We won SEC-T CTF last week!

Gotta distribute the prizes!

Reversing / Assembly Crash Course

Mattias Grenfeldt

2023-09-21

What is reverse engineering?

- (Normal) (Forward) Engineering:
  - From ideas/concepts/designs to products/software/”artifacts”
- Reverse Engineering:
  - The other way around
  - Given output, what was input?
  - Mostly just reading code and thinking
  - A basis for most technical security work
- In CTFs:
  - Binary executable reversing
  - Prerequisite for binary exploitation (pwn)
  - Crackmes: What input will make this program say “Yes”?

Binary reverse engineering

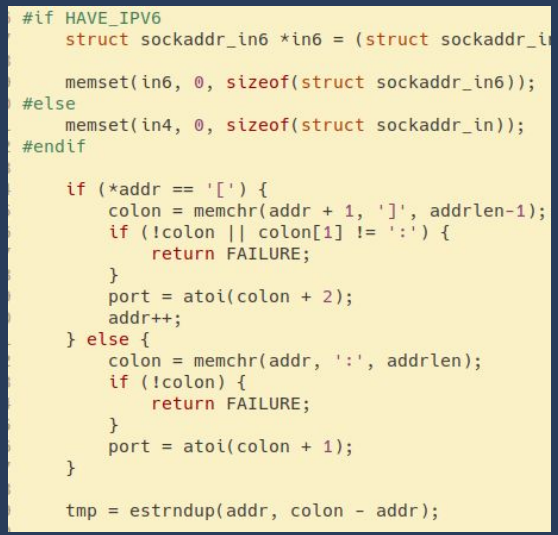
Compiler

Assembler

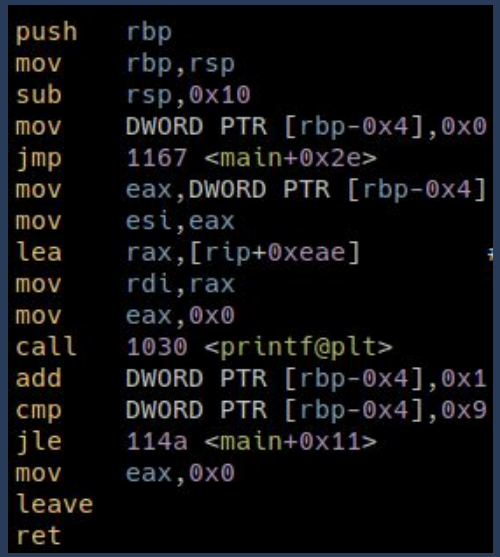
Source code (in C)

Assembly code

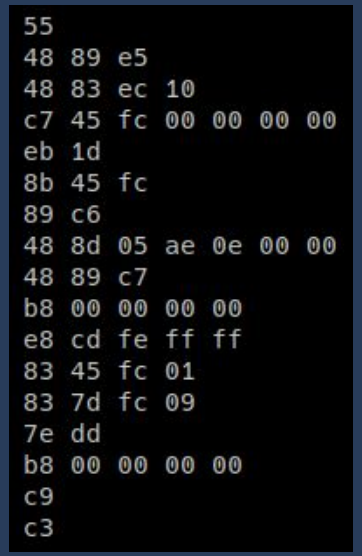
Machine code (data)



```
#if HAVE_IPV6
    struct sockaddr_in6 *in6 = (struct sockaddr_i

    memset(in6, 0, sizeof(struct sockaddr_in6));
#else
    memset(in4, 0, sizeof(struct sockaddr_in));
#endif

    if (*addr == '[') {
        colon = memchr(addr + 1, ']', addrlen-1);
        if (!colon || colon[1] != ':') {
            return FAILURE;
        }
        port = atoi(colon + 2);
        addr++;
    } else {
        colon = memchr(addr, ':', addrlen);
        if (!colon) {
            return FAILURE;
        }
        port = atoi(colon + 1);
    }

    tmp = estrndup(addr, colon - addr);
```



```
push   rbp
mov    rbp,rsp
sub    rsp,0x10
mov    DWORD PTR [rbp-0x4],0x0
jmp    1167 <main+0x2e>
mov    eax,DWORD PTR [rbp-0x4]
mov    esi,eax
lea    rax,[rip+0xeae]
mov    rdi,rax
mov    eax,0x0
call   1030 <printf@plt>
add    DWORD PTR [rbp-0x4],0x1
cmp    DWORD PTR [rbp-0x4],0x9
jle    114a <main+0x11>
mov    eax,0x0
leave
ret
```



```
55
48 89 e5
48 83 ec 10
c7 45 fc 00 00 00 00
eb 1d
8b 45 fc
89 c6
48 8d 05 ae 0e 00 00
48 89 c7
b8 00 00 00 00
e8 cd fe ff ff
83 45 fc 01
83 7d fc 09
7e dd
b8 00 00 00 00
c9
c3
```

Decompiler (not perfect)

Disassembler

What is assembly?

- What your CPU actually runs
- Many different: x86, ARM, MIPS, Power PC, SPARC, RISC-V, etc…
- We focus on x86-64
- Just a programming language
- Each instruction is very primitive
- Registers: like variables
- Memory: a bit new and spooky
- Untyped language: there are only bytes 🙏🗿🙏
  - ✨ Things are whatever you interpret them to be ✨
- Control flow (in function): “only gotos”, compares and jumps

Static vs. Dynamic reversing

- Two approaches
- Dynamic reversing
  - Running the program and investigating it
  - gdb, strace, ltrace
- Static reversing
  - Just looking at the binary and thinking
  - Disassembly, decompilation

Time for problem solving

- Guide:
  - https://github.com/RoyalRoppers/get-started/blob/main/reversing/README.md
- Problems:
  - https://royalroppers.team/problems/reversingproblems.zip

Upcoming

Next CTF

- ???

Next Meetup

- ???
- What do you want to learn?