

BÁO CÁO BÀI TẬP

Môn học: Cơ chế hoạt động của mã độc

Kỳ báo cáo: Buổi 5

Tên chủ đề: DLL injection

GV: Nghi Hoàng Khoa

Ngày báo cáo: 22/5/2023

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT230.N21.ANTN

STT	Họ và tên	MSSV	Email
1	Nguyễn Bùi Kim Ngân	20520648	20520648@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá	Người đóng góp
1	Kịch bản 01/Câu hỏi 01	100%	
2	Kịch bản 02	100%	
3	Kịch bản 03	100%	
4	Kịch bản 04	90%	
5	Kịch bản 05	60%	

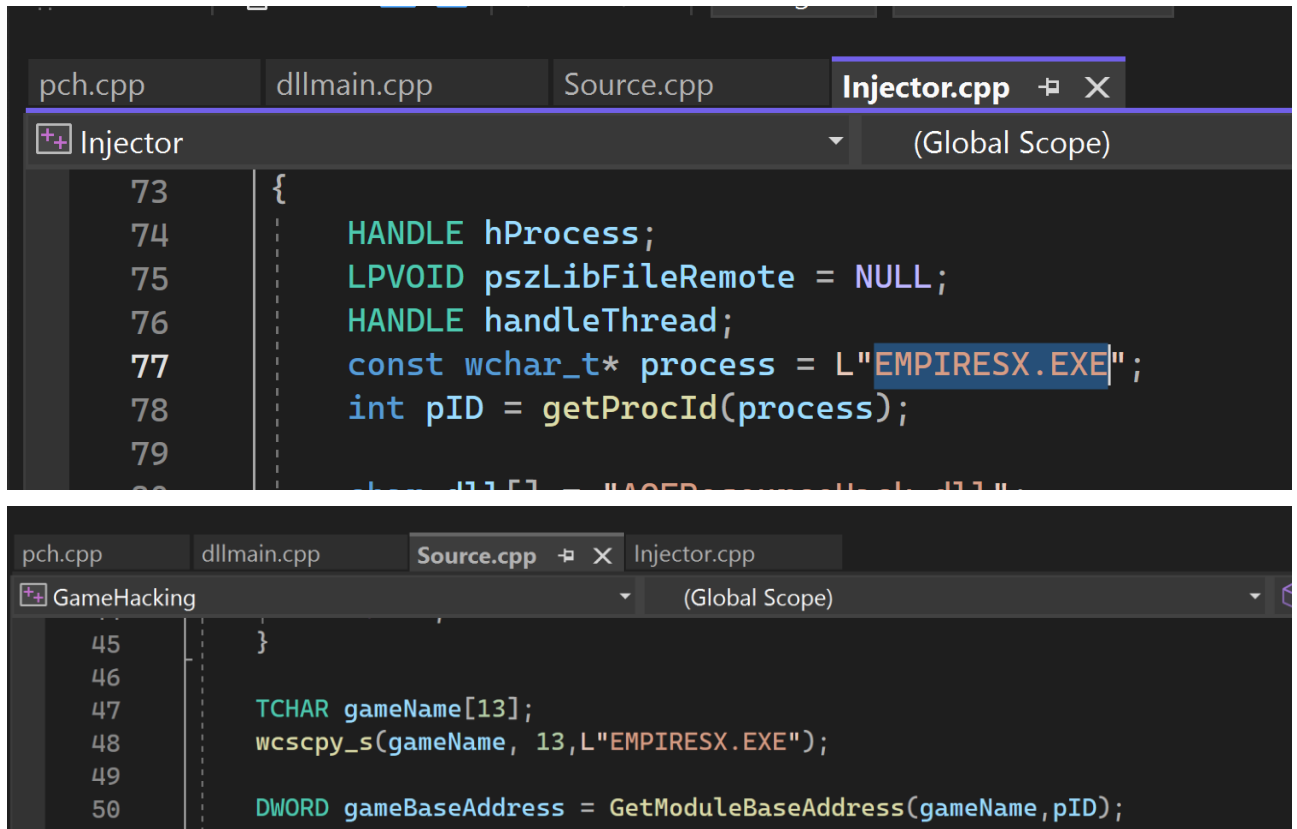
Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

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

BÁO CÁO CHI TIẾT

1. Kịch bản 01/Câu hỏi 01

Đầu tiên thì ta cần đổi tên process trong source thành EMPIRESX.EXE ở file Injector.cpp và Source.cpp



The first screenshot shows the `Injector.cpp` file in a C++ IDE. The code is in the Global Scope and includes the following lines:

```
73 {  
74     HANDLE hProcess;  
75     LPVOID pszLibFileRemote = NULL;  
76     HANDLE handleThread;  
77     const wchar_t* process = L"EMPIRESX.EXE";  
78     int pID = getProcId(process);  
79
```

The second screenshot shows the `Source.cpp` file in the same IDE. The code is also in the Global Scope and includes the following lines:

```
45 }  
46  
47 TCHAR gameName[13];  
48 wcscpy_s(gameName, 13, L"EMPIRESX.EXE");  
49  
50 DWORD gameBaseAddress = GetModuleBaseAddress(gameName, pID);
```

Về Injector.cpp, thực hiện một số công việc sau

- Lấy tên của process mình sẽ injector vào và lấy path của dll

```

Injector
(Global Scope)
main()

70
71
72 int main()
73 {
74     HANDLE hProcess;
75     LPVOID pszLibFileRemote = NULL;
76     HANDLE handleThread;
77     const wchar_t* process = L"EMPIRESX.EXE";
78     int pID = getProcId(process);
79
80     char dll[] = "AOEResourceHack.dll";
81     if (!exist(dll)) {
82         std::cout << "debuginfo: Invalid DLL path!" << std::endl;
83     }

```

- Sau đó thực hiện ghi vào bộ nhớ và tạo thread

```

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,
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;

```

Về Source.cpp, cũng lấy tên process, lấy các thông tin về ô nhớ lưu giá trị của food sau khi có cộng các offsets.

```

dllmain.cpp
Source.cpp
Injector.cpp
(Global Scope)
main()

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;

```

Và thực hiện sửa đổi giá trị theo input mình nhập.

```

dllmain.cpp Source.cpp Injector.cpp
king (Global Scope) main()
}
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 = 0;
std::cin >> newPoints;
WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);
}

```

Chúng ta đã có file source.cpp có khả năng thay đổi số lượng thức ăn trong game theo số ta nhập nên em tận dụng lại code này để thực hiện chức năng khi nhấn phím F6 thì chương trình sẽ tăng thức ăn.

Copy toàn bộ source.cpp vào dllmain.cpp và sửa một số điểm sau.

Đầu tiên, em thay tên hàm int main thành int increaseFood và gọi tới hàm này trong MainThread(), ở đây em cũng tắt MessageBox nhấn F6 để tránh out khỏi màn hình game.

```

pch.cpp dllmain.cpp Source.cpp Injector.cpp
*+ AOEResourceHack (Global Scope) Ma
25
26
27 }
28 CloseHandle(hSnapshot);
29 return dwModuleBaseAddress;
30 }
31
32
33 int increaseFood() {
34
35     HWND hGameWindow = FindWindow(NULL, L"Age of Empires Expansion");
36     if (hGameWindow == NULL) {
37         std::cout << "Start the game!" << std::endl;
38         return 0;
39     }
40     DWORD pID = NULL; // ID of our Game

```

```
87 [
88     DWORD WINAPI MainThread(LPVOID param) {
89         while (true) {
90             if (GetAsyncKeyState(VK_F6) & 0x80000) {
91                 //MessageBoxA(NULL, "F6 Pressed!", "F6 Pressed!", MB_OK);
92                 increaseFood();
93             }
94             Sleep(100);
95         }
96         return 0;
97     }
98 }
```

Tiếp theo trong hàm `increaseFood`, em chỉnh đoạn code như sau để thực hiện chức năng tăng thức ăn lên 100 khi nhấn F6

```
pch.cpp  dllmain.cpp  Source.cpp  Injector.cpp
+ AOEResourceHack (Global Scope) MainThread(LPVOID)
79 // "UI"
80 std::cout << "Age of Empires Hack" << std::endl;
81
82 std::cout << "How many points you want?" << std::endl;
83 float newPoints = currentPoint + 100;
84 //std::cin >> newPoints;
85 WriteProcessMemory(processHandle, (LPVOID)(pointsAddress), &newPoints, 4, 0);
86
87 }
```

Và cuối cùng chọn build solution

Link video <https://youtu.be/1ebE47E8zxM>

Khi quay video mình họa em có gặp trục trặc, màn hình game khi quay lại bị thu nhỏ vào góc màn hình mặc dù ở ngoài máy thật đang ở fullscreen, tuy nhiên em chưa fix được lỗi này nên em chụp lại màn hình làm mình chứng ở dưới.











Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).
Ví dụ: [NT101.K11.ANTT]-Session1_Group3.
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT