SAD - U3.1. Password Attack

Descargar estos apuntes

Índice

1. Almacenamiento de contraseñas

Una de las medidas más importantes de protección que implementa todo sistema operativo es la autenticación. Esta medida implica necesariamente el uso y almacenamiento de contraseñas.

> **Contraseñas resumidas, no cifradas**
>
> Aunque coloquialmente se hable de contraseñas cifradas, en realidad **las contraseñas no se cifran**, sino que se resumen.
>
> El resumen de una contraseña es un valor obtenido a partir de la contraseña original mediante un algoritmo de resumen. Este valor se almacena en el sistema y se compara con el resumen de la contraseña que proporciona el usuario al intentar autenticarse. Si los dos resúmenes coinciden, el usuario se autentica correctamente.

1.1. Contraseñas en Windows

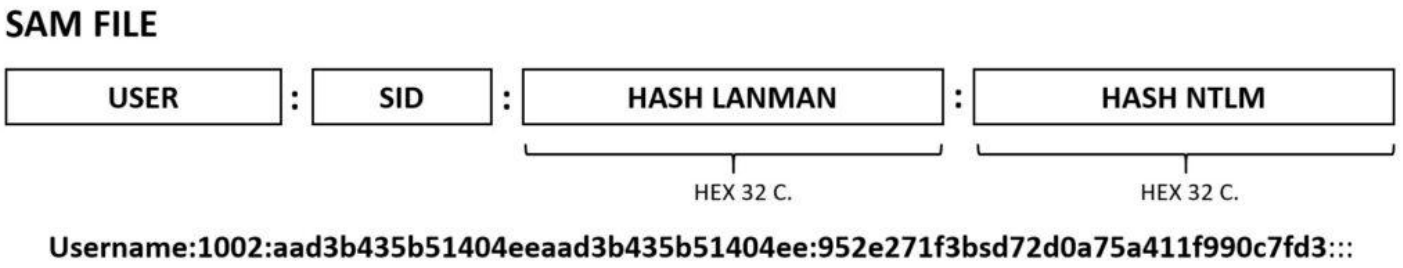
En **Windows** las contraseñas de los usuarios locales se almacenan en el fichero **SAM (Security Account Manager)**. Podemos encontrar el fichero SAM en `windows/system32/config/SAM` o en `winnt/system32/config/SAM`. Como medida de protección, **el fichero SAM al completo está cifrado** y además, cuando se está ejecutando Windows también está bloqueado.

El formato de fichero SAM actual se utiliza desde Windows 7 y sigue vigente hasta la actualidad (actualmente la última versión es Windows 11). Contiene los siguientes datos (una fila para cada cuenta de usuario):

- **User**: Contiene el nombre del usuario de cada cuenta. Este incluye las filas para los usuarios que crea Windows durante la instalación: Administrador, Invitado, etc.

- **SID**: Se corresponde a los últimos dígitos de los códigos de Identificadores de Seguridad. Estos identificadores son asignados de forma unívoca a un usuario y son utilizados para asignar permisos de acceso a recursos y objetos del sistema. Un ejemplo de SID podría ser S-1-5-21-1234567890-1234567890-1234567890-500, siendo los últimos dígitos los que se pueden ver en el contenido del SAM para cada usuario. Normalmente los dígitos de 500, 501, etc. son para los usuarios que el Sistema Operativo crea por defecto (500 para el Administrador), mientras que a partir de 1000 es para los usuarios creados por una persona (el 1000 suele ser para el primer usuario creado).
- **Hash LANMAN**: Se corresponde con el hash del LAN manager. Pertenece a un método de encriptación anterior a NTLM (ver abajo) y se conserva para permitir la compatibilidad entre sistemas heredados. Se puede considerar un valor obsoleto (no se puede considerar que sea la contraseña hash y no hay nada que obtener de él).
- **Hash NTLM**: Contiene el hash de la contraseña del usuario y se considera el sucesor del sistema LANMAN. Una vez obtenido el fichero SAM, estos hashes son los que una vez descifrados permiten obtener la contraseña original. Está formado por 32 caracteres en sistema hexadecimal, al igual que el hash LANMAN.

A parte del fichero SAM, en algunas técnicas para la obtención de los hashes también se suele utilizar el fichero `SYSTEM`. Este fichero contiene información sobre la configuración del sistema y es necesario para descifrar los hashes del fichero SAM. Se encuentra en la misma carpeta que el fichero SAM.



Ficheros bloqueados

El fichero SAM está bloqueado mientras Windows está en ejecución.

