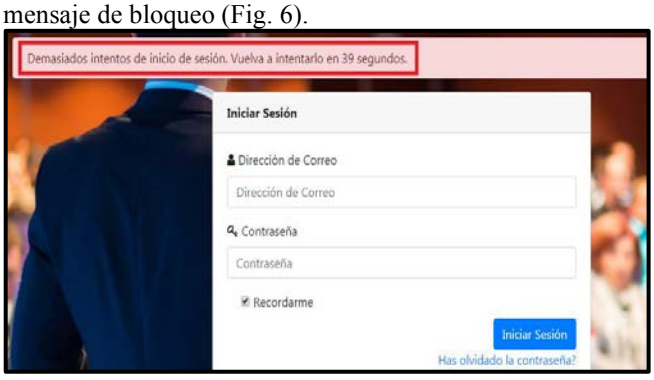


Fig. 3. Sexto intento fallido de inicio de sesión con usuario válido y contraseña incorrecta

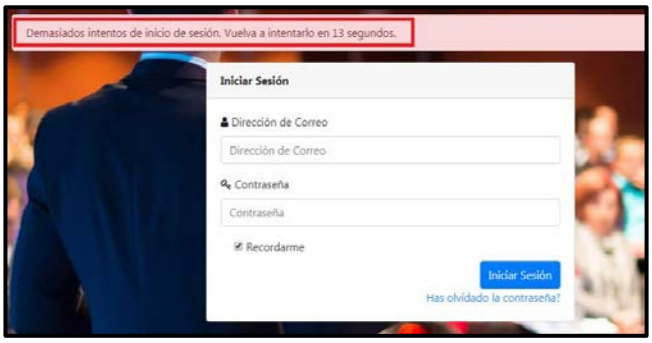


Fig. 4. Séptimo intento fallido de inicio de sesión con usuario válido y contraseña incorrecta

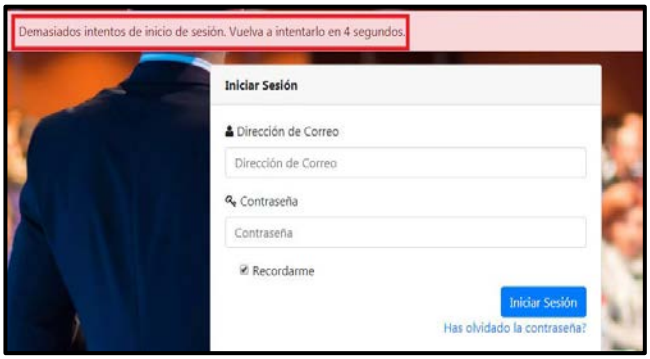


Fig. 5. Octavo intento fallido de inicio de sesión con usuario válido y contraseña incorrecta

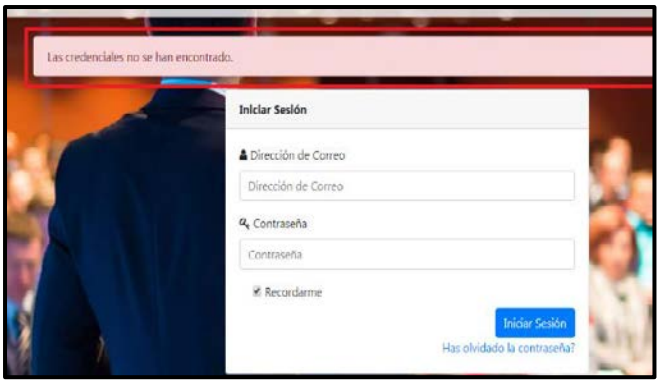


Fig. 6. Noveno intento fallido de inicio de sesión con usuario válido y contraseña incorrecta

La puntuación determinada en el sistema CVSS es de 7.5 tal como se muestra en la Tabla V, lo cual indica un nivel de severidad alto. Los factores (métricas) que contribuyen a una explotación exitosa de esta vulnerabilidad son, el vector de ataque es la red, que significa que es posible su explotación de forma remota. La complejidad de ataque es baja, porque no se requieren condiciones de acceso especiales para su explotación, solo realizar un ataque de *password spraying* para evadir el mecanismo de bloqueo. No se requiere ningún tipo de acceso ni privilegio antes del ataque. Se puede realizar la explotación sin requerir la interacción de algún usuario. En caso de una explotación exitosa, es decir, encontrar credenciales de acceso válidas, significaría una pérdida total de la confidencialidad, porque el atacante tendría acceso a información confidencial de los usuarios de la aplicación.

