EasyChair Preprint

№ 4609

Security Through an Audit Plan

Héctor Joab Ibarra Caballero, Ana Claudia Zenteno Vázquez,
María Del Carmen Santiago Díaz and
Gustavo Trinidad Rubin Linares

November 19, 2020

Seguridad Mediante un Plan de Auditorías

Héctor Joab Ibarra Caballero, Ana Claudia Zenteno Vázquez, María del Carmen Santiago Díaz, Gustavo Trinidad Rubín Linares
Facultad de Ciencias de la Computación, BUAP
14 sur y Av. San Claudio, Jardines de San Manuel,
Puebla, México
{joab.hector, anaclau.zenteno, mcsantiagodiaz, gtrubin}@gmail.com

Resumen. La seguridad se ha convertido en un proceso dentro del desarrollo de las aplicaciones web que ha tomado importancia en los últimos años, Laravel provee características de seguridad a los desarrollos de aplicaciones web. En este trabajo se usan herramientas de análisis de vulnerabilidades para evaluar la seguridad y comprobar la eficiencia del desarrollo de software obteniendo como resultado un plan de auditoría sugerido para mitigar las vulnerabilidades: caso de estudio sobre una aplicación web.

Palabras Clave: Laravel, Aplicaciones Web, Auditorías, ZAP, Burp Suite

1 Introducción

Los planes de auditoría en seguridad son una serie de pruebas de penetración y análisis de vulnerabilidades que tiene como objetivo evaluar la seguridad y las posibles vulnerabilidades de diseño, falta de configuraciones adecuadas o establecidas por defecto, la carencia de validaciones en un sistema, etc. Una evaluación de vulnerabilidades proporciona una amplia visión sobre las fallas existentes en los sistemas.

Los actuales modelos de seguridad implementados por las compañías durante los procesos de desarrollo y liberación de una aplicación web lleva a solucionar las fallas de seguridad desarrollando parches a sus sistemas en base a los reportes de fallos. En la Fig. 1 se observa que este tipo de prácticas son consideradas por los especialistas de la seguridad como una práctica ineficaz ya que existe una ventana de vulnerabilidad y explotación entre los tiempos que se descubre la falla, se crea y envía el reporte, para después notificar al usuario de la falla, generar un análisis de la falla, crear el parche, la posterior liberación del parche, para finalizar con la instalación. Es por estas prácticas que en la actualidad se ve un incremento en los reportes de fallas en seguridad de los sistemas de las empresas y otros organismos. En un reporte del "*El financiero"* menciona que *México es la segunda nación en Latinoamérica que se ha visto afectada por el cibercrimen y a nivel mundial se encuentra en los 10 países con mayores daños por este fenómeno, lo que representa pérdidas por 3 mil millones de dólares anuales*[1].

Fig. 1. Ventana de Vulnerabilidad; OWASP Testing Guide V4 [2]

Por otro lado, la seguridad web está definida como el conjunto de procedimientos, prácticas y herramientas tecnológicas para la protección web, servidores, usuarios y organizaciones. [3] Las pruebas de penetración son una serie de ejercicios en seguridad donde se busca identificar las vulnerabilidades de un sistema, fallas en seguridad que podrían ser aprovechadas durante un ataque cibernético, ataque cibernético es un acto intencional por parte de un usuario malicioso con el fin de aprovecharse de una vulnerabilidad en el diseño de una aplicación con el fin de explotar estas fallas, la explotación consiste en la elaboración o implementación de herramientas que aprovechan la falla encontrada para obtener un beneficio, que van desde el robo de información hasta la perdida en el control del sistema . En la Fig. 2 podemos observar de forma gráfica el proceso de una prueba de penetración, que parte desde el análisis del sistema web a sus diferente métodos o funciones en búsqueda de alguna vulnerabilidad, para la posterior explotación y aprovechamiento del sistema mediante herramientas, al establecer que un sistema cuenta con la seguridad web adecuada es probable que un ataque cibernético no pueda causar algún impacto en el sistema evitando la explotación del mismo.

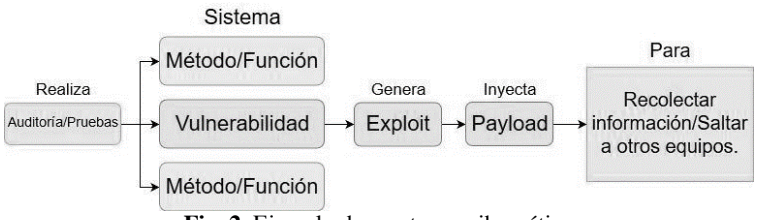

Fig. 2. Ejemplo de un ataque cibernético

1.1 Herramientas para la detección de vulnerabilidades

Para equipos de desarrollo que no tienen la suficiente experiencia ni conocimientos en realizar pruebas de penetración de forma manual se recomienda el uso de herramientas automáticas especialistas en estos temas que cumplan con alguno plan de auditorías profesional, por ejemplo, OWASP Testing Guide V4, así como también brindan al evaluador la posibilidad de presentar los resultados de forma que personas sin conocimientos en seguridad puedan entender. OWASP ZAP y Burp Suite trabajan bajo

las normativas de seguridad de OWASP por lo que se recomienda su uso dentro de los equipos de desarrollo.

