LẬP TRÌNH HỆ THỐNG– LỚP NT209.L21.ANTN

RE CHALLENGES 3: Recursive

Giảng viên hướng dẫn	Phạm Văn Hậu		ĐIỂM
Sinh viên thực hiện 1	Trần Đức Lương	19521815	

Đây là 1 file thực thi PE 32-bit. Thực hiện chạy thử chương trình cho nhập password rồi tự thoát ra. Mở file bằng IDA phân tích hàm main.

```
mov     byte ptr [ebp+var_4], 2
mov     edx, offset aPassword ; "PASSWORD: "
mov     ecx, ds:?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::ostream std::cout
call    sub_4035A0
```

Có thể thấy sub_4035A0 là hàm cout.

Chương trình lưu chuỗi password nhập vào Src. Ở đây có một loạt đoạn code điều kiện thực hiện những dòng tương tự nhau. Có chuỗi "SUCCESS!.." là chúng ta cần hướng đến, tuy nhiên ở những block code này dùng hàm sub_403190 không phải hàm cout nên ta sẽ bỏ qua phần này.

```
v6 = Src;
v7 = (void **)Src[0];
if ( v58 >= 0x10 )
  v6 = (void **)Src[0];
if ( *(_BYTE *)v6 == 49 )
{
  v60 = 0;
  v61 = 15;
  LOBYTE(Block[0]) = 0;
  sub_403190(Block, "SUCCESS! tell us how this crackme was solved", 0x2Cu);
  if ( v61 >= 0x10 )
  {
    v8 = Block[0];
    if ( v61 + 1 >= 0x1000 )
    {
      v8 = (void *)*((_DWORD *)Block[0] - 1);
      if ( (unsigned int)(Block[0] - v8 - 4) > 0x1F )
        invalid_parameter_noinfo_noreturn();
    }
```

Thực hiện rà code xuống dưới thì thấy xuất hiện hàm sub_401300 cần xem xét. Lí do vì nó xuất hiện hàm sub_4035A0 là hàm cout với dòng chữ "SUCCESS! …". Đây chính là nơi chúng ta cần hướng đến.

```
v54 = 0;
v55 = 15;
LOBYTE(v53[0]) = 0;
sub_403190(v53, "oh... sorry XD XD XD XD", 0x17u);
LOBYTE(v62) = 5;
sub_402550(Src);
sub_401300();
if ( v55 >= 0x10 )
sub_4035A0(v66, "SUCCESS! tell us how this crackme was solved");
```

Tại hàm sub_401300, điều kiện cần để đến với "SUCCESS" là v6 == 6. Ta sẽ đi kiểm tra cách hoạt động của v6.

```
LABEL_78:
  if ( v6 == 6 )
  {
    v16 = (char *)&Src;
    v17 = a5;
    if ( a6 >= 0x10 )
      v16 = (char *)Src;
    if ( a5 != 7 )
      goto LABEL_118;
    v18 = *(_DWORD *)v16;
    v19 = "stopped";
    v65 = 3;
    if ( v18 == *(_DWORD *)"stopped" )
    {
      v16 += 4;
      v65 = -1;
      v19 = "ped";
    }
```

Password nhập vào được lưu ở Src và a5 chính là độ dài chuỗi của Src. Chương trình dùng switch case với v7 sau mỗi lần lặp. Ở đây có 5 case tác động lên v6 (tăng lên 1 ở mỗi case).

```
if ( a5 )
{
  while ( v7 )
  {
    switch ( v7 )
    {
      case 1:
        sub_402550(&Src);
        v10 = Block;
        v9 = Block[0];
        if ( v69 >= 0x10 )
          v10 = (void **)Block[0];
        if ( *((_BYTE *)v10 + 1) != 116 )
        {
          if ( v69 < 0x10 )
            break;
          if ( v69 + 1 >= 0x1000 )
          {
            v9 = (void *)*((_DWORD *)Block[0] - 1);
            if ( (unsigned int)(Block[0] - v9 - 4) > 0x1F )
              goto LABEL_153;
          }
          goto LABEL_76;
```

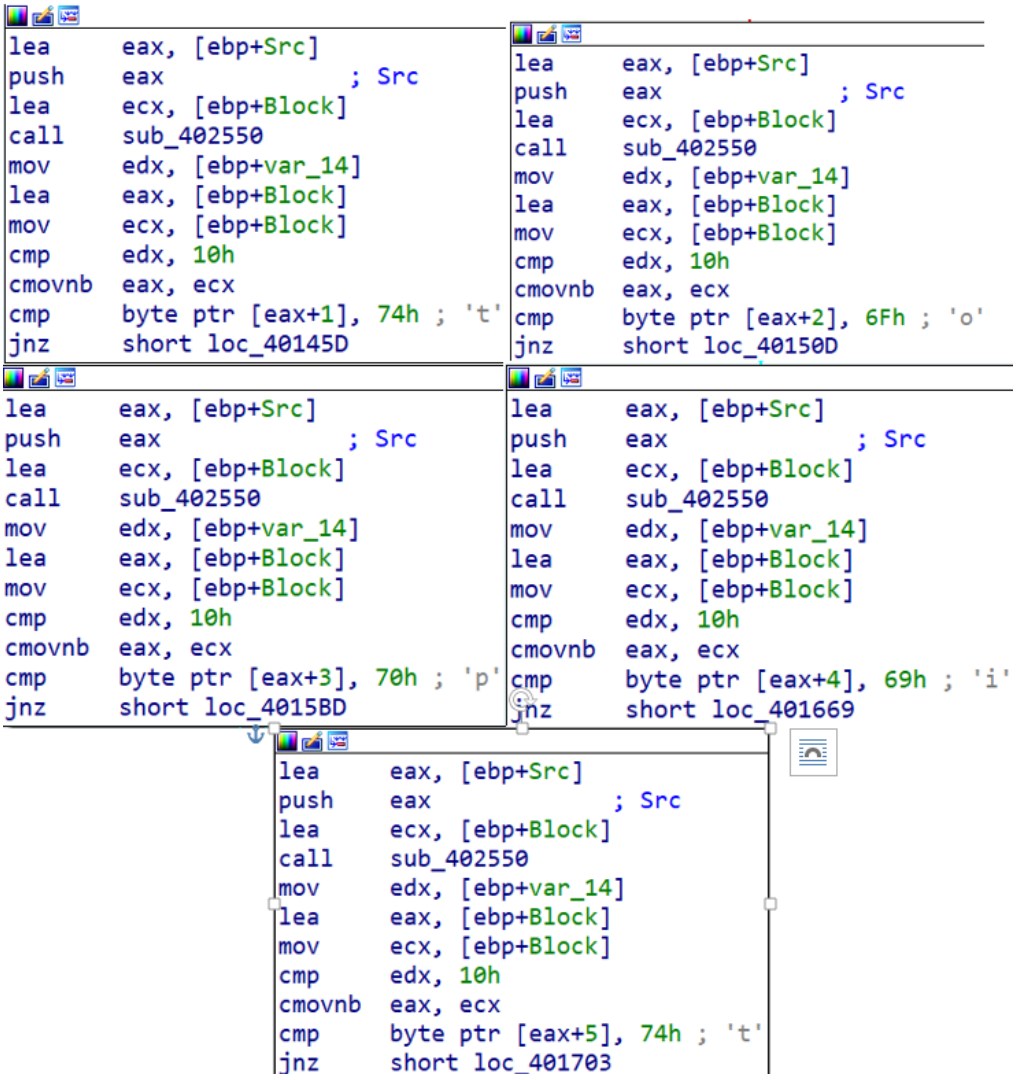
Ban đầu v7 = 0 nên chương trình nhảy xuống đoạn code này. v8 ở đây có lưu đoạn password Src chúng ta nhập vào. Vì ta cần v6 == 6 mà ở trên chỉ có 5 case có thể tăng giá trị v6 nên block code này cũng cần phải tăng v6 lên 1. Khi đó kí tự đầu tiên của password phải là ‘s’ (= 115) => Src[0] = ‘s’.

```
sub_682550(&Src);
v8 = Block;
v9 = Block[0];
if ( v69 >= 0x10 )
  v8 = (void **)Block[0];
if ( *(_BYTE *)v8 != 115 )
{
  if ( v69 < 0x10 )
    goto LABEL_77;
LABEL_45:
    sub_684015(v9);
    ++v6;
    goto LABEL_77;
  }
```

Sau đó, chương trình nhảy đến LABEL_77 để tăng v7 lên 1, nếu v7 >= a5 tức là độ dài chuỗi nhập vào thì sẽ nhảy đến LABEL_78 để kiểm tra điều kiện v6 == 6 đã nêu ở trên.

```
LABEL_77:
      if ( ++v7 >= a5 )
        goto LABEL_78;
    }
```

Chương trình tiếp tục với các case 1 đến case 5. Dưới đây là danh sách:



```
lea     eax, [ebp+Src]
push    eax             ; Src
lea     ecx, [ebp+Block]
call    sub_402550
mov     edx, [ebp+var_14]
lea     eax, [ebp+Block]
mov     ecx, [ebp+Block]
cmp     edx, 10h
cmovnb  eax, ecx
cmp     byte ptr [eax+1], 74h ; 't'
jnz     short loc_40145D
```

```
lea     eax, [ebp+Src]
push    eax             ; Src
lea     ecx, [ebp+Block]
call    sub_402550
mov     edx, [ebp+var_14]
lea     eax, [ebp+Block]
mov     ecx, [ebp+Block]
cmp     edx, 10h
cmovnb  eax, ecx
cmp     byte ptr [eax+2], 6Fh ; 'o'
jnz     short loc_40150D
```

```
lea     eax, [ebp+Src]
push    eax             ; Src
lea     ecx, [ebp+Block]
call    sub_402550
mov     edx, [ebp+var_14]
lea     eax, [ebp+Block]
mov     ecx, [ebp+Block]
cmp     edx, 10h
cmovnb  eax, ecx
cmp     byte ptr [eax+3], 70h ; 'p'
jnz     short loc_4015BD
```

```
lea     eax, [ebp+Src]
push    eax             ; Src
lea     ecx, [ebp+Block]
call    sub_402550
mov     edx, [ebp+var_14]
lea     eax, [ebp+Block]
mov     ecx, [ebp+Block]
cmp     edx, 10h
cmovnb  eax, ecx
cmp     byte ptr [eax+4], 69h ; 'i'
jnz     short loc_401669
```

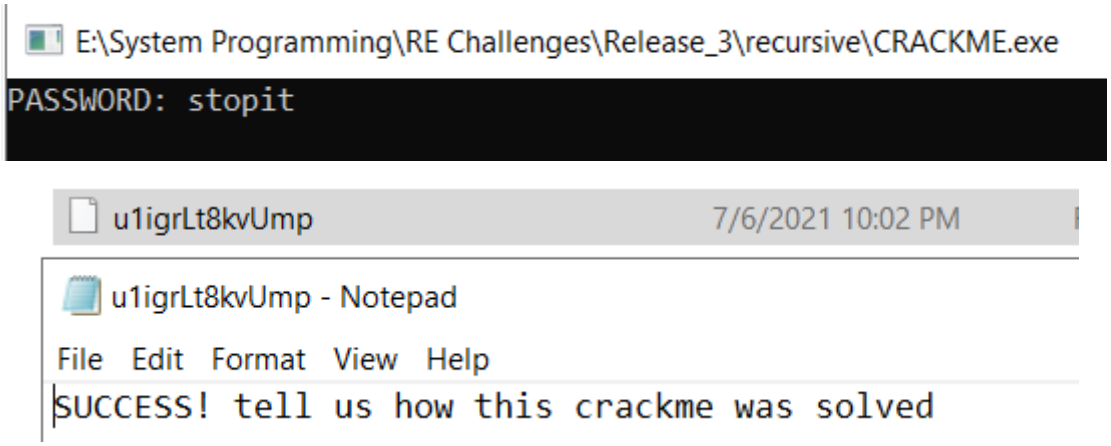
```
lea     eax, [ebp+Src]
push    eax             ; Src
lea     ecx, [ebp+Block]
call    sub_402550
mov     edx, [ebp+var_14]
lea     eax, [ebp+Block]
mov     ecx, [ebp+Block]
cmp     edx, 10h
cmovnb  eax, ecx
cmp     byte ptr [eax+5], 74h ; 't'
jnz     short loc_401703
```

Vì mục tiêu của ta là tăng v6 thêm 5 nữa nên cả 5 case này cần nhảy đến LABEL_45 để tăng v6. Khi đó dựa điều kiện kiểm tra ở các case, kết hợp với

Src[0] = ‘s’ => Src[0:6] = “stopit”. Khi đó v6 == 6. Chương trình sẽ thoát vòng lặp trên khi v7 >= a5. Để tối ưu ta sẽ chỉ cần nhập password = “stopit”. Còn nếu nhập nhiều hơn thì vẫn được vì khi đó v7 sẽ tăng đến khi >= a5 mà không ảnh hưởng đến v6.

```
LABEL_148:
            sub_401120(Block, v63, v64);
            LOBYTE(v70) = 1;
            memset(v66, 0, sizeof(v66));
            v57 = Block;
            if ( v69 >= 0x10 )
              v57 = (void **)Block[0];
            sub_4032D0(v57, v60, v61, v62);
            *(int *)((char *)v66 + *(_DWORD *)(v66[0] + 4)) = (int)&std::ofstream::`vftable';
            *(int *)((char *)&v65 + *(_DWORD *)(v66[0] + 4)) = *(_DWORD *)(v66[0] + 4) - 104;
            LOBYTE(v70) = 2;
            sub_4035A0(v66, "SUCCESS! tell us how this crackme was solved");
            std::ios::clear((char *)v66 + *(_DWORD *)(v66[0] + 4), 0, 0);
            sub_401A80(v66);
```

Chương trình đi vào block code if(v6 == 6) thì đến LABEL_148 xuất hiện các hàm sub_401120 là tạo file với tên Random ở cùng thư mục, hàm sub_4032D0 thực hiện mở file đó và sub_4035A0 chính là ghi chuỗi “SUCCESS!...” vào file đó. Thực hiện nhập password = “stopit” thì chương trình nhảy đến LABEL_148 và tạo ra file “u1igrLt8kvUmp” có nội dùng “SUCCESS!”. Điều này chứng tỏ thành công.



Có thể nhập password = “stopit123” thì chương trình vẫn thành công, miễn là 6 kí tự đầu là chuỗi “stopit”.