TABLA V. SEVERIDAD DE LA VULNERABILIDAD MECANISMO DE BLOQUEO DÉBIL

Métrica	Valor
Vector de ataque	Red
Complejidad de ataque	Baja
Privilegios requeridos	Ninguno
Interacción con el usuario	Ninguno
Alcance	Sin cambios
Confidencialidad	Alto
Integridad	Ninguno
Disponibilidad	Ninguno
Puntuación Base CVSS v3.0	7.5

Para mitigar esta vulnerabilidad se recomienda aplicar mecanismos de desbloqueo de cuentas en función del nivel de riesgo. En orden de menor a mayor seguridad: bloqueo y desbloqueo basado en el tiempo; desbloqueo de autoservicio (envía un correo electrónico de desbloqueo a la dirección de correo electrónico registrado); o desbloqueo manual por parte del administrador.

C. *Enumeración de usuarios y cuenta de usuario predecible (OTG-IDENT-004)*

A menudo, las aplicaciones web revelan cuándo existe o no un nombre de usuario en el sistema, ya sea como consecuencia de una configuración incorrecta o como una decisión de diseño. El propósito de esta prueba es verificar si es posible recopilar un conjunto de nombres de usuario válidos interactuando con el mecanismo de autenticación de la aplicación y los mensajes de errores o advertencias que emiten durante el proceso, o como en este caso de prueba a través del mecanismo de recuperación de contraseña y sus correspondientes mensajes. Un atacante puede utilizar los mensajes que se emiten por el proceso de recuperación de contraseña para obtener una lista de usuarios del sistema que luego pueden formar parte de un diccionario para ataques de fuerza bruta.

En la Fig. 7 se muestra el formulario de recuperación de contraseña donde se solicita la dirección de correo electrónico.

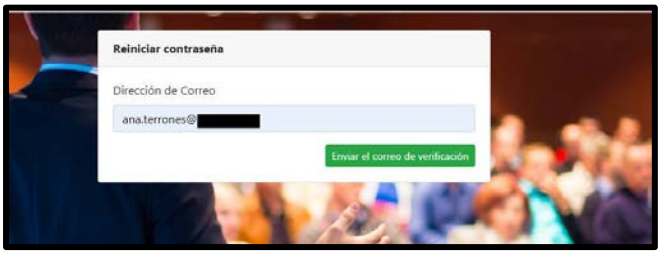


Fig. 7. Formulario de recuperación de contraseña.

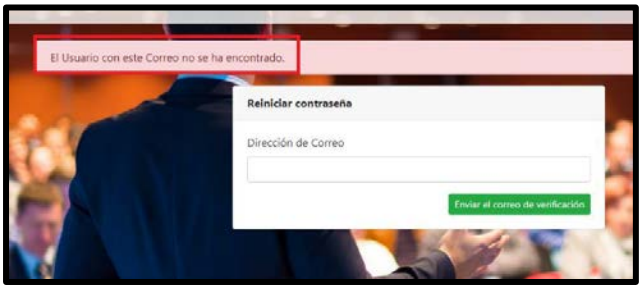


Fig. 8. Mensaje con ingreso de correo no válido.

Al ingresar un correo no válido en el formulario de recuperación de contraseña se muestra el mensaje "El usuario con este Correo no se ha encontrado" tal como se aprecia en la Fig. 8, esto es un indicador de la posibilidad de saber qué cuentas de correo electrónico son válidas o no, y para confirmar que se puede utilizar dichos mensajes para enumerar correos electrónicos válidos, se ingresa un correo válido tal como se muestra en la Fig. 9, y como se puede observar el mensaje "Le hemos enviado el enlace para el reinicio de la contraseña" es distinto al de un correo no válido, lo cual hace posible probar de forma manual o automatizada si una dirección de correo electrónico es válida o no. El propósito final de esta enumeración es la construcción de listas de usuarios válidos para ataques de ingeniería social o

