REVERSE ENGINEERING

03 – STRUCTURES AND CALLING CONVENTIONS

STRUCTURES AND CALLING CONVENTIOPNS

Lab Description: The ability of a reverse engineer to identify the use of structures and calling conventions will greatly enhance their ability to analyze a program from disassembly. This lab will require the student to identify the use of structures, create custom structures in IDA Pro and identify calling conventions.

Lab Environment: IDA Pro Educational

Lab Files that are Needed: labFile.exe

Lab – Complete the following tasks

1. Identify the use of the structure, what is the size of the structure?

2. Create a structure in IDA that represents the identified structure. Ensure that each member of your structure aligns with how it is used in the program.

3. Rename the members to something appropriate based off of your analysis.

4. In IDA View-A, add the structure offsets for each variable used.

5. There is a single regular function called in main, what calling convention does it use? What does it expect as an argument?

WHAT TO SUBMIT

Submit a Microsoft Word document or PDF that includes answers to the questions posed.

1. The size is 0x38.

```
push    ebp
mov     ebp, esp
push    ecx
push    38h ; '8'       ; Size
call    _malloc
add     esp, 4
mov     [ebp+var_4], eax
```

2.

```
00000000
00000000 student_data    struc ; (sizeof=0x38, mappedto_60)
00000000 id              dd ?                    ; XREF: _main+14/w
00000004 year            dd ?                    ; XREF: _main+22/w
00000008 gpa             dd ?                    ; XREF: _main+31/w
0000000C first           db 20 dup(?)            ; XREF: sub_401000+31/o
0000000C                                         ; _main+3E/o
00000020 last            db 20 dup(?)            ; XREF: _main+52/o
00000034 middle          dd ?                    ; XREF: _main+61/w
00000038 student_data    ends
00000038
```

3.

```
00000000
00000000 student_data    struc ; (sizeof=0x38, mappedto_60)
00000000 id              dd ?                    ; XREF: _main+14/w
00000004 year            dd ?                    ; XREF: _main+22/w
00000008 gpa             dd ?                    ; XREF: _main+31/w
0000000C first           db 20 dup(?)            ; XREF: sub_401000+31/o
0000000C                                         ; _main+3E/o
00000020 last            db 20 dup(?)            ; XREF: _main+52/o
00000034 middle          dd ?                    ; XREF: _main+61/w
00000038 student_data    ends
00000038
```

4.

```
push    ebp
mov     ebp, esp
push    ecx
push    38h ; '8'       ; Size
call    _malloc
add     esp, 4
mov     [ebp+var_4], eax
mov     eax, [ebp+var_4]
mov     [eax+student_data.id], 772114
mov     ecx, 4
mov     edx, [ebp+var_4]
mov     word ptr [edx+student_data.year], cx
mov     eax, [ebp+var_4]
movss   xmm0, ds:dword_411150
movss   [eax+student_data.gpa], xmm0
push    offset Source   ; "George"
mov     ecx, [ebp+var_4]
add     ecx, student_data.first
push    ecx             ; Destination
call    _strcpy
add     esp, 8
push    offset aSmith   ; "Smith"
mov     edx, [ebp+var_4]
add     edx, student_data.last
push    edx             ; Destination
call    _strcpy
add     esp, 8
mov     eax, [ebp+var_4]
mov     byte ptr [eax+student_data.middle], 46h ; 'F'
mov     ecx, [ebp+var_4]
push    ecx
call    print_stuff
xor     eax, eax
mov     esp, ebp
pop     ebp
retn
_main endp
```

5. The calling convention used is stdcall. It expects a pointer to the structure as an argument.