Buffer Overflow

CSE 351 Autumn 2023

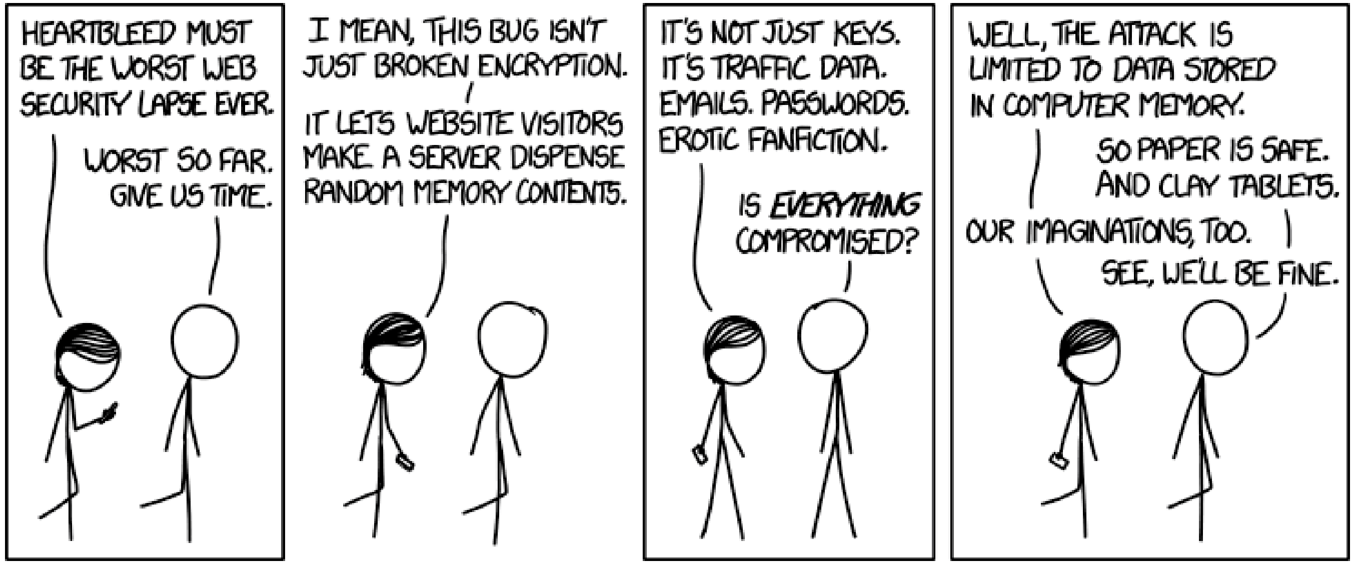
Guest Instructor:

Ellis Haker

Teaching Assistants:

Afifah Kashif
Bhavik Soni
Cassandra Lam
Connie Chen
David Dai
Dawit Hailu
Eyoel Gebre
Joshua Tan

Malak Zaki
Naama Amiel
Nayha Auradkar
Nikolas McNamee
Pedro Amarante
Renee Ruan
Simran Bagaria
Will Robertson

