Welcome to CMSC 334

Security Principles

Introduction to Computer Security

1. Introductions
2. Course logistics
3. Motivation: Why study computer security?
4. Goals of security

Introduction to Computer Security

1. Introductions
2. Course logistics
3. Motivation: Why study computer security?
4. Goals of security

Dr. David Balash

Faculty page: https://cs.richmond.edu/faculty/dbalash

Homepage: https://davidbalash.github.io

Professor Balash

"Ba-lish"

He/Him

- BS in computer engineering Iowa State
- Two-decade career as a software engineer
- MS and PhD in computer science from GW
- Research: Computer S&P

Dr. David Balash

Things I like

- Education/Learning
- Hiking
- Cycling
- Guitars
- Board games
- Programming
- Cats

Ask me anything

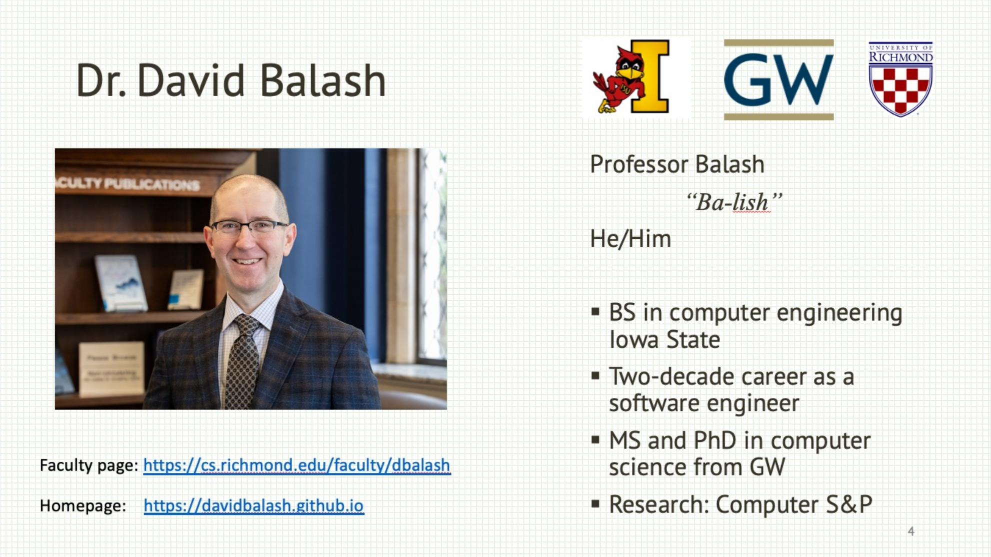
Assignment 1

Task: Create a personal introduction slide and post it to the introductions channel on the course Slack workspace

Due: Friday January 19th

Points: 5

Be Creative



Dr. David Balash

Professor Balash

"Ba-lish"

He/Him

- BS in computer engineering Iowa State
- Two-decade career as a software engineer
- MS and PhD in computer science from GW
- Research: Computer S&P

Faculty page: https://cs.richmond.edu/faculty/dbalash

Homepage: https://davidbalash.github.io

4

Name

Photo

Pronunciation

Pronouns

Personal Introduction

Student Introductions

1. Name
2. Pronouns
3. Major(s)
4. Class year
5. What topics in computer security interest you the most?

Introduction to Computer Security

1. Introductions
2. Course logistics
3. Motivation: Why study computer security?
4. Goals of security

Where all course information can be found

`https://cmsc334-s24.github.io`

How to communicate with me

Course Slack Workspace: https://cmsc334-s24.slack.com

After class or in office hours - 223 Jepson Hall

Tue 4:30PM - 5:30PM

Fri 3:00PM - 5:00PM

and by appointment https://calendly.com/davidbalash

Email: david.balash@richmond.edu

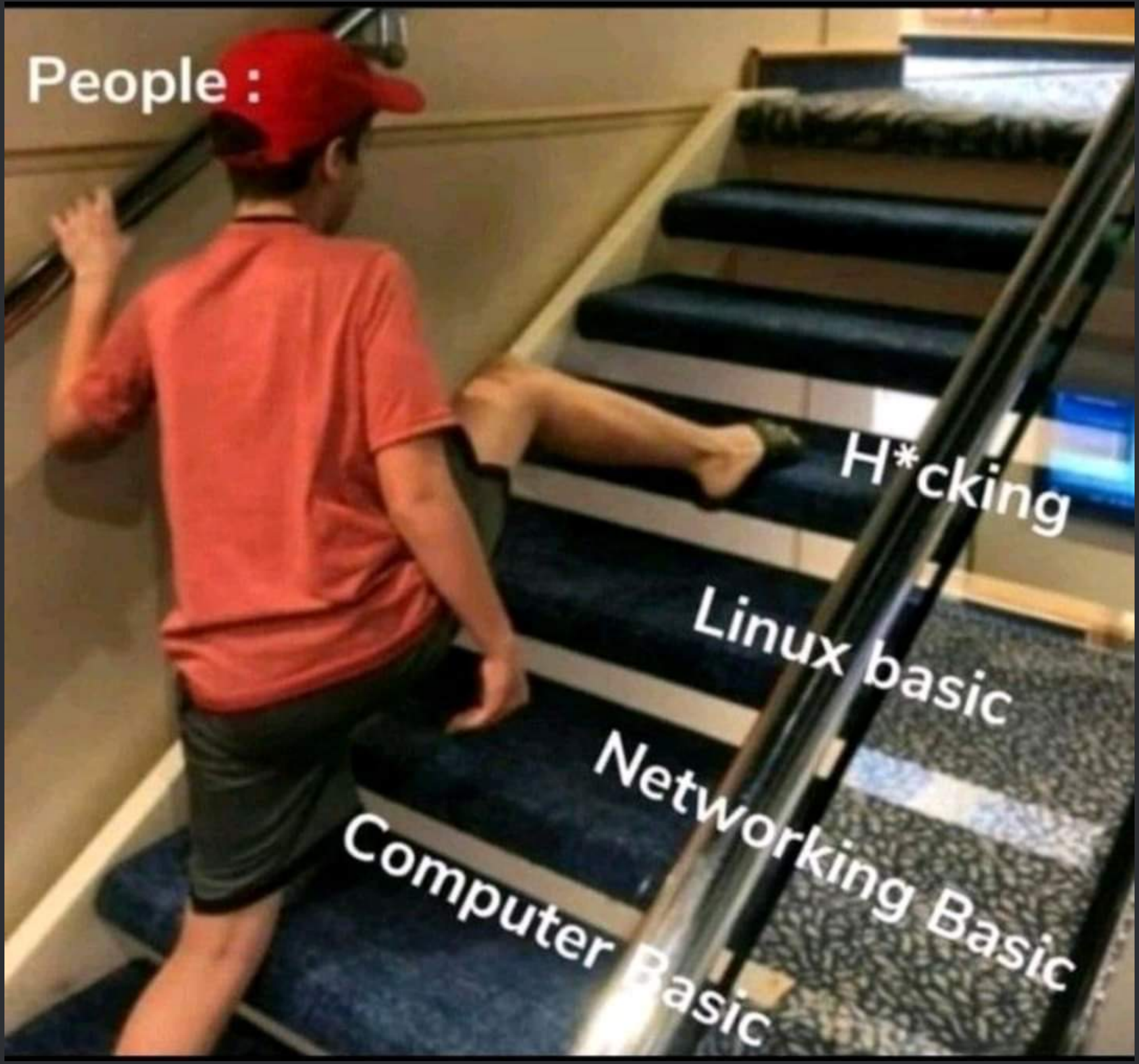
Computer Security is interdisciplinary

Draws on all areas of Computer Science

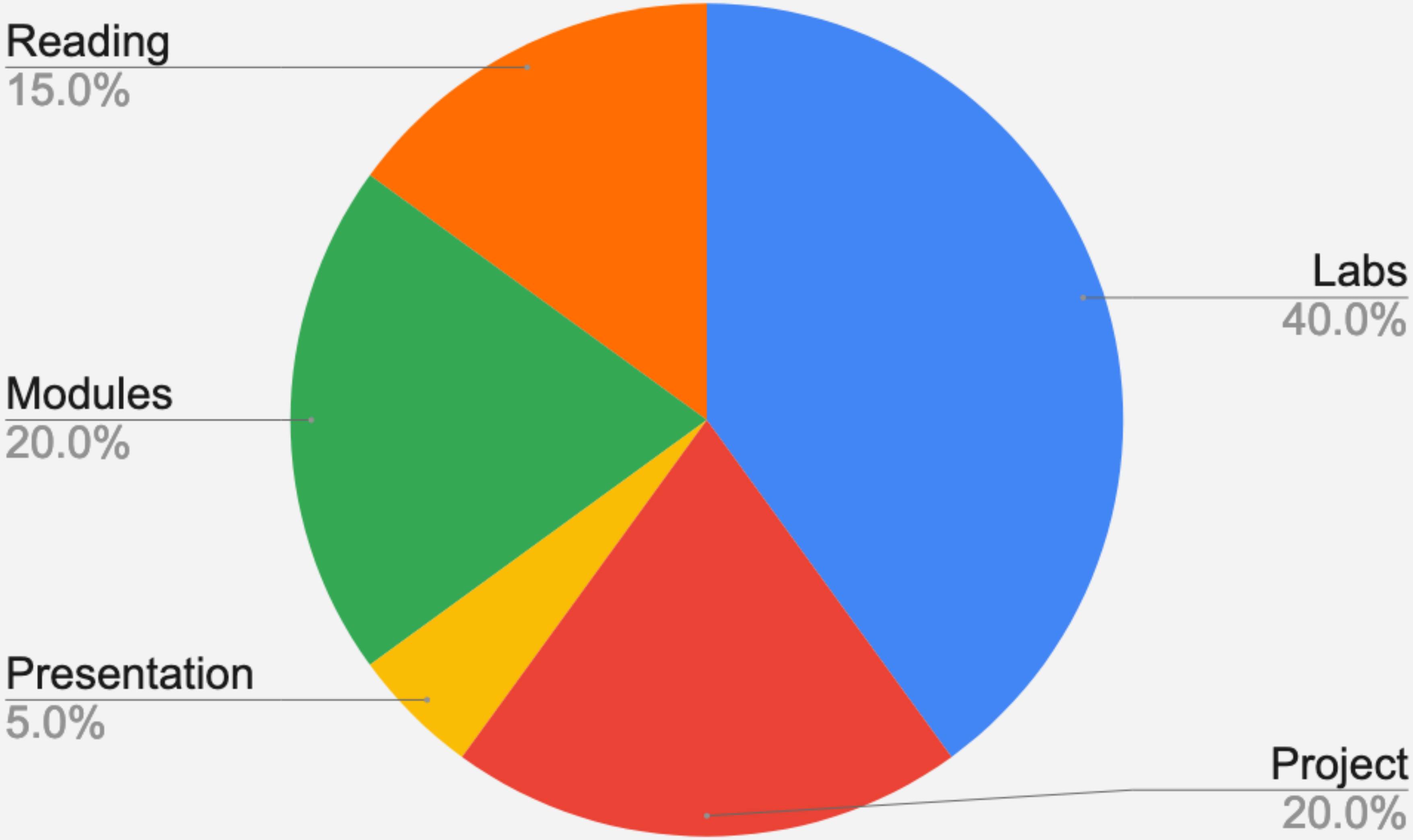
- Theory (especially cryptography)
- Networking
- Programming languages/compilers
- Operating systems
- Databases
- Machine learning / AI
- Computer architecture / hardware
- HCI, psychology

Course Outline

1. Introduction to Computer Security
    - What are some general goals when thinking about computer security?
2. Securing Software and Systems
    - How do attackers exploit insecure software? How do we defend against these attacks?
3. Applied Cryptography
    - How do we securely send information over an insecure channel?
4. Web Security
    - What are some attacks on the web, and how do we defend against them?
5. Network Security
    - What are some attacks on the Internet, and how do we defend against them?
6. Usable Security and Privacy
    - How do human factors affect security? How do we protect our privacy online?


