Informe Laboratorio 3

Sección 2

Matias Tobar
e-mail: matias.tobar@mail.udp.cl

Octubre de 2025

Índice

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

Para desarrollar las actividades se selecciono la pagina `https://en.webshare.cz/`.

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

Al analizar las peticiónes durante la creación de una nueva cuenta, se identificó que la contraseña es hasheada en el lado del cliente antes de ser enviada.

El algoritmo utilizado es MD5-crypt. Este formato es característico de sistemas Unix/Linux y su estructura es `$1$<salt>$<hash>`. Una característica clave de esta implementación es que el script del lado del cliente genera una salt aleatoria y única para cada intento de registro, lo que fortalece la seguridad del hash almacenado. En la Figura 1 se observa el payload de la petición, donde el campo `password` contiene el hash MD5-crypt.

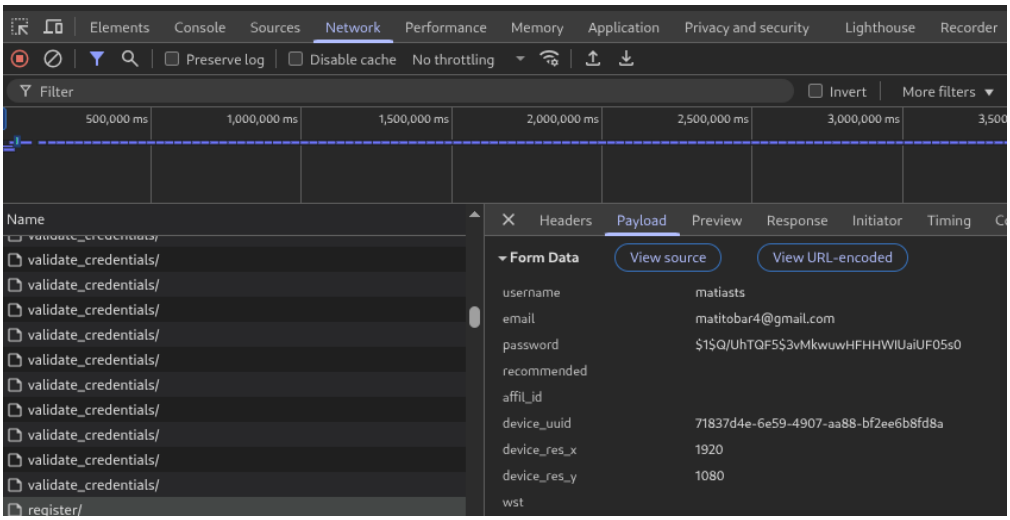

Figura 1: Payload de la petición de registro mostrando el hash MD5-crypt.

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

El algoritmo de hash utilizado para el inicio de sesión es SHA-1 sin sal. Al interceptar la petición POST al endpoint `/api/login/`, se observa que el campo `password` contiene una cadena hexadecimal de 40 caracteres, longitud característica de SHA-1.

Esta implementación es significativamente más débil, ya que al no utilizar una sal, es vulnerable a ataques de diccionario. La Figura 2 muestra el payload de la petición de login con el hash SHA-1 de la contraseña.

