

BÁO CÁO THỰC HÀNH

Môn học: Cơ chế hoạt động của Mã độc

Lab 5: DLL injection

GVHD: Nguyễn Hữu Quyền

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.022.ATCL.1- Nhóm 3

STT	Họ và tên	MSSV	Email
1	Nguyễn Ngọc Trà My	21520353	21520353@gm.uit.edu.vn
2	Bùi Hoàng Trúc Anh	21521817	21521817@gm.uit.edu.vn
3	Lê Hoàng Oanh	21521253	21521253@gm.uit.edu.vn
4	Bùi Nguyễn Phúc	21522469	21522469@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:

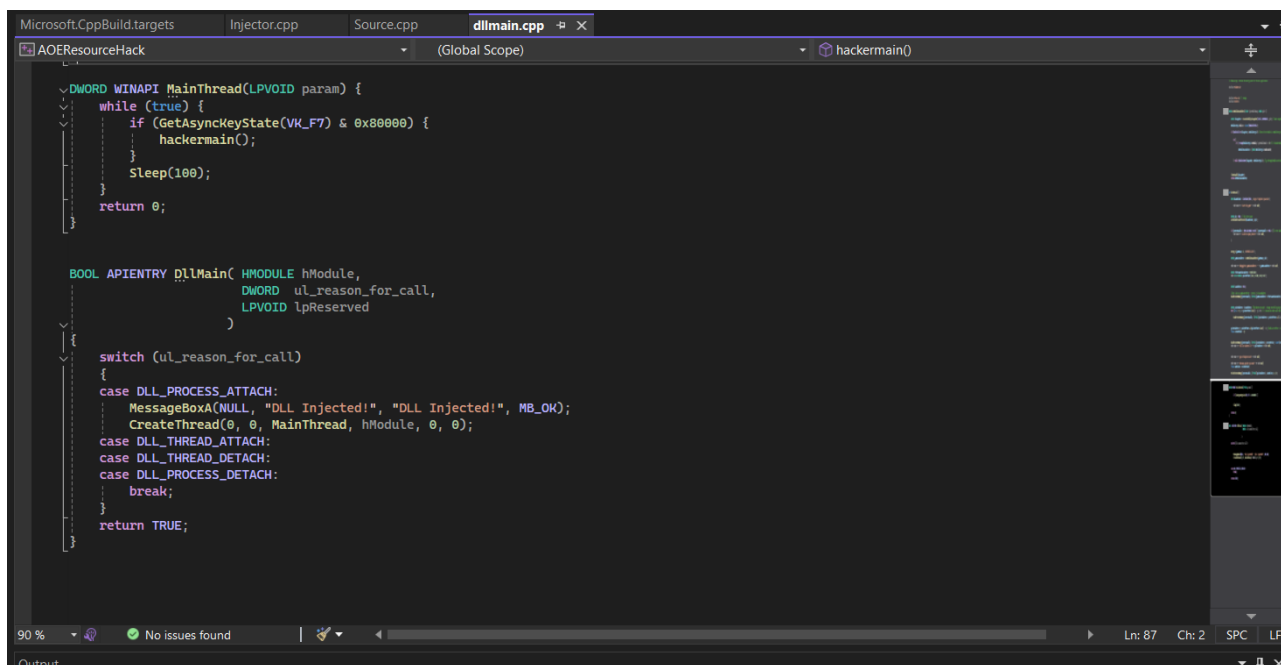
STT	Công việc	Kết quả tự đánh giá
1	Yêu cầu 1	100%

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

BÁO CÁO CHI TIẾT

Bài thực hành 1: Demo phần điểm thức ăn được nâng lên sau Inject.

Tiến hành inject một DLL để khi nhấn F6 thì game tự động nâng thức ăn lên. Ta có file dllmain.cpp như sau:








```
Microsoft.CppBuild.targets | Injector.cpp | Source.cpp | dllmain.cpp
AOEResourceHack | (Global Scope) | hackermain()

DWORD WINAPI MainThread(LPVOID param) {
    while (true) {
        if (GetAsyncKeyState(VK_F7) & 0x8000) {
            hackermain();
        }
        Sleep(100);
    }
    return 0;
}

BOOL APIENTRY DllMain( HMODULE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            MessageBoxA(NULL, "DLL Injected!", "DLL Injected!", MB_OK);
            CreateThread(0, 0, MainThread, hModule, 0, 0);
        case DLL_THREAD_ATTACH:
        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}
```

DLL sau khi được attached sẽ tiến hành tạo một thread để lắng nghe trạng thái phím F6. Đầu tiên ta copy code từ file Source.cpp sang dllmain.cpp. Sau đó đổi gameName sang EMPIRESX.EXE cho tất cả các file cpp

 EMPIRES	5/8/2024 1:27 PM	Help file	2,131 KB
 EMPIRESX.BAK	5/8/2024 1:27 PM	BAK File	1,479 KB
 EMPIRESX	5/8/2024 1:27 PM	Application	1,479 KB
 EULA	5/8/2024 1:27 PM	Rich Text Format	33 KB
 eula	5/8/2024 1:27 PM	Text Document	16 KB

Hàm để patch

```

int main() {
    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];
    wcsncpy_s(gameName, 13, L"EMPIRESX.EXE");

    DWORD gameBaseAddress = GetModuleBaseAddress(gameName, pID);

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

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

    DWORD baseAddress = NULL;

    //Get value at gamebase+offset -> store it in baseAddress
    ReadProcessMemory(processHandle, (LPCVOID)(gameBaseAddress + offsetGameToBaseAddress), &baseAddress, sizeof(baseAddress), NULL);
    std::cout << "debuginfo: baseAddress = " << std::hex << baseAddress << std::endl;
}

```

Để patch giá trị ta cần có Handle của process, địa chỉ nền của module và các giá trị offsets. Ta có thể dùng Cheat Engine để tìm các giá trị đó

Hàm xử lý và cập nhật giá trị

```

DWORD baseAddress = NULL;

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

DWORD pointsAddress = baseAddress; //the Address 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 << "debuginfo: 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);
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 = currentPoint;
newPoints = 1000;
WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);

DWORD WINAPI MainThread(LPVOID param) {
    while (true) {
        if (GetAsyncKeyState(VK_F7) & 0x80000) {
            hackermain();
        }
        Sleep(100);
    }
    return 0;
}

```

Ta đổi code ở chỗ newPoints để cộng thêm 100 với mỗi lần F6

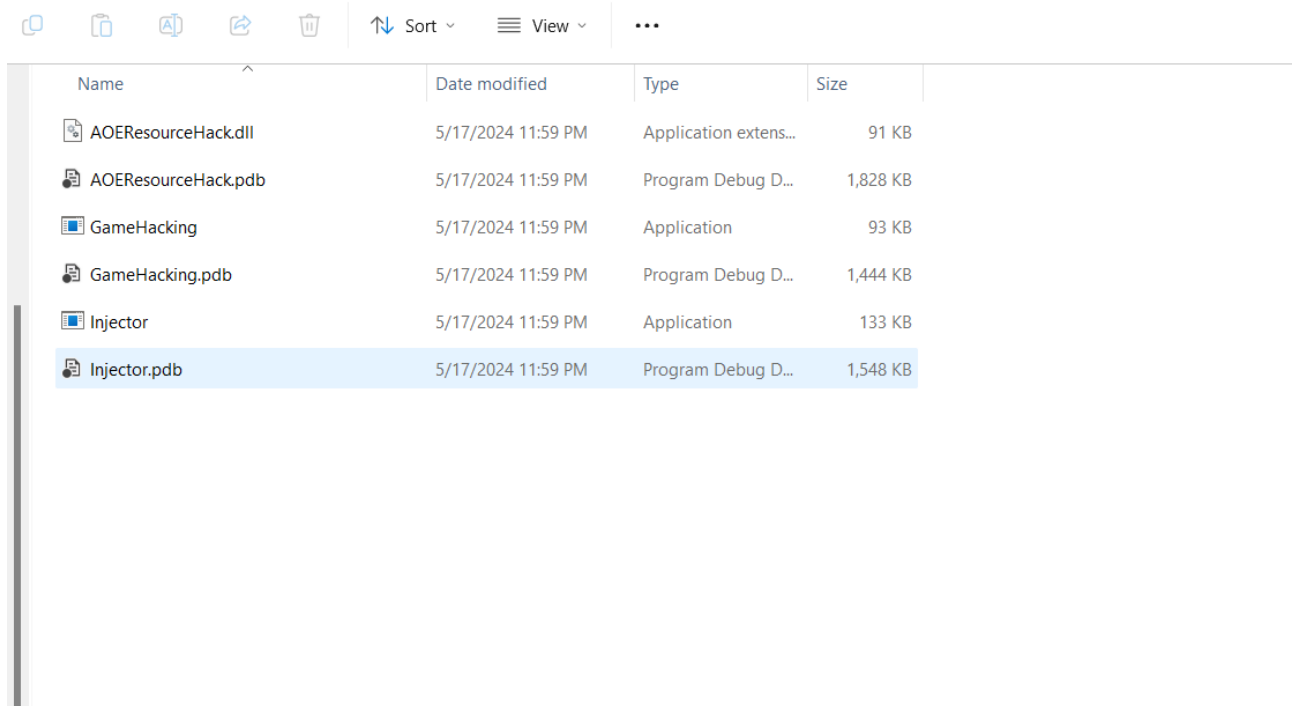
Ta quay lại hàm main() trong file dllmain.cpp và đổi tên thành hackermain() và thêm nó vào hàm Main Thread bên dưới

```

DWORD WINAPI MainThread(LPVOID param) {
    while (true) {
        if (GetAsyncKeyState(VK_F7) & 0x80000) {
            hackermain();
        }
        Sleep(100);
    }
    return 0;
}

```

Sau khi hoàn tất code, ta sẽ mở game và tiến hành build các file cpp. Sau khi build hoàn tất chạy GameHacking.exe để kiểm tra giá trị hiện tại, sau đó tắt đi và chọn Injector.exe



Name	Date modified	Type	Size
AOEResourceHack.dll	5/17/2024 11:59 PM	Application extens...	91 KB
AOEResourceHack.pdb	5/17/2024 11:59 PM	Program Debug D...	1,828 KB
GameHacking	5/17/2024 11:59 PM	Application	93 KB
GameHacking.pdb	5/17/2024 11:59 PM	Program Debug D...	1,444 KB
Injector	5/17/2024 11:59 PM	Application	133 KB
Injector.pdb	5/17/2024 11:59 PM	Program Debug D...	1,548 KB

Hoàn tất injector ta vào game và thử F6 (có trong video demo)