People :
H*cking
Linux basic
Networking Basic
Computer Basic

Coursework and Grading



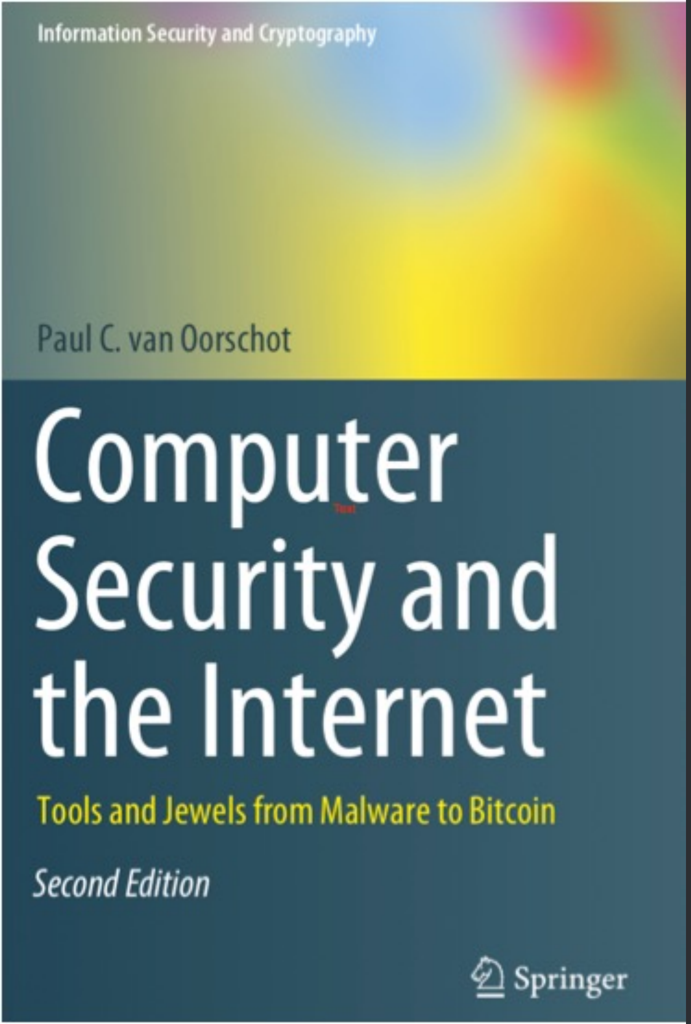
Course Textbook

Available as pdfs on the authors website:

https://people.scs.carleton.ca/~paulv/toolsjewels.html

We will read and discuss the book using:

hypothes.is



Information Security and Cryptography

Paul C. van Oorschot

Computer Security and the Internet

Tools and Jewels from Malware to Bitcoin

Second Edition

Springer

Ethics

- **Responsible Use of Knowledge**
    - Apply skills for positive, lawful purposes only
    - Avoid malicious activities or unauthorized access
- **Respect for Privacy**
    - Honor the confidentiality of personal and institutional data
    - Refrain from unauthorized data snooping or surveillance
- **Legal Compliance**
    - Adhere to all relevant laws and school policies
    - Understand the legal implications of cybersecurity actions

Principles of Responsible Disclosure

1. Ethical Reporting
   - Report vulnerabilities to the affected organization before publicizing

2. Collaborative Resolution
   - Allow reasonable time for the issue to be fixed and collaborate if necessary

3. Responsible Public Disclosure
   - Release information responsibly post-fix to inform and educate, respecting legal and ethical guidelines

Introduction to Computer Security

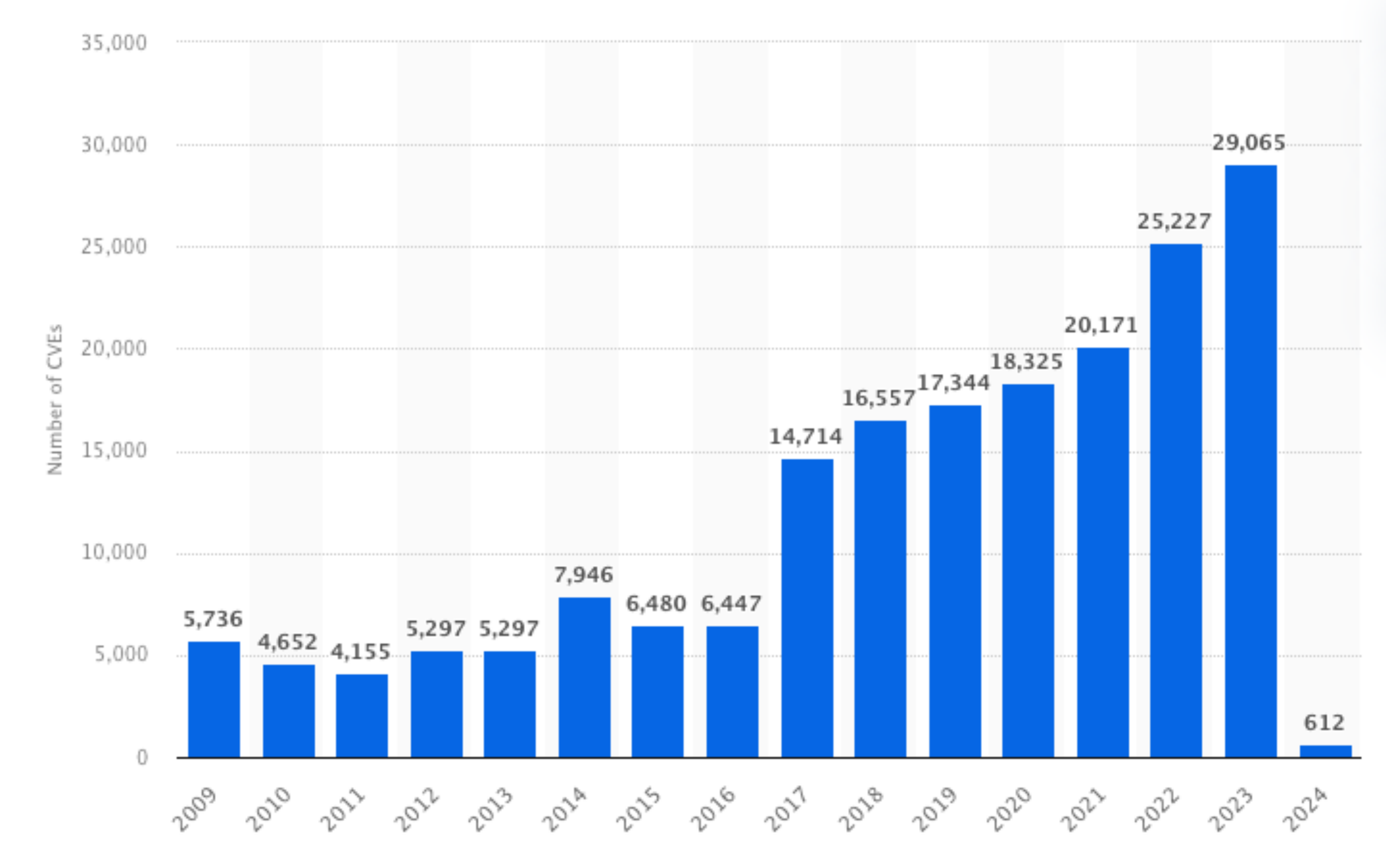
1. Introductions
2. Course logistics
3. Motivation: Why study computer security?
4. Goals of security

> The business world comes to a standstill when Internet service is disrupted. Our critical infrastructure, from power plants and electricity grids to water supply and financial systems, is dependent on computer hardware, software and the Internet. Implicitly we expect, and need, security and dependability.
>
> -- Paul C. Van Oorschot

Computer security is important

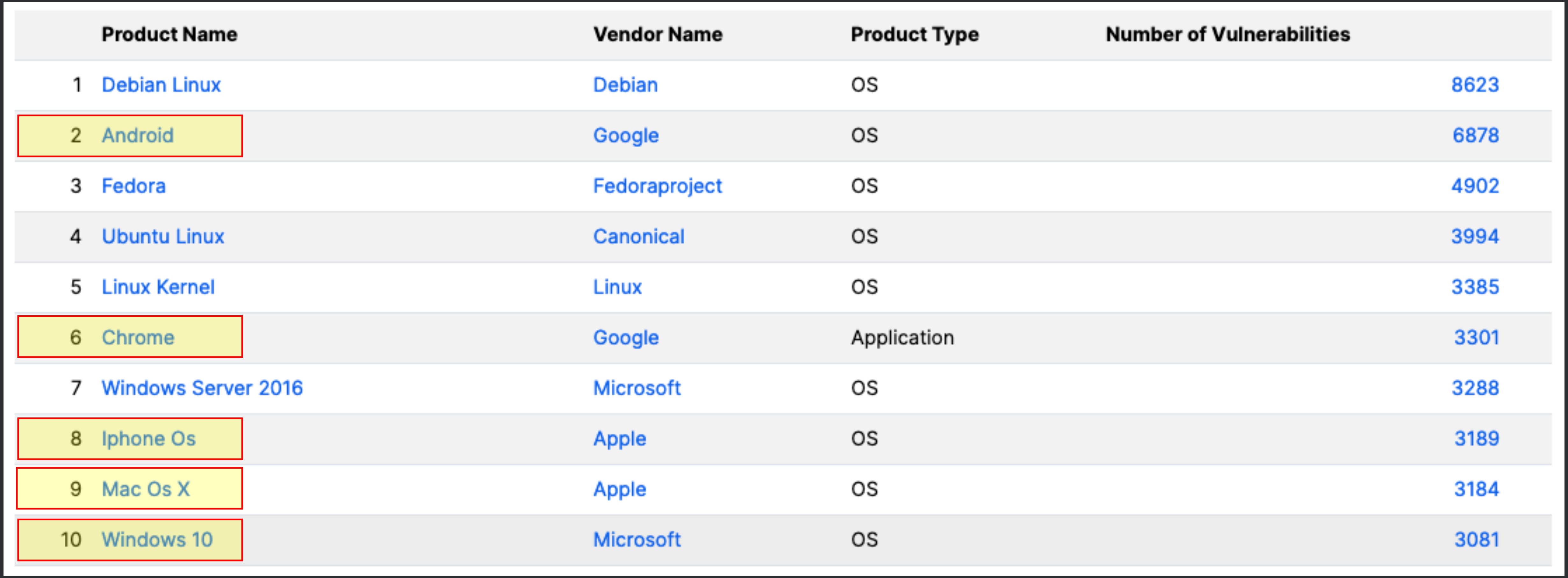
1. Number of vulnerabilities and attacks are increasing
2. Cybercrime
3. Cyberwarfare
4. Increasing government and academic interest
5. Just read the news…

Common security vulnerabilities and exposures (CVEs) worldwide from 2009 to 2024 YTD



https://www.statista.com/statistics/500755/worldwide-common-vulnerabilities-and-exposures/

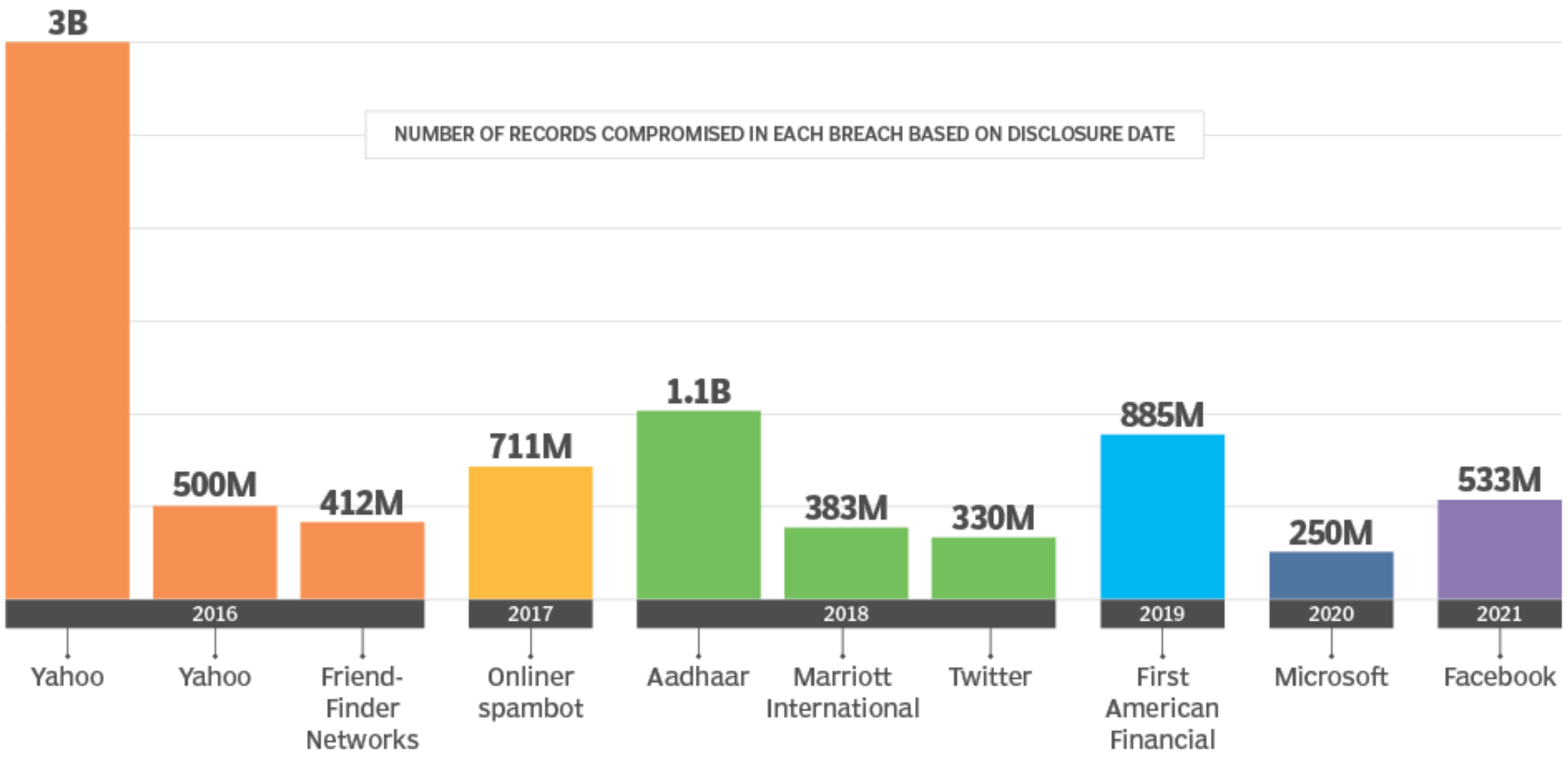
Top 10 products by total number of distinct security vulnerabilities 2014 to 2024