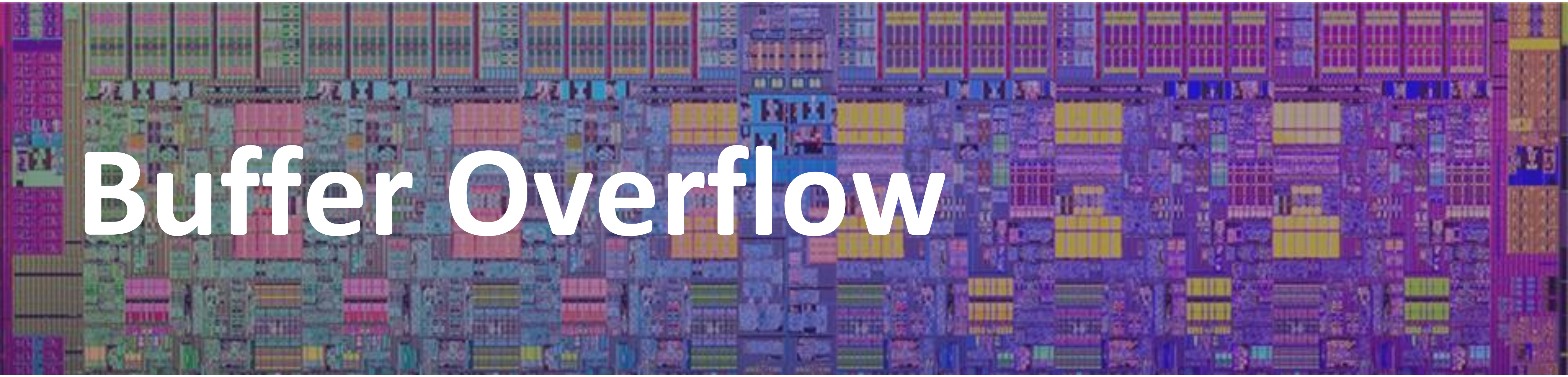

HEARTBLEED MUST BE THE WORST WEB SECURITY LAPSE EVER.
WORST SO FAR. GIVE US TIME.
I MEAN, THIS BUG ISN'T JUST BROKEN ENCRYPTION.
IT LETS WEBSITE VISITORS MAKE A SERVER DISPENSE RANDOM MEMORY CONTENTS.
IT'S NOT JUST KEYS. IT'S TRAFFIC DATA. EMAILS. PASSWORDS. EROTIC FANFICTION.
IS EVERYTHING COMPROMISED?
WELL, THE ATTACK IS LIMITED TO DATA STORED IN COMPUTER MEMORY.
SO PAPER IS SAFE. AND CLAY TABLETS.
OUR IMAGINATIONS, TOO.
SEE, WE'LL BE FINE.

Alt text: I looked at some of the data dumps from vulnerable sites, and it was ... bad. I saw emails, passwords, password hints. SSL keys and session cookies. Important servers brimming with visitor IPs. Attack ships on fire off the shoulder of Orion, c-beams glittering in the dark near the Tannhäuser Gate. I should probably patch OpenSSL.

http://xkcd.com/1353/

Relevant Course Information

- Lab 3 released today, due next Friday (11/10)
  - You will have everything you need by the end of this lecture
- Mid-Course Survey
  - Released on Sunday (11/5), closes Thursday (11/9)
- Midterm starts Thursday
  - Instructions will be posted on Ed Discussion
  - You are permitted to discuss high-level concepts and give hints, but are **not** allowed to solve the problems together
  - We will be available on Ed Discussion (private posts, please) and support hours to answer clarifying questions



Buffer Overflow

Lesson Summary (1/3)

- Buffer overflow is a bug where more data is written to a buffer (array) than there is space for
  - Can be used to attack a system by overwriting important data
- Distressingly common in real programs
  - Most common technical source of security vulnerabilities
  - Programmers keep making the same mistakes ☹
  - Recent measures make these attacks much more difficult, but not impossible!
- Exploits based on buffer overflow
  - Stack smashing: Altering the execution of a program - overwrite return address to somewhere else in instruction memory
  - Code injection: Run arbitrary code on target's computer
    - put code in buffer
    - overwrite return address to start of buffer
    - when function returns, executes buffer code

Lesson Summary (2/3)

- Array bounds checking

  safer version of gets

  - In C, check manually or use safe library functions (eg: fgets)
  - Done automagically in most modern languages (eg: Java)
- Stack canaries
  - Store a secret value in the stack before the return address, check that it wasn't overwritten before returning
- Non-executable memory regions
  - Prevent code from being executed on the stack — only prevents code injection

    - turned off in gdb
- Randomized stack offsets
  - Put a random amount of padding in memory before the stack

    -harder to predict where things are in memory

    - turned off on attu

Lesson Summary (3/3)

writing past the end of the array

modifying return to execute your own code

- Terminology:
  - Buffer, buffer overflow, stack smashing, code injection attack

