Fundamentos de la Seguridad Informática
Seguridad en Aplicaciones

Grupo de Seguridad Informática
Facultad de Ingeniería
Universidad de la República

Propósito

El propósito de las prácticas descritas en este documento es introducir a los participantes en los problemas relacionados con la seguridad de las aplicaciones. Las mismas permiten visualizar las vulnerabilidades más típicas.

1. Aplicaciones Web (WebGoat)

En esta práctica veremos los problemas comunes en aplicaciones Web, utilizando herramientas del proyecto OWASP [1].

Herramientas

- OWASP WebGoat [2]
- OWASP WebWolf [3]

1.1. Presentación

WebGoat es una herramienta (se podría decir que es un Framework) que permite crear aplicaciones inseguras en un ambiente controlado. Cada una de las aplicaciones inseguras es componente de una lección. En cada lección los usuarios deben demostrar la comprensión del problema planteado y explotar la vulnerabilidad propuesta. Por ejemplo, en una de las lecciones el usuario debe utilizar "SQL injection" para obtener números de tarjetas de crédito. Genera un ambiente de enseñanza realista y provee pistas a los usuarios.

WebWolf es un complemente del WebGoat, que simula ser un conjunto de servicios que puede utilizar el atacante como parte de su infraestructura para montar un ataque. Entre otros incluye un servidor de e-mail, un fileserver donde publicar archivos, etc.

Como se presentó en el teórico el OWASP Top Ten es un proyecto de OWASP [1] que prioriza los 10 principales riesgos sobre aplicaciones web. En el teórico se presenta la versión 2017 y el WebGoat está ordenado en base a la versión 2021. En la figura 1 se presenta la relación entre los riesgos del OWASP Top Ten 2017 y 2021.

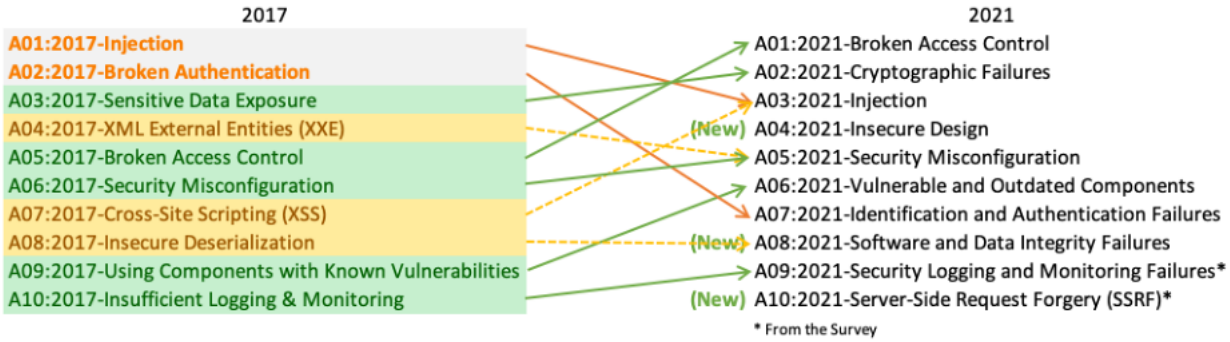

Figura 1: Relación entre los riesgos del OWASP Top Ten 2017 y 2021 (https://owasp.org/www-project-top-ten/)

1.2. Escenario

Para esta práctica se utilizará un puesto de trabajo con un navegador instalado, la máquina atacante utilizada para los otros laboratorios para acceder a la red privada y un servidor con WebGoat instalado en la dirección `10.0.x.5`, donde `x` es el número de grupo.

Guía de la práctica

Para acceder a la instalación de WebGoat del laboratorio, se debe:

- Generar un túnel ssh que conecte puertos altos (8080 y 9090 en el ejemplo) del PC personal con el puerto 80 (WebGoat) y 9090 (WebWolf) del servidor, utilizando `lulu` y el atacante como máquinas intermedias:

```
ssh -L 8080:10.0.x.5:80 -L 9090:10.0.x.5:9090 -J <usuario_fing>@lulu.fing.edu.uy fsi@192.168.44.(x+10)
```

- Conectarse a la URL del WebGoat utilizando la siguiente URL:
  `http://localhost:8080/WebGoat`[1]
  y al WebWolf con esta URL:
  `http://localhost:9090`

- Registrar un nuevo usuario en el sitio.

