Reporte de Riesgos e Impacto al negocio:
Realizando Pruebas de Penetración en un Entorno Controlado de un Sistema en Producción.

Nombre del investigador forense: Orlando Valdés

Nombre de la organización/empresa: Estudiante de Seminario por la OEA

Fecha del informe: 09/08/2023

Resumen

El documento describe una serie de ejercicios de laboratorio para probar la seguridad de aplicaciones web que utilizan PHP y MySQL. El documento implica que la aplicación web tiene un diseño inseguro y no sigue las mejores prácticas recomendadas por OWASP.

Este informe forense tiene como objetivo analizar los resultados de una investigación sobre la seguridad de una aplicación web que utiliza PHP y MySQL. Se identificaron varias vulnerabilidades y debilidades en el sistema, como un diseño inseguro, falta de autenticación y autorización, inyección SQL, ejecución de código remoto y fallas de configuración y despliegue. Se utilizó una metodología forense para reconstruir y analizar las pruebas del incidente de seguridad.

Reporte de Hallazgo

Se debe tomar en consideración que asumiremos que el documento proporcionado pudiesen ser una guía desarrollada que está siendo estudiada y se replicara el laboratorio con el fin de validar la información que contiene, tomando en consideración que el sistema será un entorno controlado para dicha prueba.

Se realiza un análisis de vulnerabilidades y amenazas desde una perspectiva forense, se han identificado a continuación:

1. Diseño inseguro: El documento describe una serie de ejercicios de laboratorio que ponen a prueba la seguridad de una aplicación web que utiliza PHP y MySQL. Estos

ejercicios implican la explotación de vulnerabilidades comunes, como la inyección SQL, errores de configuración y la ejecución de código remoto. Según el documento, la aplicación web tiene un diseño inseguro que no sigue las mejores prácticas recomendadas por OWASP.

2. Falta de autenticación y autorización: El documento muestra que la aplicación web permite el acceso no autorizado a datos y funcionalidades sensibles, como información de empleados, credenciales de la base de datos y carga de archivos. Además, se revela que la aplicación web no implementa una adecuada autenticación y gestión de sesiones de los usuarios.

3. Inyección SQL: El documento explica cómo llevar a cabo un ataque de inyección SQL en el parámetro "id" del sitio web, utilizando la herramienta sqlmap para extraer información de la base de datos "users". Se revela que la base de datos contiene los nombres de usuario y contraseñas en texto plano de los administradores del sitio, lo que facilita el acceso no autorizado al área restringida.

4. Ejecución de código remoto: El documento detalla cómo realizar un ataque de ejecución de código remoto en el sitio web, utilizando la herramienta Burp Suite para modificar la extensión y el contenido de un archivo subido al servidor. Se indica que es posible subir un archivo PHP con una shell reversa que permite ejecutar comandos en el sistema operativo del servidor, aprovechando una mala validación del tipo y tamaño de los archivos.

5. Fallas de configuración y despliegue: El documento señala que el servidor web Apache2 tiene archivos y directorios desprotegidos, como el directorio "admin", que contiene una bandera oculta. Además, sugiere que el servidor web almacena algunas variables de entorno con las credenciales de acceso a la base de datos, lo que podría ser aprovechado por un atacante para obtener información sensible.

Estos hallazgos indican claramente la presencia de múltiples vulnerabilidades y deficiencias en la seguridad de la aplicación web analizada.

Reporte Técnico

Tomando en consideración los hallazgos anteriores y validando con el documento proporcionado, podemos resumir 3 áreas de amenazas tomando como referencias manuales de buenas practicas tal como los es OWASP Top Ten 2021.

De esta manera podemos desglosar y seccionar de manera técnica los hallazgos e intentar categorizarlos para brindar una mejor comprensión de las amenazas e identificar a manera de listado las vulnerabilidades que sirvan como referencia futura a un debido saneamiento; dando como resultado un sistema (aplicación web) eficiente con seguridad y confiabilidad para el negocio del capital que este representa como un activo más de la organización.