	Product Name	Vendor Name	Product Type	Number of Vulnerabilities
1	Debian Linux	Debian	OS	8623
2	Android	Google	OS	6878
3	Fedora	Fedoraproject	OS	4902
4	Ubuntu Linux	Canonical	OS	3994
5	Linux Kernel	Linux	OS	3385
6	Chrome	Google	Application	3301
7	Windows Server 2016	Microsoft	OS	3288
8	Iphone Os	Apple	OS	3189
9	Mac Os X	Apple	OS	3184
10	Windows 10	Microsoft	OS	3081

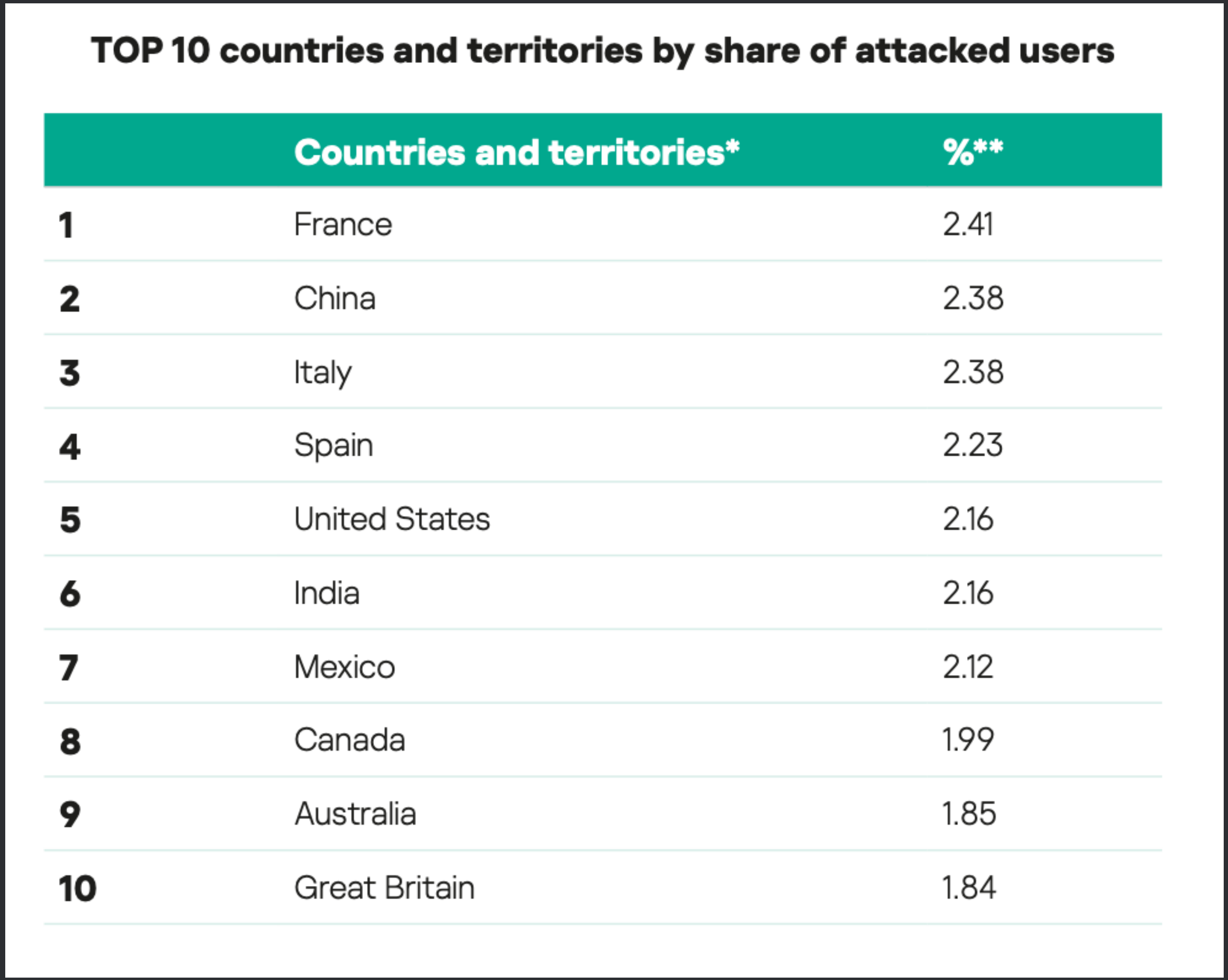
https://www.cvedetails.com/top-50-products.php

10 of the biggest data breaches in history



https://www.techtarget.com/whatis/34-Cybersecurity-Statistics-to-Lose-Sleep-Over-in-2020

It is a global problem

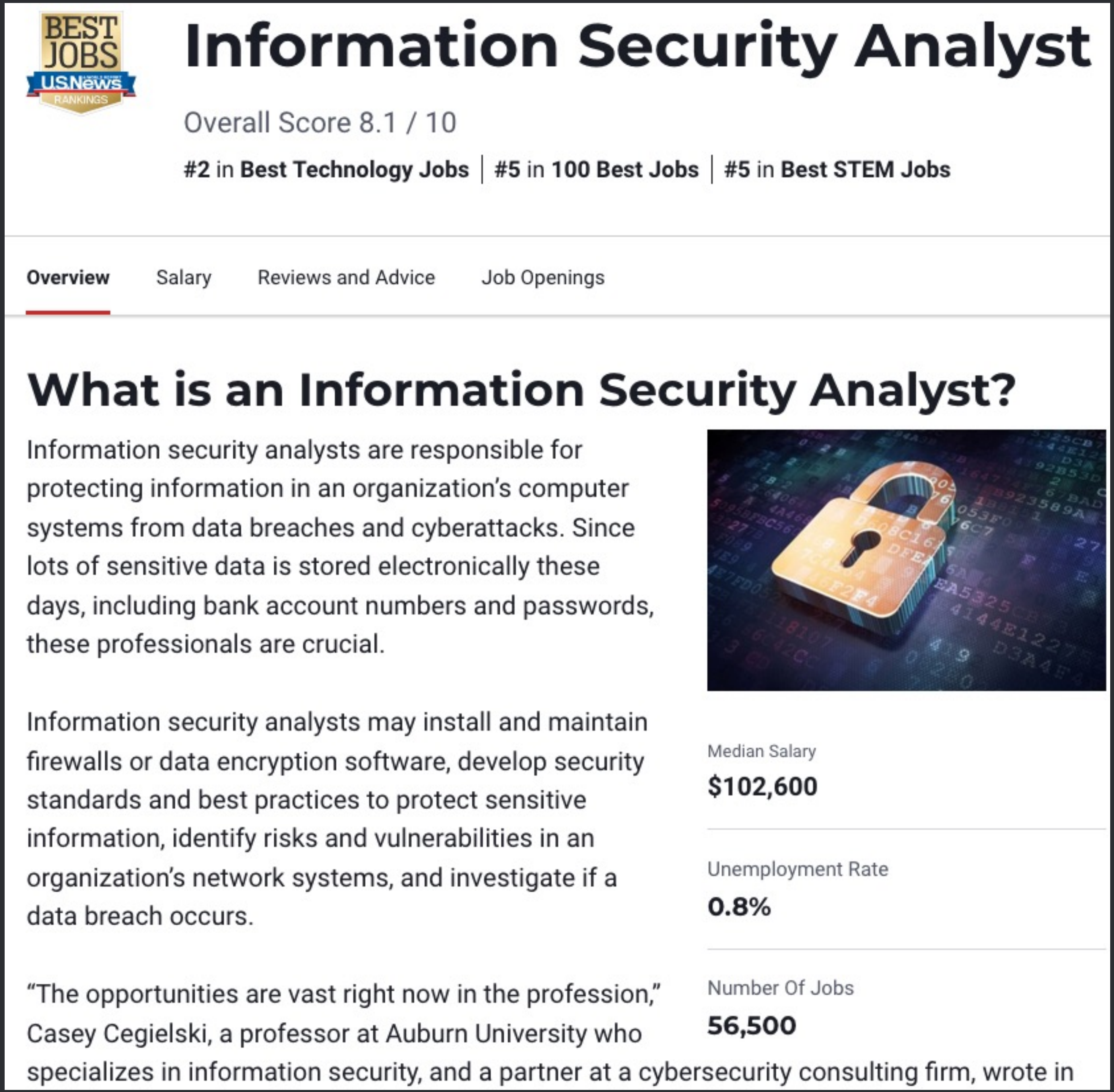


TOP 10 countries and territories by share of attacked users

	Countries and territories*	%**
1	France	2.41
2	China	2.38
3	Italy	2.38
4	Spain	2.23
5	United States	2.16
6	India	2.16
7	Mexico	2.12
8	Canada	1.99
9	Australia	1.85
10	Great Britain	1.84

Source: Kaspersky Security Bulletin 2023

Career opportunities



Information Security Analyst

Overall Score 8.1 / 10

#2 in **Best Technology Jobs** | **#5** in **100 Best Jobs** | **#5** in **Best STEM Jobs**

What is an Information Security Analyst?

Information security analysts are responsible for protecting information in an organization's computer systems from data breaches and cyberattacks. Since lots of sensitive data is stored electronically these days, including bank account numbers and passwords, these professionals are crucial.

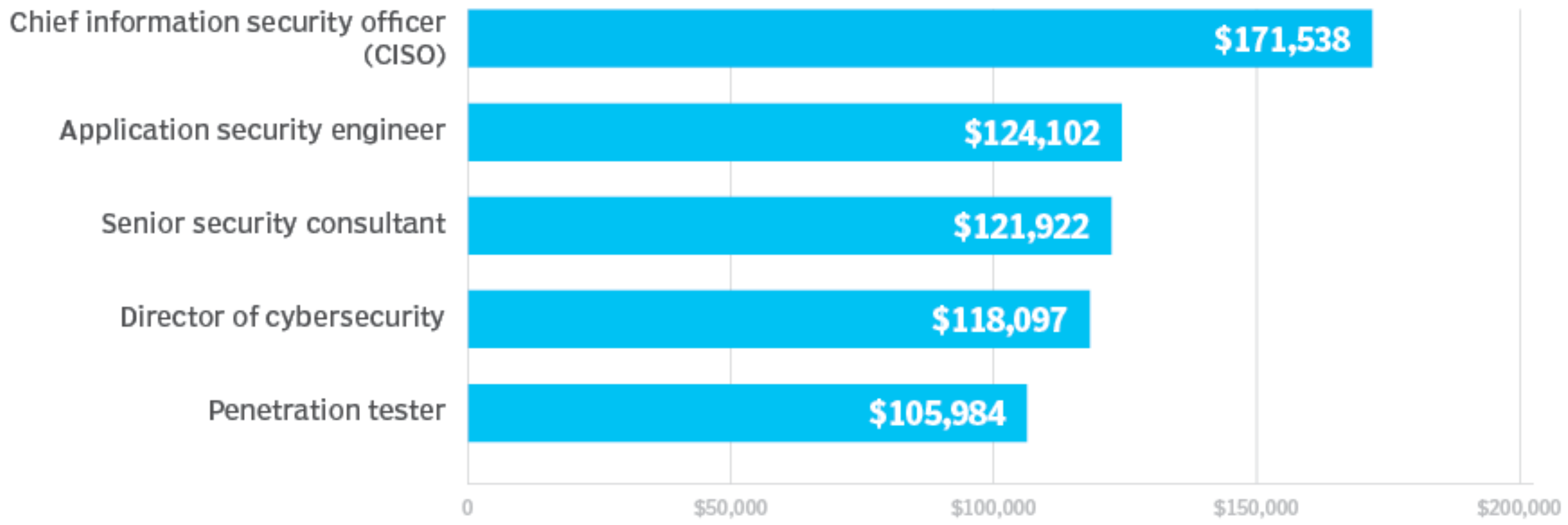
Information security analysts may install and maintain firewalls or data encryption software, develop security standards and best practices to protect sensitive information, identify risks and vulnerabilities in an organization's network systems, and investigate if a data breach occurs.

