Introduction

Insu Yun

Weclome to EE595-B Software Security!

- Lecture & Hands-on laboratory
  - Learn high level concepts for software attacks
  - Practice attacks with exercises

Goal: Lean how hackers attack software vulnerabilities!

Who should take this course?

- If you want to study this topic seriously (e.g., research or job)
  -> You **SHOULD** take this!

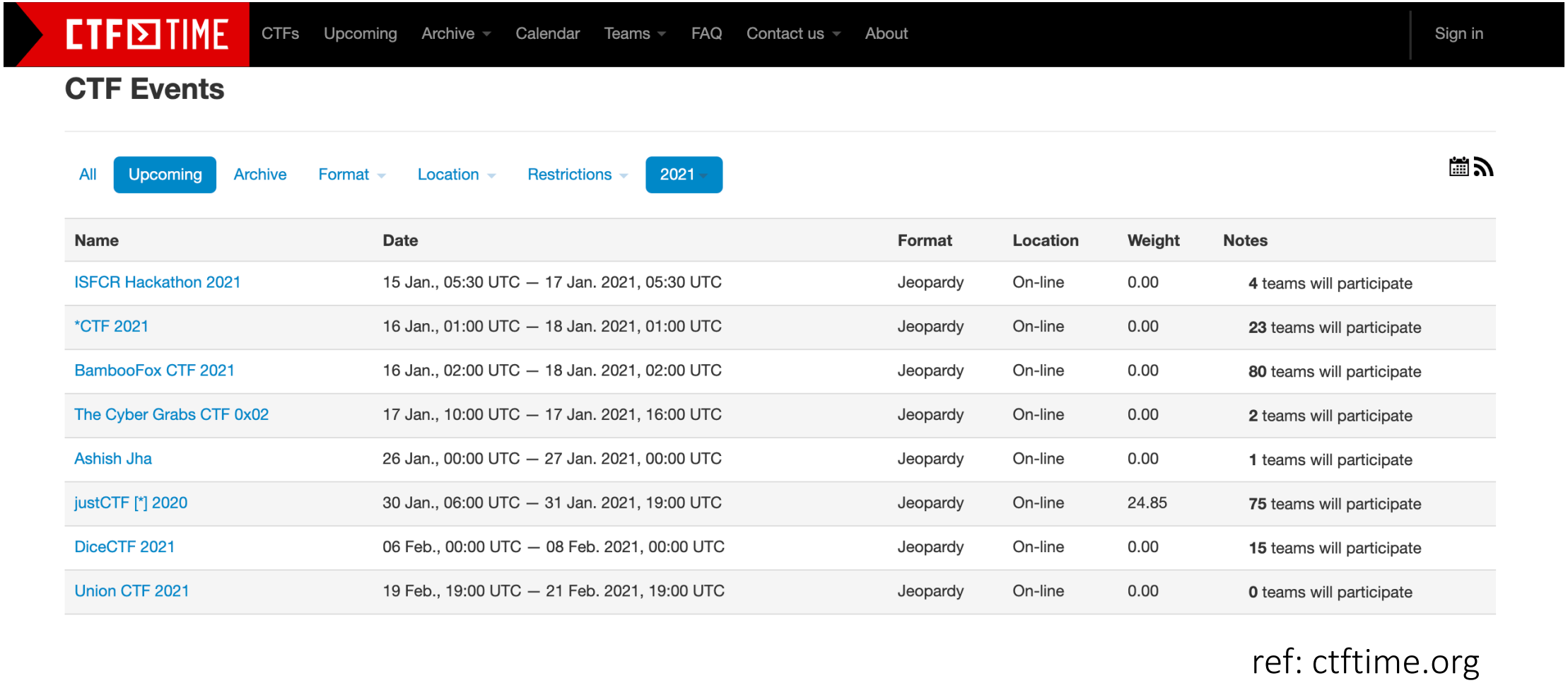
- If you are interested in this topic + you have enough free time
  -> Good to take! It would be fun!

- If you are interested in this topic + your schedule is tight
  -> Think carefully!
  This would be one of the toughest courses in your life!

Through Capture The Flag(CTF)

- Cyber game like puzzle solving
- Types: Jeopardy, Attack and defense

Many people are already enjoying CTF!