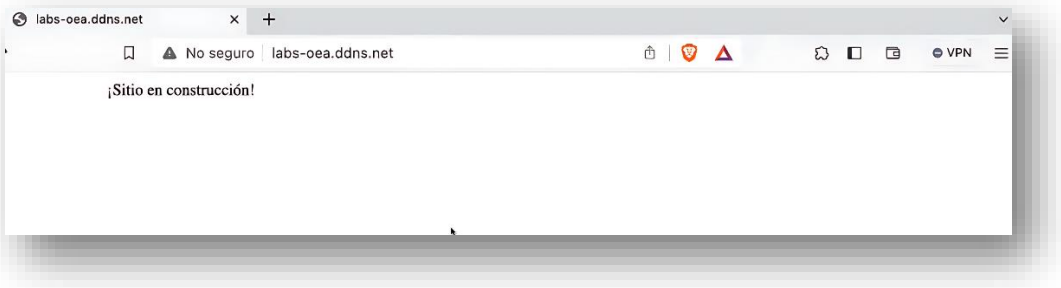

Ilustración 1. Aplicación Web en Producción.

1. Diseño inseguro:

En esta sección se identificó un diseño inseguro de la aplicación web, que expone funcionalidades sensibles y no implementa una adecuada autenticación y gestión de sesiones de los usuarios. Según el OWASP Top Ten 2021, esto podría estar relacionado con los siguientes puntos:

- A04:2021-Security Misconfiguration: Esta categoría de OWASP se refiere a la falta de configuración segura en la aplicación web, lo cual puede incluir permisos incorrectos en archivos y directorios, configuraciones predeterminadas inseguras, entre otros aspectos.

- A07:2021-Broken Authentication: Esta categoría se refiere a las vulnerabilidades relacionadas con la autenticación y gestión de sesiones. Esto puede incluir la falta de protección adecuada de las credenciales de los usuarios, la ausencia de mecanismos de autenticación fuertes, entre otros.

Para validar esta información se ejecutan las siguientes pruebas empleando un listado de herramientas y técnicas:

Nmap: Una herramienta de escaneo de puertos y rastreo de servicios ampliamente utilizada.

Dirb: Una herramienta para rastrear directorios y archivos en servidores web.