"The opportunities are vast right now in the profession," Casey Cegielski, a professor at Auburn University who specializes in information security, and a partner at a cybersecurity consulting firm, wrote in

Median Salary	**$102,600**
Unemployment Rate	**0.8%**
Number Of Jobs	**56,500**

https://money.usnews.com/careers/best-jobs/information-security-analyst

Average annual salary for cybersecurity positions



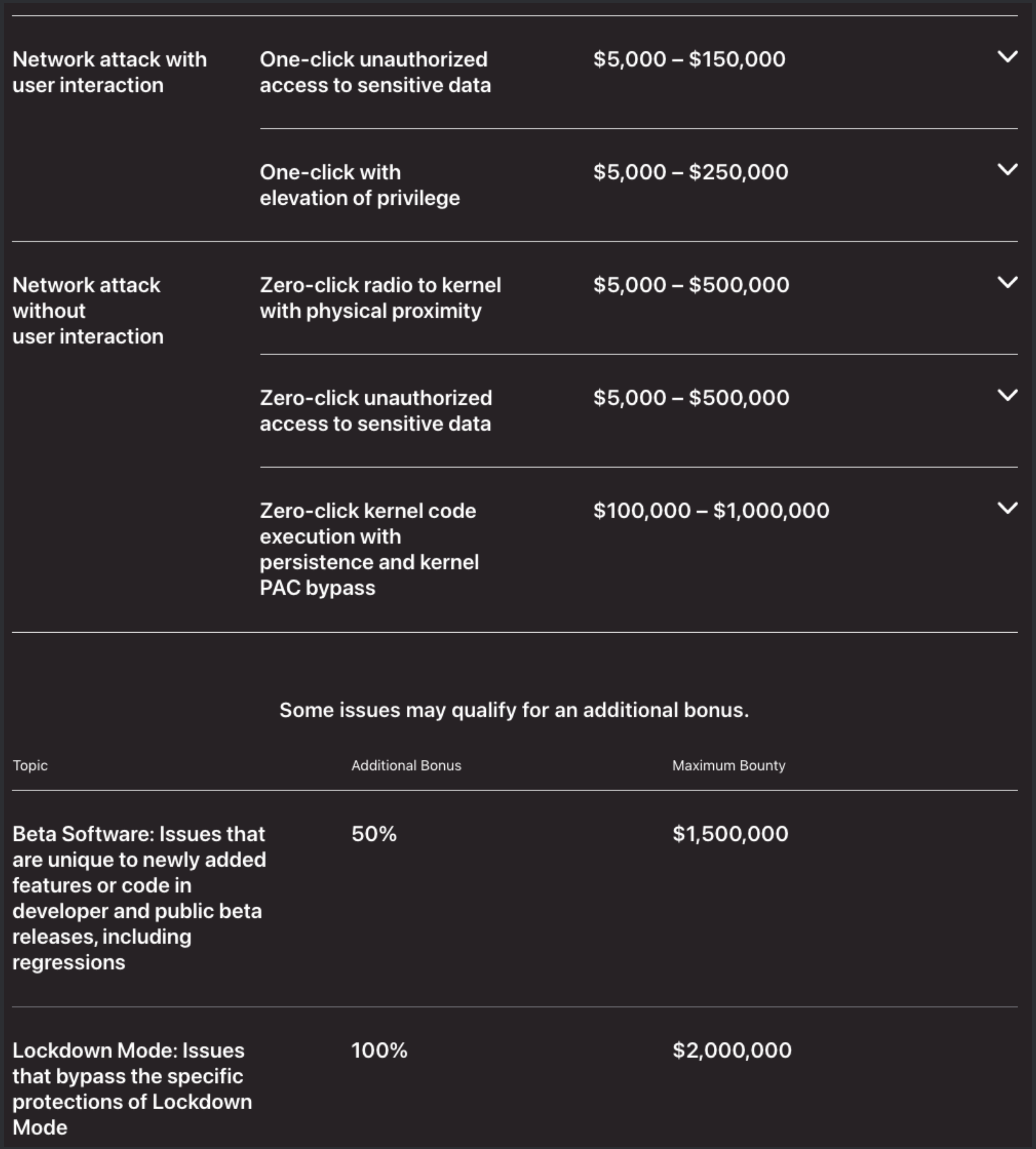
https://www.techtarget.com/whatis/34-Cybersecurity-Statistics-to-Lose-Sleep-Over-in-2020

Marketplace for vulnerabilities

Bug bounty programs

- Pwn2Own competition:	up to **$25,000**
- Google Vulnerability Reward Program:	up to **$31,337**
- Microsoft Bounty Program:	up to **$250,000**
- Apple Bug Bounty program:	up to **$2,000,000**

Apple Security Bounty Categories



Network attack with user interaction	One-click unauthorized access to sensitive data	$5,000 – $150,000
	One-click with elevation of privilege	$5,000 – $250,000
Network attack without user interaction	Zero-click radio to kernel with physical proximity	$5,000 – $500,000
	Zero-click unauthorized access to sensitive data	$5,000 – $500,000
	Zero-click kernel code execution with persistence and kernel PAC bypass	$100,000 – $1,000,000

Some issues may qualify for an additional bonus.

Topic	Additional Bonus	Maximum Bounty
Beta Software: Issues that are unique to newly added features or code in developer and public beta releases, including regressions	50%	$1,500,000
Lockdown Mode: Issues that bypass the specific protections of Lockdown Mode	100%	$2,000,000

https://security.apple.com/bounty/categories/

Introduction to Computer Security

1. Introductions
2. Course logistics
3. Motivation: Why study computer security?
4. Goals of security

What is computer security?

The art and science of protecting computer-related assets from unauthorized actions and their consequences

What is computer security?

Achieving some goal in the presence of an adversary

What is computer security?

Achieving some goal in the presence of an **adversary**

What is computer security?

Achieving some goal in the presence of an **attacker**

hacker n.

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities.
2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
3. An expert at a particular program, or one who frequently does work using it or on it; as in `a Unix hacker'.
4. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.
5. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

The New Hacker's Dictionary by Eric S. Raymond and Guy L. Steele

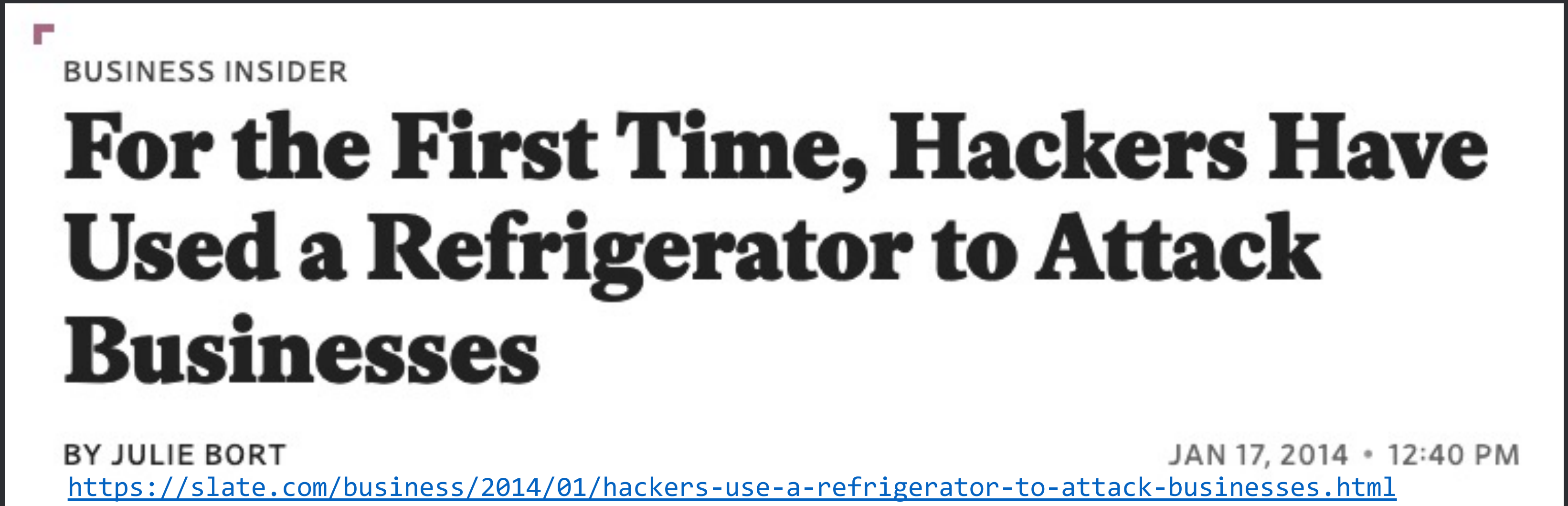
hacker n.

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities.
2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
3. An expert at a particular program, or one who frequently does work using it or on it; as in `a Unix hacker'.
4. An expert or enthusiast of any kind. One might be an astronomy hacker, for example.
5. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

The New Hacker’s Dictionary by Eric S. Raymond and Guy L. Steele

What can be attacked by our adversary?

Everything!

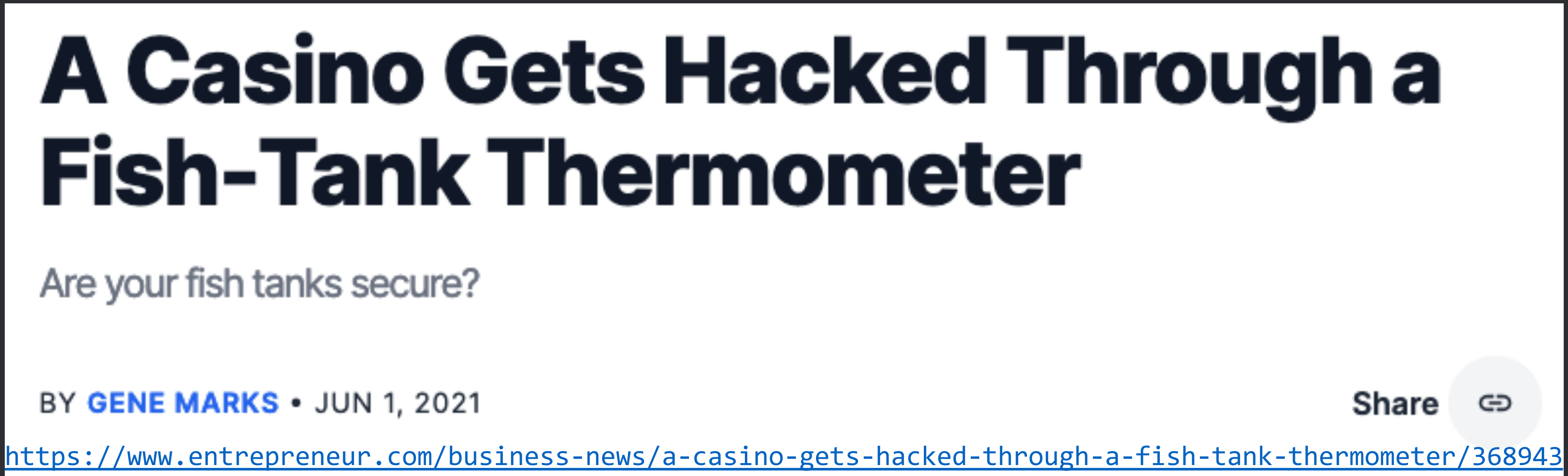


BUSINESS INSIDER

For the First Time, Hackers Have Used a Refrigerator to Attack Businesses

BY JULIE BORT JAN 17, 2014 • 12:40 PM

https://slate.com/business/2014/01/hackers-use-a-refrigerator-to-attack-businesses.html



A Casino Gets Hacked Through a Fish-Tank Thermometer

Are your fish tanks secure?

BY GENE MARKS • JUN 1, 2021 Share

https://www.entrepreneur.com/business-news/a-casino-gets-hacked-through-a-fish-tank-thermometer/368943

Why do people attack systems?

High-level plan for thinking about security

1. **Policy:** the goal you want to achieve

   Example: only the office manager should be able to read the payroll file

2. **Threat Models:** assumptions about what the attacker can do

   Example: attacker can guess passwords, but has no access to password file

3. **Mechanisms:** tools/settings that allow a system to uphold policy

   Examples: user accounts, passwords, file permissions, and encryption

Policy: the goal you want to achieve

Common Goals

1. **Confidentiality:** information is accessible only to those authorized

2. **Integrity:** data remains unaltered, except by those authorized

3. **Availability:** information and services remain accessible for use

Threat Models: what the attacker can do

Adversary Attributes

1. **Objectives:** the target of the adversary

2. **Methods:** the attack techniques, or types of attacks

3. **Capabilities:** computing resources, skills, knowledge, opportunity

Mechanisms: tools/settings

Types of Mechanisms

1. **Physical Security:** Protecting hardware and facilities
2. **Software-Based Protections:** Anti-virus, firewalls, and encryption
3. **Procedures:** User authentication protocols, security training
4. **Network Security:** Secure network protocols, intrusion detection

Why is computer security a hard problem?

