CPE457/557 Software Reverse Engineering

**05 - WINDOWS API**

# Understanding the Windows API

**Lab Description:** The following lab aims to gauge student’s ability to properly analyze executables for malicious patterns via identification of calls made under the Windows API. This may include:

* MSDN documentation
* Previous lessons (e.g. Basic Analysis Techniques, PE)
* Using static analysis tools
* Dynamic analysis

**Lab Environment:** It is recommended that the VM environment from the lesson, “Creating a Safe Lab Environment” be used. The following utilities are recommended, these all require a Windows OS to run.

* PE Studio: https://www.winitor.com/
* Dependency Walker: http://www.dependencywalker.com/

**Lab Files that are Needed:** The malicious executable should be found within the zip, “Windows\_api\_lab\_files.zip”.

### **Lab:**

Because API calls provide us with an immense knowledge of what a program does, and how it does it, it’s important that we know how to gather it. Using Dependency Walker, or your preferred PE parsing tool, answer the following questions.

**Sample 1: MD5 - f2d89336f493f4e1cc0324b81a73b6fe**  
**WARNING:** This sample is malicious, please handle accordingly.

1. Using a PE parsing tool, identify all the imported libraries. What are they?
2. Which library (or libraries) indicate that this sample will connect to the internet?
3. Based on your answer to question b (previous), what APIs does the sample import from that library? Describe what capability each API provides the sample.
4. What other evidence can you identify that this sample will utilize the internet for command and control? Be specific and provide the evidence along with a description.
5. Can you identify any domains or IP addresses this sample will use? Why or why not?

**Sample 2: MD5 - 30d3307779016426d24f3077d8f33514  
  
WARNING:** This sample is malicious, please handle accordingly.

1. Analyze this file using a PE parsing utility, what do these tools tell you about this sample?
2. What libraries and APIs does this sample import? If you can’t identify any, why do you suspect that is?  
     
   When parsing this sample, it appears dramatically different than the previous sample. Provide some discussion on specific reasons you think that is. How would this change your continued analysis of this sample?

## WHAT TO SUBMIT

Submissions should be neatly organized and formatted. Each question should provide a screenshot and a brief description, if necessary, to aide the screenshot.

**Binary 1**

1. wininet.dll, kernel32.dll

2. wininet.dll

3. InternetOpenUrlA: opens some type of resource specified by a url

InternetCloseHandle: simply closes an internet handle. You pass in the handle that you want to close.

InternetOpenA: Initializes the programs of windows internet functions.

4. In the Virus Total Section, certain engines detect something along the lines of win32.upatre.nh. A quick google search explains that the file is a Trojan that connects to various url’s.

5. I can’t see the urls, but I assume that the url is being built at runtime because there are arguments being passed into InternetOpenUrlA.

**Binary 2**

1. There are no visible imports and the strings are all gibberish.

2. I cannot identify any imports or libraries. This is likely because the sample is statically linked. It is likely a Trojan that contains a packed payload. More evidence that supports this is that certain signatures in the Virus Total Section indicate that a packer is present.

Packing is a way to essentially compress an executable. It is decompressed at runtime. This explains why PE studio does not detect any imports or libraries as well as why all of the strings appear to be random text.

The best way to deal with this going forward would be to learn how to manually unpack binaries.

You could also use IDA’s lumina to figure out the name of any imports.