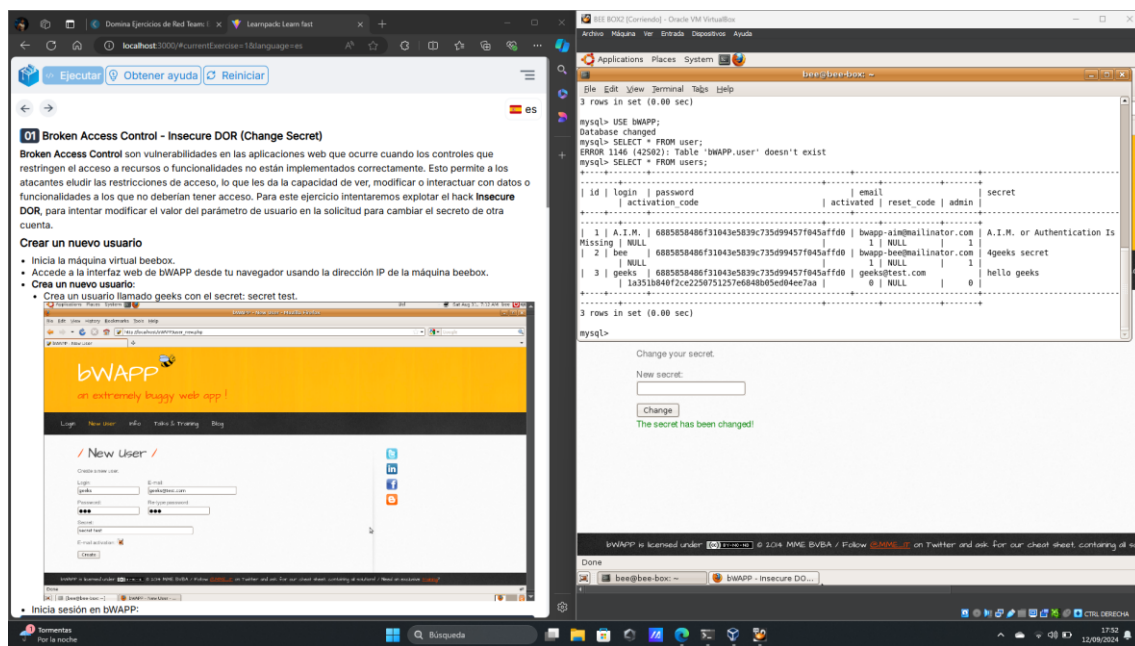


Explotación de Vulnerabilidades del OWASP Top 10

1- Broken Access Control - Insecure DOR (Change Secret)

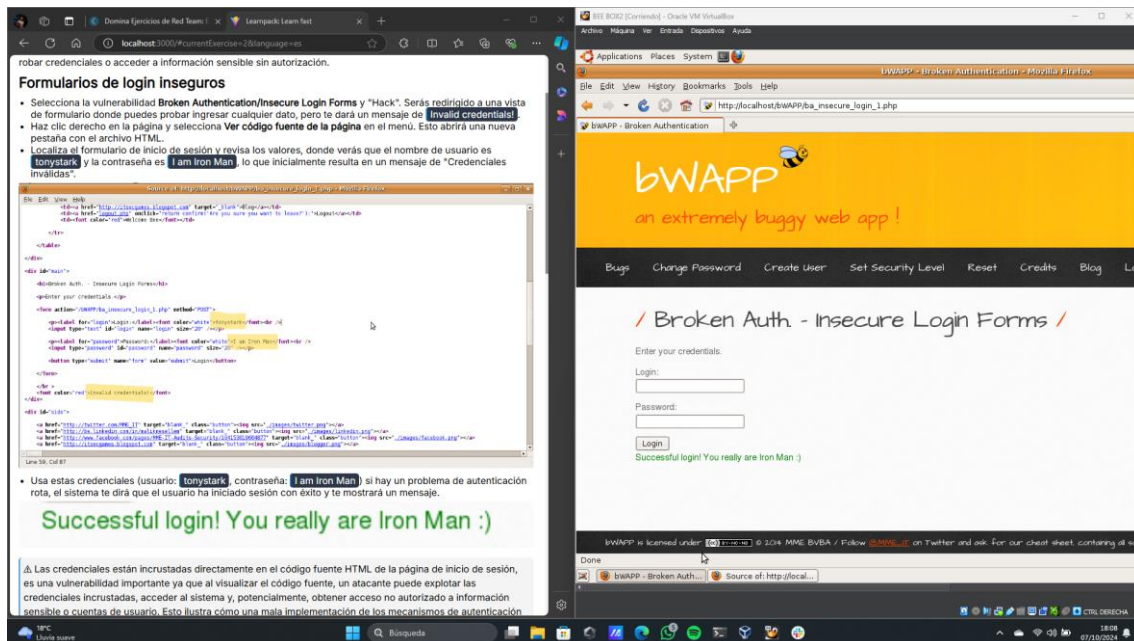
Broken Access Control son vulnerabilidades en las aplicaciones web que ocurre cuando los controles que restringen el acceso a recursos o funcionalidades no están implementados correctamente. Esto permite a los atacantes eludir las restricciones de acceso, lo que les da la capacidad de ver, modificar o interactuar con datos o funcionalidades a los que no deberían tener acceso.

Evidencia.



