Hispasec]

DumpShark

Writeup UAM

Noviembre 2023

Informe técnico

Cláusula legal

Documentación

Informe técnico

Índice

1. Datos del reto.

1.1. Especificaciones técnicas.

A continuación se detalla el alcance y el conjunto de especificaciones del reto, así como las normativas de aplicación.

Alcance	
Activo/s:	**captura.pcap**
Categoría:	Forense
Fecha de inicio:	23/11/2023.
Fecha de fin:	07/12/2023.
Cumplimiento normativo:	Las contraseñas y otra información sobre los usuarios no son almacenadas en cumplimiento con la ley de protección de datos (Reglamento General de Protección de Datos, RGPD). Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales.

2. Proceso de resolución del reto.

A continuación se detalla la manera intencionada en la que se espera que el usuario resuelva el reto. Cada entrada viene encabezada por la descripción detallada del punto en concreto, acompañado de los pasos a seguir para reproducirlo.

2.1. Flag 1

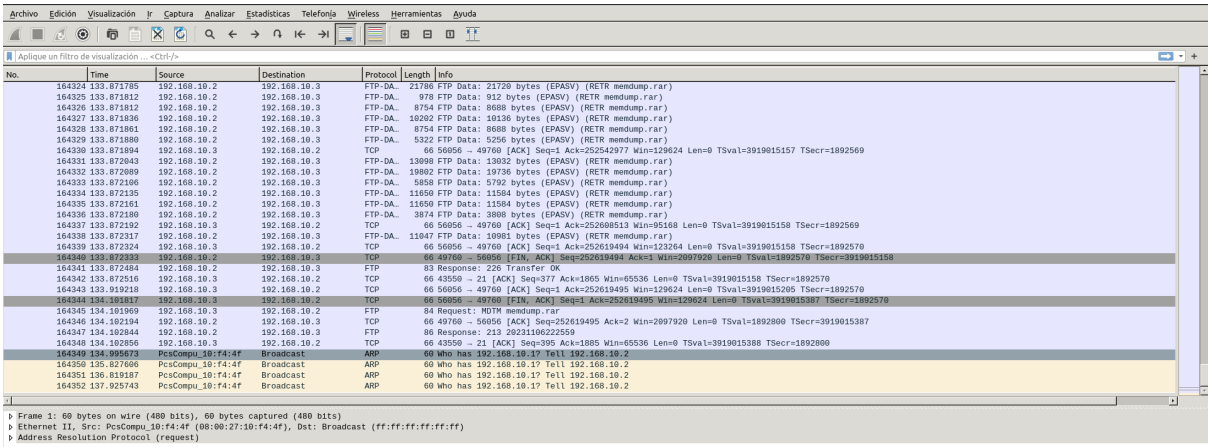
Descripción

Hemos detectado que un atacante ha conseguido vulnerar una de nuestras máquinas y, como investigador, tu misión es descubrir cuáles son los pasos que el atacante ha ido realizando de modo que finalmente podamos conocer cuál es el ataque que utilizó para poder descargar una captura de memoria que teníamos almacenada en nuestro servidor, el cual se accedía desde otro equipo para realizar una copia de seguridad de la captura de memoria. Por suerte hemos podido obtener una captura de red de los pasos que ha ido realizando el atacante, solo debes analizarla.

- Formato de la flag: UAM{SiglasVulnerabilidadUtilizada_MD5SUMdel.mem}
    - Ejemplo de flag: UAM{RCE_075372c956b6106c1fd8bca837400974}

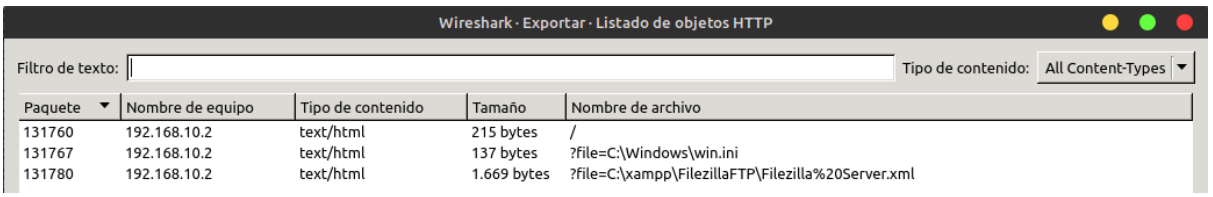
Resolución

Al abrir el archivo de captura de red con la herramienta gráfica "Wireshark" vemos que contiene gran cantidad de paquetes (**164352** en concreto).



