


LẬP TRÌNH HỆ THỐNG
BÁO CÁO LAB 4
KỸ THUẬT DỊCH NGƯỢC CƠ BẢN

Họ và tên	MSSV	Lớp
Hồ Diệp Huy	22520541	NT209.O21.ANTT.2 Nhóm 14
Mai Nguyễn Nam Phương	22521164	
Đinh Quốc Huy	22520536	

Team 14				
6		Đinh Quốc Huy		
		22520536		
		10	15:40:41 06/05/2024	
		Hồ Diệp Huy		
		22520541		
Mai Nguyễn Nam Phương				
22521164				

Hình 1: Kết quả trả lời câu hỏi

C2.1:

Kết quả:

```
huy@ubuntu: /home/huy /baste 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):
Nobody has ever shed tears without seeing a coffin
Your input hard-coded password: Nobody has ever shed tears without seeing a coffin
Congrats! You found the hard-coded secret, good job :).
```

Hình 2: Kết quả khi chạy Câu 1

Giải thích: Dựa vào hàm call `_isoc99_scanf` (màu đỏ) để lấy giá trị đầu vào, sau đó chương trình sẽ push kết quả đúng và input của ta vào stack rồi so sánh kết quả xem 2 chuỗi kí tự có tương tự nhau không nhờ call hàm `_strcmp` (màu tím), nếu 2 chuỗi là tương tự thì ta sẽ thành công (màu đen) (Dễ dàng nhận thấy chuỗi kí tự đúng sẽ là s2)

```
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_804926A ; "%[^\\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 ; "Nobody has ever shed tears without seei"...
lea     eax, [ebp+s1]
push    eax ; s1
call    _strcmp
add     esp, 10h
test    eax, eax
jnz     short loc_80487A6
call    success_1
jmp     short loc_80487AB
```

Chuỗi kí tự đúng

Hình 3: Hàm hardCode của Câu 1

C2.2:

Kết quả:

```
Congrats! You found the hard-coded secret, good job! :)
huy@asus:~/Downloads$ ./basic-reverse
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):
6 85
Your input: 6 85
Congrats! You found a secret pair of numbers :).
huy@asus:~/Downloads$
```

Hình 4: Kết quả khi chạy Câu 2

Giải thích:

```
1 int otherhardCode()
2 {
3     int v0; // edx@2
4     int result; // eax@3
5     int u2; // [sp+4h] [bp-14h]@1
6     int v3; // [sp+8h] [bp-10h]@1
7     int v4; // [sp+Ch] [bp-Ch]@1
8
9     getchar();
10    puts("Enter your 2 numbers (separated by space) (option 2):");
11    __isoc99_scanf("%d %d", &v3, &u2);
12    printf("Your input: %d %d\n", v3, u2);
13    v4 = 6;
14    if ( v3 == 6 )
15    {
16        v0 = funny_func(6, *(&funny_seq + 6));
17        if ( v0 == u2 )
18            result = success_2();
19        else
20            result = failed();
21    }
22    else
23    {
24        result = failed();
25    }
26    return result;
27 }
```

Hình 5: Hàm otherhardCode của Câu 2

- Ở lệnh if đang so sánh nếu $v3 == 6$ và $v0 == v2$ thì gọi hàm `success_2()` nên $v3$ và $v0$ là 2 số cần tìm.
- Lệnh if đầu tiên giả sử $v3 = 6$ nên số đầu tiên là 6, sau đó xét tiếp số thứ 2.
- Chương trình gán $v0 = \text{funny_func}(6, *(&\text{funny_seq} + 6))$
- Tìm đến `funny_seq` ta lấy kết quả của ô có địa chỉ là $(\&\text{funny_seq} + 6)$.

```

.rodata:08048B60      public funny_seq
.rodata:08048B60 funny_seq db 0Ah          ; DATA XREF: otherhardCode+5D↑r
.rodata:08048B61      db 0
.rodata:08048B62      db 0
.rodata:08048B63      db 0
.rodata:08048B64      db 3
.rodata:08048B65      db 0
.rodata:08048B66      db 0
.rodata:08048B67      db 0
.rodata:08048B68      db 6
.rodata:08048B69      db 0
.rodata:08048B6A      db 0
.rodata:08048B6B      db 0
.rodata:08048B6C      db 9
.rodata:08048B6D      db 0
.rodata:08048B6E      db 0
.rodata:08048B6F      db 0
.rodata:08048B70      db 1
.rodata:08048B71      db 0
.rodata:08048B72      db 0
.rodata:08048B73      db 0
.rodata:08048B74      db 4
.rodata:08048B75      db 0
.rodata:08048B76      db 0
.rodata:08048B77      db 0
.rodata:08048B78      db 7

