

Informe Laboratorio 3

Sección 2

Alumno Bruno Rosales
e-mail: bruno.rosales@mail.udp.cl

Octubre de 2025

Índice

1. Descripción de actividades	2
2. Desarrollo de actividades según criterio de rúbrica	2
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	6
2.5. Realiza el intento de login por medio del hash	8
2.6. Identifica las políticas de privacidad o seguridad	10
2.7. Comente 4 conclusiones sobre la seguridad del sitio escogido	11
2.8. Conclusiones sobre la seguridad del sitio escogido	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 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

El primer paso para encontrar un sitio para probar hash es utilizar publicwww, el cual es un motor de búsqueda de código fuente web. Se opta por buscar MD5 en este sitio, un algoritmo de hash de 16 bytes, ya que es bastante antiguo y fácil de detectar.

Se opta por el sitio moneycontrol.com, ya que es el único que coincide en tener un sistema de registro e ingreso, estar disponible en inglés por defecto (hay muchos sitios en ruso, chino

y coreano), ser SFW (Safe For Work) y además de mostrar que contiene MD5 ('MD5=...'). Esto se aprecia en la siguiente Figura 1.

Rank	Url	Snippets
49	https://ok.ru/	hotowall", "OK/utills/md5": "st.okcdn.ru/res
102	http://thetartmagazine.com/feed/summary	6-09 06:28:32.045", "md5": "18216df2-eab8-eb6
180	https://www.douyu.com/	ype: Normal Content-MD5: +rzXyx7b8c93yuhuyL
280	https://www.livejournal.com/	-revalidate Content-MD5: dXWiEpFsbuGsTJO9gr
364	https://www.accuweather.com/	ent-Length, Content-MD5, Date, X-API-Version
525	https://www.moneycontrol.com/	x1rw== x-goog-hash md5=qbFw8CWcSvi9yWfhyxx

Figura 1: Búsqueda del algoritmo de cifrado MD5 mediante el sitio 'publicwww'. Se aprecian múltiples sitios y destacado, 'moneycontrol.com', el sitio elegido.

Posteriormente, se procede al proceso de registrarse con una cuenta, utilizando las credenciales 'brunotrone12345@gmail.com' como gmail y 'Contr@generica12345' como contraseña. Esto se hace mientras se utiliza la herramienta de desarrollador de red desde el navegador, lo que permite captar el tráfico.

El primer resultado es una solicitud que verifica si el usuario (correo) ya existe en el sistema, como se aprecia en la Figura 2. Esto no se encuentra cifrado, lo cual puede conllevar fallas de seguridad.

St...	Me...	Domain	File	Initiator	Ty...	Transferr...	Si...
200	PO...	accou...	OVIM	OVIM:1 (x...	json	2.13 kB	18...
200	PO...	accou...	checkifuserexistbyemail	jquery-1...	ht...	3.31 kB	5...
200	GET	appfe...	indices&ind_id=9?callback=	jquery-1...	json	1.06 kB	19...

3 requests | 1.97 kB / 6.49 kB transferred | Finish: 11.24 s

email: "brunotrone12345@gmail.com"

Figura 2: Tráfico captado al registrarse con el correo. Se puede apreciar que no se le aplica ningún algoritmo de cifrado.

El segundo resultado es el más relevante. Luego de completar el proceso de registro en el

sitio, se logra captar con las herramientas de desarrollador de red desde el navegador el tráfico de la solicitud. En este caso, se tiene el correo sin cifrado, lo cual es un riesgo de seguridad. Por parte de la contraseña, se encuentra cifrada: '81b2e4f296ea6b696851199d4dd66eaa0abc6f5d', pero no en MD5 como se esperaba, sino que en SHA-1. Esto se deduce por la longitud característica de este algoritmo, 40 hexadecimales. Esto se aprecia en la Figura 3.

