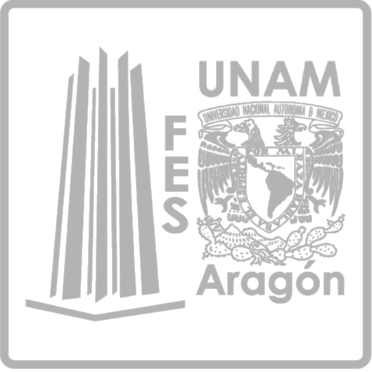
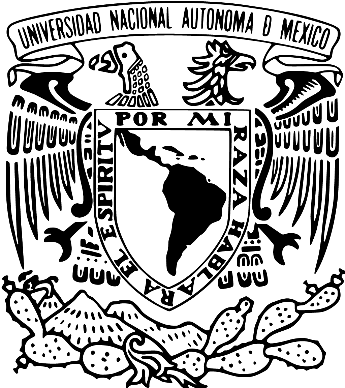
…



Texto

Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamente



# **Índice**

[Índice 2](#_Toc135871322)

[Instrucciones 3](#_Toc135871323)

[¿Qué se hizo? 11](#_Toc135871324)

[¿Cómo se resuelve? 12](#_Toc135871325)

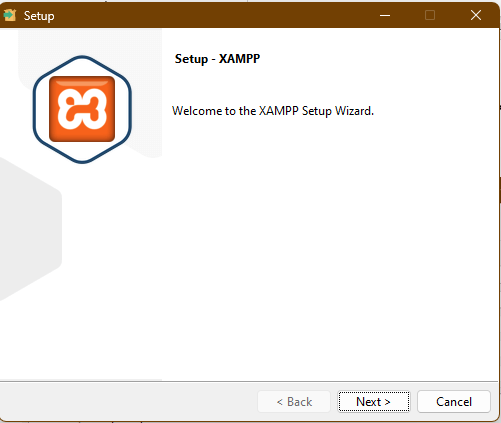
[¿Cómo se puede mitigar ese problema? 14](#_Toc135871326)

[Conclusiones 15](#_Toc135871327)

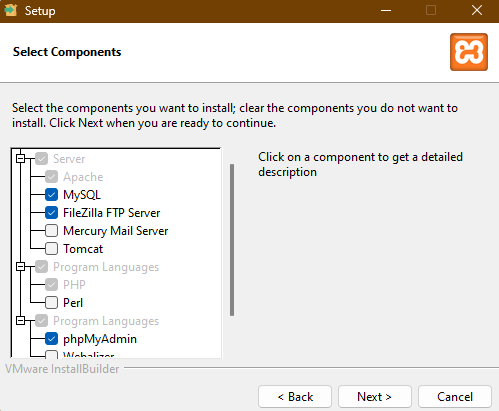
# **Instrucciones**

Nota: Únicamente para sistema operativo de Windows

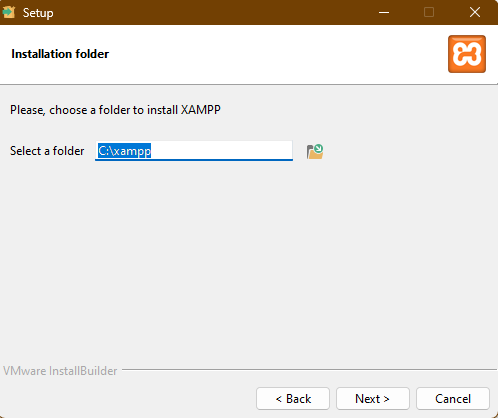
1. Clonar el repositorio <https://github.com/quintanaesc/ProyectoCokisFB.git>
2. Descargar xampp de <https://www.apachefriends.org/es/download.html>
3. Una vez descargado, comenzar la instalación.



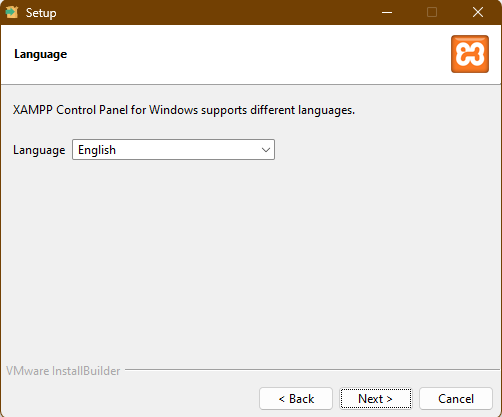
1. Seleccionar sólo lo que vamos a ocupar.