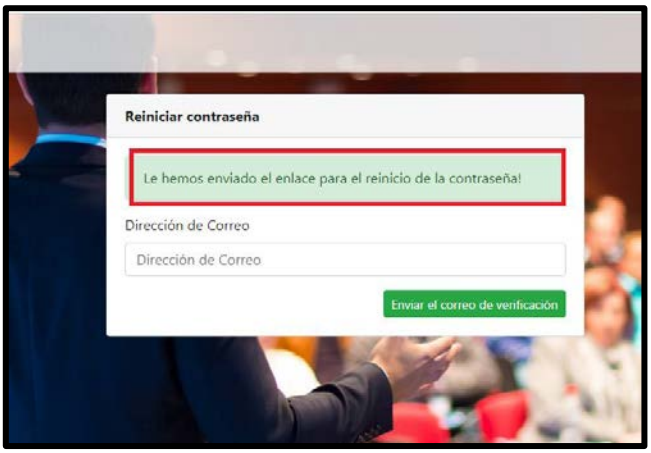


Fig. 9. Mensaje de recuperación de contraseña con ingreso de correo válido.

La puntuación determinada en el sistema CVSS es de 5.3 tal como se muestra en la Tabla VI, lo cual indica un nivel de severidad medio. Los factores (métricas) que contribuyen a una explotación exitosa de esta vulnerabilidad son el vector de ataque (red), es decir, es explotable de manera remota. No existen condiciones especializadas para su explotación, solo fue suficiente realizar una adecuada fase de reconocimiento donde se recolectaron correos electrónicos de otros sitios web del mismo dominio, los cuales sirvieron para enumerar usuarios válidos de la aplicación. No se requiere ningún tipo de acceso ni privilegio para realizar el ataque. Se puede realizar la explotación sin requerir la interacción de algún usuario. La pérdida de confidencialidad es considerada baja, porque el hecho de obtener usuarios del sistema no causa una pérdida directa o grave para la aplicación, pero cabe mencionar que esta vulnerabilidad es un factor agravante para las vulnerabilidades de mecanismo de bloqueo y credenciales débiles, ya que en conjunto ayudan al atacante a crear el escenario ideal para ataques de diccionario o fuerza bruta.

