Caso de Estudio: Mi Primer Acercamiento Metodológico al Bug Bounty

Autor: Diego Thomas Valdés Villar **Fecha:** 30 de Septiembre, 2025

Resumen

Este documento detalla el proceso metodológico que seguí durante mi primera incursión estructurada en un programa de bug bounty de una importante plataforma de e-commerce y fintech en América Latina. El objetivo principal no fue el descubrimiento de una vulnerabilidad, sino la aplicación y el aprendizaje de un flujo de trabajo profesional de pruebas de seguridad, desde la configuración del entorno hasta el análisis de defensas complejas. Este caso de estudio sirve como documentación de las habilidades prácticas adquiridas en análisis de aplicaciones web y lógica de negocio.

Descargo de responsabilidad: *Todas las pruebas se realizaron siguiendo estrictamente las reglas y el alcance definidos en el programa público de bug bounty de la organización objetivo. Este documento no revela ninguna vulnerabilidad no divulgada ni información sensible.*

Fase 1: Configuración del Laboratorio y Definición de la Estrategia

El primer paso fue establecer un entorno de pruebas profesional para interceptar y analizar el tráfico web.

- **Herramienta Principal:** Se seleccionó **Burp Suite Community Edition** como el proxy de intercepción principal.
- **Configuración del Entorno:** Se configuró un navegador (Chromium) para dirigir su tráfico a través del proxy de Burp. Esto permitió la captura y el análisis de todas las peticiones HTTP/HTTPS entre el navegador y los servidores del objetivo.
- **Análisis del Objetivo:** Se estudió en profundidad la política del programa de bug bounty para comprender el alcance, las reglas de enfrentamiento y las áreas de mayor interés para la organización, incluyendo desafíos específicos como el **Bypass de Autenticación de Dos Factores (2FA)** y promociones en nuevos flujos de negocio.

Fase 2: Análisis de Vulnerabilidades de Inyección (Búsqueda de XSS)

La investigación inicial se centró en vulnerabilidades de tipo inyección, específicamente Cross-Site Scripting (XSS), debido a su prevalencia histórica en plataformas web.

1. **Identificación de Puntos de Entrada:** Se navegaron las funcionalidades públicas de la aplicación, identificando todos los campos donde el usuario ingresa datos que luego se reflejan en la página (buscadores, filtros, etc.).
2. **Pruebas Metódicas:** Se siguió un enfoque escalonado:
    - Se comenzó con inyecciones de HTML simples (`<h1>test</h1>`) para verificar si la aplicación procesaba las etiquetas. Se observó que la entrada era **sanitizada** (los caracteres `<` y `>` eran eliminados o codificados).
    - Se intentaron payloads de XSS básicos (`<script>alert(1)</script>`), lo que resultó en un bloqueo inmediato de la petición, confirmando la presencia de un **Web Application Firewall (WAF)**.
    - Se probaron payloads codificados (`%3C...%3E`) para intentar evadir el WAF. Esto permitió que la petición pasara, pero el servidor aplicó defensas secundarias como la **normalización** (conversión a minúsculas) y la **sanitización**.
3. **Conclusión de la Fase:** Se concluyó que la aplicación contaba con defensas robustas de múltiples capas contra ataques de XSS reflejado en sus funcionalidades públicas. El aprendizaje clave fue la identificación práctica de controles de seguridad como el **WAF** y la **sanitización de entradas**.

Fase 3: Pivote hacia el Análisis de Lógica de Negocio

Al encontrar defensas de inyección sólidas, la estrategia pivotó hacia el análisis de fallas en la lógica de negocio, enfocándose en un flujo de suscripción a un programa de beneficios.

1. **Mapeo del Flujo:** Utilizando Burp Suite, se capturó y mapeó cada paso del flujo de suscripción, desde la selección del plan hasta el inicio del proceso de pago.
2. **Formulación de Hipótesis:** Se teorizó que podría ser posible manipular el precio o el plan intercambiando identificadores de transacción a mitad del proceso. La hipótesis principal fue que el sistema podría no validar el estado de la sesión en cada paso.
3. **Ejecución del Ataque:**
    - Se capturaron los identificadores de preferencia (`preference_id`) para un plan barato y uno caro.
    - Se inició un flujo de pago con el plan barato.
    - Utilizando **Burp Repeater**, se interceptó una petición `PUT` intermedia que actualizaba el estado del flujo y se reemplazó el `preference_id` del plan barato por el del plan caro.
4. **Análisis de Resultados y Descubrimiento:** La respuesta del servidor (`200 OK`) fue analizada. Se descubrió que, aunque la petición fue aceptada, el sistema **ignoró el `preference_id` inyectado** y respondió con una URL de redirección que contenía el `preference_id` original y correcto del plan barato.
5. **Conclusión de la Fase:** Se determinó que la aplicación implementa un control de seguridad robusto de **validación de estado del lado del servidor**. El sistema no confía en los datos enviados por el cliente a mitad de un flujo, sino que se basa en el

estado inicial guardado de forma segura en la sesión. Se identificó exitosamente un **falso positivo** y se documentó un control de seguridad bien implementado.

Habilidades y Herramientas Desarrolladas

Este proyecto me permitió desarrollar y demostrar las siguientes habilidades prácticas:

- **Configuración y Uso Avanzado de Burp Suite:** Interceptación de tráfico, uso de **HTTP history** para mapeo y **Repeater** para manipulación de peticiones.
- **Análisis de Peticiones HTTP/HTTPS:** Comprensión de cabeceras, cookies y cuerpos de petición (JSON).
- **Metodología de Pruebas de Inyección:** Flujo de trabajo para la búsqueda de vulnerabilidades de tipo XSS.
- **Identificación de Controles de Seguridad:** Reconocimiento práctico de **WAFs**, **sanitización de entradas** y **normalización**.
- **Análisis de Lógica de Negocio:** Mapeo de flujos de múltiples pasos y formulación de hipótesis para encontrar fallas de lógica y control de estado.
- **Identificación de Falsos Positivos:** Capacidad para analizar resultados y discernir entre una vulnerabilidad real y un control de seguridad efectivo.

Conclusión Final

Aunque no se reportó una vulnerabilidad crítica, este proyecto fue un éxito rotundo en el cumplimiento de su objetivo principal: ejecutar un ciclo de pruebas de seguridad de principio a fin y desarrollar una base sólida de habilidades prácticas. El proceso de encontrar defensas robustas fue tan educativo como lo habría sido encontrar un fallo, proporcionando una visión invaluable de cómo las aplicaciones seguras son diseñadas y protegidas.