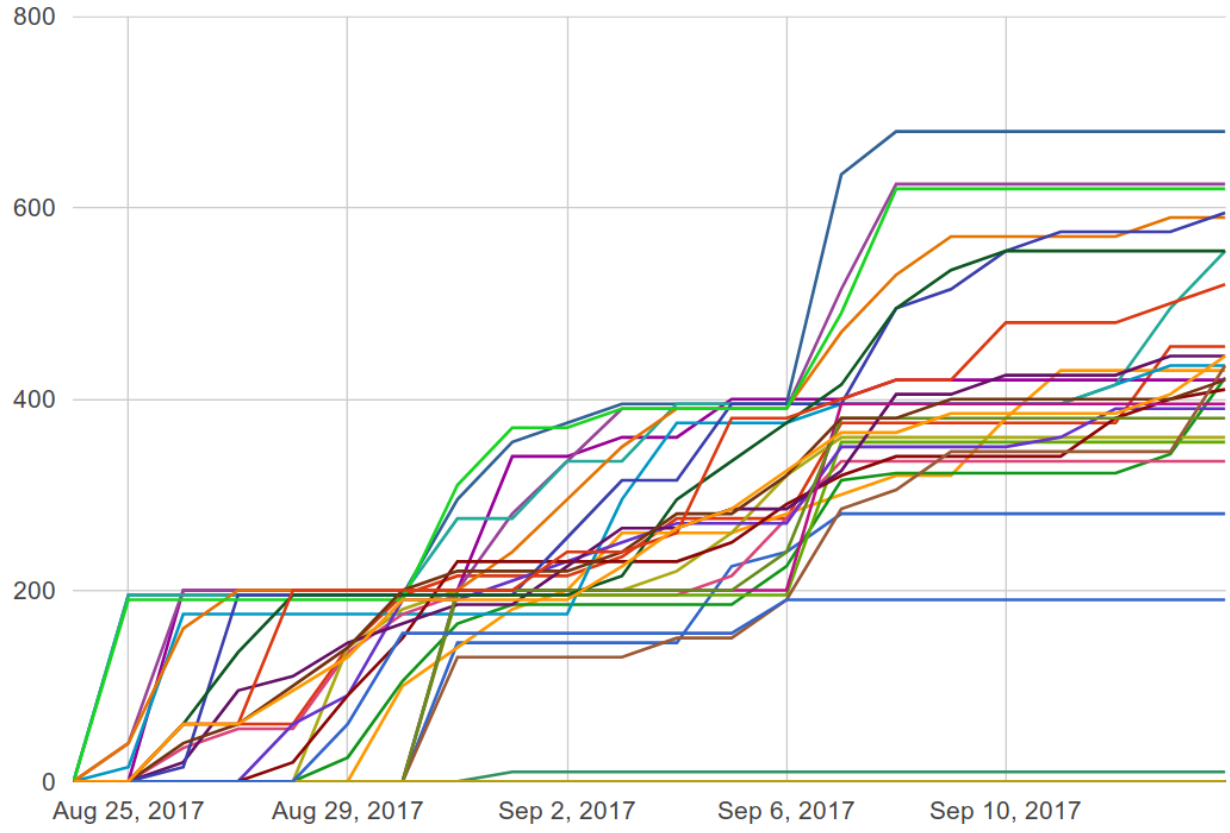
Lec04: Writing Exploits

Taesoo Kim

Scoreboard


800
600
400
200
0
Aug 25, 2017
Aug 29, 2017
Sep 2, 2017
Sep 6, 2017
Sep 10, 2017

Administrivia

- Join Piazza!
- An optional recitation at 4:30-5:30pm on Wed (in CBC 104A)
- Due: Lab03 (stack overflow) on Sept 21 at midnight
- NSA Codebreaker Challenge → Due: Nov 30 (starts tonight!)

Survival Guide for CS6265

1. Work as a group/team (find the best ones 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 hackers give a private tutoring for you!
    - But, remember! only when you ask ..

NSA Codebreaker Challenges

NSA Codebreaker Challenges (Last Year)

University	Students Participating	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Georgia Institute of Technology	149	67	57	49	39	18	5
Carnegie Mellon University	74	28	26	16	11	6	3
United States Military Academy	28	11	9	9	7	5	3
Naval Postgraduate School	15	7	7	6	6	5	1
University of Maryland, College Park	24	10	7	4	3	2	1
Lesley University	1	1	1	1	1	1	1
Williams College	1	1	1	1	1	1	1
Dakota State University	100	56	40	26	20	8	0
New Mexico Institute of Mining & Technology	27	14	14	13	12	7	0
Arizona State University	44	24	23	13	9	6	0

Showing 1 to 10 of 481 entries

Previous 1 2 3 4 5 ... 49 Next

NSA Codebreaker Challenges

The Department of Homeland Security (DHS) has requested NSA's assistance in investigating unusual network activity within a large SCADA system. The system controls critical infrastructure for multiple cities, so it's imperative that an assessment is carried out immediately. If any intrusions are found, then we need to identify how the systems were compromised and neutralize the threat. DHS is concerned that someone might be attempting to take control of the distributed sensor nodes and form a large botnet. If this happens, they could use it to wreak havoc across the cities and potentially launch DDoS attacks against other critical networks.

