Seguridad en Aplicaciones Web mediante la Implementación de Múltiples Estrategias Defensivas

Ryan E. Vega Torres
Maestría en Ciencias de Computadoras
Mentor: Dr. Jeffrey Duffany
Graduate Project EXPO - Mayo 2026

Abstracto

Este proyecto evaluó vulnerabilidades comunes en aplicaciones web dentro de un laboratorio controlado con Kali Linux. Se analizaron OWASP Juice Shop, DVWA y OWASP Mutillidae II mediante pruebas manuales y automatizadas. Las pruebas incluyeron encabezados de seguridad, inyección SQL, XSS, fuerza bruta, inyección de comandos y enumeración de directorios. Los resultados mostraron que la combinación de herramientas y controles defensivos permite reducir la exposición, aunque no reemplaza la corrección de vulnerabilidades internas.

Introducción

Las aplicaciones web están expuestas a riesgos como configuraciones débiles, entradas mal validadas y controles de seguridad incompletos. Este proyecto evalúa aplicaciones web intencionalmente vulnerables mediante pruebas manuales y automatizadas en un laboratorio controlado, con el propósito de comparar su comportamiento ante distintos escenarios de seguridad.

Contexto y Problema

Las herramientas automatizadas ayudan a detectar vulnerabilidades, pero sus resultados pueden variar según la arquitectura y configuración de cada aplicación. Por eso, depender de una sola herramienta o control defensivo puede limitar el análisis. Este proyecto compara condiciones iniciales con escenarios mitigados o más restrictivos para observar cambios en la exposición y dificultad de explotación.

Metodología

El proyecto se desarrolló en un laboratorio controlado con Kali Linux, utilizando OWASP Juice Shop, DVWA y OWASP Mutillidae II como aplicaciones intencionalmente vulnerables para comparar distintos escenarios de seguridad. La evaluación combinó pruebas manuales con Burp Suite y validaciones automatizadas con herramientas como curl, SQLMap, XSStrike, Wfuzz y Dirsearch, lo que permitió analizar tanto el comportamiento de las solicitudes HTTP como los resultados generados por cada herramienta. Cada prueba partió de una condición inicial y luego fue comparada con escenarios más restrictivos, mitigaciones o cambios en niveles de seguridad para observar cómo variaban la exposición, la detección y la dificultad de explotación.

Resultados y discusión

Los resultados evidenciaron diferencias entre las condiciones iniciales, los niveles de seguridad más restrictivos y las mitigaciones aplicadas. La combinación de pruebas manuales y automatizadas permitió validar mejor los hallazgos, mientras que los controles defensivos redujeron parte de la exposición sin sustituir la corrección directa de las vulnerabilidades internas. Las evidencias muestran que los ataques fueron más efectivos en escenarios con configuraciones débiles y más difíciles de ejecutar cuando se aplicaron controles adicionales.

Figura 1
Página inicial de DVWA

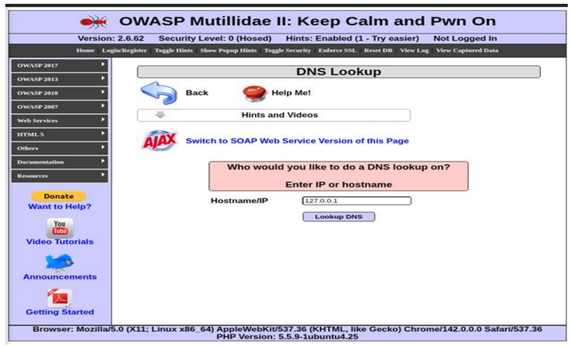

Figura 2
Página inicial de Mutillidae ll

```
  if echo "$resp" | grep -q "authentication"; then
    echo "SUCCESS with $p"
    break
  fi
  sleep 2
done
Trying: carlos1998
Trying: carlos98
Trying: carlos123
Trying: rivera1998
Trying: bayamon1998
Trying: bayamon98
Trying: rocky1998
SUCCESS with rocky1998
```

Figura 3
Evidencia de prueba automatizada de autenticación con credenciales predecibles.

Robustez de Autenticación y Resistencia a Intentos Automatizados		
Test Scenario	**Tool / Method**	**Evidence Observed**
Predictable Password Account	Automated script using profile-based password list	The script tested combinations based on profile details such as name, city, pet, and birth year.
Stronger Password Account	Automated script using other account	The script tested multiple generated combinations, but no successful login was identified.

Tabla 1
Comparación de robustez de autenticación frente a intentos automatizados.

```
Target: http://localhost:8081/vulnerabilities/brute/?username=admin&password=
FUZZ&Login=Login
Total requests: 4

ID           Response   Lines    Word       Chars       Payload

000000004:   200        108 L    250 W      4375 Ch     "test"
000000001:   200        108 L    250 W      4375 Ch     "1234"
000000002:   200        108 L    250 W      4375 Ch     "admin"
000000003:   200        108 L    254 W      4413 Ch     "password"
```

Figura 4
Evidencia de validación automatizada de fuerza bruta en DVWA mediante Wfuzz.

Autenticación por Fuerza Bruta		
Security Level	**Tool / Method**	**Evidence Observed**
Low	Wfuzz	Failed attempts showed 4375 characters, while password showed 4413 characters.
Impossible	curl with dynamic user_token	A valid user_token was extracted before each attempt. All responses returned 4636 bytes, including password.

Tabla 2
Comparación de autenticación por fuerza bruta según nivel de seguridad.

Conclusiones

Los resultados demostraron que las vulnerabilidades evaluadas pueden variar según la arquitectura, configuración y nivel de seguridad de cada aplicación. La combinación de pruebas manuales y automatizadas permitió confirmar mejor los hallazgos y entender el comportamiento real de cada escenario. Aunque las mitigaciones aplicadas redujeron parte de la exposición, una defensa efectiva requiere varias capas de protección junto con la corrección directa de vulnerabilidades internas.

Trabajo futuro

Como trabajo futuro, se propone aplicar estas estrategias en una aplicación web propia o en un sistema autorizado donde sea posible modificar directamente el código y la configuración. Esto permitiría comparar una versión vulnerable con una versión protegida, validar las mitigaciones con mayor control y evaluar cómo cambian los resultados después de corregir las vulnerabilidades internas.

Agradecimientos

Agradezco al Dr. Jeffrey Duffany por su guía y acompañamiento académico durante la preparación de este proyecto y su presentación en el *Graduate Project EXPO*.

Referencias

[1] I. Chorell and C. Ekberg, “A Comparative Analysis of Open Source Dynamic Application Security Testing Tools,” Linköping University, 2024.

[2] E. A. Altulaihan, A. Alismail, and M. Frikha, “A Survey on Web Application Penetration Testing,” Electronics, 2023, vol. 12, no. 5, p. 1229.

[3] X. Wang, J. Zhai, and H. Yang, “Detecting Command Injection Attacks in Web Applications Based on Novel Deep Learning Methods,” Scientific Reports, 2024, vol. 14.

[4] U. Kishnani and S. Das, “Securing the Web: Analysis of HTTP Security Headers in Popular Global Websites,” 2024.