Informe Pericial 1.1: Volcado y Análisis de Memoria del Proceso MEMPASS

Tarea 1.1 para la asignatura Auditoría Informática II

Autor:
Arturo Rivero Dutra

Fecha:
07/02/2026

Índice

Identificación

Caso: Actividad sospechosa en el sistema informático por parte de un usuario

Código identificador: UNI_1_1

Responsable del informe: Arturo Rivero Dutra

Entidad que solicitó el análisis: Universidad Complutense de Madrid

Persona a la que va dirigido el informe: María Inmaculada Pardines Lence

Fecha de emisión del informe: 7 de febrero de 2026

Antecedentes

La entidad que solicitó el análisis forense descrito en el presente informe detectó actividad sospechosa por parte de un usuario con acceso legítimo al sistema.

La entidad explica que en su política de generación de contraseñas seguras, todos sus usuarios han de comenzar sus contraseñas con la fecha de creación de dicha contraseña de acuerdo al siguiente formato: *YYYYMMDD*. También explica que a principios del presente año, 2026, se forzó a todos los usuarios a actualizar su contraseña a una nueva de acuerdo a la política de seguridad.

Este informe explica los hallazgos del análisis forense realizado tras un presunto acceso al sistema por parte del usuario.

Objetivos

- Adquirir el área de memoria de un proceso.
- Analizar la información del área de memoria de un proceso.

Análisis del Laboratorio Forense

El presente análisis forense ha sido realizado en un laboratorio consistente en una misma Máquina Virtual en la cual se ha generado, adquirido y analizado la evidencia digital.

Sistema Operativo (VM): Windows 11 25H2
Nivel de Parcheado/Build: Build 26200.6584
Arquitectura: x64, 12GB RAM, 30GB Disco

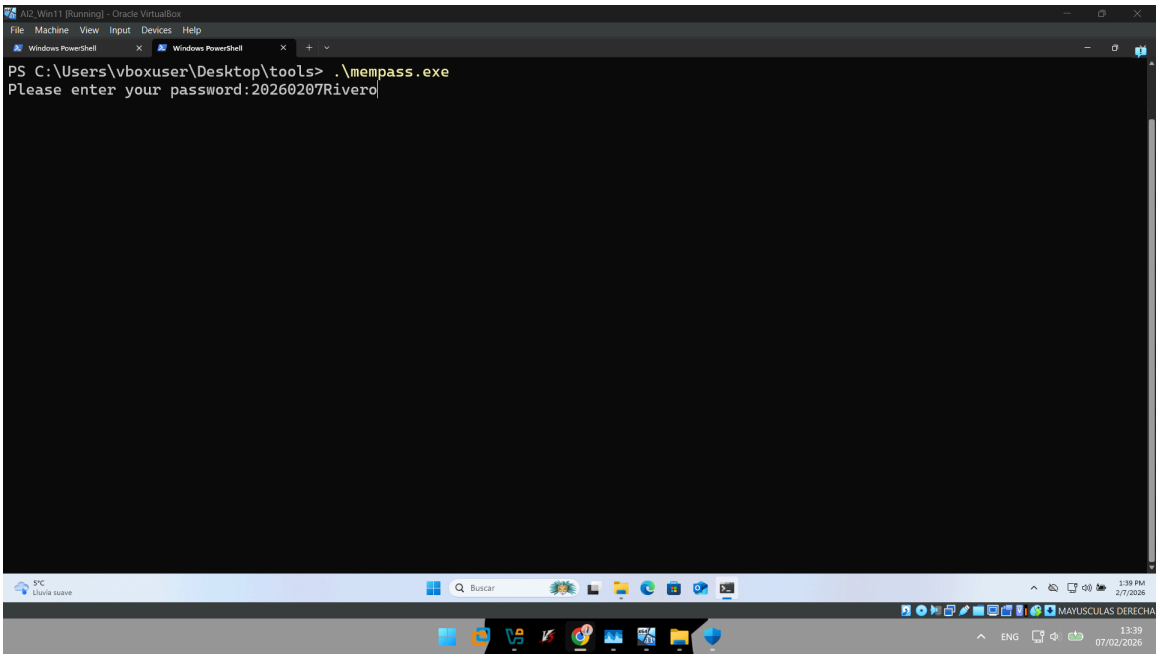
Análisis detallado del escenario:

- Instalación del Sistema Operativo en la máquina virtual completamente limpia.
- Windows Defender completamente desactivado.
- Máquina con acceso a internet por ethernet.

