|  |
| --- |
| **BỘ THÔNG TIN VÀ TRUYỀN THÔNG**  **HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  ------o0o------    **TÌM HIỂU BÀI THỰC HÀNH PTIT - OLLYDBG**  Giảng viên: Đinh Trường Duy |

## **1. Nội dung và hướng dẫn thực hiện bài thực hành**

### 1.1. Mục đích

Bài thực hành này giới thiệu công cụ OllyDBG, một công cụ debug mã hợp ngữ disassembly. Bài thực hành giúp sinh viên hiểu về các khái niệm và kỹ thuật trong lĩnh vực dịch ngược, phân tích mã nguồn, cũng như làm quen với công cụ OllyDBG.

### 1.2. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về cách thức quản lý tệp tin, bộ nhớ, quy trình của hệ điều hành Windown, ngôn ngữ lập trình Assembly, biết cách sử dụng công cụ debugging như OllyDBG để theo dõi, phân tích một chương trình máy tính.

### 1.3. Chuẩn bị môi trường để có thể làm bài thực hành

* Sinh viên tải và cài đặt một ảo Windown 10 được cung cấp sẵn để có thể thực hiện bài thực hành với máy Labtainer  Link tải máy windows7 cho Vmware:

*https://ptiteduvnmy.sharepoint.com/:f:/g/personal/trungnd\_b20at194\_stu\_ptit\_edu\_vn/El7cq wVoeHRHl\_28b4Zi9tQBeayJmdPqeHDpsERyGnYt9Q?e=u00Xut*

* Thiết lập chức năng Remote Desktop, đầu tiên mở:

## Settings -> System -> Remote Desktop -> Enable Remote Desktop -> On

 Sau khi thiết lập chức năng Remote Desktop trên máy ảo Windown 10, sinh viên tiến hành kiểm tra kết nối giữa hai máy thông qua lệnh:

Trên máy Terminal của máy Labtainer gõ lệnh:

*ping 192.168.152.125(****địa chỉ ip của máy Windown 10****)*

### 1.4. Nội dung thực hành

 Khởi động bài lab:o Vào terminal, gõ:

*labtainer ptit-ollydbg*

## (Chú ý: sinh viên sử dụng mã sinh viên của mình để nhập thông tin email người thực hiện bài lab khi có yêu cầu, để sử dụng khi chấm điểm)

Sau khi khởi động xong một terminal ảo sẽ xuất hiện, đại diện cho máy ***ptit-ollydbg***

* Trên terminal ***ptit-ollydbg*** sử dụng câu lệnh *"remmina"* để khởi chạy ứng dụng điều khiển máy tính từ xa Remmina.
* Cửa sổ ứng dụng Remmina hiện ra, sinh viên tiến hành tạo một RDP(Remote Desktop Protocol) với các tham số như : 
  + Name: ThucHanhATTT
  + Group: là nhóm của kết nối
  + Protocol: là giao thức kết nối
  + Sever: địa chỉ IP máy Windown
  + User name: Lab
  + User password: 1234@qwer

 Nhấn ***Save&Connect***để lưu và kết nối đến máy tính Windown 10

* Sinh viên làm các nhiệm vụ dưới đây để hoàn thành bài lab: o
  + ***Nhiệm vụ 1: Cài đặt OllyDBG và open file EasyCrackMe.exe***  OllyDBG là một công cụ gỡ lỗi (debugger) phổ biến được sử dụng rộng rãi trong cộng đồng phát triển phần mềm và phân tích malware. Với OllyDBG, người dùng có thể phân tích và thao tác với các chương trình máy tính chạy trên hệ điều hành Windows một cách chi tiết và linh hoạt. Công cụ này cung cấp một loạt các tính năng mạnh mẽ như xem và chỉnh sửa mã máy, theo dõi giá trị của các biến, gỡ lỗi đa luồng, tìm kiếm và thay đổi mã máy, phân tích mã, và hỗ trợ plugin mở rộng. Công cụ bao gồm các cửa sổ con như:
* **Disassembler window:** Hiển thị các đoạn mã của chương trình dưới dạng mã assembly và các chú thích kèm theo tại các dòng mã đó.
* **Register window:** Hiển thị danh sách các thanh ghi và giá trị của chúng trong quá trình thực thi chương trình.
* **Tip window:** Cung cấp các thông tin bổ sung và gợi ý cho một dòng mã cụ thể, hữu ích trong quá trình debug.
* **Dump window:** Cho phép người dùng xem và chỉnh sửa các giá trị trong bộ nhớ của chương trình đang được debug.
* **Stack window:** Hiển thị thông tin về stack của chương trình, bao gồm các giá trị và địa chỉ trên stack.

