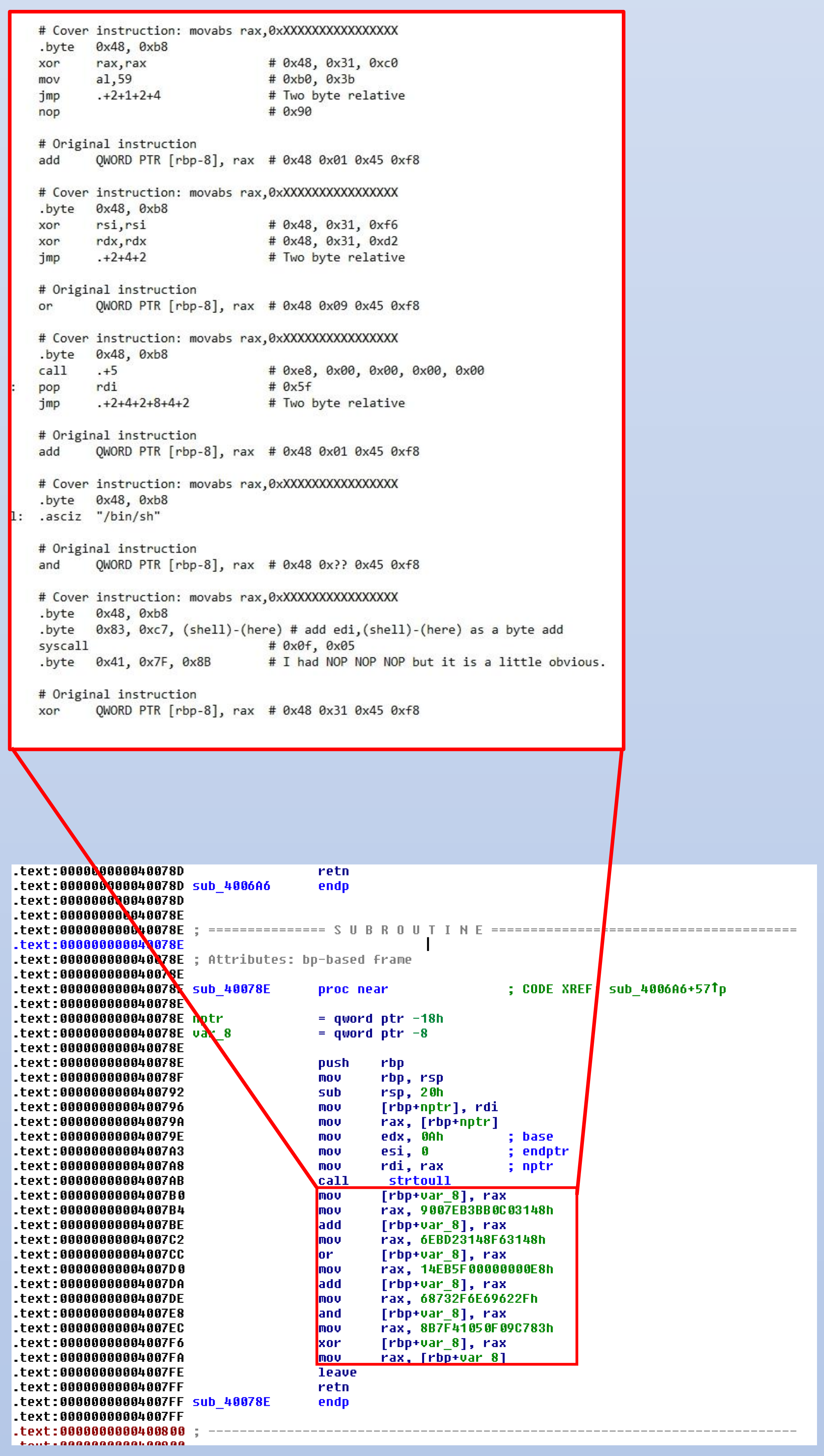
Executable Steganography

CIS 497 - Senior Project - Spring 2018



```
    # Cover instruction: movabs rax,0xXXXXXXXXXXXXXXXX
    .byte   0x48, 0xb8
    xor     rax,rax                 # 0x48, 0x31, 0xc0
    mov     al,59                   # 0xb0, 0x3b
    jmp     .+2+1+2+4               # Two byte relative
    nop                             # 0x90

    # Original instruction
    add     QWORD PTR [rbp-8], rax  # 0x48 0x01 0x45 0xf8

    # Cover instruction: movabs rax,0xXXXXXXXXXXXXXXXX
    .byte   0x48, 0xb8
    xor     rsi,rsi                 # 0x48, 0x31, 0xf6
    xor     rdx,rdx                 # 0x48, 0x31, 0xd2
    jmp     .+2+4+2                 # Two byte relative

    # Original instruction
    or      QWORD PTR [rbp-8], rax  # 0x48 0x09 0x45 0xf8

    # Cover instruction: movabs rax,0xXXXXXXXXXXXXXXXX
    .byte   0x48, 0xb8
    call    .+5                     # 0xe8, 0x00, 0x00, 0x00, 0x00
:   pop     rdi                     # 0x5f
    jmp     .+2+4+2+8+4+2           # Two byte relative

    # Original instruction
    add     QWORD PTR [rbp-8], rax  # 0x48 0x01 0x45 0xf8

    # Cover instruction: movabs rax,0xXXXXXXXXXXXXXXXX
    .byte   0x48, 0xb8
l:  .asciz  "/bin/sh"

    # Original instruction
    and     QWORD PTR [rbp-8], rax  # 0x48 0x?? 0x45 0xf8

    # Cover instruction: movabs rax,0xXXXXXXXXXXXXXXXX
    .byte   0x48, 0xb8
    .byte   0x83, 0xc7, (shell)-(here) # add edi,(shell)-(here) as a byte add
    syscall                         # 0x0f, 0x05
    .byte   0x41, 0x7F, 0x8B        # I had NOP NOP NOP but it is a little obvious.

    # Original instruction
    xor     QWORD PTR [rbp-8], rax  # 0x48 0x31 0x45 0xf8
```

```
.text:000000000040078D                 retn
.text:000000000040078D sub_4006A6      endp
.text:000000000040078D
.text:000000000040078E
.text:000000000040078E ; =============== S U B R O U T I N E =======================================
.text:000000000040078E
.text:000000000040078E ; Attributes: bp-based frame
.text:000000000040078E
.text:000000000040078E sub_40078E      proc near               ; CODE XREF: sub_4006A6+57↑p
.text:000000000040078E
.text:000000000040078E nptr            = qword ptr -18h
.text:000000000040078E var_8           = qword ptr -8
.text:000000000040078E
.text:000000000040078E                 push    rbp
.text:000000000040078F                 mov     rbp, rsp
.text:0000000000400792                 sub     rsp, 20h
.text:0000000000400796                 mov     [rbp+nptr], rdi
