

# Informe Laboratorio 3

## Sección 2

Isidora Bravo Ortiz  
e-mail: isidora.bravo2@mail.udp.cl

Octubre de 2025

## Índice

<b>1. Descripción de actividades</b>	<b>2</b>
<b>2. Desarrollo de actividades según criterio de rúbrica</b>	<b>3</b>
2.1. Identifica el algoritmo de hash utilizado al momento de registrarse en el sitio	3
2.2. Identifica el algoritmo de hash utilizado al momento de iniciar sesión . . . . .	3
2.3. Genera el hash de la contraseña desde la consola del navegador . . . . .	4
2.4. Intercepta el tráfico login con BurpSuite . . . . .	6
2.5. Realiza el intento de login por medio del hash . . . . .	7
2.6. Identifica las políticas de privacidad o seguridad . . . . .	8
2.7. Comente 4 conclusiones sobre la seguridad del sitio escogido . . . . .	9

## 1. Descripción de actividades

Su objetivo será auditar la implementación de algoritmos hash aplicados a contraseñas en páginas web desde el lado del cliente, así como evaluar la efectividad de estas medidas contra ataques de tipo Pass the Hash (PtH). Para llevar a cabo esta auditoría, deberá registrarse en un sitio web y crear una cuenta, ingresando una contraseña específica para realizar las pruebas.

Al concluir la tarea, es importante que modifique su contraseña por una diferente para garantizar su seguridad.

Dado que la cantidad de sitios chilenos que utilizan hash es limitada, se permite realizar esta tarea en cualquier sitio web a nivel mundial. En este sentido, realice las siguientes actividades:

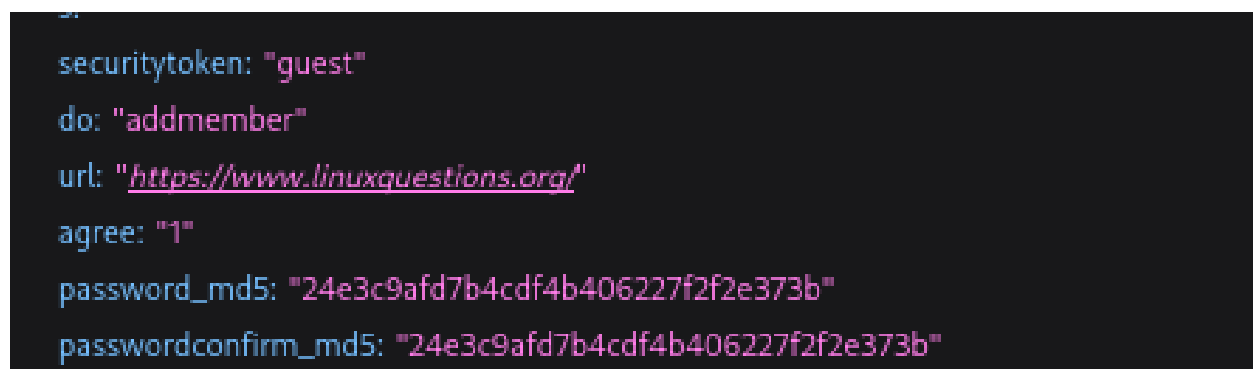
- Identificación del algoritmo de hash utilizado para las contraseñas al momento del registro en el sitio.
- Identificación del algoritmo de hash utilizado para las contraseñas al momento de iniciar sesión.
- Generación del hash de la contraseña desde la consola del navegador, partiendo de la contraseña en texto plano.
- Interceptación del tráfico de login utilizando BurpSuite desde su equipo.
- Realización de un intento de login modificando la contraseña por una incorrecta haciendo uso del hash obtenido en el punto anterior. Puede interceptar el tráfico y modificar el hash por el correcto o hacer uso del servicio repeater de BurpSuite.
- Descripción de las políticas de privacidad o seguridad relacionadas con las contraseñas, incluyendo un enlace a las mismas.
- Cuatro conclusiones sobre la seguridad o vulnerabilidad de la implementación observada.

## 2. Desarrollo de actividades según criterio de rúbrica

### 2.1. Identifica el algoritmo de hash utilizado al momento de registrarse en el sitio

Para dar inicio a esta actividad de laboratorio, procedemos a ingresar a la página **www.linuxquestions.org** donde nos registraremos e identificaremos el hash que se utiliza al momento de crear una nueva cuenta.

En esta oportunidad, nos dirigimos a inspeccionar elementos y notamos que el hash utilizado en esta página corresponde a MD5 como se puede ver en la siguiente captura:



```
securitytoken: "guest"
do: "addmember"
url: "https://www.linuxquestions.org/"
agree: "1"
password_md5: "24e3c9afd7b4cdf4b406227f2f2e373b"
passwordconfirm_md5: "24e3c9afd7b4cdf4b406227f2f2e373b"
```

Figura 1: Identificación de Hash en el registro

### 2.2. Identifica el algoritmo de hash utilizado al momento de iniciar sesión

Luego, con las credenciales creadas anteriormente, procederemos a ingresar a nuestra cuenta para poder identificar el algoritmo de hash que se utiliza al momento de iniciar sesión. Una vez ingresadas las credenciales correspondientes, encontramos el siguiente hash al inspeccionar:

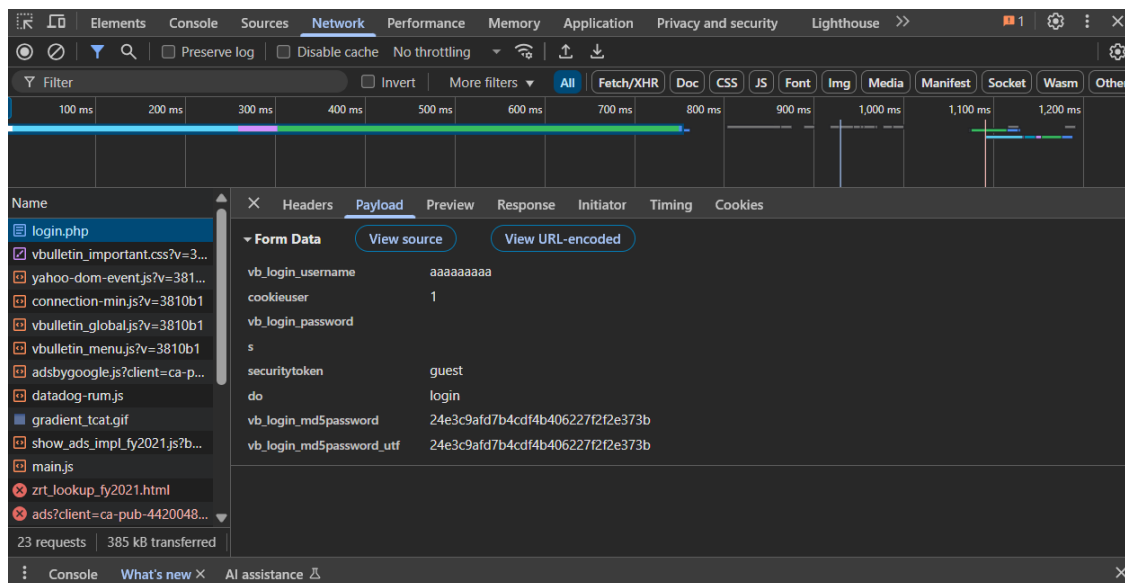


Figura 2: Hash encontrado al iniciar sesión

### 2.3. Genera el hash de la contraseña desde la consola del navegador

Para el siguiente punto, se inspeccionó el formulario de inicio de sesión y se observó que en el atributo **onsubmit** se invocaba una función de hashing del lado del cliente (**md5hash(...)**). Esto indicaba que la contraseña era transformada antes de ser enviada al servidor como se ve en la siguiente figura:

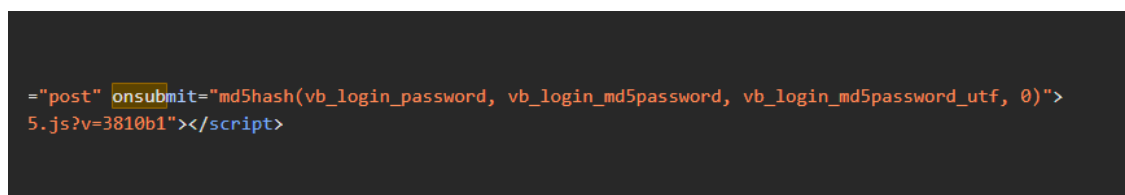


Figura 3: HTML del formulario

Posteriormente, se accedió a las herramientas de desarrollo (DevTools) y, en la pestaña Sources, se localizó el archivo JavaScript responsable de la implementación de MD5: **vbulletin\_md5.js**. En este archivo se identificó la función **hex\_md5**, lo cual confirmó la presencia del algoritmo MD5 en el código del cliente como se demuestra en la siguiente figura:

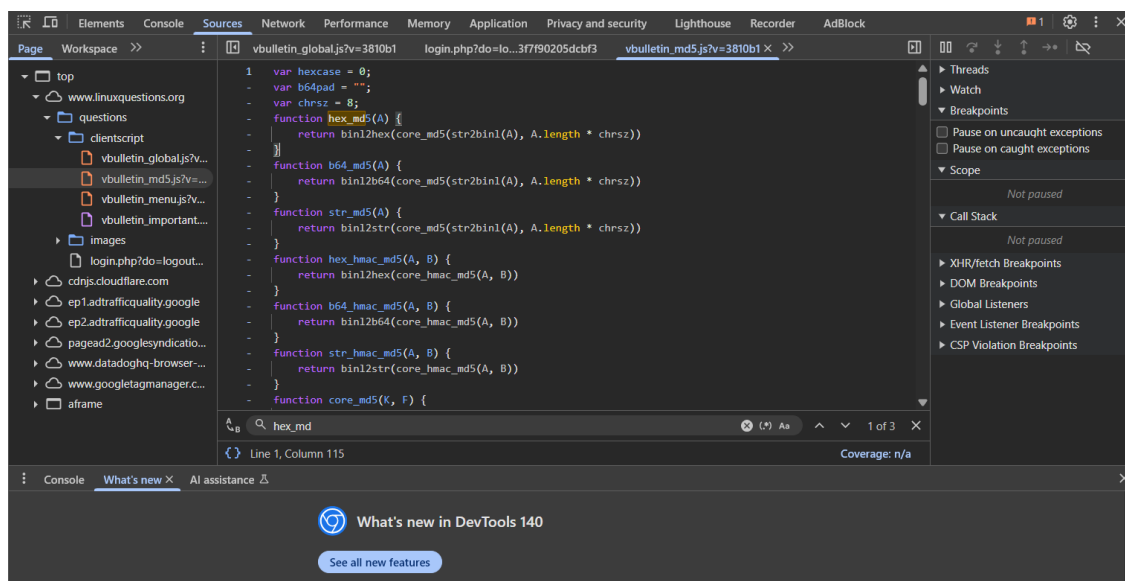


Figura 4: Implementación MD5 en cliente

Finalmente, se reprodujo el funcionamiento del hash ejecutando en la consola del navegador la instrucción `hex_md5("holagente")`, la cual devolvió el valor hexadecimal **24e3c9afd7b4cdf4b406227f2f2e373b**. Esto evidenció que el hash puede generarse localmente utilizando la misma rutina que emplea el sitio web como se puede apreciar a continuación:

