Systems and Network Security

Lec 1: Course Overview

WWW

- Course website on CourSys:
  - Syllabus, policy, schedule, slides, assignments, project
  - Discussion board, announcements
  - Submissions, grades

Communication

- taowang@sfu.ca
  - Use my email for topics that are sensitive, confidential, etc…
- Discussion Board
  - Use this if your question/discussion would be beneficial for other students
- **Please** be professional and plan ahead

Office Hours

- Thursdays (2 PM to 2:30 PM) using Zoom

What is this course about?

What’s wrong with this picture?

What is this course about?

What’s wrong with this picture?

Course Goals

- Learn how an attacker gains control of a system
- Learn how to defend a system
- Gain hands-on experience in various security topics

Topics

- System security:
  - Shellcode Development
  - Buffer overflows
  - Control-flow hijacking and defenses
  - Return-oriented programming
- Network security:
  - Network monitoring and analysis
  - ARP attacks
  - TCP/IP attacks
  - DNS attacks
  - Firewalls and VPNs

Course objectives

- How to think like an attacker
  - To develop the “security mindset”

- Technical aspects of security
  - Reproducing attacks
  - Building defensive solutions

Course objectives

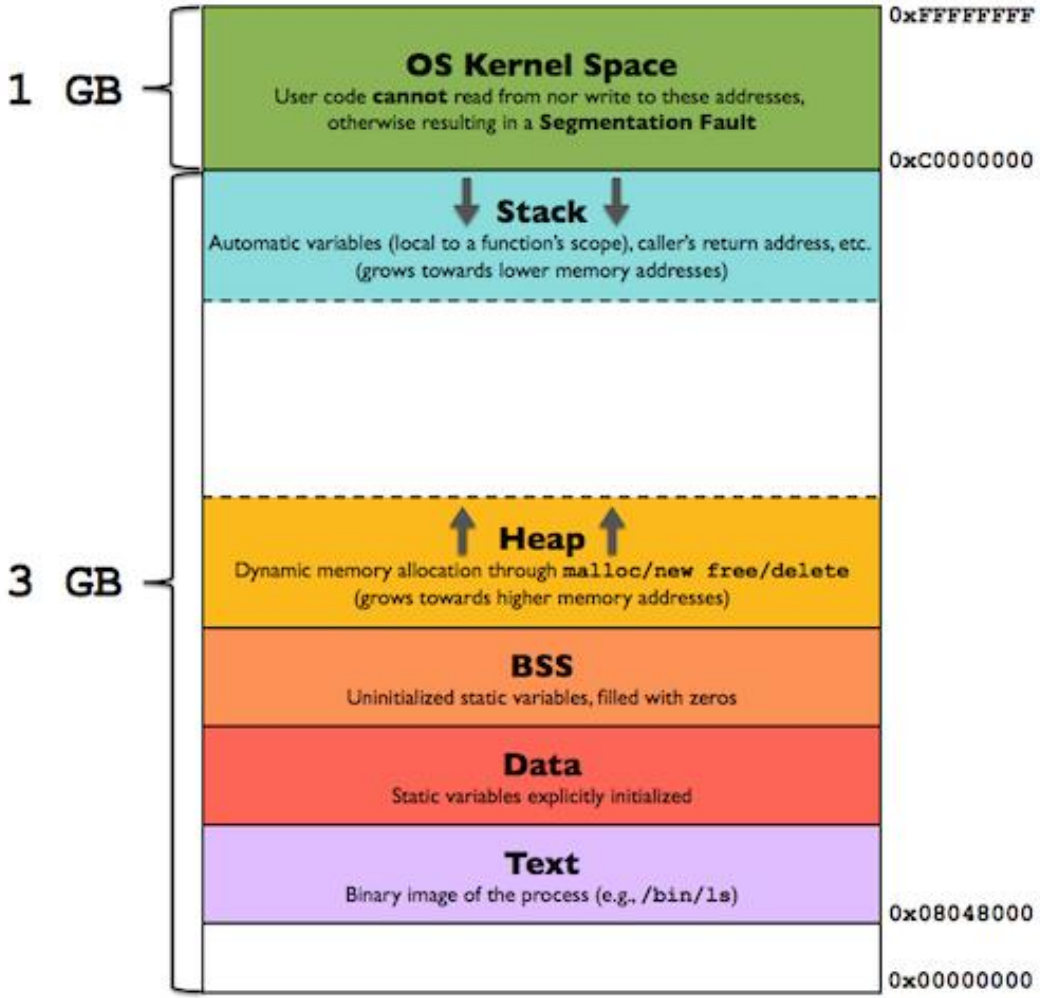
- We design the course for a deep dive into two of the hardest security topics to learn technically
  - Buffer overflow is the core of memory safety issues, around 70% of vulnerabilities
  - Almost all attacks are delivered to your computer through networks, and networks themselves are highly vulnerable
- Security is very broad, and we cannot cover, e.g.:
  - Hardware security, code analysis, web security, database security, cryptography

Prerequisites

- Assumed:
  - operating systems (e.g., memory layout, execution semantics)
  - computer networks (e.g., IP networks, Internet naming and routing)
  - strong programming skills in C/C++ and Python
  - ability to write working Assembly code
  - knowledge of software dev. tools in Linux (gcc, gdb, objdump, ld, git, etc.)
  - ability to learn new languages, tools and frameworks

- It's a six-credit course - expect to do a lot of work!

Assess your knowledge base




```
int (*func)();
func = (int (*)()) code;
(int)(*func)();
```

```
mov ebx, 42
mov eax, 0x1
int 0x80
```

Assess your knowledge base

- Networking
  - Nmap
  - Wireshark labs: https://www-net.cs.umass.edu/wireshark-labs/
  - E.g.,: TCP lab: https://www-net.cs.umass.edu/wireshark-labs/Wireshark_TCP_v7.0.pdf

Credits: Computer Networking: A Top-Down Approach, 7th ed., J.F. Kurose and K.W. Ross

Course Materials

- Materials are research papers, book chapters, related articles etc.
- Use the slides to guide your study
- References will be available at the course website

Grading

- Weekly labs: 60%
- Final Project (Group of 3): 25%
- Quizzes: 15%

Labs

- 11 Labs
- Lab every Monday afternoon
- The lab work needs to be your own
- Submit by the end of **next Sunday**

Lab Report *(important)*

- You need to submit detailed lab report and the required code.
- The report:
  - describes your work **in details**
  - documents all of your attacks/defenses including all steps, command line instructions, and console output with **proper screenshots**
  - should be **thorough** enough that your attacks can be easily **replicated**
  - should include learned lessons and interesting observations
- Recommended <20 pages

Lab Report *(important)*

- Common mistakes:
  - Missing explanations: mysterious variables, unclear method, results not shown
  - Presentation: unfocused screenshots, illogical steps, irrelevant information
  - Plagiarism (next slide)

