

# Informe Laboratorio 3

## Sección 4

José Berriós Piña  
e-mail: jose.berrios1@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 . . . . .	4
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 . . . . .	12
2.7. Demuestra 4 conclusiones sobre la seguridad . . . . .	13

## 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

Ingresamos a MMO-Champion (<https://www.mmo-champion.com>).

En el formulario de registro, hacemos clic derecho en el botón **Complete Registration** y seleccionamos la opción **Inspect**.

Una vez dentro de la pestaña **Network** de **Inspect**, activamos la opción **Preserve Logs** para que el tráfico generado al registrarse no desaparezca al ser redirigidos.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

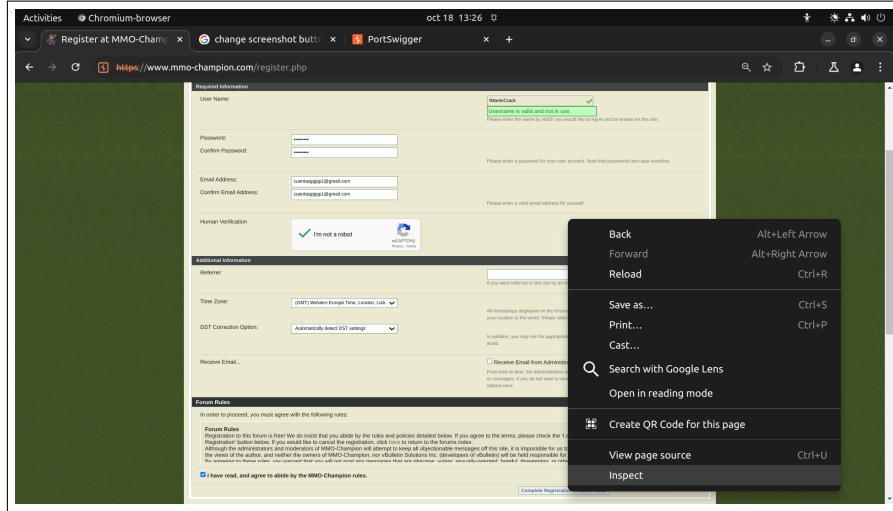


Figura 1: Formulario de registro en MMO-Champion.

Se completa y envía el formulario de registro con los siguientes datos:

- **Usuario:** 1MartinCrack
- **Clave:** varita123
- **Correo:** cuentasjpjpjp1@gmail.com

Una vez registrados, buscamos el documento **register.php?do=addmember**, dirigiéndonos al apartado **Payload** del documento seleccionado.

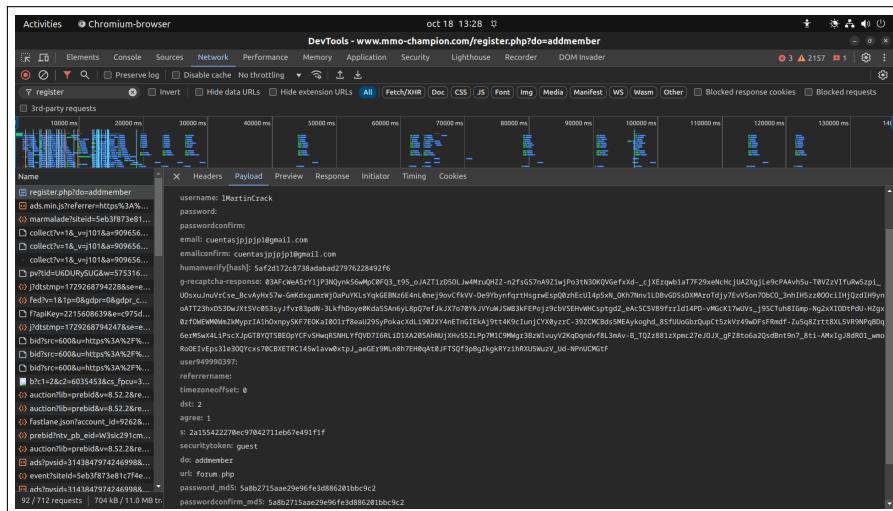


Figura 2: Inspección del Payload en el documento register.php.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Copiamos la contraseña hasheada de la Figura 2, es decir, el valor de la variable **password\_md5**:

- **password\_md5:** 5a8b2715aae29e96fe3d886201bbc9c2

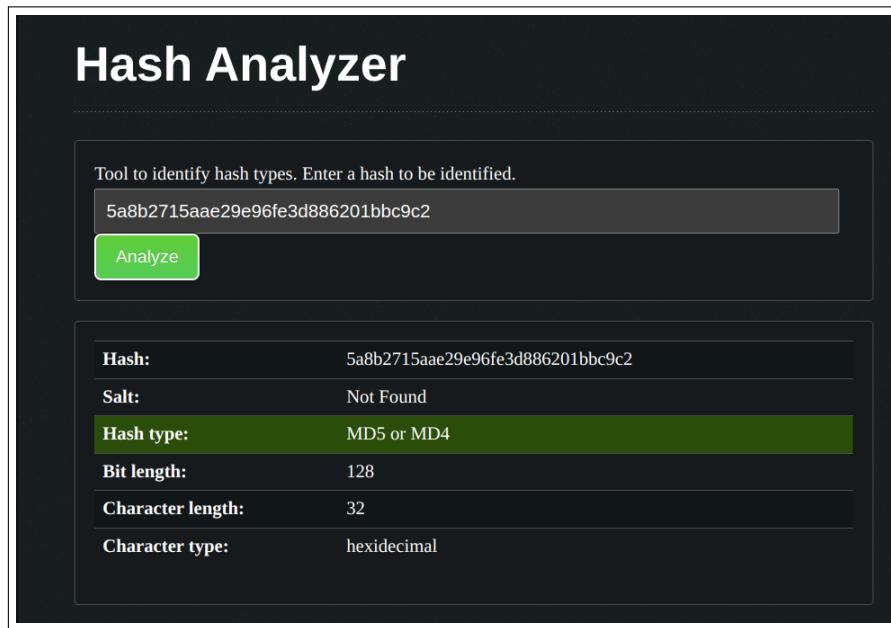


Figura 3: Análisis del hash **password\_md5**.

Utilizando la página Hash Analyzer, detectamos que el hash utilizado para el registro es MD5. Sin embargo, es importante notar que el uso de la página podría considerarse innecesario en este caso, ya que el mismo documento **register.php** incluye el sufijo **md5** en la variable **password\_md5**.

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

Ingresamos a MMO-Champion (<https://www.mmo-champion.com>).

En el formulario de inicio de sesión, hacemos clic derecho en el botón **Log in** y seleccionamos la opción **Inspect**.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

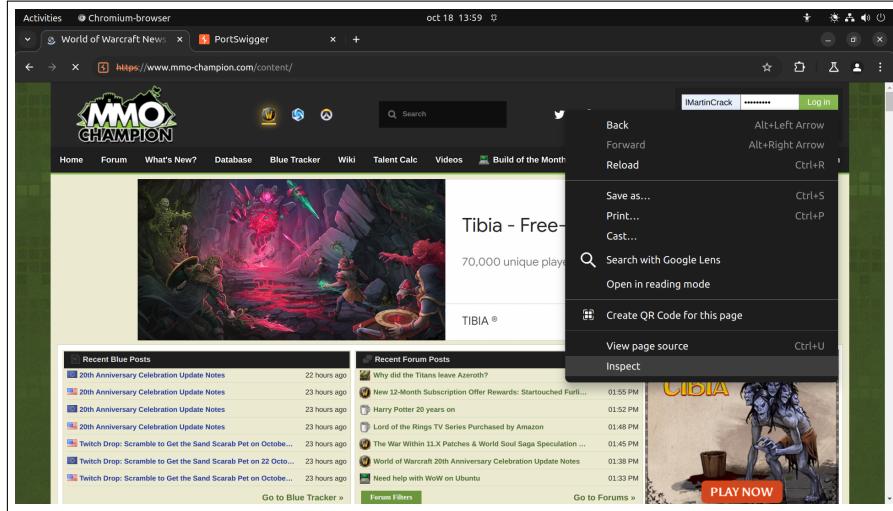


Figura 4: Inspección del inicio de sesión en MMO-Champion.

Una vez dentro del apartado **Network** de **Inspect**, activamos la opción **Preserve Logs** para que el tráfico generado al iniciar sesión no desaparezca al ser redirigidos.

Se completa y envía el formulario de ingreso con los siguientes datos:

- **Usuario:** lMartinCrack
- **Clave:** varita123

En la ventana **Network**, buscamos el documento **login.php?do=login**, y nos dirigimos al apartado **Payload** del archivo seleccionado.

