Ng Jong Ray, Edward a0216695u e0540252@u.nus.edu

HW-B-1.exe
(Hint: printf() is 0x401089, main() is 0x401060)
1. Focus on the main() method. What is stored in EAX prior to the function call at 0x40107D?
It is the first character of the argument of the executable.
2. Focus on the function that starts at 0x401000. What does 0x62, 0x63, 0x73 likely correspond to?
They correspond to the ascii characters “b”, “c” and “s”.
3. What coding construct is likely a major part of this function?
If - else statements
4. What does this function (0x401000) do?
It checks the first character of the argument against the ascii characters “b”, “c” and “s” and then output “bruges”, “chicago” and “seoul” respectively. If it does not match with any of the 3 above characters, then it will output “no city”.
5. What does this overall program do?
The program reads in the character, compares it against “b”, “c” and “s” and then uses the printf function to output “bruges”, “chicago”, “seoul” or “no city”.

Ng Jong Ray, Edward a0216695u e0540252@u.nus.edu

HW-B-3.exe
1. What is the subroutine located at 0x40117F?
It is the printf function.
2. What does the second subroutine called by main do?
This subroutine sub_401040 attempts to open the URL http://www.practicalmalwareanalysis.com/cc.htm, it will read the first 0x200 bytes of the webpage into a buffer. If this is not possible it will throw 3 errors, either “Error 2.1: Fail to OpenUrl\n”, “Error 2.2: Fail to ReadFile\n” and “Error 2.3: Fail to get command\n”.
3. What type of code construct is used in this subroutine?
If else statements as seen from the many jumps. Arrays are also used during the comparisons.
4. Are there any network-based indicators for this program?
If the url http://www.practicalmalwareanalysis.com/cc.htm is accessed and with the user agent Internet Explorer 7.5/pma
5. What is the purpose of this malware
The malware will check if there is internet connection, if there is then it will try to open the URL http://www.practicalmalwareanalysis.com/cc.htm and if it is successful, it will read in the first 0x200 bytes into a buffer. It will then check the start of the buffer for the characters ‘<!–’. The program will throw an error if something did not work at any stage.