

# Informe Laboratorio 3

## Sección 4

Renato Oscar Benjamin Contreras Carvajal  
e-mail: renato.contreras@mail.udp.cl

Octubre de 2024

## Índice

<b>1. Descripción de actividades</b>	<b>2</b>
<b>2. Desarrollo de actividades según criterio de rúbrica</b>	<b>2</b>
2.1. Identifica el algoritmo de hash utilizado al momento de registrarse en el sitio	2
2.2. Identifica el algoritmo de hash utilizado al momento de iniciar sesión . . . . .	5
2.3. Genera el hash de la contraseña desde la consola del navegador . . . . .	6
2.4. Intercepta el tráfico login con BurpSuite . . . . .	8
2.5. Realiza el intento de login . . . . .	8
2.6. Identifica las políticas de privacidad o seguridad . . . . .	11
2.7. Demuestra 4 conclusiones sobre la seguridad . . . . .	11

## 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.

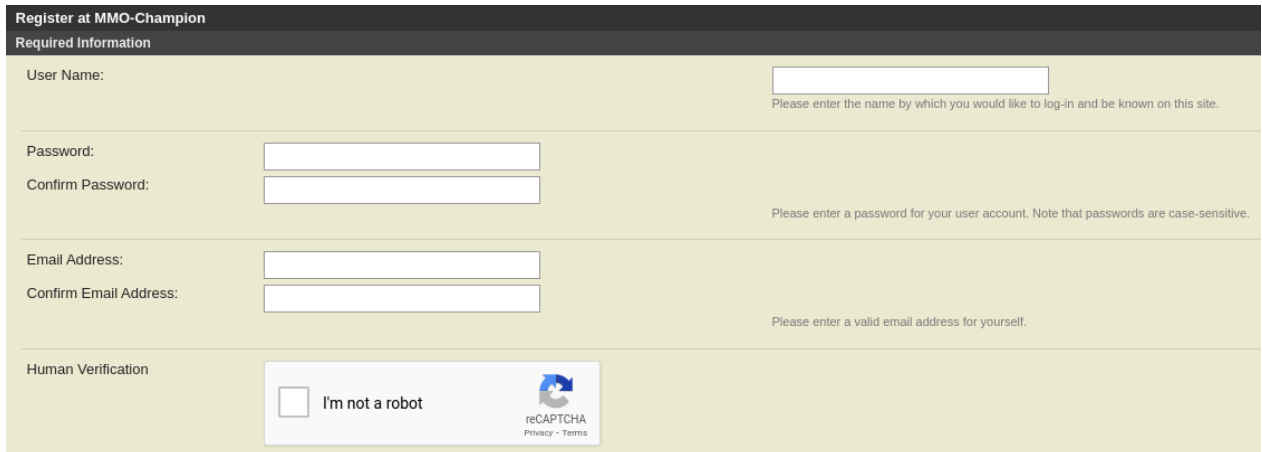
Dado que la cantidad de sitios chilenos que utilizan hash es limitada, se permite realizar esta tarea en cualquier sitio web a nivel mundial. En este sentido, realice las siguientes actividades:

- 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 una contraseña incorrecta por el hash obtenido en el punto anterior.
- 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

En este laboratorio haremos uso de la página web <https://www.mmo-champion.com/content/> en la que completaremos el formulario de registro:

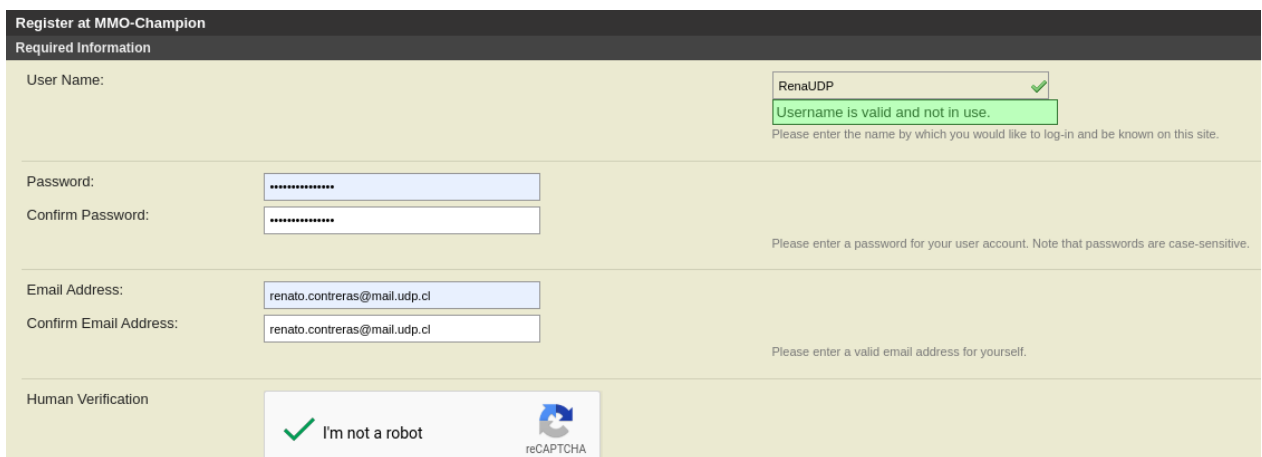


The image shows the 'Register at MMO-Champion' form with the following fields and instructions:

- User Name:** A text input field. Instruction: "Please enter the name by which you would like to log-in and be known on this site."
- Password:** A text input field.
- Confirm Password:** A text input field. Instruction: "Please enter a password for your user account. Note that passwords are case-sensitive."
- Email Address:** A text input field.
- Confirm Email Address:** A text input field. Instruction: "Please enter a valid email address for yourself."
- Human Verification:** Includes a checkbox labeled "I'm not a robot" and a reCAPTCHA logo with links for "Privacy" and "Terms".

Figura 1: Formulario de registro de MMOChampion.

Para esto se utilizará la contraseña: sMyTvQ5AB@V5Zcy.



The image shows the same registration form, but now filled out with the following data:

- User Name:** "RenaUDP". A green checkmark is visible next to the input field. A green message box below the field states: "Username is valid and not in use." Instruction: "Please enter the name by which you would like to log-in and be known on this site."
- Password:** Masked with "\*\*\*\*\*".
- Confirm Password:** Masked with "\*\*\*\*\*". Instruction: "Please enter a password for your user account. Note that passwords are case-sensitive."
- Email Address:** "renato.contreras@mail.udp.cl".
- Confirm Email Address:** "renato.contreras@mail.udp.cl". Instruction: "Please enter a valid email address for yourself."
- Human Verification:** The checkbox "I'm not a robot" is now checked with a green checkmark. The reCAPTCHA logo is still present.

Figura 2: Formulario rellenado en MMOChampion.

Aquí inspeccionaremos con la herramienta de Google Chrome y, en la sección de *Network*, identificaremos el archivo *register.php*:

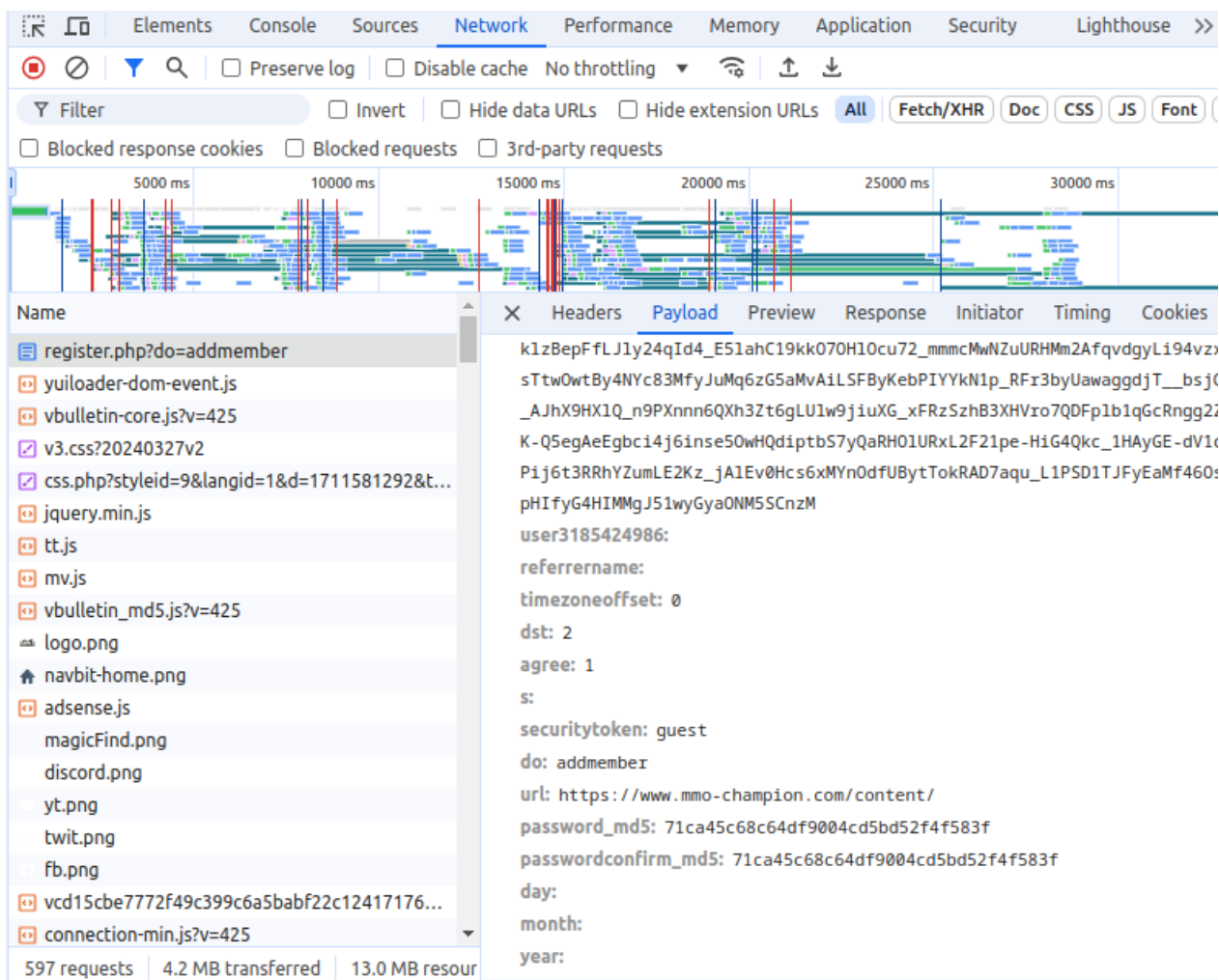


Figura 3: Tráfico al momento de realizar el registro.

De este archivo podemos observar que la contraseña está hasheada en MD5 con un valor de:

71ca45c68c64df9004cd5bd52f4f583f

Como supuestamente está hasheada en MD5 y tenemos la contraseña usada, verificaremos esto en la página web <https://10015.io/tools/md5-encrypt-decrypt>.

The image shows a web-based MD5 hash verification tool. It features two tabs: 'Encrypter' and 'Decrypter'. The 'Encrypter' tab is active, showing a 'Text' input field with the value 'sMyTvQ5AB@V5Zcy' and an 'MD5 Hash' output field with the value '71ca45c68c64df9004cd5bd52f4f583f'. A double arrow icon points from the text field to the hash field. At the bottom, there are three buttons: 'Encrypt >', 'Reset', and 'Copy'.

Figura 4: Verificación del formato de hash y su valor.

Comparando los valores obtenidos de la imagen anterior con los obtenidos al registrarnos, se observa que son iguales:

$$71ca45c68c64df9004cd5bd52f4f583f = 71ca45c68c64df9004cd5bd52f4f583f$$

Con este procedimiento verificamos que la página web utilizada usa el método de hash MD5 en su formulario de registro.

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

Para iniciar sesión se ingresan las credenciales correctas (RenaUDP, sMyTvQ5AB@V5Zcy):

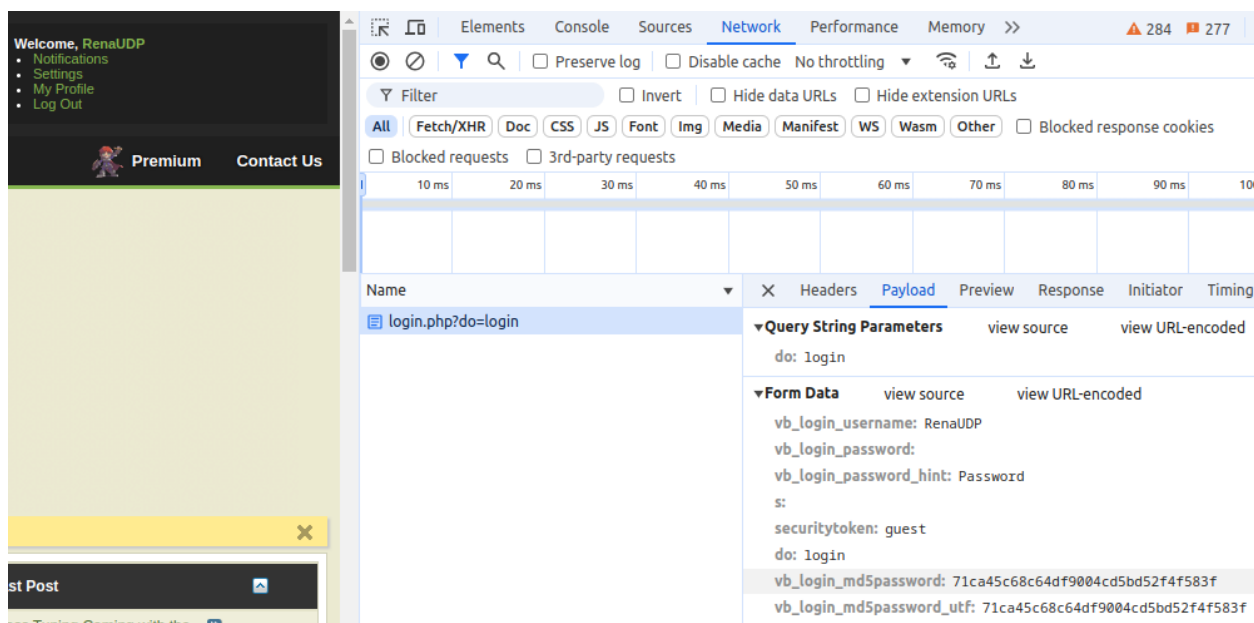


Figura 5: *Login.php*.

Nuevamente, podemos comparar el valor obtenido 'vb\_login\_md5password' que tiene el valor idéntico al hash verificado en el registro (71ca45c68c64df9004cd5bd52f4f583f).

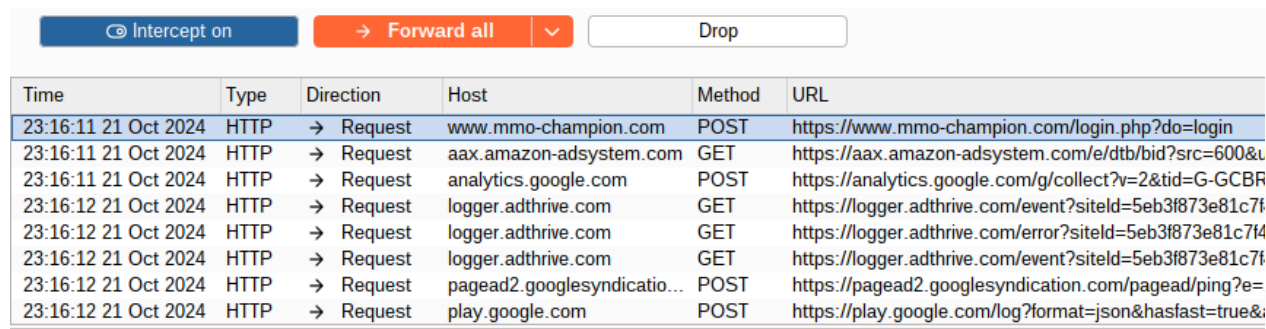
### 2.3. Genera el hash de la contraseña desde la consola del navegador