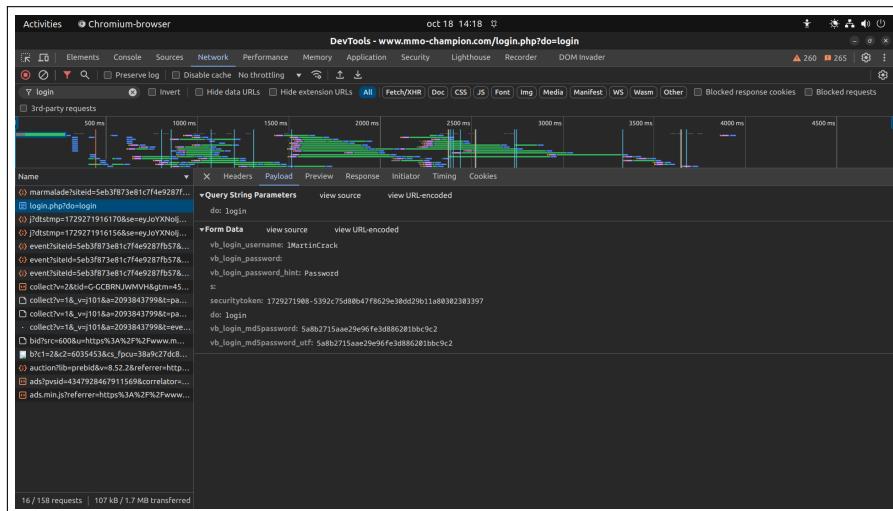


Figura 5: Archivo login.php con el usuario y su contraseña hasheada.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Se extrae que la variable con la contraseña hasheada es:

- **vb\_login\_md5password:** 5a8b2715aae29e96fe3d886201bbc9c2

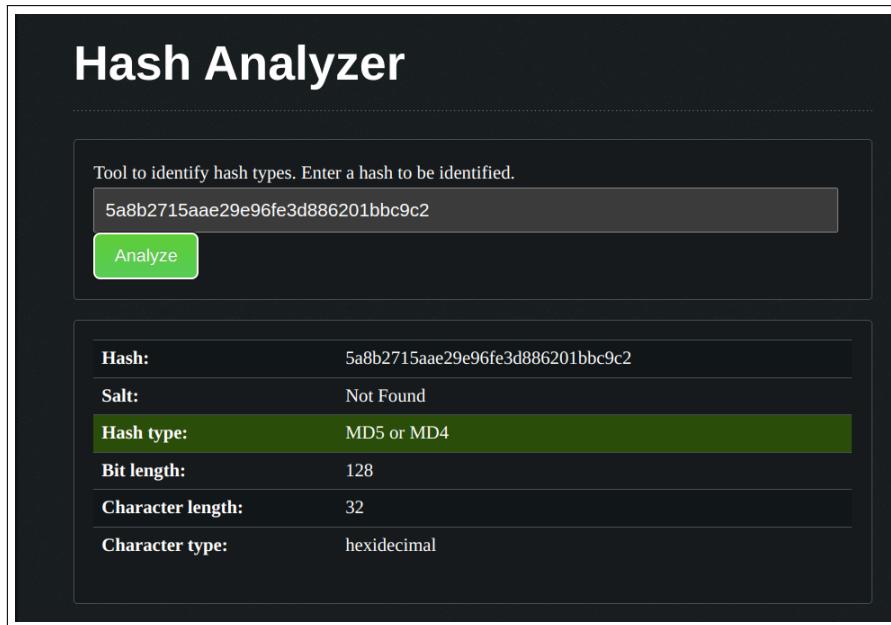


Figura 6: Análisis del hash **password\_md5**.

Utilizando la página Hash Analyzer, se detecta que el hash utilizado para el inicio de sesión es MD5. Sin embargo, es importante notar que el uso de la página podría considerarse innecesario, ya que en el documento **login.php** la variable **vb\_login\_md5password** aparece con el sufijo **md5**.

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

La página MMO-Champion utiliza un archivo JavaScript que contiene una función encargada de **hashear** la contraseña al iniciar sesión.

Este archivo JavaScript es solicitado por la página cuando se está deslogueado, de manera que esté disponible antes de iniciar sesión.

Una vez iniciada la sesión, la aplicación reemplaza el archivo que contiene la función de encriptado. El archivo **vbulletin\_md5.js** (utilizado antes del inicio de sesión para encriptar contraseñas) es sustituido por **vbulletin\_read\_marker.js**, que se encarga de gestionar los marcadores de lectura en MMO-Champion.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Activities ◊ Chromium-browser