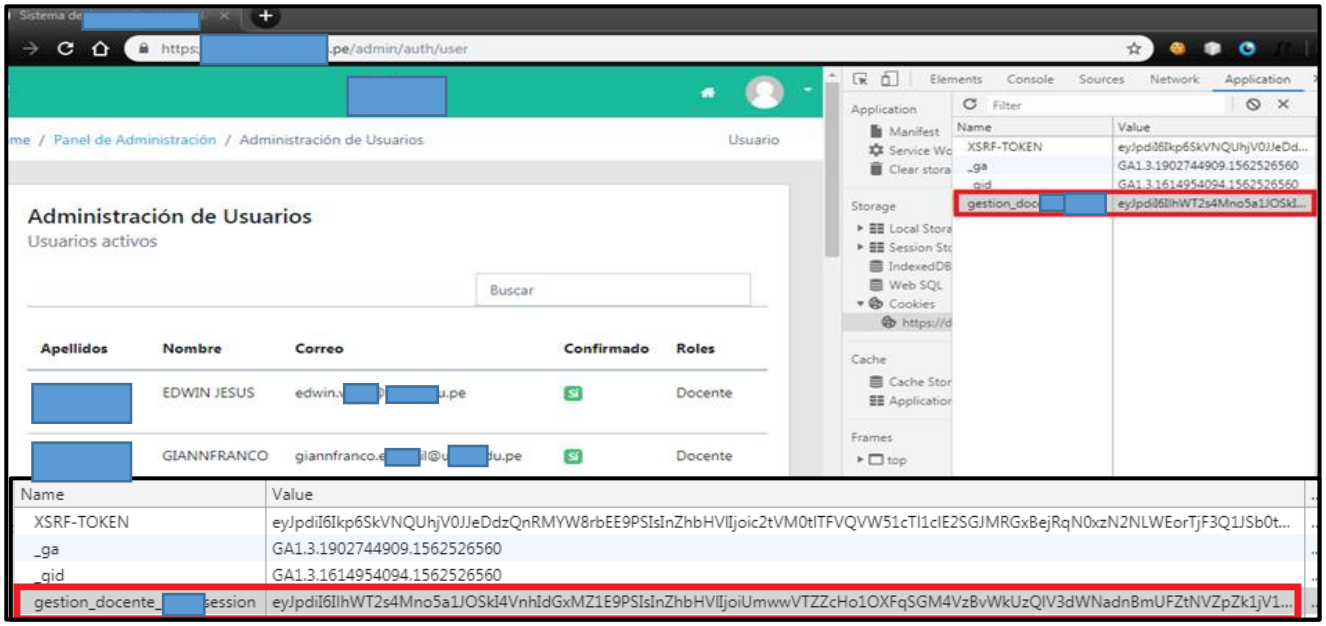
TABLA VI. SEVERIDAD DE LA VULNERABILIDAD DE ENUMERACIÓN DE USUARIOS

Métrica	Valor
Vector de ataque	Red
Complejidad de ataque	Baja
Privilegios requeridos	Ninguno
Interacción con el usuario	Ninguno
Alcance	Sin cambios
Confidencialidad	Baja
Integridad	Ninguno
Disponibilidad	Ninguno
Puntuación Base CVSS v3.0	**5.3**

Fig. 10. Cookie de la sesión original

D. Reproducción de cookies de autenticación (OTG-SESS-002)

Algunos sitios vulnerables simplemente están utilizando la cookie como token de autenticación sin tener que realizar más comprobaciones en el lado del servidor web, por lo tanto, para comprobar se inició sesión con credenciales válidas, la cual se copió la cookie, en otro navegador, se pega el valor copiado anteriormente en el campo correspondiente en la consola y como resultado se comprueba que es posible iniciar sesión solo con el valor de dicha cookie sin necesidad de ingresar un usuario y una contraseña, lo que hace que el token de autenticación sea vulnerable a los atacantes si las cookies son robadas por *malware* o inyección de JavaScript. Las Fig. 10 y 11 muestran la evidencia en diferentes navegadores.