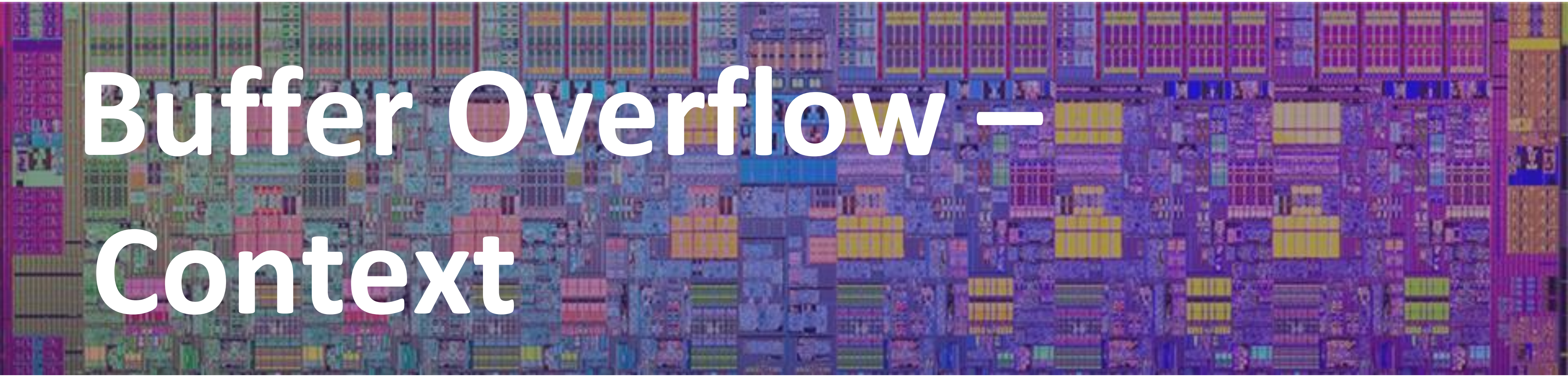
array

overwriting return

- Learning Objectives:
  - Define buffer overflow and explain how it occurs.
  - Identify elements of C programs that make them vulnerable to buffer overflow.
  - Identify methods of protecting against buffer overflow at multiple levels (hardware, OS, software).
  - Perform stack smashing and code injection exploits.

- What lingering questions do you have from the lesson?

