**Informe Técnico**

**Explotación de Vulnerabilidades XSS Almacenado y CSRF en DVWA – Entorno Controlado**

**1. Escenario**

Durante una auditoría ética en la plataforma **DVWA (Damn Vulnerable Web Application)**, se identificaron dos vulnerabilidades críticas:

* **XSS Almacenado (Stored XSS):** ejecución de código JavaScript inyectado en el campo de comentarios.
* **CSRF (Cross-Site Request Forgery):** manipulación de formularios que permite realizar acciones sensibles, como el cambio de contraseña, sin consentimiento del usuario autenticado.

El objetivo del laboratorio fue **demostrar la explotación controlada** de ambas vulnerabilidades y proponer medidas de mitigación bajo estándares OWASP.

**2. Metodología**

1. Configuración de DVWA en nivel de seguridad *Low*.
2. Inyección del payload XSS en el módulo **XSS (Stored)** y validación de su ejecución en navegadores de distintos usuarios.
3. Creación de un archivo **HTML malicioso** con un formulario oculto para explotar CSRF en el módulo de cambio de contraseña.
4. Validación de resultados observados: ejecución de alertas en navegador y modificación no autorizada de credenciales.

**3. Evidencia Técnica**

**3.1 XSS Almacenado**

* **Payload utilizado:**
* <script>alert('XSS almacenado')</script>
* **Resultado observado:** al recargar la página, el script se ejecutó automáticamente en el navegador de cualquier usuario, confirmando la persistencia del ataque.

**3.2 CSRF en cambio de contraseña**

* **Código malicioso (HTML):**
* <form action="http://localhost/dvwa/vulnerabilities/csrf/" method="POST">
* <input type="hidden" name="password\_new" value="hack123">
* <input type="hidden" name="password\_conf" value="hack123">
* <input type="hidden" name="Change" value="1">
* <input type="submit" value="Haz clic aquí">
* </form>
* **Resultado observado:** al abrir el archivo con sesión activa, la contraseña del usuario fue cambiada sin su consentimiento.

**4. Evaluación del Riesgo**

* **XSS Almacenado:** permite ejecución de scripts maliciosos, robo de cookies de sesión, redirecciones y phishing.
* **CSRF:** permite realizar acciones críticas en nombre de usuarios autenticados, comprometiendo la **integridad y disponibilidad** de las cuentas.
* **Severidad (CVSS v3.1):** Crítica (8.8 – 9.1).

**5. Recomendaciones de Seguridad**

1. **Para XSS:**
   * Escapar y sanitizar todas las entradas y salidas en HTML.
   * Implementar **Content Security Policy (CSP)**.
2. **Para CSRF:**
   * Incluir **tokens CSRF únicos y validados en el servidor**.
   * Validar el origen de las peticiones (Origin y Referer).
3. Reforzar cabeceras HTTP de seguridad (**HSTS, X-Frame-Options, X-XSS-Protection**).
4. Implementar autenticación multifactor (MFA) para cuentas críticas.
5. Educar a desarrolladores en estándares **OWASP Top 10**.

**6. Reflexión**

* **XSS almacenado vs reflejado:** el almacenado es más grave, ya que persiste en la aplicación y afecta a múltiples usuarios.
* **Peligro de CSRF:** su efectividad aumenta en usuarios autenticados, al ejecutar acciones sin su conocimiento.
* **Relación defensa:** validación de origen y tokens anti-CSRF son barreras esenciales que hubieran mitigado el ataque.

**7. Conclusión**

El análisis confirmó que DVWA es vulnerable a **XSS almacenado y CSRF**, ambos con un impacto crítico en la seguridad de usuarios y aplicaciones. Las pruebas controladas demostraron cómo un atacante podría explotar estas fallas para ejecutar código arbitrario o modificar credenciales. Este laboratorio refuerza la necesidad de implementar **validación de entradas, tokens de seguridad y cabeceras protectoras**, siguiendo estándares OWASP, para garantizar la resiliencia de aplicaciones web frente a ataques comunes.