Para encontrar la función que necesitamos, nos ubicamos en el inicio de la página sin estar logueados (ya que al iniciar sesión, los archivos con el hash se eliminan), y nos dirigimos al apartado *Sources* dentro del inspector. Una vez ahí, buscamos con **Ctrl + F** el término "md5", y localizamos las funciones correspondientes.



## 2.4. Intercepta el tráfico login con BurpSuite

Para esta sección instalamos *BurpSuite Community*, donde se interceptan paquetes al momento de intentar un inicio de sesión correcto en la página.



Time	Type	Direction	Host	Method	URL
23:16:11 21 Oct 2024	HTTP	→ Request	www.mmo-champion.com	POST	https://www.mmo-champion.com/login.php?do=login
23:16:11 21 Oct 2024	HTTP	→ Request	aax.amazon-adsystem.com	GET	https://aax.amazon-adsystem.com/e/dtb/bid?src=600&u
23:16:11 21 Oct 2024	HTTP	→ Request	analytics.google.com	POST	https://analytics.google.com/g/collect?v=2&tid=G-GCBB
23:16:12 21 Oct 2024	HTTP	→ Request	logger.adthrive.com	GET	https://logger.adthrive.com/event?siteId=5eb3f873e81c7f
23:16:12 21 Oct 2024	HTTP	→ Request	logger.adthrive.com	GET	https://logger.adthrive.com/error?siteId=5eb3f873e81c7f
23:16:12 21 Oct 2024	HTTP	→ Request	logger.adthrive.com	GET	https://logger.adthrive.com/event?siteId=5eb3f873e81c7f
23:16:12 21 Oct 2024	HTTP	→ Request	pagead2.googlesyndication.com	POST	https://pagead2.googlesyndication.com/pagead/ping?e=
23:16:12 21 Oct 2024	HTTP	→ Request	play.google.com	POST	https://play.google.com/log?format=json&hasfast=true&

Figura 7: Interceptando un login en la página con BurpSuite.

```

9 Accept-Language: en-US,en;q=0.9
10 Origin: https://www.mmo-champion.com
11 Content-Type: application/x-www-form-urlencoded
12 Upgrade-Insecure-Requests: 1
13 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
14 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://www.mmo-champion.com/content/
20 Accept-Encoding: gzip, deflate, br
21 Priority: u=0, i
22 vb_login_username=RenaUDP&vb_login_password=5vb_login_password_hint=Password&se=6securitytoken=guest&do=login&vb_login_md5password=71ca45c68c64df9004cd5bd52f4f583f&vb_login_md5password_utf=71ca45c68c64df9004cd5bd52f4f583f

```

Figura 8: Contenido del paquete interceptado.

Por favor, revisa el repositorio para ver la imagen con buena calidad. De las imágenes anteriores destacamos el *POST* realizado por la página que contiene el nombre de usuario 'RenaUDP', así como 'vb\_login\_md5password' y 'vb\_login\_md5password\_utf', que comparten el valor '71ca45c68c64df9004cd5bd52f4f583f'.

## 2.5. Realiza el intento de login

En esta parte se repiten los pasos de la sección anterior, pero en vez de realizar un login correcto, se intenta con una contraseña incorrecta (123456), como se muestra en la siguiente imagen:

```

17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://www.mmo-champion.com/login.php?do=logout&logouthash=1729563864-10f301fffe35b3699036ac39b01232236f0e3031
20 Accept-Encoding: gzip, deflate, br
21 Priority: u=0, i
22 vb_login_username=RenaUDP&vb_login_password=5vb_login_password_hint=Password&se=6securitytoken=1729563868-814099b15c3d8a124a2d8b9c7f25a1b767c022636do=login&vb_login_md5password=10adc3949ba59abbe56e057f20f883e6
23 vb_login_md5password_utf=10adc3949ba59abbe56e057f20f883e

```

Figura 9: Contenido de un login incorrecto.



Como se ve en la última línea de la imagen anterior, las variables 'vb\_login\_md5password' y 'vb\_login\_md5password\_utf' comparten el valor 'e10adc3949ba59abbe56e057f20f883e', que corresponde al hash en MD5 de 123456, como se muestra a continuación:

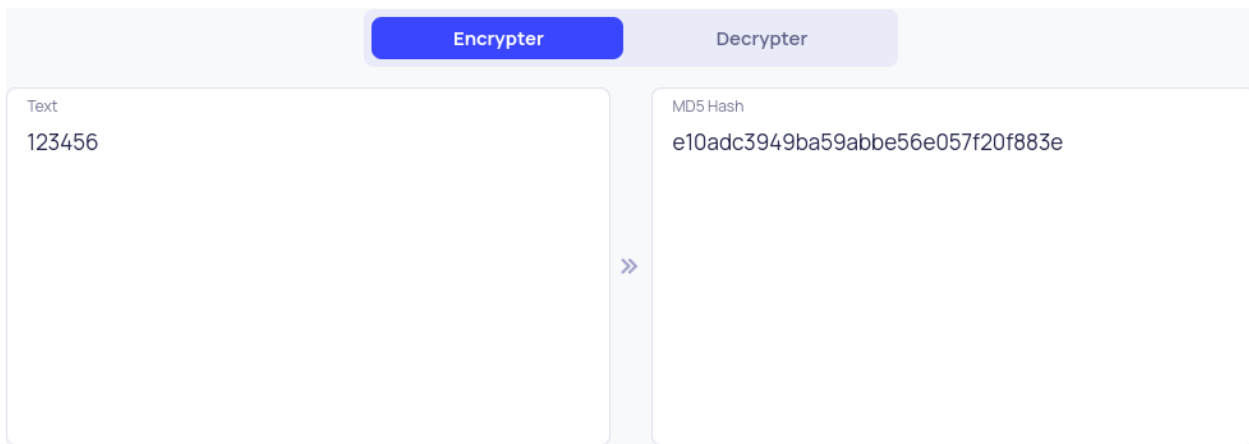


Figura 10: Encriptación con MD5 de la contraseña incorrecta.

Ahora cambiamos el valor por el correcto obtenido en la sección anterior, de la siguiente manera:

```
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://www.mmo-champion.com/login.php?do=logout&logoutash=1729563864-10f301fffe35b3699036ac39b01232236f0e3031
20 Accept-Encoding: gzip, deflate, br
21 Priority: u=0, i
22
23 vb_login_username=RenaUDP&vb_login_password=6vb_login_password_hint=Password&s=5securitytoken=1729563868-814099b15c3d8a124a2d8b9c7f25a1b767c022636do=login6vb_login_md5password=71ca45c68c64df9004cd5bd52f4f583f6
vb_login_md5password_utf=71ca45c68c64df9004cd5bd52f4f583f
```

Figura 11: Corrección al valor correcto de las variables 'vb\_login\_md5password' y 'vb\_login\_md5password\_utf'.