En ella podemos ver que hay peticiones tanto HTTP, como TCP, ARP o, la que más nos interesa, **FTP-DATA**.

Primero podemos intentar volcar los archivos con la utilidad de Wireshark "**Exportar objetos**", dentro de la pestaña "**Archivo**", pero no aparecerá el FTP-DATA, aunque sí podremos ver las peticiones HTTP que contiene la captura.



En esta podremos ver la primera parte de la flag, ya que el *endpoint* de la petición "**/?file=C:\Windows\win.ini**" nos indica que está explotando una vulnerabilidad de *Local File Inclusion* (**LFI**).

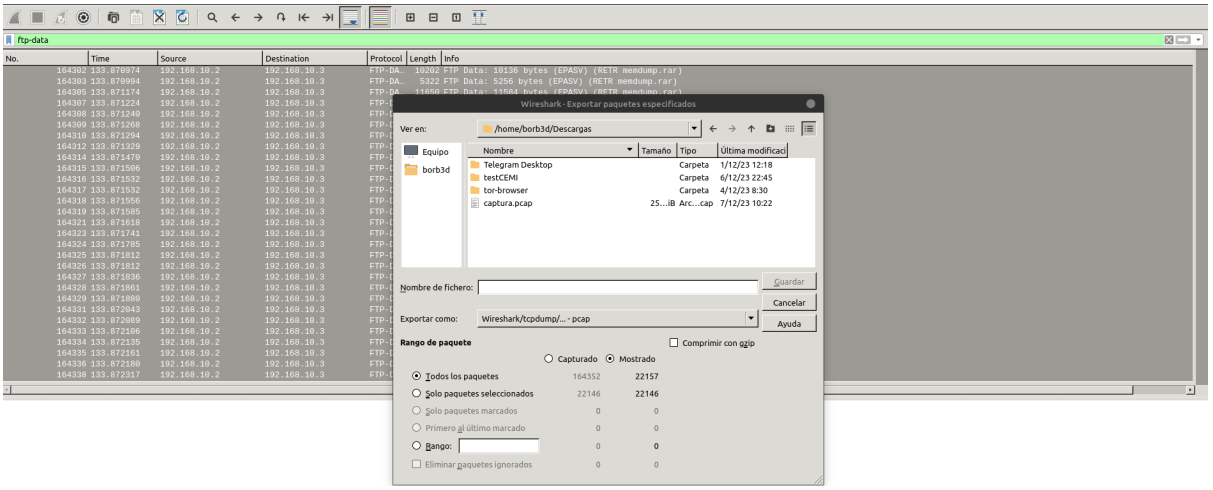
Wireshark · Seguir flujo HTTP (tcp.stream eq 131097) · captura.pcap

```
GET /?file=C:\xampp\FilezillaFTP\Filezilla%20Server.xml HTTP/1.1
Host: 192.168.10.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Mon, 06 Nov 2023 22:54:40 GMT
Server: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.2.4
X-Powered-By: PHP/8.2.4
Content-Length: 1669
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html>
        /* $file = $_GET['file']; */
</html>
<FileZillaServer>
    <Settings>
        <Item name="Admin port" type="numeric">14147</Item>
    </Settings>
    <Groups />
    <Users>
        <User Name="admin">
            <Option Name="Pass"></Option>
            <Option Name="Group"></Option>
            <Option Name="Bypass server userlimit">0</Option>
            <Option Name="User Limit">0</Option>
            <Option Name="IP Limit">0</Option>
            <Option Name="Enabled">1</Option>
            <Option Name="Comments"></Option>
            <Option Name="ForceSsl">0</Option>
            <IpFilter>
```

Posteriormente tenemos que buscar el volcado de memoria **.mem**, el cual se encuentra entre los paquetes **FTP-DATA** y que debemos volcar de alguna otra forma.

