Análise a sistemas ligados (*live*)

António Pinto
apinto@estg.ipp.pt

Nov 2022 (v3)

Sumário

Conceitos introdutórios

- Investigação digital a sistemas ligados tentar extrair **informação volátil**
- Informação de estado em memória **RAM**
- Saber o que está acontecer **no momento**
- Saber **quem** está a fazer o quê
- Deve ser visto como técnica **complementar** e não como substituta

Porquê recorrer-se à investigação *live*

- Discos de grande capacidade
  - Discos cada vez **maiores** e mais baratos
  - Discos com **TBs** de armazenamento são frequentes
  - Pesquisas e cópias muito demoradas
- Sistema que não podem ser parados (sistemas críticos)
- Sistemas podem não ser fáceis de capturar (dimensões, quantidades, localizações)
- Alguma informação só existe em RAM

Estratégias de investigação *live*

- Duas grandes estratégias
  - Recolha de informação com a execução de **utilitários ou *scripts***
  - Recolha de uma imagem da memória RAM (**RAM *dump***)
- Ambas implicam alteração do estado do equipamento em análise (**intrusivas**)

Especificidades

- **Minimizar** alterações no sistema analisado
- Ponderar **utilidade e impacto** das ferramentas
- **Momento** em que se faz a investigação é crucial
- Requer **acesso** ao sistema

Potenciais problemas

- Muitas ferramentas **dependem** de funcionalidades do sistema em análise
- Sistema pode estar corrompido ou **infetado** (troca de *drivers*, comandos, módulos de kernel, troca de *DLLs*, . . .)

Alterações

- Alterações ao sistema em análise devem ser minimizadas, mas são **inevitáveis**
- Usar aplicações **reconhecidas**
- **Evitar deteção** (intrusos podem estar a monitorizar)
- Alterações podem ser **explicadas** (ex.: impressões digitais de familiares numa cena de um crime)

Informação passível de recolha

- Processos em execução
- Ficheiros abertos
- Ligações de rede
- Cópia da memória RAM
- Outras informações

Exercício

Recolha manual de informação (20 minutos)

Utilizando comandos disponíveis no sistema operativo, recolha a seguinte informação em 2 sistemas (Windows e Linux):

- Lista de processos em execução

  Lin **ps -cafe**
  Win **tasklist** ou **wmic process list**

- Lista de ligações de rede

  Win/Lin **netstat**

- Lista de ficheiros em uso

  Lin **lsof**
  Win **handle**[1]

Submeta sua análise crítica pelo moodle (ficheiro PDF)

[1] `docs.microsoft.com/pt-pt/sysinternals/downloads/handle`

Exercício

Recolha automatizada (40 minutos)

O *ir-rescue* é um *script* de recolha de informações *Live* com versões para Linux e Windows.

Obtenha-o de `https://github.com/diogo-fernan/ir-rescue`, execute ambas as versões e tente identificar respostas para as seguintes perguntas:

- Sistema Operativo utilizado
- Ligações estabelecidas com exterior
- Portas TCP à escuta
- Processos a correr no sistema
- Configuração de rede

Submeta sua resposta pelo moodle (ficheiro PDF)

Dados em RAM

- Aplicações **não protegem** conteúdos em RAM
- Aplicações que **tratam dados sensíveis**, não foram desenhadas para tal
  - Processadores de texto, folhas de cálculo, ...
- Palavras-passe introduzidas nas aplicações (ex. *browsers*) são **replicadas** na RAM (*buffers* e *stacks*)
- *Core dumps* de **falhas** em aplicações podem conter informação sensível
- Informação pode **persistir** por algum tempo em RAM

Persistência de dados em RAM

- Chow **estudou** a persistência de dados em RAM [1]
- **Experimentou** com sistemas operativos Windows e Linux
- **Criaram programas** que
  - Inseriam vários registos com 20bytes no seu arranque
  - Cada registo incluí número de série, *timestamp*, . . .
  - Contavam diariamente o número destes registos que conseguiam encontrar em RAM

Persistência de dados em RAM (2)

- Execução
  - Imediatamente **após o término** das aplicações, identificaram-se 2 a 4MB de registos
  - **Após 14 dias**, identificaram-se entre 23KB e 3MB de registos
  - **Após 28 dias**, 7KB de registos persistiam
- *Reboot*
  - Após *soft reboot*, muitos dados **mantinham-se** em memória
  - Após *hard reboot*, num dos casos, dados mantinham-se mesmo após 30 minutos **desligado** (IBM ThinkPad T30)

Exercício

Aquisição de RAM (30 minutos)

O programs WinPmem permite obter cópias de memória RAM de sistemas Windows.

Pode ser descarregado de `https://github.com/Velocidex/WinPmem`.

Obtenha uma cópia da RAM do seu PC em formato *raw*.