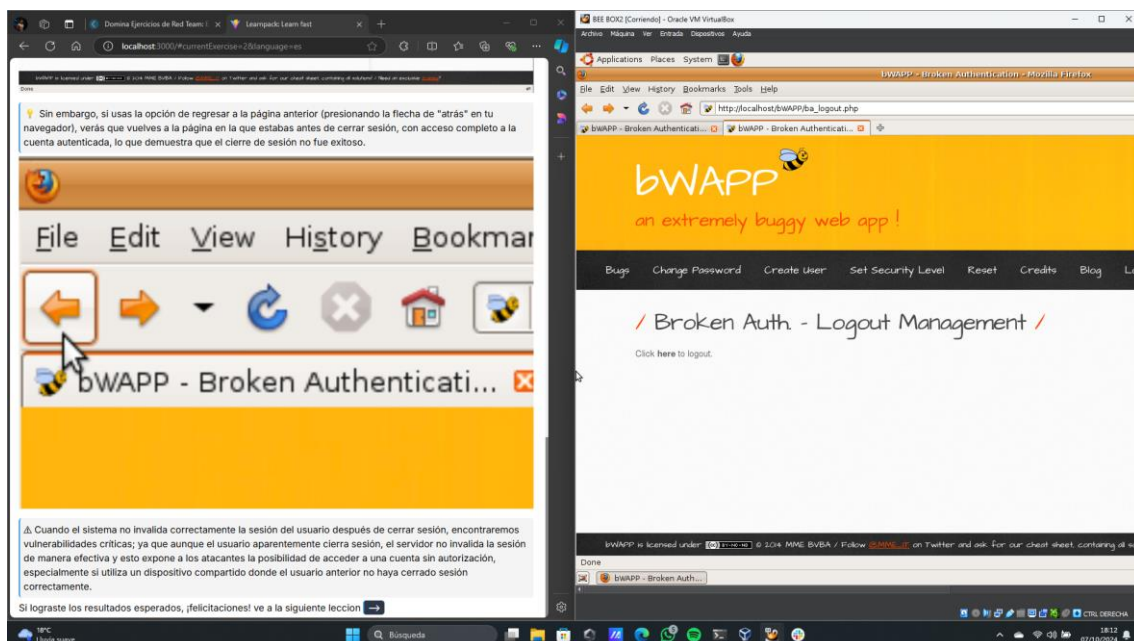
2- Identification & authentication failures - Broken Authentication

La autenticación rota es una vulnerabilidad de seguridad crítica que surge cuando las aplicaciones web no gestionan adecuadamente la autenticación de los usuarios. Puede permitir a los atacantes comprometer cuentas, robar credenciales o acceder a información sensible sin autorización.



Gestión de cierre de sesión.

Cuando el sistema no invalida correctamente la sesión del usuario después de cerrar sesión, encontraremos vulnerabilidades críticas; ya que aunque el usuario aparentemente cierra sesión, el servidor no invalida la sesión de manera efectiva y esto expone a los atacantes la posibilidad de acceder a una cuenta sin autorización, especialmente si utiliza un dispositivo compartido donde el usuario anterior no haya cerrado sesión correctamente.

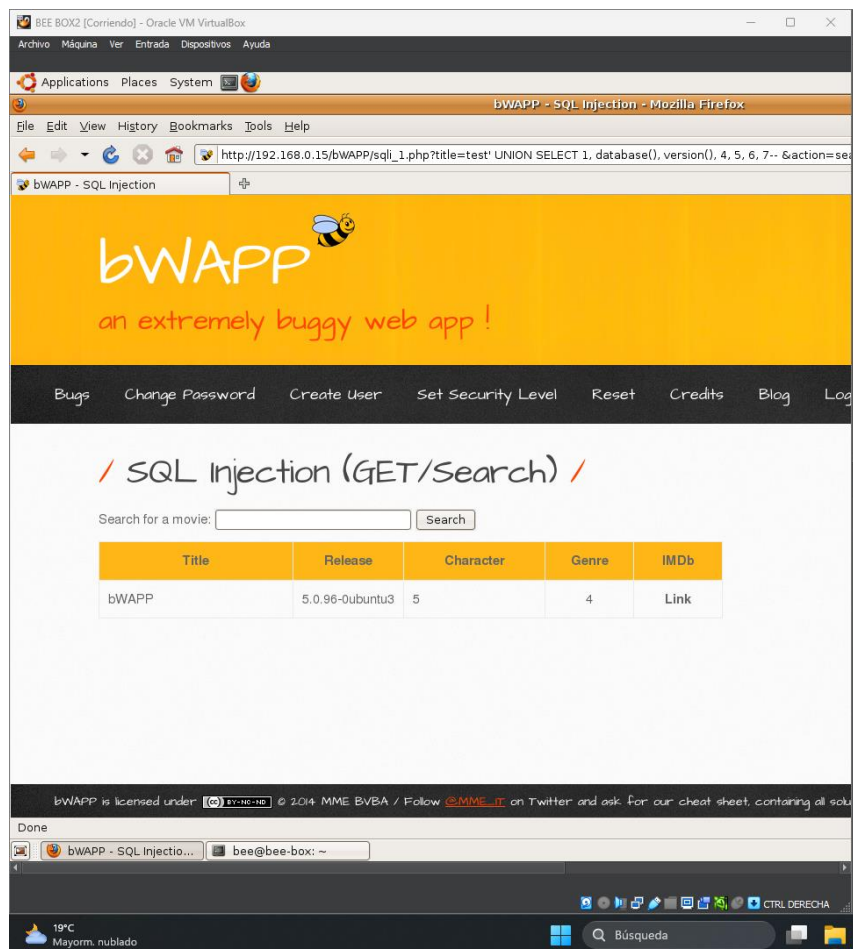


3- Injection - SQL injection.

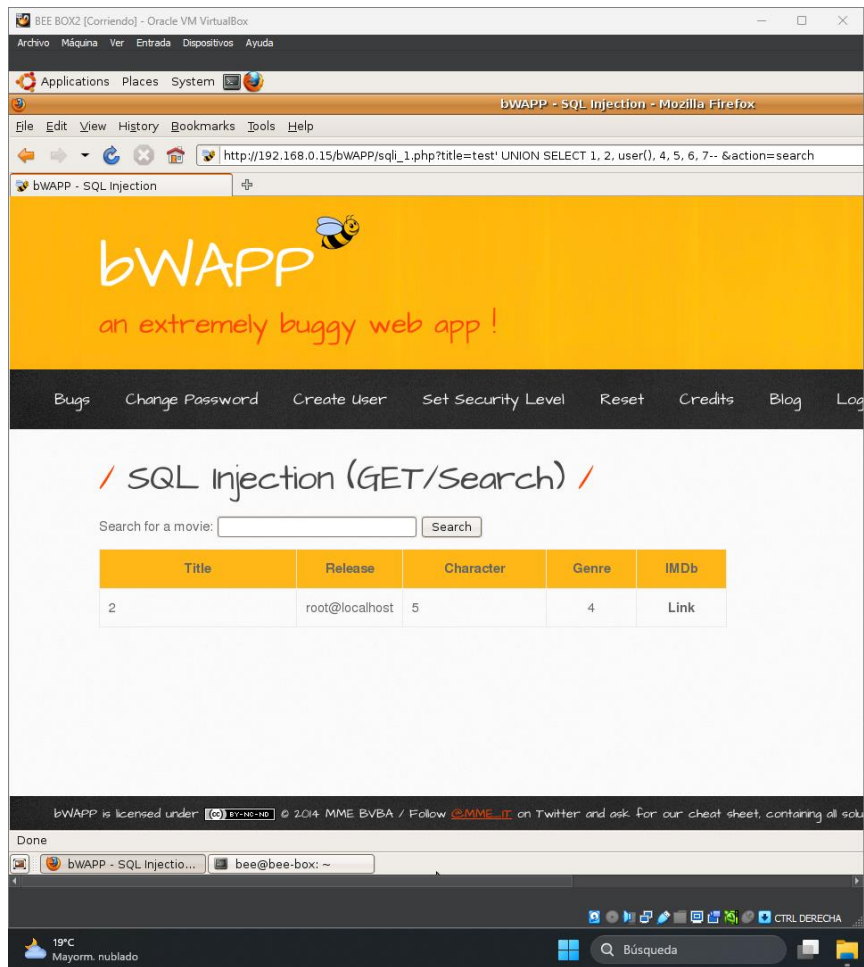
La **inyección SQL** es una vulnerabilidad de seguridad en aplicaciones web que ocurre cuando un atacante puede insertar o "inyectar" código SQL malicioso en una consulta que una aplicación envía a una base de datos. Este ataque permite que el atacante manipule o controle la consulta SQL que realiza la aplicación, lo que puede dar lugar a acceso no autorizado a datos sensibles, modificación de datos, eliminación de registros, o incluso el control total del sistema de base de datos.

Inyección con UNION SELECT.

Esto mostrará el nombre de la base de datos y la versión del servidor SQL.

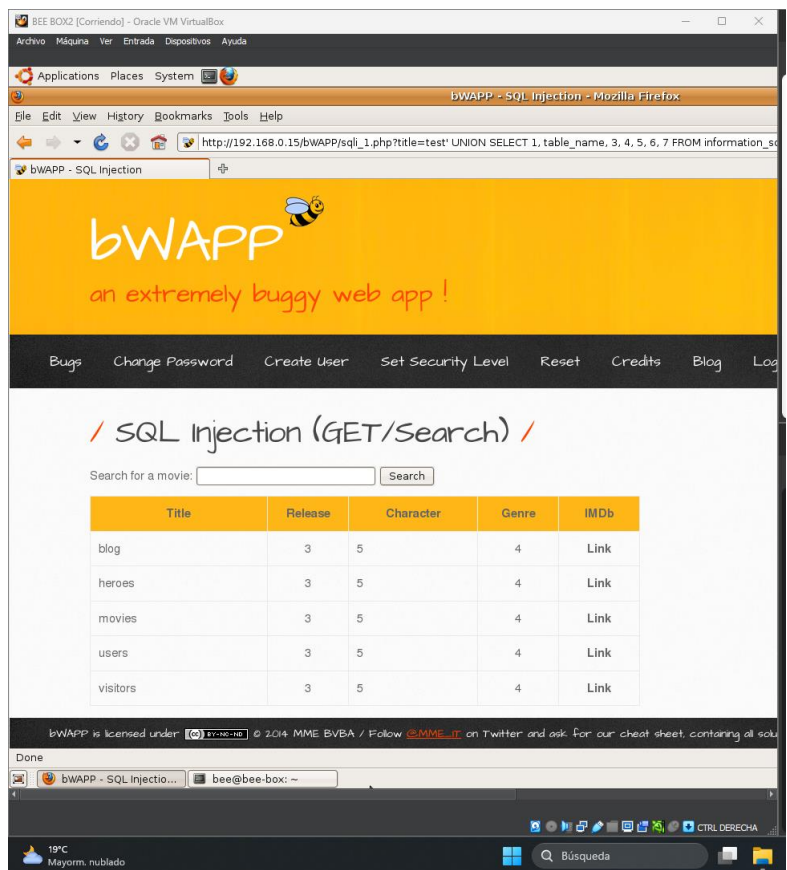


Obtención del nombre del usuario actual.

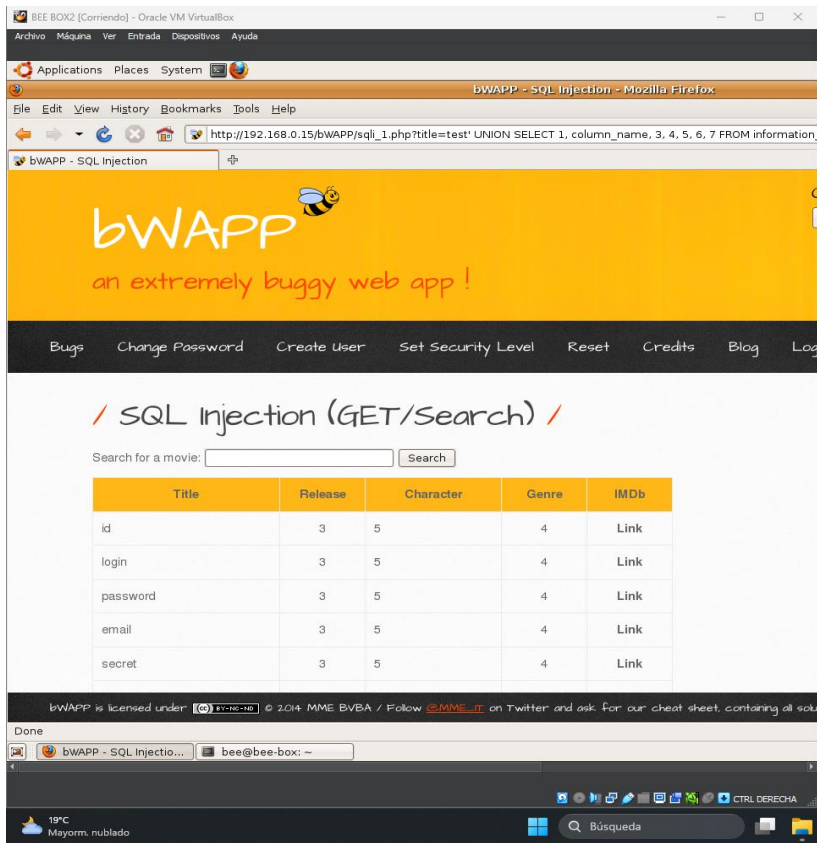


Obtención de los nombres de las tablas.

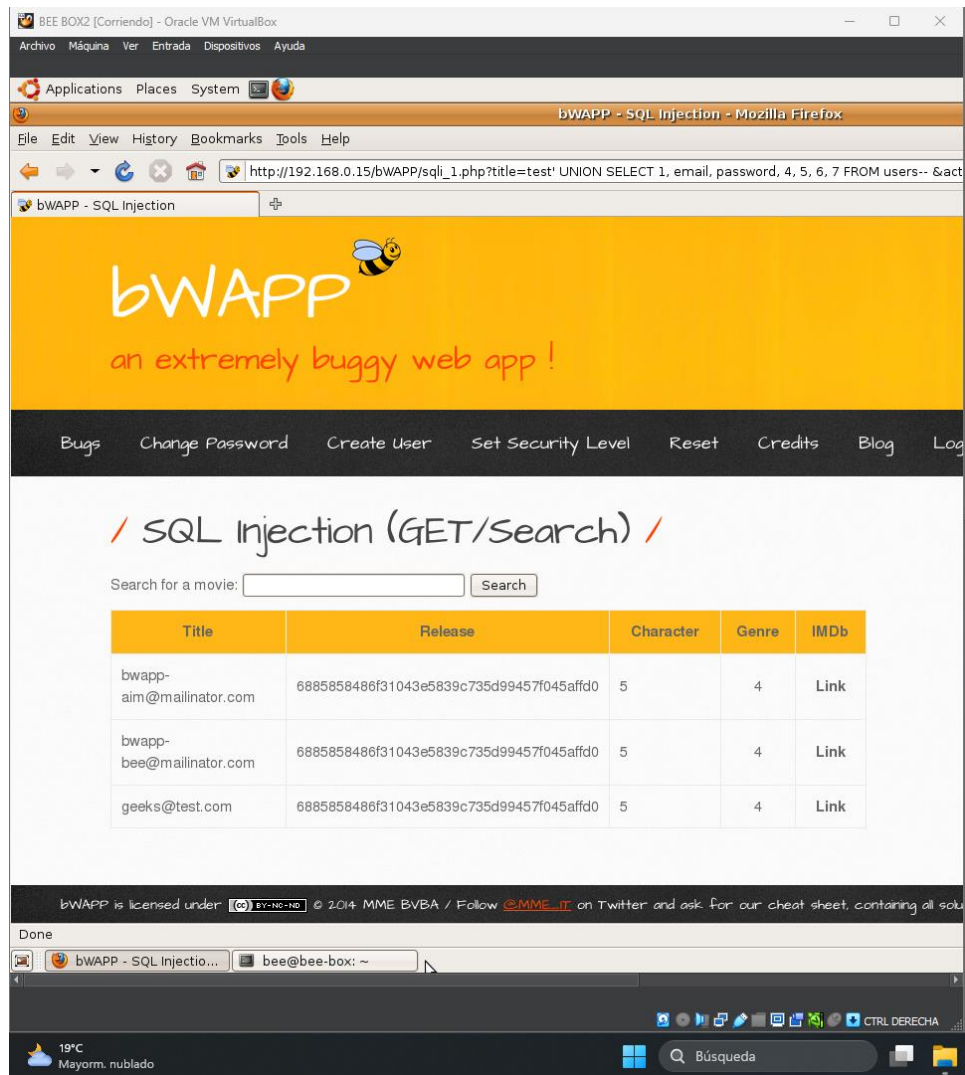
Realizamos una consulta UNION SELECT para acceder a la tabla information_schema.tables, que contiene los nombres de las tablas en la base de datos actual.



Obtener nombres de columnas.



Obtener datos sensibles.



4- Injection - Cross-site scripting (XSS) Reflected GET and POST.

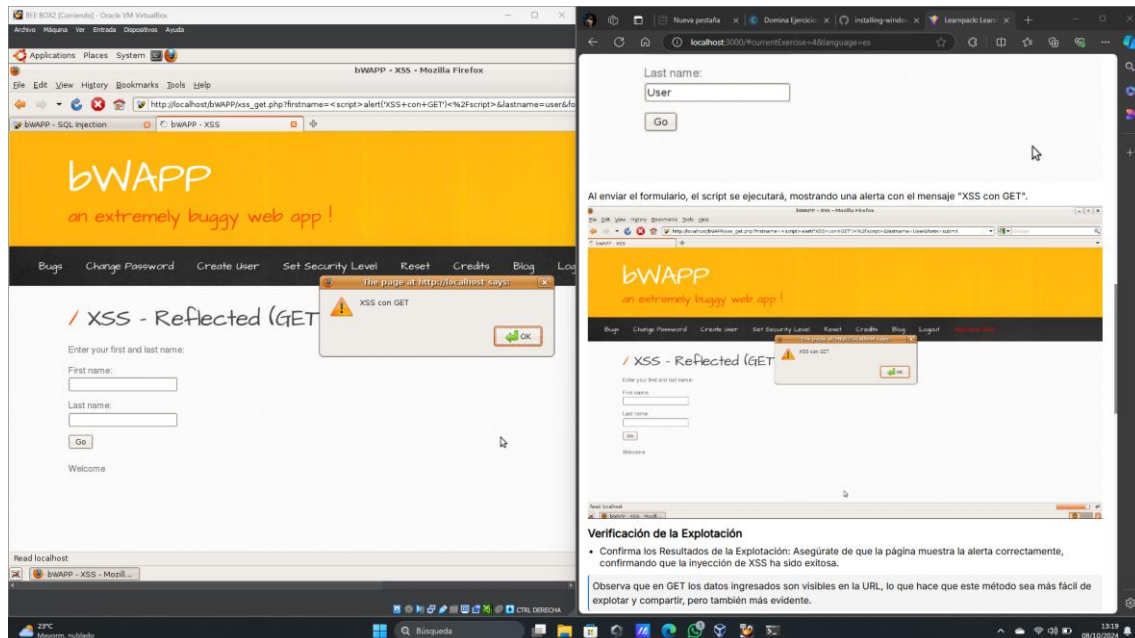
El **Cross-Site Scripting (XSS) Reflected**, o **XSS reflejado**, es un tipo de vulnerabilidad de seguridad en aplicaciones web donde un atacante inyecta código malicioso, generalmente en lenguaje JavaScript, que se refleja (se devuelve) en la respuesta de la aplicación web a un usuario.