1. Defender-attacker asymmetry
   - Negative goal
   - Defenders must protect all possible attack points; attackers only need exploit one

2. No rule book
   - Difficult to think of all possible attacks

3. Realistic threat models are open-ended
   - Intelligent and adaptive adversaries

4. Smart attackers jump the lowest bar or break the weakest link

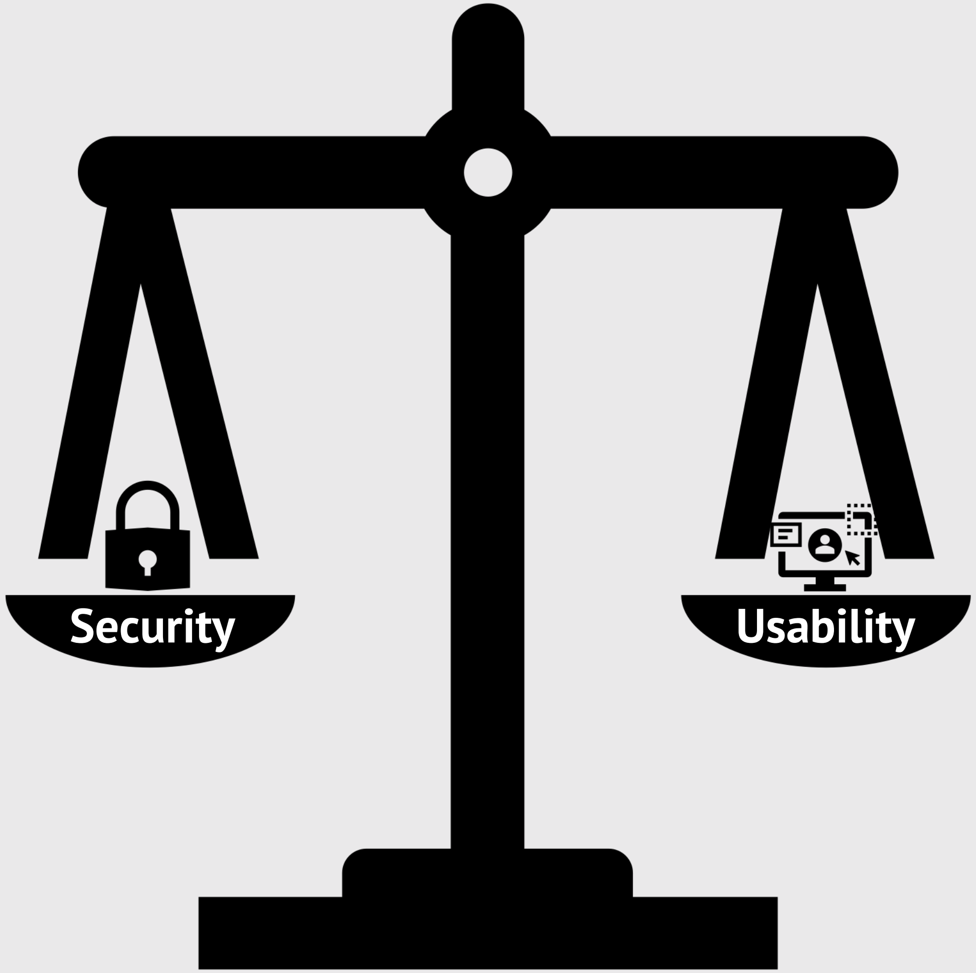
5. Your system may be secure, then the environment changes
   - Pace of technology evolution
   - Threat models change

Why is computer security a hard problem? (continued)

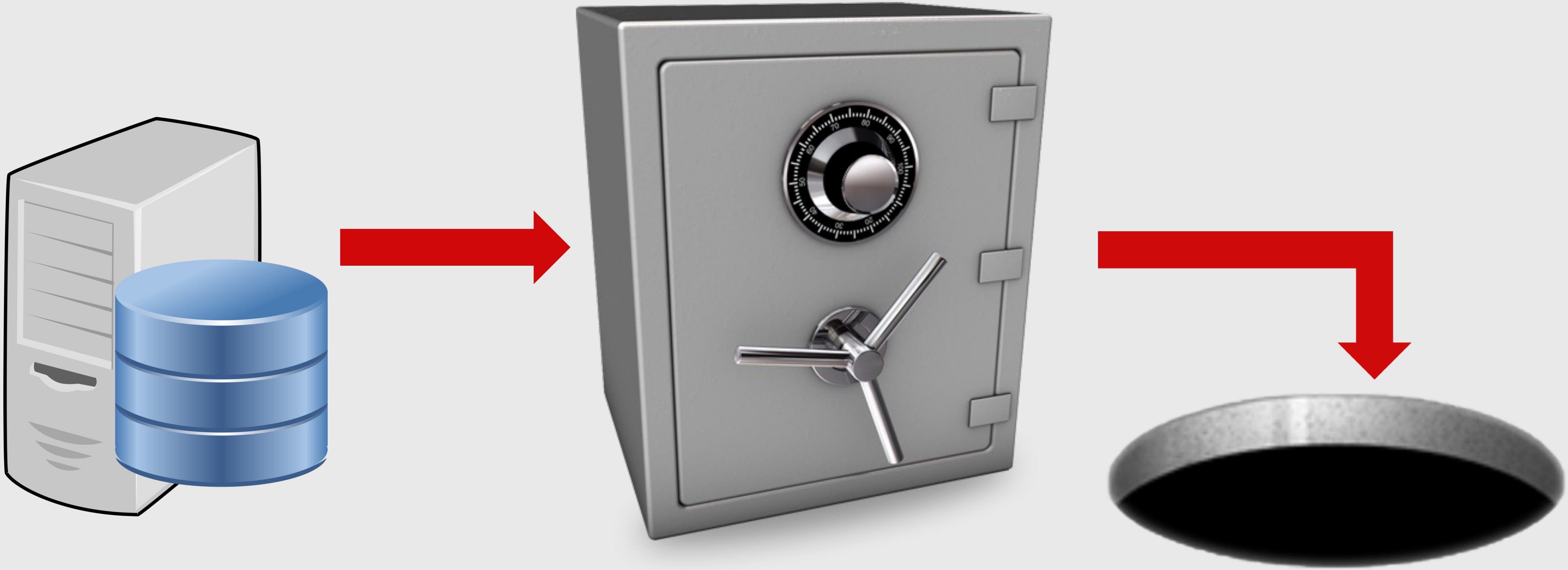
6. Ease of attacks
    - Cheap
    - Distributed, automated
    - Anonymous
    - Insider threats
7. Security not built in from the beginning
8. Humans are in the loop
9. Computers are ubiquitous
10. Society is unwilling to trade off features for security

Balancing Security with Cost and Usability

Security
Costs



Absolute security is easy to achieve!



Security is not about perfection

Security is about defenses that are good enough
to stop the threats that you are likely to encounter

Learn to think with a security mindset

1. What is “the system”?
2. How could this system be attacked?
3. What is the weakest point of attack?
4. How could this system be defended?
5. What threats am I trying to address?
6. How effective will a given countermeasure be?
7. What is the trade-off between security, cost, and usability?

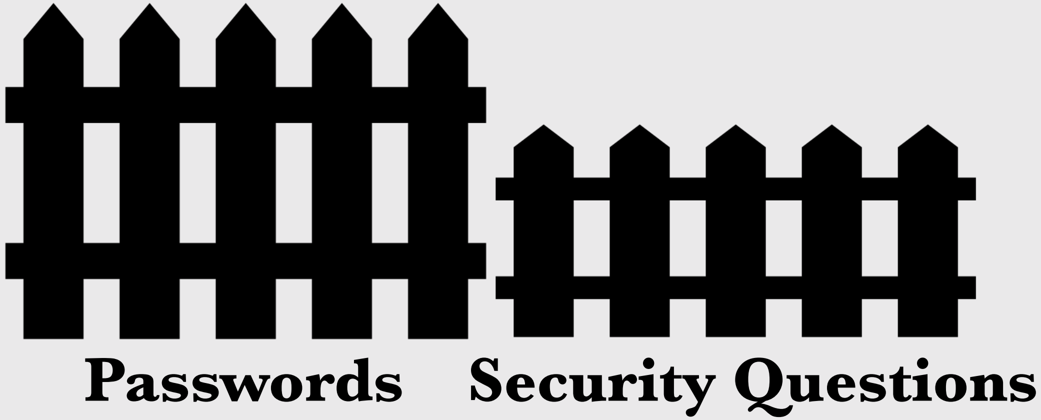
Problems with policy

Case #1 Password recovery questions

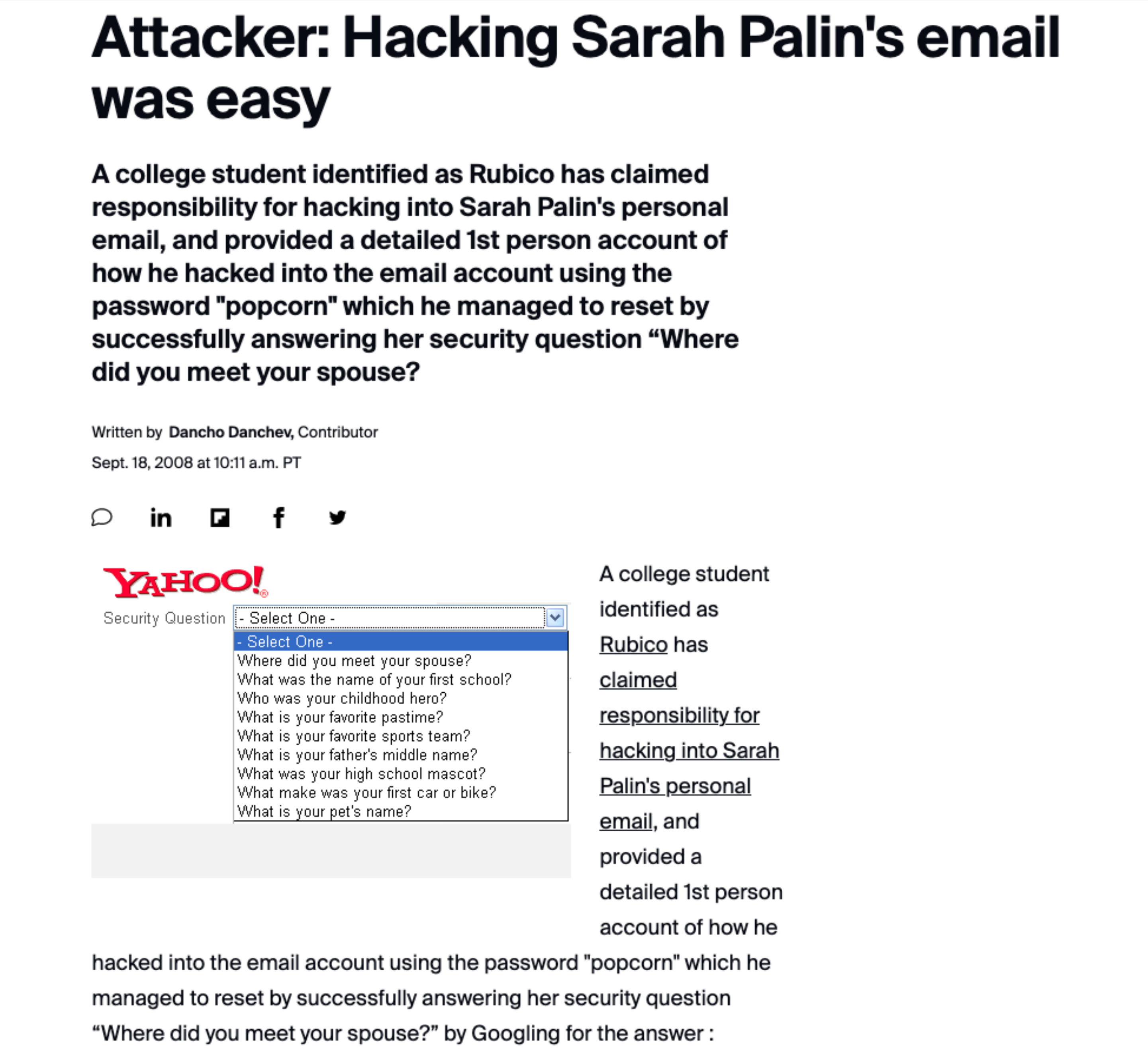
1. User can log in by supplying username and password

2. If user forgets can reset by answering security questions

3. Security questions can sometimes be easier to guess



Not equal height fences



Attacker: Hacking Sarah Palin's email was easy

A college student identified as Rubico has claimed responsibility for hacking into Sarah Palin's personal email, and provided a detailed 1st person account of how he hacked into the email account using the password "popcorn" which he managed to reset by successfully answering her security question “Where did you meet your spouse?

Written by **Dancho Danchev,** Contributor

Sept. 18, 2008 at 10:11 a.m. PT

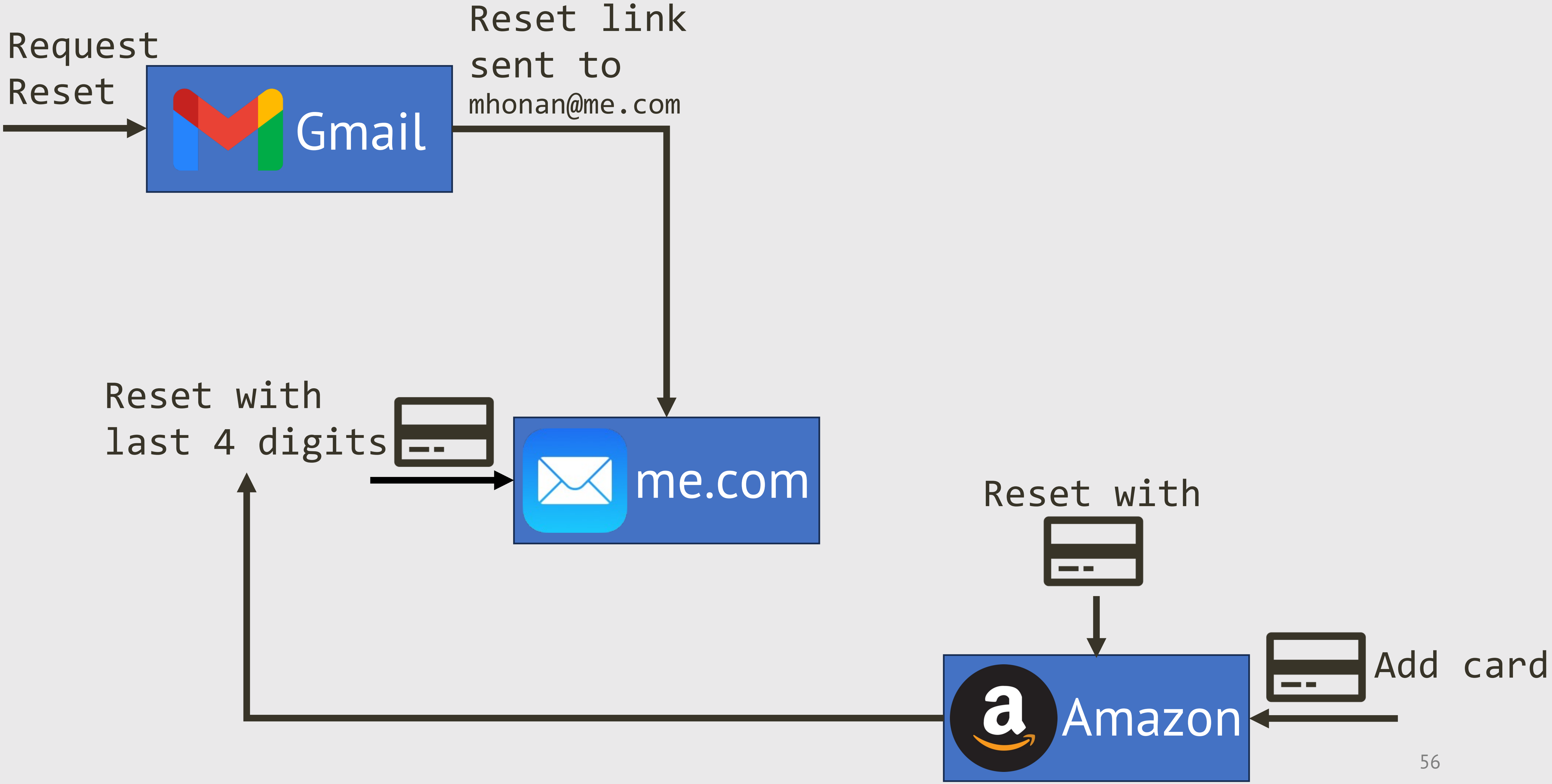
YAHOO!
Security Question
- Select One -
- Select One -
Where did you meet your spouse?
What was the name of your first school?
Who was your childhood hero?
What is your favorite pastime?
What is your favorite sports team?
What is your father's middle name?
What was your high school mascot?
What make was your first car or bike?
What is your pet's name?

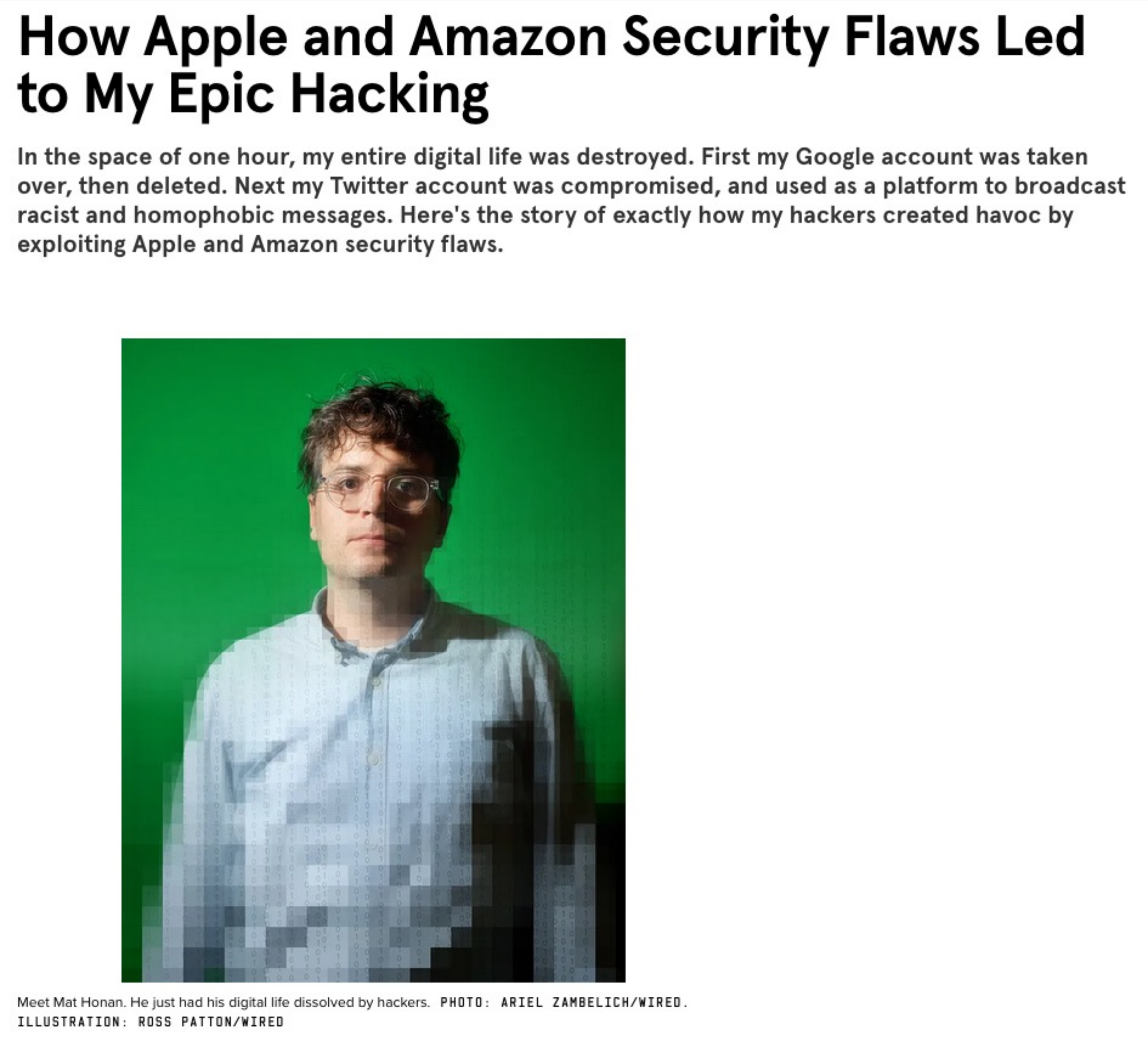
A college student identified as Rubico has claimed responsibility for hacking into Sarah Palin's personal email, and provided a detailed 1st person account of how he hacked into the email account using the password "popcorn" which he managed to reset by successfully answering her security question “Where did you meet your spouse?” by Googling for the answer :

https://www.zdnet.com/article/attacker-hacking-sarah-palins-email-was-easy/

Case #2 Reasonable policy can still fail

Gmail password reset will send a verification link to a backup email address and helpfully displays part of the backup email address


Request Reset
Gmail
Reset link sent to
mhonan@me.com
Reset with last 4 digits
me.com
Reset with
Amazon
Add card



How Apple and Amazon Security Flaws Led to My Epic Hacking

In the space of one hour, my entire digital life was destroyed. First my Google account was taken over, then deleted. Next my Twitter account was compromised, and used as a platform to broadcast racist and homophobic messages. Here's the story of exactly how my hackers created havoc by exploiting Apple and Amazon security flaws.

Meet Mat Honan. He just had his digital life dissolved by hackers. PHOTO: ARIEL ZAMBELICH/WIRED. ILLUSTRATION: ROSS PATTON/WIRED

https://www.wired.com/2012/08/apple-amazon-mat-honan-hacking/

How can we improve policies?

1. Consider implications of policy statements

2. Address all potential threats and risks

3. Periodically update to address new threats

4. Regular employee training on policy importance and practices

Problems with threat models

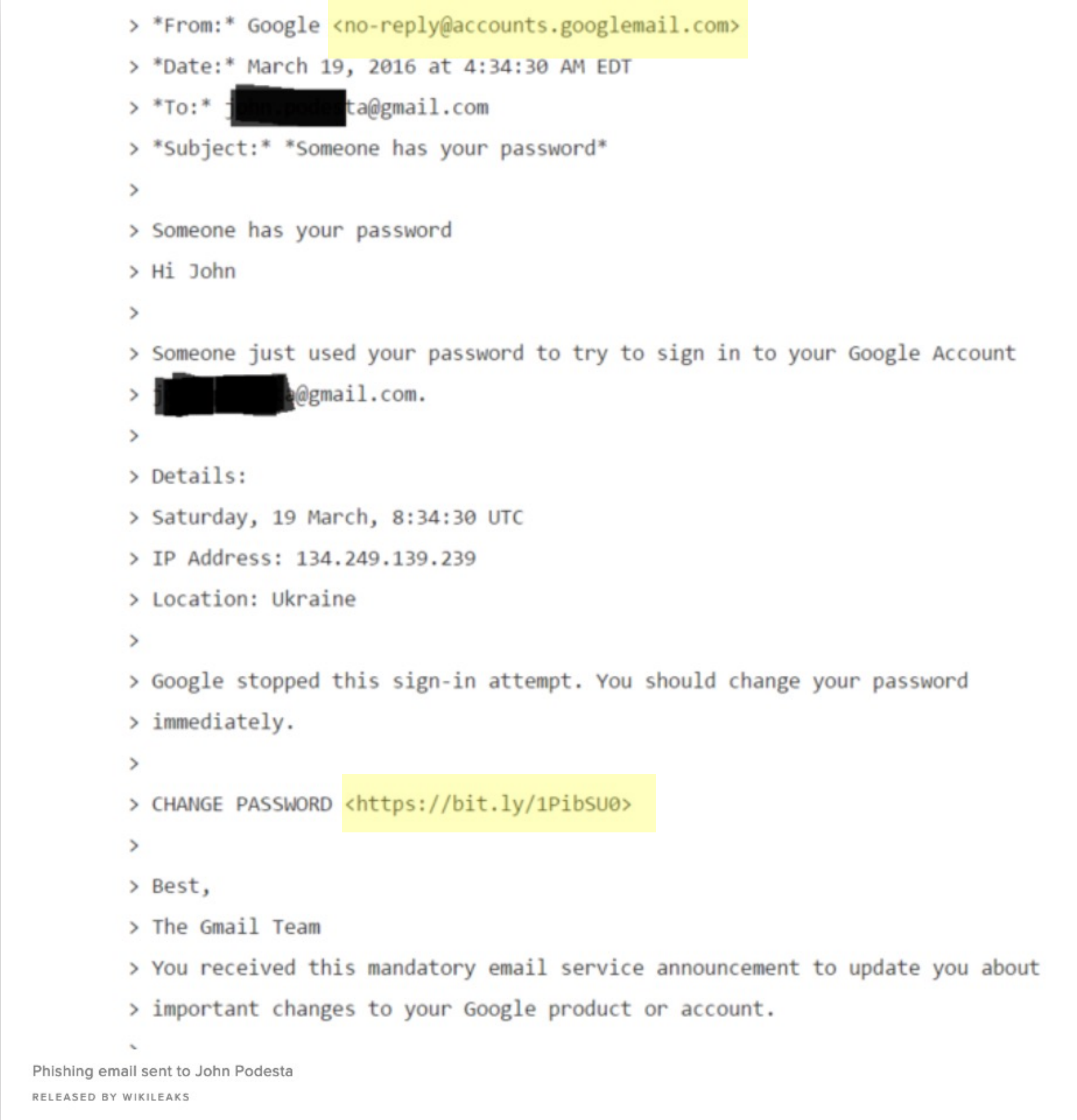
Case #1 Human factors hard to account for

