(more) Advanced Binary Exploitation

ComSec 25/11/2020

@sleepunderflow

# ~# whoami

- Graduate from Coventry University
    - Ethical Hacking and Cybersecurity
- Security Consultant at Nettitude
- @sleepunderflow at social media

Agenda

- Tools
- ret2libc
- ROP
- ROP with PWN tools
- One-address overflow
- Tips & tricks

Tools

- GDB + GEF
- pwntools (python2)
- ropper
- checksec
- Patience
- gcc-multilib

Scenario 1

- NX stack – no shellcode
- No PIE (or use information leak)
- Uses libc
  - printf
  - read
  - fgets
  - open
- Filesystem access
  - Can get libc binary

Solution – ret2libc

- Find a way to leak an address in libc at known offset
  - Good candidates are puts, write, etc.
  - PLT
- Find the offset of leaked address in libc and calculate the base libc is at
- Calculate addresses of
  - system (or execl)
  - /bin/sh
- Call system(“/bin/sh”)

Scenario 2

- NX stack – no shellcode
- No PIE (or use information leak)
- No libc

Solution – proper ROP

- Make our own execve syscall
- Use parts of the binary as building blocks for the syscall
- Chain those blocks together

Scenario 3

- All the conditions for ret2libc are met
- But
  - 64-bit
  - Strcpy used to overflow
    - NULL BYTES IN ADDRESSES!
    - Limited to a single address overflown

Solution – use the magic of leave

- Rather than placing our ROP chain on the stack, move the stack to our ROP
- Overflow saved base pointer instead of return pointer

The End

Any Questions?

@sleepunderflow