Name	Date	Format	Location	Weight	Notes
ISFCR Hackathon 2021	15 Jan., 05:30 UTC — 17 Jan. 2021, 05:30 UTC	Jeopardy	On-line	0.00	**4** teams will participate
*CTF 2021	16 Jan., 01:00 UTC — 18 Jan. 2021, 01:00 UTC	Jeopardy	On-line	0.00	**23** teams will participate
BambooFox CTF 2021	16 Jan., 02:00 UTC — 18 Jan. 2021, 02:00 UTC	Jeopardy	On-line	0.00	**80** teams will participate
The Cyber Grabs CTF 0x02	17 Jan., 10:00 UTC — 17 Jan. 2021, 16:00 UTC	Jeopardy	On-line	0.00	**2** teams will participate
Ashish Jha	26 Jan., 00:00 UTC — 27 Jan. 2021, 00:00 UTC	Jeopardy	On-line	0.00	**1** teams will participate
justCTF [*] 2020	30 Jan., 06:00 UTC — 31 Jan. 2021, 19:00 UTC	Jeopardy	On-line	24.85	**75** teams will participate
DiceCTF 2021	06 Feb., 00:00 UTC — 08 Feb. 2021, 00:00 UTC	Jeopardy	On-line	0.00	**15** teams will participate
Union CTF 2021	19 Feb., 19:00 UTC — 21 Feb. 2021, 19:00 UTC	Jeopardy	On-line	0.00	**0** teams will participate

ref: ctftime.org

I am also one of them (from DEFCON CTF)

Instructor / TA

- Instructor: Insu Yun
- TA
  - JunYoung Park
  - Yeongbin Hwang

Prerequisite

- (Strict) EE209 or other equivalent courses (e.g., CS230)
- (Recommended) Operating system, system programming, architecture

- Required skills: C, Python, C++

Lecture: In hybrid

- Offline: N1 #111
- Online: https://kaist.zoom.us/j/81246807331?pwd=M1FqWDJ3dk5tVlgwVVZFZXJuQi81UT09

- WARNING: We require time for stabilizing online session! You may feel uncomfortable due to setting issues.

General information

- Homepage: https://teemo.kaist.ac.kr/ee595/2022/
- Piazza: https://piazza.com/class/kjv59av4pi450v
  - Register now. For announcements. No KLMS.
- Discord: https://discord.gg/hsXNZH8efB
  - Use for tutorial + office hour
- Youtube: https://www.youtube.com/playlist?list=PLpYYZoHf-Y98wvXAvU7fKy39-Qv084Oav
- Email: ee595@hacking.kaist.ac.kr
  - Don’t use my or TA’s personal mail for this course

Office hour

- Please participate in this poll: FIX

- Please come to discord (For online) or #??? (For offline)
  - We will let you know place for offline after schedule

- **I strongly recommend you to join office hour!**
  - Concept != Reality
  - We will help you to tackle obstacles in reality (e.g., debugging)

Topics

- Lab01: Reverse engineering
- Lab02: Linux basic + shellcode
- Lab03: Stack overflow
- Lab04: Bypassing stack protection
- Lab05: Bypassing DEP/ASLR
- Lab06: Return-oriented programming
- Lab07: Remote exploits
- Lab08: Miscellaneous attacks
- Lab09: Heap exploits

> 10 challenges per lab

→ In total, you will solve 100 challenges in a semester

Three types of lectures

1. General lecture
   - Explain concepts of each topic
   - Slides (+ video) will be uploaded in the website
2. Lab review
   - At the day of deadline, I will briefly show you how to approach the challenge
   - Slides and videos will not be uploaded (Only live!)
3. Tutorial
   - Go through the tutorial (~ 30minutes)
   - Bring your labtop (or join discord), do yourself, and ask questions
   - Materials and videos will be available

Assignments + Exams

- Lab assignment: Tutorials (0-2) + 10 challenges
- No exam (Midterm + Final)
- Instead of exam, we will have In-class CTF

In-class CTF

- For 24 hours! instead of final exam!
- You can make a team (<= 3 people)

- Your tasks
  - Make a challenge for other students
  - Solve challenges from other students + from us

- We will share details later

Scoring

- For each challenge
  - Submit a flag with corresponding writeup
  - Total: 220 points = 200 points (10 challenges) + 20 point (one tutorial)

- In class CTF
  - Will be counted as TWO labs
  - i.e., 400 points!