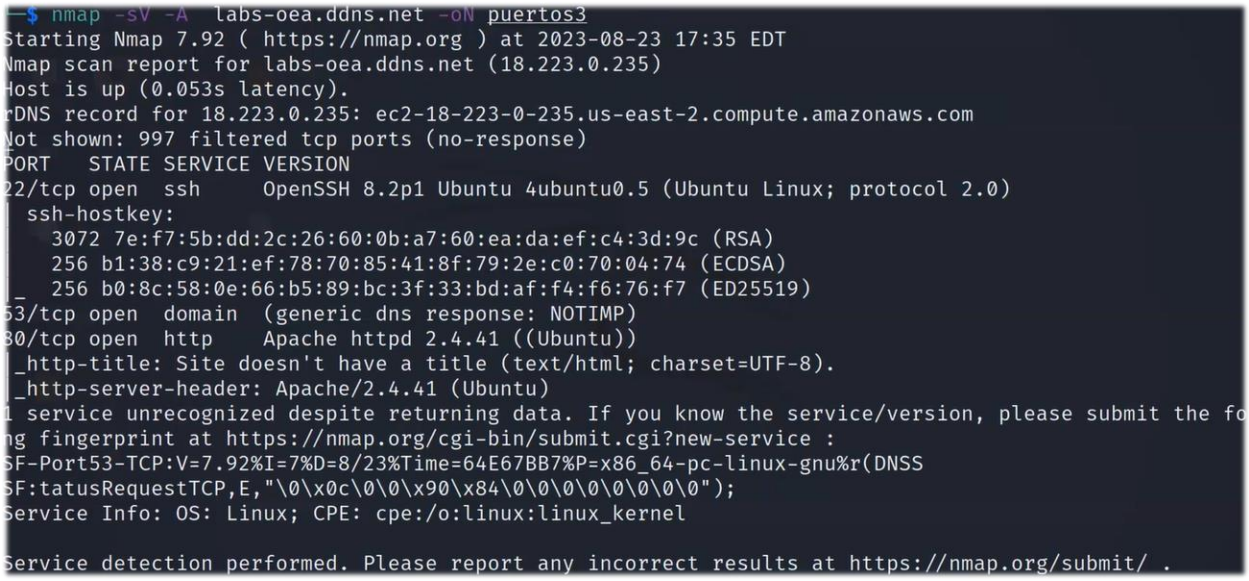

Ilustración 2. Herramienta NMAP

Según la OWASP en el punto A5-Configuración de Seguridad Incorrecta:

Los atacantes a menudo intentarán explotar vulnerabilidades sin parchear o acceder a cuentas por defecto, páginas no utilizadas, archivos y directorios desprotegidos, etc., para obtener acceso o conocimiento del sistema o del negocio.

Los servidores como apache2 de Linux/Unix son propensos a este tipo de vulnerabilidades ya que no existe alguna configuración de seguridad por defecto, para proteger los servicios.

Factores de incidencia:

CWEs mapeadas :20

Tasa de incidencia máx.:19.84 %

Tasa de incidencia prom.:4.51 %

Incidencias totales.:208,387

Explotabilidad ponderada prom.:8.12

Impacto ponderado prom.:6.56

Cobertura máx.:89.58 %

Cobertura prom.: 44.84 %

Total CVEs.:789

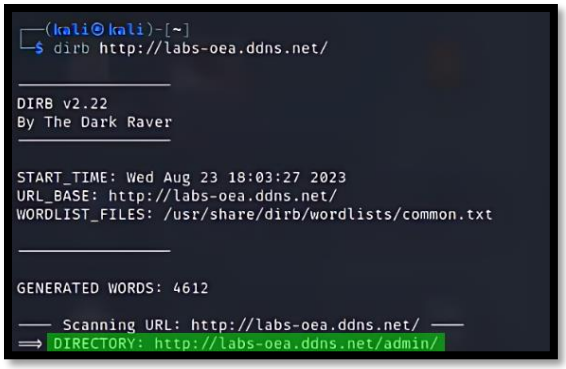

Ilustración 3. Herramienta Dirb

2. Inyección SQL:

En esta sección se identificó una vulnerabilidad de inyección SQL en el parámetro "id" de la aplicación web. Esta vulnerabilidad permite a un atacante ejecutar comandos SQL no deseados en la base de datos. Según el OWASP Top Ten 2021, esto está relacionado con el punto:

- A03:2021-Injection: Esta categoría se refiere a las vulnerabilidades de inyección, donde los datos no confiables se insertan en comandos o consultas no seguras. Esto puede incluir inyección SQL, inyección de comandos, entre otros.

Esta vulnerabilidad también es, según la OWASP: A03:2021 – Inyección, en combinación con A07:2021 – Fallas de Identificación y Autenticación: La confirmación de la identidad, la autenticación y la gestión de sesiones del usuario son fundamentales para protegerse contra ataques relacionados con la autenticación. Puede haber debilidades de autenticación si la aplicación permite ataques de fuerza bruta u otros ataques automatizados.

Factores de incidencia:

CWEs mapeadas:22

Tasa de incidencia máx.:14.84 %

Tasa de incidencia prom.:2.55 %

Explotabilidad ponderada prom.:7.40

Impacto ponderado prom. 6.50

Cobertura máx.:79.51 %

Cobertura prom.: 45.72 %

Incidencias totales:132,19

Total CVEs: 3,897

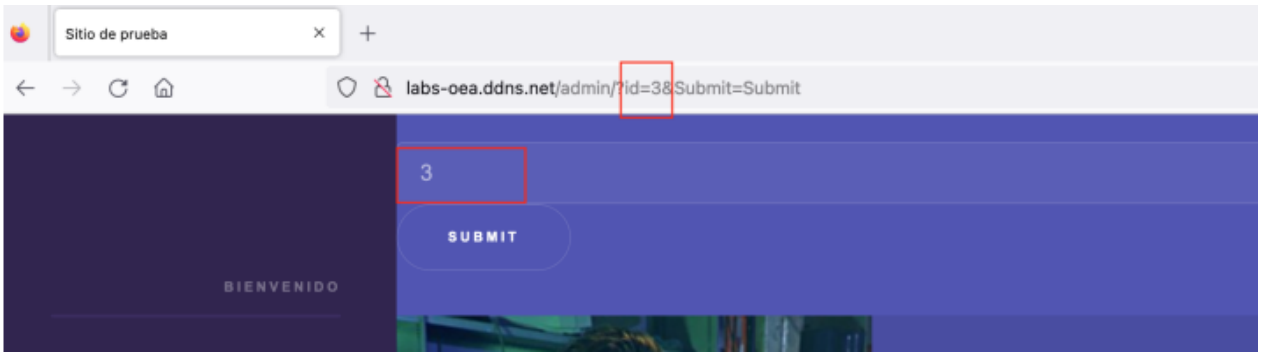

Ilustración 4. Parametros de consultas propensos a inyeccion SQLi

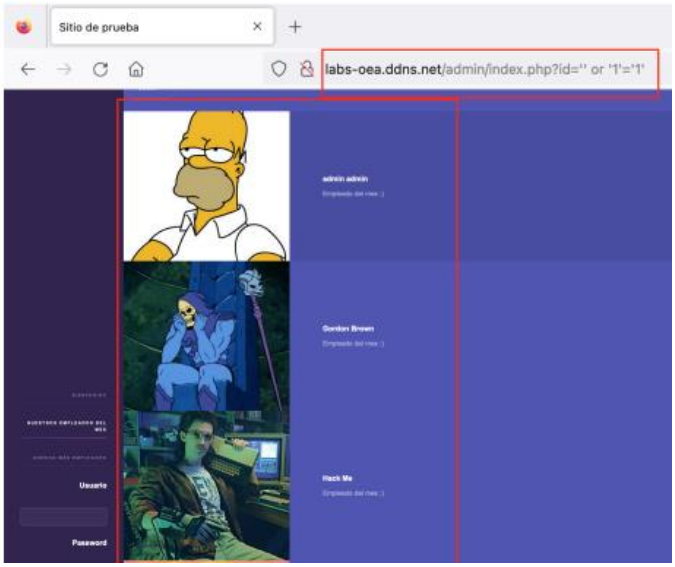

Ilustración 5. Consulta de tipo malintencionada.

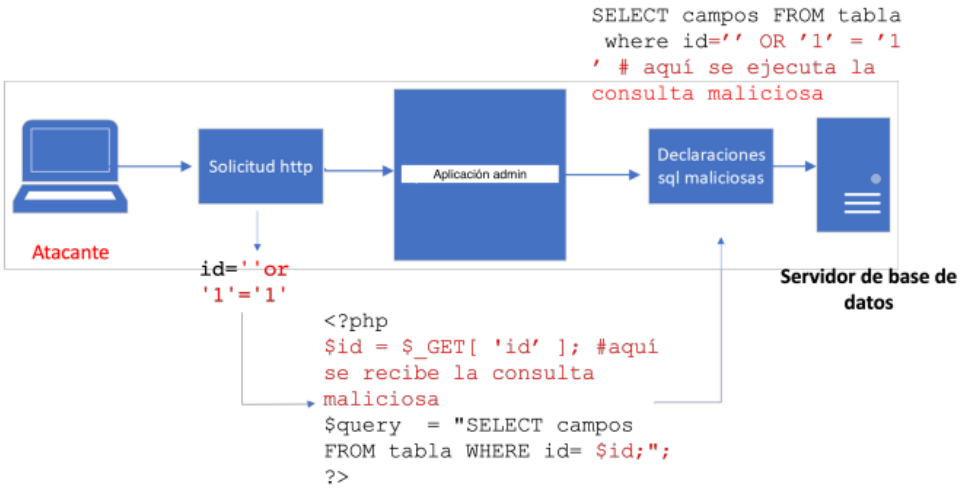

Ilustración 6. Diagrama de un ataque de inyección de SQL

Con el apoyo de herramientas es más frecuente encontrar este tipo de ataques en sitios no solo de producción, que pueden llegar a ser perjudiciales para las organizaciones, a continuación, se muestra como un escenario, el aprovechamiento de la herramienta SQL Map.

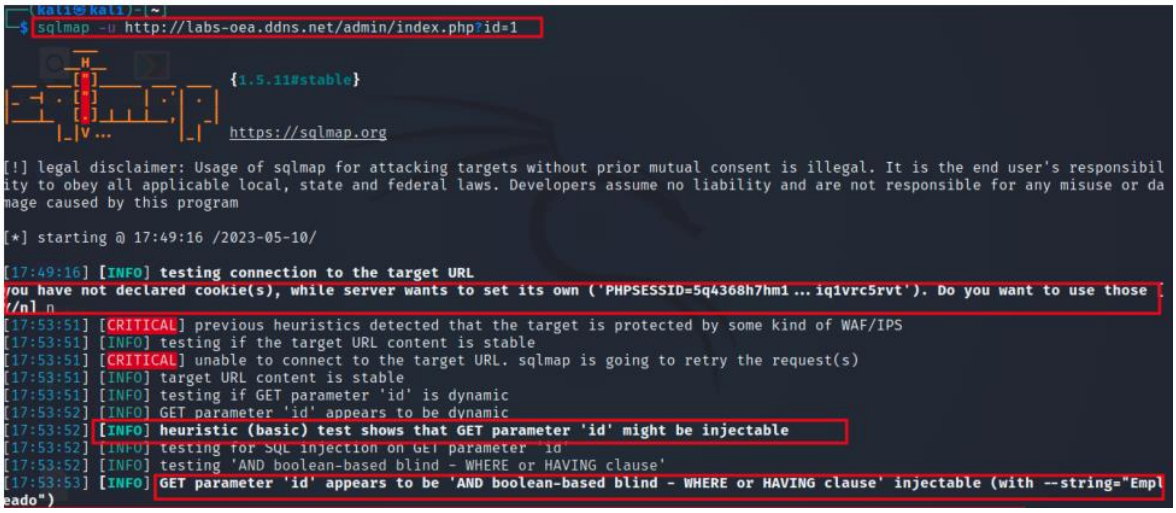

Ilustración 7. Lanzamiento de la herramienta.

Ilustración 8. Tras obtener una consulta exitosa, se muestran los tipos de consultas que se pueden realizar, funciona en cuestión como fuerza brutal a los tipos que existen.

Tras probar diferentes entradas con esta herramienta se llegan a ciertos resultados como el siguiente:

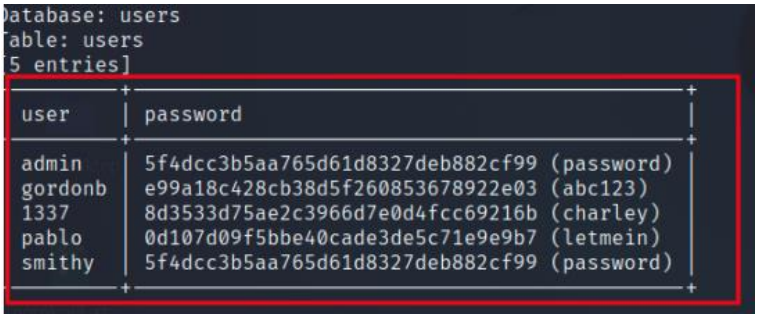

Ilustración 9. Informacion detallada de accesos al aplicativo web

3. Ejecución de Código Remoto:

En esta sección se encontró una vulnerabilidad de ejecución de código remoto, que permite a un atacante ejecutar código arbitrario en el servidor. Según el OWASP Top Ten 2021, esto podría estar relacionado con el punto:

- A03:2021-Code Injection: Esta categoría se refiere a las vulnerabilidades de inyección de código, donde los datos no confiables se insertan en funciones o lenguajes de programación no seguros. Esto puede incluir inyección de código en lenguajes como PHP, JavaScript, entre otros.

