**Task 1 - Network Analysis**

For the network analysis of the packet capture of the device from the Sales department, I used Wireshark.

The packet capture was downloaded from malware-traffic-analysis.net.

<https://www.malware-traffic-analysis.net/2022/03/21/2022-03-21-Brazil-sourced-malspam-infection.pcap.zip>.

Graphical user interface, text, application

Description automatically generated

The format of the columns which is used by default is not the most efficient and does not display all the necessary info. Therefore we will make some changes to the columns, the time will be displayed in UTC format, source port,and destination port

Graphical user interface, text, application

Description automatically generated

The next step is to filter the results. In the URL bar I made a http.request

A picture containing application

Description automatically generated

**Task 2 - Memory Analysis**

**F1**

Get the volatility tool from volatilityfoundation.org and download it. Unzip the file into a location that can be easily accessed from cmd. Rename the executable file for easier usage in the cmd as well.

Download the Image and unzip the extracted file to the same directory of the volatility executable.

You are now ready to use the tool.

*Command: vol.exe -f ./"CA2 RAM Memory Capture for Task 2.raw" imageinfo*

This will determine the profile/machine type for the memory image.

Imageinfo offers a high-level summary of the sample for analysis. Through this, you can identify the operating system, service pack, and the system’s hardware architecture, i.e. 32 or 64 bit.

**Text

Description automatically generated**

Figure 1

The profile of the image can be found in the suggested profile section. For the initial analysis, we will use Win7SP1x86 as the image profile.

This tells us that the operating system was Windows 7, Special 32 bit version. This figure profile data will allow us to access further data about the image.

**F2**

*Command : vol.exe -f "CA2 RAM Memory Capture for Task 2.raw" --profile=Win7SP1x86 pslist*

This will display the processes in the image. It will retrieve critical data about each process, such as name, Process ID, Parent Process ID, Threads, Handles, Sessions, Wow64, and the start date and time of the process.

**Text

Description automatically generated**

Figure 2

Now we can review the result of this command and examineanything that is out of the ordinary. Some of the processes are related to the operating system and are necessary for the system’s operation. Others are for applications such as the browser or background tasks such as chrome.exe and GoogleUpdate.exe.

Using *| find /c /v ""* at the end of a command can tell us how many lines are generated from the command. Therefore, using it, we found the number of processes that existed.

Forty-seven(47) processes were active.

Text

Description automatically generated

Figure 3

**F3**

***How many files were open at the time? How many jpeg files were open at the time?***

To discover the number of files that open at the time we will use the filescan command. Filescan will find the open files even if they were hidden by a rootkit. The output of the command shows the physical offset of the file\_object, file name, number of pointers to the object, number of handles to the object, and the effective permissions granted to the object.

*Command: vol.exe -f "CA2 RAM Memory Capture for Task 2.raw" --profile=Win7SP1x86 filescan*

Text

Description automatically generated

*Command: vol.exe -f "CA2 RAM Memory Capture for Task 2.raw" --profile=Win7SP1x86 filescan | find /c /v ""*

Once again by counting the number of lines generated by the command, we can easily count the opened files in the memory image.

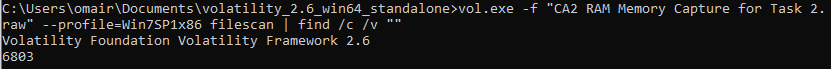
****

Figure 4

*Command: vol.exe -f "CA2 RAM Memory Capture for Task 2.raw" --profile=Win7SP1x86 filescan | findstr "jpg"*

To find the files which were formatted in the .jpg format I filtered the filescan result with the findstr command with jpg. This returns all the files which end with jpg.

Text

Description automatically generated

*Command: vol.exe -f "CA2 RAM Memory Capture for Task 2.raw" --profile=Win7SP1x86 filescan | findstr "jpg"| find /c /v ""*

To count the number of files with the .jpg format, we add the *| find /c /v “”* filter again. There were only 25 jpg files.

A screenshot of a computer

Description automatically generated with medium confidence

**F4**

***What searches appear in the Internet history? What browser was used in the search?***

To discover the number of files that open at the time, we will use the filescan command. Filescan will find the open files even if a rootkit hid them. The command’s output shows the physical offset of the file\_object, file name, number of pointers to the object, number of handles to the thing, and the effective permissions granted to the object.