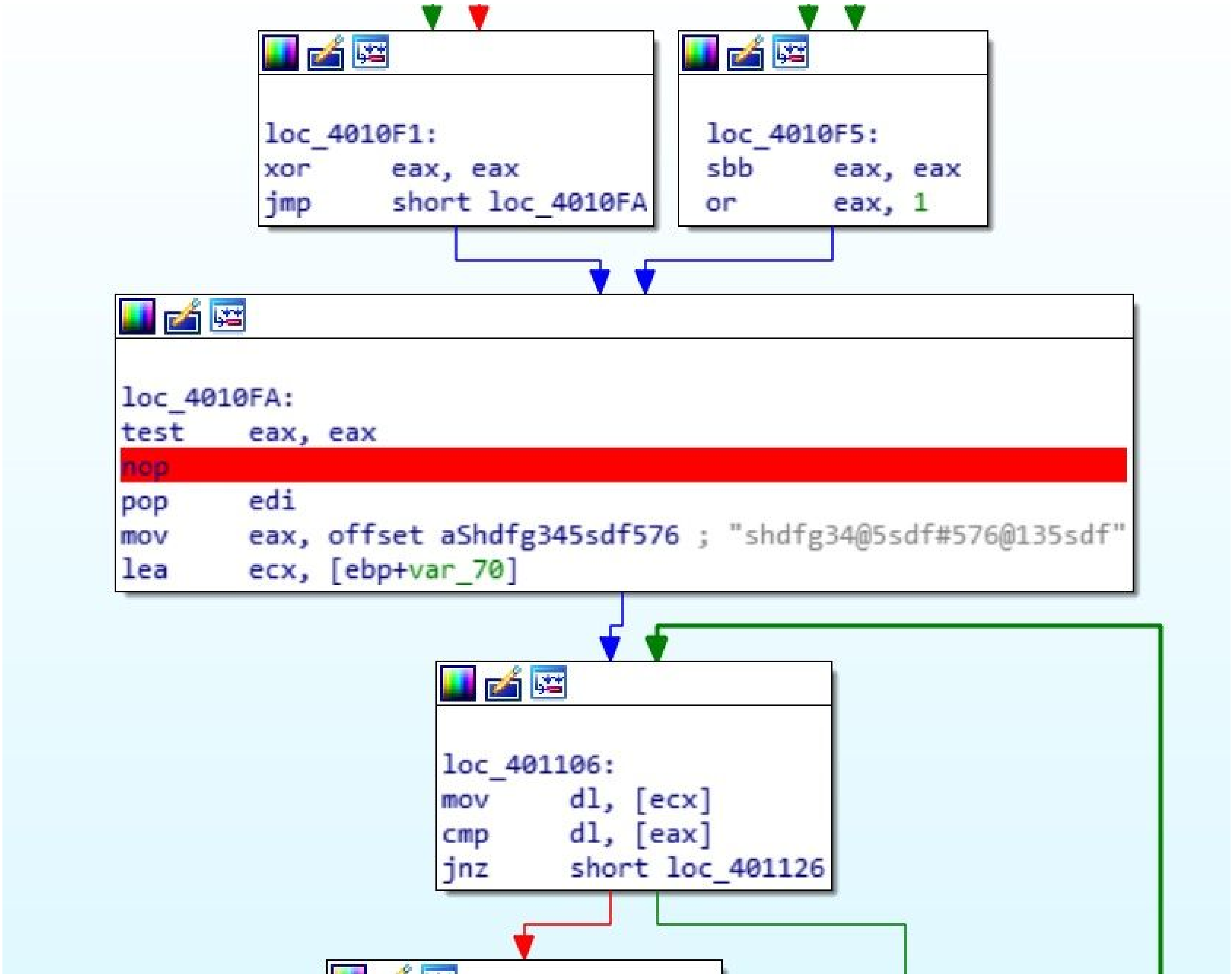
David Huang
dhuan008@ucr.edu

Part 1:
Here's your flag:**34gdfh340234**

Changed the JNZ (opcode 75) to NOP (opcode 90). Highlighted in red. No instruction to goto incorrect username, so as long as there isn't a overflow or something we will always get the flag.. I did the same thing for password check.

Address	Original bytes	Patched bytes
00000000004010FC	75	90



```
loc_4010F1:
xor     eax, eax
jmp     short loc_4010FA
```

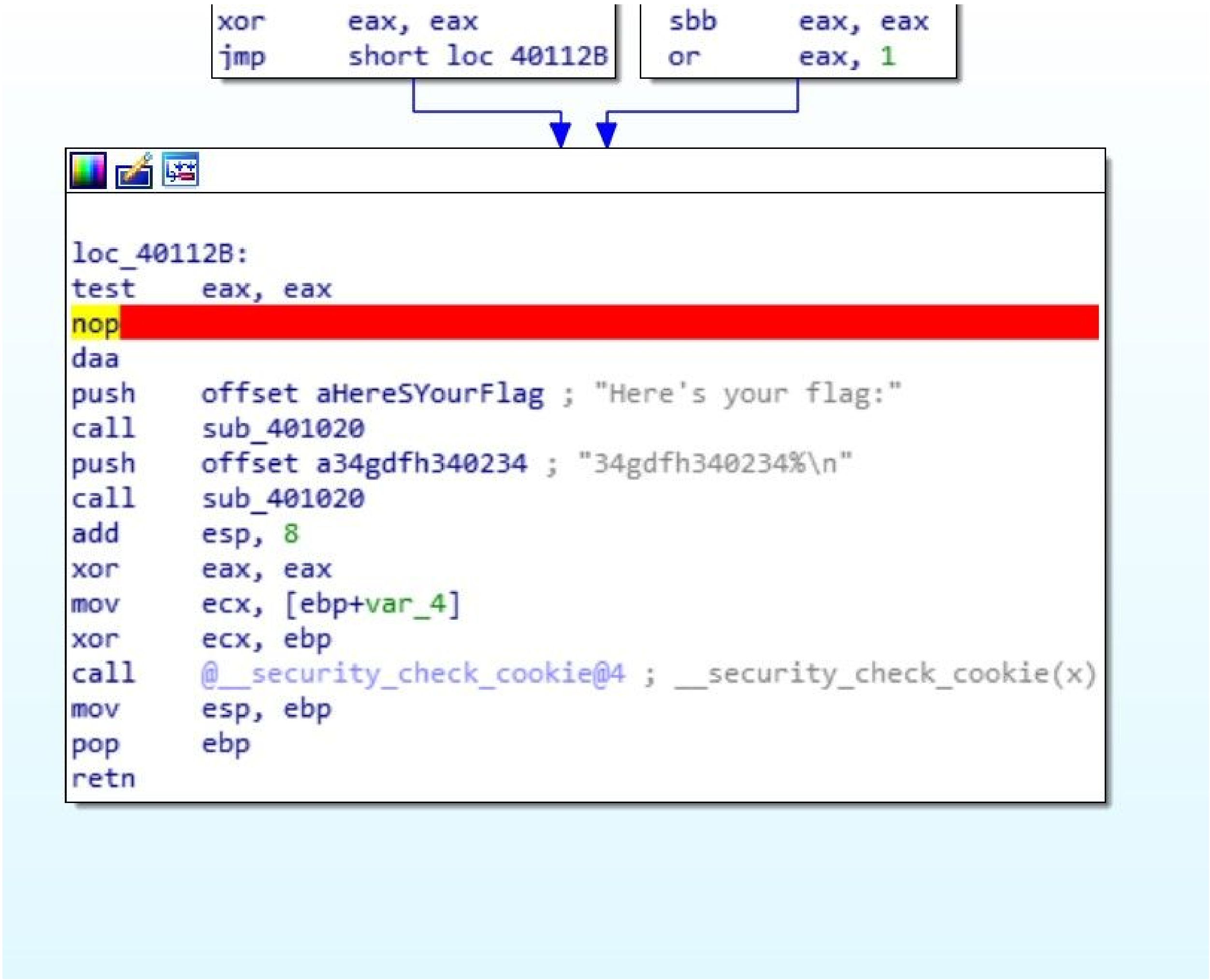
```
loc_4010F5:
sbb     eax, eax
or      eax, 1
```

```
loc_4010FA:
test    eax, eax
nop
pop     edi
mov     eax, offset aShdfg345sdf576 ; "shdfg34@5sdf#576@135sdf"
lea     ecx, [ebp+var_70]
```

```
loc_401106:
mov     dl, [ecx]
cmp     dl, [eax]
jnz     short loc_401126
```

Password - Same logic as above.

Address	Original bytes	Patched bytes
000000000040112D	75	90



```
xor     eax, eax
jmp     short loc 40112B
```

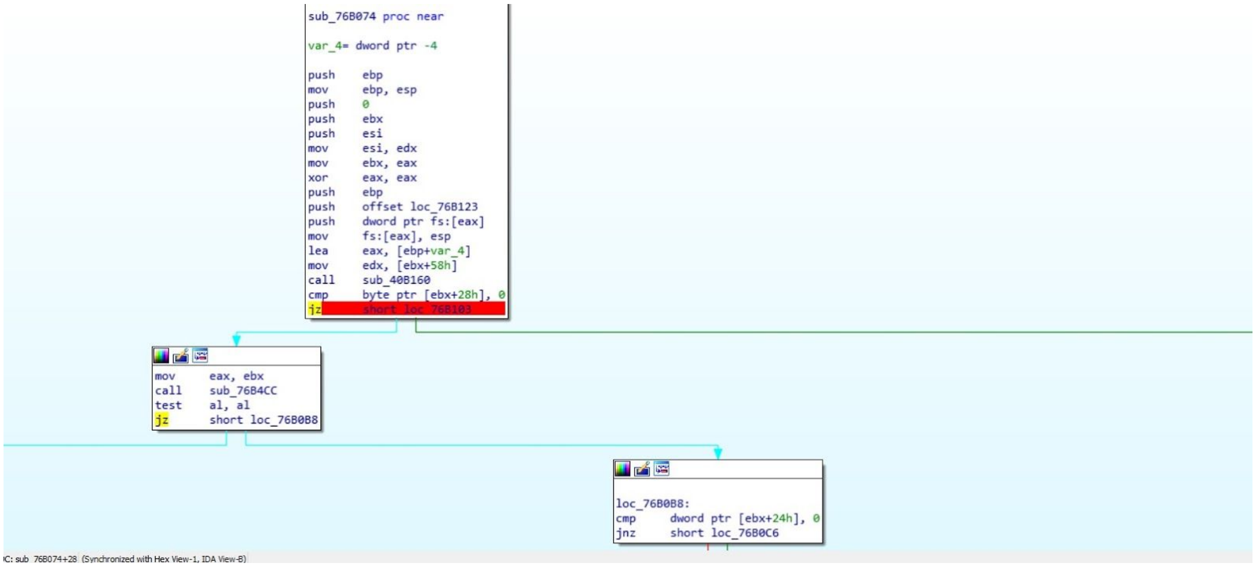
```
sbb     eax, eax
or      eax, 1
```

```
loc_40112B:
test    eax, eax
nop
daa
push    offset aHereSYourFlag ; "Here's your flag:"
call    sub_401020
push    offset a34gdfh340234 ; "34gdfh340234%\n"
call    sub_401020
add     esp, 8
xor     eax, eax
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    @__security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn
```

Part 2.

I was actually searching for the trial period is over string when i found the location of the banner modification code which contains all of the Unregistered copy and Expired copy modifications. So i changed the code to bypass all of them. Technically i did it by reversing the comparison so since its a unregistered copy it wont show as unregistered. Jnz to jz.

Address	Original bytes	Patched bytes
000000000076B074	75	74



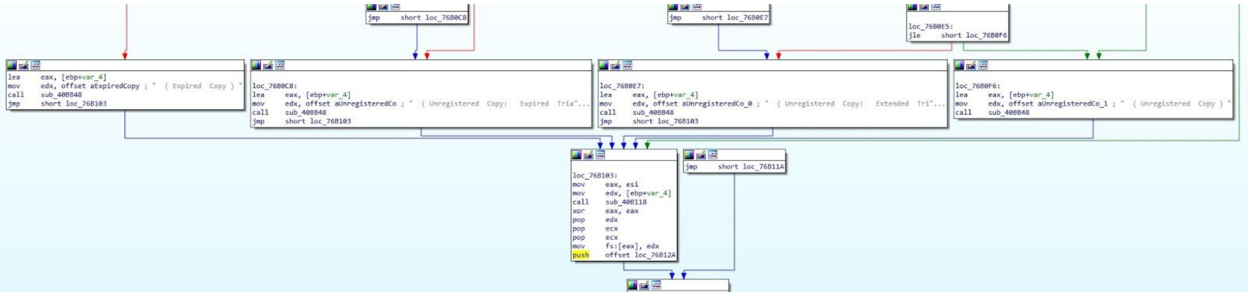
```
sub_76B074 proc near

var_4= dword ptr -4

push    ebp
mov     ebp, esp
push    0
push    ebx
push    esi
mov     esi, edx
mov     ebx, eax
xor     eax, eax
push    ebp
push    offset loc_76B123
push    dword ptr fs:[eax]
mov     fs:[eax], esp
lea     eax, [ebp+var_4]
mov     edx, [ebx+58h]
call    sub_40B160
cmp     byte ptr [ebx+28h], 0
jz      short loc_76B103
```

```
mov     eax, ebx
call    sub_76B4CC
test    al, al
jz      short loc_76B0B8
```

```
loc_76B0B8:
cmp     dword ptr [ebx+24h], 0
jnz     short loc_76B0C6
```



```
jmp     short loc_76B0C8
```

```
jmp     short loc_76B0E7
```

```
loc_76B0E5:
jle     short loc_76B0F6
```

```
lea     eax, [ebp+var_4]
mov     edx, offset aExpiredCopy ; "  ( Expired  Copy ) "
call    sub_40BB48
jmp     short loc_76B103
```

```
loc_76B0C8:
lea     eax, [ebp+var_4]
mov     edx, offset aUnregisteredCo ; "  ( Unregistered  Copy:   Expired  Tria"...
call    sub_40BB48
jmp     short loc_76B103
```

```
loc_76B0E7:
lea     eax, [ebp+var_4]
mov     edx, offset aUnregisteredCo_0 ; "  ( Unregistered  Copy:   Extended  Tri"...
call    sub_40BB48
jmp     short loc_76B103
```

```
loc_76B0F6:
lea     eax, [ebp+var_4]
mov     edx, offset aUnregisteredCo_1 ; "  ( Unregistered  Copy ) "
call    sub_40BB48
```

```
loc_76B103:
mov     eax, esi
mov     edx, [ebp+var_4]
call    sub_40B118
xor     eax, eax
pop     edx
pop     ecx
pop     ecx
mov     fs:[eax], edx
push    offset loc_76B12A
```

```
jmp     short loc_76B11A
```

Afterwards I tried to remove the call to the trial popup but it seemed to always come back despite my changes, and the requirements stated that it was either the banner or the the reminder pop-up.