- Late policy: 50% of original score (one extra week)

Grading rule: Overview

- 100%: lab assignment + in-class CTF

- We have two grading rules: General + Catch up
- Your grade = MAX(Grade $_{\text{General}}$ + Grade $_{\text{Catch up}}$)

- It will be a little bit complicated. But it is for you!

Grading rule: General

- Goal: Grade that you are doing well in general

- Grade $_{\text{Regular}}$ = Letter if your score >= Score(Letter)
- Score(Letter) = Ratio(Letter) * (20 * # of problems + 400) + 20 * # of tutorials
  - i.e., Except for tutorials, you should get Ratio(Letter) of scores. Note that 400 is the score for in-class CTF

- Ratio(“A+”) = 0.75, Ratio(“A0”) = 0.70, … (-0.05 for a lower letter)

Grading rule: Catch up

- Goal: Grade that you are catching up (even after grace period)

- Grade $_{\text{Catch up}}$ = Letter
  if your # of solved problems (including CTF) >= Number(Letter)
- Note that Grade $_{\text{Catch up}}$ can be “A-” at maximum

- Number(Letter) = # of problems * Ratio(Letter) + # of tutorials
  - i.e., Except for tutorials, you should solve Ratio(Letter) of challenges

Grading rule: Summary

- A+ >= 1850
- A0 >= 1740
- A- >= 1630
- B+ >= 1520
- B0 >= 1410
- B- >= 1300
- C+ >= 1190
- C0 >= 1080
- C- >= 970

- Otherwise F

- A- >= 68
- B+ >= 64
- B0 >= 60
- B- >= 55
- C+ >= 50
- C0 >= 46
- C- >= 42

- Otherwise F

Write-up

- You should submit a write-up for each challenge to get actual point!
- Be concise yet precise!
- You should use Markdown (https://www.markdownguide.org/) to write your writeup

- You don’t need to submit writeups for tutorials and the first lab

Write-up sample

```
## Description

In this challenge, ebp and the return address are protected by stackshield.
By doing debugging, you can see all ebp and ret values are keep tracking and
storing somewhere. However, when you make an input large enough, you will see
that a function pointer will be overwritten. And the overwritten value will be
store in EAX and make it jump at <main+96>. I put my shellcode as env, get the
address, and put it. In my case, the function pointer(0x08048b0a at 0xbffff654)
was overwritten. So we could learn, we could jump using the weakpoint even
though the stackshield is working on.

## Exploit
```python
  #!/usr/bin/env python3

  import os
  import sys

  from pwn import *

  payload = cyclic(100) + p32(0xbffff654)
  p = process(["/ee595/lab02/func_ptr/target"])
  p.sendline(payload)
  p.interactive()
```

Collaborator: Insu Yun
- I asked a question about how to get the core file from the server
```

Description

Exploit code

Collaborator

Tips

- Study in group (e.g., discussion)
- Get help from me and TAs (Office hour, Piazza)
  - Strongly recommend to use office hour!

- Manage your time
- Learn basic tools (e.g., gdb, pwntools, python)
- Try to tackle in order (not strict)

Tips (2)

- Start your assignment as soon as possible
  - Don’t assume that TAs will respond immediately
- Try to solve challenges as much as you can
  - e.g., Bad strategy: Solve only 7.5 challenge per lab
  - Later challenges will be much more difficult than earlier ones

Misconduct policy

- DO NOT SHARE YOUR CODE WITH OTHER STUDENTS
  - We encourage you to discuss, but discussion != sharing code
  - Do not copy other students' code
  - Do not copy any public code

About course material

- You should ***never*** share challenges/exploits/writeups online
- Once found → F

- Reason: It makes this course less useful for other students

Ethical hacking

- DO NOT ATTACK OTHER’s SYSTEM

- Attack your own and isolated environment
  - Use your home directory
  - DO NOT DoS our server (e.g., fork bomb)

Your account em

Registration for EE595-B: Softwar

noreply.kaist.hacking@gmail.com
insuyun에게

Dear nice_kap1tsa (insuyun@kaist.ac.kr):

Welcome to EE595-B: Software security.
You can login via the link below:

https://teemo.kaist.ac.kr:8443/api/GKQKDUKCC7OPR9W1QMXL2BT4UIJHMIWQ

Your api-key is:

GKQKDUKCC7OPR9W1QMXL2BT4UIJHMIWQ

If you did not request this, ignore this email.