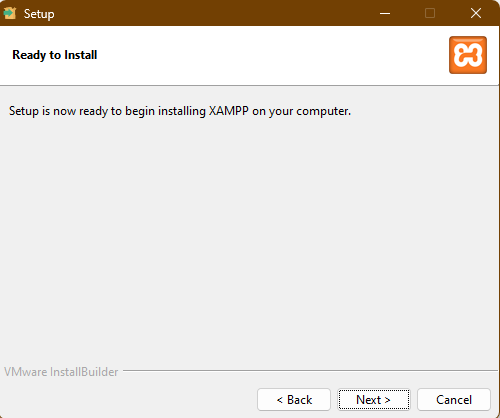
1. Seleccionar el lugar de instalación.



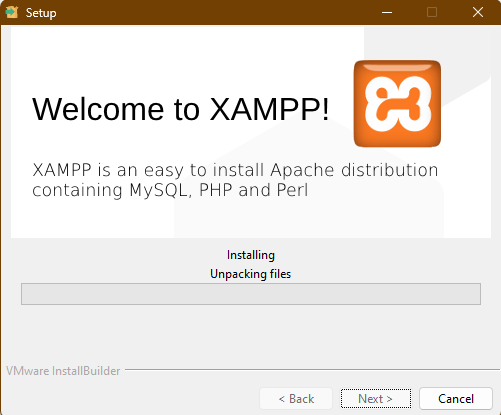
1. Seleccionar el idioma.



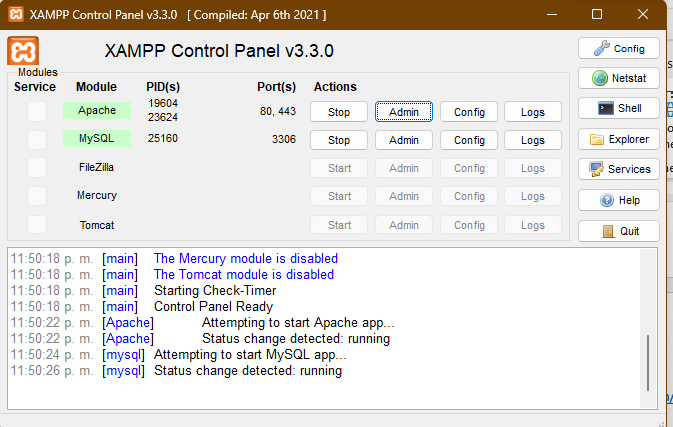
1. Una vez acabada la configuración, comenzar con la instalación.



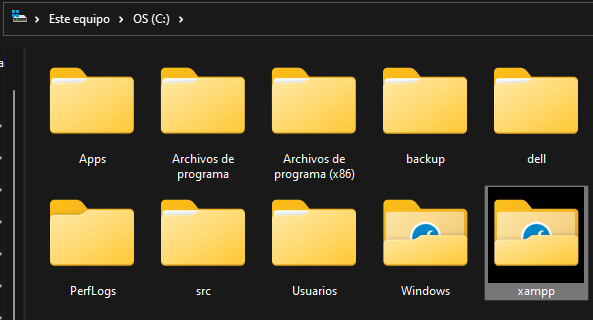
1. Esperar a que acabe la instalación.



1. Si hay algún problema en la instalación o no corre correctamente, se puede usar el siguiente vídeo de guía: <https://www.youtube.com/watch?v=xXkKeojOb1A&t=231s>
2. Una vez instalado, podemos comenzar nuestra configuración para abrir el servidor.



