

BÁO CÁO THỰC HÀNH LẬP TRÌNH HỆ THỐNG LAB3

Nhóm 12:

22520825 Nguyễn Đức Luân

22520661 Vũ Ngọc Quốc Khánh

22521110 Đoàn Hoàng Phúc

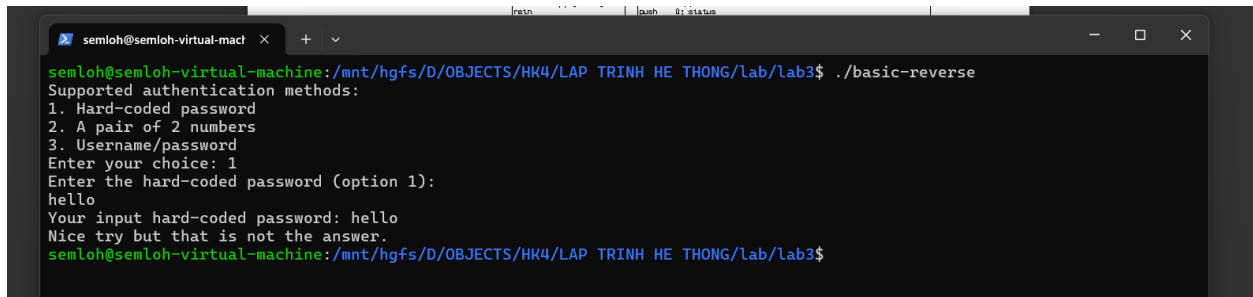
Ta xem flowchart của chương trình bằng công cụ IDA pro 32 bit:



Chương trình sử dụng 3 hàm chính (3 options) tương ứng với 3 challenges cần solve:

- hardCode()
- otherhardCode()
- userpass()

Ví dụ



```
semloh@semloh-virtual-machine: /mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/lab/lab3$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. A pair of 2 numbers
3. Username/password
Enter your choice: 1
Enter the hard-coded password (option 1):
hello
Your input hard-coded password: hello
Nice try but that is not the answer.
semloh@semloh-virtual-machine: /mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/lab/lab3$
```

Yêu cầu 1. Phân tích và tìm *passphrase cố định (option 1)* của **basic-reverse** với phương pháp chứng thực 1. Báo cáo phương pháp phân tích, input tìm được và hình ảnh minh chứng chạy file.

Xem xét hàm hardCode dưới mã assembly:

```

; Attributes: bp-based frame

; int hardCode()
public hardCode
hardCode proc near

s1= byte ptr -3F0h

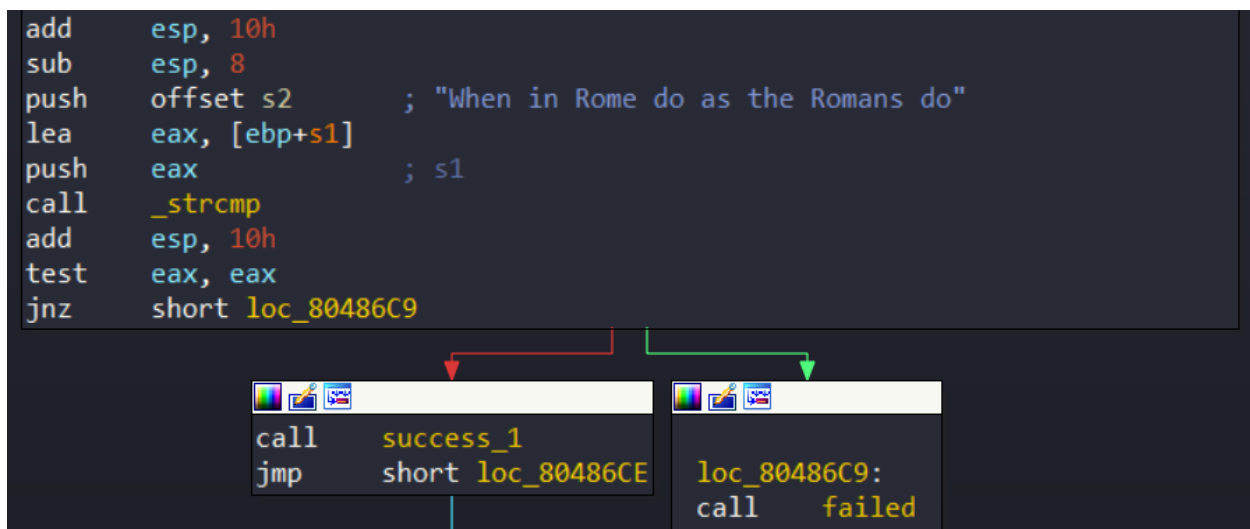
; __unwind {
push    ebp
mov     ebp, esp
sub     esp, 3F8h
call    _getchar
sub     esp, 0Ch
push    offset aEnterTheHardCo ; "Enter the hard-coded password (option 1"...
call    _puts
add     esp, 10h
sub     esp, 8
lea     eax, [ebp+s1]
push    eax
push    offset asc_804916A ; "%[^\\n]"
call    ___isoc99_scanf
add     esp, 10h
sub     esp, 8
lea     eax, [ebp+s1]
push    eax
push    offset format ; "Your input hard-coded password: %s\\n"
call    _printf
add     esp, 10h
sub     esp, 8
push    offset s2 ; "When in Rome do as the Romans do"
lea     eax, [ebp+s1]
push    eax ; s1
call    _strcmp
add     esp, 10h
test    eax, eax
jnz     short loc_80486C9

```

Hàm thực hiện tạo một biến s1 và được cấp phát. Ban đầu chương trình gọi hàm getchar() có lẽ để lấy ký tự xuống dòng “\n” trước đó. Sau đó dùng hàm thư viện C và chuẩn bị một số tham số cho hàm

_puts và in dòng chữ “Enter the hard-coded password (option 1):” được lưu trữ trong biến aEnterTheHardCo

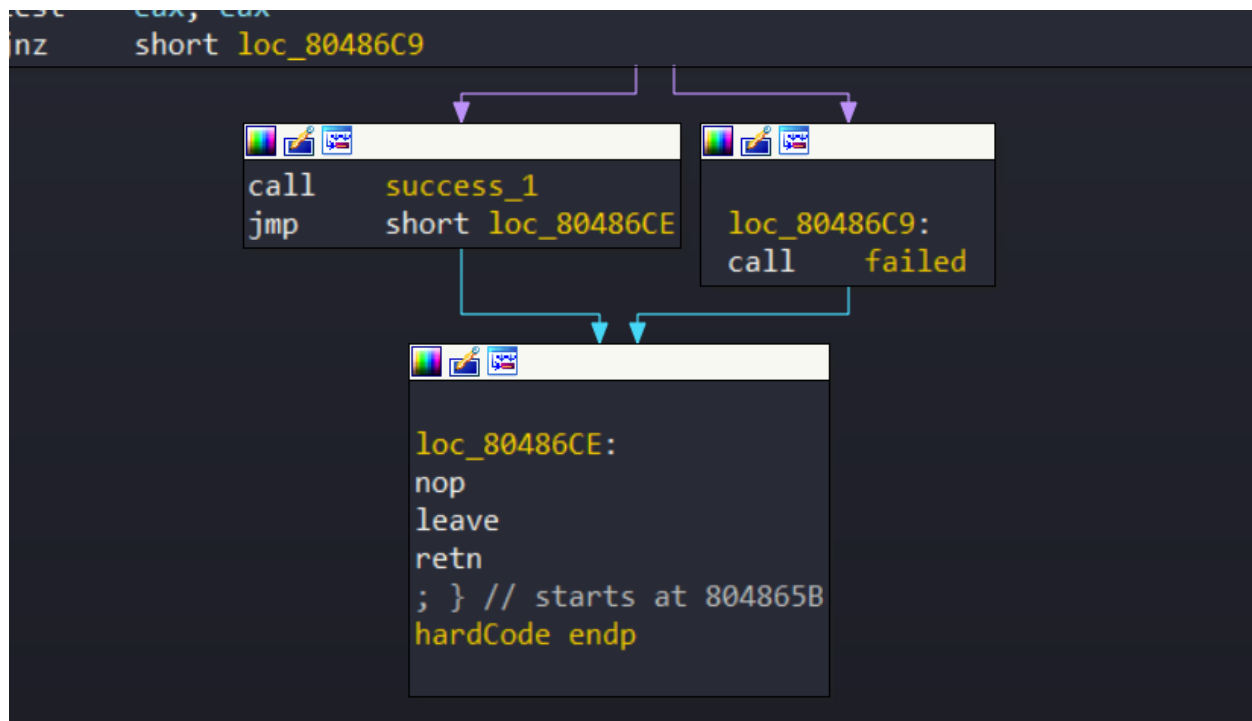
Tiếp tục, dùng __isoc99_scanf để cho người dùng nhập input. Dùng hàm _printf để in ra thông báo “Your input hard-coded password: %s” được lưu trong biến format và thay thế chuỗi “%s” với password chúng ta nhập tại s1. Cuối cùng đến đoạn chương trình chính:



Để ý kỹ 2 dòng push các tham số vào s1 (password ta nhập), s2 (chuỗi cần so sánh). Chúng ta có thể thấy chuỗi được lưu tại biến s2

When in Rome do as the Romans do

Sau đó kết quả trả về ở eax, nếu `eax = 0` thì 2 chuỗi bằng nhau thì thực hiện tiếp đến hàm `success_1` và in ra thông điệp báo thành công, còn nếu không bằng thì nhảy đến `short loc_80486C9` và thông báo không phê duyệt



Vậy password là: When in Rome do as the Romans do

Kết quả thử nghiệm:

```

semloh@semloh-virtual-machine:/mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/Lab/lab3$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. A pair of 2 numbers
3. Username/password
Enter your choice: 1
Enter the hard-coded password (option 1):
When in Rome do as the Romans do
Your input hard-coded password: When in Rome do as the Romans do
Congrats! You found the hard-coded secret, good job :).
semloh@semloh-virtual-machine:/mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/Lab/lab3$

```

Yêu cầu 2. Phân tích và tìm **cặp số nguyên (option 2)** của **basic-reverse** với phương pháp chứng thực 2. Báo cáo phương pháp phân tích, input tìm được và hình ảnh minh chứng chạy file.

Xem xét hàm otherhardCode() trên mã giả:

```

1 int otherhardCode()
2 {
3     int v0; // edx
4     int v2; // [esp+4h] [ebp-14h] BYREF
5     int v3[4]; // [esp+8h] [ebp-10h] BYREF
6
7     getchar();
8     puts("Enter your 2 numbers (separated by space) (option 2):");
9     __isoc99_scanf("%d %d", v3, &v2);
10    printf("Your input: %d %d\n", v3[0], v2);
11    v3[1] = 5;
12    if ( v3[0] == 5 && (v0 = funny_func(funny_seq[5], 5), v0 == v2) )
13        return success_2();
14    else
15        return failed();
16 }

```

Từ mã giả C cho thấy để có thể vào hàm success_2() ta phải thỏa mãn điều kiện của lệnh if trước đó:

```

if ( v3[0] == 5 && (v0 = funny_func(funny_seq[5], 5), v0 == v2) )

```

Như ta đã biết thì biến v3[0] là số thứ nhất và v2 là số thứ hai nhập vào chương trình, điều kiện của lệnh if là số thứ nhất phải là 5 và số thứ 2 phải thỏa mã hàm điều kiện là phải bằng biến v0, v0 được tính bằng hàm sau:

```

(v0 = funny_func(funny_seq[5], 5), v0 == v2) )

```

Xem xét hàm funny_func(): với hai tham số là funny_seq[5], và số 5

```

1 int __cdecl funny_func(int a1, int a2)
2 {
3     return (a1 + a2 - 1) * (a1 + a2);
4 }

```

Từ đây ta thấy hàm này trả về kết quả của biểu thức trên. Vấn đề là xác định biến funny_seq[5] có giá trị là bao nhiêu bằng cách dựa vào địa chỉ gốc của mảng funny_seq[] với offset. Mà đây là mảng int nên địa chỉ của ô nhớ $k = \text{địa chỉ mảng} + \text{offset} * 4$

```

.rodata:08048A60 funny_seq      dd 1
.rodata:08048A64                db 3
.rodata:08048A65                db 0
.rodata:08048A66                db 0
.rodata:08048A67                db 0
.rodata:08048A68                db 5
.rodata:08048A69                db 0
.rodata:08048A6A                db 0
.rodata:08048A6B                db 0
.rodata:08048A6C                db 7
.rodata:08048A6D                db 0
.rodata:08048A6E                db 0
.rodata:08048A6F                db 0
.rodata:08048A70                db 9
.rodata:08048A71                db 0
.rodata:08048A72                db 0
.rodata:08048A73                db 0
.rodata:08048A74                db 2
.rodata:08048A75                db 0

```

Từ hình trên ta xác định được địa chỉ của funny_seq[5] là 08048A74 và có giá trị bằng 2. Lúc này ta đã có được 2 tham số của hàm funny_func(): với hai tham số là funny_seq[5] = 2 và 5. Qua đó xác định được giá trị của biến v0 = 42

Vậy đáp án hai số là: 5 42

Thực nghiệm kết quả trên chương trình:

```

Supported authentication methods:
1. Hard-coded password
2. A pair of 2 numbers
3. Username/password
Enter your choice: 2
Enter your 2 numbers (separated by space) (option 2):
5 42
Your input: 5 42
Congrats! You found a secret pair of numbers :).
semloh@semloh-virtual-machine:/mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/lab/lab3$

```


Yêu cầu 3. Phân tích, tìm **username/password** phù hợp của **basic-reverse** với phương pháp chứng thực 3. Báo cáo phương pháp và input tìm được.

Lưu ý bắt buộc: **username** được tạo từ MSSV của các thành viên trong nhóm.

- Nhóm **3 sinh viên**, sắp xếp MSSV theo thứ tự tăng dần rồi lấy 3 số cuối nối nhau. Ví dụ 22520013 < 22520123 < 22521021 sẽ có username là **013123021**.
- Nhóm **2 sinh viên**, sắp theo thứ tự tăng dần và lấy 4 số cuối nối nhau bằng dấu "-". Ví dụ 22520013 < 22520123 sẽ có username là 0013-0123.
- Nhóm có **1 sinh viên** có MSSV là 2152xxxx thì username là **2152-xxxx**.

Ta có 3 MSSV theo thứ tự tăng dần: 22520661 22502825 22521110

Từ đó xác định được username là: 661825110

Xem xét hàm userpass:

```
int userpass()
{
    size_t len_user; // ebx
    size_t v2; // eax
    size_t v3; // edx
    char v4[9]; // [esp+Ah] [ebp-2Eh]
    char pass[10]; // [esp+13h] [ebp-25h] BYREF
    char user[10]; // [esp+1Dh] [ebp-1Bh] BYREF
    char token[5]; // [esp+27h] [ebp-11h] BYREF
    int i; // [esp+2Ch] [ebp-Ch]
```

```

qmemcpy(token, "7`ZK_", sizeof(token));
getchar();
puts("Enter your username:");
__isoc99_scanf("%[^\\n]", user);
getchar();
puts("Enter your password:");
__isoc99_scanf("%[^\\n]", pass);
printf("Your input username: %s and password: %s\\n", user, pass);
if ( strlen(user) != 9 )
    return failed();
len_user = strlen(user);
if ( len_user != strlen(pass) )
    return failed();
for ( i = 0; i <= 8; ++i )
{
    if ( i > 1 )
    {
        if ( i > 3 )
            v4[i] = token[i - 4];
        else
            v4[i] = user[i + 5];
    }
    else
    {
        v4[i] = user[i + 2];
    }
}
for ( i = 0; ; ++i )
{
    v2 = strlen(user);
    if ( v2 <= i || (user[i] + v4[i]) / 2 != pass[i] )
        break;
}
v3 = strlen(user);
if ( v3 == i )
    return success_3();
else
    return failed();
}

```

Vào hàm userpass(): Đoạn đầu là khai báo các biến. Tiếp theo cấp vùng nhớ và giá trị “7`ZK_” cho biến token và thực hiện tương tự công đoạn lấy dữ liệu đầu vào và hiển thị ra với 2 trường username và password: getchar() -> __isoc99_scanf -> getchar() -> puts() -> __isoc99_scanf -> printf. Xong rồi check thử xem độ dài password

đầu vào của username : len(username)== 9? và xem len(password) == 9?

```
qmemcpy(token, "7`ZK_", sizeof(token));
getchar();
puts("Enter your username:");
__isoc99_scanf("%[^\\n]", user);
getchar();
puts("Enter your password:");
__isoc99_scanf("%[^\\n]", pass);
printf("Your input username: %s and password: %s\\n", user, pass);
if ( strlen(user) != 9 )
    return failed();
len_user = strlen(user);
if ( len_user != strlen(pass) )
    return failed();
```

Sau đó chương trình tạo một biến v4 mới được hình thành như sau:

```
for ( i = 0; i <= 8; ++i )
{
    if ( i > 1 )
    {
        if ( i > 3 )
            v4[i] = token[i - 4];
        else
            v4[i] = user[i + 5];
    }
    else
    {
        v4[i] = user[i + 2];
    }
}
```

Chạy riêng đoạn code này với 2 biến user = "661825110

" và token ="7`ZK_" ta tìm được giá trị của v4 ="18107`ZK_"

Tiếp đến là phần hình thành mật khẩu để đối chiếu:

```
for ( i = 0; ; ++i )
{
    v2 = strlen(user);
    if ( v2 <= i || (user[i] + v4[i]) / 2 != pass[i] )
        break;
}
v3 = strlen(user);
if ( v3 == i )
    return success_3();
else
    return failed();
```

Từ code trên ta thấy các kí tự của password bằng trung bình cộng các giá trị trong chuỗi của hai biến user và token:

Ta có chương trình giải mã:

```

#include <iostream>
using namespace std;
int main() {
    string user="661825110";
    string test ="7`ZK_";
    char v4[9];
    for(int i=0; i<9;i++){
        if ( i > 1 )
        {
            if ( i > 3 )
                v4[i] = test[i - 4];
            else
                v4[i] = user[i + 5];
        }
        else
        {
            v4[i] = user[i + 2];
        }
    }
    for(int i=0;i<9;i++){
        cout<<v4[i];
    }
    cout<<"\n";

    for(int i=0;i<9;i++){
        char a = (user[i] + v4[i]) / 2;
        cout<<a;
    }
}

```

Password ta tìm được là: 37144JE>G

Vậy đáp án của bài này là

user: 661825110

password: 37144JE>G

Kết quả khi chạy chương trình:

```
semloh@semloh-virtual-machine:/mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/lab/lab3$ ./basic-reverse
Supported authentication methods:
1. Hard-coded password
2. A pair of 2 numbers
3. Username/password
Enter your choice: 3
Enter your username:
661825110
Enter your password:
37144JE>G
Your input username: 661825110 and password: 37144JE>G
Congrats! You found your own username/password pair :).
semloh@semloh-virtual-machine:/mnt/hgfs/D/OBJECTS/HK4/LAP TRINH HE THONG/lab/lab3$
```