

Projekt nr. 3  
KRYCY  
Autorzy: Piotr Szewczyk, Paweł Murdzek

Wybrany projekt: DumpMe

Scenariusz zadania:

Pytania dotyczące zadania:

Pytanie nr. 1

Na początku pytają się o SHA1 pobranego pliku zgodnie z praktyką. Po uruchomieniu komendy w powershellu "Get-FileHash -Algorithm SHA1" dostajemy powyższy hash i wykonujemy zadanie.

Pytanie nr. 2

Z jakim systemem mamy doczynienia? Używając Volatility3 uruchamiając komendę "windows.info" otrzymujemy informacje o systemie. Dopasowujemy informację do formatu odpowiedzi.

```
KdDebuggerDataBlock      0xf800029f80a0
NTBuildLab                7601.18741.amd64fre.win7sp1_gdr.
CSDVersion                1
```

### Pytanie nr. 3

Kolejne pytanie dotyczy PID notatnika, filtrując wyszukiwania komendą "windows.pslist | Select-String "notepad"" znajdujemy wynik:

Q3 Solved : 2966

What was the process ID of notepad.exe?

\*\*\*\*  
3032

Hints

Submit

Windows PowerShell

PS C:\Users\piotr\volatility3-develop> python vol.py -f "C:\Users\piotr\Desktop\KRYCY\_proj3\65-DumpMe\temp\_extract\_dir\Triage-Memory.mem" windows.pslist | select-String "notepad"

Progress: 100.00 PDB scanning finished

3032 1432 notepad.exe 0xfa80054f9060 1 60 1 False 2019-03-22 05:32:22.000000 UTC N/A Disabled

PS C:\Users\piotr\volatility3-develop> |

### Pytanie nr. 4

W kolejnym pytaniu dostajemy pytanie o child process wscript.exe. Używając funkcji "windows.pstree", dowiadujemy się, że PID wscript.exe = 5116. Znow używamy komendy pstree, tym razem filtrując procesy 5116.

```
Progress: 100.00 PDB scanning finished
> ** 5116 3952 wscript.exe 0xfa8005a80060 8 312 1 True 2019-03-22 05:35:32.000000 UTC N/A \Device\HarddiskVolume2\Win
ows\SysWOW64\wscript.exe "C:\Windows\
System32\wscript.exe" //B //NOLOGO %TEMP%\vhjReUDEuumpX.vbs C:\Windows\SysWOW64\wscript.exe
> *** 3496 5116 UWkpjFjDzM.exe 0xfa8005ald9e0 5 109 1 True 2019-03-22 05:35:33.000000 UTC N/A \Device\HarddiskVolume2\Use
s\Bob\AppData\Local\Temp\rad93398
.tmp\UWkpjFjDzM.exe "C:\Users\Bob\AppData\Local\Temp\rad93398.tmp\UWkpjFjDzM.exe" C:\Users\Bob\AppData\Local\Temp\rad93398.tmp\UWkpjFjDzM.exe
**** 4660 3496 cmd.exe 0xfa8005bb0060 1 33 1 True 2019-03-22 05:35:36.000000 UTC N/A \Device\HarddiskVolume2\Windows\Sys
OW64\cmd.exe C:\Windows\System32
\cmd.exe C:\Windows\SysWOW64\cmd.exe
* 4048 1432 POWERPNT.EXE 0xfa80053d3060 23 765 1 True 2019-03-22 05:35:09.000000 UTC N/A \Device\HarddiskVolume2\Pro
gram Files (x86)\Microsoft Office\roo
t\Office16\POWERPNT.EXE "C:\Program Files (x86)\Microsoft Office\root\Office16\POWERPNT.EXE" C:\Program Files (x86)\Microsoft Office\root\Office16\POWER
PNT.EXE
```

Bezpośrednim child process jest:

Q4 Solved : 2939

Name the child process of wscript.exe.

\*\*\*\*\*  
UWkpjFjDzM.exe

Hints

Submit

## Pytanie nr. 5 oraz Pytanie nr. 6

W tym zadaniu trzeba znaleźć adres IP ofiary oraz adres atakującego, komenda “nmap” pozwala na wyświetlenie wszystkich połączeń sieciowych. Po wyniku od razu widać, że tym adresami są:

**Q5** Solved : 2877

What was the IP address of the machine at the time the RAM dump was created?

\*\*\*\*\*

10.0.0.101

Hints Submit

**Q6** Solved : 2821

Based on the answer regarding the infected PID, can you determine the IP of the attacker?