NSA Codebreaker Challenges Tasks

- Task 0: Setup a test instance of the system
- Task 1: Analyze suspicious network traffic
- Task 2: Develop a network signature for an intrusion detection system
- Task 3/4: Analyze critical system components for vulnerabilities
- Task 5: Perform forensic analysis of a compromised endpoint
- Task 6: Craft an exploit to takedown the botnet server and devise a strategy to clean the infected endpoints

Lab03: Stack overflow!

.oO Phrack 49 Oo.

Volume Seven, Issue Forty-Nine

File 14 of 16

BugTraq, r00t, and Underground.Org
bring you

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Smashing The Stack For Fun And Profit
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

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.

Lab03: 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 shell/flag!
- Explore a template exploit code (PwnTool)
- In-class tutorial
  - Learning PwnTool
  - Writing your first stack overflow exploit!

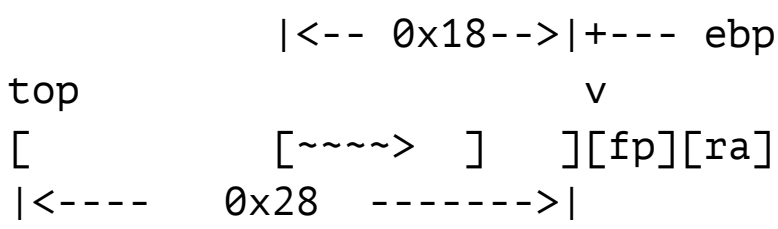
Reminder: crackme0x00

```
$ 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

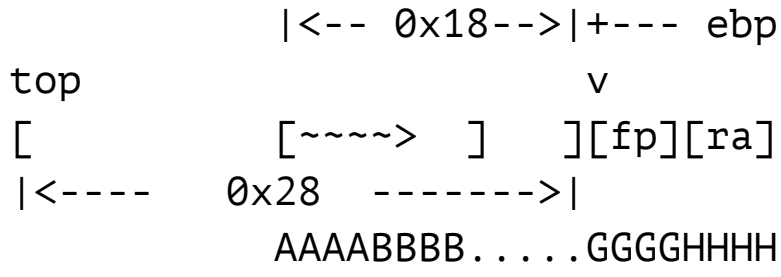
```
$ objdump -d crackme0x00
...
8048448:       8d 45 e8                lea    -0x18(%ebp),%eax
804844b:       89 44 24 04             mov    %eax,0x4(%esp)
804844f:       c7 04 24 8c 85 04 08    movl   $0x804858c,(%esp)
8048456:       e8 d5 fe ff ff          call   8048330 <scanf@plt>
```



Reminder: crackme0x00

```
main() {
  char s1[16];
  ...
  scanf("%s", &s1);
  ...
}
```

Reminder: crackme0x00



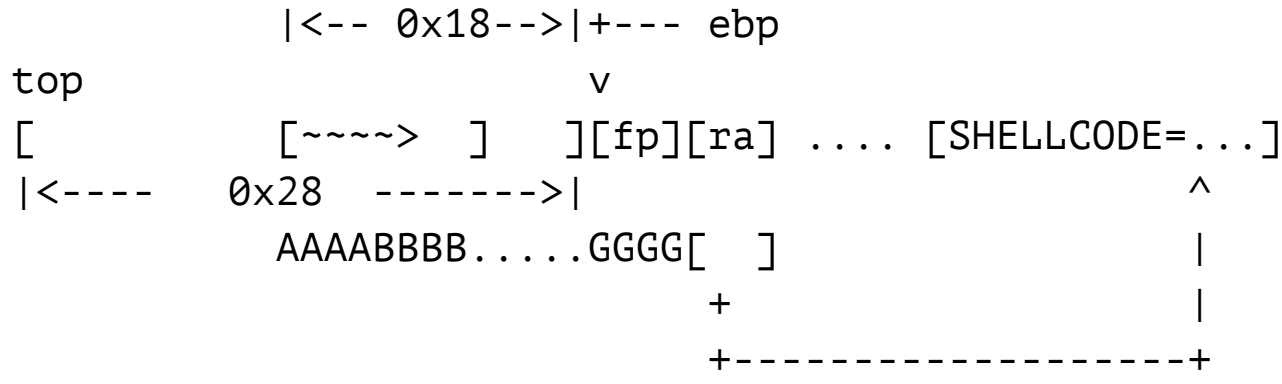
DEMO: pwntool

- cyclic
- checksec
- asm
- shellcraft
- template (exploit.py)

Where to put Shellcode?

- stack (today's tutorial)
- commandline argument
- environment vars

Example: Injecting Shellcode (e.g., env)



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

In-class Tutorial

- Step 1: Learn PwnTool
- Step 2: Play with your first exploit!

```
$ ssh YOURID@cyclonus.gtisc.gatech.edu -p 2023
$ ssh YOURID@cyclonus.gtisc.gatech.edu -p 2022
$ ssh YOURID@computron.gtisc.gatech.edu -p 2023
$ ssh YOURID@computron.gtisc.gatech.edu -p 2022

$ cd tut/lab04
$ cat README
```

References

- IDA Demo
- Phrack #49-14