Hay varias formas de realizarlo y, en este caso, vamos a separar los paquetes **FTP-DATA** en una captura aparte para parsearla con "**tshark**".

Debemos seleccionar todos los paquetes de **FTP-DATA** y acceder a "**Exportar paquetes especificados**" dentro de la pestaña "**Archivo**".



Una vez con la captura de los paquetes FTP-DATA separados de los demás, podemos utilizar tshark para extraer el archivo comprimido **.rar** en formato hexadecimal.

- **tshark -2 -r memdump.pcap -T fields -e tcp.payload > memdump**

```
└─$ tshark -2 -r memdump.pcap -T fields -e tcp.payload > memdump

└─$ file memdump
memdump: ASCII text, with very long lines (28960)
```

Ahora debemos usar la utilidad "**xxd**" para convertir el archivo en el comprimido final y así poder descomprimirlo correctamente.

- **xxd -r -p memdump > memdump.rar**
- **unrar x memdump.rar**

```
└─$ xxd -r -p memdump > memdump.rar

└─$ file memdump.rar
memdump.rar: data

└─$ unrar x memdump.rar

UNRAR 6.11 beta 1 freeware      Copyright (c) 1993-2022 Alexander Roshal


Extracting from memdump.rar

Extracting  memdump.mem                                               OK
All OK
```

Ahora sí, podemos extraer el **md5** del volcado de memoria para así construir la flag.

- **md5sum memdump.mem**

```
└─$ md5sum memdump.mem
ca93f01d55d60a3c4ea7790ed584b04f  memdump.mem
```

La flag quedaría tal que así: **UAM{LFI_ca93f01d55d60a3c4ea7790ed584b04f}**

2.2. Flag 2

Descripción

¡ENHORABUENA! Has conseguido volcar en tu máquina la copia de seguridad de la captura de memoria que teníamos almacenada y que el atacante ha robado, ahora debes continuar con la investigación averiguando, entre la captura de red y el archivo de memoria, si el atacante ha podido hacerse con algún dato sensible o contraseña que pudiese contener.

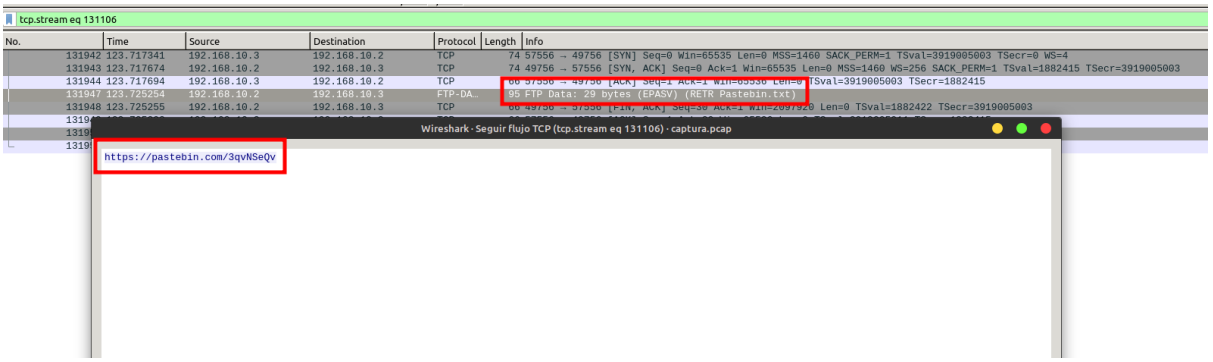
Formato de la flag: UAM{}

Resolución

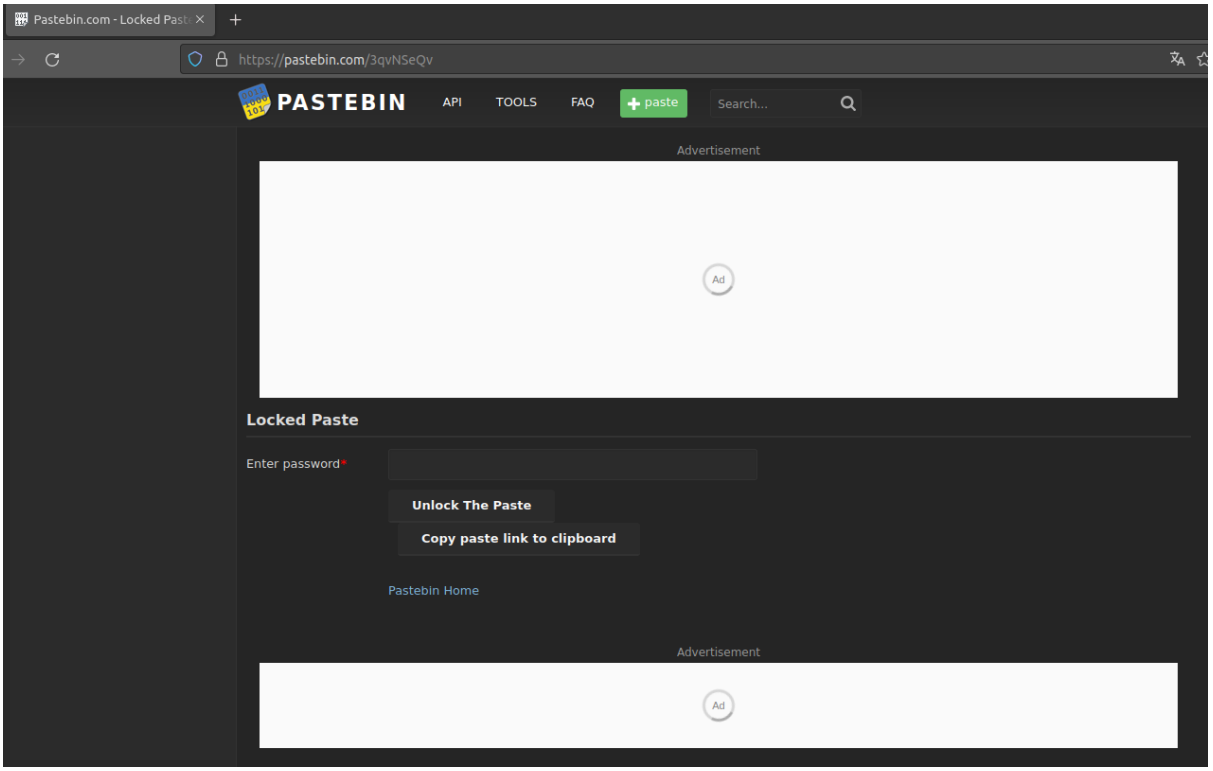
Para esta segunda flag debemos utilizar **Volatility3**, ya que, en principio, el perfil de Windows 11 no podría conseguirse de manera estandar (también es posible sacar la flag de manera mucho más fácil, que veremos después de esta solución).

Lo primero que tenemos que pensar es, ¿qué tenemos que buscar dentro del volcado de memoria? Ya que con un fichero tan grande no es posible analizarlo todo.

En la captura de red de la flag anterior podemos ver, en **FTP-DATA,** otro fichero el cual hace referencia a **Pastebin** y en el cual podemos encontrar una URL.



Este **Pastebin** requiere una contraseña, por lo que seguramente tengamos que encontrarla dentro del volcado de memoria.



Ya conociendo que es lo que creemos que tenemos que buscar generamos un perfil propio del sistema operativo del volcado de memoria (no es 100% necesario).

- **python3 volatility3/vol.py --save-config config.json -f memdump.mem windows.info**

```
└─$ python volatility3/vol.py --save-config config.json -f memdump.mem windows.info
Volatility 3 Framework 2.5.2
Progress:  100.00               PDB scanning finished
Variable        Value

Kernel Base     0xf80267a1f000
DTB     0x1ae000
Symbols file:///home/borb3d/Descargas/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/CF32DE2E4A334C7C06FB63FCB6FAFB5C-1.json.xz
Is64Bit True
IsPAE   False
layer_name      0 WindowsIntel32e
memory_layer    1 FileLayer
KdVersionBlock  0xf802686289a0
Major/Minor     15.22621
MachineType     34404
KeNumberProcessors      4
SystemTime      2023-11-06 22:22:57
NtSystemRoot    C:\Windows
NtProductType   NtProductWinNt
NtMajorVersion  10
NtMinorVersion  0
PE MajorOperatingSystemVersion  10
PE MinorOperatingSystemVersion  0
PE Machine      34404
PE TimeDateStamp        Tue Jun 17 09:32:46 2036
```

