Phòng thí nghiệm An toàn Thông tin InsecLab
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: Cơ chế hoạt động của mã độc

Tên chủ đề: DLL injection

GVHD: Ngô Đức Hoàng Sơn

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.O21.ANTT.1

STT	Họ và tên	MSSV	Email
1	Nguyễn Đình Luân	21521105	21521105@gm.uit.edu.vn
2	Trần Thanh Triều	21522713	21522713@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Nội dung	Tình trạng
1	Yêu cầu 1	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Link video demo: https://youtu.be/-JtEjvYjtEE

Yêu cầu 1:
Đầu tiên ta phân tích code file source.cpp:

Ta thấy đầu tiên chương trình thực hiện lấy các thông tin về tiến trình:

```
HWND hGameWindow = FindWindow(NULL, L"Age of Empires Expansion");
if (hGameWindow == NULL) {
    std::cout << "Start the game!" << std::endl;
    return 0;
}
DWORD pID = NULL; // ID of our Game
GetWindowThreadProcessId(hGameWindow, &pID);
HANDLE processHandle = NULL;
processHandle = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pID);
if (processHandle == INVALID_HANDLE_VALUE || processHandle == NULL) { // error handling
    std::cout << "Failed to open process" << std::endl;
    return 0;
}

TCHAR gameName[13];
wcscpy_s(gameName, 13,L"EMPIRESX.EXE");

DWORD gameBaseAddress = GetModuleBaseAddress(gameName,pID);

std::cout << "debugginfo: gameBaseAddress = " << gameBaseAddress << std::endl;

DWORD offsetGameToBaseAdress = 0x003C4B18;
std::vector<DWORD> pointsOffsets{ 0x3c, 0x100, 0x50, 0x0 };


DWORD baseAddress = NULL;
```

Tiếp theo thực hiện đọc và thao tác trên bộ nhớ:

```
//Get value at gamebase+offset -> store it in baseAddress
ReadProcessMemory(processHandle, (LPVOID)(gameBaseAddress+ offsetGameToBaseAdress), &baseAddress, sizeof(baseAddress), NULL);
std::cout << "debugginfo: baseaddress = " << std::hex << baseAddress << std::endl;

DWORD pointsAddress = baseAddress; //the Adress we need -> change now while going through offsets
for (int i = 0; i < pointsOffsets.size() - 1; i++) // -1 because we dont want the value at the last offset
{
    ReadProcessMemory(processHandle, (LPVOID)(pointsAddress + pointsOffsets.at(i)), &pointsAddress, sizeof(pointsAddress), NULL);
    std::cout << "debugginfo: Value at offset = " << std::hex << pointsAddress << std::endl;
}
pointsAddress += pointsOffsets.at(pointsOffsets.size() - 1); //Add Last offset -> done!!
float currentPoint = 0;

std::cout << sizeof(currentPoint) << std::endl;
ReadProcessMemory(processHandle, (LPVOID)(pointsAddress), &currentPoint, sizeof(currentPoint), NULL);
```

Đọc giá trị và thay thế giá trị food:

```
std::cout << sizeof(currentPoint) << std::endl;
ReadProcessMemory(processHandle, (LPVOID)(pointsAddress), &currentPoint, sizeof(currentPoint), NULL);
std::cout << "The last address is:" << pointsAddress << std::endl;
std::cout << "Current value is:" << currentPoint << std::endl;
//"UI"
std::cout << "Age of Empires Hack" << std::endl;

std::cout << "How many points you want?" << std::endl;
float newPoints = 0;
std::cin >> newPoints;
WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);
```

Phân tích file injector.cpp

- Lấy tên process và đường dẫn file dll

```
int main()
{
    HANDLE hProcess;
    LPVOID pszLibFileRemote = NULL;
    HANDLE handleThread;
    const wchar_t* process = L"EMPIRESX.EXE";
    int pID = getProcId(process);

    char dll[] = "AOEResourceHack.dll";
    if (!exist(dll)) {
        std::cout << "debuginfo: Invalid DLL path!" << std::endl;
    }
```

- Tiếp theo là ghi vào bộ nhớ và tạo các thread hỗ trợ việc chèn DLL:

```
char dllPath[MAX_PATH] = { 0 }; // full path of DLL
GetFullPathNameA(dll, MAX_PATH, dllPath, NULL);
std::cout << "debuginfo: Full DLL path: " << dllPath << std::endl;


hProcess = OpenProcess(PROCESS_QUERY_INFORMATION |PROCESS_CREATE_THREAD | PROCESS_VM_OPERATION |PROCESS_VM_WRITE, 0, pID);

if (hProcess) {
    pszLibFileRemote = VirtualAllocEx(hProcess, NULL, strlen(dllPath) + 1, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
}
else {
    std::cout << "error: Could not open process handle with id:" << pID << std::endl;
}
if (pszLibFileRemote == NULL) std::cout << "error: Cannot allocate memory" << std::endl;

int isWriteOK = WriteProcessMemory(hProcess, pszLibFileRemote, dllPath, strlen(dllPath) + 1, NULL);
if (!isWriteOK) std::cout << "error: Failed to write" << std::endl;

handleThread = CreateRemoteThread(hProcess, NULL, NULL, (LPTHREAD_START_ROUTINE)LoadLibraryA, pszLibFileRemote, NULL, NULL);
if (handleThread == NULL) {
    std::cout << "error: Failed to create thread" << std::endl;
    ErrorExit(_T("CreateRemoteThread"));
}

WaitForSingleObject(handleThread, INFINITE);
CloseHandle(handleThread);
VirtualFreeEx(hProcess, dllPath, 0, MEM_RELEASE);
CloseHandle(hProcess);

return 0;
```

Chỉnh sửa file dllmain.cpp để có thể thực hiện tăng điểm khi nhấn phím F6
Đầu tiên ta sửa lại tên của process trong 2 file injector và source từ Empiresx.exe thành EMPIRESX.EXE:
injector.cpp:

```
HANDLE handleThread;
const wchar_t* process = L"EMPIRESX.EXE";
int pID = getProcId(process);
```

source.cpp:

```
TCHAR gameName[13];
wcscpy_s(gameName, 13,L"EMPIRESX.EXE");

DWORD gameBaseAddress = GetModuleBaseAddress(gameName,pID);
```

Tiếp đó ta copy toàn bộ nội dung file source.cpp dán vào file dllmain.cpp:

```
// dllmain.cpp : Defines the entry point for the DLL application.
#include "pch.h"
#include <Windows.h>

#include <Windows.h>
#include<TlHelp32.h>
#include <iostream>
#include <tchar.h> // _tcscmp
#include <vector>


DWORD GetModuleBaseAddress(TCHAR* lpszModuleName, DWORD pID) {
    DWORD dwModuleBaseAddress = 0;
    HANDLE hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE, pID); // make snapshot of all modules within process
    MODULEENTRY32 ModuleEntry32 = { 0 };
    ModuleEntry32.dwSize = sizeof(MODULEENTRY32);

    if (Module32First(hSnapshot, &ModuleEntry32)) //store first Module in ModuleEntry32
    {
        do {
            if (_tcscmp(ModuleEntry32.szModule, lpszModuleName) == 0) // if Found Module matches Module we look for -> done!
            {
                dwModuleBaseAddress = (DWORD)ModuleEntry32.modBaseAddr;
                break;
            }
        } while (Module32Next(hSnapshot, &ModuleEntry32)); // go through Module entries in Snapshot and store in ModuleEntry32


    }
    CloseHandle(hSnapshot);
    return dwModuleBaseAddress;
}


int increaseFoodBy100() {

    HWND hGameWindow = FindWindow(NULL, L"Age of Empires Expansion");
    if (hGameWindow == NULL) {
        std::cout << "Start the game!" << std::endl;
```

Thực hiện chỉnh sửa tên hàm main thành increaseFoodBy100:

```
int increaseFoodBy100() {

    HWND hGameWindow = FindWindow(NULL
    if (hGameWindow == NULL) {
        std::cout << "Start the game!"
        return 0;
```

Chỉnh sửa việc nhập chỉ số thức ăn thành tự động cộng thêm 100 vào giá trị trước đó:

```
  //"UI"
 // std::cout << "Age of Empires Hack" << std::endl;

 // std::cout << "How many points you want?" << std::endl;
  float newPoints = currentPoint + 100;
  //std::cin >> newPoints;
  WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);
```

Cuối cùng thêm việc chạy hàm increaseFoodBy100 khi ta bấm F6:

```
DWORD WINAPI MainThread(LPVOID param) {
    while (true) {
        if (GetAsyncKeyState(VK_F6) & 0x80000) {
            //MessageBoxA(NULL, "F6 Pressed!", "F6 Pressed!", MB_OK);
            increaseFoodBy100();
        }
        Sleep(100);
    }
    return 0;
```

Kết quả Demo ở video