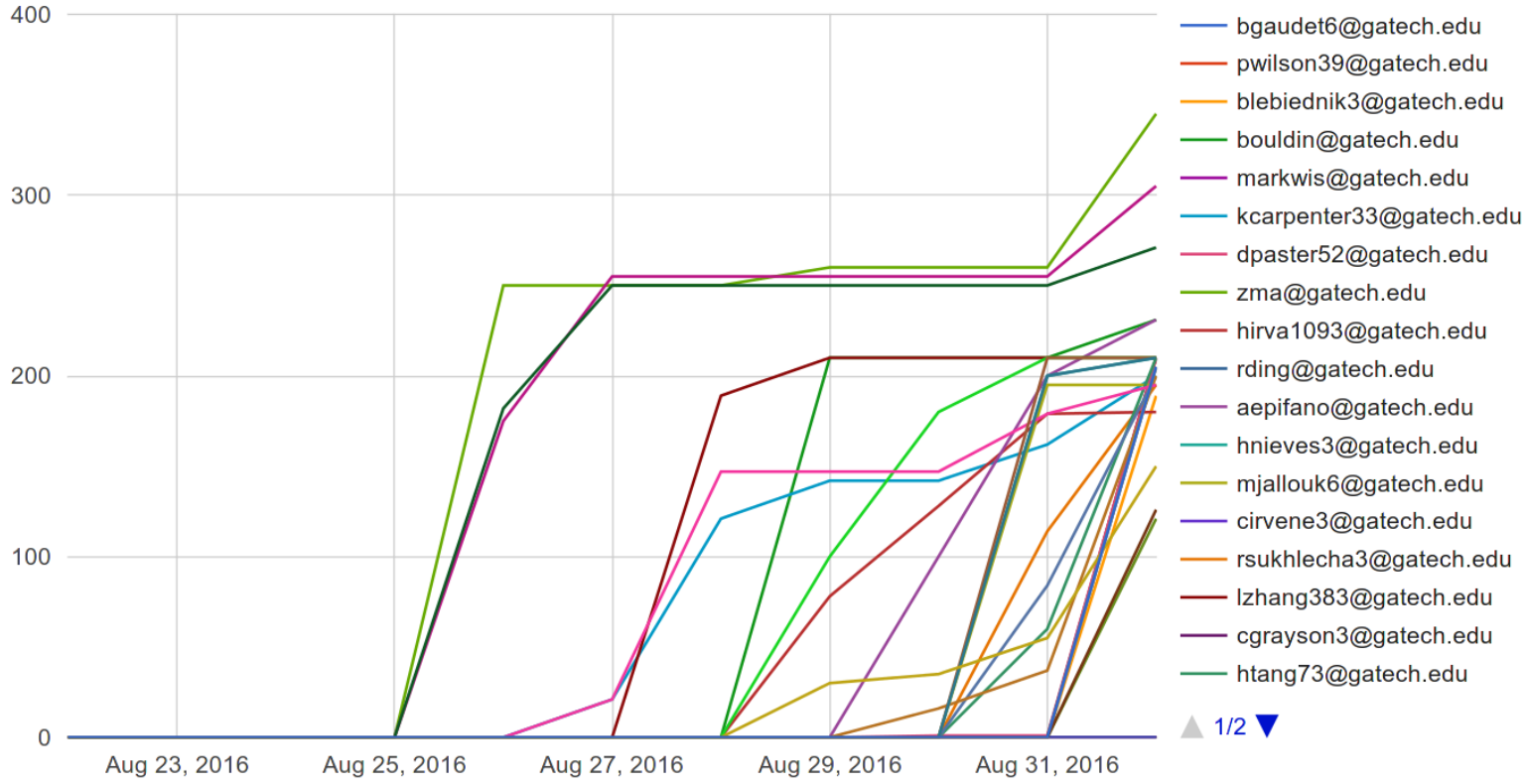
Lec02: x86_64 / Shellcode / Tools

Taesoo Kim

Scoreboard


400
300
200
100
0
Aug 23, 2016
Aug 25, 2016
Aug 27, 2016
Aug 29, 2016
Aug 31, 2016
bgaudet6@gatech.edu
pwilson39@gatech.edu
blebiednik3@gatech.edu
bouldin@gatech.edu
markwis@gatech.edu
kcarpenter33@gatech.edu
dpaster52@gatech.edu
zma@gatech.edu
hirva1093@gatech.edu
rding@gatech.edu
aepifano@gatech.edu
hnieves3@gatech.edu
mjallouk6@gatech.edu
cirvene3@gatech.edu
rsukhlecha3@gatech.edu
lzhang383@gatech.edu
cgrayson3@gatech.edu
htang73@gatech.edu
1/2

Administrivia

- Survey: how many hours did you spend? (<3h, 6h, 10h, >20h)
- Join Piazza
- An optional recitation at 5-6pm on every Wed (in Klaus 1447)
- Lab02 is out!
- Due: Sept 8th at midnight

Course Grading (No Grouping!)