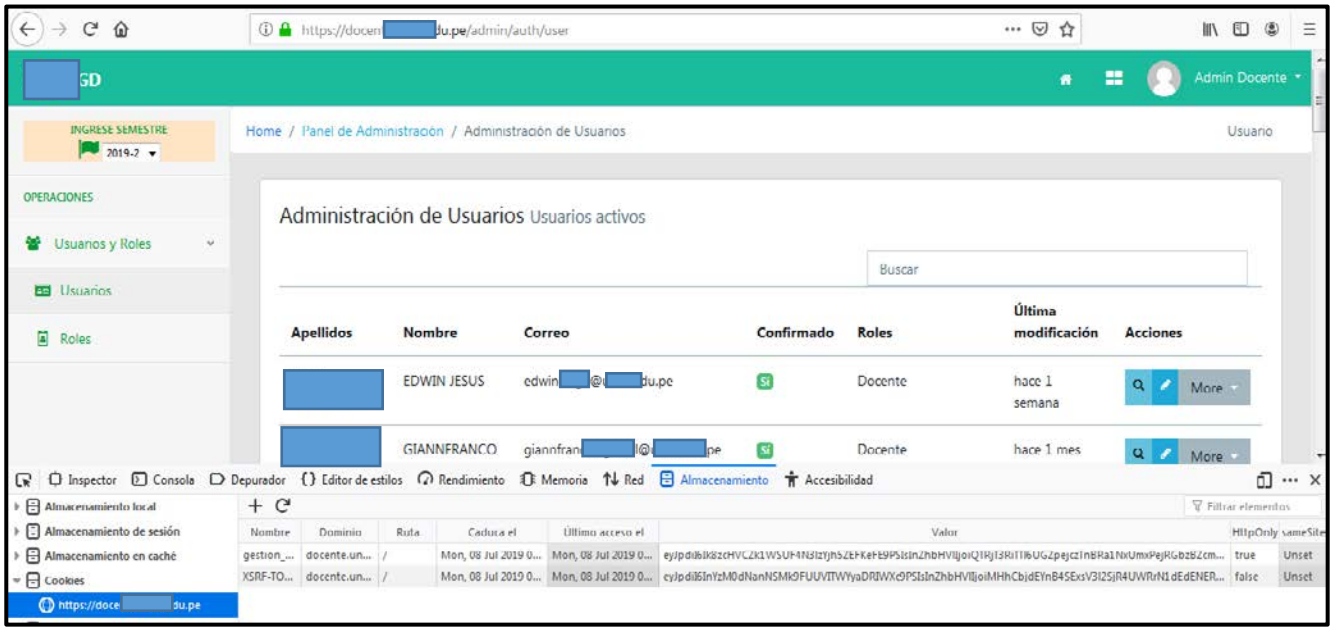


Fig. 11. Reproducción de la sesión reutilización la cookie de sesión en el Sistema de Gestión Docente

TABLA VII. SEVERIDAD DE LA VULNERABILIDAD EN LA COOKIE DE SESIÓN

Métrica	Valor
Vector de ataque	Red
Complejidad de ataque	Baja
Privilegios requeridos	Ninguno
Interacción con el usuario	Requerido
Alcance	Sin cambios
Confidencialidad	Bajo
Integridad	Ninguno
Disponibilidad	Ninguno
Puntuación Base CVSS v3.0	**4.3**

La puntuación en el sistema CVSS es de 4.3, tal como se muestra en la Tabla VII, lo cual indica un nivel de severidad medio, ya que para lograr una explotación exitosa es necesario recrear un escenario donde el atacante capture la cookie y luego logre secuestrar la sesión del usuario.

Se recomienda que además de usar la cookie como token de autenticación, realizar más comprobaciones en el lado del servidor web (validar usuario y contraseña).

E. Funcionalidad de cierre de sesión (OTG-SESS-006)

La terminación de la sesión es una parte importante del ciclo de vida de la sesión. Algunas aplicaciones web se basan únicamente en la cookie de sesión para identificar al usuario que ha iniciado sesión. Al cerrar sesión, la cookie de sesión se elimina del navegador, sin embargo, dado que la aplicación no realiza ningún seguimiento, no sabe si una sesión está desconectada o no. Por lo tanto, al reutilizar una cookie de sesión es posible obtener acceso a la sesión autenticada. Para esta prueba se inició sesión con usuario y contraseña válidos, luego se invocó la función de cierre de sesión y a continuación se navegó a una página que solo está visible en una sesión autenticada, mediante el uso del botón atrás del navegador, volviendo a acceder al panel de administración, lo ideal es que la aplicación redirija al formulario de autenticación, lo cual debería ser necesario para la seguridad de la aplicación, pero el establecimiento de cookies de sesión a nuevos valores después del cierre de sesión generalmente se considera una buena práctica.

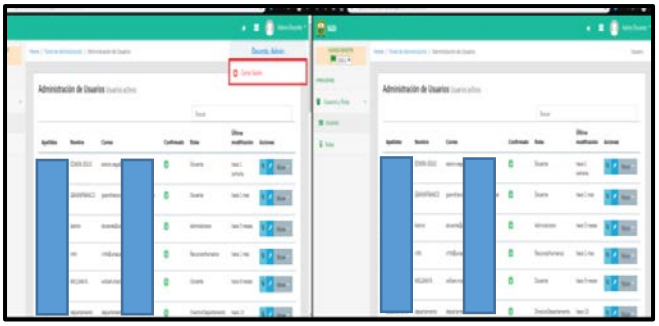


Fig. 12. Prueba de terminación de sesión del lado del servidor en el Sistema de Gestión Docente

Producto de la prueba anterior (reproducción de cookies de sesión) en la que se comprobó que es posible reproducir una sesión usando el mismo valor de la cookie, se verifico la funcionalidad de cierre de sesión cerrando la sesión original (victima) y se verificó que aún es posible mantener la otra sesión activa, lo cual no implica un alto riesgo, a menos que un atacante logre capturar la cookie y así logre secuestrar la sesión del usuario, tal como se evidencia en las Fig. 12, Fig. 13 y Fig. 14.

