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

BÁO CÁO THỰC HÀNH

Môn học: Cơ chế mã độc

Tên chủ đề: DLL Injection

GVHD: Ngô Đức Hoàng Sơn

Nhóm: 12

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 Triệu Thiên Bảo	21520155	21520155@gm.uit.edu.vn
2	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn
3	Huỳnh Minh Khuê	21522240	21522240@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Nội dung	Tình trạng	Trang
1	Yêu cầu 1	100%	2 – 6
Điểm tự đánh giá			**?/10**

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

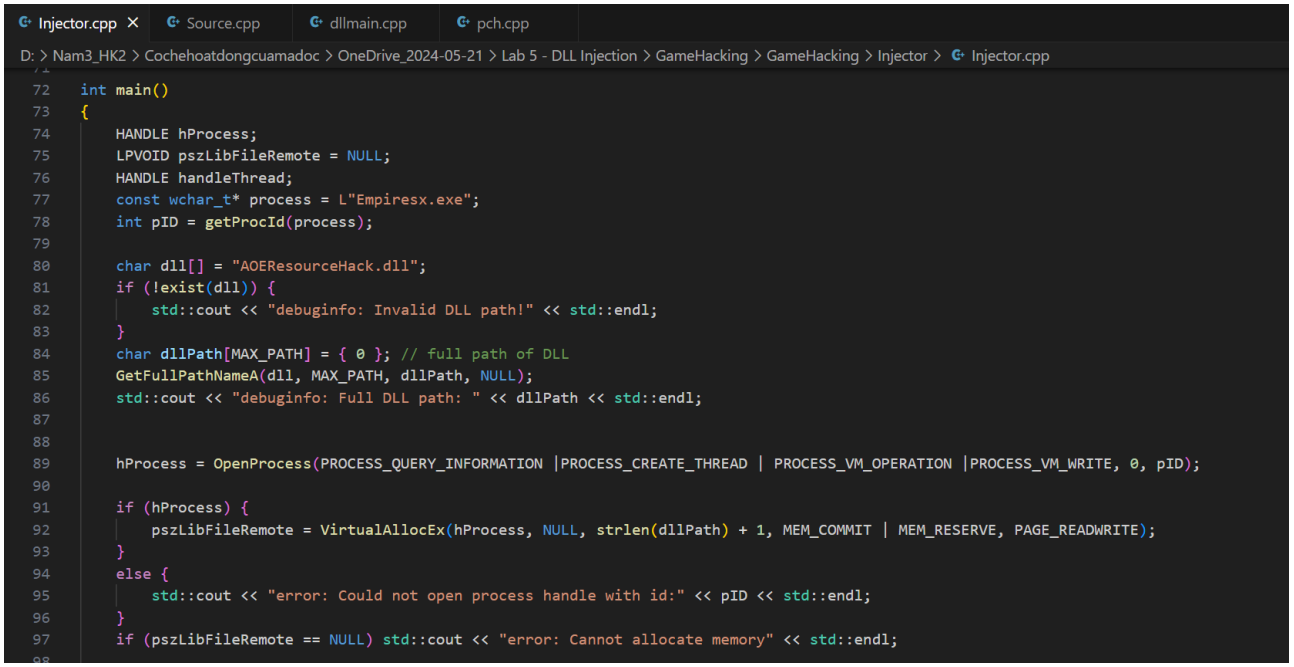
[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Link video thực hiện:
https://www.youtube.com/watch?v=R9QITAEV9SA

Các file cần để thực hiện chèn code vào file EMPIRESX.EXE:

File Injector.cpp



File Source.cpp

```
Injector.cpp | Source.cpp | dllmain.cpp | pch.cpp
D: > Nam3_HK2 > Cochehoatdongcuamadoc > OneDrive_2024-05-21 > Lab 5 - DLL Injection > GameHacking > GameHacking > GameHacking > Source.cpp
31  int main() {
37      }
38      DWORD pID = NULL; // ID of our Game
39      GetWindowThreadProcessId(hGameWindow, &pID);
40      HANDLE processHandle = NULL;
41      processHandle = OpenProcess(PROCESS_ALL_ACCESS, FALSE, pID);
42      if (processHandle == INVALID_HANDLE_VALUE || processHandle == NULL) { // error handling
43          std::cout << "Failed to open process" << std::endl;
44          return 0;
45      }
46
47      TCHAR gameName[13];
48      wcscpy_s(gameName, 13,L"Empiresx.exe");
49
50      DWORD gameBaseAddress = GetModuleBaseAddress(gameName,pID);
51
52      std::cout << "debugginfo: gameBaseAddress = " << gameBaseAddress << std::endl;
53
54      DWORD offsetGameToBaseAdress = 0x003C4B18;
55      std::vector<DWORD> pointsOffsets{ 0x3c, 0x100, 0x50, 0x0 };
56
57
58      DWORD baseAddress = NULL;
59
60      //Get value at gamebase+offset -> store it in baseAddress
61      ReadProcessMemory(processHandle, (LPVOID)(gameBaseAddress+ offsetGameToBaseAdress), &baseAddress, sizeof(baseAddress), NULL);
62      std::cout << "debugginfo: baseaddress = " << std::hex << baseAddress << std::endl;
```

File maindll.cpp:

```
Injector.cpp | launch.json | c_cpp_properties.json | dllmain.cpp | Source.cpp
GameHacking > GameHacking > AOEResourceHack > dllmain.cpp > MainThread(LPVOID)
1   // dllmain.cpp : Defines the entry point for the DLL application.
2   #include "pch.h"
3   #include <Windows.h>
4
5   #include <Windows.h>
6   #include<TlHelp32.h>
7   #include <iostream>
8   #include <tchar.h> // _tcscmp
9   #include <vector>
10
11
12  DWORD_PTR GetModuleBaseAddress(TCHAR* lpszModuleName, DWORD pID) {
13      DWORD_PTR dwModuleBaseAddress = 0;
14      HANDLE hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE, pID); // make snapshot of all modules within
15      MODULEENTRY32 ModuleEntry32 = { 0 };
16      ModuleEntry32.dwSize = sizeof(MODULEENTRY32);
17      //BYTE mBA[sizeof(DWORD)] = *(ModuleEntry32.modBaseAddr);
18      //memcpy(mBA, &dwModuleBaseAddress, sizeof(DWORD));
19      if (Module32First(hSnapshot, &ModuleEntry32)) //store first Module in ModuleEntry32
20      {
21          do {
22              if (_tcscmp(ModuleEntry32.szModule, lpszModuleName) == 0) // if Found Module matches Module we look
23              {
24                  dwModuleBaseAddress = (DWORD_PTR)ModuleEntry32.modBaseAddr;
```

Các bước để chương trình chèn code vào game:

B1: Dùng hàm getProcID để lấy process ID của tiến trình. Hàm này giúp xác định tiến trình của game có đang chạy hay không. Nếu việc tìm tiến trình mục tiêu ở hàm này thất bại, sẽ dẫn đến các hành động sau đó (mở tiến trình, cấp phát bộ nhớ, ...) không được thực hiện, trả về các thông báo không thể thực thi ở các bước và kết thúc chương

trình. Hàm getProcID được truyền vào giá trị tham số là biến process đã được gán sẵn giá trị ở hàm main, tạo một vòng lặp để so sánh tham số này với các tên tiến trình đang hoạt động, nếu tìm được tên tiến trình giống với tham số thì kết thúc vòng lặp, hàm sẽ trả về process ID của tiến trình đó.

```
int getProcId(char * procName) {
    DWORD pID = 0;
    PROCESSENTRY32 pe32;
    pe32.dwSize = sizeof(PROCESSENTRY32);
    // Take snapshots of everything
    HANDLE hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
    if (hSnapshot == INVALID_HANDLE_VALUE) {
        return 0;}
    //wchar_t * path;
    //path = (wchar_t *)GetWC(pe32.szExeFile);
    if (Process32First(hSnapshot, &pe32)){
    do {
        if (strcmp(pe32.szExeFile, procName) == 0) {
            CloseHandle(hSnapshot);
            //pID = pe32.th32ParentProcessID;
            pID = pe32.th32ParentProcessID;
            break;
        }
    } while (Process32Next(hSnapshot, &pe32));}
    CloseHandle(hSnapshot);
    return pID;
}
```

B2: Kiểm tra thư mục đang làm việc có chứa file AOEResourceHack.dll không. Nếu tìm được thì in ra đường dẫn đầy đủ của file để xác định file thực sự tồn tại.

```
char dll[] = "AOEResourceHack.dll";

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

B3: Mở tiến trình của game với các quyền truy cập cần thiết.

```
hProcess = OpenProcess(PROCESS_QUERY_INFORMATION |PROCESS_CREATE_THREAD | PROCESS_VM_OPERATION |PROCESS_VM_WRITE
```

B4: Cấp phát bộ nhớ trong tiến trình game cho đường dẫn DLL nếu có thể mở được tiến trình.

```
if (hProcess) {
    pszLibFileRemote = VirtualAllocEx(hProcess, NULL, strlen(dllPath) + 1, MEM_COMMIT | MEM_RESERVE, PAGE_READ
}
```

B5: Ghi đường dẫn file .dll vào bộ nhớ tiến trình.

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

B6: Tạo thread từ xa để gọi hàm LoadLibraryA nhằm tải dll vào không gian địa chỉ của tiến trình.

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

B7: Hàm foodHack, hàm này dùng để tăng số lượng "thịt". Copy code từ file Source.cpp và đổi tên từ hàm main thành foodHack. Sau đó compile thành file .o rồi chuyển thành file .dll. Chạy code của file Injector.cpp để tiêm file .dll "độc" vào trong tiến trình EMPIRESX.EXE.

Injector.cpp | launch.json | dllmain.cpp | Source.cpp

ameHacking > GameHacking > AOEResourceHack > dllmain.cpp > foodHack()

```
37  int foodHack() {
73
74      DWORD_PTR pointsAddress = baseAddress; //the Adress we need -> change now while going through offsets
75      for (int i = 0; i < pointsOffsets.size() - 1; i++) // -1 because we dont want the value at the last offset
76      {
77          ReadProcessMemory(processHandle, (LPVOID)(pointsAddress + pointsOffsets.at(i)), &pointsAddress, sizeof(pointsAddress
78          std::wcout << "debugginfo: Value at offset = " << std::hex << pointsAddress << std::endl;
79      }
80      pointsAddress += pointsOffsets.at(pointsOffsets.size() - 1); //Add Last offset -> done!!
81      float currentPoint = 0;
82
83      std::wcout << sizeof(currentPoint) << std::endl;
84      ReadProcessMemory(processHandle, (LPVOID)(pointsAddress), &currentPoint, sizeof(currentPoint), NULL);
85      std::wcout << "The last address is:" << pointsAddress << std::endl;
86      std::wcout << "Current value is:" << currentPoint << std::endl;
87      //"UI"
88      std::wcout << "Age of Empires Hack" << std::endl;
89
90      std::wcout << "How many points you want?" << std::endl;
91      float newPoints = currentPoint + 100;
92      std::wcin >> newPoints;
93      WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);
94
```

Trạng thái ban đầu khi chưa inject code:

Trạng thái sau khi inject code thành công, khi bấm F6, ta thấy số thịt ban đầu đã tăng lên thêm 100 miếng.