Sobre GET y POST en XSS:

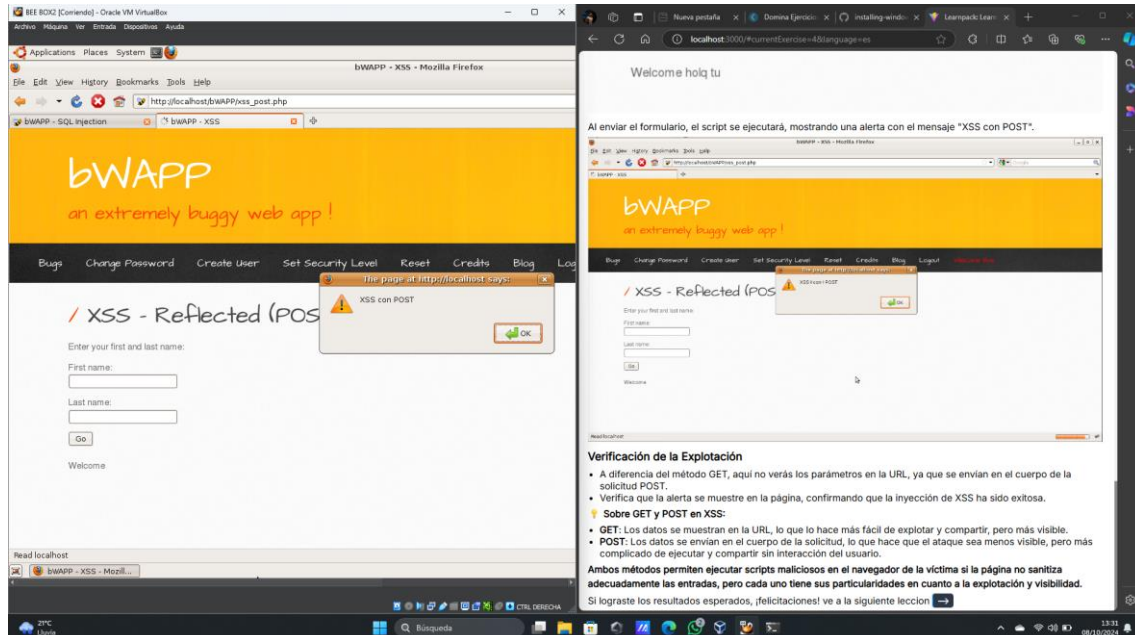
- **GET:** Los datos se muestran en la URL, lo que lo hace más fácil de explotar y compartir, pero más visible.

- **POST:** Los datos se envían en el cuerpo de la solicitud, lo que hace que el ataque sea menos visible, pero más complicado de ejecutar y compartir sin interacción del usuario.

XSS con GET.



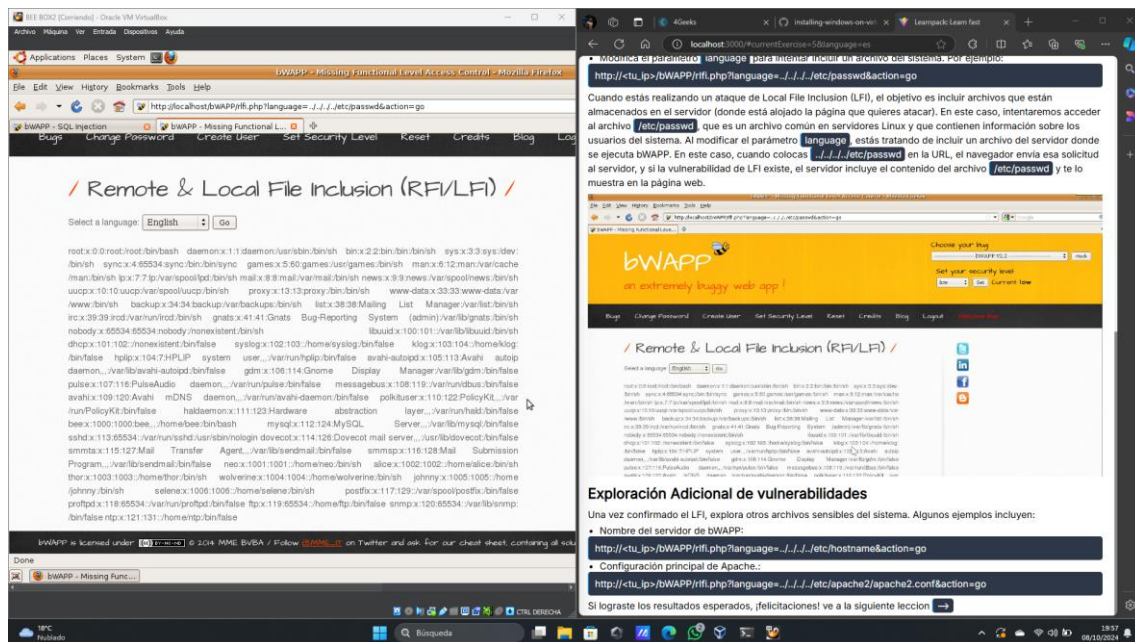
XSS con POST.



