UNC Universidad Nacional de Córdoba
FAMAF Facultad de Matemática, Astronomía, Física y Computación

# Práctico 2 - Seguridad Web

**Deadline parte ENTREGABLE: 30 de Septiembre 23:59**
Sugerencia: Trabajar desde la máquina virtual con Kali instalada en el TP0.

1. Practicar los diferentes tipos de vulnerabilidades dentro de la ruta */dvwa* que se encuentra en el servidor web: “OWASP Broken Web Applications”. (Credenciales por defecto: admin:admin)

2. Encontrar la información (flag) que se encuentra en: http://paste.ubuntu.com/p/HnGHwGk4rQ/

3. Practicar las diferentes vulnerabilidades estudiadas de tipos: SQL injection, XSS, CSRF, CORS, Insecure Deserialization, Directory Traversal mediante la plataforma: https://portswigger.net/web-security

4. Instalar Docker: https://docs.docker.com/get-docker/
   Instalar Docker Compose: https://docs.docker.com/compose/install/

   Descargar el archivo .tar.gz:
   https://drive.google.com/file/d/14SGjIBK0msHeoIZ5q-v-gqGCTiU_DXdA/view

   Descomprimir y dentro de la carpeta ‘amarelo-designs’ ejecutar:
   make install

   Luego ingresar a la aplicación a través de:
   http://localhost:10008/admin

   Acceder a la plataforma con admin:admin.

   Capturar los headers empleando un proxy, buscar la cookie sessionId.

   Decodificar la cookie y deserializar la misma usando la función loads, del módulo pickle de python.

   Tratar de explotar la aplicación web, ejecutando el comando ping y capturandolo (como se realizó en el práctico 0 mediante la herramienta tcpdump).

5. Determinar qué valor o valores de entrada del servidor OWASPVWA en la ruta /cgi-bin/test.cgi (instalada en el Práctico 0) son vulnerables a shellshock. Verificar utilizando un web-proxy.

# Parte entregable

1. Ingresar a la bóveda "Gringotts" y recuperar el secreto del usuario bob.
   http://143.0.100.198:5000

   **********
   * Para este problema no deberá utilizar herramientas automáticas. *
   **********

   a. Reporte el proceso de obtención del secreto de bob.
   b. ¿Se puede obtener la cantidad de usuarios del sistema?
   c. ¿Se puede obtener cuales son?

2. Acceder a: http://143.0.100.198:6789

   a. ¿Qué tecnologías usa este servicio web?
   b. ¿Qué vulnerabilidad/es tiene?
   c. ¿Ejemplifica una manera de explotarla?

3. Acceder a: http://143.0.100.198:5010

   a. ¿Qué vulnerabilidad/es existen en dicha aplicación?
   b. ¿Es posible conseguir el código de la aplicación?

4. Analizar el siguiente programa en PHP en busca de vulnerabilidades.
   a. Describa el proceso y un request válido que trate de explotarlas.

```php
<?php
class LangMgr{
    public function newLang(){
        $lang = $this->getBrowserLang();
        $sanitizedLang = $this->sanitizeLang($lang);
        require_once("/lang/$sanitizedLang");
    }

    private function getBrowserLang(){
       $lang = $_SERVER['HTTP_ACCEPT_LANGUAGE'] ?? 'en';
       return $lang;
    }

    private function sanitizeLang($lang){
        return str_replace('../', '', $lang);
    }
}

(new LangMgr())->newLang();
?>
```

   b. Entregar un parche del código para mitigar las vulnerabilidades encontradas.

5. Realizar un ataque XSS a alguna de las rutas de la máquina virtual OWASPVWA:
   - dvwa/vulnerabilities/xss_r/
   - dvwa/vulnerabilities/xss_s/

   Documentar y entregar un ataque que capture las teclas presionadas por un usuario de la aplicación vulnerable.

6. La empresa ficticia Setbrains se jacta de que tienen 0 vulnerabilidades de autenticación. ¿Puedes demostrarles que esto no es cierto?
   http://143.0.100.198:5001

   **********
   * Para este problema no deberá utilizar herramientas automáticas. *
   **********

   a. Reportar la flag que demuestra que han logrado vulnerar a la aplicación.
   b. Escribir una breve descripción de cuál fue el error.

   TIP: Si bien no tienen que usar herramientas automáticas para resolver este challenge. Deberán investigar un poco los distintos paths para encontrar la pista que les dará el inicio para la vulnerabilidad.