Informatyka śledcza

Laboratorium nr 6

Spis treści

Wstęp

Poniższe laboratorium ma na celu zaprezentowanie w praktyce narzędzi umożliwiających wykonanie zrzutu pamięci RAM oraz jej późniejszej analizy. W trakcie laboratorium student zaznajomi się z podstawowymi informacjami, zawartymi w pamięci operacyjnej. Do laboratorium dołączony jest plik (memory3.vmem) zawierający dane, które podlegają analizie. Proszę o przygotowanie raportu z wykonanych zadań w formacie pdf oraz opisanie uzyskanych rezultatów z użycia poszczególnych pluginów frameworka Volatility.

Wykorzystywane narzędzia w trakcie laboratorium:

1. **FTK Imager**
2. **AVML**
3. **Volatility**

Zadanie 1 – Tworzenie zrzutu pamięci z systemu Windows

1. Przy wykorzystaniu systemu Windows zainstaluj darmowy program FTK (https://accessdata.com/product-download/ftk-imager-version-4-5).
2. Uruchom ww. program i z zakładki file wybierz opcje „Capture Memory". Operacja ta spowoduje utworzenie zrzutu z pamięci uruchomionej stacji z zainstalowanym systemem Windows.

Zadanie 2 – Tworzenie zrzutu pamięci z systemu Linux

1. Pobierz przy pomocy stacji z zainstalowanym systemem Linux program avml (https://github.com/microsoft/avml/releases).
2. W pobranym pliku zmień uprawnienia (chmod 755).
3. Przy wykorzystaniu przeglądarki internetowej wywołaj dowolną stronę internetową.
4. Przy wykorzystaniu programu graficznego otwórz dowolne zdjęcie lub plik JPG.

5. Wykonaj zrzut pamięci poleceniem: sudo ./avml nazwa.dmp.
6. Wykorzystując utworzony plik nazwa.dmp użyj polecenia strings do wyświetlenia zawartości pamięci. Odpowiedz, czy jesteś w stanie odnaleźć w pamięci informacje o wywołanej stronie oraz pliku graficznym bez pomocy dodatkowego filtra?
7. Przy pomocy filtra grep wyszukaj wcześniej wywołaną stronę oraz plik graficzny.

Zadanie 3 – Analiza pamięci przy wykorzystaniu programu Volatility

1. Wykorzystując wirtualną (lub natywną) maszynę z systemem Linux pobierz program Volatility (https://github.com/volatilityfoundation/volatility).
2. Pobierz z uczelnianej strony (UPEL) plik z przygotowanym obrazem memory3.vmem.
3. Przejdź do pliku z pobranym frameworkiem Volatility.
4. Sprawdź, czy ww. program nie potrzebuje dodatkowych bibliotek (jak np. Pythona, który jest niezbędny do uruchomienia Volatility) na wykorzystywanej maszynie z Linuxem (parametr -h). Jeżeli wyświetlane zostaną „pomoce" bez błędów sprawdź podstawowe informacje znajdujące się na badanym obrazie:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem imageinfo
Volatility Foundation Volatility Framework 2.6.1
```

5. Odpowiedz na pytania:
a. Jakie sugerowane profile są aktualnie podpowiadane przez program?
b. Do czego wykorzystywany jest adres KDBG?
c. DTB (Directory Table Base) – jest używany do translacji wirtualnego adresu na jaki adres?
d. O czym świadczą dane zawarte w KPCR (Kernel Processor Control Region) w odniesieniu do badanego obrazu?
6. Volatility wymaga do prawidłowej analizy wskazania profilu badanego obrazu. Wywołaj funkcje wyświetlenia listy procesów systemu:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.6.1
Offset(V)  Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
---------- -------------------- ------ ------ ------ -------- ------ ------ ------------------------------ ----------
0x810b1660 System                    4      0     58      183 ------      0
0xff2ab020 smss.exe                544      4      3       21 ------      0 2010-08-11 06:06:21 UTC+0000
0xff1ecda0 csrss.exe               608    544     10      369      0      0 2010-08-11 06:06:23 UTC+0000
```