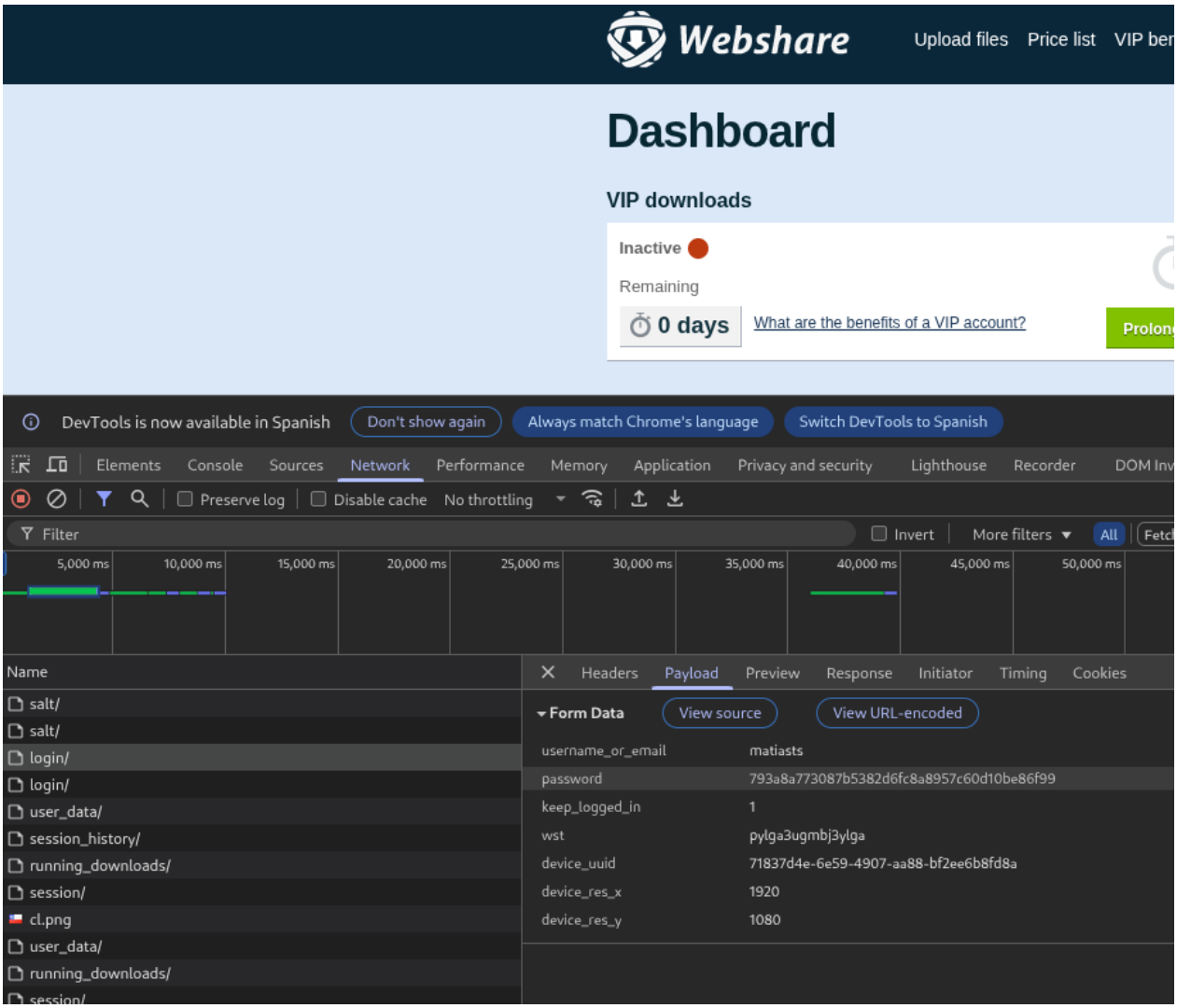

Figura 2: Payload de la petición de login mostrando el hash SHA-1.

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

Para validar se utilizó una función JavaScript que implementa el algoritmo **MD5-crypt**. Al proporcionarle la contraseña de prueba (`"prueba123"`) y la sal capturada en la petición (`"Q/UhTQF5"`), la función generó un hash idéntico al enviado al servidor, como se muestra en la Figura 3.

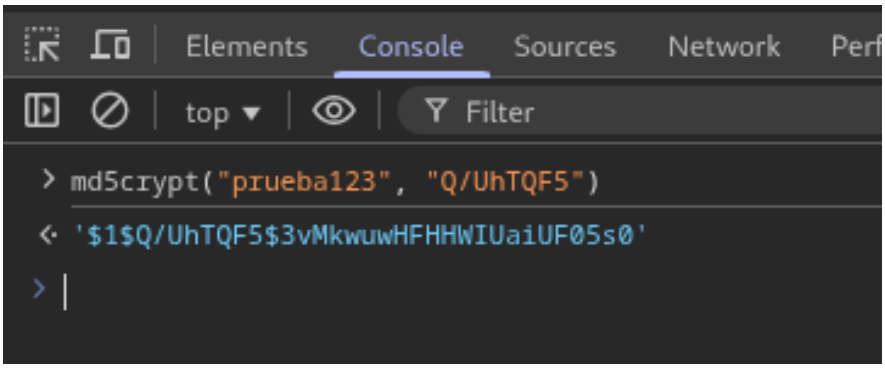

Figura 3: Generación del hash MD5-crypt en la consola.