1. Ubicamos la carpeta donde se instaló Xampp



1. Buscamos la carpeta llamada “htdocs”

Interfaz de usuario gráfica, Aplicación

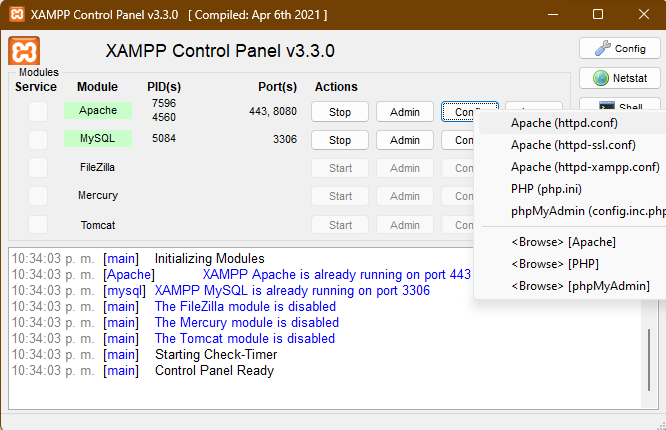
Descripción generada automáticamente

1. Copiamos los archivos que descargamos del repositorio y los pegamos en esa carpeta:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Ya teniendo los archivos ahí, nos dirigimos a Xampp y apretamos el botón “Config” de “Apache”.



1. Presionamos la primera opción (Apache (httpd.conf))
2. Nos va a abrir un bloc de notas, en el cual con el buscador (Ctrl + b o Ctrl + f), ubicaremos la palabra “htdocs”.

Captura de pantalla de un celular

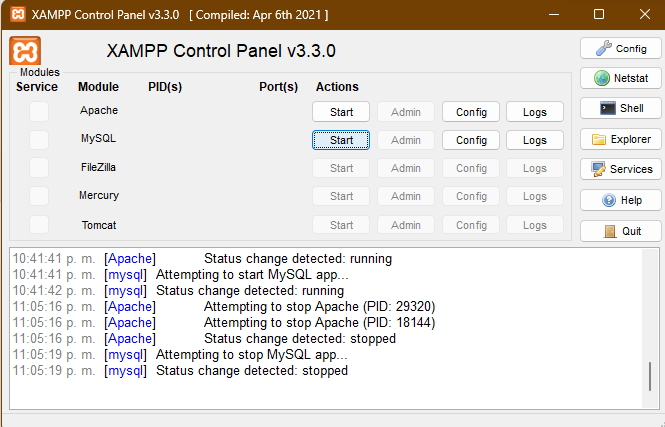
Descripción generada automáticamente

1. Le agregamos el nombre de nuestro directorio y guardamos.

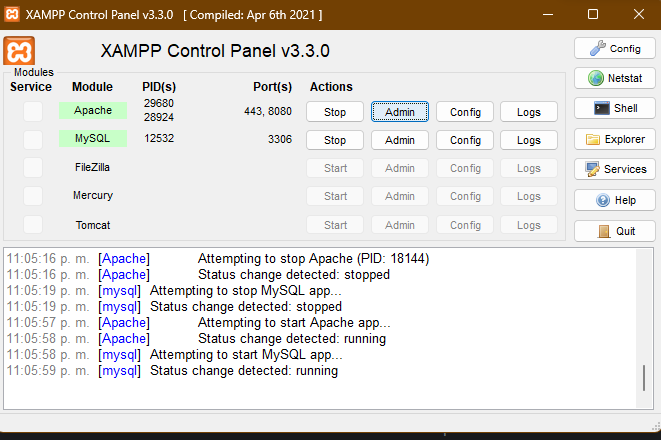
Texto

Descripción generada automáticamente

1. Reiniciamos Apache y MySQL.



1. Y ya tendremos listos nuestros servicios. Podemos entrar de varias maneras:
2. Dando click en “Admin” de “Apache”

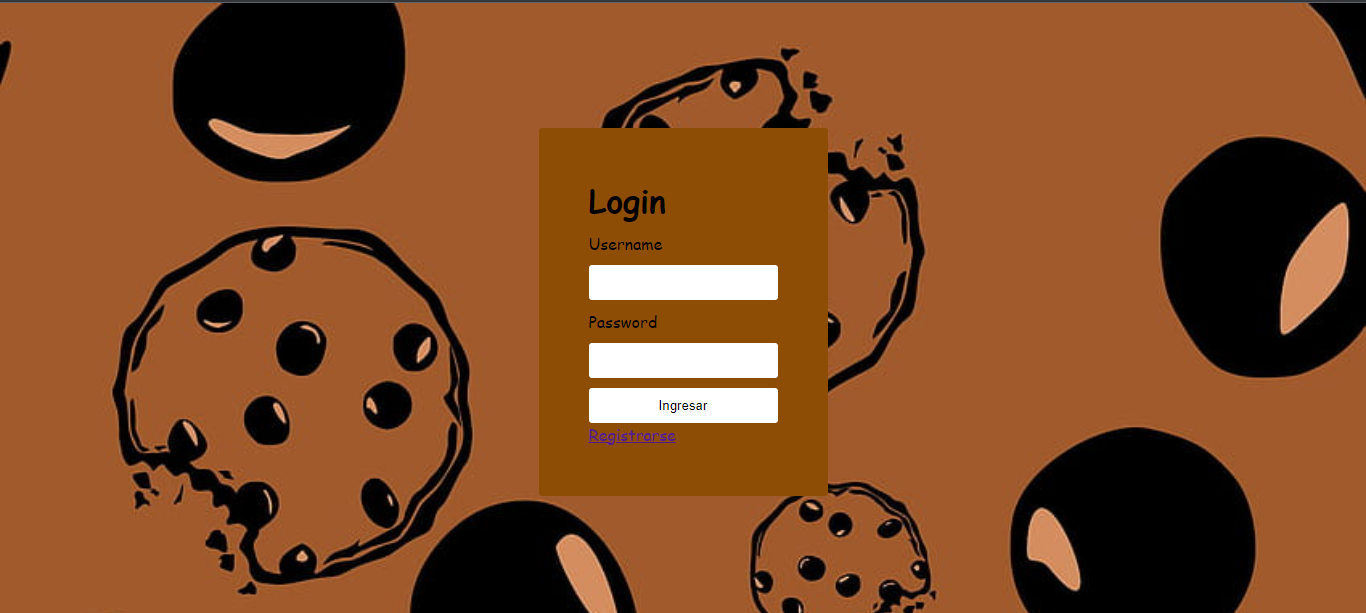


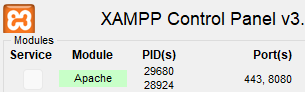
Y nos va a abrir el localhost

Interfaz de usuario gráfica, Texto, Correo electrónico

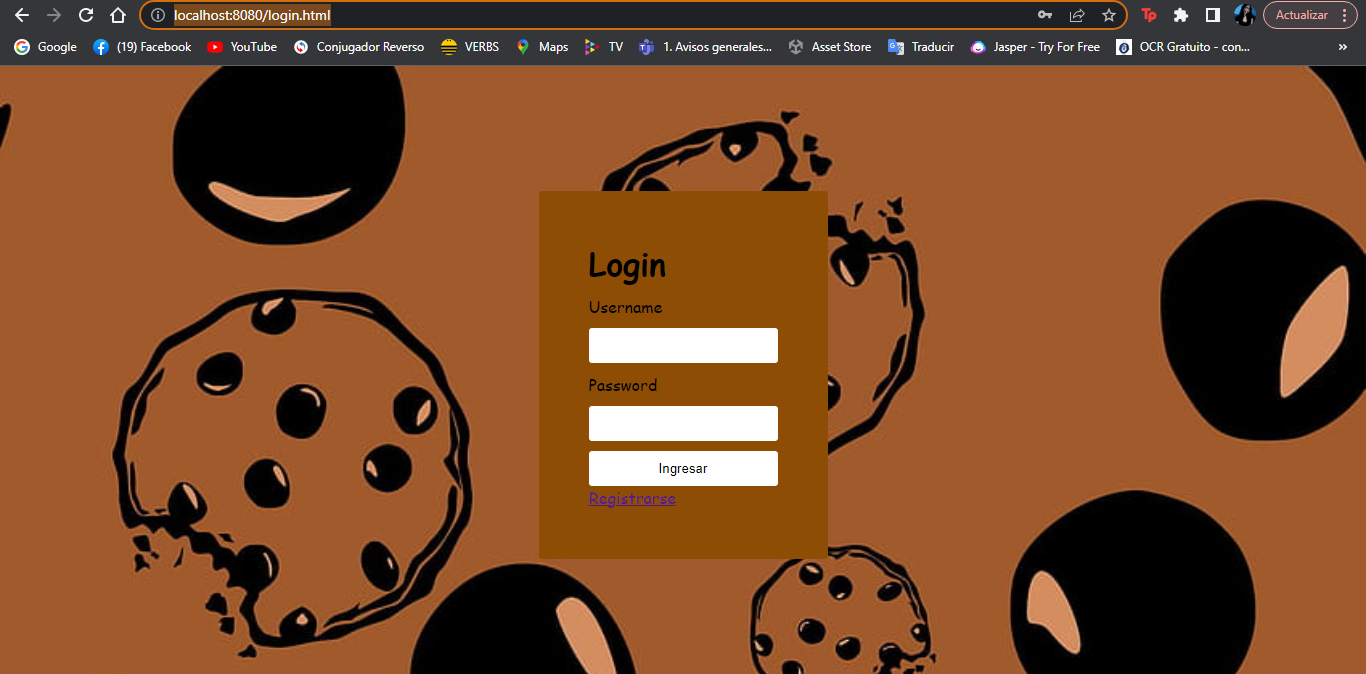
Descripción generada automáticamente

Ya sólo seleccionamos el html de login y ya tenemos acceso a la página de ingreso.



1. Nos vamos directamente al navegador de su preferencia y colocamos en el buscador “localhost”, el cual nos va a volver a dirigir a la página que muestra los archivos, seleccionamos el html de login y accedemos.
2. En el mismo navegador colocamos en el buscador “localhost” acompañado del puerto, el cual lo va a indicar Xampp: 

En éste caso, colocamos “localhost:8080” para que nos rediriga directamente a la página de login.



# **Imagen en blanco y negro Descripción generada automáticamente con confianza media¿Qué se hizo?**

A lo largo del proyecto, se llevaron a cabo una serie de actividades y tareas que contribuyeron a su desarrollo. A continuación, se describe de manera general lo que se hizo:

En cuanto a la instalación del servidor, se realizó una investigación sobre la instalación de un servidor local en Windows. Después de evaluar diferentes opciones, se decidió utilizar XAMPP como distribución de Apache para administrar el servidor. Se procedió a instalar y configurar XAMPP, y se creó un repositorio en GitHub para utilizarlo como sistema de control de versiones para el servidor. Además, se instaló el repositorio localmente dentro de la carpeta "htdocs".

En relación con el diseño de la página web, se obtuvo una plantilla básica de inicio de sesión y una página web simple copiando la estructura y los archivos de estilo. Se implementó el uso de Bootstrap para estilizar las páginas. Se realizaron modificaciones en los estilos de la página de inicio de sesión y la página de correos para personalizarlas según las necesidades del proyecto. Además, se diseñó y creó una página de registro.