Esta vulnerabilidad también es según la OWASP: A03:2021 – Inyección, en combinación con A04:2021 – Diseño Inseguro: El diseño inseguro es una categoría amplia que representa diferentes debilidades, expresadas como "diseño de control faltante o ineficaz". Uno de los factores que contribuyen al diseño inseguro es la falta de perfiles de riesgo empresarial inherentes al software o sistema que se está desarrollando y, por lo tanto, la falta de determinación del nivel de diseño de seguridad que se requiere.

Factores de incidencia:

CWEs mapeadas :40	Explotabilidad ponderada prom. :6.46	Cobertura prom.: 42.51 %
Tasa de incidencia máx.:24.19 %	Impacto ponderado prom. 6.78	Incidencias totales:262,407
Tasa de incidencia prom.:3.00 %	Cobertura máx.:72.25 %	

Total CVEs : 2,691

En el caso de las pruebas realizas se empleó la herramienta Burp suite, para interferir con la información y la comunicación entre el usuario y el aplicativo, por lo que, incluso se es capaz de introducir una Shell reverse.

Al obtener credenciales anteriormente, se accede como un administrador donde se encuentra una sección para la carga de archivos. En donde como anteriormente se emplea una herramienta para el apoyo de lograr crear una Shell reverse, para poder

manipular el servidor del aplicativo y demás alcances-repercusiones que por la astucia e imaginación del atacante se pudiesen perpetrar.

Ilustración 10. Aplicativo web tras entrar como administrador, y visualizar la seccion de carga de archivos. Nota: el archivo solo puede ser de tipo imagen en formatos (JPG, PNG).

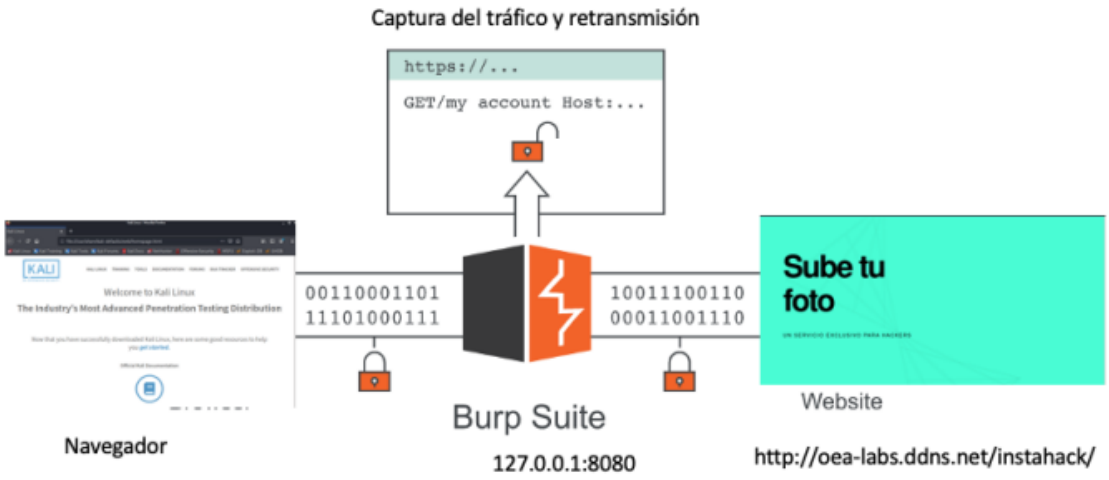

Ilustración 11. Diagrama de funcionamiento de la herramienta Burp Suite.

Ilustración 12. Manipulación y comprometimiento de la integridad del aplicativo con la herramienta.