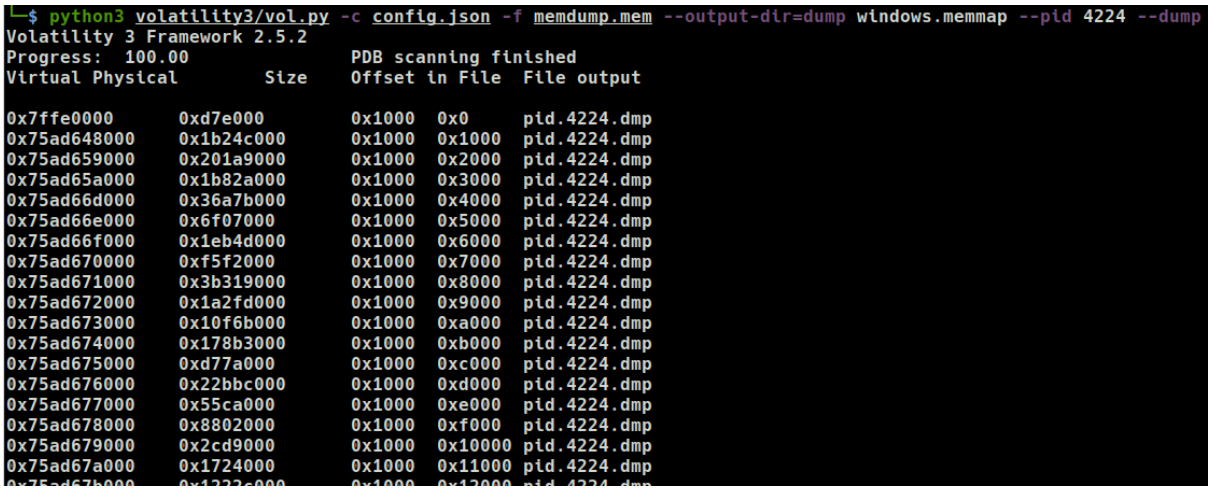
Posteriormente, debemos utilizar este perfil para realizar las consultas necesarias al volcado. Primero listaremos los procesos de la misma.

- **python3 volatility3/vol.py -c config.json -f memdump.mem windows.pslist.PsList**

```
└─$ python3 volatility3/vol.py -c config.json -f memdump.mem windows.pslist.PsList | grep -i note
4224ress936100.0Microsoft.Note  0x96029fdba0c0  39      -       1       False   2023-11-06 22:22:19.000000      N/A     Disabled
```

Teniendo el "**PID**" del proceso, podemos proceder a realizar un volcado del mismo.

- **python3 volatility3/vol.py -c config.json -f memdump.mem --output-dir=dump windows.memmap --pid 4224 --dump**



