La *memory forensics*, ossia le indagini sul contenuto della memoria RAM di un dispositivo digitale come un computer, rappresentano una materia abbastanza complessa, infatti occorre conoscere molto bene come funzionano i processi ed il sistema operativo, al fine di comprendere cosa è in esecuzione e quali file sono coinvolti.
In quest'articolo si tratterà un piccolo esempio di "*malware hunting*" (caccia al malware) per illustrare come con degli strumenti open source ed online, si può trovare il software maligno che infesta un computer, chiaramente è un percorso semplice per fini descrittivi e didattici, dato che scovare i malware e le loro varie declinazioni spesso è molto più complicato.

di Nanni Bassetti

UN ESEMPIO PRATICO DI *MEMORY FORENSICS* CON L'*OPEN SOURCE*

Nanni BASSETTI è laureato in Scienze dell'Informazione, libero professionista specializzato in digital forensics, fondatore di CFI (Computer Forensics Italy) e project manager di CAINE Linux/GNU Live distro per indagini informatiche. Docente, relatore in parecchi corsi ed eventi, autore di molti articoli tecnici e di un paio di libri.

1. *Introduzione*

Di seguito vengono descritti i passaggi operativi che occorre effettuare per il "*malware hunting*", iniziando con il download del file immagine di una RAM che appartiene ad un computer compromesso da un noto malware, in questo caso si tratta del "vecchio" SpyEye:

https://code.google.com/archive/p/volatility/wikis/SampleMemoryImages.wiki

ATTENZIONE i file contengono veramente il malware, quindi operate in ambiente di test a vostro rischio.

Il file in questione è: **spyeye.vmem (512Mb).** Il percorso operativo sarà così composto:

1. Cerchiamo le connessioni in esecuzione.
2. Cerchiamo i processi che le creano.
3. Cerchiamo il malware iniettato nel processo.
4. Estraiamo il malware.
5. Lo controlliamo su https://virustotal.com/.
6. Estraiamo le stringhe contenute nel malware al fine di cercare la chiave di registro che lo manda in esecuzione.
7. Analizziamo la chiave di registro per trovare qual'è il file eseguibile e dov'è allocato nel computer.

I programmi utilizzati saranno:

- OS CAINE 8.0
- Volatility Framework 2.5
- VirusTotal.com
- Strings

2. *Procedura operativa*

PASSO 1: Cerchiamo le connessioni in esecuzione

```
volatility -f spyeye.vmem connscan
```

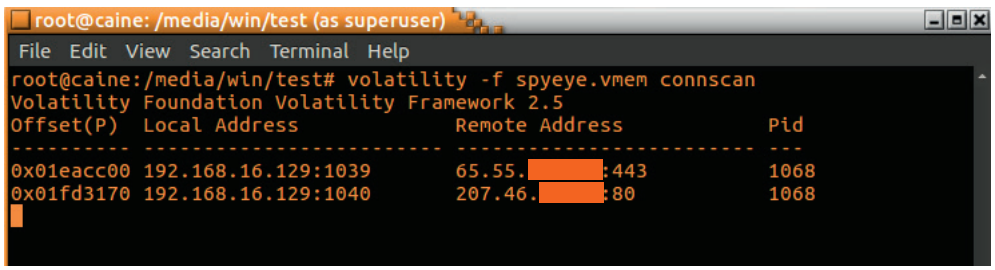

Figura 1

Troviamo la connessione all'indirizzo IP 65.55.xxx.xxx sulla porta 443 ID del Processo: 1068 (figura 1).
Troviamo la connessione all'indirizzo IP 207.46.xxx.xxx sulla porta 80 ID del Processo: 1068 (figura 1).

PASSO 2: Cerchiamo i processi che le creano

```
volatility -f spyeye.vmem pslist
```

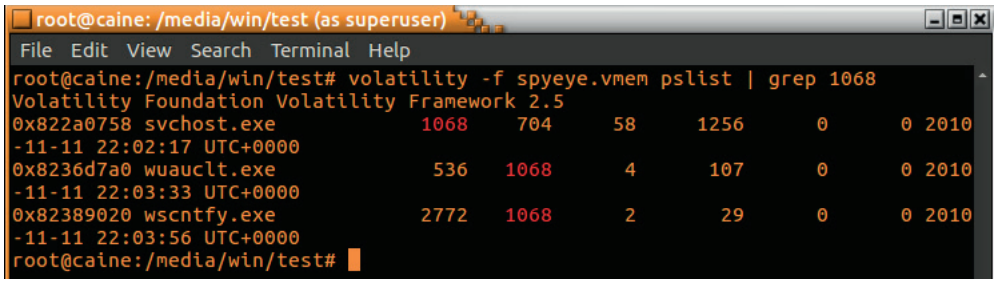

Figura 2

Scopriamo che il processo è: 0x822a0758 **svchost.exe 1068** (figura 2).
SVCHOST è un esecutore di servizi, per dirla in maniera semplice ed è normale che sia in esecuzione.

