CSE 127 Week 3 Discussion

Kevin Yu (Winter 2021)
Legal Notice

This lecture will be recorded and made available to registered students on Canvas.
Agenda

1) PA2 GDB Demo
2) PA3 Overview
3) Questions/OH
Using GDB

- Read memory at a given address as an instruction (i.e., print the instruction located at a specific address)

  \[ x/i \ <\text{address}> \]

  \[ x/x \ <\text{address}> \] (reads \*<address> as hex digits)

- Printing address of a buffer

  \[ p \ \&\text{buf} \]
Side Channel Attacks

- The goal of this assignment is to gain hands-on experience with exploiting memory and timing side channels.
- Side channel attacks, in general, are attacks carried out using unintended information leak (side channels) caused by the implementation of the target itself.
- Common side channels include: memory (cache) attacks, timing attacks...etc...
PA3 Overview

Three Parts:

1) Memhack (50 pts) *memhack.c*
   a) Generate a signal on correct guess
2) Timehack (50 pts) *timehack.c*
   a) Using timing as signal for correct guess
3) Combined Writeup
Structure of Assignment

- You are given a VM image with the starter files `memhack.c` and `timemhack.c`.
- Both files import a library `sysapp.c` which contains the side channel vulnerability you are trying to exploit. You DO NOT NEED TO MODIFY this file.
sysapp.c

- check_pass() takes a char pointer and matches the password character by character. If all characters match the correct password, return 1, else 0.
  - An artificial delay() is before checking each character, meaning the more characters that are correct, the longer the check takes to succeed/fail
- hack_system() is the function you will need to call once you have correctly deduced the password.
memhack.c

- You are given a buffer set up (more details next slide) that will cause a segfault if the program tries to access certain specific bytes
- You are also given demonstrate_signals() which should show you how to catch segfaults in your program.
Memhack Buffer

Protected bytes

page 1

page 2

page 3

buffer

page_start
The password you are trying to crack

- At most 32 characters
- ASCII symbols from ASCII 33 ("!") to ASCII 126 ("~") can be used in the password.
Memhack hints

- Referencing protected memory bits will raise a fault
- Capturing a fault can be used as a hint of a (partially) correct guess
- ex: If the password is “hello” and you place

```
page 1 h page 2 page 3
```

- `check_pass(my_guess)` faults. Why?

```
page 1 a page 2 page 3
```

- `check_pass(my_guess)` does not fault and returns 0. Why?
Catching faults

- Use sigsetjmp/siglongjmp
- sigsetjmp
  - Set jump point for siglongjmp to jump to later
  - Returns 0 when you call it to set the returning point
  - Returns non-zero value when it returns via a call to siglongjmp()
- siglongjmp
  - Used to return to the point at which sigsetjmp was called
- Avoid calling sigsetjmp in a helper function, because if the function sigsetjmp was called returns before siglongjmp is called, causes undefined behavior.
Catching faults

```c
signal(SIGSEGV, SIG_DFL);

signal(SIGSEGV, &handle_SEGV);
```

- This tells the system that whenever it hits a SIGSEGV segmentation fault, to call the function `handle_SEGV()`.

- There are two calls because the documentation requires the signal to be set to default (SIG_DFL) before being set to a handler function.
Execution time of check_pass depends on how many characters you guess correctly.

- rdtsc returns processor cycle count in long. Treat as a running timer
  - Use as timer by calling before and after and finding the difference in cycles -> Cycles that have passed
- There may be a lot of noise with each check_pass call, so you want to do multiple samples
Dealing with noise

- Avoid using printf’s in your code, as they can cause huge variances in execution time
- Use the median and not the mean for the multiple trials you run for a given guess
  - Outliers are every extreme, so you want to avoid using the mean
  - qsort may be helpful
- If time is not continuing to increase as you progress through characters, then you probably made a bad guess earlier. Backtrack
Questions / Office Hours

good luck :D