## Bảng 1.1: Một số lệnh trong OllyDBG

|  |  |
| --- | --- |
| Câu lệnh | Mô tả |
| Run (F9) | Bắt đầu thực thi chương trình |
| Pause (F12) | Tạm dừng thực thi chương trình |
| Step Into (F7) | Thực hiện lệnh hiện tại và di chuyển đến lệnh tiếp theo, bao gồm cả trong các hàm con |
| Step Over (F8) | Thực hiện lệnh hiện tại và di chuyển đến lệnh tiếp theo, nhưng không bước vào trong các hàm con. |
| Set Breakpoint  (F2) | Đặt một điểm dừng tại địa chỉ hiện tại. |
| Toggle Breakpoint  (Space) | Chuyển đổi trạng thái của điểm dừng tại địa chỉ hiện tại. |
| Run to Cursor  (Ctrl+F9) | Chạy chương trình đến địa chỉ đang được chỉ định bởi con trỏ. |
| Search for (Ctrl+B) | Tìm kiếm một chuỗi byte hoặc một hằng số trong mã máy. |
| CPU Window  (Alt+C) | Hiển thị cửa sổ CPU, nơi bạn có thể xem và chỉnh sửa các thanh ghi CPU và trạng thái của chúng. |
| Disassembler Window (Alt+D) | Hiển thị cửa sổ Disassembler, nơi bạn có thể xem mã máy của chương trình. |
| Memory Window  (Alt+M) | Hiển thị cửa sổ Memory, nơi bạn có thể xem và chỉnh sửa nội dung của bộ nhớ. |
| Registers Window  (Alt+R) | Hiển thị cửa sổ Registers, nơi bạn có thể xem và chỉnh sửa các thanh ghi CPU. |
| Stack Window  (Alt+S) | Hiển thị cửa sổ Stack, nơi bạn có thể xem nội dung của stack. |

* Sau khi hoàn thành bước truy cập từ xa được vào máy Windown, sinh viên khởi chạy công cụ OllyDBG v1.10.
* Sử dụng OllyDBG mở file chương trình Easy\_CrackMe.exe bằng cách nhấn: *F3 -> Downloads -> Easy\_CrackMe.exe*
* Sinh viên quan sát các cửa sổ của ứng dụng OllyDBG khi mở file chương trình

o ***Nhiệm vụ 2: Sử dụng OllyDBG để unpack chương trình***

* Sau khi mở file chương trình xong, nhấn F9 để chạy chương trình, khi chạy chương trình sẽ hiện ra một cửa sổ, nhập vào mật khẩu cho chương trình. Ví dụ: B20DCAT001

 Sau khi nhập mật khẩu, thông báo của chương trình sẽ được đưa ra, đến bước này sinh viên sẽ cần phải tìm các thông tin của đoạn chuỗi văn bản thông báo hiện ra màn hình như địa chỉ bộ nhớ, đoạn mã assembly, text.

##  Sinh viên bấm chuột trái tại cửa sổ **Dissambler window** **-> Search for -> All referenced text string**

* Tại cửa sổ hiện ra, trỏ đến đoạn text thông báo ra màn hình trước đó rồi thực hiện: *Chuột trái -> Copy to Clipboard -> Whole line*  để sao chép thông tin của đoạn chương trình này
* Ghi lại đoạn nội dung trên vào dưới phần “*Nhiem vu 2:”* trong file

ATTT.txt

o ***Nhiệm vụ 3: Thực hiện Debug chương trình***

* Click double vào chuỗi “Incorrect Password” để đi đến đoạn code trong chương trình sử dụng chuỗi này, quan sát thấy có hai trường hợp xử lý khi người dung nhập thông tin vào message box đó là *“Congratulation”* và *“Incorrect Password”.*
* Di chuyển đến đoạn code đầu tiên của câu lệnh in ra thông báo

### “Incorrect Password” có địa chỉ 0x00401135 để đi đến lệnh JNZ đầu tiên: Chuột trái -> Go to -> JNZ form

 Quan sát các câu lệnh của chương trình ta thấy có câu lệnh:

CALL DWORD PTR DS:[<&USER32.GetDlgItemTextA>]

Với dòng comment được bôi đỏ có nội dung *GetDlgItemTextA*

 Có thể đây chính là câu lệnh thực hiện nhiệm vụ đọc password khi ta nhập vào.

Đặt một breakpoint tại địa chỉ 0x004010AA bằng cách nhấn F2, rồi sau đó chạy lại chương trình, nhập mật khẩu, ấn nút “??” ta sẽ thấy chương trình dừng tại breakpoint vừa đặt.Nhấn F8 để chạy được qua hàm *GetDlgItemTextA,* quan sát Stack(Stack SS:[0019F7E1]=32 ('2')) và câu lệnh hiện tại thấy có sự so sánh giá trị giữa ESP+5 và 0x61, trong đó địa chỉ ESP+5 là kí tự thứ 2 trong password nhập vào.

* Tiếp tục nhấn F8 và điều chỉnh cờ **Z** trong cửa sổ Registers(FPU) bằng việc double click vào giá trị **0** ở cờ **Z**, giá trị được sét lại giá trị **1** để di chuyển đến câu lệnh

LEA ECX,DWORD PTR SS:[ESP+A]

Quan sát Stack thấy có sự so sánh giá trị giữa ESP+A với chuỗi ASCII của chương trình, trong đó ESP+A là các kí tự thứ 3 và 4 của chuỗi password nhập vào.

* Qua quá trình Debug, ta đã tìm được các kí tự số 2,3,4 của mật khẩu chương trình.
* Lưu lại đoạn mã code so sánh, trên dưới phần “*Nhiem vu 3:”* vào file

ATTT.txt

o ***Nhiệm vụ 4: Tìm chuỗi nội dung của Message Box***

* Tiếp tục nhấn F8 và điều chỉnh cờ **Z** trong cửa sổ Registers(FPU) bằng việc double click vào giá trị **0** ở cờ **Z**, giá trị được sét lại giá trị **1** để di chuyển đến câu lệnh

LEA EAX,DWORD PTR SS:[ESP+10]

Quan sát Stack thấy có sự so sánh giữa ESP+10 với chuỗi ASCII của chương trình, trong đó địa chỉ ESP+10 là các kí tự thứ 5 trở đi của chuỗi password nhập vào.

* Qua quá trình trên , ta đã tìm được các kí tự thứ 5 trở đi của mật khẩu chương trình.

Lưu lại đoạn mã code so sánh, các kí tự ASII được so sánh trong code chương trình dưới phần “*Nhiem vu 4:”* vào file ATTT.txt

o ***Nhiệm vụ 5: Tìm ra mật khẩu của File chương trình***

* Tiếp tục nhấn F8 và điều chỉnh cờ **Z** trong cửa sổ Registers(FPU) bằng việc double click vào giá trị **0** ở cờ **Z**, giá trị được sét lại giá trị **1** để di chuyển đến câu lệnh

CMP BYTE PTR SS:[ESP+4],45

Quan sát Stack thấy có sự so sánh giữa ESP+5 và 0x45, trong đó địa chỉ ESP+4 là các kí tự đầu tiên của chuỗi password nhập vào.

* Qua qua đó đã tìm được kí tự đầu tiên của mật khẩu chương trình, kết hợp các bước trên có thể suy ra được mật khẩu.
* Chạy lại chương trình, bỏ breakpoint và nhập mật khẩu vừa lấy được.Nếu thông báo hiện *“Congratulation”* thì mật khẩu của chương trình đúng, nếu ngược lại mật khẩu của chương trình là sai và sinh viên cần thực hiện lại các bước trên.
* Ghi lại mật khẩu của chương trình và lưu file ATTT.txt.

* Kết thúc bài lab:
  + Trên ***terminal*** đầu tiên sử dụng câu lệnh sau để kết thúc bài lab: *stoplab ptit-ollydbg*
  + Khi bài lab kết thúc, một tệp zip lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới stoplab.
* Khởi động lại bài lab:
  + Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, dùng câu lệnh:

### labtainer -r ptit-ollydbg