PASSO 3: Cerchiamo il malware iniettato nel processo sospetto

```
volatility -f spyeye.vmem malfind -p 1068
```

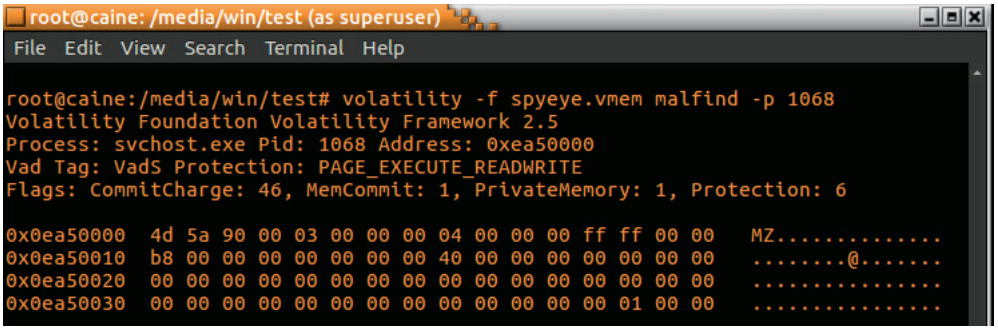

Figura 3

Il -p 1068 serve per indicare il processo numerato come 1068 ossia SVCHOST (figura 3).

```
Volatility Foundation Volatility Framework 2.5
Process: svchost.exe Pid: 1068 Address: 0xea50000
Vad Tag: VadS Protection: PAGE_EXECUTE_READWRITE
Flags: CommitCharge: 46, MemCommit: 1, PrivateMemory: 1, Protection: 6

0x0ea50000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 MZ..............
0x0ea50010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ........@.......
0x0ea50020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0x0ea50030 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 ................
```

Notiamo che il plugin malfind di Volatility trova all'indirizzo 0xea50000 un file eseguibile, perchè inizia con "MZ", la tipica sequenza di caratteri con la quale iniziano tutti i file EXE.
Inoltre notiamo che la VADS PROTECTION è PAGE_EXECUTE_READWRITE, ossia il Virtual Address Descriptor, che rappresenta ciascun file presente in memoria e sul disco, è eseguibile (EXECUTE).

PASSO 4: Estraiamo il malware

```
volatility -f spyeye.vmem vaddump -p 1068 -b 0xea50000 -D /home/caine/dump

Volatility Foundation Volatility Framework 2.5
Pid Process Start End Result
---------- -------------------- ---------- ------
1068 svchost.exe 0x0ea50000  0x0ea7dfff /home/caine/dump/svchost.exe.22a0758.0x0ea50000-
0x0ea7dfff.dmp
```

Abbiamo chiesto a Volatility di estrarre il nodo VAD che inizia all'indirizzo 0xea50000 e salvarlo nella directory /home/caine/dump, il software l'ha nominato automaticamente in **svchost.exe.22a0758.0x0ea50000-0x0ea7dfff.dmp** (figura 4).