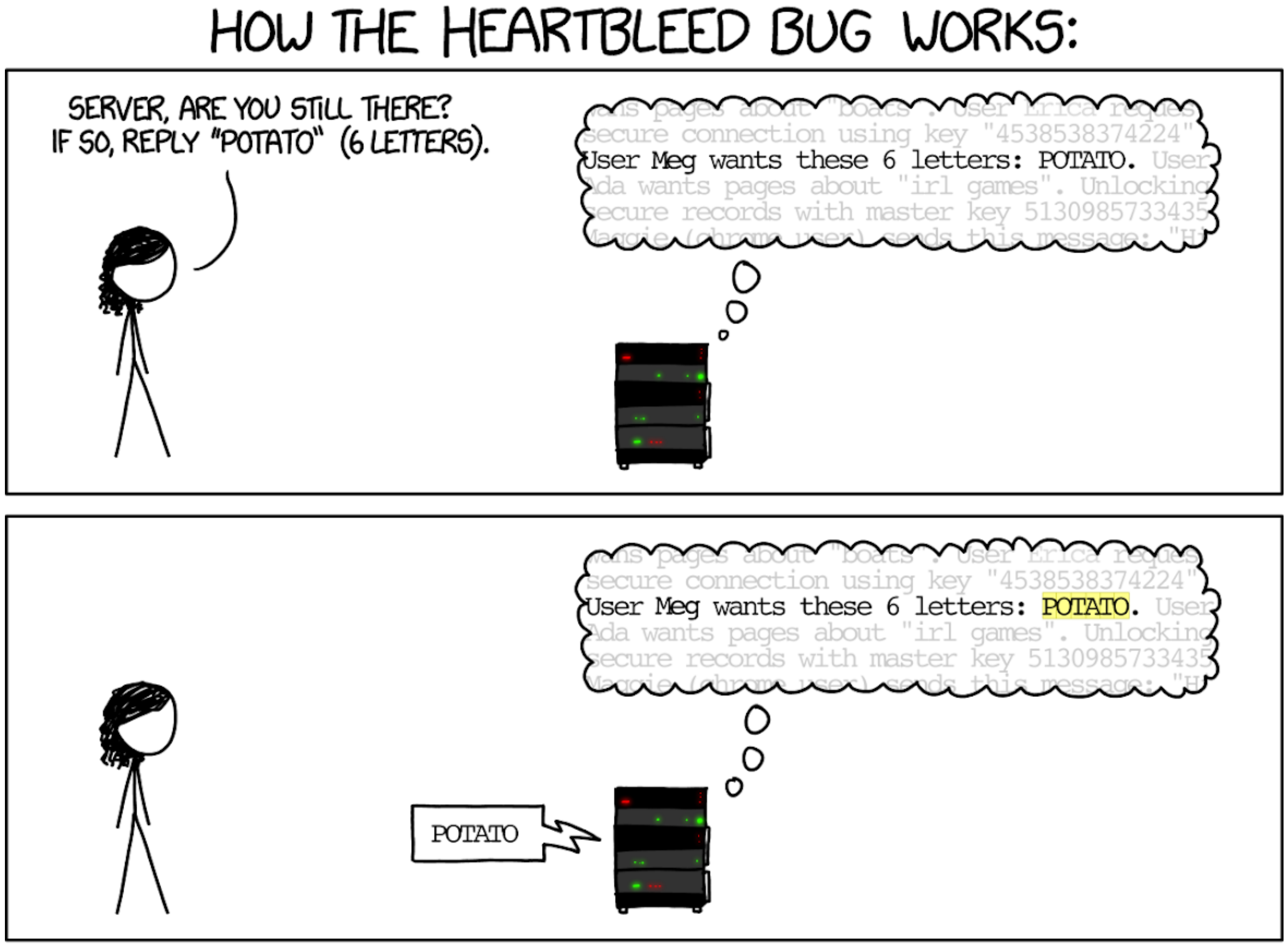
Buffer Overflow – Context



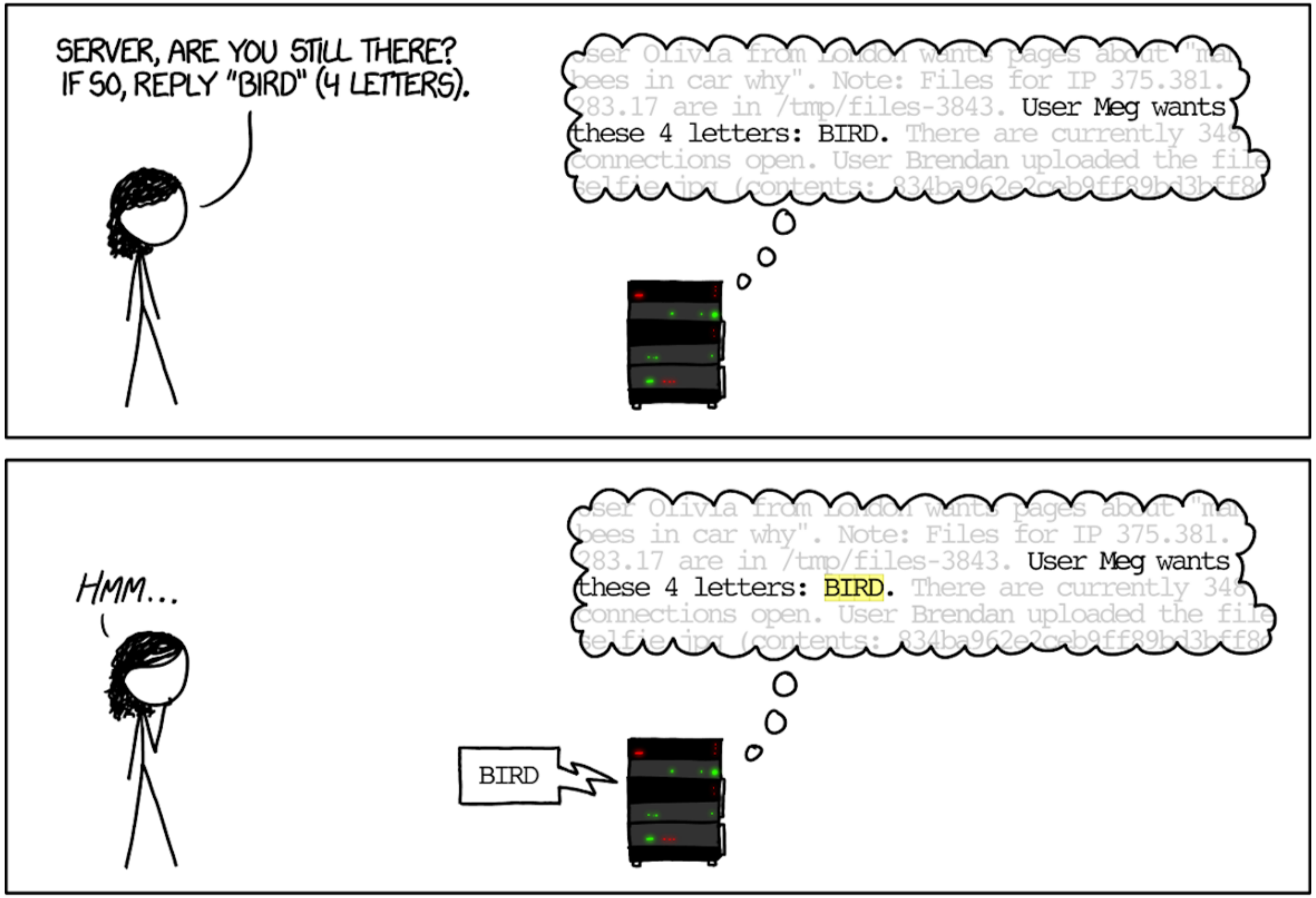
Example: the original Internet worm (1988)