Estos ficheros se hallan en el directorio del volumen principal C:/Windows/system32/config y **se deniega cualquier tipo de interacción con ellos** pues su alteración podría suponer una grave afectación al sistema. Es por ello que se necesitan métodos especiales para obtener su información, ya sea mediante la `extracción con discos de arranque` o con `herramientas especializadas` que se indican a continuación.

Primero se debe obtener una copia de estos ficheros y luego se puede proceder a su análisis.

Para obtener un volcado de la memoria RAM de un sistema en ejecución se puede utilizar la herramienta Mimikatz, pwdump o Impacket.

- Mimikatz permite obtener los hashes de las contraseñas almacenadas en la memoria RAM del sistema, lo que puede ser útil para realizar ataques de fuerza bruta o para obtener las contraseñas de los usuarios del sistema.

```
# Mimikatz
C:\mimikatz.exe
lsadump::sam /system:C:\Windows\system32\config\SYSTEM /sam:C:\Windows\system32\config\SAM
```

- pwdump es una herramienta que permite extraer los hashes de las contraseñas almacenadas en el fichero SAM de un sistema Windows.

```
# pwdump
C:\pwdump8.exe -sam C:\Windows\system32\config\SAM -system C:\Windows\system32\config\SYSTEM
```

- Impacket es una colección de herramientas que permiten interactuar con sistemas Windows y realizar ataques.

```
# Extracción del Registro en Windows
C:\reg save hklm\sam C:\sam
C:\reg save hklm\system C:\system
# Uso de Impacket en Kali Linux
$> impacket-secretsdump -sam <fichero_sam> -system <fichero_system> LOCAL
```

Una vez obtenidos los hashes de las contraseñas del archivo SAM, es necesario utilizar herramientas especializadas como **SamInside**, **Ophcrack** o **John The Ripper** para atacar las contraseñas y recuperar las contraseñas originales.

Estas herramientas permiten recuperar las contraseñas almacenadas en el fichero SAM mediante **ataques de fuerza bruta**, **ataques de diccionario** o bien mediante el uso de **tablas rainbow**.

Es posible que en sistemas Windows Server se utilice una política de seguridad que impida la extracción de los hashes de las contraseñas. En estos casos, es necesario utilizar técnicas de **escalada de privilegios** para obtener los permisos necesarios para extraer los hashes de las contraseñas.

1.2. Rainbow Tables

El método de las `Rainbow Tables o Tablas Arcoíris` consiste en emplear unas tablas especiales que contienen pares de contraseñas y hashes precalculados que se denominan cadenas arcoíris. El sistema consiste en comprobar si el hash objetivo forma parte de alguna de estas cadenas, es decir, comprobar si el hash está en la tabla.

Este método podría parecer muy simple si no fuera porqué el formato de las tablas rainbow no es en texto plano, y tanto para su creación como para su sistema de búsqueda (optimizado) se emplean algoritmos especiales que nada tienen que ver con el método de fuerza bruta.

Para conocer más sobre su contenido así como para ver la explicación del algoritmo se recomienda las siguientes páginas:

- Fichero Rainbow Tables
- Funcionamiento creación y algoritmo Rainbow Tables

Como la comprobación se efectúa directamente hash objetivo contra hash de la tabla la velocidad es muchísimo mayor que con la fuerza bruta. Si el hash de la contraseña objetivo se halla en la tabla arcoíris es posible obtener el resultado en unos pocos minutos u horas.

La herramienta por antonomasia para hacer tareas con las Rainbow Tables es **Rainbowcrack**, aunque en realidad se compone de varias herramientas. Otra herramienta que nos permite trabajar con tablas arcoíris, de forma gráfica, es **Ophcrack**.

Las tablas arcoíris se pueden generar en Kali Linux con una herramienta especial (rtgen), aunque lo recomendando es buscarlas y descargarlas de Internet. Las tablas arcoiris se pueden encontrar en diferentes formatos y tamaños, dependiendo de la longitud de las contraseñas y del algoritmo de cifrado utilizado.

Tablas arcoíris

> Las tablas arcoíris ocupan mucho espacio en disco, por lo que es necesario disponer de un disco duro con suficiente capacidad para almacenarlas.

Algunas páginas donde se pueden encontrar y descargar tablas arcoíris son:

- Tablas para Ophcrack
- Tablas del proyecto Rainbowcrack

1.3. Contraseñas en Linux

En **Linux** un par de ficheros se encargan de la gestión de usuarios y contraseñas del sistema.

- `/etc/passwd`: Es el más importante, es donde se registran los usuarios del sistema.
- `/etc/shadow`: Almacena las contraseñas cifradas. **El único usuario con permiso de lectura de este fichero es el root**.