Se procedió al diseño e instalación de una base de datos. Se diseñaron las tablas "empleado" y "correos", y se generaron los scripts SQL para crear la base de datos "email\_servicio" y para insertar datos en las tablas correspondientes. Estos scripts SQL se ejecutaron en el administrador de base de datos phpMyAdmin del servidor para crear y probar la base de datos. También se verificaron las credenciales de usuario.

Se diseñaron los archivos PHP necesarios para el inicio de sesión y el registro de usuarios. Se generó la conexión con la base de datos, cuyo código se reutilizó para el resto de los archivos PHP del sistema. Se diseñaron las consultas SQL correspondientes y se investigó sobre el inicio de sesión mediante cookies, lo que llevó a la generación de la cookie de inicio de sesión.

Se diseñaron los diferentes sitios dentro de la aplicación web. Se estableció una estructura común para los servicios y se determinó la función de cada página, como "correos", "enviar" y "cuenta". Para cada sitio, se diseñó una estructura específica y una sección "about" utilizando HTML. Por ejemplo, en la página de "correos" se reservó un espacio para una tabla futura que se cargaría mediante una función en JavaScript. En la página "enviar" se creó un formulario para enviar correos, que se insertaban en la tabla correspondiente de la base de datos. En la página "cuenta" se diseñó un formulario para el cambio de datos de la cuenta, que modificaba el registro en la tabla "empleado" de la base de datos.

Se decidió utilizar la cookie de sesión para reflejar el usuario logueado mediante un letrero en la sección de cabecera de las páginas de "correos" y "enviar". Para lograr esto, se generó el archivo "usuario.php" que se encargaba de obtener y devolver el usuario logueado. Se diseñó una función JavaScript para obtener el resultado de "usuario.php" y mostrarlo en las páginas correspondientes.

Finalmente, se diseñó la funcionalidad de las secciones "about" de las páginas. En el caso de la página "correos", se generó el archivo PHP correspondiente para realizar consultas a la base de datos, obteniendo información sobre los correos.