5- Security Misconfiguration - Local File Inclusion (LFI).

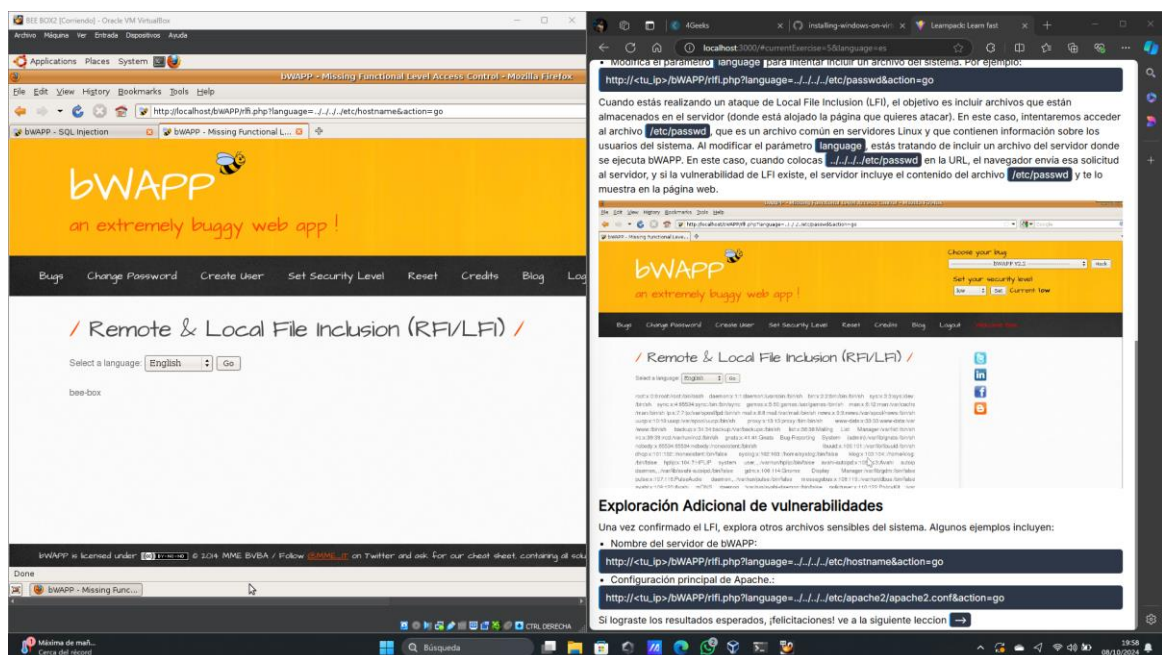
son dos vulnerabilidades que pueden comprometer la seguridad de una aplicación web.

Cuando estamos realizando un ataque de Local File Inclusion (LFI), el objetivo es incluir archivos que están almacenados en el servidor (donde está alojado la página que queremos atacar).



Exploración Adicional de vulnerabilidades.

Nombre del servidor de bWAPP.



Configuración principal de Apache.

The screenshot shows a virtual machine environment with two windows. The left window displays the Apache configuration file, which includes comments about the main configuration file and the use of the # symbol for directives. The right window shows the bWAPP application interface, which is a web application with a yellow header and a navigation menu. The application is titled "bWAPP" and "an extremely buggy web app!". It has a "Remote & Local File Inclusion (RF/LFI)" section. Below this section, there are two input fields for the URL. The first field contains the URL "http://localhost/bWAPP/rfcli.php?language=J.J.J.J/etc/hostname&action=go". The second field contains the URL "http://ctu_ip/bWAPP/rfcli.php?language=J.J.J.J/etc/apache2/apache2.conf&action=go".

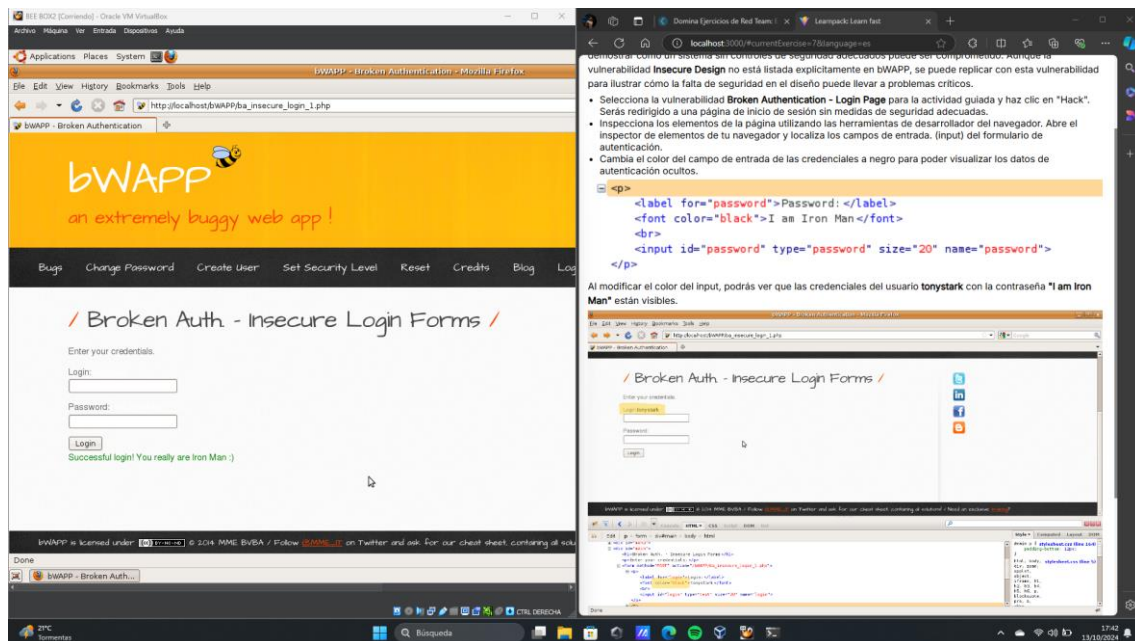
6- Server side request forgery - port scan.

