Karla Beatriz Aguilar Martínez – AM20072

Práctica2

Volcado de Memoria RAM – Informe

Fase 1: Preparación del Entorno y Verificación de Integridad

Antes del análisis, se valida la integridad del archivo de memoria y se preparan los recursos necesarios para la correcta ejecución de Volatility.

1.1 Verificación de Hash SHA-256

Se calcula la huella digital del archivo para garantizar que no ha sido alterado.

Comando:

```
sha256sum memdump1.mem
```

Resultado:
0bd1f0647ab9b0cad7be20aa50017ffe... memdump1.mem

Interpretación:
Esta firma garantiza que el archivo es auténtico; cualquier modificación generaría un hash distinto.

1.2 Descarga de Símbolos PDB

Volatility necesita símbolos del kernel de Windows para interpretar adecuadamente estructuras internas.

Comando:

```
wget https://downloads.volatilityfoundation.org/volatility3/symbols/windows.zip
```

El archivo descargado se mueve al directorio volatility3/symbols.

Fase 2: Identificación del Sistema Operativo

Se obtiene información del sistema operativo a partir de la imagen de memoria.

Comando:

```
python3 vol.py -f memdump1.mem windows.info
```

Datos relevantes encontrados:

- **Sistema operativo:** Windows 10
- **Arquitectura:** 64 bits
- **SystemTime:** 2025-11-24 05:05:03 UTC

Importancia:
La fecha del sistema permite ubicar el incidente en un contexto temporal preciso.

Fase 3: Revisión de Procesos Activos (PsList)

Se analiza qué procesos estaban ejecutándose cuando se creó el volcado.

Comando:

```
python3 vol.py -f memdump1.mem windows.pslist.PsList
```

Procesos identificados:

1. **System (PID 4):** Proceso núcleo del sistema.
2. **lsass.exe (PID 564):** Maneja autenticación y credenciales; esencial para el análisis.
3. **FTK Imager.exe (PID 2608):** Indica que esta herramienta generó el volcado de RAM.

Fase 4: Extracción de Hashes (Hashdump)

Al estar cargado *lsass.exe*, se pueden extraer hashes NTLM residentes en memoria.

Comando:

```
python3 vol.py -f memdump1.mem windows.hashdump.Hashdump
```

Hallazgos:

- **Usuario principal analizado:** *Bea*
- **Hash NTLM encontrado:** 65a73c83a606313c7137b6b1a6ec729c

Fase 5: Descifrado del Hash NTLM

Se utiliza CrackStation para intentar recuperar la contraseña original.

Procedimiento:

1. Se ingresa el hash en la plataforma.
2. Se detecta como hash tipo NTLM.
3. **Contraseña recuperada:** *contraseña*

Fase 6: Validación del Resultado

Para confirmar la exactitud del análisis, se realiza una prueba en el sistema original.

Acción:
Se inicia Windows 10, se selecciona el usuario "Bea" y se introduce la contraseña *contraseña*

Resultado:
El acceso es exitoso, demostrando que la contraseña obtenida desde la memoria RAM era correcta.

Herramientas Utilizadas

- **Sistema Operativo Base:** Kali Linux
- **Framework Forense:** Volatility 3 (Python 3)
- **Evidencia Analizada:** *memdump1.mem*
- **Utilidades Complementarias:**
    - *wget* para descarga de símbolos
    - CrackStation para descifrado de hashes NTLM