\*\*\*\*\*

10.0.0.106

Hints Submit

|             |       |            |       |               |      |             |      |                                |     |
|-------------|-------|------------|-------|---------------|------|-------------|------|--------------------------------|-----|
| 0x13e300e0  | UDPv4 | 10.0.0.0   | 83790 | *             | 0    | -           | -    | 2019-03-22 03:43:47.000000 UTC |     |
| 0x13e397190 | TCPv4 | 10.0.0.101 | 49217 | 10.0.0.106    | 4444 | ESTABLISHED | 3496 | UWkpjFjDzM.exe                 | N/A |
| 0x13e3986d0 | TCPv4 | -          | 49378 | 213.209.1.129 | 25   | CLOSED      | -    | -                              | -   |

## Pytanie nr. 7

W tym pytaniu mamy za zadanie odnaleźć ile procesów jest związanych z VCRUNTIME140.dll. Do sprawdzania powiązań z dll służy komenda “windows.dlllist”. Filtrujemy po VCRUNTIME140.dll i liczymy ilość wystąpień.

**Q7** Solved : 2695

How many processes are associated with VCRUNTIME140.dll?

\*

5

Hints Submit

```
PS C:\Users\piotr\volatility3-develop> python vol.py -f "C:\Users\piotr\Desktop\KRYCY_proj3\65-DumpMe\temp_extract_dir\Triage-Memory.mem" windows.dlllist |
Select-String "VCRUNTIME140.dll"
Progress: 100.00
1136 OfficeClickToR 0x7fefa5c0000 0x16000 VCRUNTIME140.dll C:\Program Files\Common Files\Microsoft Shared\ClickToRun\VCRUNTIME140.dll -1 2
019-03-22 05:32:0 9.000000 UTC Disabled
1272 EXCEL.EXE 0x745f0000 0x15000 VCRUNTIME140.dll C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll - 2248
-03-13 01:40:10.000000 UTC Disabled
3688 OUTLOOK.EXE 0x745f0000 0x15000 VCRUNTIME140.dll C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll - 2090
-04-30 15:36:50.000000 UT Disabled
2780 iexplore.exe 0x745f0000 0x15000 VCRUNTIME140.dll C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll - D
isabled
4048 POWERPNT.EXE 0x745f0000 0x15000 VCRUNTIME140.dll C:\Program Files (x86)\Microsoft Office\root\Office16\VCRUNTIME140.dll - D
isabled
```

## Pytanie nr. 8

Pytanie dotyczy o hash md5 zainfekowanego procesu. Z wcześniejszych logów wynika, że tym procesem jest PID = 3496, czyli UWkpjFjDzM.exe. Zatem trzeba odzyskać ten plik. Wtyczka dumpfiles pozwala na zapisanie takiego pliku. Używając komendy “windows.dumpfiles --pid 3496” zapisujemy pliki związane z tym procesem. Wyszukujemy plik z fragmentem w/w .exe w nazwie i odczytujemy hash’a md5. Niestety uzyskany hash nie pasował do klucza odpowiedzi. Wyczytaliśmy, że Volatility2 inaczej podchodzi do zapisu plików niż wersja 3. Niemniej jednak podejście jest wydajne się ok.

Pytanie nr. 9

Pytanie dotyczące LM hash'a Bob'a. Użycie komendy "hashdump" pozwoliło na odczytanie zawartości rejestru (gałęzi SAM i SYSTEM) z pamięci. Zidentyfikowano konto użytkownika Bob, dla którego hash LM ma wartość **aad3b435b51404eeaad3b435b51404ee**, czyli nie istnieje - przestarzała metoda hashowania haseł.

Q9 Solved : 2490

What is the LM hash of Bob's account?

\*\*\*\*\*

Hints Submit

Pytanie nr. 10

W tym pytaniu zadaniem jest sprawdzenie jakie uprawnienia ma konkretny węzeł VAD. VAD - Virtual Address Descriptor - część jądra windowsa mówiąca o pozwoleniach procesów w konkretnym zakresie pamięci. Do znajdowania informacji związanych z VAD wykorzystujemy komendę "vadinfo".

```
Select-String "0xfffffa800577ba10" -Context 0 5
Progress: 100.00 PDB scanning finished
> 820 svchost.exe 0xfffffa800577ba10 0x30000 0x33fff Vad PAGE_READONLY 0 0 0xfa800577c8e0 N/A Disabled
```

Q10 Solved : 2356

What memory protection constants does the VAD node at 0xfffffa800577ba10 have?

\*\*\*\*

Hints Submit

Pytanie nr. 11

Pytanie dotyczyło kolejnego node'a tym razem zaczynającego i kończącego się zgodnie z pytaniem. Podobne zadanie do potrzebnego, trzeba było zacząć od tego adresu.

Q11 Solved : 2305

What memory protection did the VAD starting at 0x00000000033c0000 and ending at 0x00000000033dffff have?