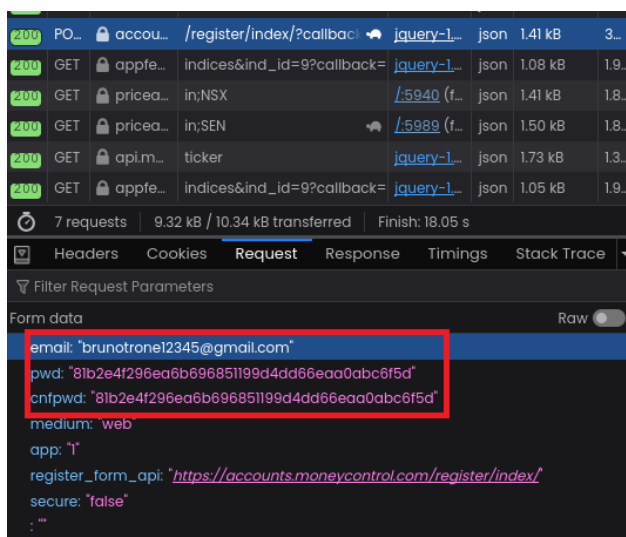


Figura 3: Tráfico captado al registrarse con el correo y contraseña. Se puede apreciar que no se le aplica ningún algoritmo de cifrado al correo, sin embargo, la contraseña y la confirmación de la contraseña se encuentran cifradas.

Para verificar esto, se opta por utilizar otra herramienta de desarrollador del navegador; en este caso, la consola. Desde aquí, se pueden estimular las funciones de Javascript del sitio. En este caso se logra estimular la función del algoritmo de cifrado 'SHA-1' con la contraseña utilizada en el registro, obteniendo el mismo hash captado:

- '81b2e4f296ea6b696851199d4dd66eaa0abc6f5d'

Esto se ve en detalle en la Figura 1.

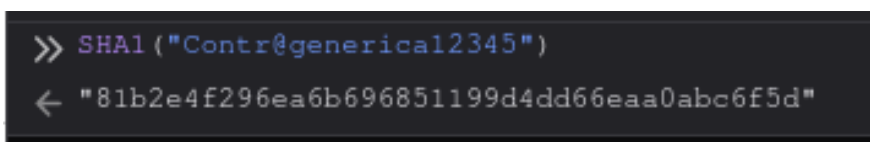


Figura 4: Función Javascript 'SHA-1' estimulada desde consola en el navegador. Se aprecia el mismo hash capturado en el proceso de registro.

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

El proceso para identificar el algoritmo de ingreso de sesión es el mismo que el de registro.

Para esto se capta el tráfico desde la parte de ingreso de sesión de la página web. Se utilizan las credenciales 'brunotrone12345@gmail.com' como gmail y 'Contr@generica12345' como contraseña. Luego de un ingreso correcto, se logra capturar el tráfico de la solicitud enviada, obteniendo el campo de correo (sin cifrar, al igual que en el caso del registro) y el campo de la contraseña (como 'pwd') cifrada en SHA-1: '81b2e4f296ea...'. Los resultados se pueden apreciar en la Figura 5.

