CMSC 491/691 Malware Analysis Midterm

Name: Daniel Roh
Due: 3/30/2020 by 5:00pm

Download and extract midterm2020.7z on a virtual machine. The password to the zip file is "infected", without the quotes. The file contains midterm2020q1.exe and midterm2020q2.exe, which are live malware samples.

How to turn this in for grading: You can edit your answers right into this file. Submit the homework a .docx or .pdf to RJ via email.

The midterm is graded out of 100 points, and there are 110 points available.

Part 1: midterm2020q1.exe (50 pts)

1) List three indicators that midterm2020q1.exe is packed. What packer was used to pack midterm2020q1.exe? (10 pts)

- **a.** In viurstotal, the section names are not standard names (UPX0, UPX1) as well as the sizes of the sections are also odd. The raw size is 0 but the virtual size is 49,152.

- **b.** The entropy for one of the sections is 7.52, which is a indicator of packing

- **c.** The file imports kernel32.dll which uses LoadLibraryA, GetProcAddress, which means that this file is using runtime linking. Which is also an indication that the file is packed to hide imports

Name of packer: UPX(3.95)

2) Unpack midterm2020q1.exe and describe how you unpacked it. What is the md5 of the unpacked file? (7 pts)

To unpack midterm2020q1.exe, I went to the file location and opened a CMD window. From there, ran the command "ipx -d midterm2020q1.exe" to unpack the file. IPX automatically was able to find the packer that was used and unpacked the file

MD5 Hash: 2486b321908578ca531fee0cd775bc0e

Follow the FakeNet-NG guide to configure the network settings of your Flare VM. You should only run the malware while your VM is not connected to the internet! Run midterm2020q1.exe and answer questions 3 - 6.

3) What port does midterm2020q1 listen on? How did you find this information? (8 pts)

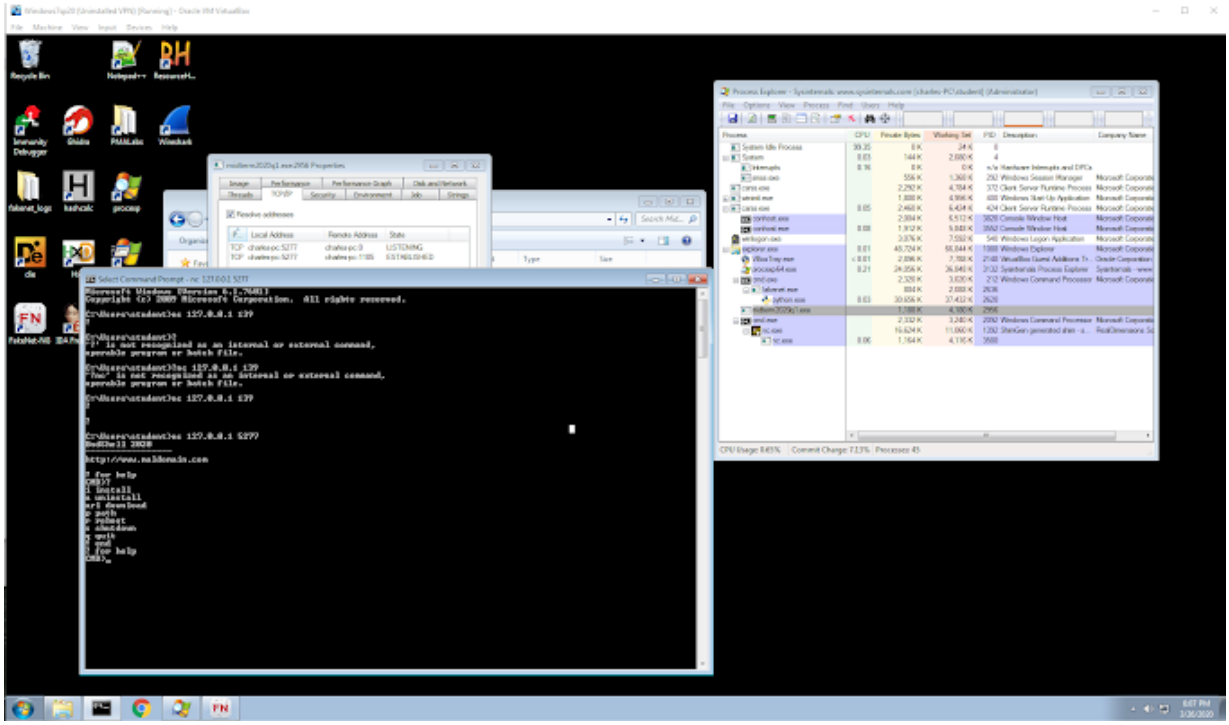
Port: 5277
This information was found by using process explorer and looking at the properties of midterm2020q1.exe while it was running. In the properties of miterm2020q1.exe in process explorer, under the TCP/IP tab, shows the port that midterm2020q1.exe was using 5277 to listen.