Załącz wykonany zrzut z ww. polecenia i odpowiedz na pytania:
a. Jakie informacje zawierają poszczególne kolumny: Offset(V), PID, PPID, Thds, Hnds, Sess, Wow64, Start i Exit?
b. O czym świadczy znacznik (V) w rubryce Offset?
c. Który z niżej opisanych procesów został zakończony i kiedy?
d. Dlaczego procesy „System" i „smss.exe" nie posiadają informacji w rubryce Sess?
e. Który numer procesu należy do VMwareUser.exe?
7. Wykonaj polecenie:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 pslist -P
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Name                PID   PPID   Thds     Hnds   Sess  Wow64 Start                          Exit
```

Jaką zmianę wywołał wskaźnik *-P*? Porównaj zmianę w procesie VMwareUser.exe.

8. Wyświetlając listę procesów w formie „drzewa”:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 pstree
Volatility Foundation Volatility Framework 2.6.1
```

Odpowiedz na pytania:

a. Co oznaczają wyświetlone wcięcia i kropki?
b. Jakiego identyfikatora nie znajdziemy w prezentowanych tabelach?
c. Procesem nadrzędnym procesu smss.exe jest…?
d. Za co odpowiedzialny jest proces smss.exe?

9. Wykorzystując wskaźnik -h odszukaj i wyświetl załadowane biblioteki dll w badanym obrazie na podstawie procesu wscntfy.exe (Podpowiedź: do wyszukanego wskaźnika dodaj -p i podaj id procesu wscntfy.exe).

10. Przy pomocy polecenia dlldump wypakuj pliki dll w nowo utworzonym folderze:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 dlldump -D ~/Desktop/dll_dump/
Volatility Foundation Volatility Framework 2.6.1
```

Czy oddało się odzyskać plik: module.124.113f368.77f60000.dll?

11. Wyświetl otwarte powiązania „uchwyty” we wskazanym procesie i odpowiedz na pytania:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 handles -p 1668 -t Process
Volatility Foundation Volatility Framework 2.6.1
```

a. Do jakiego procesu należy wskazany PID (1168)?
b. Z jakim procesem wskazany PID (1168) posiada aktywny „uchwyt”?
c. Podaj PID odnalezionego aktywnego powiązanego procesu.

12. Polecenie Getsids wyświetla identyfikatory SID (Security Identifiers) powiązany z procesem. W ten sposób jesteśmy w stanie uchwycić procesy, które mają złośliwy charakter i mogą eskalować uprawnienia. Do jakich uprawnień należy wskaźnik (S-1-5-32-544)?

13. Przy wykorzystaniu wtyczki verinfo jesteśmy w stanie wyświetlić informacje o wersjach które zostały osadzone w plikach PE (nie wszystkie pliki posiadają te informacje). Odpowiedz na pytania:

a. Jaką wersje posiada plik: C:\WINDOWS\system32\SAMLIB.dll?
b. Podaj jego OS.
c. Podaj wersje pliku: *C:\ProgramFiles\VMware\VMware\Tools\TPAutoConnect.exe.*
d. Podaj LegalCopyright ww. pliku.

14. Wykorzystaj wtyczkę odpowiedzialną za przeglądarkę internetową IE i odpowiedz na pytania:

a. Podaj PID procesu IEXPLORE.EXE.
b. O której została uruchomiona przeglądarka?
c. Czy została wyświetlona strona www.yahoo.com?
d. Czy została wyświetlona strona www.bing.com?

15. Proszę o wyeksportowanie procesu pod nazwą wuauclt.exe:

```
┌──(kali㉿kali)-[~/Desktop/volatility]
└─$ python vol.py -f ~/Desktop/memory3.vmem --profile=WinXPSP3x86 procdump -p 468 -D ~/Desktop/Virus/
```

Poprawnie wykonane polecenie zwróci do utworzonego folderu plik (executable.468.exe) z procesu. Wykonaj jego analizę poprzez sprawdzenie sumy kontrolnej (np. md5sum) i poddaj go weryfikacji pod kątem obecności złośliwego oprogramowania (www.virustotal.com). Proszę o załączenie wyników z wykonanego działania.

Rozwiązania zadań muszą zawierać zrzuty ekranów ze wszystkich wykonanych elementów oraz szczegółowy opis uzyskanych rezultatów.