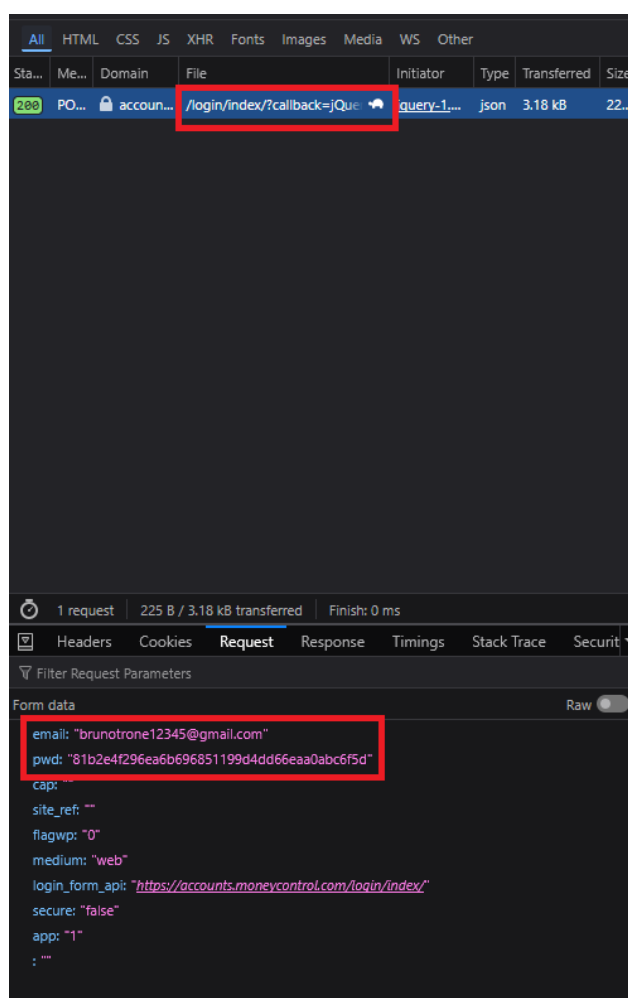


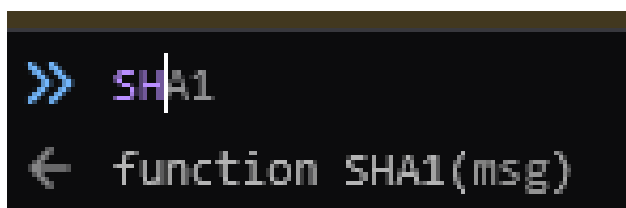
Figura 5: Tráfico captado al ingresar con el correo y contraseña. Se puede apreciar que no se le aplica ningún algoritmo de cifrado al correo, sin embargo, la contraseña se encuentra cifrada.

Para verificar esto se hace uso del mismo metodo, estimulación de la función Javascript 'SHA-1' detectada. En este caso, coincide el hash capturado con el hash generado por el

algoritmo SHA-1, por lo que se comprueba la presencia de este metodo de cifrado. Se puede ver en la Figura 3 el resultado obtenido.

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

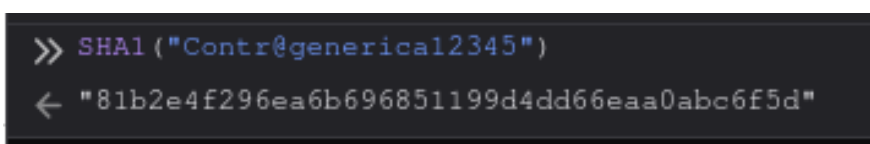
Para generar el hash, se utiliza el metodo que se viene usando desde la primera actividad. En primera instancia se comprueba la existencia de alguna función existente de cifrado (SHA-1, SHA-256, MD5) mediante su nombre en la consola. Esto permite saber si existe la función y llamarla. En el caso de SHA-1, como la función 'SHA-1', se detecta una función por consola, con el parámetro de 'msg' como input de la función. Esto es una mala práctica, ya que siempre se han de ofuscar las funciones relevantes a usar en el código. Sin embargo, esto es de utilidad para la actividad, por lo que se utiliza como se puede apreciar la Figura 6.



```
>> SHA1
< function SHA1(msg)
```

Figura 6: Función detectada 'SHA-1' al buscar por la consola del navegador distintos algoritmos de cifrado.

Esto nos da como resultado el hash obtenido en la actividad 2.1 y 2.2: '81b2e4f296e...'. Este hash es una salida de la función estimulada con el 'msg' como 'Contr@generical2345', y al coincidir nos esta indicando que se obtuvo de manera exitosa el hash de la contraseña mediante la consola. El resultado se ve en la Figura 7.



```
>> SHA1("Contr@generical2345")
< "81b2e4f296ea6b696851199d4dd66eaa0abc6f5d"
```

Figura 7: Función estimulada con la contraseña usada, mediante la consola del navegador. Se observa como se obtiene el mismo hash que el capturado en actividades previas.

2.4. Intercepta el tráfico login con BurpSuite

Para interceptar tráfico del ingreso en el sitio 'moneycontrol.com', se utiliza BurpSuite, una herramienta para identificar y analizar vulnerabilidades, usada en actividades anteriores. Específicamente, se utiliza el navegador que provee, la herramienta de historial HTTP para ver las solicitudes y el apartado de 'Intruder', que permite modificar solicitudes HTTP con valores propios. Esto nos permite extraer y analizar todo el tráfico HTTP desde la página. De esta manera, se pueden obtener tanto las solicitudes, hash, correo, entre otras variables y usarlas para probar la funcionalidad del hash.

El primer paso es abrir BurpSuite e iniciar el navegador que este suite provee. Posteriormente, se navega hasta la página y se realizan 2 intentos de login, uno incorrecto, con la contraseña 'password' y uno correcto con la contraseña 'Contr@generica12345', todo esto mientras se captura el tráfico HTTP mediante el historial HTTP del software. Esto nos permite obtener 2 hash; 1 incorrecto y otro correcto, que se usarán posteriormente. Esto se puede apreciar en la Figura 8 y Figura 9 con la contraseña errónea, y en la Figura 10 y Figura 11 con la contraseña correcta.

