hw4

dj$_{\text{dvorak}}$

April 4, 2020

Contents

1 Homework 4 MARE

1.1 (Chapter 6 from Practical Malware Analysis

2 Lab 06-01

2.1 Question 1

The main function calls in main is `InternetGetConnectedState`. If the host can connect to the internet, it calls an internal function `sub 0x40105F`, which proceeds to run this massive code construct.

2.2 Question 2

This massive code construct at `sub 0x40105F` that seems to be a Microsoft Visual C++ runtime Library call based on the it's complexity and numerous internal functions. There are lots of dynamic memory calls related to `malloc` or `heaps`.

2.3 Question 3

Performing dynamic analysis shows that the program simply prints "Connected to internet" if the can and "Error 1.1 No Internet" depending on the outcome of `InternetGetConnectedState`. One might conclude then, the funcition at `sub 0x40105F` is something along the lines of a `printf` statement from the system library.

3 Lab 06-02

3.1 Question 1

The first subroutine makes a call to check for internet connection.

3.2 Question 2

In the subroutine located at `0x40117F`, we see a call to call to another subroutine from a seemingly static library, `_stbuf` which seems to allocate some dynamic buffer. Then, there seems to be some file handling and file pointer parsing. Finally, the `_frbuf` seems to free the dynamically allocated buffer. Looks like an unlabeled `printf()` call again.

3.3 Question 3

The second call is pretty obvious. It tries to connect to an external URL, `practicalmalwareanalysis`, read 200h bytes and then parse it for some kind of command. Looks like it's parsing HTML code. There is a series of checks that checks for the string "<!–". The argument is saved in register AL.

3.4 Question 4

A series of if conditionals are used.

3.5 Question 5

Looking at the imports, we see that there are strings like `InternetReadFile` and `Success Internet Connection`. There's also a URL pointed to the `practicalmalwareanalysis.com`. There's also a string for `Internet Explorer 7.5`. We can probably check for any traffic to this URL with this browser.

3.6 Question 6

Looks like the program tries to connect to the internet, connect get some HTML page. Parse a comment for a command and then sleep for a minute.

4 Lab 06-03

4.1 Question 1

Compared to the function call from Lab 06-02, there is a new subroutine that plays with registers, `sub_401130`. It tries to open a register of windowsversion and set the key to "malware" and the data to point to some file at `C:\Temp\cc.exe`. There is also a jump table which seems take a file and drop it into that specific directory. Seems highly suspicious.

4.2 Question 2

The new function takes a pointer to a string containing a filename. Also an argument, which is probably the command parsed from the HTML source from before.

4.3 Question 3

The construct is a jump table. Probably a switch. Can do things like create new directory, copy file, delete file, set registers or sleep or exit.

- New dir
- Copy
- delete
- set regs
- sleep
- exit (error parsing)

4.4 Question 4

This function can change register values. Sets the `malware` value in `CurrentVersion\Run` to `C:\Temp\cc.exe`.

4.5 Question 5

Seems to open a link to `http://practicalmalwareanalysis.com/cc.htm` and also creates a file at `C:\Temp\cc.exe`.

4.6 Question 6

Purpose seems to grab a file from the internet and copy it into to known location for future use. Allows persistence for backdoor commands.

5 Lab 06-04

5.1 Compare the `main` method with that of 06-02, 06-03.

Well, this one seems to be built of 06-03 with a new loop mechanism.

5.2 What new code has added to main()?

There is now a loop function for attempting to parse the command. This loop iterates 1440 times after the internet connection passes.

5.3 How is this HTML parse function different from the previous labs?

This parse function is a little different in the user agent has a %d. There is a call to print string which seems to be the loop iteration of the program prefixed with `pma`.

5.4 How long will the program run?

Runs for 1440 iterations with each iteration lasting a minute. This means the program runs for an entire day.

5.5 Network-based indicators?

Network based indicators is a repeated request to `http://practicalmalwareanalysis.com/cc.htm` using the `Internet Explorer 7.5` client agent.

5.6 What is the purpose of this malware?

Perform realtime coordination of a potential botnet for a single day.