OWASP Zed Attack Proxy Project (ZAP) es una aplicación liberada de forma gratuita que trabaja con un proxy de intercepción, lo que nos permite realizar pruebas activas y pasivas a la aplicación web, cuenta con un escáner automático de vulnerabilidades que permite tener un resultado aproximado de las fallas encontradas dentro de una aplicación web.

Burp Suite es otra herramienta para el análisis de vulnerabilidades, cuenta con funciones para conocer las vulnerabilidades encontradas dentro del sistema. Es un programa complejo con una consola que nos permite ver, editar y enviar los mensajes interceptados con el proxy, así como enviar partes de los mensajes interceptados a otras herramientas dentro de Burp Suite para obtener una evaluación más completa de la aplicación web.

El framework Laravel permite el desarrollo de aplicaciones web seguras, es objeto de este artículo, conocer el grado de cumplimiento de estos estándares de seguridad. Comentaremos sobre la aplicación web desarrollada en Laravel con fines de pruebas para realizar el escaneo de vulnerabilidades utilizando las herramientas ZAP y Burp Suite.
En la sección dos se describe la estructura de desarrollo de aplicaciones seguras propuesta por Laravel. En la sección 3 se describe el plan de auditorías para analizar la aplicación desarrollada, y en la sección 4 se muestran conclusiones y trabajo futuro.

2 Desarrollo de Aplicación en Laravel

Es de conocimiento general que el uso de los distintos Framework y metodologías del desarrollo nos facilitan los procesos de desarrollo. Para determinar qué tan seguro es usar un Framework en un proyecto y por qué es necesario evaluar la seguridad de las aplicaciones web se propone analizar la seguridad y las herramientas de desarrollo que provee el Framework Laravel. En la Fig. 3 se muestra cómo se realizó una instalación pura del Framework utilizando la distribución Debian 8, además de los componentes necesarios para utilizar Laravel.

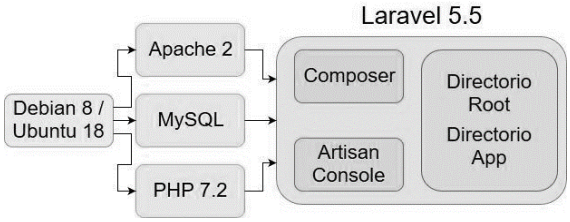

Fig. 3. Entorno de trabajo para Laravel Framework en Debian 8/Ubuntu 18

2.1 Estructura de la aplicación

La mayoría de los servicios web en línea trabajan con un sistema de entrada y registro dentro de sus estructuras principales, por lo que durante el desarrollo de nuestro proyecto se crearon vistas básicas y se usan las diferentes herramientas proporcionadas por Laravel para realizar una aplicación segura, así como añadir funciones para validar la información recibida. En la Fig. 4, se muestra estructura de la aplicación siguiendo las recomendaciones de desarrollo propuestas por Laravel, el esquema Ruta, Modelo, Vista, Controlador (RMVC).

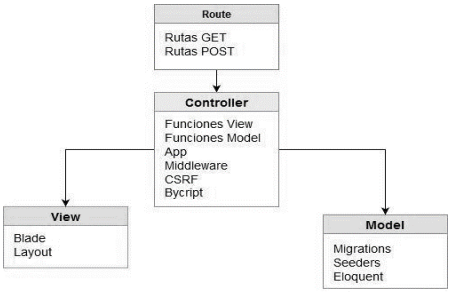

Fig. 4. RMVC de la aplicación web usando Laravel

Dentro de las rutas declaradas en el sistema podemos encontrar dos tipos de peticiones, las peticiones tipo GET son las que retornan las vistas de los formularios al usuario, mientras que las peticiones de tipo POST que realiza llamadas al controlador para la modificación y actualización del modelo. Para el desarrollo con formularios se incluyen *token csrf*, que permiten la validación de entrada y salida, donde es posible comprobar que no se incluya código javascript o sql. También se recomienda el uso de token de sesión el cual se mantiene hasta que el usuario cierra la sesión actual.

3 Plan de auditoría

Se propone un plan de auditoría con una serie de pruebas de penetración para analizar vulnerabilidades en la aplicación desarrollada basado en los 11 puntos principales de la seguridad web de la OWASP Testing Guide V4:

1. Recopilación de Información
2. Gestión de Configuración e Implementación
3. Gestión de identidad
4. Autenticación
5. Autorización
6. Gestión de Sesión
7. Validación de entrada
8. Manejo de errores
9. Criptografía
10. Lógica de negocios
11. Lado del Cliente

3.1 Mapeo de aplicación

Consiste en hacer una navegación manual de la aplicación web que se desea evaluar, para realizar un mapeo del sitio de forma pasiva se utiliza con Burp Suite. Este tipo de exploración dentro de la aplicación nos ayuda a tener un panorama de cómo responde el sistema, muestra las peticiones que trabaja, los tipos de entradas que recibe y la tecnología que usa. Burp Suite muestra un árbol de navegación con las carpetas y archivos a los que se han accedido por esta navegación.
ZAP crea un mapa del sitio al hacer una exploración activa con el escáner, donde podemos obtener distintas rutas como archivos a los cuales se puede acceder y evaluar.