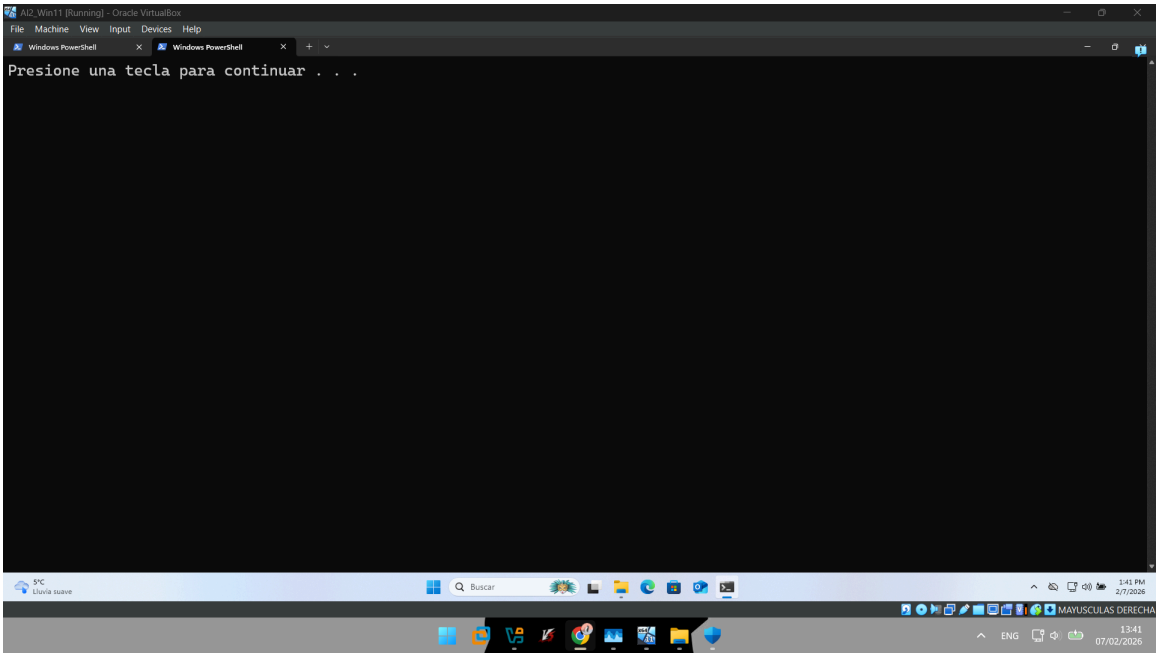
Ciclo de Vida de la Evidencia Digital

Generación de la Evidencia

Con tal de simular el uso de las credenciales del usuario, se ejecutó el programa mempass.exe, el cual indicaba que se introdujera una contraseña.



Una vez se introdujo la contraseña y se presionó enter, el programa insertó la cadena de texto en su región de memoria correspondiente y procedió a pausarse.