oct 18 14:15 ⓘ

DevTools - www.mmo-champion.com/login.php?do=login

Page Top Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder DOM Invader

13 6486 4

Threads Watch Breakpoints

Pause on uncaught exceptions Pause on caught exceptions

Scope Not passed

Call Stack Not passed

XHR/Fetch Breakpoints DOM Breakpoints Global Listeners Event Listener Breakpoints CSP Violation Breakpoints

```
var hexValue = 0;
var b64split = '';
var chsz = 8;
function hex2binstr(A) {
    return bin2binstr(core_md5(str2bin1(A), A.length * chsz));
}
function hex4_md5(A) {
    return bin2binstr(core_md5(str2bin1(A), A.length * chsz));
}
function str_md5(A) {
    return bin2binstr(core_md5(str2bin1(A), A.length * chsz));
}
function hex_hmac_md5(A, B) {
    return bin2binstr(core_hmac_md5(A, B));
}
function hex4_hmac_md5(A, B) {
    return bin2binstr(core_hmac_md5(A, B));
}
function str_hmac_md5(A, B) {
    return bin2binstr(core_hmac_md5(A, B));
}
function core_md5(k1, F) {
    K[0] >= 5 ? 128 << ((F & 32) + 4) : 128 << ((F & 32) + 4) + 14;
    var j = 172584193;
    var I = 271738879;
    var H = 1732584194;
    var T = 3233486789;
    for (var C = 0; C < K.length; C += 16) {
        var E = 0;
        var D = 0;
        var B = H;
        var A = T;
        G = MD5_F1(G, J, I, H, K[C], 8, -68876936);
        G = MD5_F1(G, J, I, H, K[C + 1], 12, 389564586);
        H = MD5_F1(H, J, I, H, K[C + 2], 17, 68618519);
        I = MD5_F1(I, H, G, J, K[C + 3], 22, -1844525539);
        T = MD5_F1(T, G, H, I, K[C + 4], 7, 30151429);
        G = MD5_F1(G, J, I, H, K[C + 5], 12, 1208884261);
        H = MD5_F1(H, J, I, H, K[C + 6], 17, 1473231341);
        I = MD5_F1(I, H, G, J, K[C + 7], 22, -45785983);
```

Figura 7: Archivo JavaScript de la ventana `source` con la función `vbulletin_md5.js` encargado del cifrado mediante la función `hex_md5(x)`.

El nombre específico de la función encargada del hash es **hex\_md5(x)**, como se observa en la Figura 7. Esta función se ejecutara mediante la consola, obteniendo el hash de la palabra **varita123**.

```
Activities ◊ Chromium-browser
          oct 18 15:44 □
DevTools - www.mmo-champion.com/login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6
Elements Console Sources Network Performance Memory Application Security Lighthouse Recorder DOM Inspector
Default levels ▾ 1,544 issues: 1514 open 30 1 hidden
[! top | ↻ | Filters]
⚠ A parser-blocking cross site (i.e. different document) script, https://www.mmo-champion.com/login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6, is login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6.js?v=425 in the network request for this script MAY be blocked by the browser in this or a future page load due to poor network connectivity. If blocked in this page load, it will be confirmed in a subsequent console message. See https://www.chromestatus.com/feature/718547246799104 for more details.
⚠ A parser-blocking cross site (i.e. different document) script, https://ajax.googleapis.com/ajax/libs/yui/2.9.0/build/connection/connection-min.js?v=425, is login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6.js?v=425 invoked via document.write. The network request for this script MAY be blocked by the browser in this or a future page load due to poor network connectivity. If blocked in this page load, it will be confirmed in a subsequent console message. See https://www.chromestatus.com/feature/718547246799104 for more details.
This browser is AJAX compatible vbulletin-core_js?v=425_1
Fire System Init vbulletin-core_js?v=425_1
Fire vB_XHTML_Ready vbulletin-core_js?v=425_1
Fetch Cookie :: vbulletin_cookie [null] vbulletin-core_js?v=425_1
⚠ fire-hooks: referenced 'adpod' but it was never created regedit_main.js
adsp-1111-origins adsp-1111-origins adsp-1111-origins > Array(1)
⚠ join-3rd-party: Permissions Policy feature join-all-interest-group will not be enabled by default in cross-origin iframes or same-origin iframes nested in cross-origin iframes. Calling joinInterestGroup() will be rejected with NotAllowedError if it is not explicitly enabled join_3rdparty_ifr_1_to_innerOpnpx.js
Request successful > Response
Powered by AMP + HTML - Version 24062416252800 https://www.mmo-champion.com/login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6
⚠ 1003: Third-party cookie will be blocked in future Chrome versions as part of Privacy Sandbox. join_3rdparty_ifr_1_to_innerOpnpx.js
Powered by AMP + HTML - Version 24062416252800 https://www.mmo-champion.com/login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6
⚠ Attestation check for Attribution Reporting on https://echo.quantumt.cc failed. VMI029249_atmoads_vb_mis_2
Powered by AMP + HTML - Version 24062416252800 https://www.mmo-champion.com/login.php?d=logout&logouthash=1729276794-0d7ab0f3ab556e5a06c5c859d39bc1de30c7d6
> hex_md5('varlist23') VMI03119_atmoads_vb_mis_2
> *5a8b2715aae79e96fe3d886201bb0c9c2'
```

Figura 8: Aplicación del hash vía consola a la contraseña **varita123**.