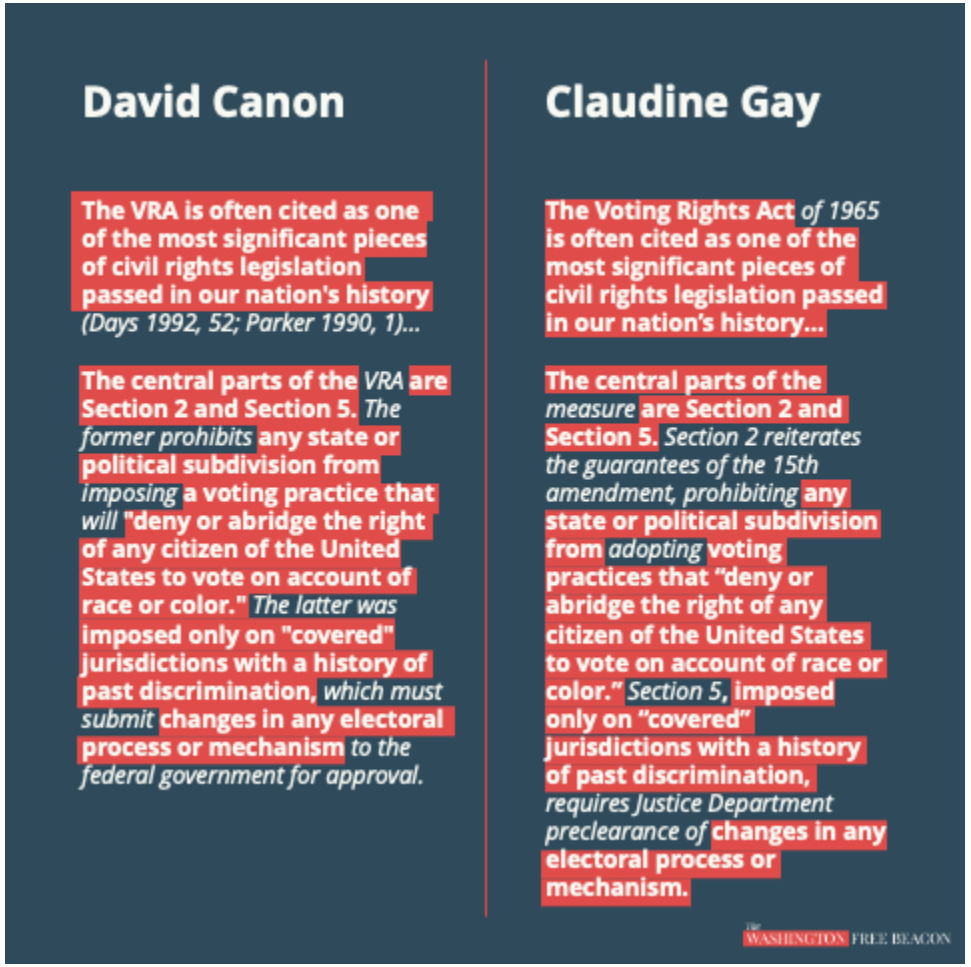
Plagiarism

- What is plagiarism?
- You are encouraged to discuss the assignment and ask for advice from other students, but:
  - Do not copy any text, code, or images
  - Do not show anything on a screen
  - Do not send files
  - Do not write down the solution
- There are lab questions for which I expect unique answers from each and every student

Plagiarism

- I am required to report all incidents of academic dishonesty, and the consequences are not up to me
  - At minimum, a serious penalty on the report after re-submitting it
- ***The plagiarising student and the plagiarised student cannot be distinguished and will receive the same consequences***
  - If you show someone your code/report and ask them not to copy it, but they do, you will still be reported for academic dishonesty

Example of Plagiarism



David Canon

The VRA is often cited as one of the most significant pieces of civil rights legislation passed in our nation's history *(Days 1992, 52; Parker 1990, 1)...*

The central parts of the *VRA* are Section 2 and Section 5. *The former prohibits* any state or political subdivision from *imposing* a voting practice that *will* "deny or abridge the right of any citizen of the United States to vote on account of race or color." *The latter was* imposed only on "covered" jurisdictions with a history of past discrimination, *which must submit* changes in any electoral process or mechanism *to the federal government for approval.*

Claudine Gay

The Voting Rights Act *of 1965* is often cited as one of the most significant pieces of civil rights legislation passed in our nation's history...

The central parts of the *measure* are Section 2 and Section 5. *Section 2 reiterates the guarantees of the 15th amendment, prohibiting* any state or political subdivision from *adopting* voting practices that "deny or abridge the right of any citizen of the United States to vote on account of race or color." *Section 5,* imposed only on "covered" jurisdictions with a history of past discrimination, *requires Justice Department preclearance of* changes in any electoral process or mechanism.

THE WASHINGTON FREE BEACON

Final Project

- This is your opportunity to explore or dig deeper in a specific security-related topic.
  - Related to **systems** and/or **networking** topics
  - Can be a research-related project
  - Reproducing known and recent attacks, or security-related systems
  - Searching for a vulnerability: Analysis of a program, misconfiguration in the network
  - Other topics: Smart home security, ML-based Firewalls IDS

Final Project

- Has to have a heavy implementation component
- Highly recommended to discuss with the instructor and/or in the discussion board
- Four major milestones/checkpoints
  - The first one is on Feb 12th
- Details on the website

Quizzes

- Two quizzes
  - Their dates are posted on the Schedule page

- Every quiz will cover **everything** in the course so far

Participation

- I expect students to take active and regular roles in discussion, asking/answering questions, etc.

- Discussion board:
  - discuss the assignments and projects and other class materials
  - you can also use it to exercise the “security mindset”
    - Discussing recent security incidents
    - Posting and discussing resources and news
    - …

Late Submission Policy

- Late submissions will not be graded.
- Unless
  (1) there is an excused absence (e.g., illness with sick note, emergency) **and**
  (2) student made arrangements with the instructor prior to the deadline.

Ethics

As Uncle Ben said…

Don't try this at home

- Never attack a system without the express consent of the owner
- Never use any of the attacks on a network connected to the Internet!
  - Even if it seems simple (e.g., TCP RST)
- Project/assignments?
  - code should run in an isolated env (e.g., VM)
- If in doubt, please contact me!

Ethics Forms

To receive a non-zero grade in this course, you must sign the ethics form by 11:59pm on February 1, 2024.

- The form is available on CourSys.
- Late forms will not be accepted.

Introduction to Security

What is Security?

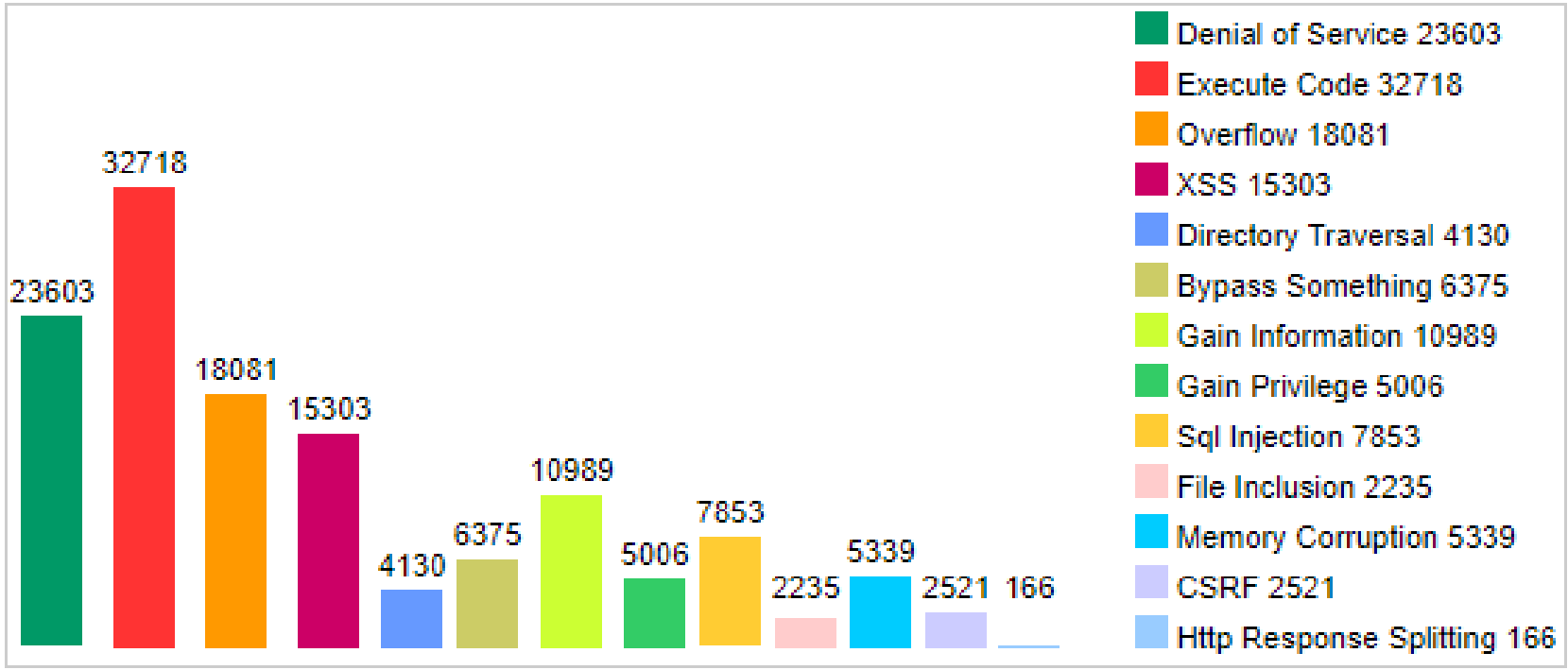
What is Security?