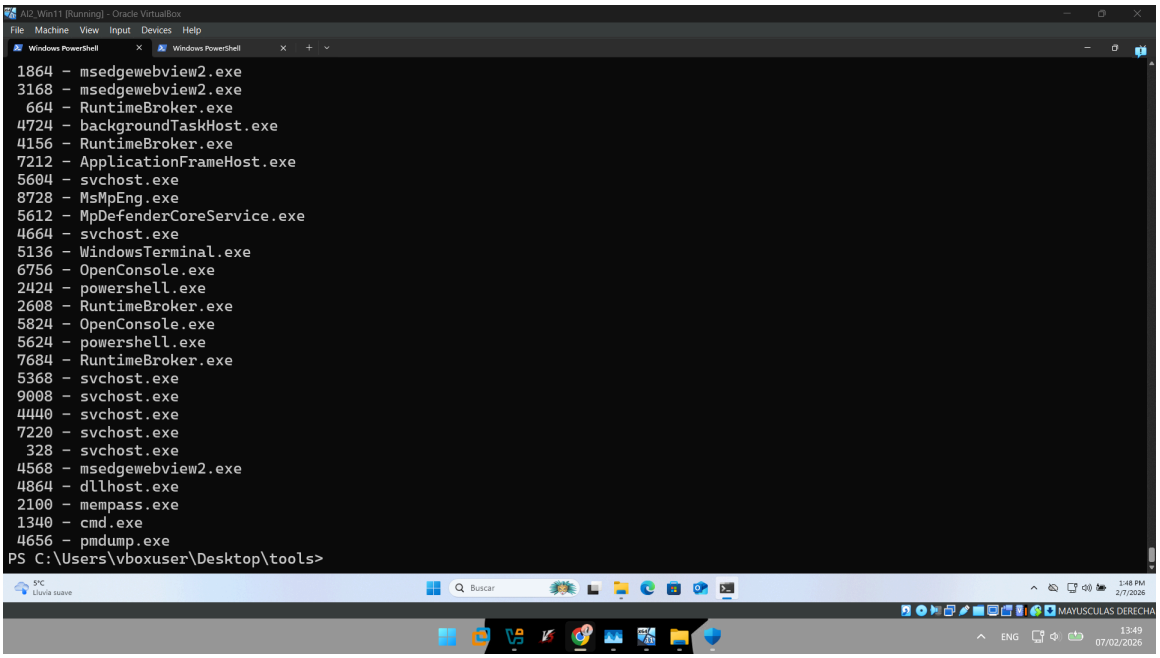


Se dejó el proceso pausado para mantenerlo en memoria y se abrió una nueva terminal desde donde trabajar.

Adquisición de la Evidencia

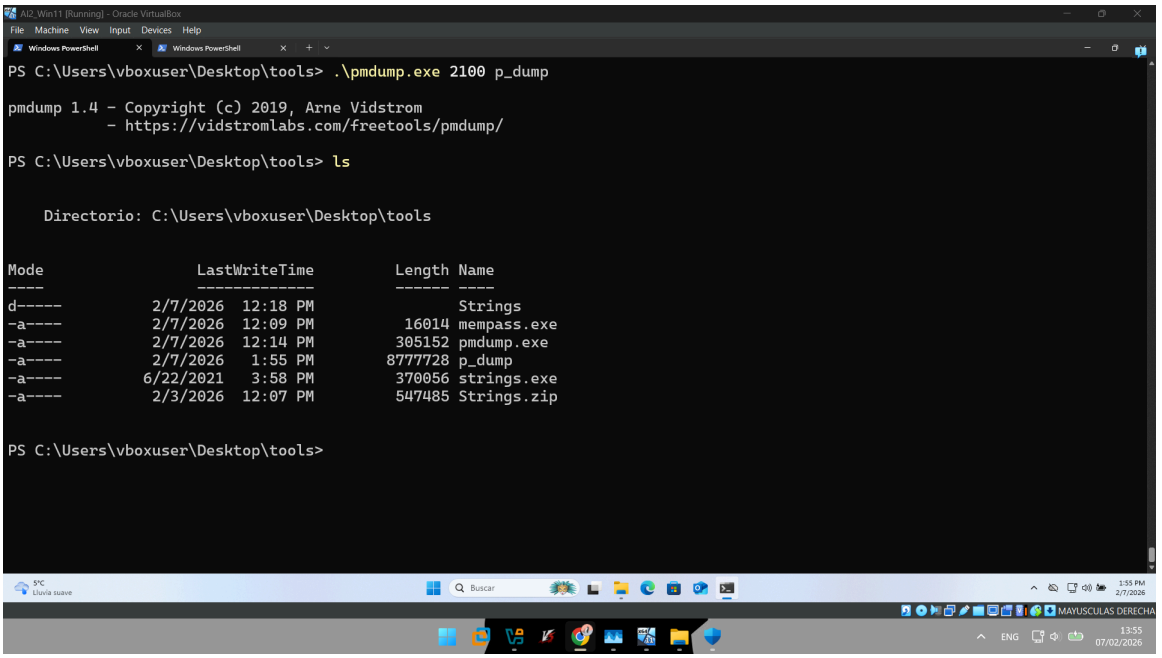
En esta segunda fase se procedió a identificar el *pid* del proceso de *mempass.exe* con el programa [pmdump.exe]:

> *.\pmdump.exe -list*

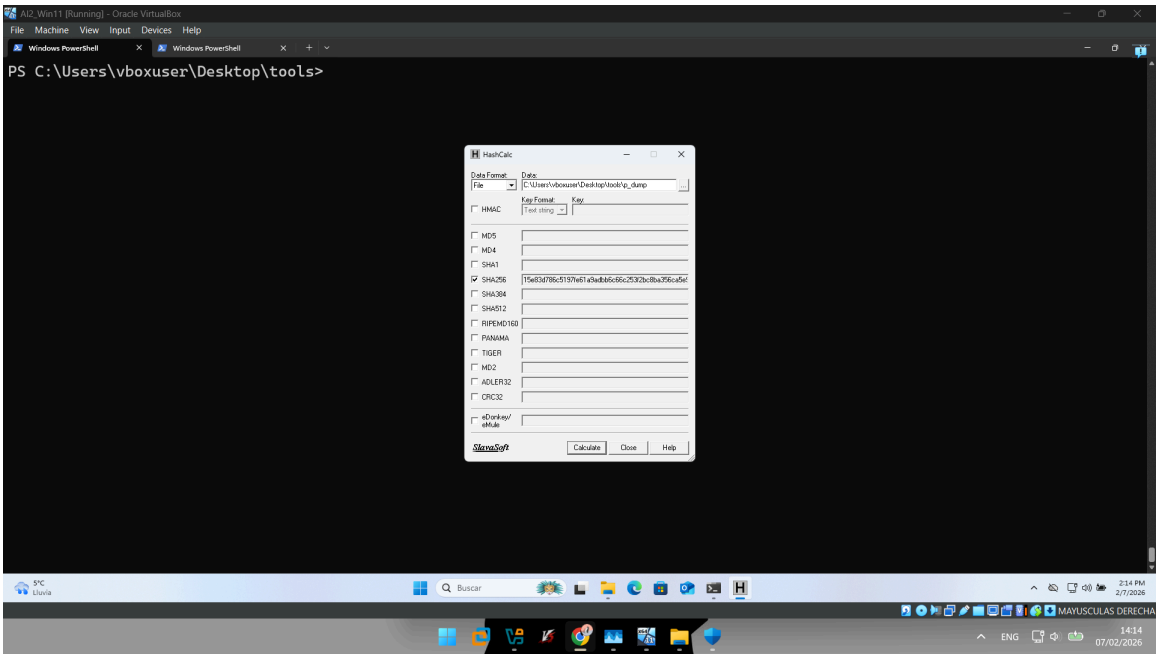


Se descubrió que el *pid* de *mempass.exe* era 2100. Una vez se obtuvo esa información, se pudo continuar con el volcado de memoria de ese proceso en concreto obteniendo así la evidencia digital que fue nombrada como *p_dump*:

> *.\pmdump.exe 2100 p_dump*



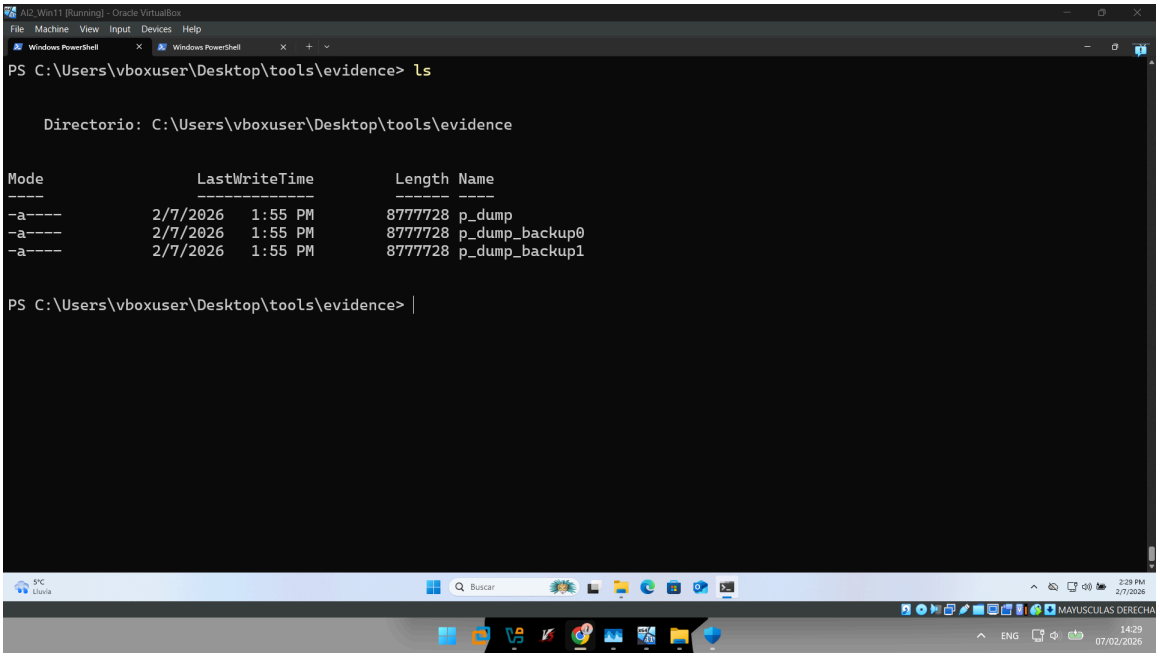
Acto seguido, se procedió a realizar el *hash SHA-256* de la evidencia obtenida para capturar la integridad de la evidencia original. Para ello, se empleó el programa HashCalc:



Obteniendo así el siguiente *hash*:

15e83d786c5197fe61a9adbb6c66c253f2bc8ba356ca5e9e382ce1333734babe

Después, con tal de asegurar la calidad en el análisis forense cumpliendo con las buenas prácticas descritas en la *ISO/IEC 27037*, se procedió a realizar una primera copia de la evidencia bajo el nombre de *p_dump_backup0* la cual fue debidamente resguardada en un entorno aislado. Posteriormente se realizó una segunda copia nombrada *p_dump_backup1* en la cual fue donde se procedió a continuación con el análisis.



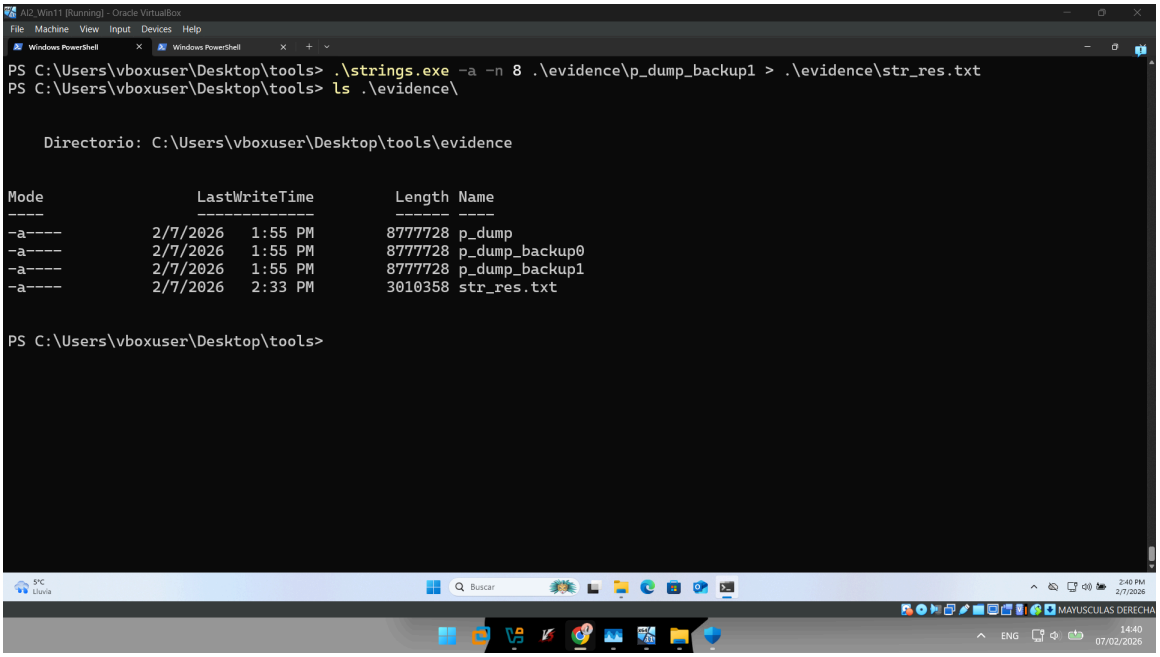
Análisis de la Evidencia

Se comenzó el análisis haciendo uso de la herramienta strings.exe mediante la cual se pudo *parsear* el volcado de memoria ya mencionado conforme a los siguientes parámetros:

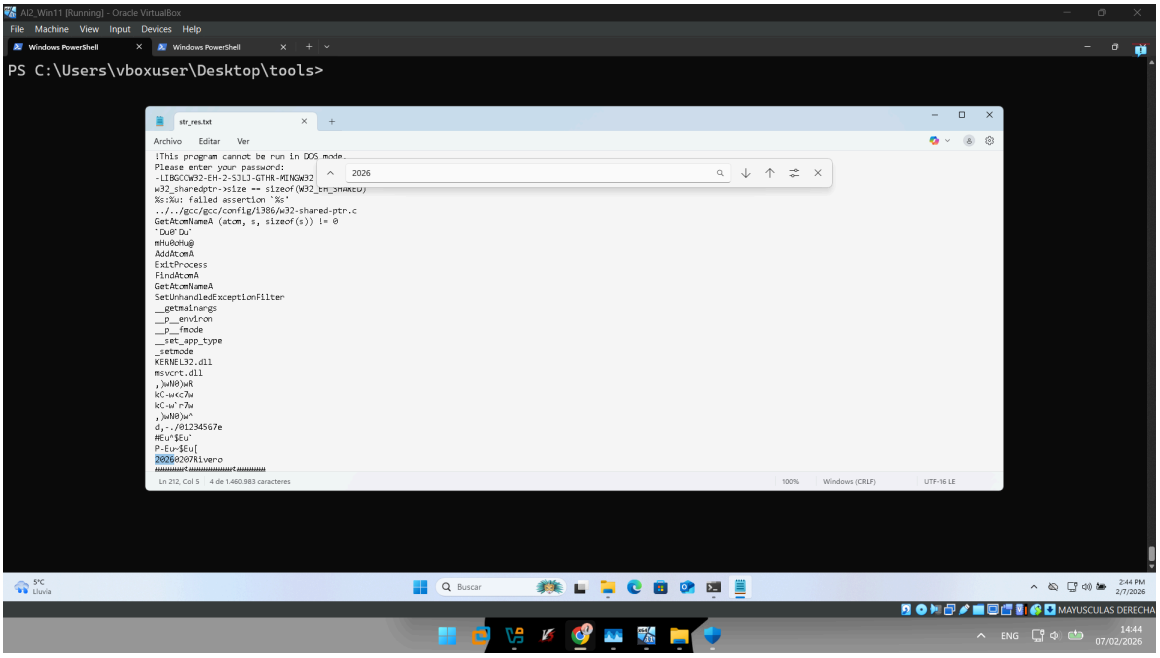
- *-a*: Para encontrar cadenas ASCII.
- *-n 8*: Para filtrar por cadenas de longitud de 8 caracteres como mínimo ya que ya se sabía a priori que la contraseña a encontrar debía contener una fecha en el formato *YYYYMMDD*.
- *str_res.txt*: El nombre del fichero con la salida del programa.

El comando exacto utilizado fue:

> .\strings.exe -a -n 8 .\evidence\p_dump_backup1 > .\evidence\str_res.txt



Por último, se abrió la salida *str_res.txt* con el bloc de notas y se buscó la cadena de texto *2026* dado que sabíamos a priori que el formato de la contraseña contenía la fecha de su creación y que ésta fue creada en el año 2026:



Así, se reveló la información buscada, la cadena de texto que representaba la contraseña:

20260207Rivero

Descripción de la Evidencia Capturada

Relación detallada del elemento capturado:

Nombre Evidencia	Hash SHA-256	Fecha Adquisición	Analista	Fecha Análisis
p_dump	15e83d786c5197fe61a9adbb6c66c253f2bc8ba356ca5e9e382ce1333734babe	07/02/2026	Arturo Rivero	07/02/2026

Anotación: El análisis se realizó sobre la copia de trabajo p_dump_backup1, verificada como copia de p_dump.

Herramientas Utilizadas

En esta sección se recopila el glosario de las herramientas utilizadas junto a una breve descripción de su funcionalidad:

- mempass.exe: Utilidad con interfaz de línea de comandos la cual recibe como entrada una cadena de texto por parte del usuario que representa una contraseña y la introduce en el mapa de memoria de su proceso.
- pmdump.exe: Utilidad con interfaz de línea de comandos que realiza un volcado de la región de memoria asignada a un proceso.
- HashCalc: Utilidad con interfaz gráfica de usuario que permite computar el *hash* para una entrada sea esta una cadena de texto o un archivo con una selección de algoritmos disponibles.
- Strings.exe: Utilidad con interfaz de línea de comandos que permite buscar cadenas de caracteres válidas en un archivo binario siguiendo diferentes patrones de búsqueda.

Conclusiones

Como se ha demostrado en el análisis forense descrito en el presente informe, es posible recuperar información tal como cadenas de texto de especial relevancia como puede ser una contraseña de la región de memoria de un proceso. Esto puede ser especialmente útil ya que como se ha visto anteriormente, muchas veces, esta información no se encuentra cifrada y por tanto puede ser fácilmente identificada por un programa que encuentre cadenas de texto válidas como *strings.exe*. Un ejemplo de programa que podría manejar este tipo de información sensible en un entorno realista sería el navegador, que comúnmente gestiona credenciales de acceso del usuario.

Por último, queda de manifiesto la vital importancia de capturar eficazmente evidencias digitales que se encuentran en entornos volátiles, como puede ser la memoria RAM, ya que con frecuencia son de especial relevancia y su naturaleza volátil puede destruirlas con facilidad; y, precisamente por ello, se han de tomar las precauciones adecuadas para su preservación una vez capturada, a saber, realizar una copia de seguridad primera que sea correctamente resguardada y una segunda copia de seguridad en la cual es donde se realiza el análisis que potencialmente puede afectar la integridad de la evidencia.

Firmado digitalmente por:

Arturo Rivero Dutra,
a 7 de febrero de 2026