Informe de Práctica de Seguridad Informatica: CSRF

Nombre: Daniel Alejandro Trigo Tabilo

1. Introducción

El presente informe documenta una práctica de laboratorio enfocada en la explotación de una vulnerabilidad de tipo Cross-Site Request Forgery (CSRF). Se utilizó la plataforma vulnerable DVWA para comprender el funcionamiento e impacto real de este tipo de ataques.

2. Entorno de trabajo

- Aplicación utilizada: Damn Vulnerable Web Application (DVWA)
- Nivel de seguridad: Low
- Navegador utilizado: Firefox
- Sistema Operativo: Kali Linux
- Entorno: VirtualBox

3. Desarrollo del ataque

El objetivo del ataque fue realizar una modificación de contraseña de un usuario autenticado en DVWA, sin su consentimiento. Para lograrlo se llevaron a cabo los siguientes pasos:

1. Inicio de sesión en DVWA como usuario legítimo.
2. Se accedió a la sección "CSRF" con nivel de seguridad Low.
3. Se identificó un formulario vulnerable que permite cambiar la contraseña sin verificar la contraseña anterior ni el origen de la petición.
4. Se diseñó un código HTML malicioso, alojado en una página externa, que realiza la petición automáticamente usando un formulario oculto.

Payload utilizado:

```
<form action="http://localhost/dvwa/vulnerabilities/csrf/" method="POST">
 <input type="hidden" name="password_new" value="123456" />
 <input type="hidden" name="password_conf" value="123456" />
 <input type="submit" value="Enviar" />
</form>
<script>
 document.forms[0].submit();
</script>
```

5. Al visitar la página con el payload (con sesión activa en DVWA), el cambio de contraseña se ejecuta sin interacción del usuario.

4. Análisis del impacto

Este ataque demuestra cómo un atacante puede realizar acciones maliciosas aprovechándose de la confianza que un servidor tiene en el navegador de un usuario autenticado. En un entorno real, podría:
- Cambiar contraseñas.
- Transferir fondos.
- Eliminar información.
- Alterar configuraciones sensibles.

5. Medidas de mitigación

Para prevenir este tipo de ataques, se recomienda implementar:
- Tokens anti-CSRF en formularios.
- Validación del encabezado Origin o Referer.
- Uso de cookies con atributo SameSite=Strict o Lax.
- Verificación de la contraseña actual en formularios críticos.
- Restricción del uso de métodos GET para acciones sensibles.

6. Conclusión

La práctica realizada permitió evidenciar una de las vulnerabilidades web más comunes y peligrosas. Comprender su funcionamiento y mitigación resulta esencial para el desarrollo de aplicaciones web seguras.

7. Referencias

- OWASP: https://owasp.org/www-community/attacks/csrf
- DVWA GitHub: https://github.com/digininja/DVWA
- Mozilla Docs: https://developer.mozilla.org/en-US/docs/Web/Security/SameSite