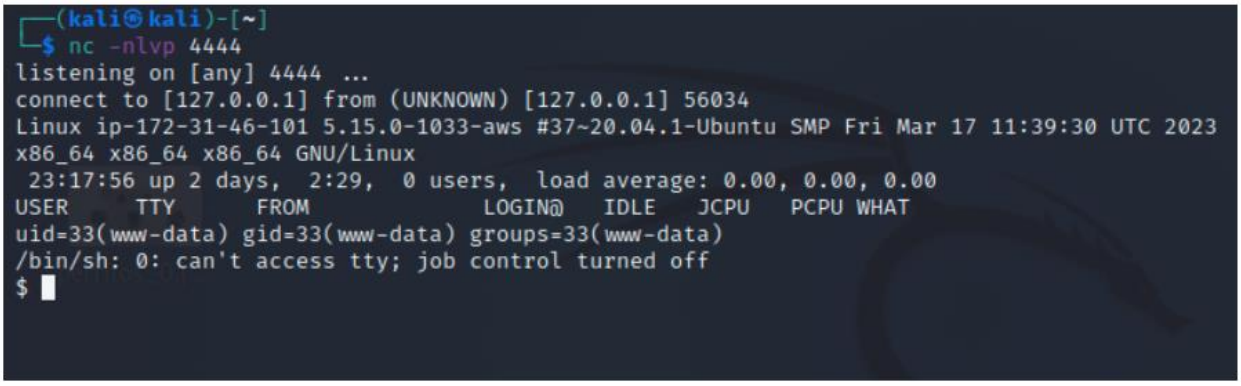

Ilustración 13. Netcar recibiendo la shell reverse.

Debemos mencionar que, para lograr el cometido mencionado anteriormente, se empleo el trafico de datos, para interceptar y modificar el archivo que contenía la Shell, ¿De qué manera?, El aplicativo lee el archivo de tipo JPG, pero antes de mandarlo a guardar al servidor de alojamiento, se cambia su formato a un archivo php, que contendrá instrucciones maliciosas y permitirán la creación de la Shell per se.

Para poder lograr un escenario ideal, se necesita un agente en Internet que captura la información del servidor (víctima) y la devuelva al cliente (atacante). Inicie la herramienta ngrok en el puerto local 4444 del protocolo TCP e identifique el puerto remoto generado en la dirección de la herramienta.

```
#Kali Linux
 $ ./ngrok tcp 4444
```

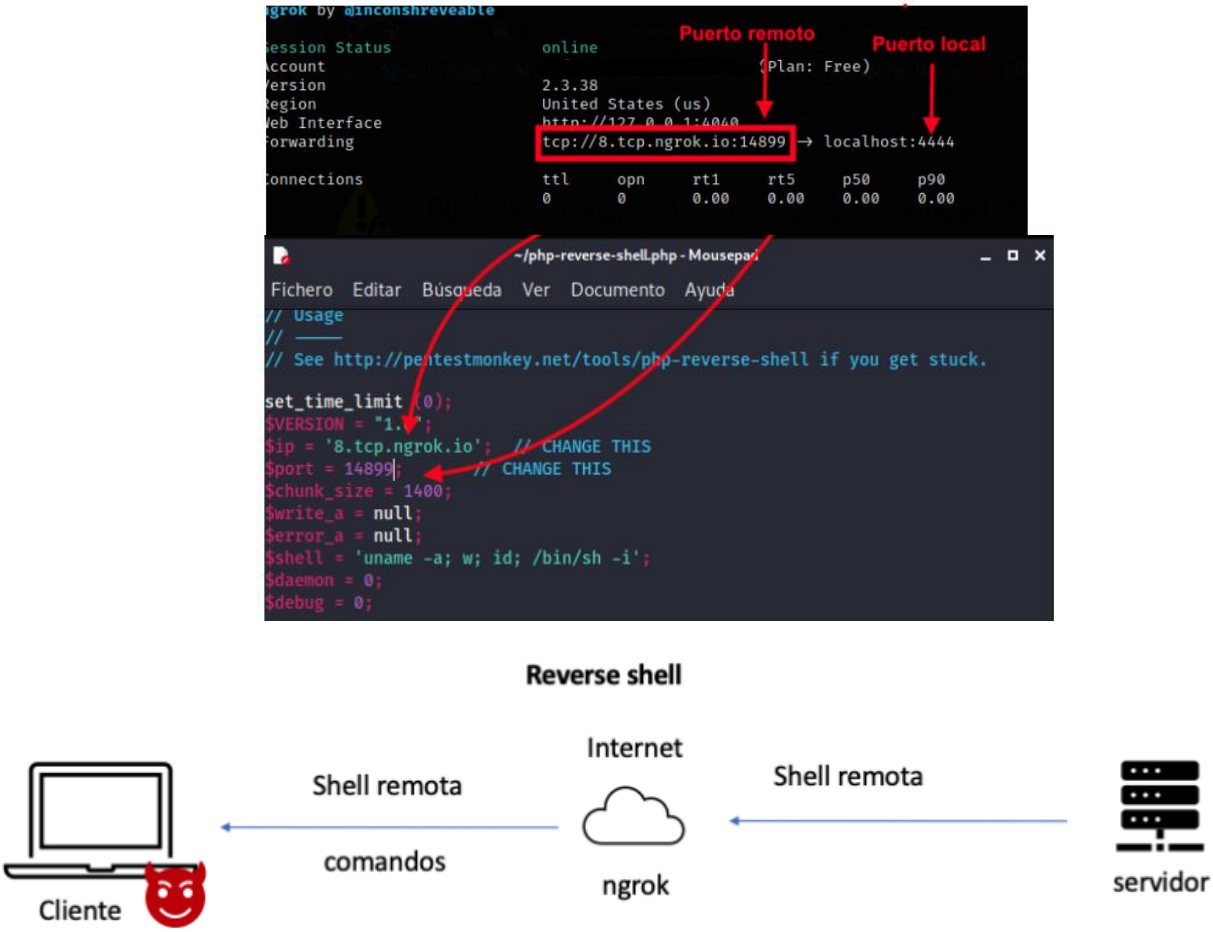

Ilustración 14.Escenario utilizando la herramienta ngrok.

Conclusiones

Con el presente informe se puede concluir que nos encontramos con una guía que detalla minuciosamente la aplicación web, que representa como un Sitio o Sistema en Producción; por lo que podemos encontrarnos ciertamente con amenazas y vulnerabilidades para el negocio.

De esta manera podemos comprender que la guía que ha sido sujeto de estudio como parte del análisis forense brinda detalles específicos que tientan contra la integridad del aplicativo por lo que se detalla o concluye de la siguiente manera:

1. El aplicativo web analizado presenta múltiples vulnerabilidades y deficiencias en su seguridad, incluyendo un diseño inseguro, falta de autenticación y autorización, inyección SQL y ejecución de código remoto.
2. Estas vulnerabilidades representan un riesgo significativo para la confidencialidad, integridad y disponibilidad de los datos y funcionalidades sensibles del aplicativo.
3. La falta de implementación de las mejores prácticas de seguridad recomendadas por OWASP es un factor clave en la existencia de estas vulnerabilidades.

Por lo que proponiendo ir más allá y brindando una opinión argumentativa tras los hallazgos encontrados sería tomar medidas para realizar un saneamiento que puede inferirse en algunas de las siguientes premisas:

1. Realizar una revisión exhaustiva del diseño del aplicativo web y aplicar las mejores prácticas de seguridad recomendadas por OWASP, como la configuración segura de servidores, la implementación adecuada de autenticación y autorización, y la validación de entradas de usuario para prevenir ataques de inyección.
2. Realizar pruebas de penetración regulares en el aplicativo web para identificar y corregir posibles vulnerabilidades antes de que sean explotadas por atacantes.
3. Implementar un proceso de gestión de parches y actualizaciones para mantener el aplicativo web y sus componentes actualizados con las últimas correcciones de seguridad.
4. Capacitar al personal encargado del desarrollo y mantenimiento del aplicativo web en buenas prácticas de seguridad, como la codificación segura y la gestión adecuada de contraseñas.
5. Establecer un proceso de monitoreo y registro de eventos de seguridad para detectar y responder rápidamente a posibles incidentes de seguridad.
6. Considerar la contratación de servicios de consultoría en seguridad informática para realizar auditorías periódicas y obtener recomendaciones específicas para mejorar la seguridad del aplicativo web.