Social Engineering

1. Phishing attacks

2. User gets email asking to renew account or transfer money

3. Tech support gets a call from convincing user to reset password

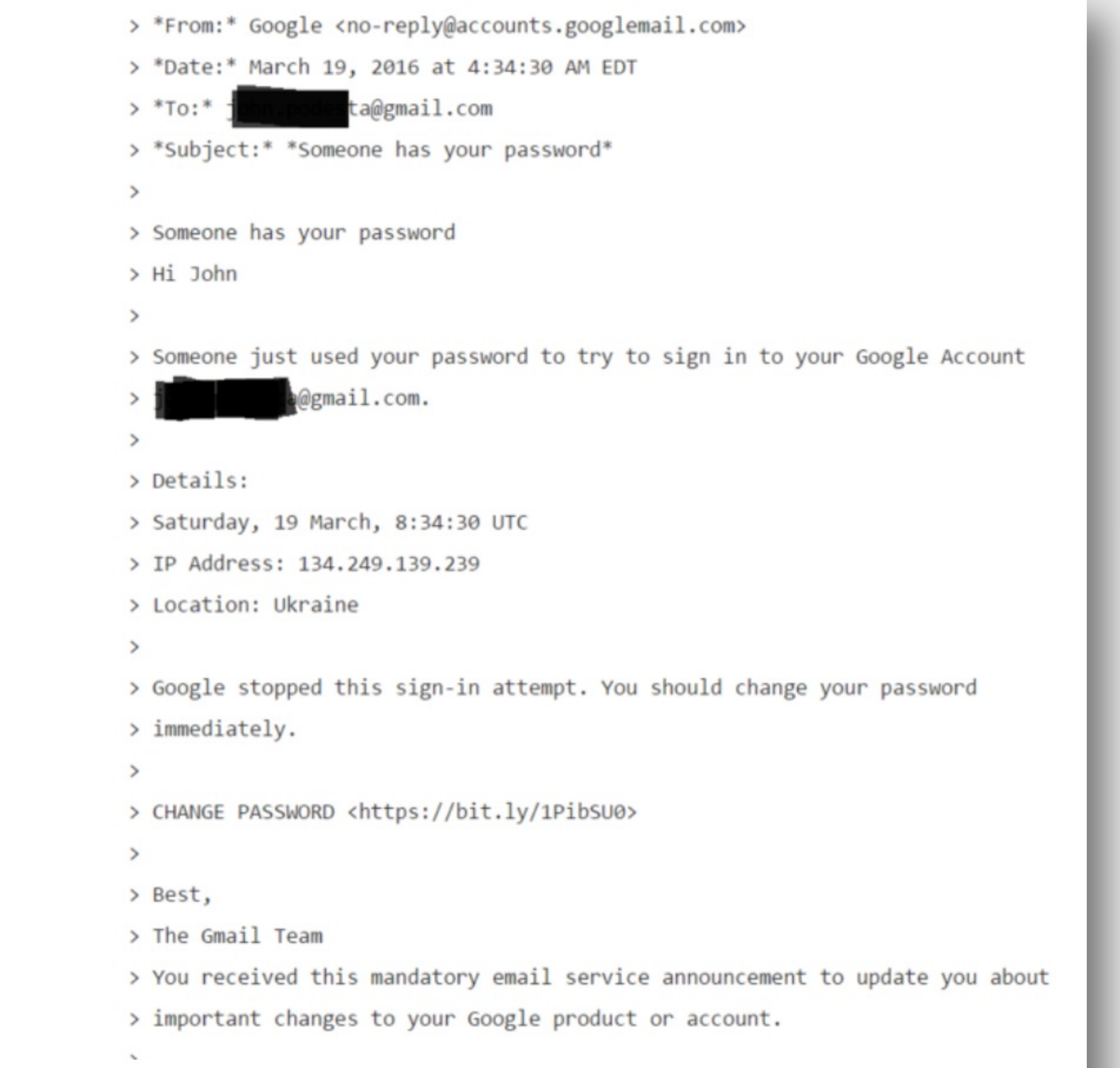


```
> *From:* Google <no-reply@accounts.googlemail.com>
> *Date:* March 19, 2016 at 4:34:30 AM EDT
> *To:* [redacted]ta@gmail.com
> *Subject:* *Someone has your password*
>
> Someone has your password
> Hi John
>
> Someone just used your password to try to sign in to your Google Account
> [redacted]@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>
> CHANGE PASSWORD <https://bit.ly/1PibSU0>
>
> Best,
> The Gmail Team
> You received this mandatory email service announcement to update you about
> important changes to your Google product or account.
```

Phishing email sent to John Podesta

RELEASED BY WIKILEAKS

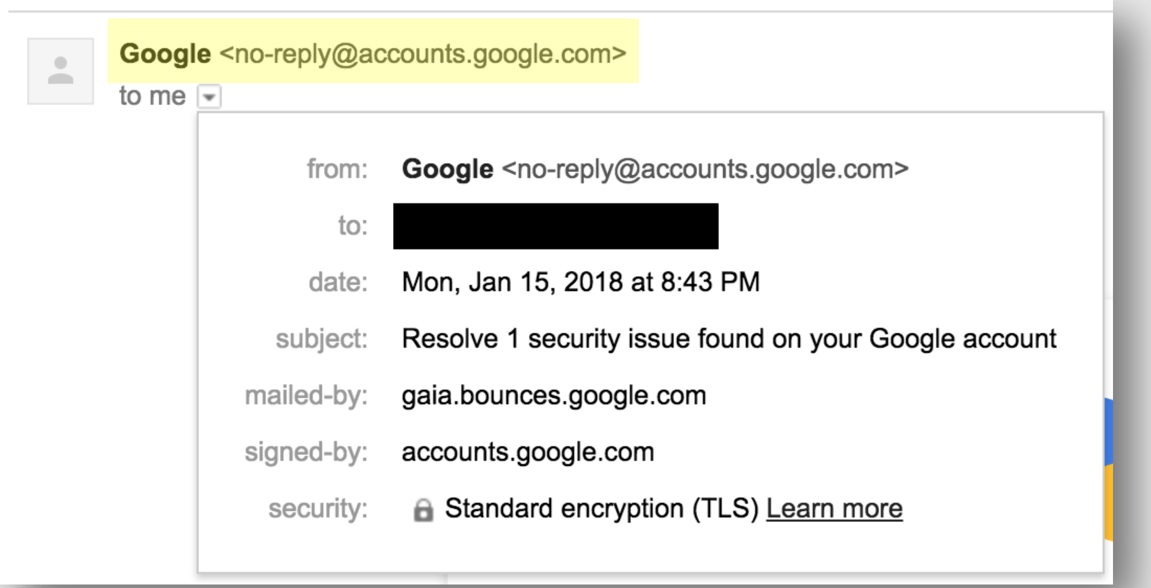
https://www.cbsnews.com/news/the-phishing-email-that-hacked-the-account-of-john-podesta/



```
> *From:* Google <no-reply@accounts.googlemail.com>
> *Date:* March 19, 2016 at 4:34:30 AM EDT
> *To:* [redacted]ta@gmail.com
> *Subject:* *Someone has your password*
>
> Someone has your password
> Hi John
>
> Someone just used your password to try to sign in to your Google Account
> [redacted]@gmail.com.
>
> Details:
> Saturday, 19 March, 8:34:30 UTC
> IP Address: 134.249.139.239
> Location: Ukraine
>
> Google stopped this sign-in attempt. You should change your password
> immediately.
>
> CHANGE PASSWORD <https://bit.ly/1PibSU0>
>
> Best,
> The Gmail Team
> You received this mandatory email service announcement to update you about
> important changes to your Google product or account.
```

Phishing email sent to John Podesta

RELEASED BY WIKILEAKS



Google <no-reply@accounts.google.com>

to me

from:	**Google** <no-reply@accounts.google.com>
to:	[redacted]
date:	Mon, Jan 15, 2018 at 8:43 PM
subject:	Resolve 1 security issue found on your Google account
mailed-by:	gaia.bounces.google.com
signed-by:	accounts.google.com
security:	🔒 Standard encryption (TLS) Learn more

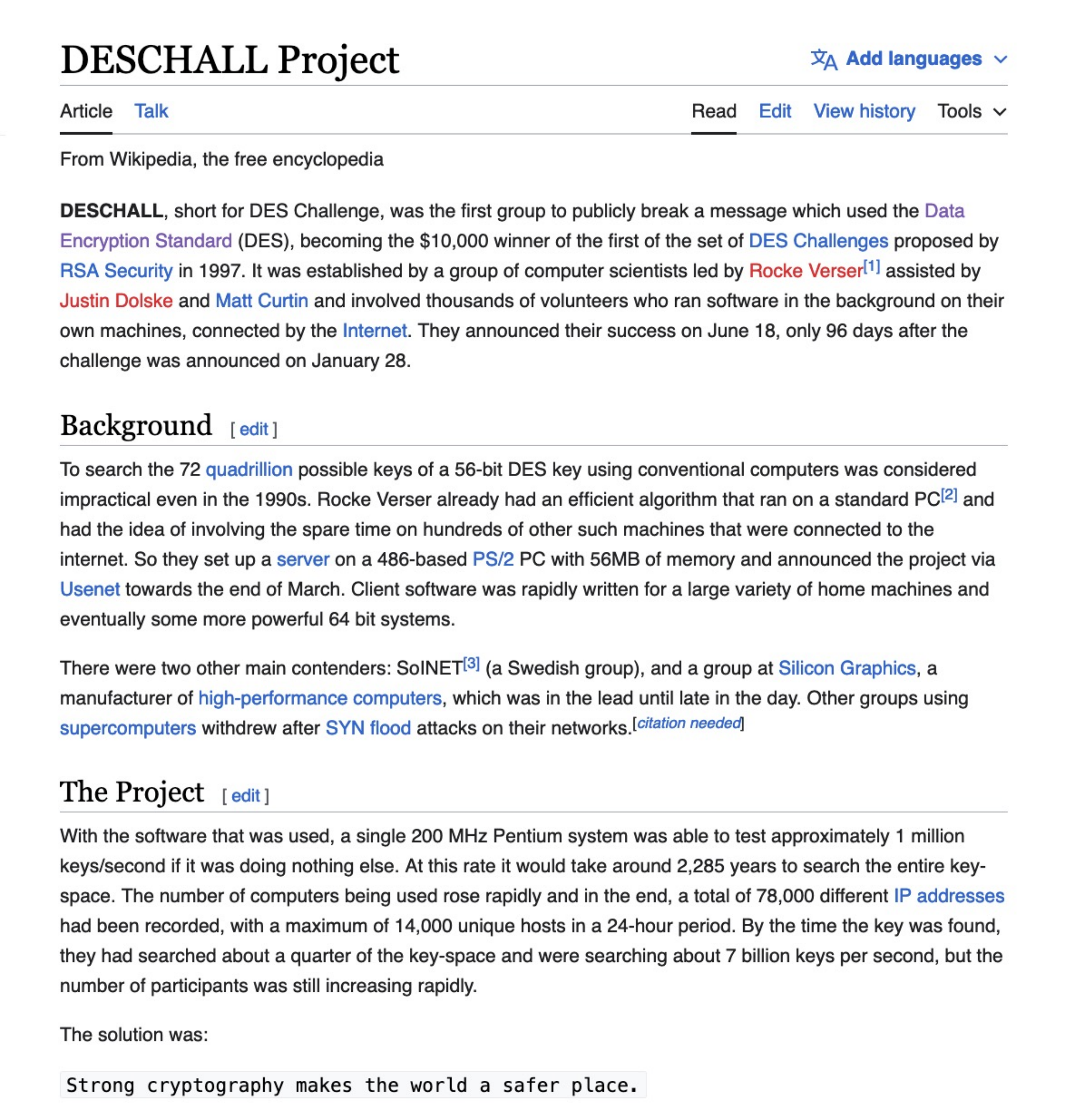
Case #2 Computational assumptions change

Data Encryption Standard (DES)

1. Was a widely-used encryption standard

2. Key length: 56 bits → 2^{56} possible keys

3. At the time it seemed fine to assume that an adversary could not use brute-force to check all keys, but that is no longer reasonable with today's computational power



DESCHALL Project

From Wikipedia, the free encyclopedia

DESCHALL, short for DES Challenge, was the first group to publicly break a message which used the Data Encryption Standard (DES), becoming the $10,000 winner of the first of the set of DES Challenges proposed by RSA Security in 1997. It was established by a group of computer scientists led by Rocke Verser[1] assisted by Justin Dolske and Matt Curtin and involved thousands of volunteers who ran software in the background on their own machines, connected by the Internet. They announced their success on June 18, only 96 days after the challenge was announced on January 28.

