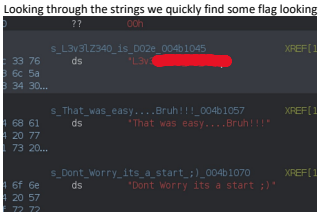
REloaded

Thursday, February 17, 2022 11:01 AM

Level 0

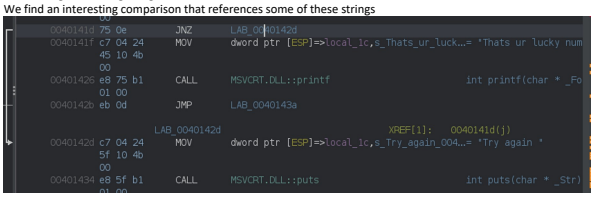
Looking through the strings we quickly find some flag looking strings



Level 1

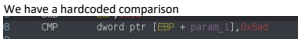
Checking the strings again, we find some interesting items

But nothing resembling a flag...
We find an interesting comparison that references some of these strings

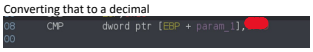


It looks like we want to get to the "That's ur lucky number"

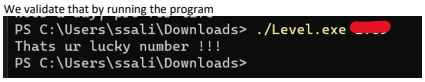
We have a hardcoded comparison



Converting that to a decimal

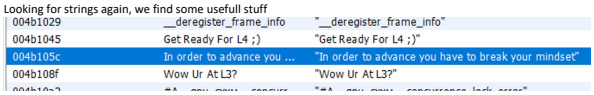


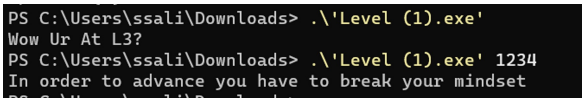
We validate that by running the program



Level 2

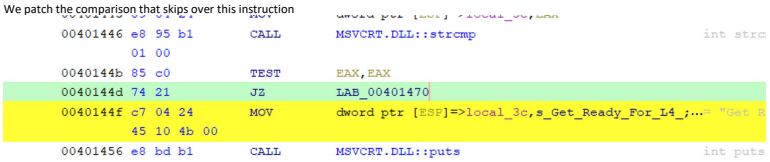
Looking for strings again, we find some usefull stuff



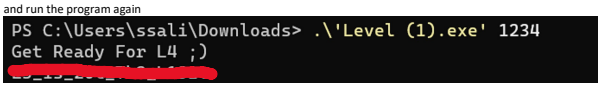


It looks like we want the "Get Ready For L4" string

We patch the comparison that skips over this instruction

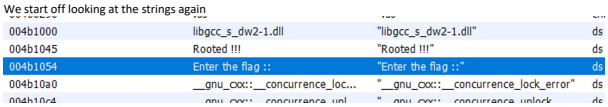


and run the program again

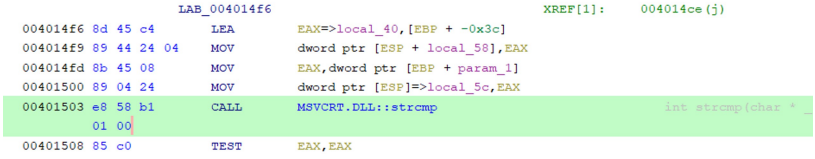


Level 3

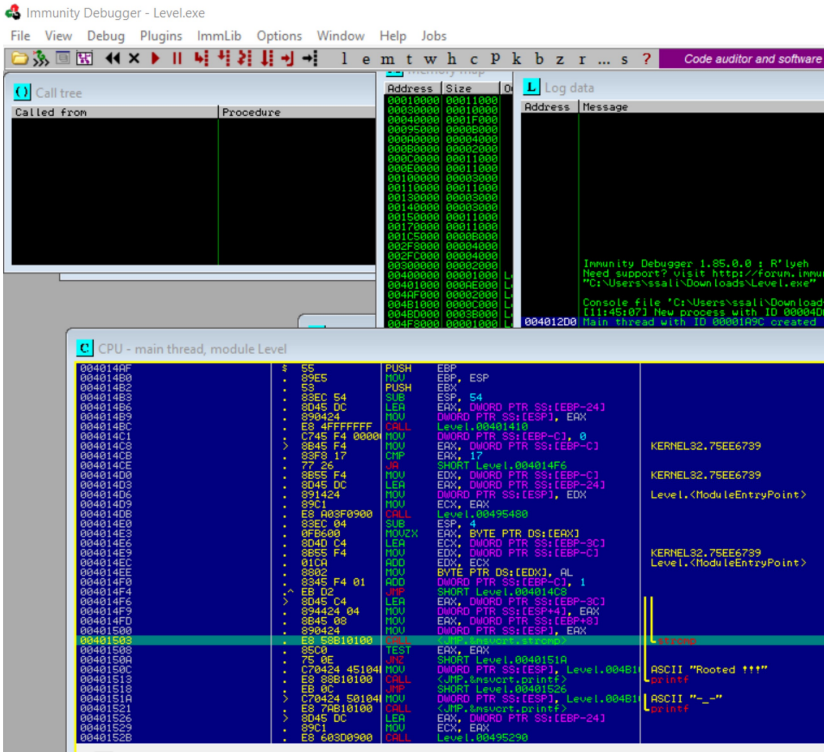
We start off looking at the strings again



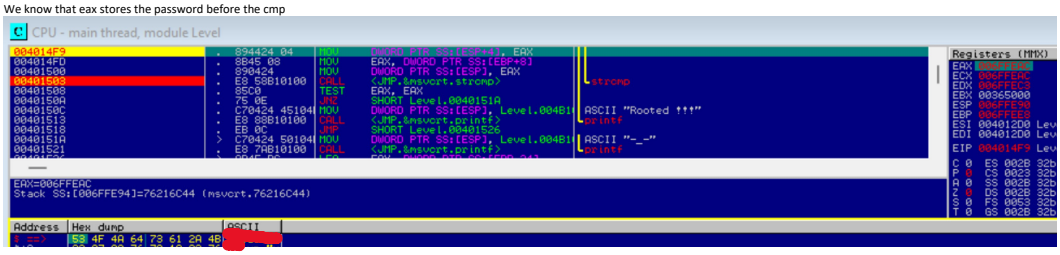
If we run the program in a debugger, we should be able to see the flag once it reaches this point



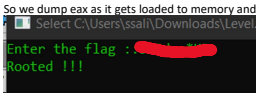
We throw the program into Immunity Debugger and find the strcmp again



We know that eax stores the password before the cmp



So we dump eax as it gets loaded to memory and find the password



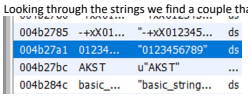
Password: SC[redacted]

Getting the Rooted !!! message did not give use the flag...so we keep reversing and find some strings in another function that look like a flag

```
                    FUN_00401410                                    XREF[1]
00401410 55              PUSH    EBP
00401411 89 e5           MOV     EBP,ESP
00401413 53              PUSH    EBX
00401414 83 ec 34        SUB     ESP,0x34
00401417 c7 45 e5        MOV     dword ptr [EBP + local_1f],"cMHT"
         54 48 4d 63
0040141e c7 45 e9        MOV     dword ptr [EBP + local_1b],"L-ft"
         74 66 2d 4c
00401425 66 c7 45        MOV     word ptr [EBP + local_17],'4'
         ed 34 00
0040142b c7 45 f0        MOV     dword ptr [EBP + local_14],'\a'
         07 00 00 00
00401432 c7 45 f4        MOV     dword ptr [EBP + local_10],0x0
         00 00 00 00
```

Level 4

Looking through the strings we find a couple that stand out



AKST:0123456789

Running the program we get the following

```
PS C:\Users\ssali\Downloads> .\Level_Final.exe
Amcm$QBu^YP+`lD$V1pvY^BdR
PS C:\Users\ssali\Downloads>
```

AmcmĮQBu^YP+`lDĮV1pvY^BdR

Lets find where this is getting printed

```
void __cdecl FUN_00401453(char *param_1)

{
  size_t sVar1;
  undefined4 uVar2;
  int local_10;

  sVar1 = strlen(param_1);
  for (local_10 = 0; local_10 < (int)sVar1; local_10 = local_10 + 1) {
    uVar2 = FUN_00401410(local_10);
    if ((char)uVar2 == '\0') {
      param_1[local_10] = param_1[local_10] ^ '7';
    }
    else {
      param_1[local_10] = param_1[local_10] ^ (byte)local_10;
    }
  }
  printf("%s",param_1);
  return;
}
```

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

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

```
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;
}
```