CS4238 Lab: (1) Basic Static Analysis

The **goal** of this lab's **first part** is to get familiar with **basic static analysis techniques** (including hashing, strings, packers, packer detection, header inspection) of PE files.

Lab Set-Up

You will need a FireEye Flare VM for this lab. You may either:

- Install manually via Github
- Download a prepared image for virtualbox via Google Drive.

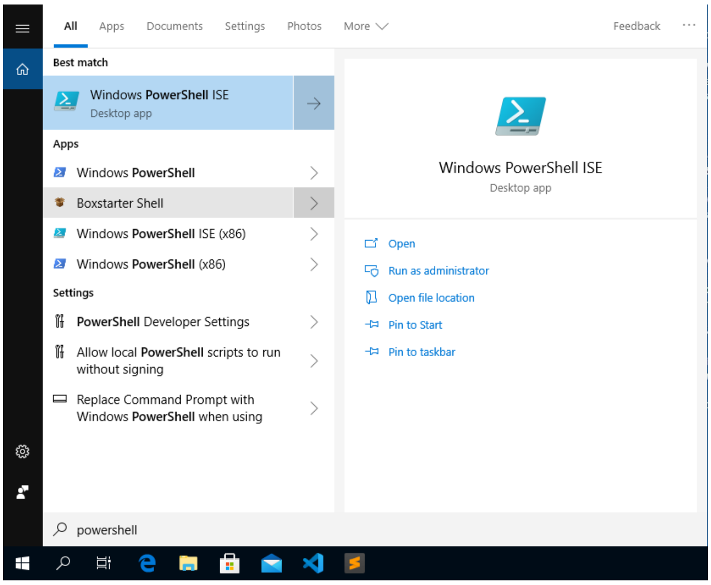
The VM password is "*Passw0rd!*". To learn more about the Flare VM, you can check this video: https://www.youtube.com/watch?v=B7PEDJV4ouM.

A helloword PE file will be used for analysis (Download `pe_helloworld.zip` from Canvas' Files).

Task 1-1: Using `Strings`

The goal of this task is to use **`strings` command**, as well as to get familiar with the Flare VM environment.

1. Click the *search* icon in the lower left corner.
2. Search for *Powershell*, and run it.



3. Commands such as `cd` and `ls` also work in *Powershell.*
4. In the terminal, run `strings` command on a PE file:

   `strings` *<PE-filename>*

Task 1-2: Hashing

The goal of this task is to perform a **hashing** on PE files.

1. In PowerShell, run:
   `Get-FileHash` *<PE-filename>*

2. By default the command uses SHA256.
   To change the hash algorithm, use `-Algorithm` option:
   `Get-FileHash -Algorithm` *<MD5/SHA1/...>* *<PE-filename>*

3. For more examples, please check link.

Task 1-3: Using Packer and Packing Detection

This task aims to **pack** a PE file using UPX, and **detect** whether a file is packed using `PEiD`. You can additionally **unpack** the packed file.

1. First, create a file backup before packing your PE file:

   `cp` *<PE-filename>* *<PE-filename.bak>*

2. Pack the PE file:

   `upx` *<PE-filename>*

3. Test the packed PE file:
   `upx -t` *<PE-filename>*

4. Run the packed PE file and check if it is still functional.

5. Check the packed PE file's information using `PEiD`.
   `PEiD.exe`

6. (*Optional*) Unpack the packed PE file:
   `upx -d` *<PE-filename>*

   Is the resulting unpacked PE file the same as the original one?
   Compare them by hashing!

Task 1-4: Inspecting PE Header

The goal of this task is to **browse PE header** in a PE file.

1. Use `PEView` or `CFF Explorer` to load the original PE file.
2. Answer the following questions:
   a. Is it a 32-bit executable file or a 64-bit one?
   b. What is the timestamp of this file?
3. Further, check the `.rsrc` section by using `Resource Hacker`.

CS4238 Lab: (2) Basic Dynamic Analysis

The **goal** of this lab's **second part** is to get familiar with **basic dynamic analysis techniques**, including running DLLs, and process monitoring.

Lab Set-Up

You will need the **FireEye's Flare VM** for this lab. A HelloWorld DLL file will be used for analysis (get `rundll_example.zip` from Canvas' Files).

Task 2-1: Running DLLs

The goal of this task is to **run** a function inside a given DLL file named `helloworld_dll.dll`.

1. `helloworld_dll.dll` has two functions: `func_1()` and `func_2()`. You can check its given source code for details.
2. In Powershell, use `rundll32.exe` to run `func_1()` as follows.

   ```
   rundll32.exe helloworld_dll.dll, func_1
   ```

Task 2-2: Process Monitoring

The goal of this task is to perform a **process monitoring** of your PE file using either :

- Process Monitor (download it from here); or
- Process Hacker.

The steps are:

1. Run the command in Step 2 of Task 2-1, and inspect the processes.
2. How many related processes are there?
   What's their process tree look like?

Task 2-3: Registry-Activity Monitoring

The task aims to **monitor** registry changes.

1. Open Regshot, then create the 1st registry shot.
2. Download and install FakeNet.
3. Create the 2nd registry shot in Regshot.
4. Compare two shots, and print the output.
   Did the software that you installed modify the registry?

Task 2-4: Network-Activity Monitoring

The goal of this basic task is to **monitor DNS requests** and **replies** using a **fake DNS**.

1. Download ApateDNS.
2. Run ApateDNS and attach it to the suitable network adaptor.
3. Start the server.
4. Ping www.google.com in the command line.
5. Change the Reply IP to 127.0.0.1.
6. Restart the server and make another ping.
   Observe the results.

The optional advanced task is to **trick & monitor** the malware's all network traffic including DNS using **FakeNet**.

1. Run FakeNet as administrator.
2. Run ping again.
3. Change the reply DNS IP in `./config/default.ini` under `[DNS Server]`.
4. For other usages on other protocols, please check their GitHub.