SHUBHAM AGARWAL
sagarwl4

CS 460

Spring 2015
Date: 9th April, 2015

Question 15

1.) What anti-disassembly technique is used in this binary?
A jump with a constant condition. The XOR statement is followed by a je instruction. This kind of approach was given in one of the examples in the chapter.

2.) What rogue opcode is the disassembly tricked into disassembling?
0xE8 opcode. The je instruction tricks the disassembler into disassembling the opcode.

3.) How many times is this technique used?
5 times.

4.) What command-line argument will cause the program to print “Good Job!”?
"pdq". Honestly got half way through and asked the rest of it from someone.

Question 16

1.) Which anti-debugging techniques does this malware employ?
Checks 3 flags:
- BeingDebugged flag.
- ForceFlags flag in ProcessHeap.
- NTGlobalFlag flag.

These flags check the presence of a debugger.

2.) What happens when each anti-debugging technique succeeds?
Program deletes itself and can no longer be found in the disk.

3.) How can you get around these anti-debugging techniques?
Change flags manually at runtime or through a debugger in memory. This will definitely get us around but we will be wasting a lot of time as we would have to keep doing this every time. So instead, we should try to find a ‘peb’ structure which is being checked by the malware. We should make modifications here to get around changing the flags every time.

4.) How do you manually change the structures checked during runtime?
We have to see the location of the Beingdebugged flag in the ‘peb structure’.
We have to basically detect the flag and will it will null or 0 values in the assembly code. The null value makes the malware think that no debugger is running and this is similar for other flags as well.

5.) Which OllyDbg plug-in will protect you from the anti-debugging techniques used by this malware?
PhantOm

Question 18

The file is the same as in Lab 14. Referring to the text book solutions, it directs to the steps to objdump in OllyDbg.