Server-Side Request Forgery (SSRF) es una vulnerabilidad en la que un atacante puede manipular a un servidor vulnerable para que realice solicitudes en su nombre a otros recursos, tanto internos como externos. Este comportamiento es particularmente peligroso cuando se utiliza para realizar un escaneo de puertos en la red interna.

The screenshot shows a virtual machine environment with two windows. The left window displays the bWAPP application interface, which is a web application with a yellow header and a navigation menu. The application is titled "bWAPP" and "an extremely buggy web app!". It has a "Remote & Local File Inclusion (RF/LFI)" section. Below this section, there are two input fields for the URL. The first field contains the URL "http://localhost/bWAPP/rfcli.php?language=http://localhost/evil/1.txt&action=go". The second field contains the URL "http://localhost/evil/ssrf-1.txt". The right window shows the bWAPP application interface, which is a web application with a yellow header and a navigation menu. The application is titled "bWAPP" and "an extremely buggy web app!". It has a "Remote & Local File Inclusion (RF/LFI)" section. Below this section, there are two input fields for the URL. The first field contains the URL "http://localhost/bWAPP/rfcli.php?language=http://localhost/evil/1.txt&action=go". The second field contains the URL "http://localhost/evil/ssrf-1.txt".

7- Insecure Design - Login Page.

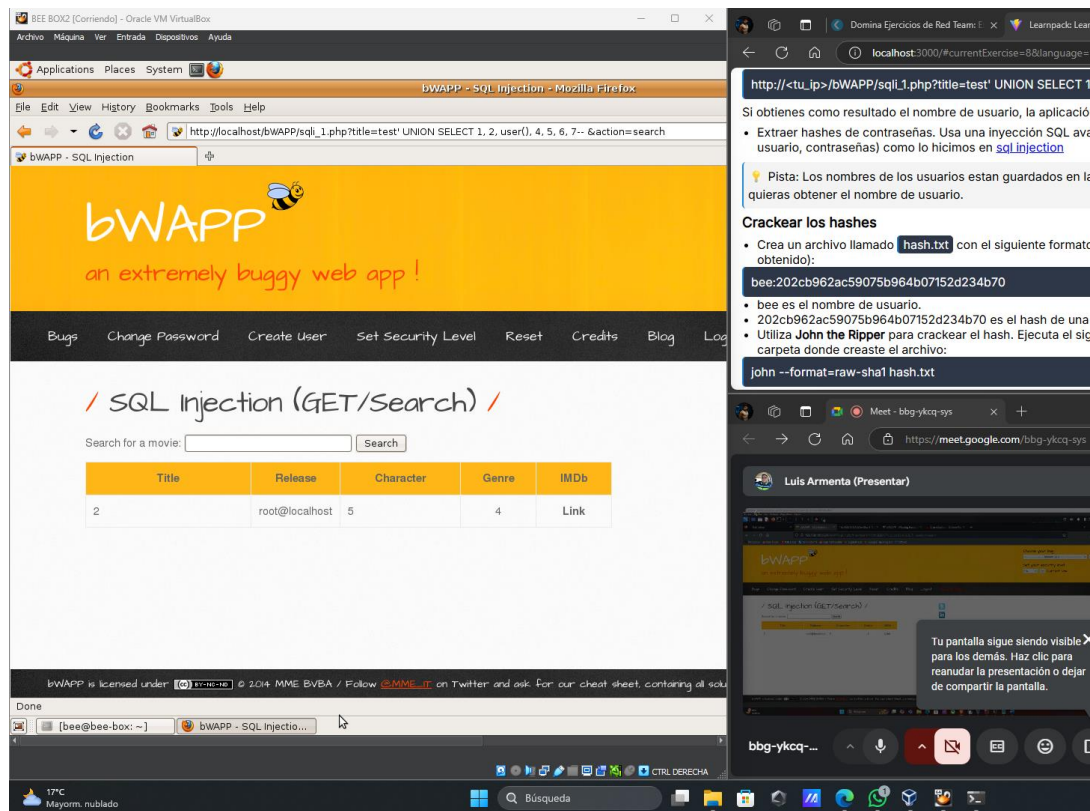
Insecure Design en una **página de inicio de sesión** (Login Page) se refiere a fallos o debilidades en el diseño arquitectónico de la funcionalidad de inicio de sesión que la hacen vulnerable a ataques. En lugar de ser simplemente errores de implementación (como el uso de malas prácticas de codificación), el problema aquí radica en cómo se conceptualiza o estructura el proceso de autenticación desde el principio.



8- Fallos de criptografía - Hashing Débil de Contraseñas.

Una implementación débil de hashing de contraseñas representa una vulnerabilidad crítica de **fallos criptográficos**, ya que, en lugar de almacenar las contraseñas en texto plano, estas deben ser hasheadas (convertidas a un valor no reversible mediante un algoritmo criptográfico) antes de ser almacenadas en una base de datos.

Obtención de hashes de contraseñas.