The malware sample starts a telnet server on the port you found in question 3. This allows the

malware actor to gain remote access to the infected computer. While midterm2020q1.exe is running, open a command prompt and run the following command:

`nc 127.0.0.1 [port number from question 3]`

4) Once you have run the nc command, type "?" to list the different commands that the midterm2020q1.exe can execute. Provide a screenshot of the list of commands. (7 pts)



5) Use the "i" command to make midterm2020q1.exe install itself. What is the registry key that the malware uses for persistence? What is the full path of the file that is made persistent? How did you find this information? (10 pts)

Path:
HKU\S-1-5-21-419295087-873244694-3473264175-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}

This information was found by running regshot before running the installation and taking a snapshot. Then after the install was completed, taking another snapshot and then running the compare operation and seeing what was changed

6) What Windows privilege does midterm2020q1.exe require in order to successfully execute the "s" command? How did you find this information? (8 pts)

Midterm2020q1.exe requires the privilege of "SeShutdownPrivilege" to successfully run the "s" command.

This information was found by taking a look at midterm2020q1.exe's properties under process explorer. Then taking a look at the security tab and looking at the bottom at the windows privileges that were given to the .exe. There, the only non-default enabled privilege was "SeShutdownPrivilege"

in which given that the “s” command shuts down the system, it would make sense that this privilege would be required to allow the command to work.

Part 2: midterm2020q2.exe (30 pts)

Use Ghidra to answer questions 1 - 6 about midterm2020q2.exe.

1) How many DLL files does midterm2020q2.exe import functions from? (3 pts)

Midterm2020q2.exe imports 4 dll files.

2) What’s the address of the string “ntdll.dll”? (3 pts)

Address: 004044b0

3) How many times is FUN_00401d30 called? (3 pts)