Thanks,
ee595@hacking.kaist.ac.kr

첨부파일 2개

id_ed25519.pub

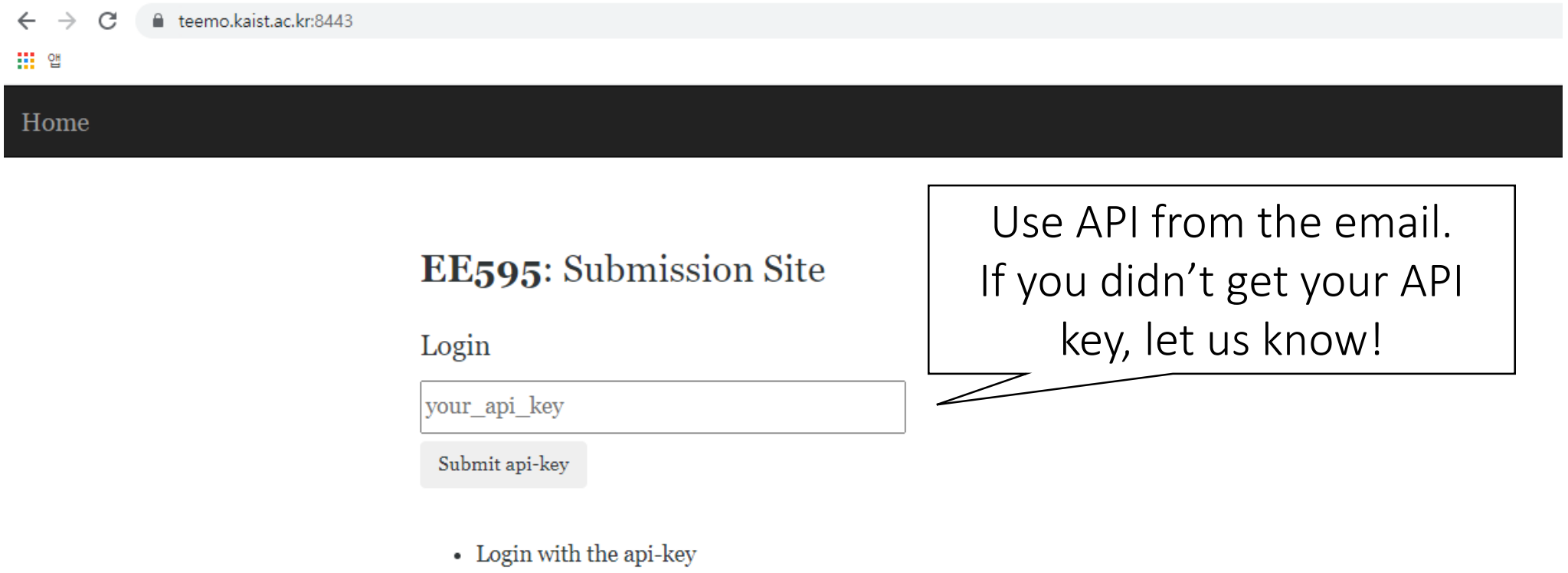
id_ed25519

Your ID: pseudonym
(Using Docker's generation mechanism)