.text:000000000040079A                 mov     rax, [rbp+nptr]
.text:000000000040079E                 mov     edx, 0Ah        ; base
.text:00000000004007A3                 mov     esi, 0          ; endptr
.text:00000000004007A8                 mov     rdi, rax        ; nptr
.text:00000000004007AB                 call    _strtoull
.text:00000000004007B0                 mov     [rbp+var_8], rax
.text:00000000004007B4                 mov     rax, 9007EB3BB0C03148h
.text:00000000004007BE                 add     [rbp+var_8], rax
.text:00000000004007C2                 mov     rax, 6EBD23148F63148h
.text:00000000004007CC                 or      [rbp+var_8], rax
.text:00000000004007D0                 mov     rax, 14EB5F00000000E8h
.text:00000000004007DA                 add     [rbp+var_8], rax
.text:00000000004007DE                 mov     rax, 68732F6E69622Fh
.text:00000000004007E8                 and     [rbp+var_8], rax
.text:00000000004007EC                 mov     rax, 8B7F41050F09C783h
.text:00000000004007F6                 xor     [rbp+var_8], rax
.text:00000000004007FA                 mov     rax, [rbp+var_8]
.text:00000000004007FE                 leave
.text:00000000004007FF                 retn
.text:00000000004007FF sub_40078E      endp
.text:00000000004007FF
.text:0000000000400800 ; ---------------------------------------------------------------------------
```

Executable Steganography is an Honor's research thesis that seeks to enhance the field of code obfuscation. By taking advantage of long operand lengths afforded by the x86_64 architecture, assembly instructions can be hidden inside the operands of longer instructions. By including a sequence of jumps from the inside of one operand to the next, a second, hidden execution path can be formed. The objectives of this thesis are to perform an attack analysis of the hidden code and determine the technique's relative stealth, resilience, and potency as a means of defeating reverse engineering.

Research Advisor

Todd McDonald, Ph. D.
Professor of Computer Science
University of South Alabama

Project Researcher

Ryan Creel
Major: Computer Science

When he's not frantically attempting to finish his thesis on time, Ryan is interested in cybersecurity capture the flag competitions, reverse engineering, and attack defense competitions.
In his free time, he enjoys playing video games, taking things apart, and wondering why he has extra pieces when he puts them back together.