# **¿Cómo se resuelve?**

El proyecto en cuestión se presenta como un reto CTF, en el cual los participantes se registran en una página web y se convierten en usuarios dentro de esta plataforma simulada. Una vez dentro, el objetivo del reto consiste en vulnerar los datos de otros usuarios registrados en la página para obtener su información personal a través de la manipulación o configuración de las cookies. Es importante destacar que todos estos usuarios e información son ficticios, creados por nosotros mismos para el propósito del proyecto.

La resolución del reto implica modificar el valor de las cookies de manera iterativa, utilizando el método de fuerza bruta, hasta encontrar al usuario objetivo que proporcionará una "bandera" o indicador de éxito en el reto.

Dado que la técnica de fuerza bruta puede llevar mucho tiempo, especialmente cuando se enfrenta a un gran número de registros, se decidió implementar un script automatizado para agilizar este proceso. Una vez que el participante ha iniciado sesión con su propio usuario en la página, puede copiar y pegar el script en la consola del navegador. A partir de ese momento, el script se encargará de realizar ciclos repetitivos con el fin de probar diferentes valores de cookies hasta que se encuentre el usuario objetivo y se obtenga la bandera que indica el éxito del reto.

El uso de este enfoque automatizado mediante el script proporciona una forma eficiente de resolver el reto al reducir el tiempo y los recursos necesarios para llevar a cabo el proceso de fuerza bruta manualmente. Al delegar esta tarea al script, los participantes pueden centrarse en otras actividades relacionadas con el reto.

Es importante tener en cuenta que este proyecto se desarrolla en un entorno controlado y seguro, utilizando únicamente usuarios y datos ficticios creados por los participantes del proyecto. El objetivo principal es permitir a los participantes adquirir habilidades y conocimientos relacionados con la seguridad informática, comprendiendo las vulnerabilidades potenciales y aplicando técnicas para fortalecer la protección de la información.

En resumen, la resolución de este proyecto de ciberseguridad implica el uso de un script automatizado para realizar un proceso de fuerza bruta en la modificación de las cookies, con el fin de encontrar al usuario objetivo y obtener la bandera con éxito.

**¿Qué tecnologías se usaron?**

En la realización de nuestro proyecto de seguridad informática, utilizamos diversas tecnologías para asegurar el funcionamiento y la eficiencia del mismo. A continuación, se detallan las principales tecnologías que implementamos:

1. XAMPP para Windows: Utilizamos el paquete de software XAMPP, que incluye el servidor web Apache, el lenguaje de programación PHP, el sistema de gestión de bases de datos MySQL y otros componentes necesarios para la creación de aplicaciones web.
2. HTML, CSS y JavaScript: Estos son los lenguajes fundamentales para el desarrollo de sitios web. Utilizamos HTML para la estructura y el contenido de las páginas, CSS para el diseño y la presentación visual, y JavaScript para agregar interactividad y funcionalidad a la página.
3. Git y GitHub: Utilizamos Git como un sistema de control de versiones para gestionar los cambios en el código de nuestro proyecto además de la documentación. GitHub, como plataforma basada en la nube, nos proporciona un repositorio remoto para almacenar y colaborar en el desarrollo del proyecto.
4. Visual Studio: Utilizamos Visual Studio como nuestro IDE principal. Proporciona herramientas y características que nos ayudan a escribir, depurar y mantener el código de manera eficiente.
5. Opera GX y su consola: Utilizamos el navegador Opera GX por sus características de privacidad y seguridad avanzadas. Además, aprovechamos su consola de desarrollo para realizar pruebas y depurar el código de la aplicación web.
6. Entorno Windows: El sistema operativo Windows es utilizado como el entorno principal en el cual desarrollamos nuestro proyecto de seguridad informática. Aprovechamos las características de compatibilidad de Windows para asegurar la estabilidad y el rendimiento del proyecto.

Estas tecnologías nos permitieron construir el proyecto de seguridad informática, brindando la base necesaria para poder realizar nuestro desafío de CTF.

# **¿Cómo se puede mitigar ese problema?**

Medidas que podemos tomar para mitigar este problema y los riesgos relacionados con la manipulación no autorizada de cookies y el acceso no autorizado a los datos:

1. Utilizar el atributo HttpOnly: Asegurarnos de configurar todas las cookies con el atributo HttpOnly. Esto evita que sean accesibles mediante scripts del lado del cliente, reduciendo así el riesgo de ataques de secuestro de cookies.