Cada línea de `/etc/passwd` almacena información sobre un usuario del sistema. Estas líneas tienen una estructura similar a la que se muestra a continuación:

```
usuario:x:1000:1000:NombreUsuario,DescripciónUsuario,:/home/usuario:/bin/bash
```

- **usuario**: Nombre del usuario.
- **x**: Contraseña cifrada del usuario. La contraseña cifrada se almacena en `/etc/shadow`.
- **1000**: Identificador de usuario (UID).
- **1000**: Identificador de grupo (GID).
- **NombreUsuario**: Nombre completo del usuario.
- **DescripciónUsuario**: Descripción del usuario.
- **/home/usuario**: Directorio personal del usuario.
- **/bin/bash**: Shell por defecto del usuario.

En `/etc/shadow` se almacenan las contraseñas cifradas de los usuarios. Cada línea de `/etc/shadow` almacena información sobre un usuario del sistema. Estas líneas tienen una estructura similar a la que se muestra a continuación:

```
usuario:HashContraseña:FechaCambio:MinDíasCambio:MaxDíasCambio:DíasAvisoCaducidad:DesactivarCuenta:FechaCaducidad:Res
alumno:$6$52450745$k5ka2p8bFuSmoVT1tzOyyuaREkkKBcCNqoDKzYiJL9RaE8yMnPgh2XzzF0NDrUhgrcLwg78xs1w5pJiypEdFX/:18600:0:999
```

- **usuario (alumno)**: Nombre del usuario.
- **HashContraseña (6)**: Contraseña cifrada del usuario.
- **FechaCambio (18600)**: Último cambio de contraseña. Días transcurridos desde el 1 de enero de 1970.
- **MinDíasCambio (0)**: Mínimo número de días hasta que se permita el cambio. Mínimo número de días que tienes que pasar entre cambios de contraseña.
- **MaxDíasCambio (99999)**: Máximo número de días hasta que se exija un cambio de contraseña. Máximo número de días de validez de la contraseña.
- **DíasAvisoCaducidad (7)**: Número de días antes de la caducidad para avisar al usuario.
- **DesactivarCuenta ()**: Número de días antes de desactivar la cuenta. Días que, tras caducar la contraseña, se espera para deshabilitar la cuenta.
- **FechaCaducidad ()**: Fecha de caducidad de la contraseña. Días transcurridos desde el 1 de enero de 1970 hasta que se desactiva la cuenta.

- **Reserva (0)**: Campo reservado para futuras ampliaciones.

Los primeros caracteres del hash de la contraseña indican el algoritmo de cifrado utilizado. Los algoritmos de cifrado más comunes se pueden consultar en la siguiente dirección: Algoritmos de cifrado de contraseñas en Linux.

En Kali Linux hay una herramienta, **hashid**, que permite identificar el algoritmo de cifrado utilizado en un hash de contraseña. La herramienta nos proporciona información sobre el algoritmo de cifrado, la longitud del hash y el tipo de hash, además de indicarnos el **modo de uso** que se debe usar en `hashcat` para crackear la contraseña o el **formato del hash** para usarlo con `John The Ripper`.

```
hashid -m "HashContraseña"
```

1.4. Listas de contraseñas

Igual que en el caso de las tablas arcoíris, para realizar ataques de fuerza bruta o de diccionario es necesario disponer de listas de contraseñas. Estas listas contienen contraseñas comunes, contraseñas filtradas de bases de datos robadas, contraseñas de diccionario, etc.

Las listas de contraseñas se pueden encontrar en diferentes formatos y tamaños, dependiendo de la longitud de las contraseñas y del tipo de ataque que se quiera realizar. Algunas de las listas de contraseñas más conocidas son:

- RockYou 2024
- Awesome Wordlists
- Wordlists
- Openwall Wordlists
- Diccionarios de palabras de diferentes idiomas (Palabras RAE).

Las listas de contraseñas se pueden utilizar con herramientas como **John The Ripper** o **Hashcat** para realizar ataques de fuerza bruta o de diccionario.

2. Herramientas de ataques a contraseña

Aunque lo más habitual es usar algunas de las herramientas más conocidas para realizar ataques a contraseñas, también es posible realizar estos ataques de forma manual. A continuación se describen algunas de las herramientas más utilizadas para realizar ataques a contraseñas.

> **CrackStation**
>
> CrackStation es un servicio en línea que proporciona una base de datos de hash de contraseñas para ayudar a los usuarios a recuperar contraseñas olvidadas o perdidas.
> La base de datos de CrackStation contiene una gran cantidad de hashes de contraseñas previamente descifrados y almacenados en una base de datos de búsqueda rápida. Los usuarios pueden ingresar un hash de contraseña en la base de datos de CrackStation y la herramienta buscará la contraseña correspondiente en texto plano.

2.1. John The Ripper

2.2 Hashcat