```

Hình 6: Stack funny_seq

- Tìm đến hàm **funny_func()** ta thấy hàm **funny_func(6, *(&funny_seq + 6))** trả về kết quả như sau: $6 * (6 + 7) + 7 = 85$

```

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

```

Hình 7: Hàm funny_func

=> Vậy hai số cần tìm là 6 và 85.

C2.3:

Kết quả:

```
huy@asus:~/Downloads$ ./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:
164541536
Enter your password:
1543G3K4F
Your input username: 164541536 and password: 1543G3K4F
Congrats! You found your own username/password pair :).
huy@asus:~/Downloads$
```

Giải thích:

```

1 int userpass()
2 {
3     size_t v0; // ebx@2
4     int result; // eax@3
5     long double v2; // fst7@13
6     size_t v3; // eax@15
7     size_t v4; // edx@16
8     char v5[9]; // [sp+1Ah] [bp-2Eh]@6
9     char v6[10]; // [sp+23h] [bp-25h]@1
10    char s[10]; // [sp+2Dh] [bp-1Bh]@1
11    char v8[5]; // [sp+37h] [bp-11h]@1
12    unsigned int i; // [sp+3Ch] [bp-Ch]@4
13    |
14    v8[0] = 87;
15    v8[1] = 54;
16    v8[2] = 98;
17    v8[3] = 54;
18    v8[4] = 90;
19    getchar();
20    puts("Enter your username:");
21    __isoc99_scanf("%[^\n]", s);
22    getchar();
23    puts("Enter your password:");
24    __isoc99_scanf("%[^\n]", v6);
25    printf("Your input username: %s and password: %s\n", s, v6);
26    if ( strlen(s) == 9 && (v0 = strlen(s), v0 == strlen(v6)) )
27    {
28        for ( i = 0; (signed int)i <= 8; ++i )
29        {
30            if ( (signed int)i > 1 )
31            {
32                if ( (signed int)i > 3 )
33                    v5[i] = v8[8 - i];
34                else
35                    v5[i] = s[i + 2];
36            }
37            else
38            {
39                v5[i] = s[i + 5];
40            }
41        }

```

```

42     for ( i = 0; ; ++i )
43     {
44         v3 = strlen(s);
45         if ( v3 > i )
46         {
47             v2 = ceil((long double)((s[i] + v5[i]) / 2));
48             if ( (long double)v6[i] == v2 )
49                 continue;
50         }
51         break;
52     }
53     v4 = strlen(s);
54     if ( v4 == i )
55         result = success_3();
56     else
57         result = failed();
58 }
59 else
60 {
61     result = failed();
62 }
63 return result;
64 }

```

- Dựa trên đoạn mã giả thì ta có thể chuyển về code C/C++ để dễ tìm ra password thông qua username hơn:

```

6  int main()
7  {
8      size_t v0; // ebx@2
9      int result; // eax@3
10     long double v2; // fst7@13
11     size_t v3; // eax@15
12     size_t v4; // edx@16
13     char v5[9]; // [sp+1Ah] [bp-2Eh]@6
14     char v6[10]; // [sp+23h] [bp-25h]@1
15     string s; // [sp+2Dh] [bp-18h]@1
16     char v8[5]; // [sp+37h] [bp-11h]@1
17     unsigned int i; // [sp+3Ch] [bp-Ch]@4
18     v8[0] = 33;
19     v8[1] = 60;
20     v8[2] = 55;
21     v8[3] = 63;
22     v8[4] = 97;
23     cout << "Enter your username:";
24     cin >> s;
25     if ( s.size() == 9 && (v0 = s.size() ) )
26     {
27         for ( i = 0; (signed int)i <= 8; ++i )
28         {
29             if ( (signed int)i > 1 )
30             {
31                 if ( (signed int)i > 3 )
32                     v5[i] = v8[8 - i];
33                 else
34                     v5[i] = s[i + 2];
35             }
36             else
37             {
38                 v5[i] = s[i + 5];
39             }
40         }
41         for ( i = 0; ; ++i )
42         {
43             v3 = s.size();
44             if ( v3 > i )
45             {
46                 v2 = ceil((long double)((s[i] + v5[i])) / 2);
47                 (long long) v6[i] == v2;
48             }
49             break;
50         }
51         for(int i =0; i< 9; i++)
52             cout << v6[i];
53     }

```


- Kết quả của đoạn code:

```
Enter your username:164541536
1543G3K4F
Process returned 0 (0x0)   execution time : 8.335 s
Press any key to continue.
```

- Phân tích:

- Đoạn code này nhằm giúp ta tìm ra được password cần nhờ vào username nhập vào
- Đầu tiên ta có khai báo một số biến cần thiết, bao gồm **v8** chứa một mảng các ký tự đã được khai báo sẵn (ở đây, các phần tử được khai báo trong mảng **v8** chính là mã ASCII), **s** chứa tên người dùng nhập vào và **v6** là mật khẩu cần tìm
- Sau đó tiến hành kiểm tra xem độ dài của tên người dùng có phù hợp không (9 ký tự).
- Nếu phù hợp thì sẽ tiếp tục lần lượt kiểm tra điều kiện của 1 biến **i** (khởi tạo **i** = 0, cho **i** chạy dần tới 8) nhằm tìm ra lần lượt các ký tự của chuỗi ký tự **v5**
- Tiếp tục ta lại xét 1 biến **i** khác (vẫn khởi tạo bằng 0 rồi cho chạy dần) ở đây ta khởi tạo biến **v3** = chiều dài của username (= 9) rồi cho chạy vòng lặp so sánh dần dần với **i**, nếu **v3** > **i** thì mới tiếp tục
- Thực hiện lấy các ký tự của chuỗi password nhờ lấy trung bình cộng của tổng mã ASCII giữa từng ký tự chuỗi **s** và chuỗi **v5**

- Thực hiện giải tay:

Dựa trên đề cho thì ta có mảng chứa username có 9 ký tự: 164541536, ta sẽ được mã ASCII như sau:

i	0	1	2	3	4	5	6	7	8
s[i]	1	6	4	5	4	1	5	3	6
ASCII	49	54	52	53	52	49	53	51	54

- Xét vòng lặp for đầu tiên: **for unsigned i = 0; i <= 8; ++i**, ta được giá trị của các phần tử mảng **v5** như sau (chú thích: hàng 3 là giá trị cụ thể của mỗi phần tử **v5[i]** bao gồm ký tự hiển thị và mã ASCII tương ứng)

i	0	1	2	3	4	5	6	7	8
v5[i]	s[5]	s[6]	s[4]	s[5]	v8[4]	v8[3]	v8[2]	v8[1]	v8[0]
Giá trị v4[i]	'1' 49	'5' 53	'4' 52	'1' 49	'Z' 90	'6' 54	'b' 98	'6' 54	'W' 87

- Xét vòng lặp for tiếp theo for **i = 0;i<9;++i** ta được phần tử của password như sau:

Lưu ý: các số thập phân ở đây làm tròn xuống

i	0	1	2	3	4
v6[i]	$(s[0]+v5[0])/2$ $= (49+49)/2 =$ 49	$(s[1]+v5[1])/2$ $= (53+54)/2 =$ 53	$(s[2]+v5[2])/2$ $= (52+52)/2 =$ 52	$(s[3]+v5[3])/2$ $= (53+49)/2 =$ 51	$(s[4]+v5[4])/2$ $= (90+52)/2 =$ 71
Giá trị của v6[i]	'1'	'5'	'4'	'3'	'G'

5	6	7	8
$(s[5]+v5[5])/2$ $= (49+54)/2 =$ 51	$(s[6]+v5[6])/2$ $= (53+98)/2 =$ 75	$(s[7]+v5[7])/2$ $= (51+54)/2 =$ 52	$(s[8]+v5[8])/2$ $= (87+54)/2 =$ 70
'3'	'K'	'4'	'F'

Từ bảng trên ta có thể có được password là 1543G3K4F