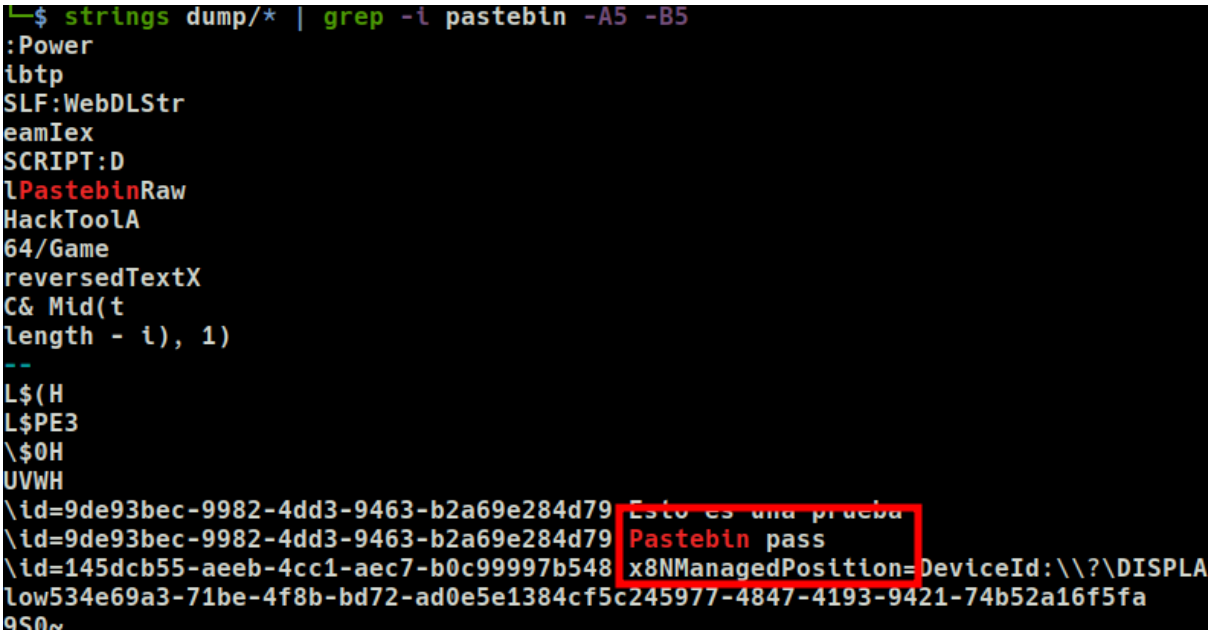
Ahora sí, con todo este volcado podemos revisarlo para extraer los datos que queremos y, como sabemos que queremos encontrar una contraseña de un Pastebin, podemos filtrar por esta palabra.

- **strings dump/* | grep -i pastebin -A5 -B5**

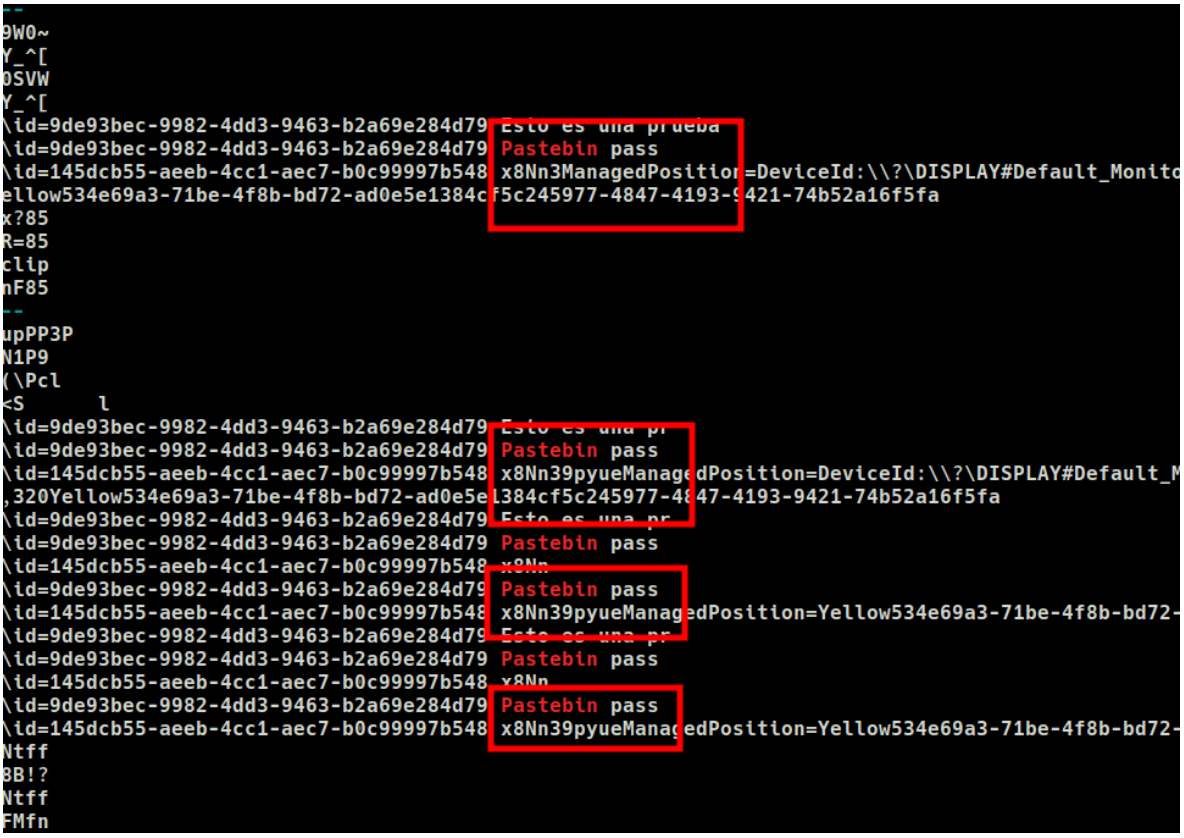
En los volcados que hemos extraído podemos observar que aparece el texto "**Pastebin pass**" el cual nos indica que por ahí debe estar la contraseña y, si está dentro de un proceso de notas, es posible que tenga un formato parecido al siguiente:

```
Pastebin pass
Esta es la contraseña
```