- Contraseña: varita123
  - Resultado del hash: 5a8b2715aae29e96fe3d886201bbc9c2

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

### 2.4. Intercepta el tráfico login con BurpSuite

Se inicia **BurpSuite** con las configuraciones básicas. Luego, navegamos a la sección **Proxy/Intercept** y se inicia el navegador integrado de la aplicación, lo que permite ahorrar configuraciones adicionales de licencias de navegadores.

En el navegador de **BurpSuite**, ingresamos a MMO-Champion y nos dirigimos al formulario de login.

Antes de presionar el botón **Log in** con el usuario registrado y la contraseña correcta, en **BurpSuite** debemos comenzar la captura presionando **Intercept**.

Analizamos la lista de paquetes capturados y seleccionamos el que corresponde al login, es decir, el paquete **www.mmo-champion.com POST**.

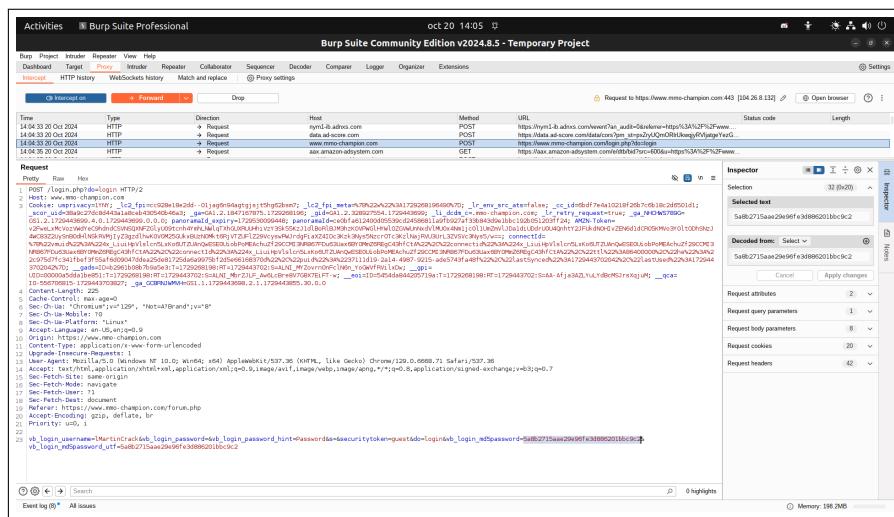


Figura 9: POST /login.php?do=login HTTP/2.

Del paquete capturado, se obtiene lo siguiente:

- **vb\_login\_md5password:** 5a8b2715aae29e96fe3d886201bbc9c2

Al comparar la contraseña hasheada con la ingresada, es decir, **varita123**, podemos comprobar que se ha aplicado el hash utilizando la función **hex\_md5('varita123')**, tal como verificamos anteriormente a través de la consola (ver Sección 2.3).

### 2.5. Realiza el intento de login

Se inicia **BurpSuite** con las configuraciones básicas. Luego, navegamos a la sección **Proxy/Intercept** y se inicia el navegador de la aplicación, lo que permite evitar configuraciones adicionales de licencias de navegadores.

En el navegador de **BurpSuite**, ingresamos a MMO-Champion y nos dirigimos al formulario de login.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

Se realizará un **intento de login incorrecto** con los siguientes datos:

- **Usuario:** lMartinCrack
- **Clave:** arco123

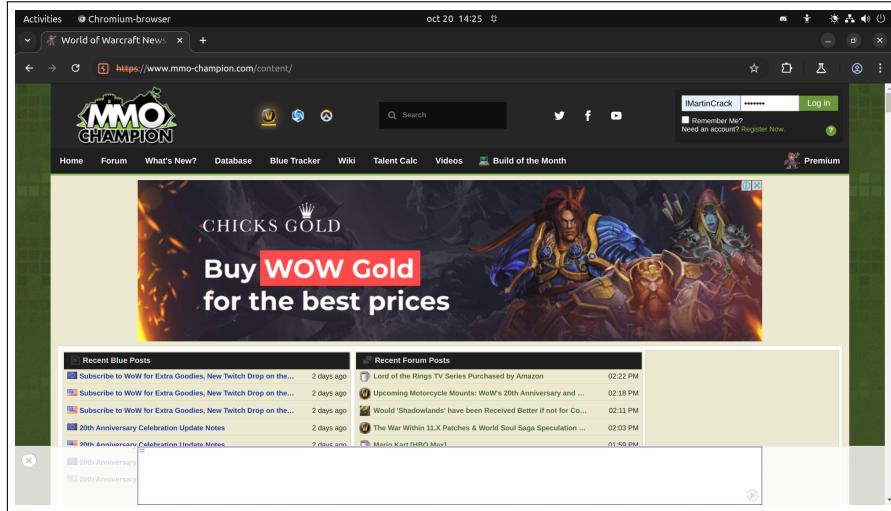


Figura 10: Login con usuario correcto y contraseña incorrecta.

Se comienza la captura en BurpSuite presionando '**Intercept On**', y luego se presiona el botón **Log In** de MMO-Champion con los datos anteriores.

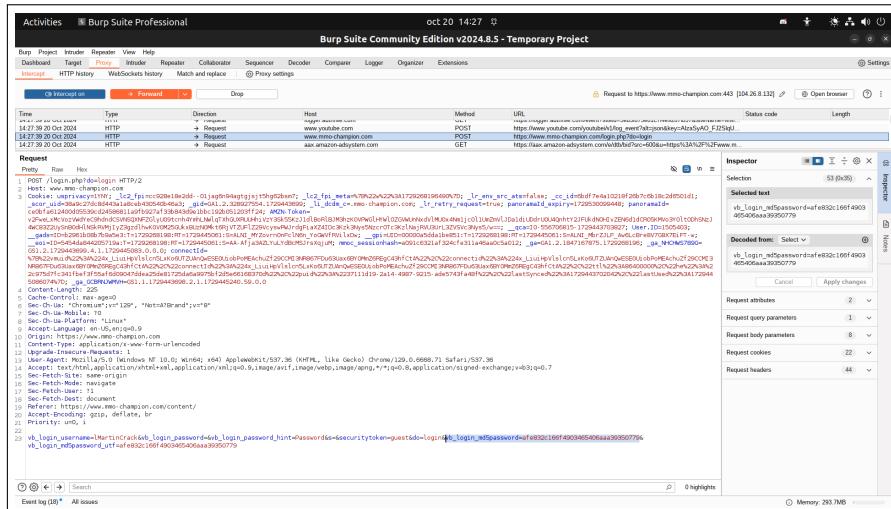


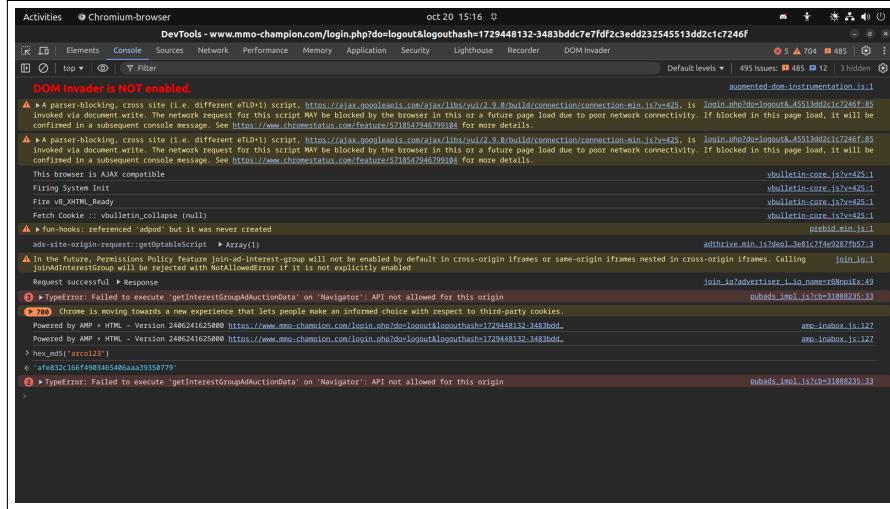
Figura 11: POST /login.php?do=login HTTP/2.

De la captura del paquete, se obtiene la siguiente información:

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

- **vb\_login\_md5password:** afe832c166f4903465406aaa39350779
- **vb\_login\_md5password\_utf:** afe832c166f4903465406aaa39350779

Se verifica que la clave ingresada fue **arco123**, esto debido a la utilización de la función de hash **hex\_md5('arco123')**.