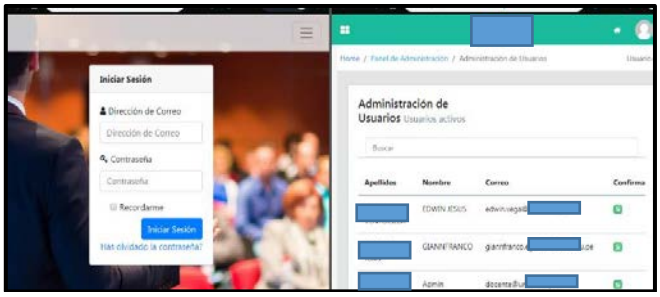


Fig. 13. Prueba de funcionalidad de cierre de sesión

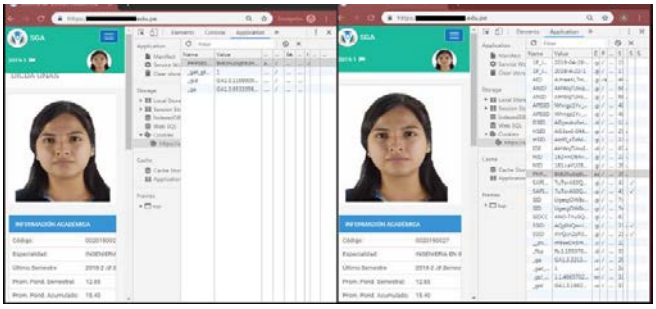


Fig. 14. Reproducción de la sesión utilizando la cookie en el Sistema de Gestión Académica

La puntuación en el sistema CVSS es de 4.3, tal como se muestra en la Tabla VIII, lo cual indica un nivel de severidad medio, ya que para lograr una explotación exitosa es necesario recrear un escenario donde el atacante capture la cookie y luego logre secuestrar la sesión del usuario.

TABLA VIII. SEVERIDAD DE LA VULNERABILIDAD DE LA FUNCIONALIDAD DE CIERRE DE SESIÓN

Métrica	Valor
Vector de ataque	Red
Complejidad de ataque	Baja
Privilegios requeridos	Ninguno
Interacción con el usuario	Requerido
Alcance	Sin cambios
Confidencialidad	Bajo
Integridad	Ninguno
Disponibilidad	Ninguno
Puntuación Base CVSS v3.0	**4.3**

Se recomienda, que como buena práctica se establezca las cookies de sesión a nuevos valores después del cierre de sesión, además de destruir dicha variable al invocar la función de cierre de sesión. Asimismo, implementar el indicador HttpOnly, que permite enviar cookies solo a través del protocolo http, pero no de JavaScript. Establecer la marca HttpOnly considerando como base el siguiente ejemplo:

```
<?php
ini_set("session.cookie_httponly", "True");
session_start();
session_regenerate_id();
if(!isset($_SESSION['admin_loggedin']))
{
    header('Location: index.php');
}
if(isset($_GET['search']))
{
    if(!empty($_GET['search']))
```

```
    {
        $text = $_GET['search'];
    }
    else
    {
        $text = "No text Entered";
    }
}
?>
```

Implementar el indicador “secure”, en el encabezado set-cookie para enviar cookies solo a través de HTTPS. Establecer la marca “secure” considerando como base el siguiente ejemplo:

```
<?php
ini_set("session.cookie_secure", "True");
session_start();
session_regenerate_id();
if(!isset($_SESSION['admin_loggedin']))
{
    header('Location: index.php');
}
if(isset($_GET['search']))
{
    if(!empty($_GET['search']))
    {
        $text = $_GET['search'];
    }
    else
    {
        $text = "No text Entered";
    }
}
?>
```

V. RESULTADOS

El análisis de vulnerabilidades presentado como un caso de estudio enfatiza la necesidad de realizar pruebas mas allá de la aplicación de herramientas automatizadas; debido a que como se evidencia, no existe herramienta que detecte y valide la existencia de vulnerabilidades que no sean tecnológicas completamente.

Se ha mostrado que puede existir un impacto significativo con el tipo de vulnerabilidades orientado a la lógica de negocio ya que corresponden a, cómo las aplicaciones funcionan, qué datos transmiten, cómo las transmiten, que privilegios tiene cada rol de una aplicación, manejo de sesiones, manejo de credenciales, etc. Este tipo de fallas están relacionadas a la manera en que se han construido las aplicaciones y sitios web, por lo que la solución por lo general suele ser complicada en términos de tiempo y adaptación.