- Se puede utilizar libremente las *hints* de las lecciones que las tengan.

[1] El sistema es *case-sensitive*, por lo que se debe usar W y G mayúsculas.

Como mínimo se deben realizar los siguientes ejercicios de las lecciones indicadas:

Ejercicios recomendados		
Categoría	**Lección**	**Ejercicios**
Introduction	WebGoat	Todos
	WebWolf	Todos
General	HTTP Basics	Todos
	HTTP Proxies	Todos
	Developer Tools	Todos
	CIA Triad	Todos
Ejercicios obligatorios		
Categoría	**Lección**	**Ejercicios**
(A1) Broken Access Control	Insecure Direct Object Reference	Todos
	Missing Function Level Access Control	Todos
	Spoofing an Authenticaton Cookie	Todos
(A2) Cryptographic Failures	Crypto Basics	Todos excepto el 8
(A3) Injection	SQL Injection (intro)	Todos
	SQL Injection (advanced)	Todos
	SQL Injection (mitigation)	Todos
	Path Trasversal	Todos
	Cross Site Scripting	Todos
(A5) Security Misconfiguration	XXE	Todos
(A6) Vuln & Outdated Components	Vulnerable Components	Todos excepto el 12
(A7) Identity & Auth Failure	Authentication Bypasses	Todos
	Insecure Login	Todos
	JWT Tokens	1 al 10
	Password Reset	Todos
	Secure Passwords	Todos
(A9) Security Logging Failures	Logging Security	Todos excepto el 4
(A10) Server-side Request Forgery	Cross-site request forgeries	Todos
	Server-side request foregery	Todos
Client Side	Bypass frontend restrictions	Todos
	Client side filetering	Todos
	HTML Tampering	Todos

Objetivos de Aprendizaje

- Comprender del Top Ten de OWASP al menos los siguientes riesgos: inyección, pérdida de autenticación y problemas criptográficos.
- Entender las vulnerabilidades catalogadas como *Client Side* y la diferencia entre controles de seguridad versus usabilidad

2. Análisis de seguridad mediante mecanismos de *debugging* de aplicaciones

Objetivos

Introducir los conceptos básicos de análisis de código y problemas de programación.

En el desarrollo de software, errores en la programación pueden derivar consecuentemente en una vulnerabilidad crítica que llegue a poner en peligro la seguridad de un sistema. Sin embargo, muchas veces una vulnerabilidad de seguridad no se debe a un solo error, sino a una secuencia de errores que ocurren durante el ciclo de desarrollo: hay un defecto en la codificación, éste permanece sin ser detectado durante las fases de pruebas y los controles existentes no detectan ni previenen un ataque que permite vulnerar el sistema.

Herramientas

Para esta práctica se utilizará una máquina con el debugger *gdb* instalado y la aplicación a analizar. La aplicación se encuentra con los archivos asociados al laboratorio 4.

Guía de la práctica

1. La aplicación lee un string de entrada y la compara con una clave ("`GSIlab4`"). La primer parte consiste en hacer que la aplicación acepte la clave "`ESTlab4`".
2. Modificar el código fuente entregado para eliminar la vulnerabilidad encontrada.
3. Agregar nueva/s funcionalidad/es a la aplicación que agreguen 2 nuevos problemas de seguridad de índole diferente al encontrado.

Se pide:

- Parte a)
  - Compilar la aplicación, utilizando la opción de gcc `-fno-stack-protector`.
  - Explicar detalladamente cómo logró autenticarse con la clave pedida, indicando el string de entrada utilizado
- Parte b)
  - Entregar el código fuente modificado con la solución.
  - Entregar un patch que se pueda aplicar sobre el código original para solucionar el problema.
- Parte c)
  - Entregar el código fuente modificado considerando los nuevos problemas.
  - Explicar los cambios realizados, describiendo los problemas de seguridad agregados.

Forma de Entrega

Se debe entregar el archivo entrega4.tar.gz conteniendo los archivos requeridos en cada práctica.

La página de entrega es la correspondiente en EVA.

Referencias

[1] OWASP Foundation. Owasp project. http://www.owasp.org.

[2] OWASP Foundation. Owasp webgoat project. http://www.owasp.org/index.php/Category:OWASP_WebGoat_Project.

[3] OWASP Foundation. Webwolf the small helper. https://owasp.org/www-project-webgoat/#div-webwolf.