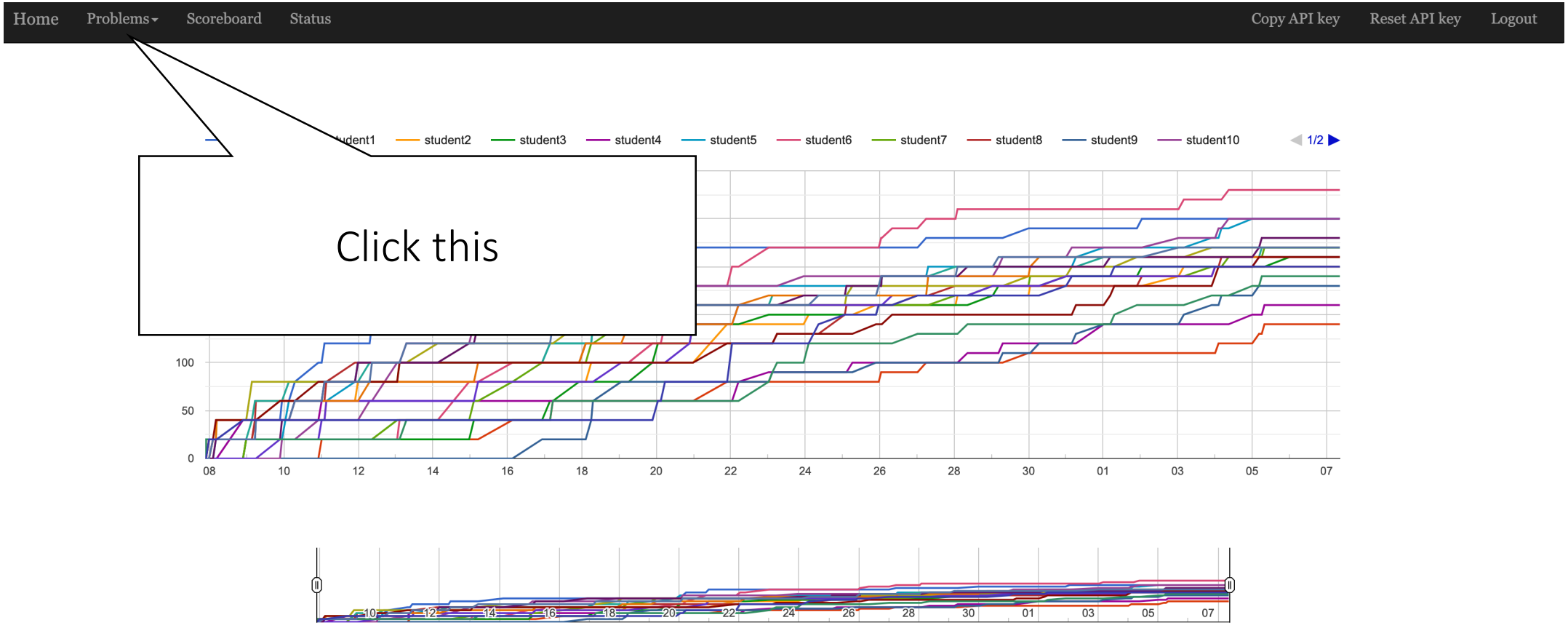
Your API Key

Your SSH key for challenge sever will be attached!

Submission : Login



Course homepage: Home



Course homepage: Lab

Home Problems Scoreboard Status Copy API key Reset API key Logout

Lab01: warm-up1

This is a warm-up lab that prepares you with the basic techniques used throughout this course. It is also a good chance to familiarize yourself with our submission and scoring system.

In this problem, your task is to defuse the bomb and get the flag. The binary _bomb_, is an executable that consists of multiple _phases_. Each phase expects you to enter a particular string (i.e., password) on stdin. If you enter the expected phrase, then the bomb is defused. Otherwise, it explodes, and you get _five_ points **deducted**. A thorough understanding of how each phase works at the binary level is required to solve this challenge without losing your points.

Note: you must maintain the Internet connection when you are solving this problem as it will update your progress (i.e., bomb defused/exploded) to the submission site, so be careful and not let the bomb explode! But be **creative** yet **careful** not to lose any points!