En el primero que nos encontramos no vemos nada que pueda ser una contraseña, aunque si que podría empezar por la palabra siguiente a "**Pastebin pass**".

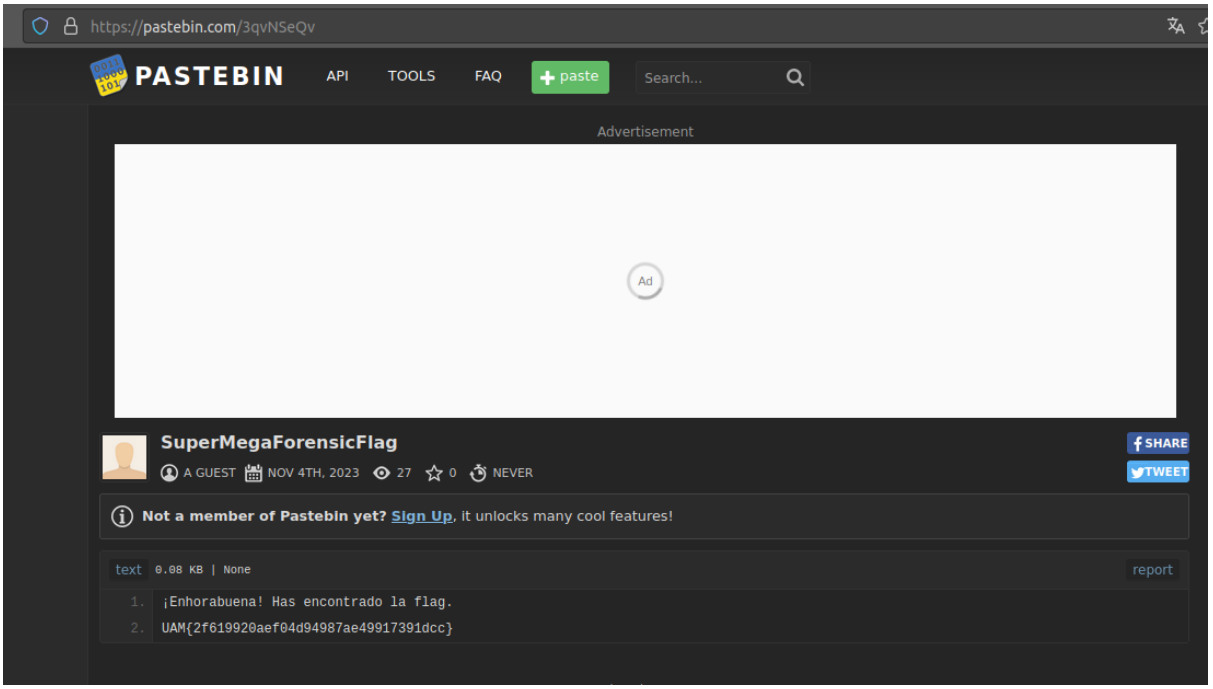


Sin embargo, si seguimos bajando en el filtrado realizado, podemos observar que la contraseña comienza a aparecer completa.