- Early versions of the finger server (`fingerd`) used `gets()` to read the argument sent by the client:
  - `finger droh@cs.cmu.edu`
- Worm attacked `fingerd` server with phony argument:
  - `finger` "*exploit-code padding new-return-addr*"
  - Exploit code: executed a root shell on the victim machine
- Scanned for other machines to attack
  - Invaded ~6000 computers in hours (10% of the Internet)
    - See June 1989 article in *Comm. of the ACM*
  - The author of the worm (Robert Morris) was prosecuted
    - First conviction under Computer Fraud and Abuse Act
    - Now an MIT professor

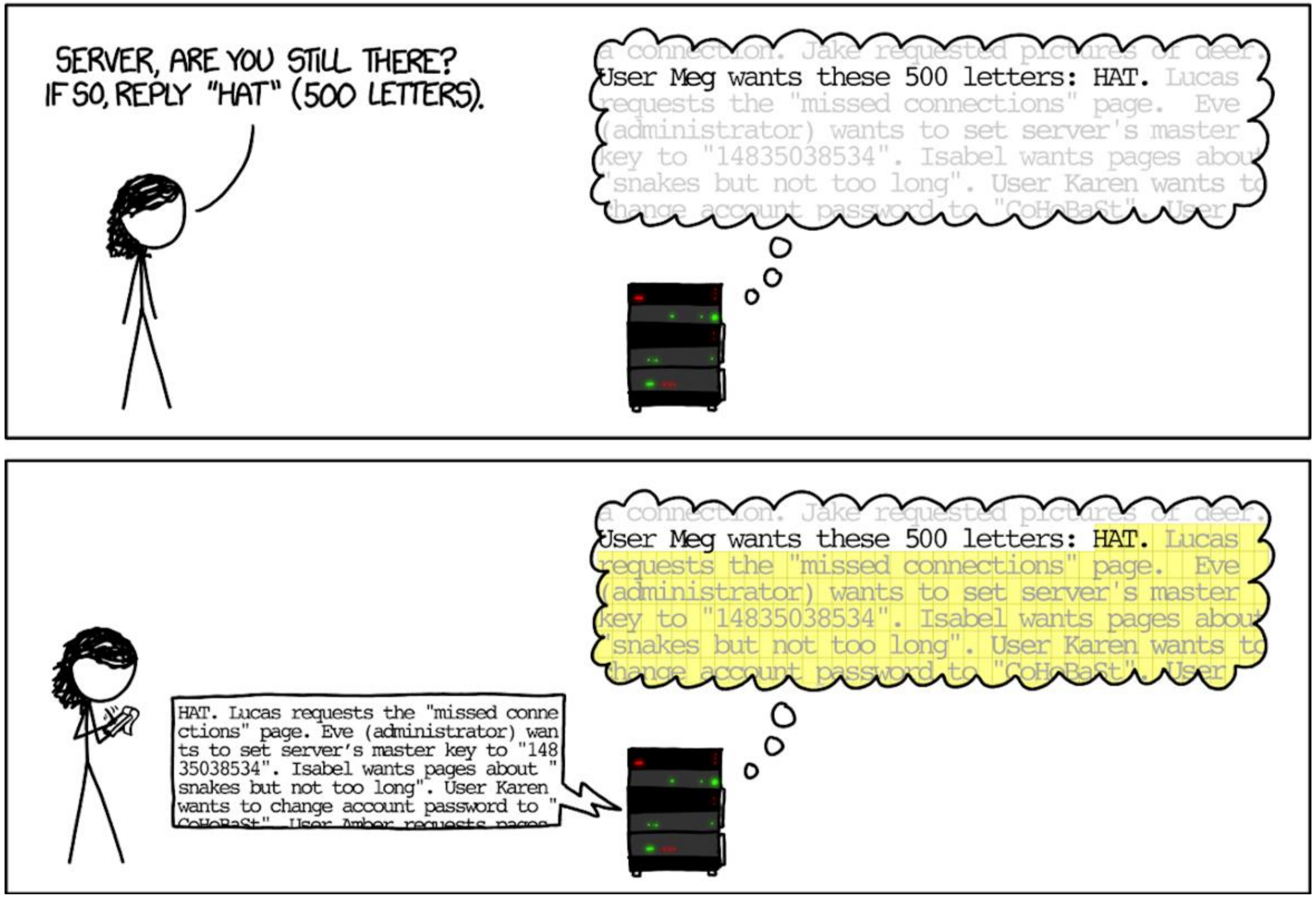
Example: Heartbleed (2014)