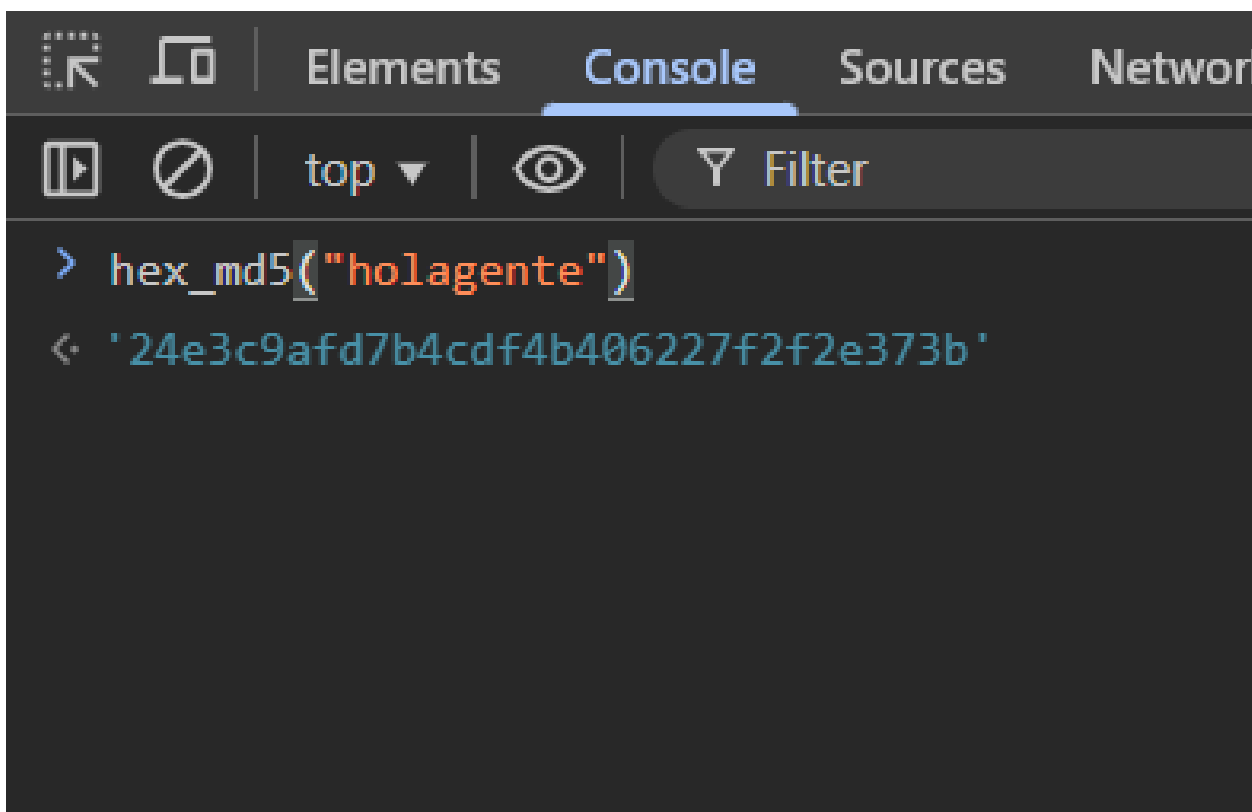


Figura 5: Generación del hash desde la consola

## 2.4. Intercepta el tráfico login con BurpSuite

A continuación, se interceptará el tráfico de la página con la herramienta BurpSuite, para esto se abrió BurpSuite y, empleando el navegador integrado, se navegó hasta la página objetivo. Tras iniciar sesión con las credenciales creadas anteriormente, se activó Proxy y luego Intercept: On, con lo que las peticiones y respuestas quedaron detenidas en Burp para ser visualizadas como se ve a continuación:

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

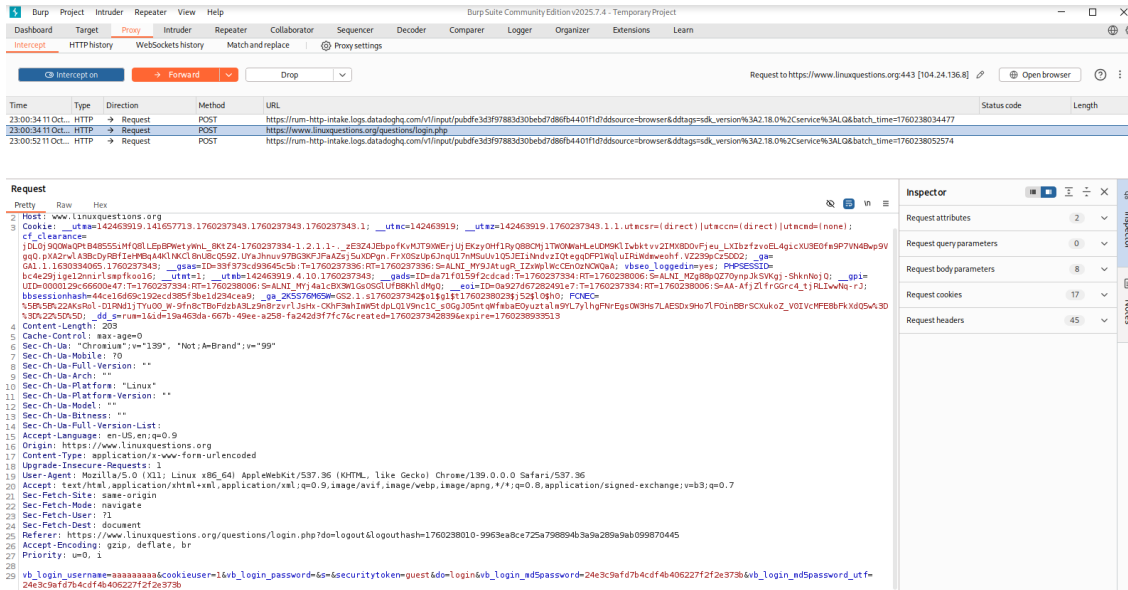


Figura 6: Tráfico interceptado con BurpSuite

### 2.5. Realiza el intento de login por medio del hash

Después de interceptar el tráfico, se intentó iniciar sesión con una contraseña incorrecta para generar la petición de autenticación fallida.

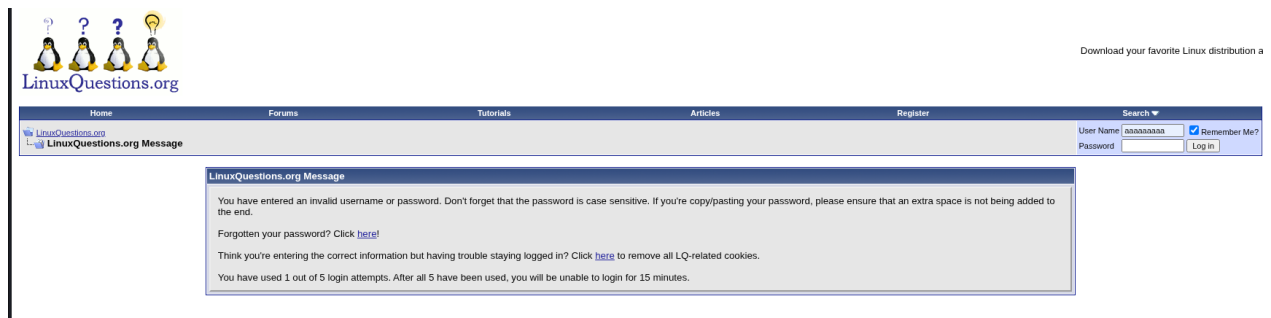


Figura 7: Ingreso contraseña incorrecta en la página

La petición interceptada se examinó en BurpSuite para localizar el campo de la contraseña (en este caso subrayado con negro) :



Figura 8: Ingreso contraseña incorrecta en BurpSuite

A continuación, ese campo se reemplazó manualmente por el hash obtenido anteriormente y se reenvió la solicitud al servidor, simulando un ataque Pass-the-Hash.