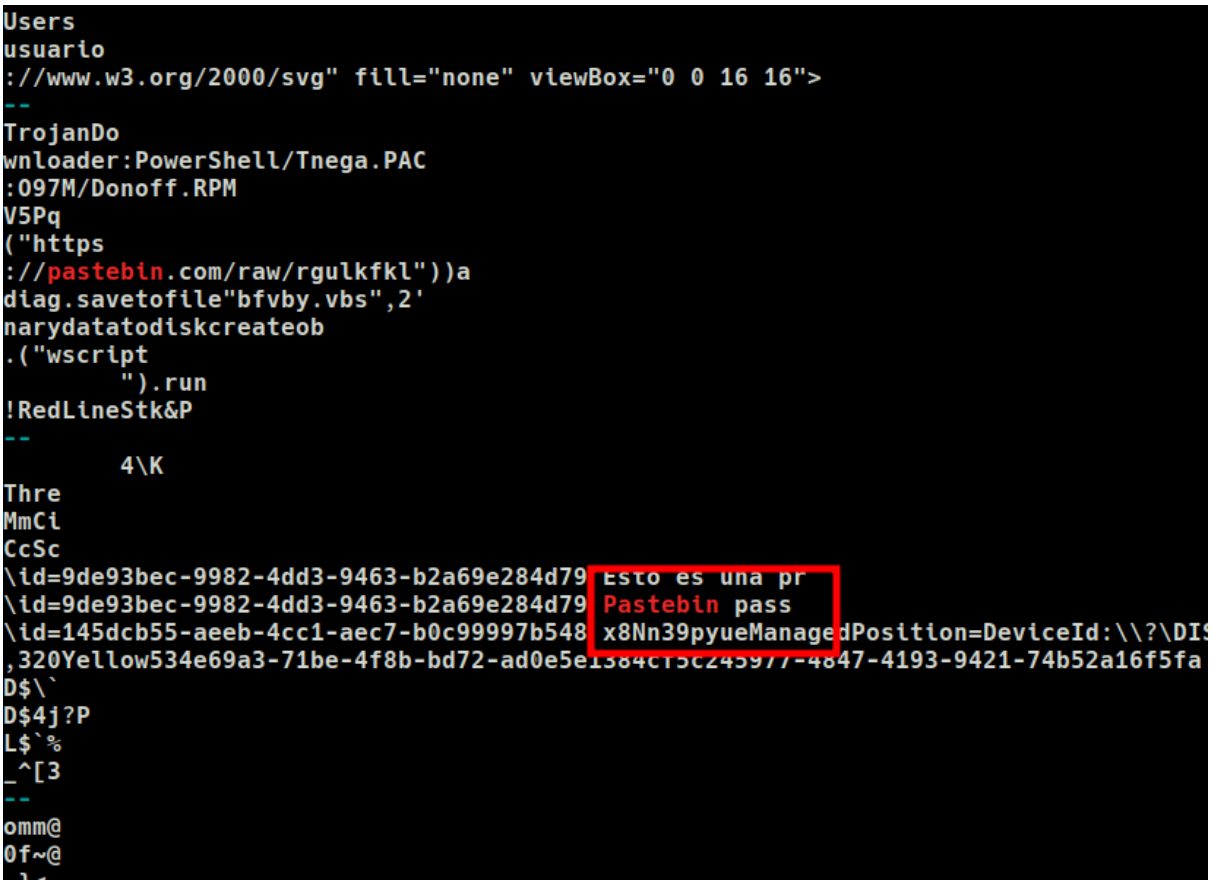
La probamos en la URL de Pastebin encontrada anteriormente y hemos conseguido la flag.

- URL de Pastebin: https://pastebin.com/3qvNSeQv
- Contraseña de Pastebin: **x8Nn39pyue**



La opción alternativa (y más sencilla) es sin utilizar **Volatility**, simplemente realizando un **Strings** y filtrando de la misma manera que en el paso anterior (aunque en este caso, habría mucha más información que ir filtrando).

- **strings memdump.mem | grep -i pastebin -A5 -B5**



Hispasec]

Hispasec Sistemas S.L.
C/ Severo Ochoa, 10 - 29590, Málaga
Telf: (+34) 952 020 494
info@hispasec.com