The screenshot shows the Chrome DevTools console output. The command `hex_md5('arco123')` was run, resulting in the output `afe832c166f4903465406aaa39350779`. The console also displays various other developer information and errors.

Figura 12: Clave **arco123** hasheada a través de la consola.

Ya identificado el paquete **www.mmo-champion.com** POST, procedemos a modificar los valores correspondientes al ingreso del usuario, es decir:

- **vb\_login\_md5password**
- **vb\_login\_md5password\_utf**

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

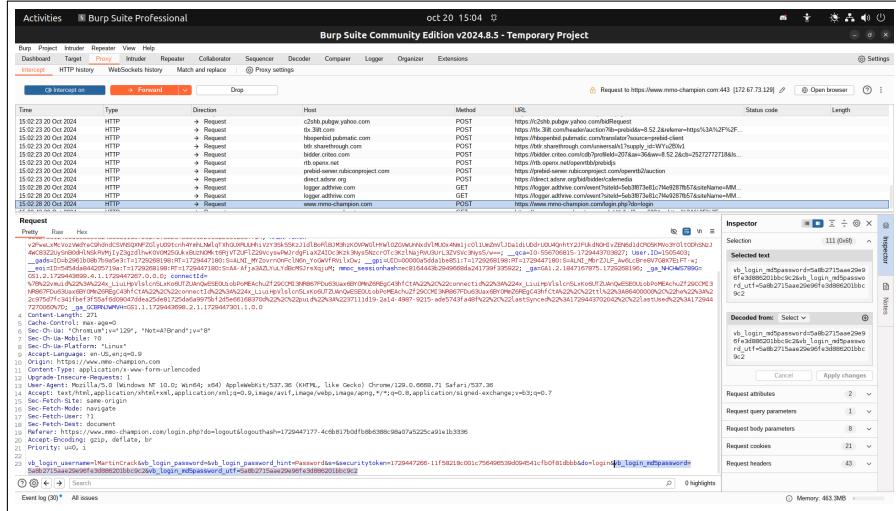


Figura 13: Modificación del POST login con contraseñas correctas.

Una vez modificadas las claves, enviamos el paquete POST presionando **Forward** en la misma ventana de **Proxy**, y detenemos la captura para asegurar una redirección sin problemas.

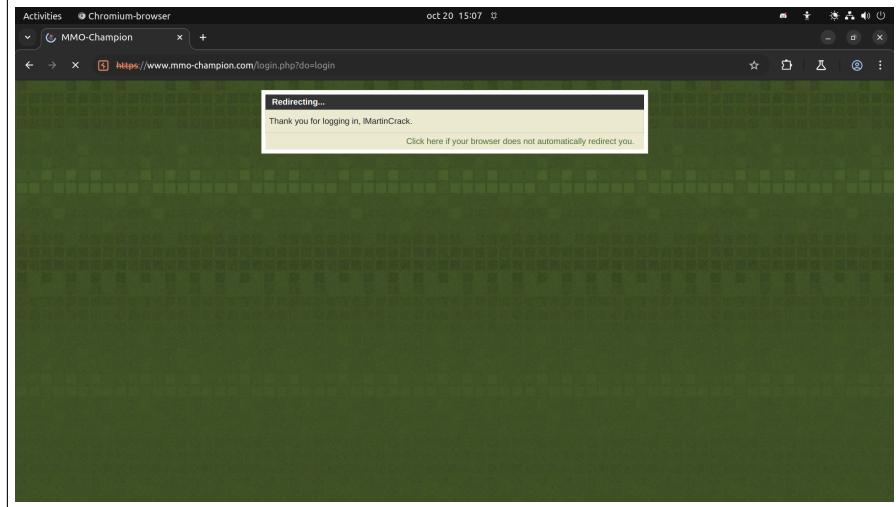


Figura 14: Visualización de la página al enviar el paquete con los campos modificados correctamente.

## 2 DESARROLLO DE ACTIVIDADES SEGÚN CRITERIO DE RÚBRICA

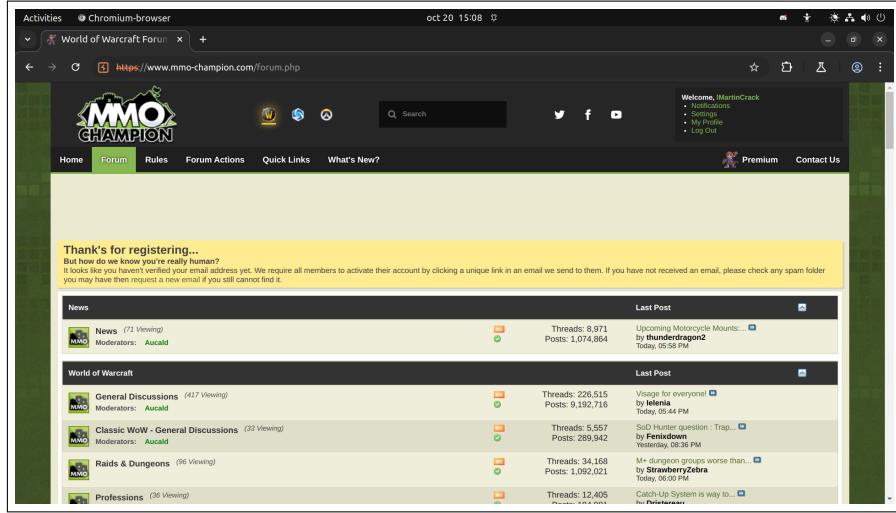


Figura 15: Redirección final al Lobby del usuario 1MartinCrack.

El ingreso a la cuenta del usuario fue exitoso, intercambiando el hash de la contraseña incorrecta por el hash de la contraseña correcta.

### 2.6. Identifica las políticas de privacidad o seguridad

La política de privacidad del sitio MMO-Champion está regulada por Magic Find, Inc., estableciendo directrices claras en cuanto a la protección de la información personal y los mecanismos de seguridad implementados.

Según la política de privacidad, existen los siguientes datos recolectados:

- **Datos proporcionados voluntariamente:** Información que el usuario entrega, como nombre de usuario, correo electrónico, detalles de la cuenta, etc.
- **Datos recopilados automáticamente:** Datos enviados automáticamente por los dispositivos del usuario al interactuar con la página web. Esto incluye la dirección IP, las páginas visitadas, y el tiempo dedicado a cada una, tipo de navegador, etc.

Por otro lado, Magic Find implementa medidas de seguridad razonables para proteger la información personal de accesos no autorizados o alteraciones.

Aunque reconocen que ningún sistema es 100 % seguro, y citando textualmente: “*Although we will do our best to protect the personal information you provide to us, we advise that no method of electronic transmission or storage is 100 % secure and no one can guarantee absolute data security.*”, luego de esa cita, sugieren que los usuarios adopten buenas prácticas de seguridad, como la selección de contraseñas robustas y mantener la confidencialidad de estas.

Magic Find tiene políticas relacionadas con el uso de cookies y la compartición de datos con terceros para fines publicitarios, donde Magic Find colabora con servicios externos, como

Braintree para el manejo de pagos y Raptive para publicidad, que también recolectan información personal.

Magic Find asegura el cumplimiento de regulaciones de protección de datos para transferencias internacionales fuera del Área Económica Europea.

**Enlace a la política de privacidad:** <https://www.magicfind.us/privacy/>.

### 2.7. Demuestra 4 conclusiones sobre la seguridad

1. **Debilidades del algoritmo de hash MD5:** A lo largo del informe se observó que tanto durante el registro como en el inicio de sesión en MMO-Champion se utilizó el algoritmo MD5 para hashear las contraseñas. En la actualidad MD5 es un algoritmo considerado inseguro debido a su vulnerabilidad a ataques de colisión y fuerza bruta, lo cual facilita el descifrado de contraseñas si los atacantes logran capturar los hashes.
2. **Possibles riesgos en la transmisión de datos:** Aunque las contraseñas son hasheadas antes de ser enviadas al servidor, interceptar el tráfico con BurpSuite demostró que, si un atacante puede obtener el hash de la contraseña, podría intentar un ataque del tipo "Pass the Hash" (PtH), utilizando el hash directamente para acceder al sistema sin necesidad de descifrar la contraseña.
3. **Dependencia en la seguridad del cliente:** La implementación de hash se realiza del lado del cliente a través de JavaScript, lo que implica que la lógica de seguridad es visible y modificable por cualquier usuario con acceso a las herramientas de desarrollo del navegador. Esto se identificaría como una vulnerabilidad, ya que cualquier usuario podría alterar el código de la página para manipular el proceso de login o modificar el hash.
4. **Políticas de privacidad y seguridad adecuadas, pero mejorables:** Si bien el sitio cuenta con políticas de privacidad robustas y claras en cuanto a la protección de los datos personales de los usuarios, el uso de un algoritmo obsoleto como MD5 no está alineado con las mejores prácticas de seguridad modernas. Se recomienda que se implemente un algoritmo más seguro como **SHA-256** o alargamiento de claves para fortalecer la seguridad de las contraseñas.