# Modifying binary executables

**Overview:**

**This paper will cover the modification of binary executables, integers, bool and conditional statements.**

Modifying strings:

The offset in strings is refereed to as *radix.*The syntax to obtain the radix is as follows

```
strings -t d binary | grep thingYourLookingFor
4455 thingYourLookingFor
```

4455 in this case is the *radix.*

Next create a text file that will serve as a string to be inserted.

```
echo “insersion string” > inject.txt
```

now to use dd to input it

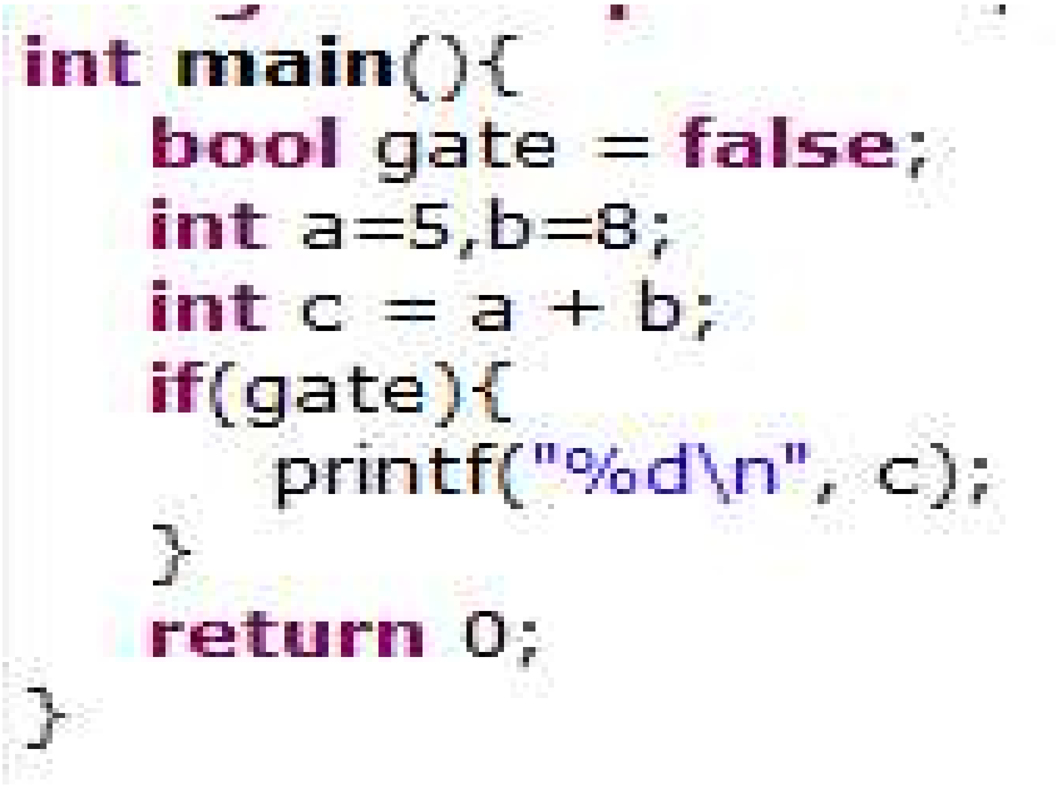
```
dd if=inject.txt of=binary obs=1 seek=4455 conv=notrunc
```

The binary will now execute with the contents of inject.txt in place of thingYourLookingFor

Breaking into if statements:

There are two tools required; objdump and a hexeditor.