2.4. Intercepta el tráfico login con BurpSuite

Se utilizó la herramienta Burp Suite para interceptar y analizar el tráfico de red durante el inicio de sesión. La Figura 4 muestra una petición POST al endpoint `/api/login/` capturada por Burp. Se puede observar claramente que el parámetro `password` no contiene la contraseña en texto plano, sino su correspondiente hash SHA-1 (`a15af...`), validando que el hashing ocurre en el cliente.

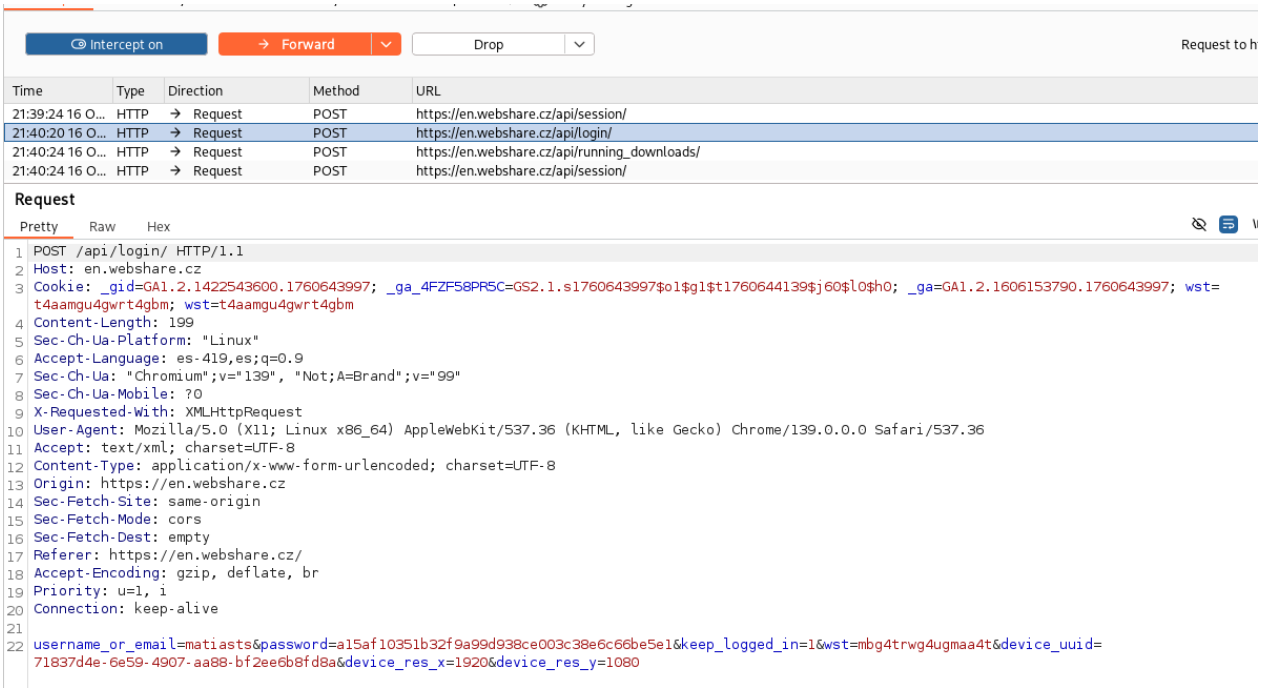

Figura 4: Petición de login interceptada en Burp Suite.

2.5. Realiza el intento de login por medio del hash

Para determinar si el sistema es vulnerable a un ataque de Pass the Hash (PtH). El procedimiento fue el siguiente:

1. Se interceptó una petición de login con una contraseña incorrecta.
2. Se modificó manualmente el hash SHA-1 incorrecto del campo `password` y se reemplazó por el hash SHA-1 correcto de la contraseña de prueba (`793a8...`).
3. Se envió la petición modificada al servidor.

El resultado fue un inicio de sesión exitoso. El servidor validó el hash y concedió acceso a la cuenta, como se muestra en la Figura 6. Esto confirma que el sistema de autenticación es críticamente vulnerable a ataques Pass the Hash. La Figura 5 muestra la petición modificada y enviada a través de Burp.

