CS165 Computer Security
Project 2, **Part 2**

Team 19: Steven Truong, Taghreed Alanazi.

Part 2: Real-world authentication bypass

In part 2, we removed the trial banner and the popup. We first searched for the keyword “Expired” on IDA to look for the assembly instructions that showed the trial banner. From there, we inspected the function that sets the trial banner in graph view. We modified the jump instructions so that execution of the application never reaches the instructions that set the trial banner.

Trial Banner:

Address	Original Instruction	New Instruction
.text:0076B0C4	jmp short loc_76B0C8	jmp short loc_76B103
.text:0076B0A7	jz short loc_76B0B8	jmp short loc_76B103
.text:0076B0E3	jmp short loc_76B0E7	jmp short loc_76B103
.text:0076B0E1	jbe short loc_76B0F6	jmp short loc_76B103
.text:0076B0E5	jle short loc_76B0F6	jmp short loc_76B103
.text:0076B0C6	jle short loc_76B0D7	jmp short loc_76B103

Next, to remove the popup we first searched for the string “registry” in IDA because the project description hinted at considering the registry. We inspected label sub_76A624. Then we set a breakpoint from the beginning of the function and ran the debugger. We were looking for the instruction that makes the popup appear.

```
.itext:00858C79                 cmp     dword ptr [eax+24h], 0
.itext:00858C7D                 jnz     short loc_858C87
.itext:00858C7F                 cmp     dword ptr [eax+20h], 2Ch ; ','
.itext:00858C83                 jbe     short loc_858CB9
.itext:00858C85                 jmp     short loc_858C89
.itext:00858C87 ; ---------------------------------------------------------------------------
.itext:00858C87
.itext:00858C87 loc_858C87:                             ; CODE XREF: .itext:00858C7D↑j
.itext:00858C87                 jle     short loc_858CB9
.itext:00858C89
.itext:00858C89 loc_858C89:                             ; CODE XREF: .itext:00858C85↑j
.itext:00858C89                 mov     eax, off_863454
.itext:00858C8E                 mov     eax, [eax]
.itext:00858C90                 call    sub_81637C
.itext:00858C95                 test    al, al
.itext:00858C97                 jnz     short loc_858CB9
.itext:00858C99                 xor     edx, edx
.itext:00858C9B                 mov     eax, offset aTrialExit ; "Trial Exit"
.itext:00858CA0                 call    sub_838EB8
.itext:00858CA5                 call    sub_84CF18
.itext:00858CAA                 mov     eax, off_8635E0
.itext:00858CAF                 mov     byte ptr [eax], 1
.itext:00858CB2                 mov     eax, [ebx]
.itext:00858CB4                 call    sub_63DE50
.itext:00858CB9
.itext:00858CB9 loc_858CB9:                             ; CODE XREF: .itext:00858C72↑j
.itext:00858CB9                                         ; .itext:00858C83↑j ...
.itext:00858CB9                 xor     edx, edx
.itext:00858CBB                 xor     eax, eax
.itext:00858CBD                 call    sub_838EB8
.itext:00858CC2                 mov     eax, off_8635E0
.itext:00858CC7                 cmp     byte ptr [eax], 0
```

After stepping through the program with the debugger, we discovered that call sub_81637C was causing the popup to appear. We also discovered that the instructions in loc_858CB9 initialized the program after the popup.

To remove the popup at the beginning, we had to prevent the program from jumping into loc_858C89 because it contained the call instruction that spawned a popup on startup. So we looked at the previous instructions and removed any instance of jump to loc_858C89 (function that has the popup) and replaced it with jump to loc_858CB9 (function that initializes the main program).

Function sub_828E08 also had a call instruction to sub_81637C, so we modified the instructions there too so that sub_81637C never gets called. This change prevents the popup from appearing periodically while using the program.

Popup:

Address	Original Instruction	New Instruction
.text:00858C85	jmp short loc_858C89	jmp short loc_858C8B9
.text:0082912C	test esi, esi	xor esi,esi
.text:0082912E	jnz loc_8291D9	jz loc_8291D9