En PHP, al crear una cookie, puedes usar la función setcookie y establecer el parámetro httponly en true para habilitar el atributo HttpOnly.

1. Configurar tiempos de expiración razonables para las cookies. Evita que las cookies permanezcan activas durante períodos prolongados, lo que reduce la ventana de oportunidad para posibles ataques.
2. Validar los datos de las cookies: Antes de utilizar los datos almacenados en las cookies, asegurémonos de validar y limpiar cuidadosamente la información. Esto ayuda a prevenir inyecciones de código malicioso o manipulación no autorizada de los datos.
3. Realiza actualizaciones periódicas de tus aplicaciones y servidores, incluyendo los parches de seguridad más recientes.
4. Realiza pruebas regulares de seguridad en tu página para identificar posibles vulnerabilidades. Esto te permite corregirlas antes de que sean aprovechadas por atacantes y fortalecer la seguridad.
5. Asegurarse de que los datos almacenados en la base de datos estén encriptados de manera adecuada. Esto ayuda a proteger la información confidencial en caso de que se produzca una brecha de seguridad.

Al implementar estas medidas, estaremos tomando acciones concretas para mitigar los riesgos asociados con la manipulación no autorizada de cookies y el acceso no autorizado a los datos de tus usuarios.

# **Imagen en blanco y negro Descripción generada automáticamente con confianza mediaConclusiones y experiencias**

En conclusión, la realización del CTF es una excelente manera de reforzar y poner a prueba nuestras habilidades y conocimientos en la asignatura, y crear un desafío propio agrega un nivel adicional a nuestra creatividad y personalización.

En este caso, el desafío se centró en automatizar cookies por fuerza para encontrar el usuario.

Bajo el contexto de un CTF donde realizamos pruebas en un entorno controlado, el uso de fuerza bruta puede ser una técnica que nos puede ayudar a resolver desafíos y aprender sobre las vulnerabilidades que existen con medidas adecuadas de seguridad.

Es importante destacar que el objetivo principal de nuestro CTF es promover el aprendizaje y mejorar nuestras habilidades en esta área. Al diseñar y realizar este desafío propio, no comprometemos la seguridad o privacidad de otros sistemas o personas fuera de nuestro entorno controlado del CTF.

En general, nuestra experiencia fue

* Paola: Mi experiencia a lo largo del desarrollo de este CTF fue un constante aprendizaje, puesto que en mi caso todo se me hizo complicado y al mismo tiempo todo se me hizo sumamente interesante. Me gustó ver cómo se seleccionaron, implementaron y fusionaron las herramientas para la creación del reto, pero además de ello lo que más me intereso fue la manera en que se trató la vulnerabilidad, sobre todo, porque en algunos ejercicios que hacíamos de Pico CTF, trabajamos con las cookies, sin embargo, fue diferente solo resolverlo a hacerlo.
* Ricardo: Durante el desarrollo del proyecto me enfrenté a varios desafíos que con ayuda de mis compañeros pude resolver, por ejemplo, aplicar nuevas tecnologías como XAMPP, PHP, MySQL, HTML, CSS y JavaScript en el desarrollo de un CTF de ciberseguridad. Aprendí sobre la instalación de servidores, el diseño de servicios web, la manipulación de cookies y la protección de datos. Fue una experiencia muy interesante, aunque bastante desafiante ya que desconocía mucho del tema, me ayudó a fortalecer mis habilidades y conocimientos en esta área.
* Diana: Realizar este proyecto de ciberseguridad ha sido un desafío emocionante y algo difícil. Superé varios retos, como el de enfrentarme a tecnologías que jamás había usado, como XAMPP, apache, php, entre otras tecnologías, apoyada del conocimiento de mis compañeros y de tutoriales e investigación. También de retos técnicos al aplicar técnicas de ciberseguridad, como la manipulación de cookies, la fuerza bruta y el desarrollo de un script para automatizar el proceso. Además, me permitió reflexionar sobre cuestiones éticas relacionadas con la privacidad y la protección de datos.

Este proyecto me brindó experiencia para trabajar en equipo, y colaborar con mis compañeros fue enriquecedor, ya que pudimos compartir ideas y encontrar soluciones efectivas.

* Roberto: