Week 7 Thursday
Assembly III

Fill in the check-in form on cs107a.stanford.edu!

https://xkcd.com/292/

CS 107A, Spring 2022
Andrew Benson (adbenson@)
Starter Code

git clone /afs/ir/class/archive/cs/cs107a/cs107a.1226/WWW/exercises/assembly3
Announcements

- CS 107 Friday lecture: more assign5 practice, highly recommended to get some guidance from Nick
- 2nd round of 1:1s, sign up NOW pls
  - [https://calendly.com/adbenson/cs107a-1-1](https://calendly.com/adbenson/cs107a-1-1)
- Please fill out a mid-quarter evaluation for CS 107A
  - Link on CS 107A homepage or [https://forms.gle/1swzDjS4y6nB7SfX6](https://forms.gle/1swzDjS4y6nB7SfX6)
- assign5 walkthrough out
Agenda

- Warmup Questions
- assign5 Tour
- How to Approach an Assembly Puzzle
- nanobomb1 practice
Warmup Questions
assign5 Tour
Data (11 Spr21 Students) and Advice from last spring

- ATM Part a: 1.77 hours
- ATM Part b: 3.09 hours
- ATM Part c: 4.09 hours
- Dataset Aggregation: 1.16 hours
- Secure Vault Level 1: 1.43 hours
- Secure Vault Level 2: 4.32 hours
- Secure Vault Level 3: 6.64 hours
- Secure Vault Level 4: 9.95 hours

Note: Data from last winter is significantly less time 🤷

Advice from a spring student: “Start with vault first. I did this and am so glad I did. Most people will start with ATM, so you will be able to get help in office hours. Near the deadline, it will be difficult to get help with vault. Also, you don’t want to do vault when you are stressed and tired near the deadline; save ATM for that.”
DO NOT ****ING RUN THE BOMB VAULT

Watch the walkthrough if you want to know what happens when you do so.
How to Approach an Assembly Puzzle
How to Setup Working on an Assembly Puzzle

- [CS 107A only] Run `make` if you haven't already done so.
- Run `gdb` on the executable
  - For CS 107A: `gdb ./nanobomb1`
- Switch `gdb` to view assembly: `layout asm`
- For safety, put a breakpoint on the explode function
  - For CS 107A: `b nano_explode`
- Put a breakpoint on the function that reads a password and checks it
  - For CS 107A: `b try_defuse`
- Run the executable
  - For CS 107A: `r`, For CS 107: `r input.txt`
Guidelines for Solving Assembly Puzzles

- Start by reading over the function's assembly. Try to identify:
  - Any calls to the explode function (ex. `nano_explode`)
  - Where the function returns (ex. the `retq` instruction)
  - Any notable function calls (ex. a call to `read_64_words` gives you a clue what the function on the whole does)
- Try to use the jumps in the assembly to break down the execution flow into phases
  - See if you can identify if-else checks
  - Maybe you can identify a loop structure
- Determine what execution path you must follow to successfully return without calling the explode function
Guidelines for Solving Assembly Puzzles (cont)

- Try to identify the first check done on your password in the necessary execution flow
  - Each time you form a hypothesis, create a password that should pass the check, and try it, verifying that the execution proceeds as you expect
  - Once you pass the check, write down your understanding of what is being checked since you’ll likely need to change the password later on
- Keep going!
nanobomb1 practice