Figura 9: Ingreso manual del Hash obtenido

Se comprobó la respuesta y el comportamiento de la aplicación para verificar si el servidor aceptaba el valor modificado y creaba la sesión, al recibir una respuesta satisfactoria, se confirmó el acceso a la página desde el navegador.

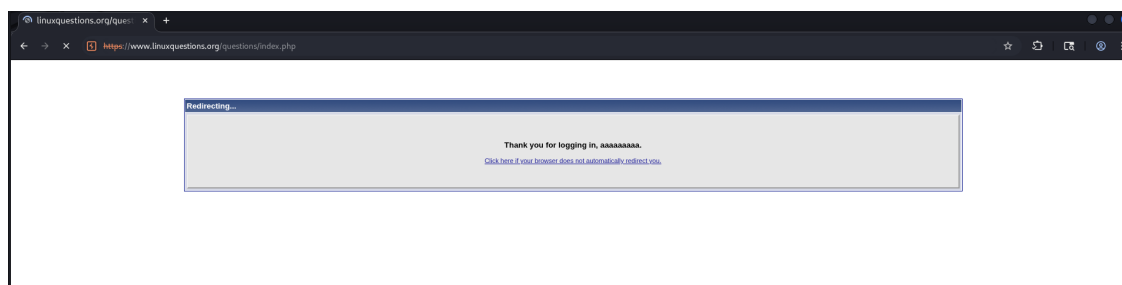


Figura 10: Ingreso exitoso a la página con el hash modificado

## 2.6. Identifica las políticas de privacidad o seguridad

El sitio **www.linuxquestions.org** establece en su política de privacidad que recopila información personal como correo electrónico, nombre de usuario, contraseña e IP, utilizada únicamente para operar y mantener el servicio. Aunque no se detallan mecanismos específicos sobre cómo se almacenan las contraseñas, la política menciona que existen medidas de seguridad para evitar la pérdida, mal uso o alteración de los datos bajo su control. Sin embargo, durante la realización del laboratorio se evidenció que las contraseñas son transformadas mediante MD5 desde el lado del cliente, lo que sugiere que el sitio no emplea algoritmos modernos ni prácticas recomendadas como el uso de “salts” o hashing del lado del servidor. Esto representa una vulnerabilidad potencial, ya que el hash generado



podría ser reutilizado en ataques del tipo Pass-the-Hash. Enlace a la política de privacidad: <https://www.linuxquestions.org/linux/privacy.html>

## 2.7. Comente 4 conclusiones sobre la seguridad del sitio escogido

Durante la realización de esta experiencia de laboratorio se dejó en evidencia el funcionamiento del proceso de autenticación del sitio **www.linuxquestions.org** y la manera en que se implementa la protección de contraseñas. A partir de los análisis realizados, se pudieron identificar diversos aspectos técnicos y de seguridad relevantes, los cuales se resumen a continuación:

1. **Uso de hashing en el cliente:** Se comprobó que el sitio aplica el algoritmo MD5 sobre la contraseña antes de enviarla al servidor. Este método evita la transmisión en texto plano, pero no proporciona una seguridad adecuada frente a ataques modernos.
2. **Debilidad del algoritmo MD5:** El uso de MD5 representa una vulnerabilidad, ya que este algoritmo es considerado obsoleto por su susceptibilidad a colisiones y ataques de fuerza bruta. Por ello, no es recomendable para la protección de contraseñas.
3. **Riesgo de ataque Pass-the-Hash (PtH):** Debido a que el hash MD5 se transmite directamente durante el proceso de inicio de sesión, un atacante que intercepte el tráfico podría reutilizar dicho valor para autenticarse sin conocer la contraseña original, exponiendo una falla de seguridad crítica.
4. **Falta de prácticas modernas de seguridad:** Aunque la política del sitio menciona medidas de seguridad generales, no se evidenció el uso de mecanismos actualizados como salts, hashing del lado del servidor o algoritmos más robustos (por ejemplo, Argon2id, bcrypt, scrypt, PBKDF2-HMAC-SHA256.), los cuales son actualmente considerados estándares para la protección de contraseñas.