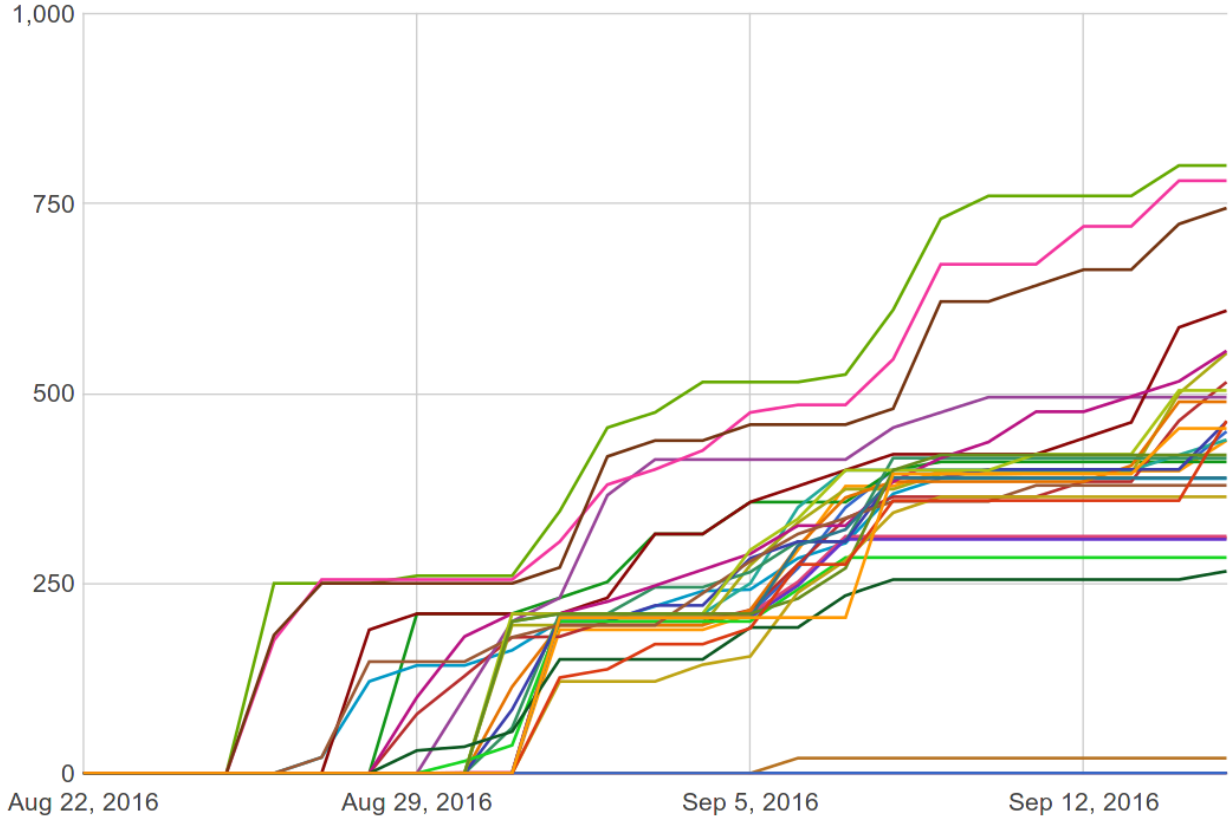
Lec04: Writing Exploits

Taesoo Kim

Scoreboard



Administrivia

- Join Piazza!
- An optional recitation on every Wed
  - 5:00-6:00pm (in Klaus 1447)
  - 6:00-6:30pm (in Klaus 3126)
- Due: Lab03 (stack overflow) on Sept 22 at midnight
- NSA Codebreaker Challenge → New due: Oct 13

Course Grading (Expectation for A/B)

1. Game:
   - 40% → A
   - 30-40% → B
2. Self competition as well:
   - 8 on average → A
   - 6 on average → B
3. Currently, ~10 (Lab1), ~9.5 (Lab2), so all A!
4. Please don't give up! Here we are to help you succeed!

Survival Guide for CS6260

1. Work as a group/team (find the best one around you!)
    - NOT each member tackles different problems
    - All members tackle the same problem (and discuss)
2. Ask questions wisely
    - Explain your assumption first
    - Explain your problem second
3. Take advantage of four TAs standing next you to help!
    - World-class (literally) hackers give a private tutoring for you!
    - But, remember! only when you ask ..

NSA Codebreaker Challenges

University	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Carnegie Mellon University	18	15	7	4	4	2
Georgia Institute of Technology	41	31	29	20	3	1
Naval Postgraduate School	4	4	4	4	3	0
Dakota State University	48	34	19	15	1	0
University of Maryland, Baltimore County	23	17	8	7	1	0
Purdue University	7	6	5	4	1	0
University of Tulsa	5	5	3	2	1	0
University of Maryland, College Park	4	4	3	2	1	0
University of Maryland - University College	3	2	2	1	1	0
Palm Beach State College	1	1	1	1	1	0

Showing 1 to 10 of 215 entries

Previous 1 2 3 4 5 ... 22 Next

NSA Codebreaker Challenges Tasks

- Task 1: Compute a hash and identify IED network ports
- Task 2: Refine IED network traffic signature
- Task 3: Decrypt IED key file
- Task 4: Disarm an IED with the key
- Task 5: Disarm any IED without a key
- Task 6: Permanently disable any IED

Lab04: Stack overflow!