“Managing a malicious adversary [and] guaranteeing **properties** even if a malicious adversary tries to attack” – Adrian Perrig

Security is Hard

1. Lack of security-driven designs
   - For many software systems and network protocols
   - Focusing on functionality not security!
2. Finding vulnerability has become a business
3. Side-channel attacks
4. Too many threats
5. …

Lack of security-driven designs



https://www.cvedetails.com/vulnerabilities-by-types.php

Lack of security-driven designs

Top 50 Products By Total Number Of "Distinct" Vulnerabilities in 2019

Go to year: 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 All Time Leaders

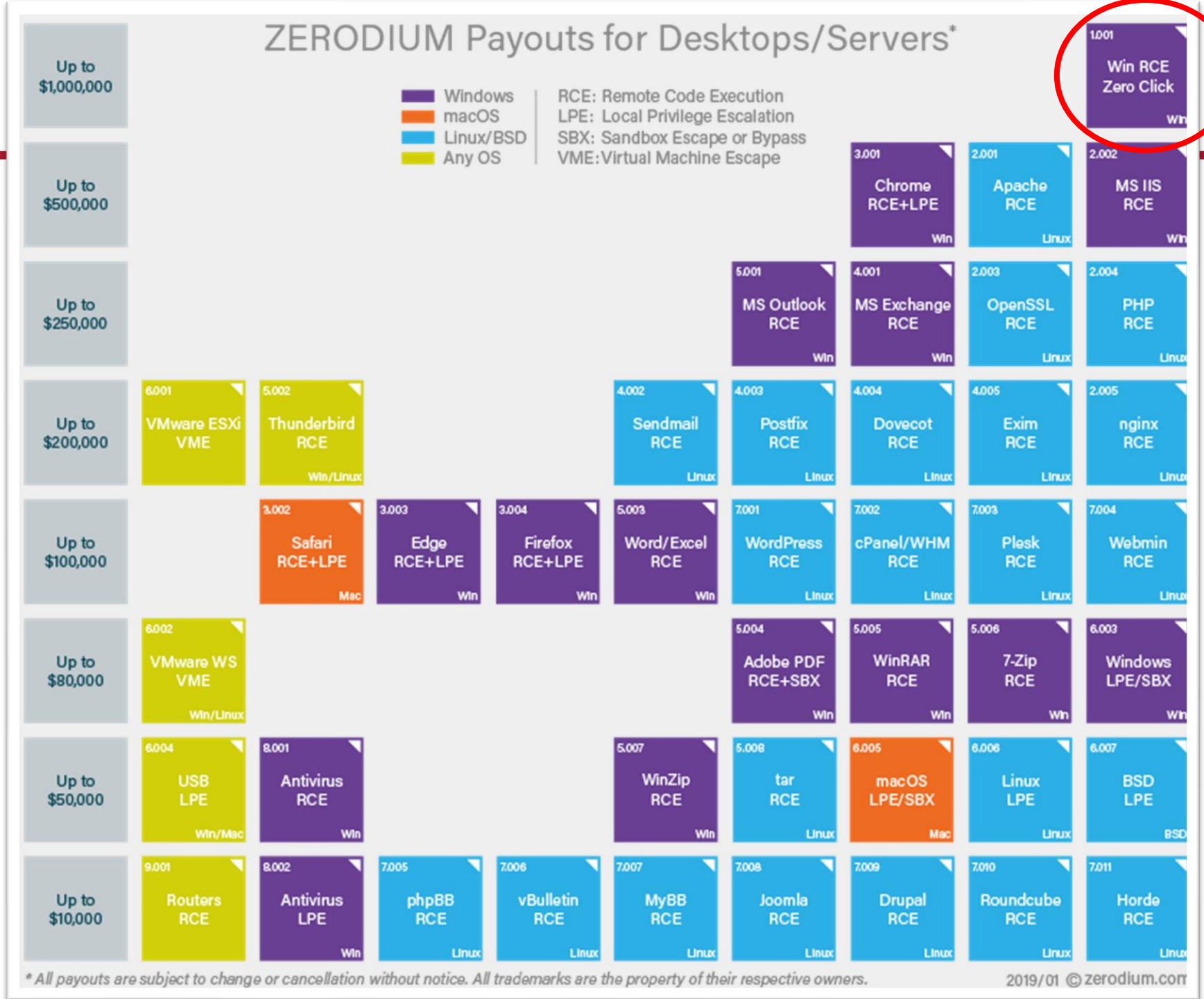
	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Android	Google	OS	414
2	Debian Linux	Debian	OS	360
3	Windows Server 2016	Microsoft	OS	357
4	Windows 10	Microsoft	OS	357
5	Windows Server 2019	Microsoft	OS	351
6	Acrobat Reader Dc	Adobe	Application	342
7	Acrobat Dc	Adobe	Application	342
8	Cpanel	Cpanel	Application	321
9	Windows 7	Microsoft	OS	250
10	Windows Server 2008	Microsoft	OS	248

https://www.cvedetails.com/top-50-products.php?year=2019

Finding vulnerability has become a business

- Bug bounty programs
  - Google Vulnerability Reward Program: up to $31,337
  - Microsoft Bounty Program: up to $100K
  - Apple Bug Bounty program: up to $200K