Análise de cópias de memória

- Volatility é uma ferramenta open source que permite extrair informação de *dumps* de memória RAM
- É multiplataforma (Windows, Linux, MacOS)

+info `https://www.volatilityfoundation.org/`

`https://github.com/volatilityfoundation/volatility/wiki/Command-Reference`

Identificar tipo de dump

```
1 aap@aap$ volatility imageinfo -f mem.bin
2 Volatile Systems Volatility Framework 2.2
3 Determining profile based on KDBG search...
4
5 Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86  (Instantiated
6 with WinXPSP2x86)
7 AS Layer1 : JKIA32PagedMemoryPae (Kernel AS)
8 PAE type : PAE
9 DTB : 0x324000L
10 KDBG : 0x80545ce0
11 Number of Processors : 1
12 Image Type (Service Pack) : 3
13 KPCR for CPU 0 : 0xffdff000
14 KUSER_SHARED_DATA : 0xffdf0000
15 Image date and time : 2013-09-29 17:03:37 UTC+0000
16 Image local date and time : 2013-09-29 18:03:37 +0100
```


Alguns comandos

- **pslist**: lista de processos
- **hivelist**: posições de memória de partes do *Registry*
- **hashdump**: extração de *hashes* de *passwords* (Windows)
- **pstree**: lista de processos, organizados em árvore
- **psscan**: lista de processos por pesquisa (identificar processos escondidos)
- **psxview**: lista de processos alternativa
- **connections**: lista de ligações de rede
- **connscan**: lista de ligações de rede (alternativa)
- **netscan**: lista de ligações de rede (Win10)
- **notepad**: ver texto visível no notepad
- **clipboard**: mostra o conteúdo do *clipboard* (win)
- ...

Obter lista de processos em execução (pslist)

```
aap@aap$ volatility -f mem.bin - -profile=Win10x64_10586
pslist

Volatility Foundation Volatility Framework 2.6
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start
Exit
------------------ ------------------
0xffffe0009067d680 System 4 0 111 0 ------ 0 2018
0xffffe00091a387c0 smss.exe 288 4 2 0 ------ 0 2018
0xffffe000924e07c0 csrss.exe 368 356 8 0 0 0 2018
0xffffe0009295c2c0 smss.exe 420 288 0 -------- 1 0 2018
```


Localizar partes do *registry* no dump

```
aap@aap$ volatility hivelist -f mem.bin
    –profile=WinXPSP2x86
Volatile Systems Volatility Framework 2.2
Virtual Physical Name
---------- ---------- ----
...
0xe16106b8 0x0e0566b8 \Device\HarddiskVolume1\
    WINDOWS\system32\config\software
0xe160d758 0x0e052758 \Device\HarddiskVolume1\
    WINDOWS\system32\config\default
0xe1622008 0x0e0aa008 \Device\HarddiskVolume1\
    WINDOWS\system32\config\SAM
0xe1610b60 0x0e056b60 \Device\HarddiskVolume1\
    WINDOWS\system32\config\SECURITY
0xe1035b60 0x02ba9b60 \Device\HarddiskVolume1\
    WINDOWS\system32\config\system
...
```

Obter hash de passwords contidos no dump

```
1 aap@aap$ volatility hashdump -f mem.bin
      –profile=WinXPSP2x86 -y 0xe1035b60 -s
      0xe1622008
2 Volatile Systems Volatility Framework 2.2
3 Administrator:500:[ELIMINADO]:::
4 Guest:501:aad3b435b51404eeaad3b435b51404ee:31
      d6cfe0d16ae931b73c59d7e0c089c0:::
5 HelpAssistant:1000:
      cf299ebdc62704b31c651cdc95def456:19732
      c3a9cfebd2c0115a3b282ad392a:::
6 SUPPORT_388945a0:1002:[ELIMINADO]:::
7 AAP:1003:[ELIMINADO]:::
```

Nota: Exemplo assume que o *SYSTEM hive* está no endereço 0xe1035b60, e que o *SAM hive* está no endereço 0xe1622008.

Exercício

Extração de passwords (30 minutos)

Descarregue o dump de memória disponível no moodle (Bob.vmem).

Descubra o sistema operativo desta cópia de memória RAM e extraia o hashdump da mesma. De seguida, descubra as password dos vários utilizadores.

Submeta um write-up pelo moodle

Processos potencialmente maliciosos (malfind)

```
1 aap@aap$ volatility malfind -f ./stuxnet.vmem
      –profile=WinXPSP3x86
2 Volatility Foundation Volatility Framework 2.6
3 Process: csrss.exe Pid: 600 Address: 0x7f6f0000
4 Vad Tag: Vad Protection:
      PAGE_EXECUTE_READWRITE
5 Flags: Protection: 6
6
7 0x7f6f0000 c8 00 00 00 1f 01 00 00 ff ee ff ee 08 70 00
      00 0x7f6f0020 00 02 00 00 00 20 00 00 8d 01 00
      00 ff ef fd 7f 0x7f6f0030 03 00 08 06 00 00 00 00
      00 00 00 00 00 00 00 00
8 0x7f6f0000 c8000000 ENTER 0x0, 0x0
9 0x7f6f0004 1f POP DS
10 0x7f6f0005 0100 ADD [EAX], EAX
11 0x7f6f0009 ee OUT DX, AL
12 ...
```


Exercício

Identificação de *malware* (30 minutos)

Descarregue o dump de memória disponível em:
`https://tinyurl.com/yaaw2o4e`

Descubra o sistema operativo desta cópia de memória RAM. Extraia (opção -D) processos potencialmente maliciosos. Valide se os processos extraídos contém malware.

Submeta um write-up, identificando o malware, pelo moodle

Capture The Flag (CTF)

Memory Forensics

`https://bit.ly/2QTJIZN`

A flag tem o formato flag{.....}

Submeta um write-up pelo moodle