- [5 points] whenever we notice that you explode a bomb

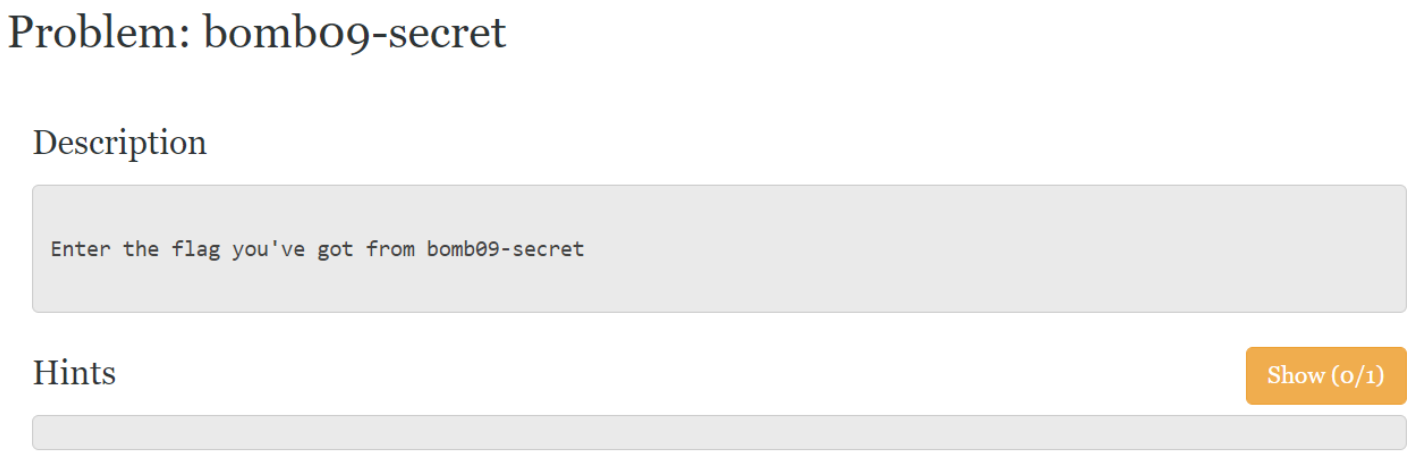
Course homepage: Lab

Name	Points	# Solved	Released (UTC-4)	Deadline (UTC-4)	Flag	Write-up
tut01-crackme	20	18	2021-01-08 00:00:00	2021-01-22 00:00:00	20 / 20 pts	Submit
bomb101-strcmp	20	17	2021-01-08 00:00:00	2021-01-22 00:00:00	20 / 20 pts	Submit
bomb102-funcall	20	17	2021-01-08 00:00:00	2021-01-22 00:00:00	20 / 20 pts	Submit
bomb103-password	20	12	2021-01-08 00:00:00	2021-01-22 00:00:00	10 / 20 pts	Submit
bomb104-quick	20	13	2021-01-08 00:00:00	2021-01-22 00:00:00	Submit	Submit
bomb105-jump	20	11	2021-01-08 00:00:00	2021-01-22 00:00:00	20 / 20 pts	Submit
bomb106-binary	20	15	2021-01-08 00:00:00	2021-01-22 00:00:00	20 / 20 pts	Submit
bomb107-array	20	15	2021-01-08 00:00:00	2021-01-22 00:00:00	10 / 20 pts	Submit

Note on flag

- Format: ee595{...}
  - e.g., ee595{thi5_i5_s4mple_fla9_f0r_y0u}
- Usually, locate at /ee595/[lab_name]/[challenge_name]/flag
  - e.g., /ee595/lab01/tut01-crackme/flag
- Sometimes, a binary embeds the flag itself
  - e.g., bomlab in lab01
- If you cannot find where the flag is, don’t hesitate ask us

Hint system for you!



- Some challenges have hints for you.
- If you want, feel free to open it!

Status

Status for student01

Lab	Problems Solved	Writeups Submitted	Total Score
lab01	10 / 11	0 / 0	180 / 220
lab02	8 / 11	6 / 10	100 / 220
lab03	10 / 12	9 / 10	180 / 240
lab04	11 / 11	9 / 10	190 / 220
lab05	11 / 11	9 / 10	180 / 220
lab06	12 / 12	10 / 10	200 / 240
lab07	11 / 11	10 / 10	210 / 220
lab08	9 / 10	9 / 10	180 / 200
lab09	10 / 11	9 / 10	180 / 220
CTF	15.0	N/A	300
Total (Grade)	-	-	1900 (A+)

- Grade will be dynamically changed based on the current max score
  - e.g., In lab01, your A+ bar will not be 1850, but 170!
- Total score reflects writeup status (i.e., no writeup == zero!)

CTF server

- ssh YOUR_ID@teemo.kaist.ac.kr –p 9000 –i YOUR_PRIVATE_KEY
- cd /ee595/lab01
- cat README

- If you are using Windows, please install WSL2 (https://docs.microsoft.com/en-us/windows/wsl/install) for using linux