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

by Aleph One
aleph1@underground.org

`smash the stack` [C programming] n. On many C implementations
it is possible to corrupt the execution stack by writing past
the end of an array declared auto in a routine. Code that does
this is said to smash the stack, and can cause return from the
routine to jump to a random address. This can produce some of
the most insidious data-dependent bugs known to mankind.
Variants include trash the stack, scribble the stack, mangle
the stack; the term mung the stack is not used, as this is
never done intentionally. See spam; see also alias bug,
fandango on core, memory leak, precedence lossage, overrun screw.

Lab04: Stack overflow!

- It's time to write real exploits (i.e., control hijacking)
- TONS of interesting challenges!
    - e.g., lack-of-four, frobnicated, upside-down ..

Today's Tutorial

- Example: exploit crackme0x00 to get a flag!
- Explore a template exploit code
- In-class tutorial
  - IDA (how many people are using?)
  - Extending the exploit template

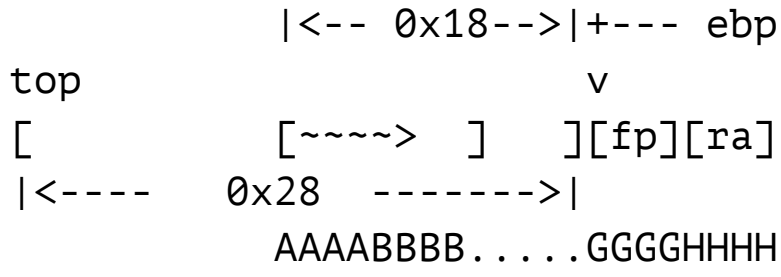
Reminder: crackme0x00

```
$ objdump -d crackme0x00
...
8048414:       55                          push   %ebp
8048415:       89 e5                       mov    %esp,%ebp
8048417:       83 ec 28                    sub    $0x28,%esp

                          +--- ebp
top                       v
[                       ][fp][ra]
|<---   0x28  ------->|
```

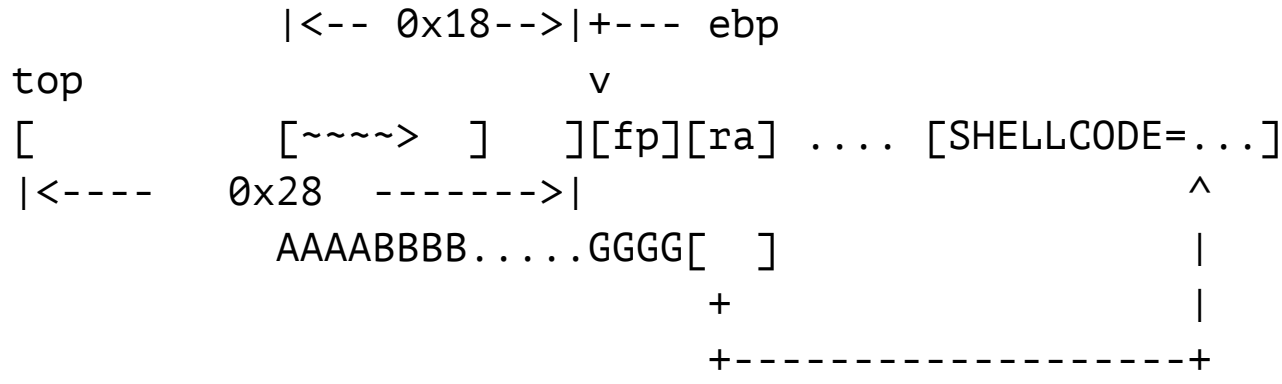
Reminder: crackme0x00

```
$ objdump -d crackme0x00
...
8048448:       8d 45 e8                lea    -0x18(%ebp),%eax
804844b:       89 44 24 04             mov    %eax,0x4(%esp)
804844f:       c7 04 24 8c 85 04 08    movl   $0x804858c,(%esp)
8048456:       e8 d5 fe ff ff          call   8048330 <scanf@plt>

           |<-- 0x18-->|+--- ebp
top                    v
[          [~~~~>  ]   ][fp][ra]
|<----   0x28  ------->|
```

Reminder: crackme0x00



Example: Injecting Shellcode



1) How to decide the address of an environment variable? (changing!)
2) How to inject (or manipulate) environment variables?

DEMO: Exploiting crackme0x00!

- core dump
  - ulimit -c unlimited
  - gdb -c core
- shell commands/tools
  - env
  - export
  - hexedit
  - dmesg

In-class Tutorial

- Step 1: Bruteforcing
- Step 2: Play with your first exploit!

```
$ git git@clone tc.gtisc.gatech.edu:seclab-pub cs6265
or
$ git pull
$ cd cs6265/lab04
$ ./init.sh

$ cd tut
$ cat README
```

References

- IDA Demo
- Phrack #49-14