3.2 Seguridad de Métodos de entrada y salida de datos

Es conveniente evaluar los distintos escenarios en los métodos de entrada y salida de la aplicación, agregando campos no esperados a los formularios del sitio para ver su respuesta. Un Spider (herramienta automatizada creada para el reconocimiento y mapeo de sitios web) puede ayudar a recorrer cada uno de los formularios.

3.3 Seguridad en las sesiones

Pruebas masivas de ingreso en los campos de usuario y contraseña son vulnerados cuando no se define un límite de intentos fallidos con ataques de fuerza bruta. Los inicios de sesión con guardados en tokens que suelen usar cifrado hash o valores hexadecimales y pueden ser descifrados con la herramienta decoder de Burp Suite.

Auditar consultas a las bases de datos con plugin CO2 de Burp suite para intentar hacer inyecciones SQL muestra si el sitio ha sido asegurado o no.

Además, ZAP realiza pruebas de activas y pasivas de vulnerabilidades sobre las generalidades del sitio como errores en los encabezados, manejo de cookies, protección del navegador deshabilitada, entre otras.

4 Conclusiones y trabajos futuros

En este trabajo concluimos que es recomendable utilizar un Framework del lado del servidor (Back-End), ya que facilita la incorporación y uso de herramientas que nos permiten generar aplicaciones web seguras. Herramientas que incluyen el cifrado de contraseñas, validación de datos de entrada a la base de datos, etc. Después de evaluar las herramientas que utiliza Laravel Framework llegamos a la conclusión que éste es un Framework que cubre los requisitos básicos de seguridad, mediante sus herramientas y su metodología de desarrollo RMVC. Las funciones del Framework se encuentran bien documentadas y son fáciles de usar, por lo que se recomienda trabajar con Laravel Framework en proyectos profesionales de desarrollo web. ZAP es una herramienta útil para el análisis de vulnerabilidades, proporciona información de la falla y cómo puede solucionarse facilitando la corrección de errores. Burp Suite es una herramienta compleja, puede ayudarnos a encontrar vulnerabilidades más específicas y complejas de una aplicación Web así como proporcionarnos las herramientas para la explotación de la vulnerabilidad. Juntas estas dos herramientas nos permiten generar reportes de seguridad basados en análisis de vulnerabilidades. Finalmente se propone un plan de auditorías basado en realizar dos pasos, primero implementar el uso de ZAP para la evaluación mediante el escáner activo de vulnerabilidades y evaluar los resultados con las más comunes fallas en seguridad en una aplicación web listadas en OWASP Top 10 The Ten Most Critical Web Application Security Risks. El segundo paso es usar Burp Suite e implementar el uso del proxy para evaluar las peticiones a aplicación web, la función de la herramienta Spider nos ayuda a obtener un mapeo completo del sitio, permitiendo la búsqueda de errores de diseño. El uso de algoritmos de fuerza bruta que se incluyen en el programa permite evaluar la seguridad de los controladores y las herramientas de sesión. Como trabajo futuro es posible realizar un proyecto de investigación donde se apliquen planes de auditoría a los diferentes Framework basados en diferentes lenguajes de programación, incluyendo pruebas a los entornos de trabajo en donde son almacenadas las aplicaciones web, además de realizar comparaciones entre los distintos Framework. Proponer nuevos planes de auditoría basados en distintas herramientas en pruebas de penetración y metodologías para el análisis de vulnerabilidades, así como crear manuales para el uso de distintas herramientas de penetración

Referencias

1. Brian Hatch & James Lee (2003). *Hackers en Linux. Secretos y Soluciones para la Seguridad del Linux* 2 Edición. España: Mcgraw Hill.
2. Pecoraro C. (2015) *Mastering Laravel*, Packt Publishing.
3. Negus C. (2015) *Linux Bible* Eighth Edition, John Wiley & Sons, Inc.
4. Spafford G.(2001) *Web Security, Privacy & Commerce* 2nd Edition, O´Reilly Media Inc
5. Niederst J. (2012). *Learning Web Design* Fourth Edition, Canada, O´Reilly Media, Inc.
6. Stauffer M. (2016) *Laravel: Up and Running: A Framework for Building Modern PHP* Apps, O´Reilly Media.
7. Bean M. (2015*) Laravel 5 Essentials*, Packt Publishing.
8. Offensive Security OSCP (2019). *OWASP Testing Guide V4* Extraído de https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents
9. Pastor O. (2010). *Aplicaciones Web , Un enfoque práctico*, Editorial Omega.
10. Kim P. (2015). *The Hacker Playbook 2: Practical Guide to Penetration Testing*, Secure Planet LLC.
11. Hertzog R. (2015) *The Debian Administrator's HandBook*.