Las vulnerabilidades identificadas en este caso de estudio tienen un nivel de severidad entre 4.3 (medio) a 7.5 (alto) según CVSS lo cual implica tomar las acciones necesarias para mitigarlas. Comúnmente este tipo de vulnerabilidades están asociadas al manejo de credenciales y sesiones de las aplicaciones.

VI. CONCLUSIONES

En un proceso de análisis de vulnerabilidades se deben consideran tanto las relacionadas a vulnerabilidades tecnológicas como las vulnerabilidades de lógica de negocio, por lo que es importante la capacidad de análisis del pentester o auditor además del manejo de las herramientas adecuadas.

El conocimiento del funcionamiento de las aplicaciones es vital para un análisis exhaustivo y preciso que permita detectar vulnerabilidades importantes que pueden afectar significativamente a la organización. Estas vulnerabilidades pueden ser las más complicadas de identificar.

REFERENCIAS

[1] B. Babak D. y Z. A. S. Hessa Mohammed , «A study on penetration testing process and tools,» de *Long Island Systems, Applications and Technology Conference (LISAT)*, Farmingdale, NY, 2018.

[2] Y. Stefinko, A. Piskozub y R. Banakh, «Manual and automated penetration testing. Benefits and drawbacks. Modern tendency,» de *13th International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, Lviv, Ukraine, 2016.

[3] R. E. L. d. Jiménez, «Pentesting on web applications using ethical - hacking,» de *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, San José, 2016.

[4] S. Smith, «Recognizing and Preparing Loss Estimates from Cyber-Attacks,» *Information System Security,* vol. 12, nº 6, pp. 46-57, 2004.

[5] P. Aguilera Lopez, Seguridad Informática. Ciclos formativos, Editex, 2010, p. 240.

[6] Open Web Application Security Project, «OWASP Testing Guide v4,» Abril 2016. [En línea]. Available: https://www.owasp.org/images/1/19/OTGv4.pdf.

[7] Open Web Application Security Project, «OWASP Top 10 2017,» 20 noviembre 2017. [En línea]. Available: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf.

[8] Open Web Application Security Project, «Penetration testing methodologies,» 19 Febrero 2019. [En línea].

[9] N. Sangeeta y S. Kurkure, «Vulnerability Assessment and Penetration Testing of Web Application,» de *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, 2017.

[10] Open Web Application Security Project, «OWASP ZAP 2.8,» 20 Marzo 2019. [En línea]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.

[11] SANS Institute Reading Room site, «Burp Suite(up) with fancy scanning mechanisms,» 20 Febrero 2019. [En línea]. Available: https://www.sans.org/reading-room/whitepapers/tools/burp-suiteup-fancy-scanning-mechanisms-36557.

[12] P. S. Shinde y S. B. Ardhapurkar, «Cyber security analysis using vulnerability assessment and penetration testing,» de *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, Coimbatore, 2016.

[13] P. S. Anton, R. H. Anderson, R. Mesie y M. Scheiern, The Vulnerability Assessment & Mitigation, N. D. R. Institute, Ed., p. 134.

[14] FIRST.org, Inc, «Common Vulnerability Scoring System v3.1: Specification Document,» 2019. [En línea]. Available: https://www.first.org/cvss/v3.1/specification-document. [Último acceso: enero 2019].

William-Rogelio Marchand-Niño, Ingeniero de Sistemas otorgado por la Universidad Nacional del Centro del Perú, con maestría en Dirección Estratégica de TI de la Universidad de Piura, con 18 años de experiencia académica en UNAS, UDH, UPLA. Desde el año 2004 es profesor asociado en la UNAS. Ha impartido más de 90 cursos de pregrado en diferentes universidades. Instructor CISCO por 12 años. Posee múltiples certificaciones de la Industria como PMP, ITIL Foundation, CCNA, MTA. Director del Centro de Tecnologías de Información y Comunicación de la Universidad Nacional Agraria de la Selva. Miembro Senior de la IEEE.

Ana Claudia Terrones Murga, Estudiante de Ingeniería en Informática y Sistemas de la Universidad Nacional Agraria de la Selva (UNAS). Certificada Cisco CCNA R&S. Miembro del Grupo de Investigación de Redes y Seguridad de la Facultad de Ingeniería en Informática y Sistemas (FIIS-UNAS)