- Acquiring vulnerabilities
  - Zerodium: up to $2M for iOS, $500K for Android

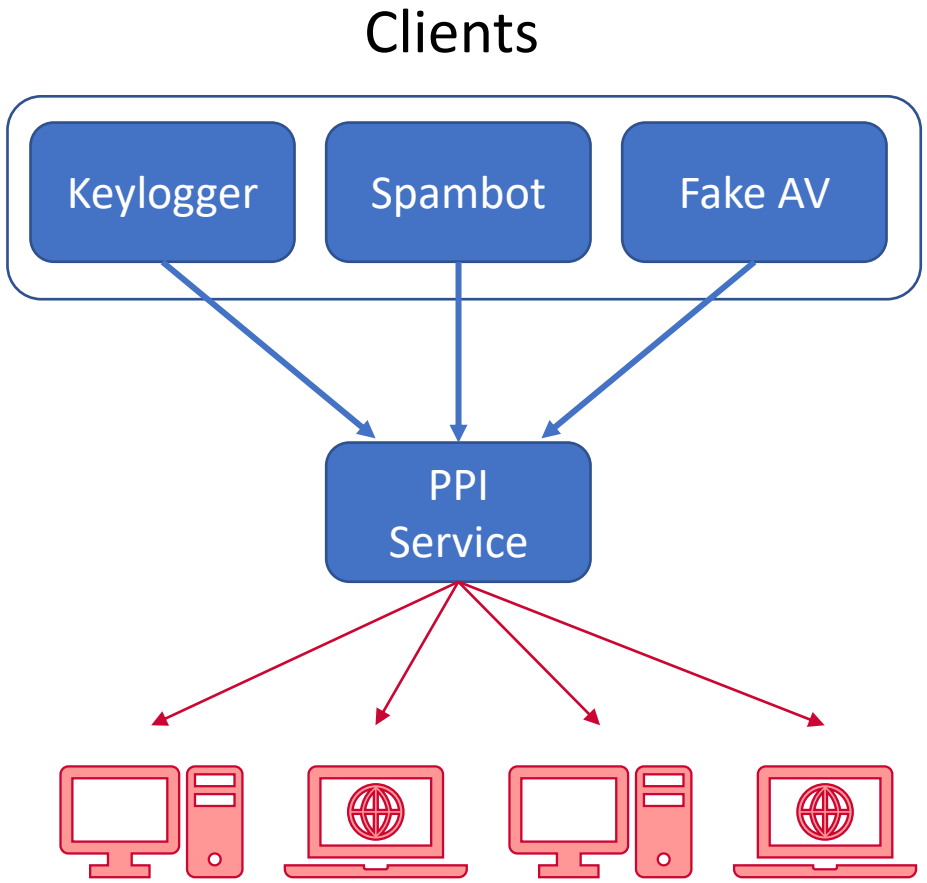

ZERODIUM Payouts for Desktops/Servers*
Windows
macOS
Linux/BSD
Any OS
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
VME: Virtual Machine Escape
Up to $1,000,000
1.001 Win RCE Zero Click Win
Up to $500,000
3.001 Chrome RCE+LPE Win
2.001 Apache RCE Linux
2.002 MS IIS RCE Win
Up to $250,000
5.001 MS Outlook RCE Win
4.001 MS Exchange RCE Win
2.003 OpenSSL RCE Linux
2.004 PHP RCE Linux
Up to $200,000
6.001 VMware ESXi VME
5.002 Thunderbird RCE Win/Linux
4.002 Sendmail RCE Linux
4.003 Postfix RCE Linux
4.004 Dovecot RCE Linux
4.005 Exim RCE Linux
2.005 nginx RCE Linux
Up to $100,000
3.002 Safari RCE+LPE Mac
3.003 Edge RCE+LPE Win
3.004 Firefox RCE+LPE Win
5.003 Word/Excel RCE Win
7.001 WordPress RCE Linux
7.002 cPanel/WHM RCE Linux
7.003 Plesk RCE Linux
7.004 Webmin RCE Linux
Up to $80,000
6.002 VMware WS VME Win/Linux
5.004 Adobe PDF RCE+SBX Win
5.005 WinRAR RCE Win
5.006 7-Zip RCE Win
6.003 Windows LPE/SBX Win
Up to $50,000
6.004 USB LPE Win/Mac
8.001 Antivirus RCE Win
5.007 WinZip RCE Win
5.008 tar RCE Linux
6.005 macOS LPE/SBX Mac
6.006 Linux LPE Linux
6.007 BSD LPE BSD
Up to $10,000
9.001 Routers RCE
8.002 Antivirus LPE Win
7.005 phpBB RCE Linux
7.006 vBulletin RCE Linux
7.007 MyBB RCE Linux
7.008 Joomla RCE Linux
7.009 Drupal RCE Linux
7.010 Roundcube RCE Linux
7.011 Horde RCE Linux
* All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.
2019/01 © zerodium.com

…or even worse: A Marketplace for owned machines

- Pay-per-install (PPI) services

- PPI operation:

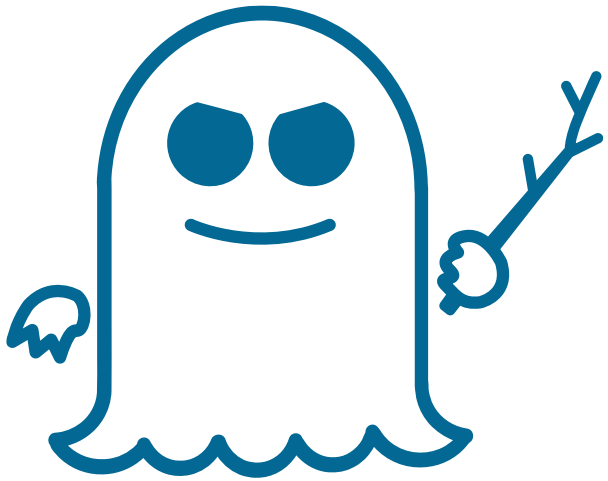
1. Own victim machine
2. Download and install client program
3. Charge client



Cabalerro et al. (www.icir.org/vern/papers/ppi-usesec11.pdf)

Side-channel attacks

- Attacks that are based on implementation of a system
  - Timing attacks
  - Power analysis attacks
  - Electromagnetic attacks
  - Caching attacks



Too many threats…

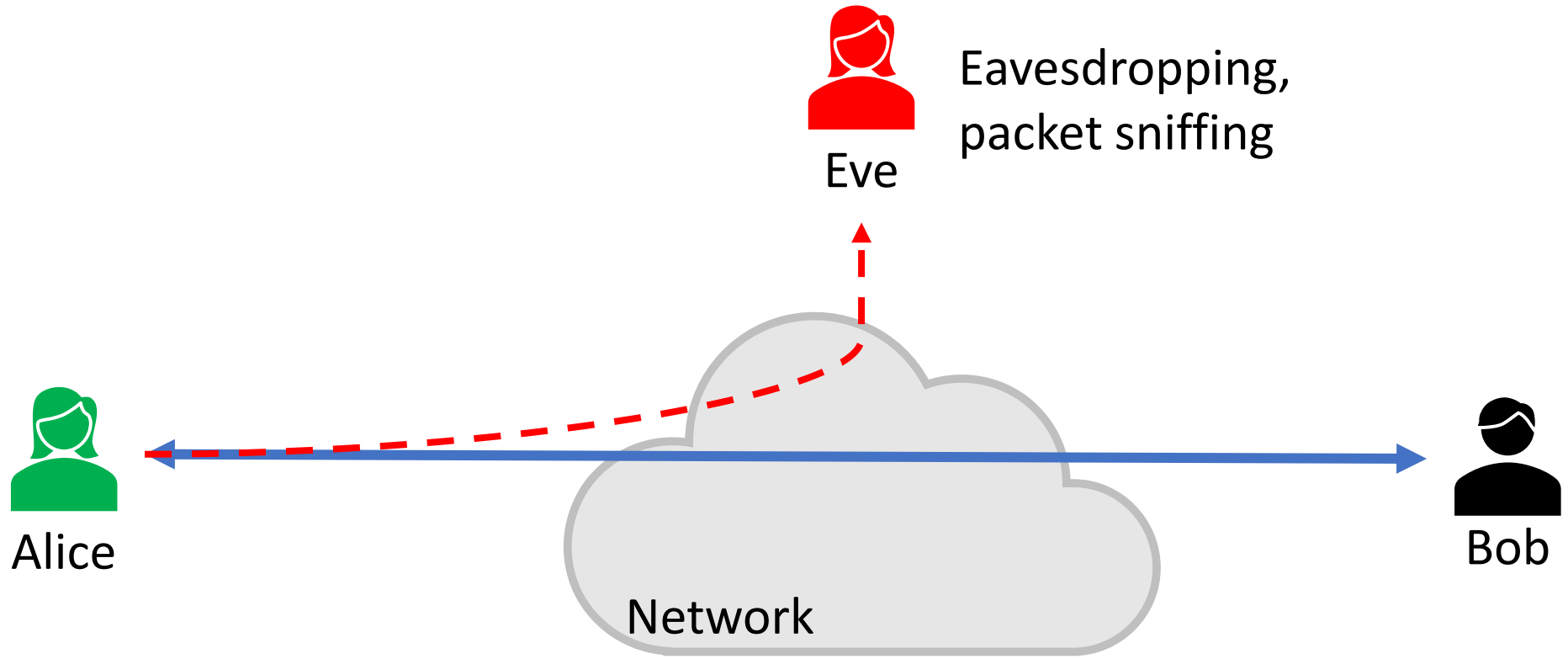
- Consider the Internet
  - Every host, router, middlebox is a potential threat
  - Esp. when they become Zombies

Security Goals

- Common general security goals: “CIA”
  - **C**onfidentiality
  - **I**ntegrity
  - **A**uthenticity
  - Availability

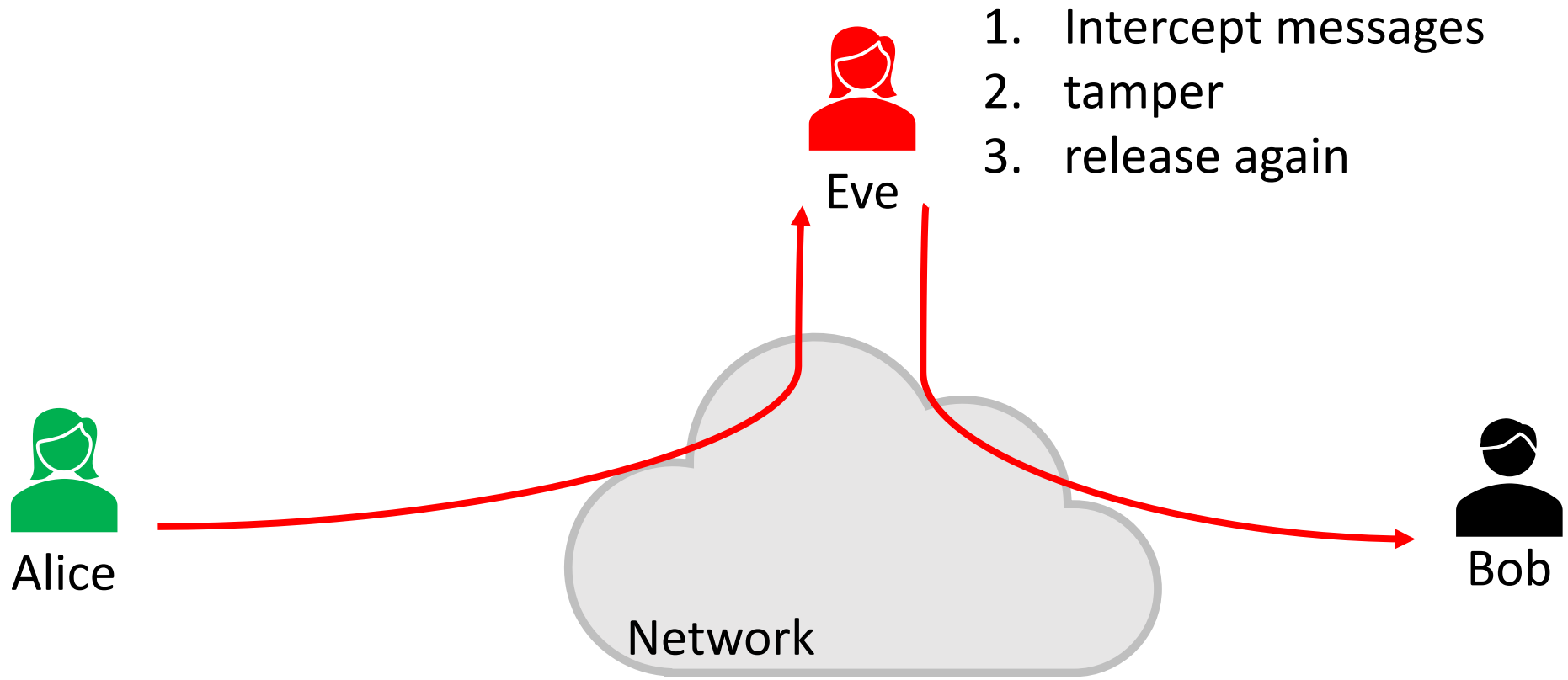
Confidentiality (Privacy)

- Confidentiality is **concealment of information**.