Si obtienes como resultado el nombre de usuario, la aplicación

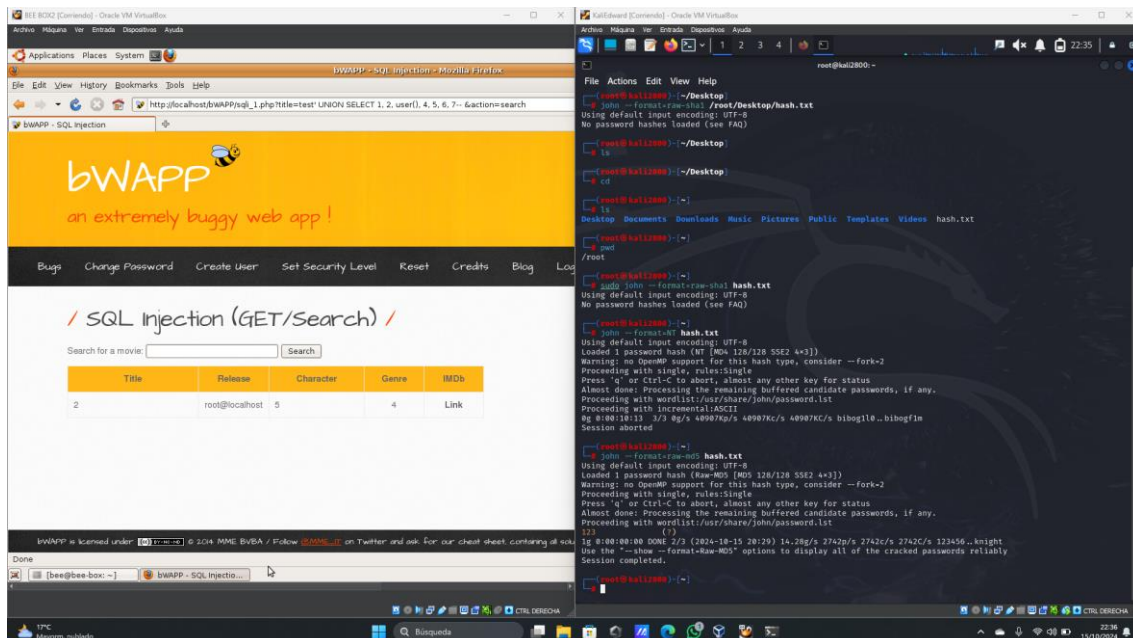
- Extraer hashes de contraseñas. Usa una inyección SQL avanzada (usuario, contraseñas) como lo hicimos en [sql injection](#)

Pista: Los nombres de los usuarios están guardados en la base de datos. Quieras obtener el nombre de usuario.

Crackear los hashes

- Crea un archivo llamado **hash.txt** con el siguiente formato obtenido):
`bee:202cb962ac59075b964b07152d234b70`
- bee es el nombre de usuario.
- 202cb962ac59075b964b07152d234b70 es el hash de una contraseña.
- Utiliza **John the Ripper** para crackear el hash. Ejecuta el siguiente comando en la carpeta donde creaste el archivo:
`john --format=raw-sha1 hash.txt`

Crackear los hashes



```
root@kali2000:~# john --format=raw-sha1 /root/Desktop/hash.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)

root@kali2000:~# john --format=raw-sha1 /root/Desktop/hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (NT [MD4 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASC2I
ig 0:00:10:13 3/3 8g/s 48907K/s 48907K/s b10g10...b10g10
Session aborted

root@kali2000:~# john --format=raw-sha1 /root/Desktop/hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
Proceeding with incremental:ASC2I
ig 0:00:00:00 DONE 2/3 (2024-10-15 20:29) 14.28g/s 2742p/s 2742C/s 123456...night
Use the --show --format=raw-MD5 options to display all of the cracked passwords reliably
Session completed.

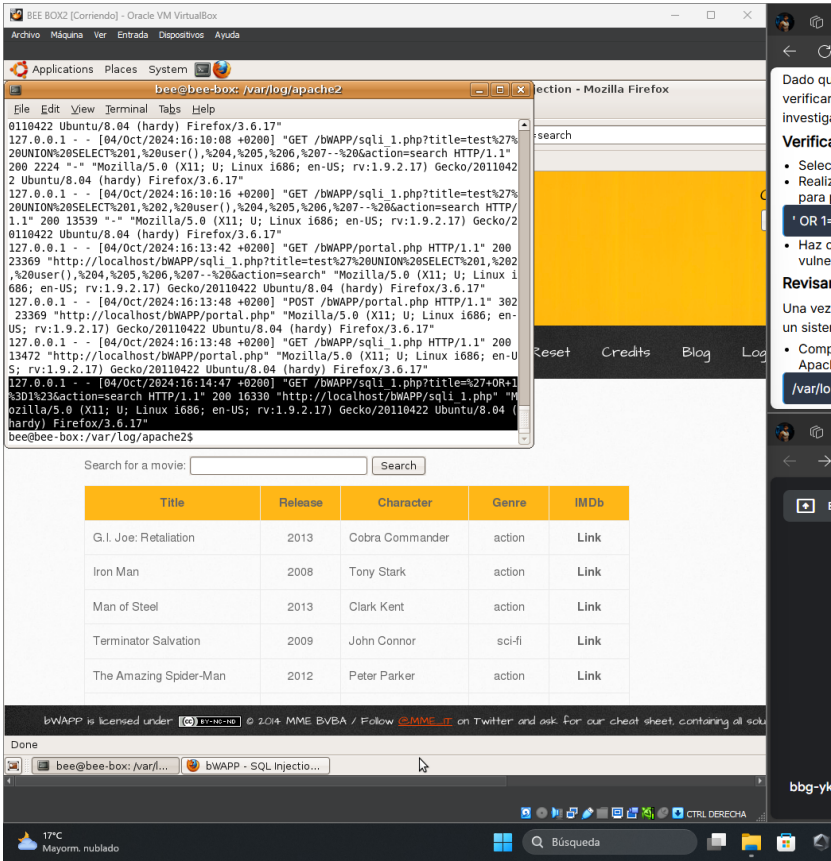
root@kali2000:~#
```

9- Security Logging and Monitoring Failures.

Security Logging and Monitoring Failures se refiere a la falta de un adecuado registro y monitoreo de eventos críticos en una aplicación. Cuando una aplicación no

registra actividades importantes, como intentos de ataque, o no monitorea estos registros de manera efectiva, los atacantes pueden pasar desapercibidos. Esto aumenta el riesgo de que un ataque comprometa la seguridad sin ser detectado.

Verificación de Intentos de SQL Injection en bWAPP.



10- vulnerable and outdated components.

También conocido como **Stored XSS**, es una vulnerabilidad de seguridad en aplicaciones web que ocurre cuando un atacante inyecta código malicioso (normalmente JavaScript) en un sitio web, y ese código se almacena permanentemente en el servidor. Posteriormente, cuando los usuarios legítimos acceden a la página web, el código malicioso se ejecuta en sus navegadores, sin que el usuario tenga que interactuar directamente con el ataque, como suele ser necesario en otros tipos de XSS (por ejemplo, XSS reflejado).