Luego, se envían todos los paquetes.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

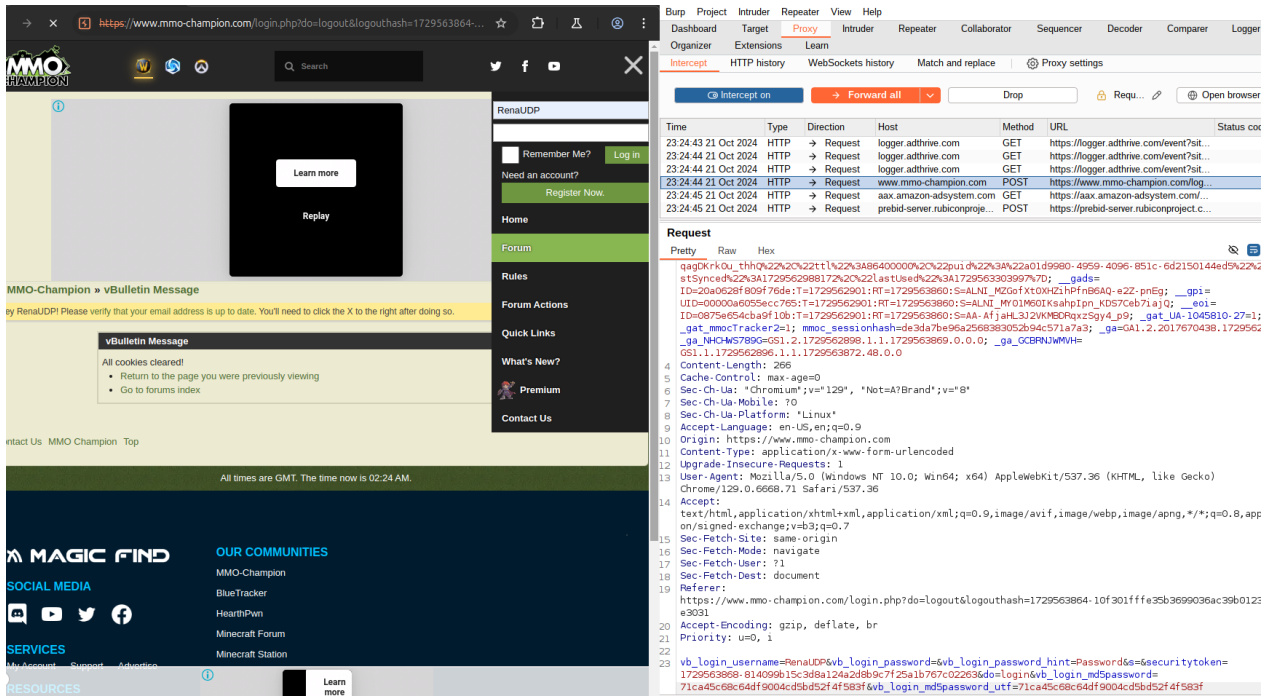


Figura 12: Antes del *Forward* en BurpSuite.