HOW THE HEARTBLEED BUG WORKS:
SERVER, ARE YOU STILL THERE? IF SO, REPLY "POTATO" (6 LETTERS).
ants pages about "boats". User Erica reques
secure connection using key "4538538374224"
User Meg wants these 6 letters: POTATO. User
Ada wants pages about "irl games". Unlocking
secure records with master key 5130985733435
Maggie (chrome user) sends this message: "H
ants pages about "boats". User Erica reques
secure connection using key "4538538374224"
User Meg wants these 6 letters: POTATO. User
Ada wants pages about "irl games". Unlocking
secure records with master key 5130985733435
Maggie (chrome user) sends this message: "H
POTATO

Example: Heartbleed (2014)

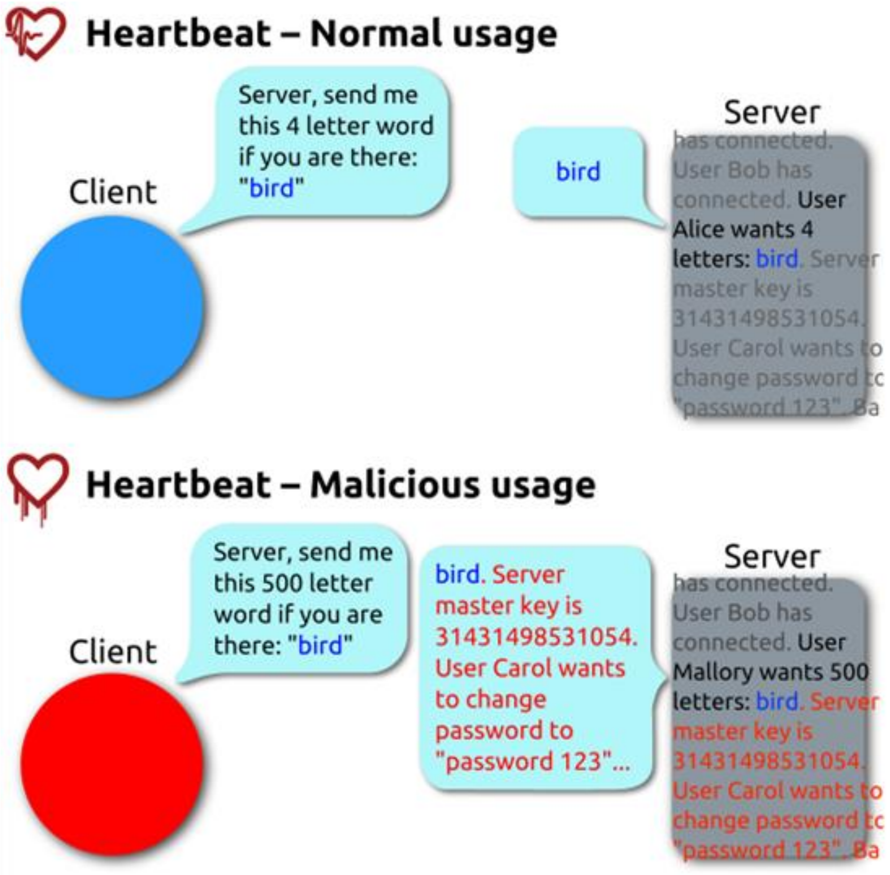

SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
User Olivia from London wants pages about "ma bees in car why". Note: Files for IP 375.381. 283.17 are in /tmp/files-3843. User Meg wants these 4 letters: BIRD. There are currently 348 connections open. User Brendan uploaded the file selfie.jpg (contents: 834ba962e2ceb9ff89bd3bff8
HMM...
User Olivia from London wants pages about "ma bees in car why". Note: Files for IP 375.381. 283.17 are in /tmp/files-3843. User Meg wants these 4 letters: BIRD. There are currently 348 connections open. User Brendan uploaded the file selfie.jpg (contents: 834ba962e2ceb9ff89bd3bff8
BIRD

Example: Heartbleed (2014)


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
a connection. Jake requested pictures of deer. User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User
a connection. Jake requested pictures of deer. User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "CoHoBaSt". User
HAT. Lucas requests the "missed conne ctions" page. Eve (administrator) wan ts to set server's master key to "148 35038534". Isabel wants pages about " snakes but not too long". User Karen wants to change account password to " CoHoBaSt". User Amber requests pages

Heartbleed Details

- Buffer over-read in OpenSSL
  - Open source security library
  - Bug in a small range of versions
- "Heartbeat" packet
  - Specifies length of message
  - Server echoes it back
  - Library just "trusted" this length
  - Allowed attackers to read contents of memory anywhere they wanted
- Est. 17% of Internet affected
  - "Catastrophic"
  - Github, Yahoo, Stack Overflow, Amazon AWS, ...


Heartbeat – Normal usage
Client
Server, send me this 4 letter word if you are there: "bird"
bird
Server
has connected. User Bob has connected. User Alice wants 4 letters: bird. Server master key is 31431498531054. User Carol wants to change password tc "password 123". Ba
Heartbeat – Malicious usage
Client
Server, send me this 500 letter word if you are there: "bird"
bird. Server master key is 31431498531054. User Carol wants to change password to "password 123"...
Server
has connected. User Bob has connected. User Mallory wants 500 letters: bird. Server master key is 31431498531054. User Carol wants to change password tc "password 123". Ba

By FenixFeather - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=32276981

Hacking Cars (2010)

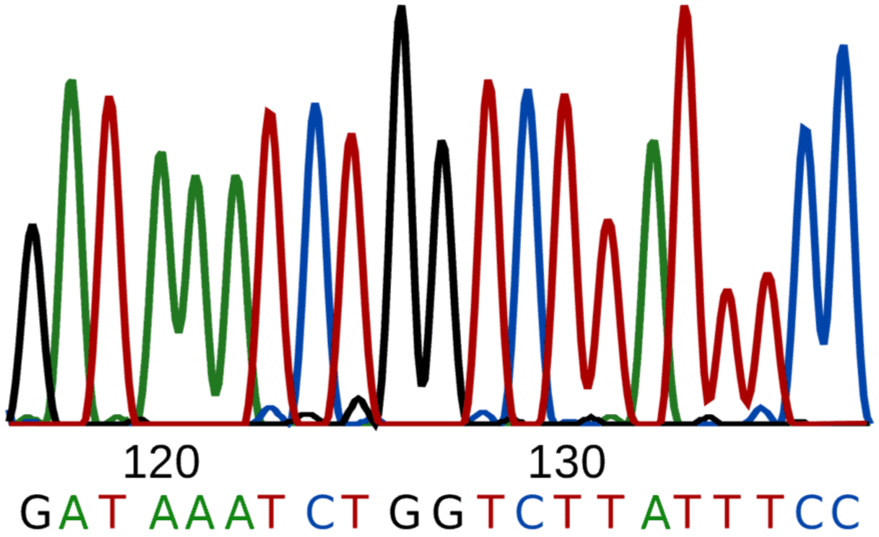
- UW CSE research demonstrated wirelessly hacking a car using buffer overflow
  - http://www.autosec.org/pubs/cars-oakland2010.pdf
- Overwrote the onboard control system's code
  - Disable brakes, unlock doors, turn engine on/off