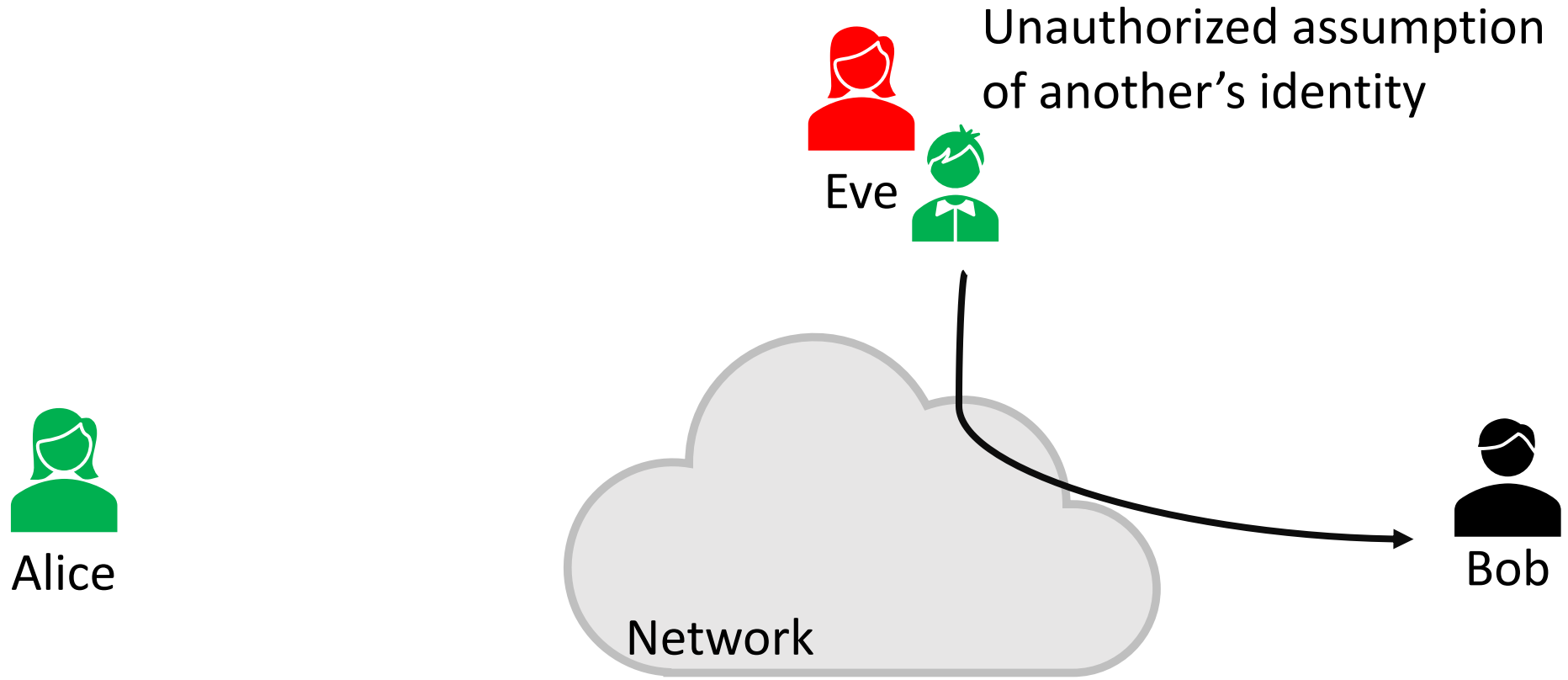
Integrity

- Integrity is **prevention of unauthorized changes.**



Authenticity

- Authenticity is **knowing who you are talking to.**

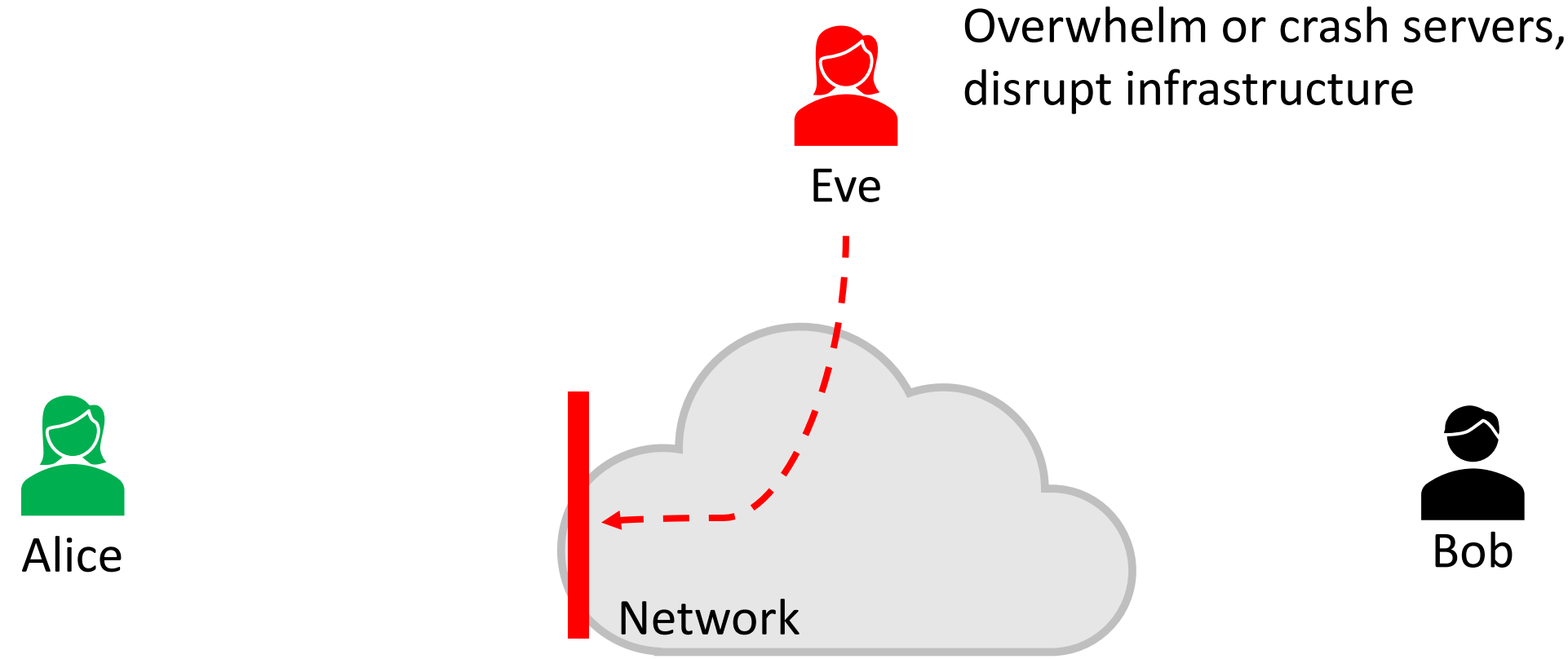


Authenticity

"On the Internet, nobody knows you're a dog."

Availability

- Availability is **ability to use information or resources.**



Technical Enablers

- Cryptography
- Roots of trust
  - Trusted hardware
  - Trusted hypervisor
- Program Analysis/Verification
- (Anomaly) Detection Algorithms

Cryptography Primitives

1. Encryption/Decryption

2. Digital Signatures

3. One-way hash functions

- Applications?

Security Approaches

- Prevention
  - Stop an attack
- Detection
  - Detect an ongoing or past attack
- Incident Response
  - Respond to attacks

Attack Phases

- Reconnaissance
- Scanning & Enumeration
- Gaining Access (or Exploitation)
- Maintaining Access (or Persistence)
- Covering Tracks

Attacker Asymmetric Advantage

Attacker Asymmetric Advantage

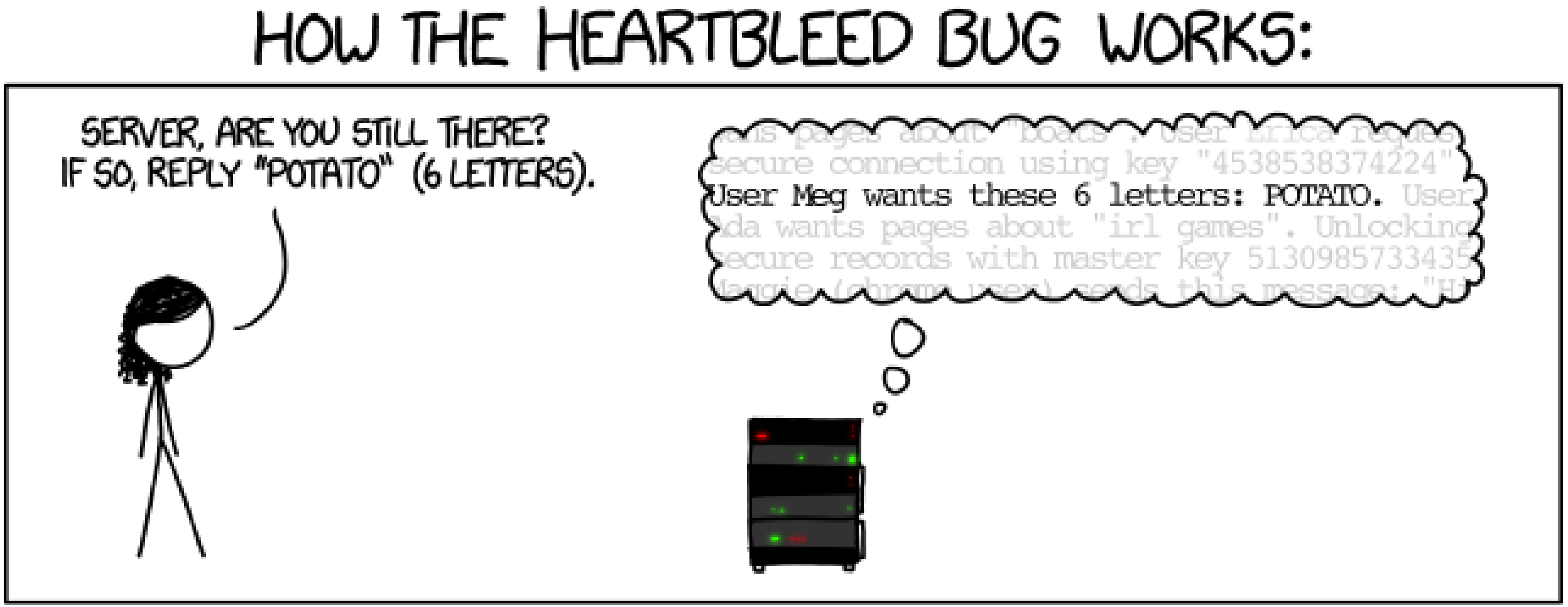
- Attacker only needs to win in one place
- Defender’s response: Defense at every layer