- 100% Lab (if you didn't turn in a single lab, you will get F)
- No midterm and final exams
- 40%: A , 30-40%: B , 30-20%: C and below
- If you solve 8 on average in all labs, you will get A !
- See Game Rules

Best Write-ups for Lab01

- zma@gatech.edu
- yskalli@gatech.edu

Bomb Stats

- Bombs exploded ?? times in total?
- in ?? phases?
- ?? people exploded at least once?

Bomb Stats

- Bombs exploded 70 times in total (70 x -5 = -350 pts)
- in ALL phases!
- 25 people exploded at least once! (so how many alive?)
  - Each lab: 11/04/04/10/01/04/05/05/04/07 people
  - Each lab: 15/05/05/14/01/04/06/07/04/09 times

Discussion 0

1. How was your experience on handling bomb?

Discussion 1

1. How did you prevent bombs from explosion?

DEMO

- GDB: breakpoint (command, set $eax=0, return, continue)
- Binary patching (cc/ret)
- Hijacking curl invocation (strace/ltrace)
- Command injection

Discussion 2

1. What was the most difficult/annoying phase?

Discussion 3

1. How did you find 'secret_phrase'?

Discussion 4

1. Any tricky assembly?

Discussion 5

1. Any useful trick to share with other students?

ASM showcases 1

```
 1ac1: e8 ba ef ff ff          call   a80 <__x86.get_pc_thunk.bx>
 1ac6: 81 c3 3a 35 00 00       add    ebx,0x353a
 ...

00000a80 <__x86.get_pc_thunk.bx>:
  a80: 8b 1c 24                mov    ebx,DWORD PTR [esp]
  a83: c3                      ret
```

ASM showcases 2

```
1525: 65 a1 14 00 00 00        mov    eax,gs:0x14
152b: 89 84 24 24 04 00 00     mov    DWORD PTR [esp+0x424],eax
1532: 31 c0                    xor    eax,eax
...
15cb: 8b 84 24 1c 04 00 00     mov    eax,DWORD PTR [esp+0x41c]
15d2: 65 33 05 14 00 00 00     xor    eax,DWORD PTR gs:0x14
15d9: 75 1c                    jne    15f7 <print_key+0xfa>
...
 5f7: e8 14 0c 00 00           call   2210 <__stack_chk_fail_local>
```

ASM showcases 3

```
2144: 8d a4 24 d4 e7 ff ff    lea    esp,[esp-0x182c]
214b: 83 0c 24 00             or     DWORD PTR [esp],0x0
214f: 8d a4 24 0c 10 00 00    lea    esp,[esp+0x100c]
```

Lab02: Bomb Lab2 / Shellcode

- Another Bomblab (be extra careful this time)!
- Writing five different shellcodes
  - x86, x86_64, both!, ascii, minimal size (competition)
  - Bonus : the smallest shellcode gets extra **20 pts**!

Today's Tutorial

- x86 shellcode overview
- In-class tutorial
  - PEDA (a fancy gdb plugin)
  - Walk over x86 shellcode (+ excercise!) and various tools

DEMO: PEDA commands

- (python: gdb.execute())
- context
- telescope/xinfo
- checksec/aslr
- vmmap/find
- elfheader, elfsymbol, hexdump
- pdisass, nearpc
- deacvtive
- rop: asmsearch, dumprop, ropgadget …

shellcode (in C)

```c
#include <stdio.h>
#include <unistd.h>

int main() {
  char *sh = "/bin/sh";
  char *argv[] = {sh, NULL};
  char *envp[] = {NULL};
  execve(sh, argv, envp);
  return 0;
}
```

DEMO: shellcode.S

- explain: asm, structure
- `man syscall` (about convention)
- execve()
- tutorial: /bin/sh to /bin/cat /proc/flag

In-class Tutorial

- Step 1: Install PEDA
- Step 2: Play with shellcode!

```
$ git git@clone tc.gtisc.gatech.edu:seclab-pub cs6265
or
$ git pull
$ cd cs6265/lab02
$ ./init.sh

$ cd tut
$ cat README
```

References

- Assembly
- x86
- x86_64
- PEDA