Hacking DNA Sequencing Tech (2017)

Computer Security and Privacy in DNA Sequencing

Paul G. Allen School of Computer Science & Engineering, University of Washington

- Potential for malicious code to be encoded in DNA!
- Attacker can gain control of DNA sequencing machine when malicious DNA is read
- Ney et al. (2017): https://dnasec.cs.washington.edu/



Discussion Questions

- Discuss the following question(s) in groups of 3-4 students
  - I will call on a few groups afterwards so please be prepared to share out
  - Be respectful of others' opinions and experiences

- If they're so well-known, why do buffer overflow attacks still happen?
  - Why do we still use unsafe languages like C?
  - What kinds of incentives dissuade tech companies from prioritizing security?

Group Work Time

- During this time, you are encouraged to work on the following:
  - If desired, continue your discussion
  - Work on the lesson problems (solutions at the end of class)
  - Work on the homework problems

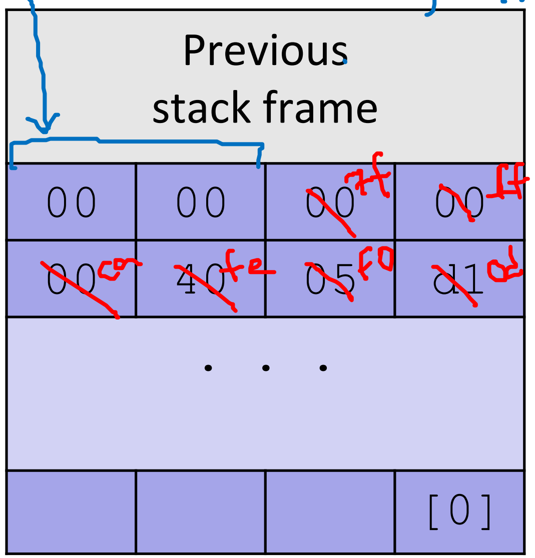
- Resources:
  - You can revisit the lesson material
  - Work together in groups and help each other out
  - Course staff will circle around to provide support

Practice Question

gets(char * dest) — dest = pointer to buffer
- reads input from user and stores it in dest

- buggy is vulnerable to stack smashing!
- What is the minimum number of characters that gets must read in order for us to change the return address to a stack address?
  - For example: (0x00 00 7f ff ca fe f0 0d)

don't need to write leading 0s because they're already there


Previous stack frame
00 00 ~~00~~ 7f ~~00~~ ff
~~00~~ ca ~~40~~ fe ~~05~~ f0 ~~d1~~ 0d
. . .
[0]

64-16 = 48 bytes in buffer
+ 6 bytes for return address
54

```
buggy:        64
   subq   $0x40, %rsp
    ...
   leaq   16(%rsp), %rdi
   call   gets
    ...
```

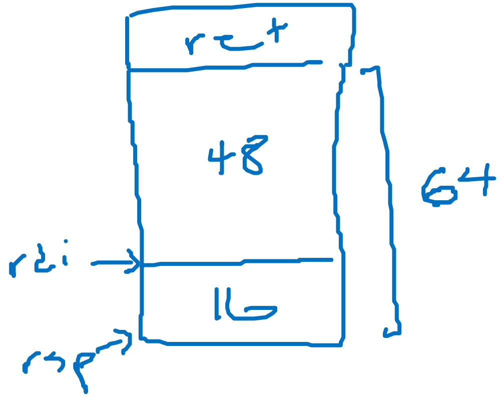
A. 27

B. 30

C. 51

[✓ D. 54]

E. We're lost…



Think this is cool?

- You'll love Lab 3 😉
  - Released Today, due next Friday (11/10)
  - Some parts *must* be run through GDB to disable certain security features
- Take CSE 484 (Security)
  - Several different kinds of buffer overflow exploits
  - Many ways to counter them
- Nintendo fun!
  - Using glitches to rewrite code: https://www.youtube.com/watch?v=TqK-2jUQBUY
  - Flappy Bird in Mario: https://www.youtube.com/watch?v=hB6eY73sLV0

↗ super cool! 🙂