\*\*\*\*

Hints Submit

Pytanie nr. 12

Zadanie dotyczy nazwy skryptu VBS uruchomionego na maszynie. We wcześniejszych zrzutach ekranu widać przy procesie wscript.exe (pytanie nr. 4) widać od razu nazwę skryptu w folderze TEMP.

Q12 Solved : 2347

There was a VBS script that ran on the machine. What is the name of the script? (submit without file extension)

\*\*\*\*\*

Hints Submit

Pełni funkcję loader'a.

### Pytanie nr. 13

Zadanie polegało na znalezieniu uruchomionej aplikacji w konkretnym czasie. Użyto pluginu shimcache i przejrano wynik pod kątem danego wystąpienia. Shimcache zawiera metadane wszystkich uruchomionych, nawet historycznych programów .exe

```
Q13 ✔ Solved : 2254

An application was run at 2019-03-07 23:06:58 UTC. What is the name of the program? (Include extension)

*****


Skype.exe

💡 Hints Submit
```

## Pytanie nr. 14

W tym pytaniu pytają się nas o to co zostało napisane w notatniku bez jego zapisu. Z wcześniejszego zadania znamy już PID notatnika = 3032. Zadanie było ułatwione o tyle że w masce w zadaniu widać było znaki <> i że to właśnie chodzi o flag<>. Bez tego trzeba by było filtrować bardziej złożenie, żeby móc coś wyłapać (key, password itp.). Udało się to rozszyfrować za pomocą komendy Get-Content.


```
PS C:\Users\piotr\volatility3-develop> Get-Content -Path "C:\Users\piotr\Desktop\dumped\pid.3032.dmp" -Encoding Unicode | Select-String "flag<"
```


Q14  Solved : 2139

What was written in notepad.exe at the time when the memory dump was captured?

\*\*\*\*<\*\*\*\*\* \*\* \*\*\*\*>\_\_\_\_\_

flag<REDBULL\_IS\_LIFE>

 Hints

 Submit

## Pytanie nr. 15

W pytaniu 15 zadaniem jest znalezienie krótkiej nazwy pliku z rekordu 59045. Czyli trzeba znaleźć konkretny rekord z Master File Table (MFT). Definicja krótkiej nazwy:

Google

short name 8.3

AllImagesVideosShort videosNewsForumsMoreTools

AI Overview

"Short name 8.3" refers to the legacy file naming convention (8 characters for the name, 3 for the extension, like MYFILE.TXT ) from MS-DOS, used for backward compatibility in modern Windows systems, where long names create an 8.3 alias (e.g., MYFILE~1.TXT ) for older programs that can't handle long names.

Używając komendy "windows.mftscan.MFTScan" udało dostać się do szukanej danej.

```
PS C:\Users\piotr\volatility3-develop> python vol.py -f "C:\Users\piotr\Desktop\WRYCY_proj3\65-DumpMe\temp_extract_dir\Triage-Memory.mem" windows.mftscan |
Select-String "59045"
usage: vol.py [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]]
              [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG]
              [-o OUTPUT_DIR] [-q] [-f FILE] [--write-config]
              [--save-config SAVE_CONFIG] [--clear-cache]
              [--cache-path CACHE_PATH] [--offline] [-u URL]
              [--filters FILTERS] [--hide-columns [HIDE_COLUMNS ...]]
              [-r RENDERER] [--single-location SINGLE_LOCATION]
              [--stackers [STACKERS ...]]
              [--single-swap-locations [SINGLE_SWAP_LOCATIONS ...]]
              PLUGIN ...
vol.py: error: argument PLUGIN: plugin windows.mftscan matches multiple plugins (windows.mftscan.ADS, windows.mftscan.MFTScan, windows.mftscan.ResidentData)
PS C:\Users\piotr\volatility3-develop>
```

**Q15** Solved : 2139

What is the short name of the file at file record 59045?

\*\*\*\*\*~\* \*\*\*

EMPLOY~1.XLS

Hints Submit

Pytanie nr. 16

Ostatnie pytanie dotyczy zainfekowanego procesu, już wcześniej było wiadomo, że jest to PID = 3496.

**Q16** Solved : 2245

This box was exploited and is running meterpreter. What was the infected PID?

\*\*\*\*

3496

Hints Submit

Podsumowanie:

Analizowany incydent to wieloetapowy atak typu LoTL, który rozpoczął się od otwarcia złośliwego arkusza EMPLOY~1 . XLS, co umożliwiło wykonanie skryptu VBS `vhjReUDEuumrX` i pełne przejęcie kontroli nad systemem przez proces `UWkpjFjDzM.exe` (PID 3496) działający z agentem Meterpreter, prowadząc ostatecznie do kradzieży danych uwierzytelniających użytkownika Bob oraz zaawansowanej manipulacji strukturami pamięci VAD.