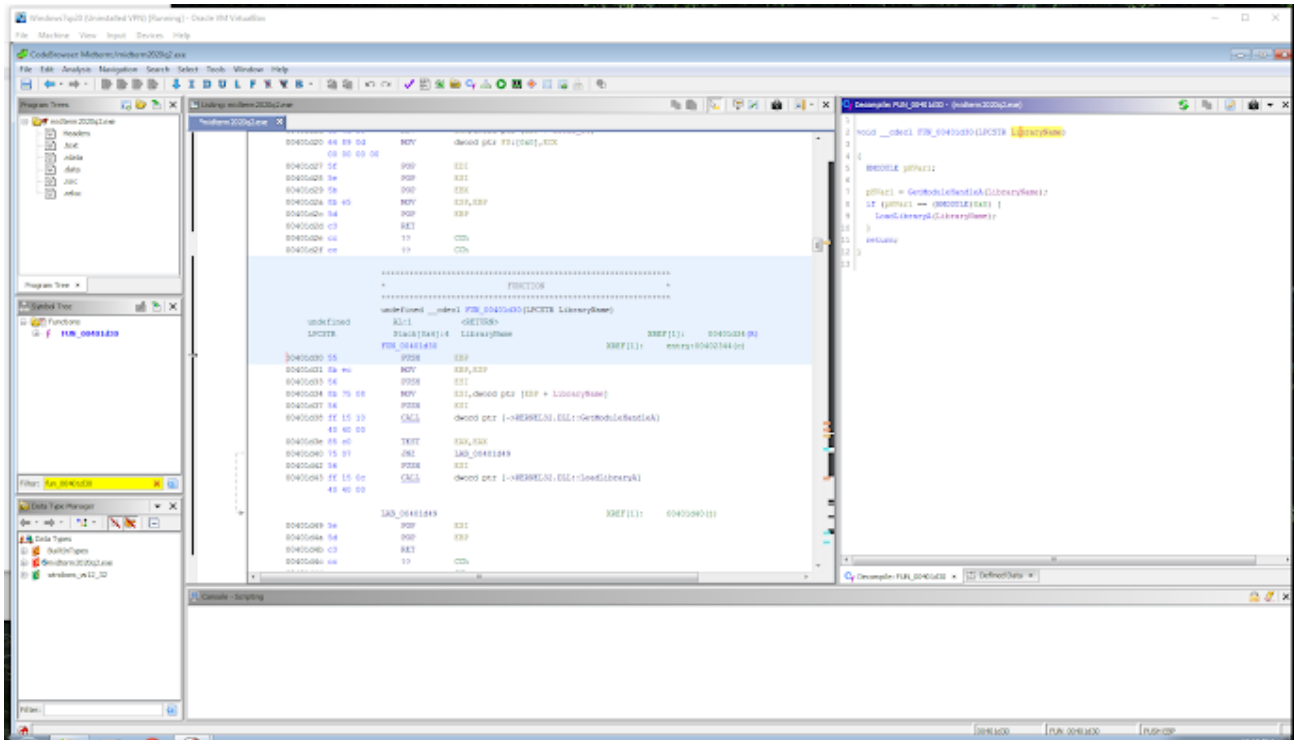
Function FUN_00401d30 is called once

4) How many arguments are passed to FUN_00401d30? (3 pts)

Function FUN_00401d30 passes 1 argument

5) Rename param_1 in FUN_00401d30 to something more descriptive. Justify the variable name you chose and provide a screenshot. (8 pts)

Param_1 was renamed to LibraryName. I choose to use this variable name due to the variable being used on a load library path function later on in the code. So likely the variable is a library that the code is trying to load in for run time linking.



6) Describe what the code between 0x0040239d and 0x004023b9 does. (10 pts)

The code is checking the type of environment that the code is being run in. In this case, the code checks if the process is being run in a x64 environment. The code continues if the code is being run in a x64 environment or exits if otherwise. This is likely due to the code using libraries requiring the x64 instruction set.

Part 3: Assembly (30 pts)

Answer questions 1 - 3 about the assembly program below.

```
start:                                  /*Function set up*/
      push   ebp                        //Store base pointer
      mov    ebp,esp                    //Move base pointer
      sub    esp,0x10                   //Move stack pointer
      mov    DWORD PTR [ebp-0x4],0x0 //Function argument
      jmp    loc_2

Loc_1:                                  /*Loop*/
      mov    edx,DWORD PTR [ebp-0x4] //Get function argument arg_0
      mov    eax,DWORD PTR [ebp+0x8] //Get local variable
      add    eax,edx                    //Place the character byte location in eax
      movzx  eax,BYTE PTR [eax]         //Get byte from eax
      lea    ecx,[eax-0x11]             //Subtract 17 from eax and load in to ecx
      mov    edx,DWORD PTR [ebp-0x4] //Get function argument
      mov    eax,DWORD PTR [ebp+0x8] //Get local variable
      add    eax,edx                    //Place the character byte location in eax
      mov    edx,ecx                    //Place decoded character in edx
      mov    BYTE PTR [eax],dl       //Place decoded character into eax
      add    DWORD PTR [ebp-0x4],0x1 //Point to next character in arg_0

loc_2:                                  /*Loop Check*/
      mov    edx,DWORD PTR [ebp-0x4] //Get function argument arg_0
      mov    eax,DWORD PTR [ebp+0x8] //Get local argument
      add    eax,edx                    //Place a string in eax
      movzx  eax,BYTE PTR [eax]         //Get a byte of string from eax
      test   al,al                   //Test if 0 or NULL
      jne    loc_1                   //Move to loc_1 if al != al
      nop
      leave
      ret
```

1) In a few sentences, explain what the function does. (7 pts)

This function takes an encoded byte string and goes character by character and subtracts 0x11 from the character to decode the character. Once the character is decoded, it is placed back into the string. The loop continues until a NULL character is reached at the end of the string and then exits.

2) Write a function in C that's equivalent to the above assembly. (10 pts)

```
void func(char* arg_0){
        int x = 0; //counter
        char temp;

        while(arg_0[x] != NULL){
                temp  = arg_0[x] - 0x11; //Decode
                x++;
        }
}
```

3) Let arg_0 be a pointer to the null terminated string "\x72\x84\x84\x76\x7e\x73\x7d\x8a\x31\x7a\x84\x7f\x38\x85\x31\x73\x72\x75". What is the value of the string pointed to by arg_0 when the function completes? (8 pts)

The value of the string pointed to by arg_0 is 100 which is the last character on the string "d". As the pointer is last updated before hitting the null.