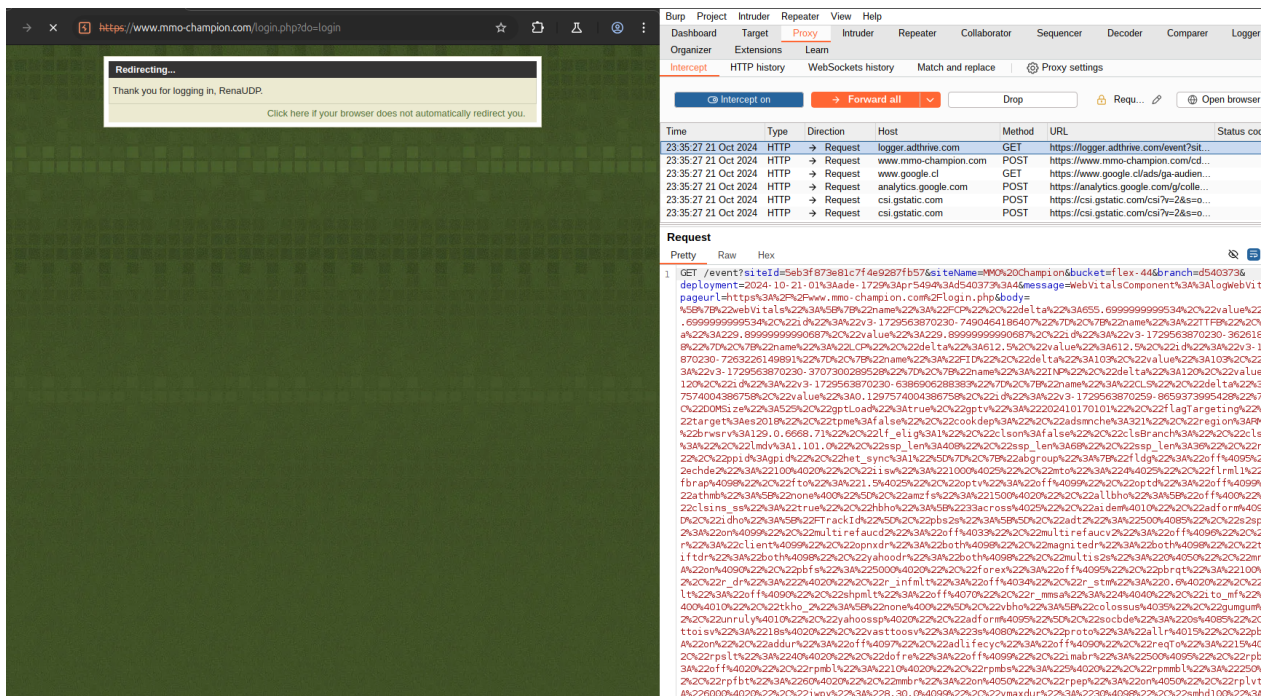


Figura 13: Después del *Forward* en BurpSuite.

Como se observa, se logra realizar un ataque de tipo *Pass the Hash*, que permite a un

atacante autenticarse sin conocer la contraseña original. En lugar de obtener la contraseña en texto plano, el atacante captura el valor hash y lo reutiliza para autenticarse.

## 2.6. Identifica las políticas de privacidad o seguridad

Las políticas de privacidad y seguridad de la página MMOChampion están descritas en <https://www.magicfind.us/privacy/>. Dichas políticas abordan el manejo de la información personal, el uso de cookies y la seguridad de los datos.

Respecto a las contraseñas, el sitio menciona que son protegidas mediante cifrado, pero no especifica el tipo de cifrado o el algoritmo hash utilizado. Este es un punto a mejorar, ya que la falta de transparencia puede ocultar vulnerabilidades, como el uso de algoritmos obsoletos como MD5 (observado en este laboratorio).

Algunos aspectos a destacar de las políticas pueden ser:

- Los datos personales no se comparten con terceros sin el consentimiento del usuario.
- Aunque las contraseñas están cifradas, no se detallan las metodologías o algoritmos empleados, lo que genera incertidumbre sobre la seguridad real.
- No se menciona el uso de técnicas avanzadas como *salting* o algoritmos modernos como SHA-256.

## 2.7. Demuestra 4 conclusiones sobre la seguridad

Cuatro conclusiones sobre la seguridad o vulnerabilidad de la implementación observada pueden ser:

1. **Vulnerabilidad al Pass the Hash:** El uso de MD5, un algoritmo hash vulnerable, permite ataques de *Pass the Hash*. Esto compromete las cuentas de los usuarios, ya que los atacantes pueden reutilizar los valores hash sin conocer la contraseña original.
2. **Falta de hashes con sal:** La ausencia de *salting* aumenta la vulnerabilidad ante ataques de tablas arcoiris y colisiones. Usar un *salt* único por usuario sería lo ideal para mejorar la seguridad.
3. **Debilidades en las políticas de seguridad:** Las políticas de privacidad no son claras respecto a los mecanismos de protección de contraseñas, lo que puede dar una falsa sensación de seguridad. Es necesario mayor detalle y transparencia.
4. **Algoritmos obsoletos:** El uso de MD5 es inadecuado debido a su vulnerabilidad a ataques modernos. Se recomienda migrar a algoritmos más seguros como bcrypt o SHA-256 para una mayor protección.

En conclusión, la implementación actual de seguridad es básica e insuficiente frente a amenazas modernas. Por lo que es recomendable que el sitio actualice sus prácticas de seguridad.