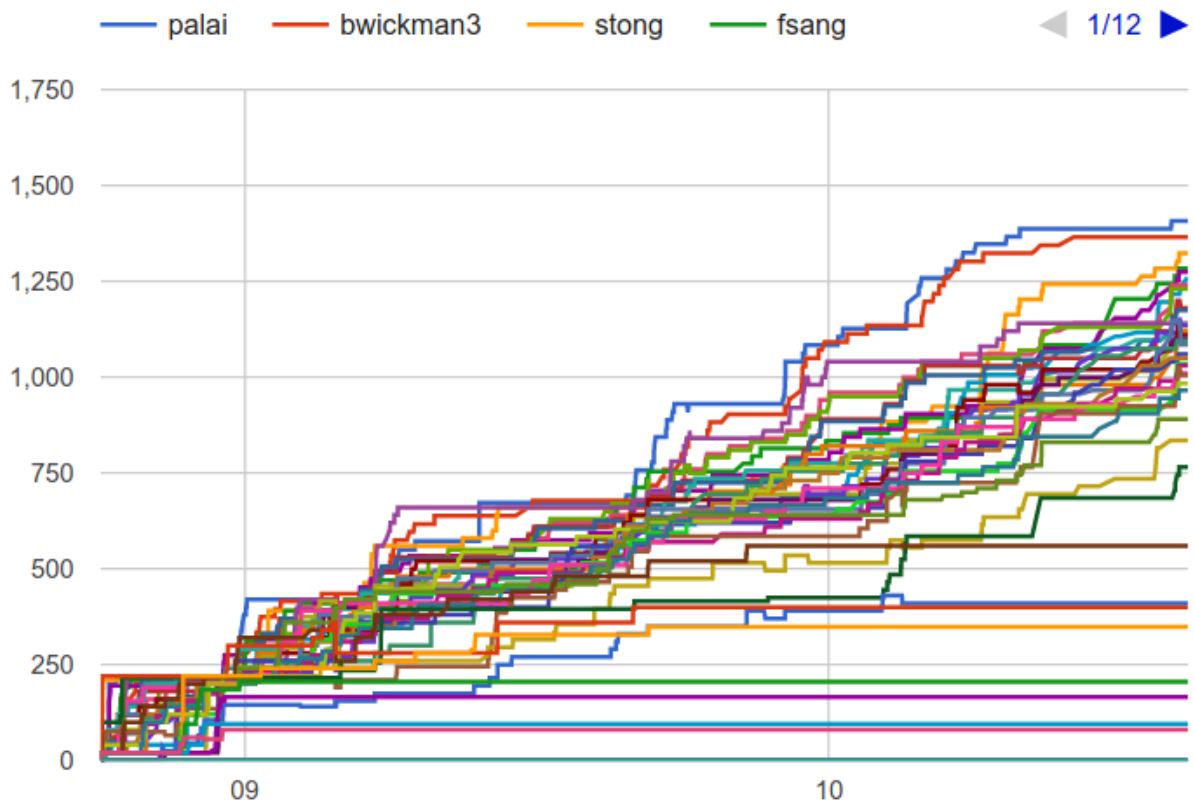
Lec08: Remote Exploit

Taesoo Kim

Scoreboard


palai
bwickman3
stong
fsang
1/12
1,750
1,500
1,250
1,000
750
500
250
0
09
10

Administrivia

- In-class CTF on Nov 16-17 (24 hours)!
- Due: form your team and submit your CTF challenge by Nov 13!
- Due: Lab07 is out and its due on Nov 2 (two weeks!)
- NSA Codebreaker Challenge → Due: Nov 29

Best Write-ups for Lab05

libbase	**gkamuzora3, stong**
moving-target	**nhicks6, sfriedfertig**
fmtstr-digging	**riya, burak**
fmtstr-read	
fmtstr-write	
brainfxxk	
fd-const	**stong, palai**
fmtstr-heap	**seulbae, riya**
profile	**palai, burak**
mini-sudo	**palai, stong**

Discussion: Lab05

- What's the most "annoying" bug or challenge?
- What's the most "interesting" bug or challenge?
- So, DEP and ASLR are not so effective?

Discussion: libbase

- What do you learn from ./check?

```
$ ./check
stack   : 0xff930aa0
system(): 0xf7521c50
printf(): 0xf7536670

$ ./check
stack   : 0xff930250
system(): 0xf755dc50
printf(): 0xf7572670
```

Discussion: libbase

Discussion: moving-target

- What's "check-aslr.sh" and pie.c?
- How many times should we try to exploit?

Discussion: moving-target

Discussion: fmtstr-*?

- fmtstr-read/write/digging are relatviely easy

How to Prevent fmtstr-*?

How to Prevent fmtstr-*?

1. Non-POSIX compliant (e.g., Windows)
    - Discarding %n
    - Limiting width (e.g., “%.512x” in XP, “%.622496x” in 2000)
2. Dynamic: enabling FORTIFY in gcc (e.g., Ubuntu)
3. Static: code annotation (e.g., Linux)

FORTIFY (-D_FORTIFY_SOURCE=2)

- Ensuring that all positional arguments are used
  - e.g., %2$d is not ok without %1$d
- Ensuring that fmtstr is in the read-only region (when %n)
  - e.g., “%n” should not be in a writable region

```
$ ./fortify-yes %2$d
*** invalid %N$ use detected ***

$ ./fortify-yes %n
*** %n in writable segment detected ***
```

Discussion: brainfxxk

Discussion: brainfxxk

Discussion: fd-const

- What's the bug?
- How to exploit?

Discussion: profile

- What's program about?
- What's the bug?

Discussion: profile

Discussion: profile

Discussion: profile

Discussion: mini-sudo (CVE-2012-0809)

- What is '-D9' for?

Discussion: mini-sudo (CVE-2012-0809)

```
void sudo_debug(int level, const char *fmt, ...) {
  va_list ap;
  char *fmt2;

  if (level > debug_level) return;

  /* Backet fmt with program name and a newline to make it
     a single write */
  easprintf(&fmt2, "%s: %s\n", getprogname(), fmt);
  va_start(ap, fmt);
  vfprintf(stderr, fmt2, ap);
  va_end(ap);
  efree(fmt2);
}
```

CVE-2013-1848: Linux ext3

```
void ext3_msg(struct super_block *sb, const char *prefix,
        const char *fmt, ...)
{
  struct va_format vaf;
  va_list args;

  va_start(args, fmt);

  vaf.fmt = fmt;
  vaf.va = &args;

  printk("%sEXT3-fs (%s): %pV\n", prefix, sb&#8594;s_id, &vaf);

  va_end(args);
}
```

CVE-2013-1848: Linux ext3

```
// @get_sb_block()
ext3_msg(sb, "error: invalid sb specification: %s", *data);

// @ext3_blkdev_get()
ext3_msg(sb, "error: failed to open journal device %s: %ld",
        __bdevname(dev, b), PTR_ERR(bdev));
```

Take-outs from DEP/ASLR?

- Do you think DEP/ASLR make attackers' life more difficult?
- Is still possible to exploit? why?
- Although we can't place shellcode into stack/heap, we can still hijack the control flow of a program in many interesting ways

Discussion: Modern Exploit on ASLR (PIE)

- Leak (or infer) code pointers (so map into library or code)
- Construct ROP (today's topic)
- (although there are a few proposals, such as CFI, to mitigate ROPs)

Today's Tutorial

- About the in-class CTF challenge
- In-class tutorial:
  - Socket programming in Python
  - Your first remote exploit!

About: In-class CTF

- In-class CTF on Nov 16-17 (24 hours), starting in the class!
- 3-4 persons as a team
- Award prizes!
- Submit your CTF challenge by Nov 13!

About: Docker Template/Sample

```
$ ssh lab07@computron.gtisc.gatech.edu -p 9007
$ ssh lab07@cyclonus.gtisc.gatech.edu -p 9007
Password: lab07

$ cd tut-remote
$ cat README
```

Remote Challenges

- Use techniques learned from Lab01-Lab07
- But targeting the remote server (e.g., online services)!

Lab07: Remote Challenges

DEMO: about how remote challenges work

- nc
- exploit.py

In-class Tutorial

- Step1: nc
- Step2: brute force attack
- Step3: guessing attack

```
$ ssh lab07@computron.gtisc.gatech.edu -p 9007
$ ssh lab07@cyclonus.gtisc.gatech.edu -p 9007
Password: lab07

$ cd tut-remote
$ cat README
```