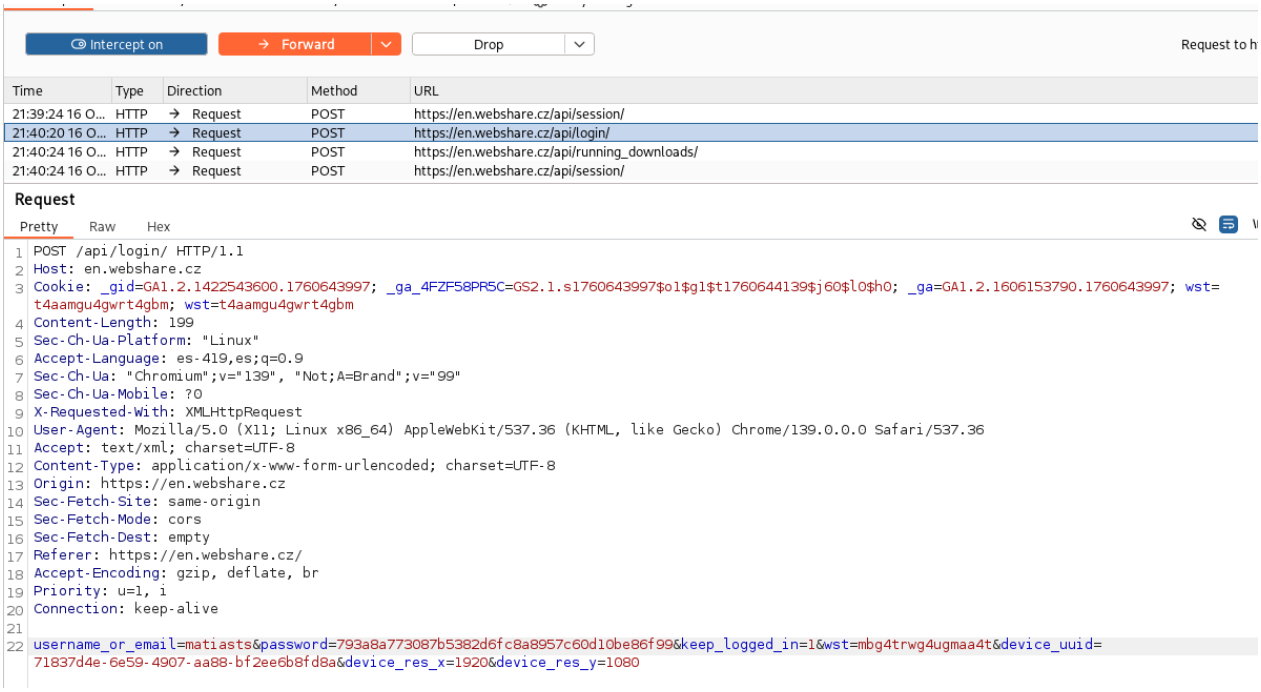

Figura 5: Petición modificada en BurpSuite para el ataque Pass the Hash.

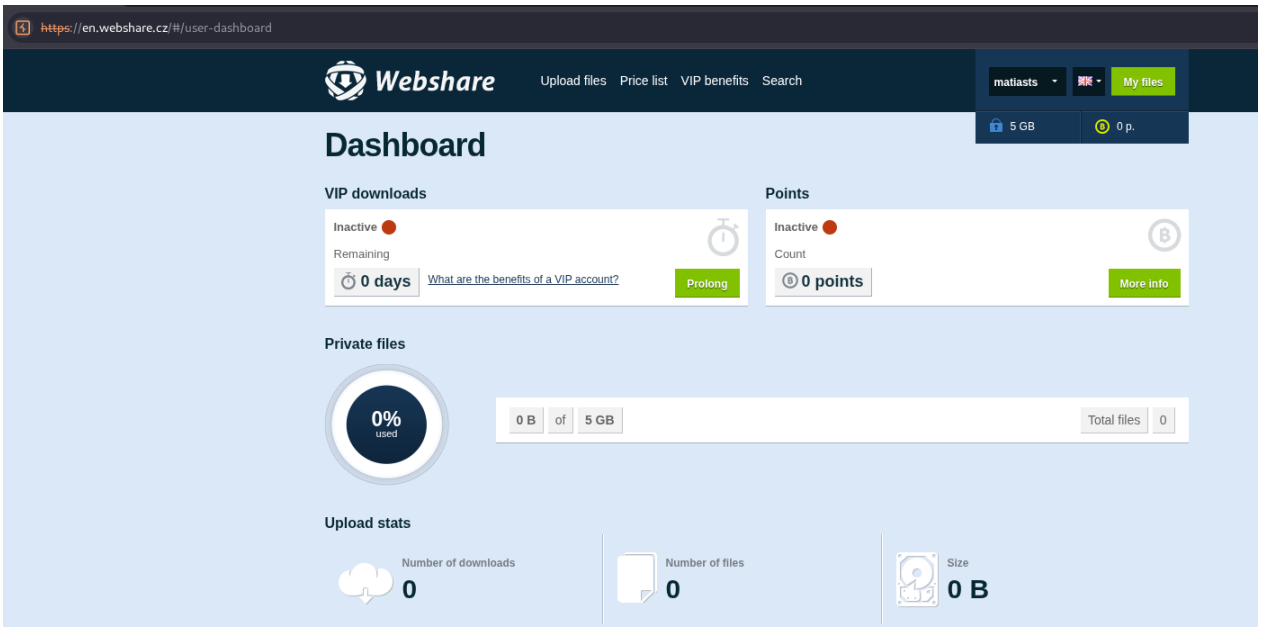

Figura 6: Dashboard del usuario tras un inicio de sesión exitoso.

2.6. Identifica las políticas de privacidad o seguridad

El sitio web auditado, Webshare.cz, dispone de una política de privacidad y protección de datos accesible en su portal. El documento, originalmente en checo, establece los siguientes puntos clave en relación con la seguridad de la información del usuario:

- **Responsable de los datos:** La empresa THINKSMART, s.r.o. es la proveedora del servicio y la responsable de la gestión de la información.
- **Información recopilada:** Recopilan tanto la información proporcionada por el usuario durante el registro (nombre, e-mail) como datos obtenidos automáticamente durante el uso del servicio (dirección IP, tipo de navegador, cookies, datos de comportamiento y localización).
- **Medidas de seguridad:** El proveedor declara que protege la información del usuario en la medida del desarrollo técnico, utilizando "tecnologías de cifrado modernas.[en] una base de datos segura. Sin embargo, explícitamente **no se hace responsable** de intervenciones no autorizadas de terceros que puedan resultar en un acceso ilícito a los datos.
- **Uso de los datos:** La información se utiliza para la prestación y mejora de los servicios, así como para fines de marketing y comerciales, tanto propios como de terceros, previo consentimiento del usuario.

- **Consentimiento:** El usuario otorga su consentimiento para el tratamiento de sus datos personales marcando una casilla en el formulario de registro. Este consentimiento es voluntario y puede ser revocado en cualquier momento.

El enlace a la página principal desde donde se accede a estas políticas es: `https://en.webshare.cz/`.

2.7. Comente 4 conclusiones sobre la seguridad del sitio escogido

Basado en el análisis realizado, se extraen las siguientes conclusiones sobre la seguridad del sitio Webshare.cz:

1. **Vulnerabilidad Crítica a Pass the Hash (PtH):** La conclusión más importante es que el mecanismo de inicio de sesión es **altamente vulnerable a ataques de Pass the Hash**. Un atacante que logre interceptar el hash SHA-1 de la contraseña (por ejemplo, a través de un ataque Man-in-the-Middle en una red no segura o accediendo a la memoria del navegador) puede autenticarse exitosamente sin necesidad de conocer la contraseña real.

2. **Uso de un Algoritmo Criptográficamente Roto:** El uso de SHA-1 para la autenticación es una práctica obsoleta y peligrosa. SHA-1 se considera criptográficamente roto desde 2017 debido a la existencia de ataques de colisión prácticos. Su uso en un sistema de autenticación moderno representa un riesgo de seguridad.

3. **Inconsistencia en las Prácticas de Seguridad:** Existe una notable inconsistencia entre la seguridad del registro y la del inicio de sesión. Mientras que para el registro se implementa un mecanismo robusto y seguro (MD5-crypt con sal dinámica), para la autenticación se recurre a un método débil e inseguro (SHA-1 sin sal). Esta disparidad crea un punto débil que compromete la seguridad general de la cuenta.

4. **Falsa Sensación de Seguridad por Hashing en el Cliente:** Aunque hashear la contraseña en el cliente puede parecer una medida de seguridad adicional, su implementación en este caso es defectuosa. Al no utilizar un desafío del servidor (nonce) o un protocolo seguro como SRP, el envío de un hash predecible (SHA-1) no ofrece protección real contra la intercepción y reutilización de credenciales, generando una falsa sensación de seguridad.

- Repositorio de github: https://github.com/matiasts4/Lab-Criptografia