Lec04: Bypass Stack Protections

Chengyu Song

Slides based on CS6265 taught by Prof. Taesoo Kim
Scoreboard
Administrative

• Due: Lab04 is out and its due on Feb 16 at midnight
• 7 normal challenges + 3 bonus challenges
Best Write-ups for Lab03

• Hang Zhang [hzhan033@ucr.edu]
Discussion 0

• What's the most "annoying" bug or challenge?
• What's the most "interesting" bug or challenge?
• What did you learn in general?
Discussion 1

• How to set env variable with non-printable characters?
Discussion 2

• How to you find the address of your shellcode?
Discussion 3

- Where do you put your shellcode?
Discussion 4

• argc0
  • When argv is empty, envp replaces its position
  • What is the minimal env variables needed?
Discussion 5

• jmp-to-where
  • Special character(s) in your shellcode?
  • Remember there are two names!!
Recall: How to Prevent Stack Overflow?

- Canary

```
|<-- 0x14 ------------------|+---- ebp
  top
[ [ [canary][fp][ra][....]
|<---- 0x30 ------------------|
      XOXOXO XXXX
      (corrupted?)
```
Rationales

• Based on existing attacks

  1. Sequential overwritten
  2. return address and frame pointer is important
Subtle Design Choices

- Where to put the canary? (e.g., right above ra? fp? local vars?)
- Which value should I use as a canary? (e.g., secrete? random? per exec? per func?)
- How to compare the canary value? (e.g., xor? cmp?)
- What to do after you find the canary value is corrupted? (e.g., crash? report?)
In-class Tutorial

• Step 1: Understanding GCC's Stack Protector
• Step 2: Let's exploit 0xdeadbeef canary!
Lab 04

+ [level 01] xor [20 points]
+ [level 02] weak-random [20 points]
+ [level 03] terminator [20 points]
+ [level 04] gs-random [20 points]
+ [level 05] stackshield [20 points]
+ [level 06] assassination [20 points]
+ [level 07] mini-heartbleed [20 points]
+ [level 08] pltgot [20 points]
+ [level 09] ssp [20 points]
+ [level 10] fd [20 points]
Additional Design Challenges

- Multi-thread program: time-of-check-to-time-of-use attack
- Fault-tolerant program: hacking blind
- Mismatches between CALL and RET
Alternative Approaches

- **The Performance Cost of Shadow Stacks and Stack Canaries**
  - Shadow Stack: switch, parallel
  - Safe Stack
  - CFI
  - Hardware Assisted
References

- Bypassing StackShield