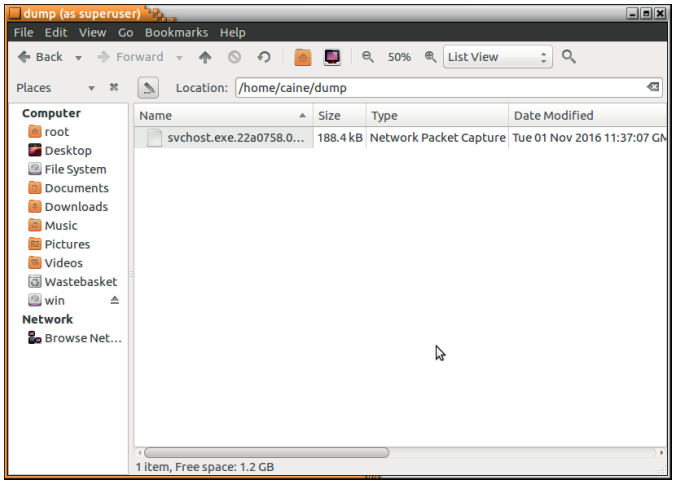

Figura 4

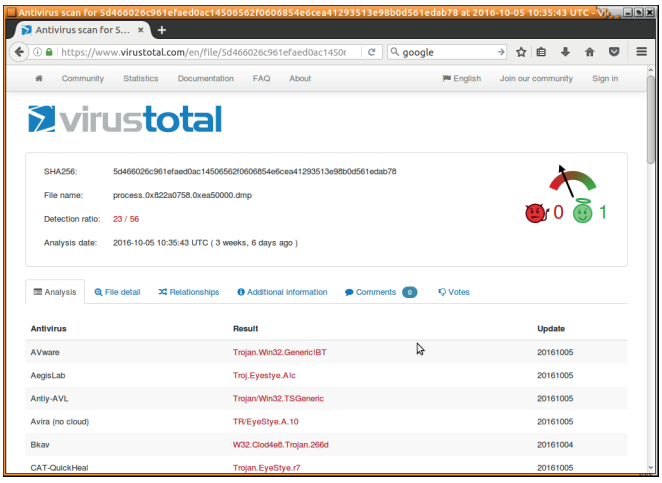

Figura 5

PASSO 5: Controlliamo il file su VirusTotal.com

Carichiamo il file **svchost.exe.22a0758.0x0ea50000-0x0ea7dfff.dmp** su **VirusTotal.com,** un grande meta-motore di ricerca che invia il file a vari scanner antivirus ed otteniamo il riconoscimento del tipo di malware (figura 5).

PASSO 6: Estraiamo le stringhe contenute nel malware al fine di cercare la chiave di registro che lo manda in esecuzione

```
strings svchost.exe.22a0758.0x0ea50000-0x0ea7dfff.dmp > strings.txt
```

Osservando le stringhe estratte possiamo trovare molte informazioni, come il nome del file eseguibile e la chiave di registro di Windows che lo manda *in running* (figura 6).

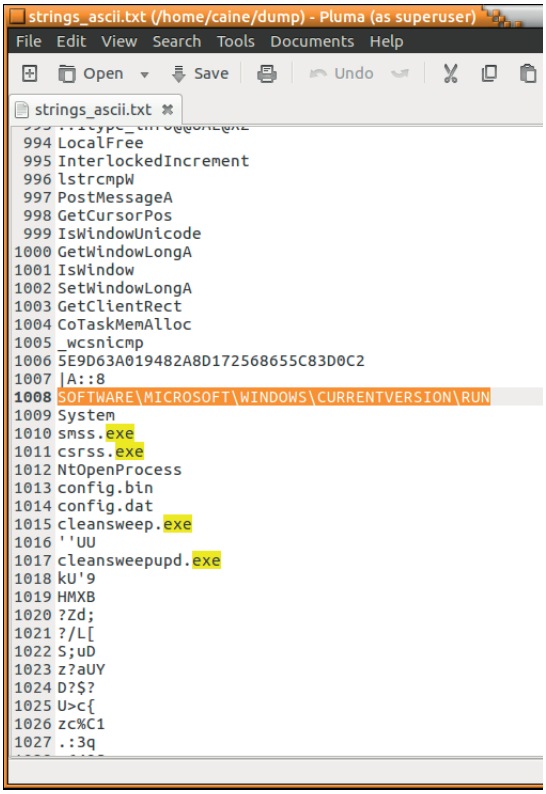

Figura 6

PASSO 7: Individuiamo il file

```
volatility -f spyeye.vmem printkey -K "SOFTWARE\
MICROSOFT\WINDOWS\CURRENTVERSION\RUN"
Volatility Foundation Volatility Framework 2.5
Legend: (S) = Stable (V) = Volatile

REG_SZ cleansweep.exe : (S) C:\cleansweep.exe\
cleansweep.exe
```

Abbiamo scovato il "cattivone"! Si trova in **C:\cleansweep.exe\ cleansweep.exe** e così possiamo andarlo a prendere e cancellarlo o analizzarlo, ecc.

3. Conclusioni

È sempre emozionante usare gli strumenti a "*basso livello*", ossia senza troppi automatismi ed interfacce grafiche, perché sembra proprio di scavare nelle informazioni digitali e capire meglio come sono organizzate, ma ricordiamo che il presente articolo è utile solo per dare un esempio veloce e comprensibile di come si possa fare della "memory" e "malware forensics", che forse sono le branche più ostiche di tutta la digital forensics, poiché comportano conoscenze veramente approfondite dei sistemi, delle reti, del reverse engineering anche in Assembly, pertanto a chi vuole approcciare questi temi, non ci resta che augurare buona caccia!

BIBLIOGRAFIA

- The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory - Michael Hale Ligh, Andrew Case, Jamie Levy, AAron Walters - 2014 WILEY
- http://trickandtipsforpc.blogspot.it/2015/07/malware-memory-forensics-introduction.html
- http://securityxploded.com/malware-memory-forensics.php
- http://www.behindthefirewalls.com/2013/07/zeus-trojan-memory-forensics-with.html. ©