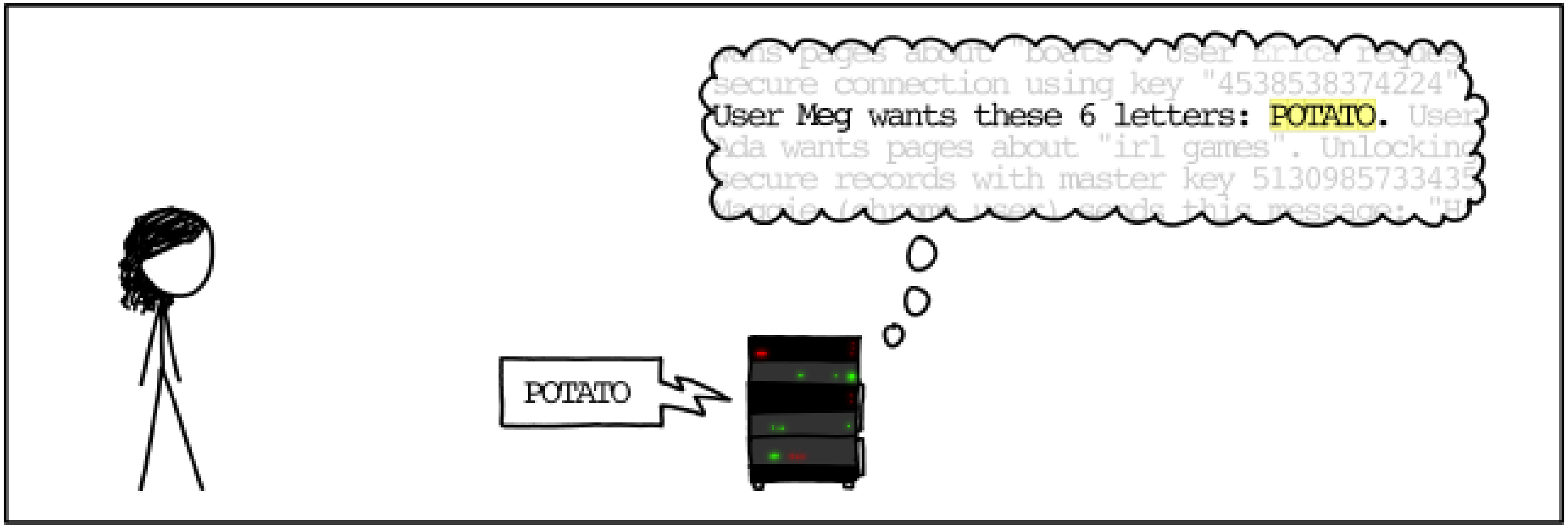
Whole System is Critical

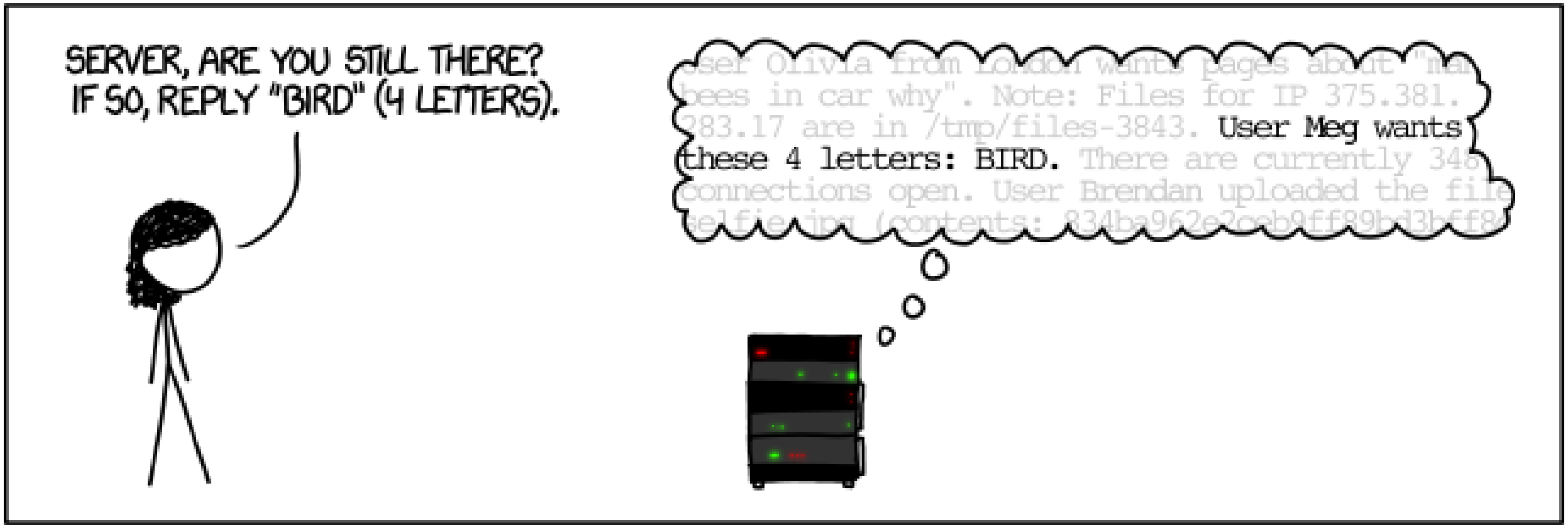
- Securing a system involves a whole-system view
  - Cryptography
  - Implementation
  - People
  - Physical security
  - Everything in between

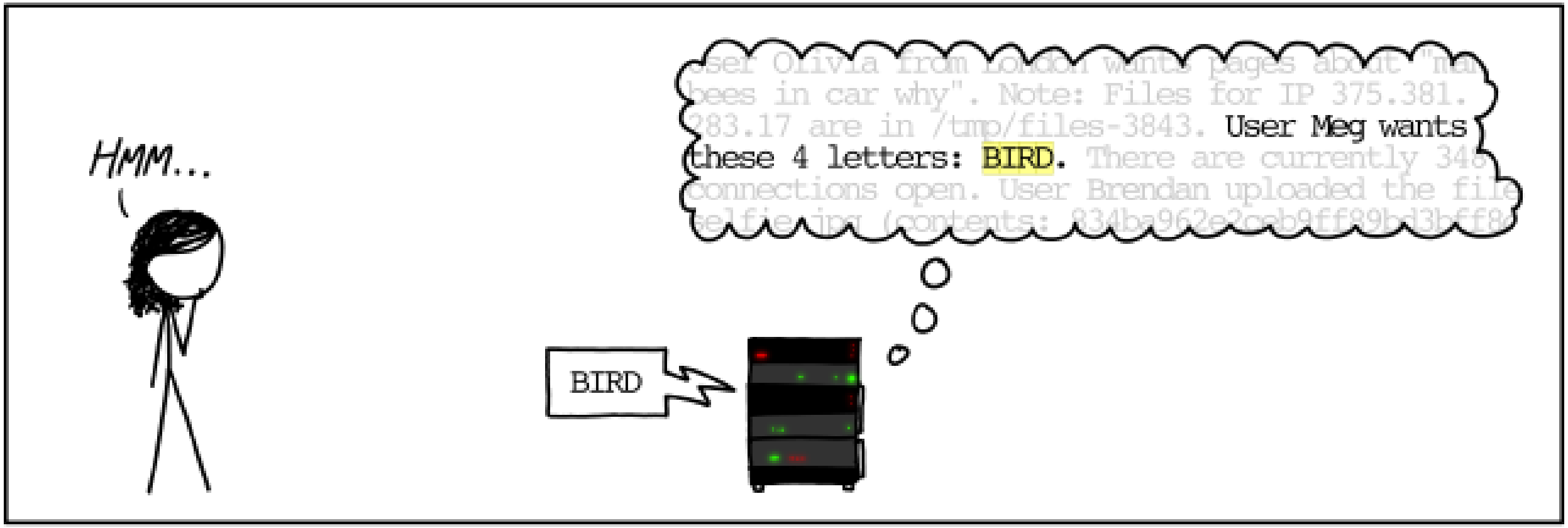
- No reason to attack the strongest part of a system if you can walk right around it.

