Año 2025

Desarrollo Seguro de Aplicaciones
Práctica 4
Configuración incorrecta de seguridad, Componentes vulnerables y obsoletos, Fallos de identificación y autenticación

Fecha de entrega: 24/06/2024 a las 17:59hs

Términos relacionados:

- OWASP top ten 2021: A5, A6 y A7

Guía para hacer la práctica:

1. Continuaremos utilizando lo que vimos en la práctica anterior: el repositorio grupal de código en GitHub y Docker con docker-compose.
2. Ejecute **git pull origin master** en la carpeta del repositorio para bajar los últimos cambios.
3. Ejecute los ejercicios accediendo a la carpeta /**practica4** y corriendo el comando ***./run.sh***
4. Para acceder vía web a cada reto deberá ingresar al puerto que corresponda.

Ejercicio 1 - 2021-A05: Configuración incorrecta de seguridad

Reto **Default password** (http://localhost:14001)

Levante la aplicación con:

```
cd practica4/ejercicio1; docker-compose up
```

➔ Detecte qué aplicación es y encuentre las credenciales por defecto del administrador de dicho sistema.

Ejercicio 2 - 2021-A06: Componentes vulnerables y obsoletos

Reto **Joomla** (http://localhost:14002/):

Levante la aplicación con:

```
cd practica4/ejercicio2; docker-compose up
```

- ➔ Identifique la versión de Joomla. Busque vulnerabilidades para esa versión en https://www.exploit-db.com/
- ➔ Explote una vulnerabilidad exitosamente.

***Hint de los argumentos para explotar la vulnerabilidad** (para ahorrar tiempo):*

```
--risk=3 --level=5 --random-agent -p list[fullordering] --technique E
--dbs
```

Ejercicio 3 - 2021-A07: **Fallos de identificación y autenticación**

Reto **Python login** (http://localhost:14003/):

Levante la aplicación con:

```
cd practica4/ejercicio3; ./run.sh
```

Explotación:

- ➔ Explote el login vulnerable a fuerza bruta para conseguir la password del usuario "**admin**" utilizando el diccionario "rockyou.txt" (https://github.com/brannondorsey/naive-hashcat/releases/download/data/rockyou.txt)
- ➔ Para explotar la vulnerabilidad puede usar:
  - ◆ Hydra, reemplazando los valores
    `hydra -l admin -P <PATH_A_ROCKYOU.txt> "http-form-post://<IP>:<PUERTO>/<ENDPOINT>:<USERNAMEPARAM>=^USER^&<PASSWORDPARAM>=^PASS^:<STRING_LOGIN_INCORRECTO>"`
  - ◆ Hacer su propio script de fuerza bruta en python, utilizando la librería requests

Fix:

- ◆ Fixee el código para que no sea vulnerable a fuerza bruta
- ◆ Que se pueda loguear normalmente con el **usuario admin y su password original**
- ◆ Que **no** sea posible realizar un ataque simple de fuerza bruta utilizando el diccionario rockyou.txt (en Kali: /usr/share/wordlists/rockyou.txt)
- ◆ Si implementa captcha, debe controlar en server side, no solamente en javascript del cliente.
- ◆ No es necesario una implementación muy compleja, sólo tiene que pasar la prueba que haremos para verificar el ejercicio.

Ejercicio 4 - CTF

- ➔ Resuelva los ejercicios del CTF de la categoría "Práctica 4"
  - ◆ https://loginbruteforce.dsa.linti.unlp.edu.ar/

Ejercicio 5 - Entrega de Informe

- ➔ Desarrolle un informe en el que explique cómo logró explotar cada una de las vulnerabilidades de los ejercicios 1, 2 y 3.
- ➔ Explique qué solución aplicó en el ejercicio 3.
- ➔ Entregue la tarea en https://catedras.linti.unlp.edu.ar en formato PDF.