

REloaded

Thursday, February 17, 2022 11:01 AM

Level 0

Looking through the strings we quickly find some flag looking strings

```
004b1033 77 00 00 00 s_L3v3l2340_is_0026_004b1045 XREF:libgcc_s_dw2-1.dll
004b1033 53 76 ds "L3v3l2340_is_0026_004b1045"
004b1033 6c 5a ds "That was easy...Bruh!!!_004b1057 XREF:libgcc_s_dw2-1.dll
004b1033 84 30 ds "That was easy...Bruh!!!"
004b1033 68 61 ds "That was easy...Bruh!!!_004b1057 XREF:libgcc_s_dw2-1.dll
004b1033 20 77 ds "That was easy...Bruh!!!"
004b1033 73 20 ds "That was easy...Bruh!!!"
004b1033 6f 6e ds "Dont worry_its_a_start_:_004b1070 XREF:libgcc_s_dw2-1.dll
004b1033 20 57 ds "Dont worry_its_a_start_:)"
004b1033 69 39 ds "Dont worry_its_a_start_:)"
```

Level 1

Checking the strings again, we find some interesting items

004b1000	libgcc_s_dw2-1.dll	"libgcc_s_dw2-1.dll"	ds
004b1013	register_frame_info	"register_frame_info"	ds
004b1029	deregister_frame_info	"deregister_frame_info"	ds
004b1045	Thats ur lucky number !!!	"Thats ur lucky number !!!"	ds
004b1059	Try again	"Try again"	ds
004b106a	N00b a day, pro for life	"N00b a day, pro for life"	ds

But nothing resembling a flag...

We find an interesting comparison that references some of these strings

```
00401410 75 06 JNZ LAB_0040142d
00401417 c7 04 24 MOV dword ptr [ESP]=local_1c,s_Thats_ur_lucky_num
00401417 45 10 4b MOV dword ptr [ESP]=local_1c,s_Thats_ur_lucky_num
00401426 e8 75 b1 CALL MSVCRT.DLL::printf int printf(char * _Format, ...);
00401426 01 00 JMP LAB_0040143a
0040142d LAB_0040142d XREF[1]: 0040141d(j)
0040142d e7 04 24 MOV dword ptr [ESP]=local_1c,s_Try_again_004_="Try again "
0040142d 45 10 4b MOV dword ptr [ESP]=local_1c,s_Try_again_004_="Try again "
00401434 e8 5f b1 CALL MSVCRT.DLL::puts int puts(char * _Str);
```

It looks like we want to get to the "That's ur lucky number"

We have a hardcoded comparison

```
00401434 81 00 CMP dword ptr [EBP + param_1],00000000
```

Converting that to a decimal

```
00401434 81 00 CMP dword ptr [EBP + param_1],0
```

We validate that by running the program

```
PS C:\Users\ssali\Downloads> .\Level.exe
Thats ur lucky number !!!
PS C:\Users\ssali\Downloads>
```

Level 2

Looking for strings again, we find some usefull stuff

004b1029	deregister_frame_info	"deregister_frame_info"	ds
004b1045	Get Ready For L4 ;)	"Get Ready For L4 ;)"	ds
004b105c	In order to advance you ...	"In order to advance you have to break your mindset"	ds
004b108f	Wow Ur At L3?	"Wow Ur At L3?"	ds

```
PS C:\Users\ssali\Downloads> .\'Level (1).exe'
Wow Ur At L3?
PS C:\Users\ssali\Downloads> .\'Level (1).exe' 1234
In order to advance you have to break your mindset
```

It looks like we want the "Get Ready For L4" string

We patch the comparison that skips over this instruction

```
00401446 e8 95 b1 CALL MSVCRT.DLL::strcmp int strcmp(char * _Str1, char * _Str2);
00401446 01 00 JMP LAB_00401470
0040144b 85 c0 TEST EAX,EAX
0040144d 74 21 JZ LAB_00401470
0040144f e7 04 24 MOV dword ptr [ESP]=local_3c,s_Get_Ready_For_L4_;"Get Ready For L4 ;)"
0040144f 45 10 4b MOV dword ptr [ESP]=local_3c,s_Get_Ready_For_L4_;"Get Ready For L4 ;)"
00401456 e8 bd b1 CALL MSVCRT.DLL::puts int puts(char * _Str);
```

and run the program again

```
PS C:\Users\ssali\Downloads> .\'Level (1).exe' 1234
Get Ready For L4 ;)
```

Level 3

We start off looking at the strings again

004b1000	libgcc_s_dw2-1.dll	"libgcc_s_dw2-1.dll"	ds
004b1045	Rooted !!!	"Rooted !!!"	ds
004b1034	Enter the flag ::	"Enter the flag ::"	ds
004b10a0	__gnu_cxx::__concurrence_lock...	"__gnu_cxx::__concurrence_lock_error"	ds
004b10a4	__gnu_cxx::__concurrence_unlock...	"__gnu_cxx::__concurrence_unlock_error"	ds

If we run the program in a debugger, we should be able to see the flag once it reaches this point

```
LAB_004014f6 XREF[1]: 004014ce(j)
004014f6 8d 45 c4 LEA EAX=>local_40,[EBP + -0x3c]
004014f9 89 44 24 04 MOV dword ptr [ESP + local_58],EAX
004014fd 8b 45 08 MOV EAX,dword ptr [EBP + param_1]
00401500 89 04 24 MOV dword ptr [ESP]=local_5c,EAX
00401503 e8 58 b1 CALL MSVCRT.DLL::strcmp int strcmp(char * _Str1, char * _Str2);
00401503 01 00 JMP LAB_00401508
00401508 85 c0 TEST EAX,EAX
```

We throw the program into Immunity Debugger and find the strcmp again


```

1
2 void __cdecl FUN_00401453(char *param_1)
3
4 {
5     size_t sVar1;
6     undefined4 uVar2;
7     int local_10;
8
9     sVar1 = strlen(param_1);
10    for (local_10 = 0; local_10 < (int)sVar1; local_10 = local_10 + 1) {
11        uVar2 = FUN_00401410(local_10);
12        if ((char)uVar2 == '\0') {
13            param_1[local_10] = param_1[local_10] ^ '7';
14        }
15        else {
16            param_1[local_10] = param_1[local_10] ^ (byte)local_10;
17        }
18    }
19    printf("%s",param_1);
20    return;
21 }
22

```

Looks like param_1 is enc_flag

and enc_flag is getting passed into this function

```
void __cdecl FUN_00401453(char *enc_flag)
```

Looking for cross-references

References to FUN_00401453 - 1 locations [CodeBrowser: RE/Level_Final.exe]

Edit Help

References to FUN_00401453 - 1 locations

Location	Label	Code Unit	Context
00401532		CALL FUN_00401453	UNCONDITIONAL_CALL

```

undefined4 FUN_004014e6(void)
{
    undefined4 local_2a;
    undefined4 local_26;
    undefined4 local_22;
    undefined4 local_1e;
    undefined4 local_1a;
    undefined4 local_16;
    undefined2 local_12;

    FUN_0040c000();
    local_2a = 0x6e616c41;
    local_26 = 0x72755420;
    local_22 = 0x20676e69;
    local_1e = 0x20736157;
    local_1a = 0x65472061;
    local_16 = 0x7375696e;
    local_12 = 0x65;
    FUN_00401453((char *) &local_2a);
    return 0;
}

```

Renaming that function for clarity

```

FUN_0040c000();
local_2a = 0x6e616c41;
local_26 = 0x72755420;
local_22 = 0x20676e69;
local_1e = 0x20736157;
local_1a = 0x65472061;
local_16 = 0x7375696e;
local_12 = 0x65;
print_enc_flag((char *) &local_2a);
return 0;

```

Converting the hex stored at local_2a to strings, we get our flag

Address	Disassembly	Comment
14ef e8 0c ab	CALL	FUN_0040c000
00 00		
14f4 c7 44 24	MOV	dword ptr [ESP + local_2a], "████"
16 41 6c		
61 6e		
14fc c7 44 24	MOV	dword ptr [ESP + local_26], "████"
1a 20 54		
75 72		
1504 c7 44 24	MOV	dword ptr [ESP + local_22], "████"
1e 69 6e		
67 20		
150c c7 44 24	MOV	dword ptr [ESP + local_1e], "████"
22 57 61		
73 20		
1514 c7 44 24	MOV	dword ptr [ESP + local_1a], "████"
26 61 20		
47 65		
151c c7 44 24	MOV	dword ptr [ESP + local_16], "████"
2a 6e 69		
75 73		
1524 66 c7 44	MOV	word ptr [ESP + local_12], 'e'

```

undefined4 FUN_004014e6(void)
{
    undefined4 local_2a;
    undefined4 local_26;
    undefined4 local_22;
    undefined4 local_1e;
    undefined4 local_1a;
    undefined4 local_16;
    undefined2 local_12;

    FUN_0040c000();
    local_2a = L'\x6e616c41';
    local_26 = L'\x72755420';
    local_22 = L'\x20676e69';
    local_1e = L'\x20736157';
    local_1a = L'\x65472061';
    local_16 = L'\x7375696e';
    local_12 = L'e';
    print_enc_flag((char *) &local_2a);
    return 0;
}

```