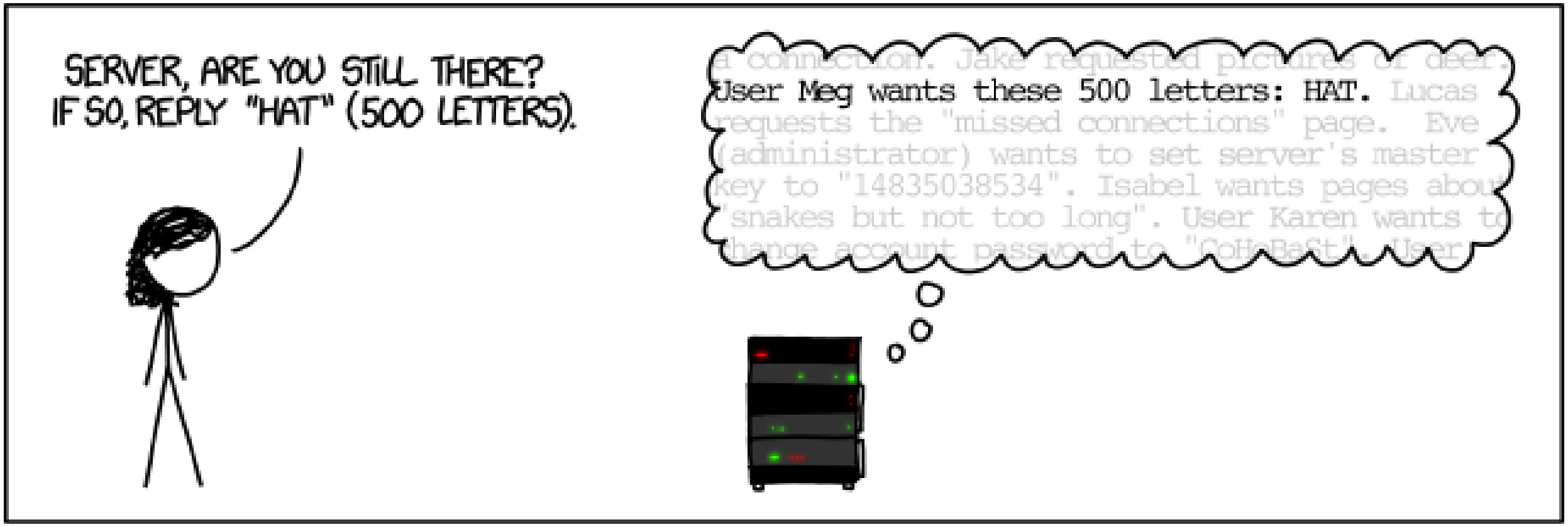
Example 1 – Heartbleed (Why Crypto != Security)

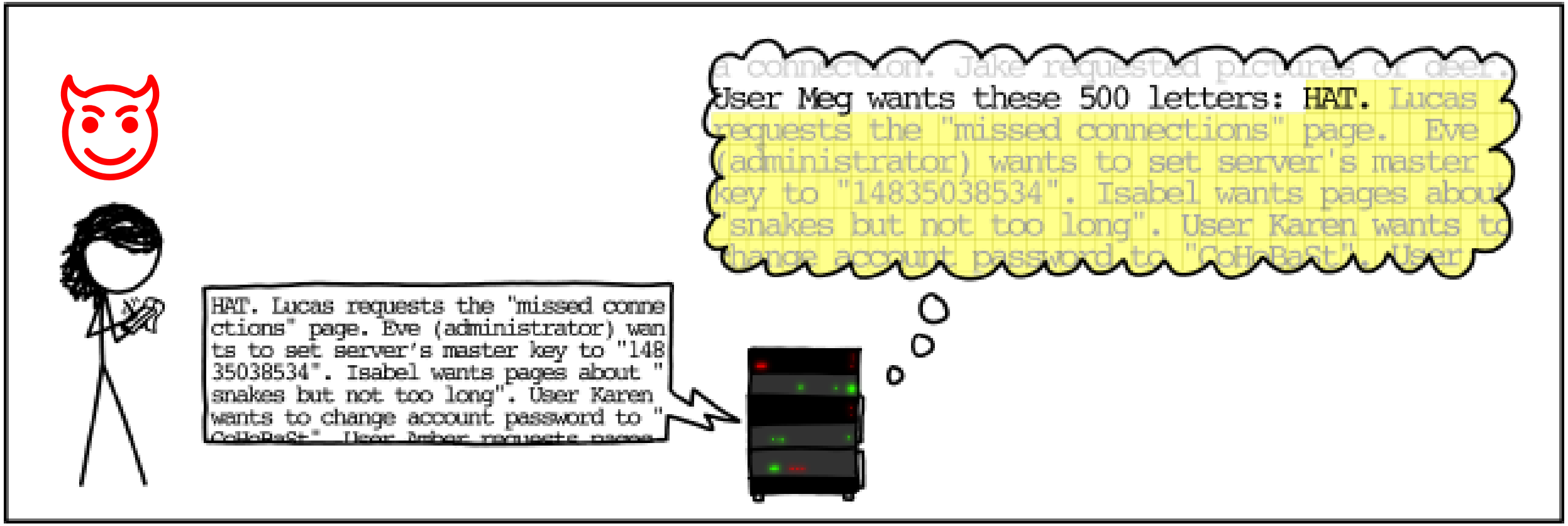



User Meg wants these 6 letters: POTATO.
POTATO


SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
these 4 letters: BIRD.


HMM...
User Olivia from London wants pages about "man 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


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
User Meg wants these 500 letters: HAT.


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 "
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

Example 2 – Linux Backdoor

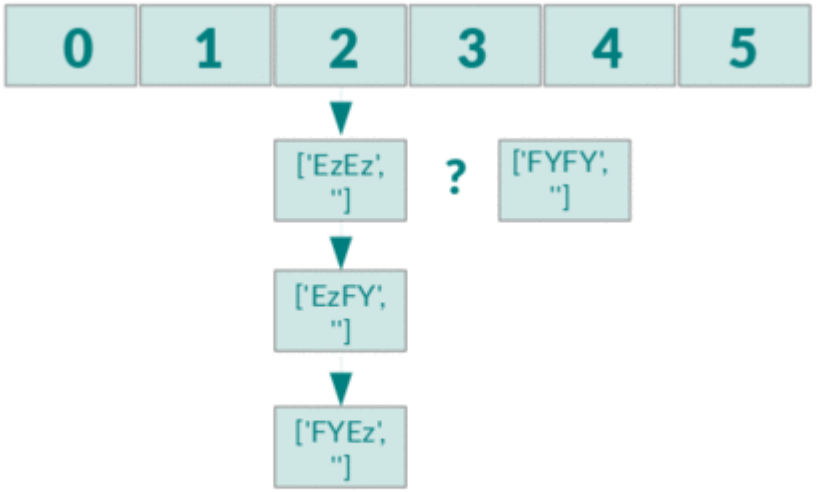
```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
        retval = -EINVAL;
```

- Was never pushed to Linux master copy in BitKeeper
- Was noticed by a developer in CVS

https://lwn.net/Articles/57137/
https://freedom-to-tinker.com/2013/10/09/the-linux-backdoor-attempt-of-2003/

Example 3 – PHP Hash Collision DoS

- PHP stores arrays using hash tables
- If an attacker controls the input in a specific way, all the inputs will collide



- number of elements to traverse is quadratic (for every insertion)

→ more CPU cycles (3000X delay compared to normal operation)

→ Resulting in a DoS attack

https://fahrplan.events.ccc.de/congress/2011/Fahrplan/attachments/2007_28C3_Effective_DoS_on_web_application_platforms.pdf

PHP Hash Collision DoS

- How did PHP solve this problem?
  - Set max. number of inputs

- Is this a good solution?
- What is the root cause of the attack?
- How do other languages address this vulnerability?

Todo

- Read and understand the syllabus
- Sign the Ethics Form
- Get to know your classmates, and form project groups
- Start thinking about project ideas

Questions?