- Hash incorrecto de 'password': '5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8'
- Hash correcto de actividades pasadas: '81b2e4f296ea6b696851199d4dd66eaa0abc6f5d'

26...	https://accounts.moneyc...	POST	/login/index/?callback=jQueryL...	✓	200	3096	script
26...	https://sync.lrxio	GET	/usersyncz/rmpssp?sub=360...	✓	302	398	HTML
26...	https://cs.visiblemeasure...	GET	/pbserver?gdpr=&gdpr_conse...	✓	404	155	text
26...	https://sync.targeting.unr...	GET	/csync/RX-773e8612-6ab3-4b...	✓	302	546	HTML
26...	https://gattpmn.io	GET	/sync/redirect?gdpr=&gdpr_c...	✓	302	827	

Figura 8: Tráfico interceptado mediante BurpSuite, donde se aprecia la ruta 'login' usada para el ingreso.

Name	Value	
email	brunotrone12345@g...	>
pwd	5baa61e4c9b93f3f0...	>

Figura 9: Variables capturadas de la solicitud de login. Se aprecia el campo de correo sin cifrar y el campo de contraseña, con un hash distinto.

4351	https://accounts.moneyc...	POST	/login/index/?callback=jQueryL...	✓	200	5382	script
------	----------------------------	------	-----------------------------------	---	-----	------	--------

Figura 10: Tráfico interceptado mediante BurpSuite, donde se aprecia la ruta 'login' usada para el ingreso.

Name	Value	
email	brunotrone12345@g...	>
pwd	81b2e4f296ea6b696...	>

Figura 11: Variables capturadas de la solicitud de login. Se aprecia el campo de correo sin cifrar y el campo de contraseña, con un hash que coincide con el de actividades previas.

2.5. Realiza el intento de login por medio del hash

Para realizar un intento de login mediante el hash, se modifica un intento de login correcto (ya capturado en el punto anterior) cambiando el hash de la contraseña por uno incorrecto (también capturado). También se realiza el opuesto, osea, un intento incorrecto se le cambia a un hash correcto. Esto se hace mediante la funcionalidad de 'Intruder' de BurpSuite.

En primera instancia se realiza el cambio por un hash incorrecto. Para esto utilizamos el 'Intruder', el cual nos permite cambiar variables de la solicitud y enviar ataques al sitio con estos nuevos valores. En este caso se modifica el valor del hash correcto por uno incorrecto, resultando en una solicitud rechazada o 'Acces Denied' como respuesta, debido a que no se puede verificar la integridad de la contraseña. Esto se aprecia en Figura 12, Figura 13 y Figura 17.

```
20 email=brunotrone1234540gmail.com&pwd=$81b2e4f296ea6b696851199d4dd66eaa0abc6f5ef cap=&site_ref=&flagwp=0&medium-web&  
21 login_form_api=https%3A%2F%2FAccounts%2Flogin%2Fapi%3Fapp=1&false&app=1&
```

Figura 12: Variables capturadas de la solicitud de login. Se aprecia el campo de correo sin cifrar y el campo de contraseña, con un hash correcto a modificar.

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

Paste

Load...

Remove

Clear

Deduplicate

Add

Add from list... [Pro version only]

5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8

Figura 13: Variable con el valor del hash de la contraseña incorrecta capturada. Esta variable reemplazará el valor del hash correcto.

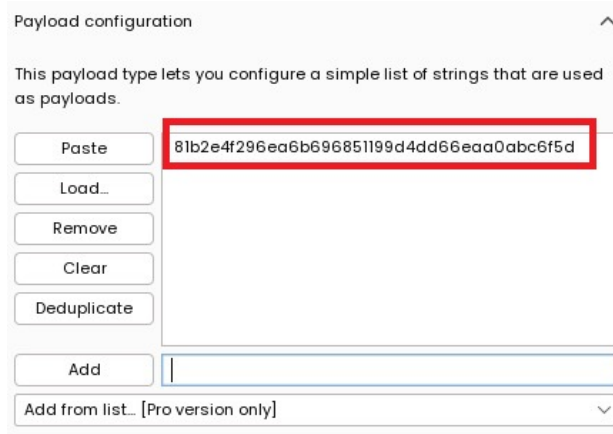


Figura 16: Variable con el valor del hash de la contraseña correcta capturada. Esta variable reemplazará el valor del hash incorrecto.