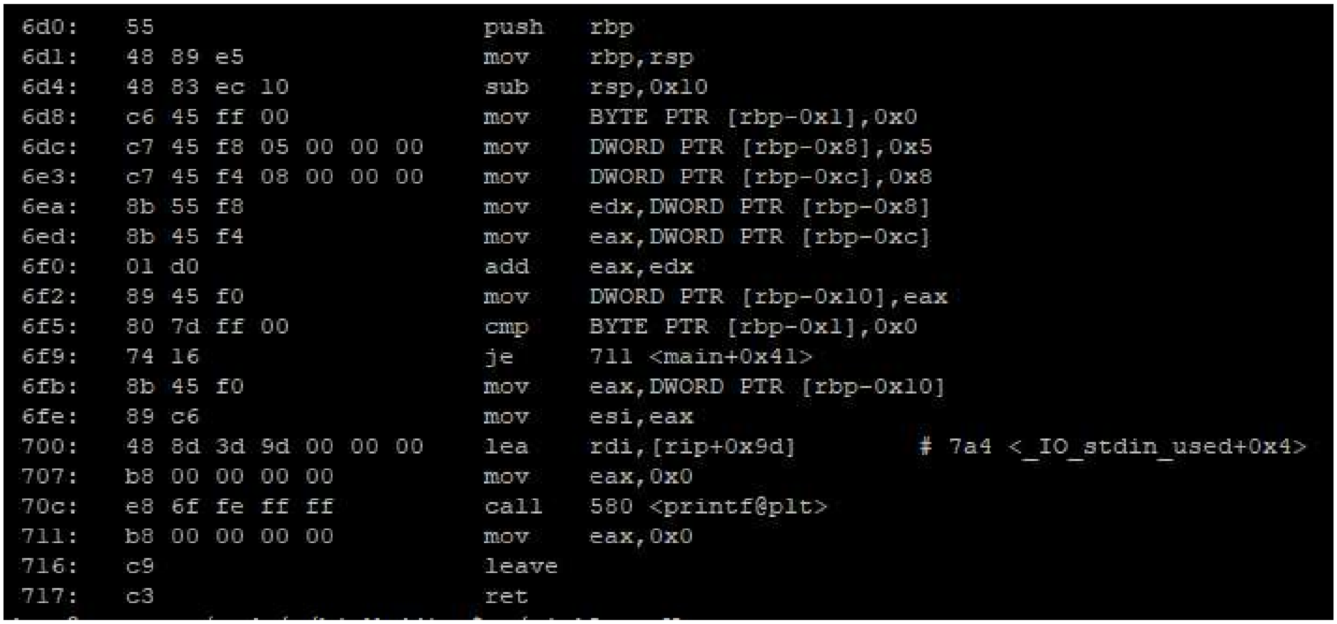
For this example we will include the source code:



```
int main(){
    bool gate = false;
    int a=5,b=8;
    int c = a + b;
    if(gate){
        printf("%d\n", c);
    }
    return 0;
}
```

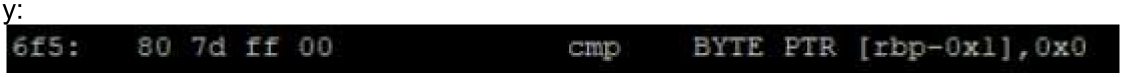
Note: There is no getting into that if statment without modifying the compiled binary.

Next we use objdump to dump the main function in assembler.
objdump -f -D -M intel pickApartVars.o | grep main.: -A20



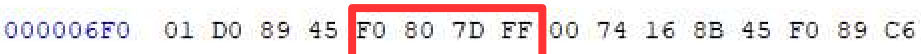
```
6d0:   55                      push   rbp
6d1:   48 89 e5                mov    rbp,rsp
6d4:   48 83 ec 10             sub    rsp,0x10
6d8:   c6 45 ff 00             mov    BYTE PTR [rbp-0x1],0x0
6dc:   c7 45 f8 05 00 00 00    mov    DWORD PTR [rbp-0x8],0x5
6e3:   c7 45 f4 08 00 00 00    mov    DWORD PTR [rbp-0xc],0x8
6ea:   8b 55 f8                mov    edx,DWORD PTR [rbp-0x8]
6ed:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
6f0:   01 d0                   add    eax,edx
6f2:   89 45 f0                mov    DWORD PTR [rbp-0x10],eax
6f5:   80 7d ff 00             cmp    BYTE PTR [rbp-0x1],0x0
6f9:   74 16                   je     711 <main+0x41>
6fb:   8b 45 f0                mov    eax,DWORD PTR [rbp-0x10]
6fe:   89 c6                   mov    esi,eax
700:   48 8d 3d 9d 00 00 00    lea    rdi,[rip+0x9d]        # 7a4 <_IO_stdin_used+0x4>
707:   b8 00 00 00 00          mov    eax,0x0
70c:   e8 6f fe ff ff          call   580 <printf@plt>
711:   b8 00 00 00 00          mov    eax,0x0
716:   c9                      leave
717:   c3                      ret
```

From the programs binary we've disassembled we can identify the compiled if statement in assembly:



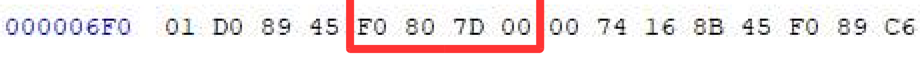
```
6f5:   80 7d ff 00             cmp    BYTE PTR [rbp-0x1],0x0
```

We can force our way into this if statement by modifying the binary executable in a hex editor:



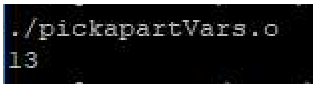
```
000006F0  01 D0 89 45 F0 80 7D FF 00 74 16 8B 45 F0 89 C6
```

Change the FF to 00



```
000006F0  01 D0 89 45 F0 80 7D 00 00 74 16 8B 45 F0 89 C6
```

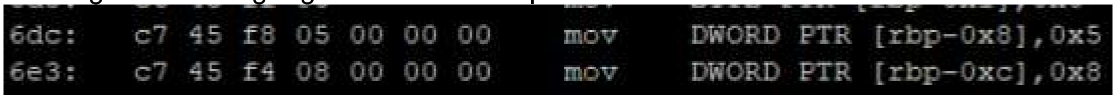
We have know snipped the bolt on the if statement



```
./pickapartVars.o
13
```

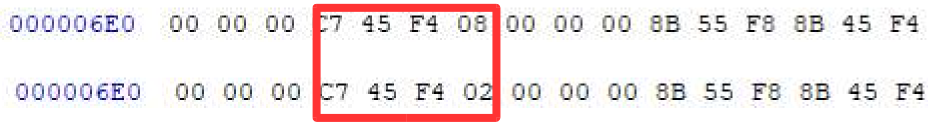
Modifying integers:

The integers are being registered as dword ptr's.

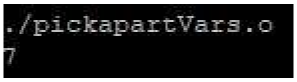


```
6dc:   c7 45 f8 05 00 00 00    mov    DWORD PTR [rbp-0x8],0x5
6e3:   c7 45 f4 08 00 00 00    mov    DWORD PTR [rbp-0xc],0x8
```

We can modify the values compiled in the program here after c7 45 f8 the following value stored in hex transforms into the decimal number. Lets change b that was assigned to 8 to 2:



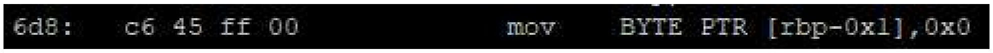
```
000006E0  00 00 00 C7 45 F4 08 00 00 00 8B 55 F8 8B 45 F4

000006E0  00 00 00 C7 45 F4 02 00 00 00 8B 55 F8 8B 45 F4
```



```
./pickapartVars.o
7
```

modifying boolean values:

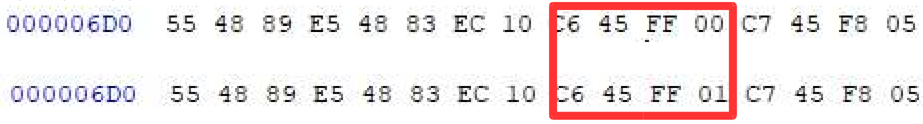
The next example is recompiled from source code and removes previous modifications made in the hex editor. In this example we will enter the if statement by changing the declaration of the boolean value to true.



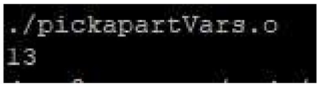
```
6d8:   c6 45 ff 00             mov    BYTE PTR [rbp-0x1],0x0
```

The last byte in that line is 0 (false).

Let's make that true:



```
000006D0  55 48 89 E5 48 83 EC 10 C6 45 FF 00 C7 45 F8 05

000006D0  55 48 89 E5 48 83 EC 10 C6 45 FF 01 C7 45 F8 05
```



```
./pickapartVars.o
13
```