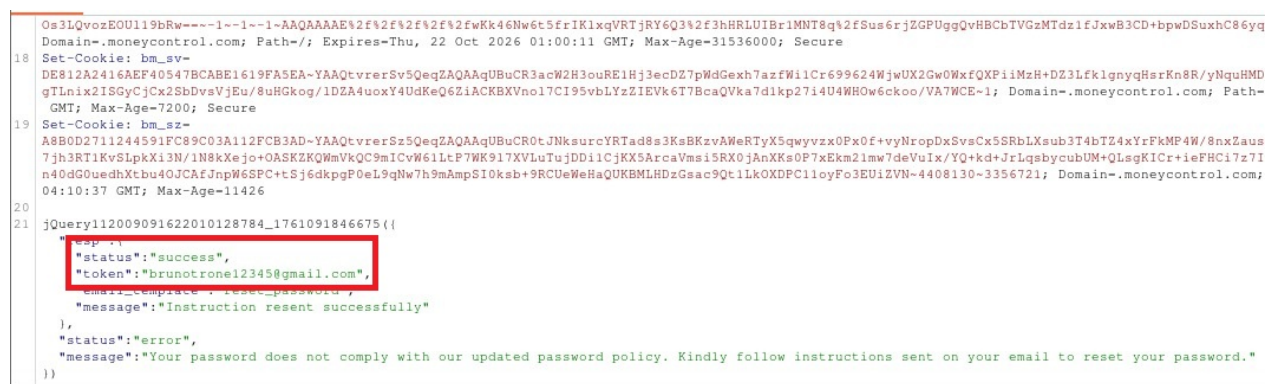


Figura 17: Resultado del cambio del valor del hash. Se aprecia el mensaje de respuesta 'success' en el estado de la solicitud, junto al token con el correo de la sesión.

2.6. Identifica las políticas de privacidad o seguridad

Las políticas de privacidad se pueden encontrar al navegar al footer de la página, o en el enlace [privacypolicy](#). Estas son normas diseñadas para proteger la información, las credenciales (contraseñas, por ejemplo) y los sistemas frente a accesos no autorizados, pérdidas o vulneraciones.

Estas políticas establecen cómo manejar datos sensibles, incluyendo contraseñas, mediante medidas como cifrado, autenticación multifactor y almacenamiento seguro. La política del sitio 'moneycontrol' es una de recopilación y uso de datos personales y de autenticación, la protección técnica y organizativa frente a accesos no autorizados, los derechos de los usuarios sobre sus datos y procedimientos para asegurar la integridad y confidencialidad de contraseñas y OTP (One-Time Password).

2.7. Comente 4 conclusiones sobre la seguridad del sitio escogido

Como resultado del análisis y las pruebas realizadas, se concluye que el sitio no presenta un nivel de seguridad adecuado. En primer lugar, las solicitudes y respuestas no están completamente cifradas, lo que expone información sensible, como correos o credenciales, a posibles interceptaciones. En segundo lugar, el código del sitio no se encuentra ofuscado, facilitando que un atacante pueda identificar funciones críticas o vulnerabilidades en el lado del cliente. Además, la ausencia de mecanismos robustos de autenticación y validación en el servidor incrementa el riesgo de ataques de tipo fuerza bruta o inyección. Finalmente, se recomienda implementar cifrado completo en las comunicaciones, ofuscación del código y mejores prácticas de protección de contraseñas para fortalecer la confidencialidad e integridad de los datos.

2.8. Conclusiones sobre la seguridad del sitio escogido

Como resultado del ataque realizado, se puede ver que las políticas de privacidad y las credenciales no son muy confidenciales ni seguras. Se podría mejorar la seguridad ofuscando las funciones y el código. También se podría cifrar tanto el correo como la solicitud completa (respuesta). Estas soluciones son factibles de aplicar, sin embargo, muchos sitios no optan por motivos económicos o de accesibilidad. Se concluye que el sitio 'moneycontrol' no es seguro y no se recomienda.