Background [edit]

To search the 72 quadrillion possible keys of a 56-bit DES key using conventional computers was considered impractical even in the 1990s. Rocke Verser already had an efficient algorithm that ran on a standard PC[2] and had the idea of involving the spare time on hundreds of other such machines that were connected to the internet. So they set up a server on a 486-based PS/2 PC with 56MB of memory and announced the project via Usenet towards the end of March. Client software was rapidly written for a large variety of home machines and eventually some more powerful 64 bit systems.

There were two other main contenders: SolNET[3] (a Swedish group), and a group at Silicon Graphics, a manufacturer of high-performance computers, which was in the lead until late in the day. Other groups using supercomputers withdrew after SYN flood attacks on their networks.[*citation needed*]

The Project [edit]

With the software that was used, a single 200 MHz Pentium system was able to test approximately 1 million keys/second if it was doing nothing else. At this rate it would take around 2,285 years to search the entire key-space. The number of computers being used rose rapidly and in the end, a total of 78,000 different IP addresses had been recorded, with a maximum of 14,000 unique hosts in a 24-hour period. By the time the key was found, they had searched about a quarter of the key-space and were searching about 7 billion keys per second, but the number of participants was still increasing rapidly.

The solution was:

```
Strong cryptography makes the world a safer place.
```

https://en.wikipedia.org/wiki/DESCHALL_Project

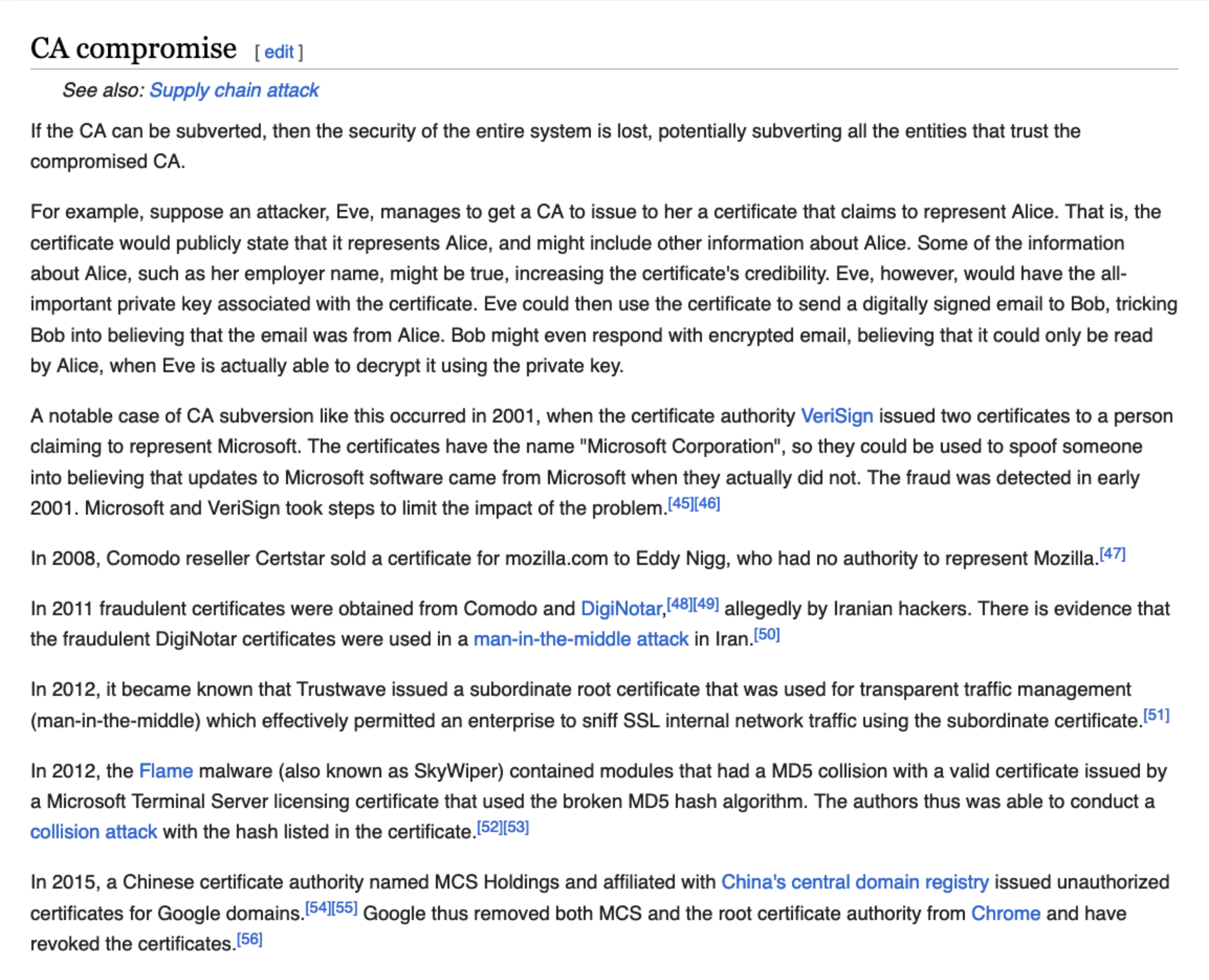
Case #3 Misplaced Trust

Can SSL Certificate Authorities be trusted?

1. Connect to an SSL-enabled web site

2. Web browser verifies the certificate with a certificate authority

3. Certificate contains server’s hostname and a cryptographic key, signed by a trusted certificate authority (CA)



CA compromise [edit]

See also: Supply chain attack

If the CA can be subverted, then the security of the entire system is lost, potentially subverting all the entities that trust the compromised CA.

For example, suppose an attacker, Eve, manages to get a CA to issue to her a certificate that claims to represent Alice. That is, the certificate would publicly state that it represents Alice, and might include other information about Alice. Some of the information about Alice, such as her employer name, might be true, increasing the certificate's credibility. Eve, however, would have the all-important private key associated with the certificate. Eve could then use the certificate to send a digitally signed email to Bob, tricking Bob into believing that the email was from Alice. Bob might even respond with encrypted email, believing that it could only be read by Alice, when Eve is actually able to decrypt it using the private key.

A notable case of CA subversion like this occurred in 2001, when the certificate authority VeriSign issued two certificates to a person claiming to represent Microsoft. The certificates have the name "Microsoft Corporation", so they could be used to spoof someone into believing that updates to Microsoft software came from Microsoft when they actually did not. The fraud was detected in early 2001. Microsoft and VeriSign took steps to limit the impact of the problem.[45][46]

In 2008, Comodo reseller Certstar sold a certificate for mozilla.com to Eddy Nigg, who had no authority to represent Mozilla.[47]

In 2011 fraudulent certificates were obtained from Comodo and DigiNotar,[48][49] allegedly by Iranian hackers. There is evidence that the fraudulent DigiNotar certificates were used in a man-in-the-middle attack in Iran.[50]

In 2012, it became known that Trustwave issued a subordinate root certificate that was used for transparent traffic management (man-in-the-middle) which effectively permitted an enterprise to sniff SSL internal network traffic using the subordinate certificate.[51]

In 2012, the Flame malware (also known as SkyWiper) contained modules that had a MD5 collision with a valid certificate issued by a Microsoft Terminal Server licensing certificate that used the broken MD5 hash algorithm. The authors thus was able to conduct a collision attack with the hash listed in the certificate.[52][53]

In 2015, a Chinese certificate authority named MCS Holdings and affiliated with China's central domain registry issued unauthorized certificates for Google domains.[54][55] Google thus removed both MCS and the root certificate authority from Chrome and have revoked the certificates.[56]

https://en.wikipedia.org/wiki/Certificate_authority

How can we improve threat models?

1. More comprehensive threat identification

2. Less reliance on certain assumptions
   - For example, trust models that don't assume fully-trusted CAs

3. Update models to reflect new threats and vulnerabilities

Problems with mechanisms

(bugs that lead to vulnerabilities)

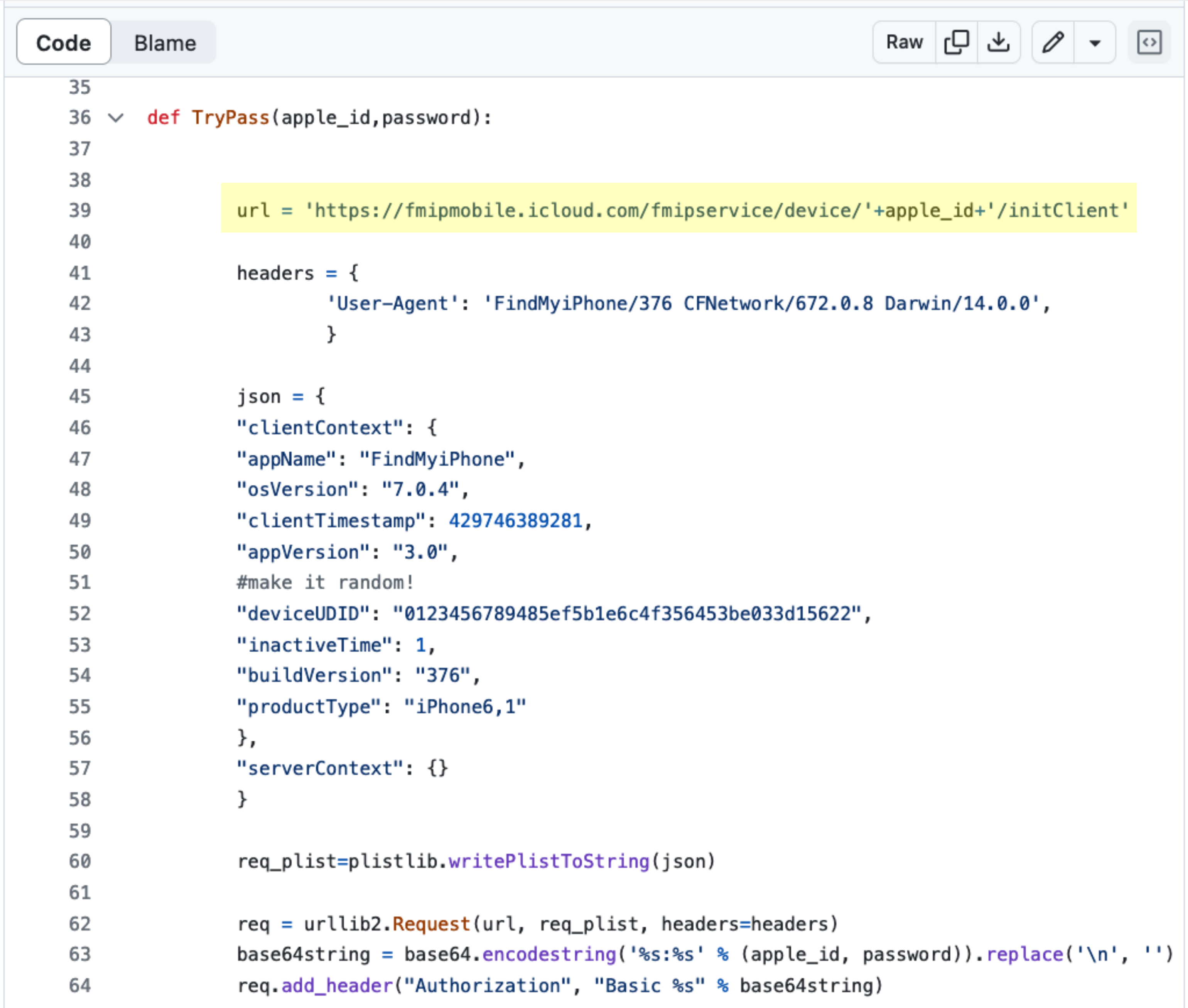
Case #1 Apple iCloud password-guessing

1. People often use weak passwords
    - Often guessed within 1K to 1M attempts

2. Most services, including Apple iCloud rate-limit login attempts

3. Apple iCloud service has many APIs and one API (Find My iPhone) forgot to implement rate-limiting

4. Adversaries could make millions of guesses per day



```python

def TryPass(apple_id,password):


        url = 'https://fmipmobile.icloud.com/fmipservice/device/'+apple_id+'/initClient'

        headers = {
                'User-Agent': 'FindMyiPhone/376 CFNetwork/672.0.8 Darwin/14.0.0',
                }

        json = {
        "clientContext": {
        "appName": "FindMyiPhone",
        "osVersion": "7.0.4",
        "clientTimestamp": 429746389281,
        "appVersion": "3.0",
        #make it random!
        "deviceUDID": "0123456789485ef5b1e6c4f356453be033d15622",
        "inactiveTime": 1,
        "buildVersion": "376",
        "productType": "iPhone6,1"
        },
        "serverContext": {}
        }

        req_plist=plistlib.writePlistToString(json)

        req = urllib2.Request(url, req_plist, headers=headers)
        base64string = base64.encodestring('%s:%s' % (apple_id, password)).replace('\n', '')
        req.add_header("Authorization", "Basic %s" % base64string)
```

https://github.com/hackappcom/ibrute

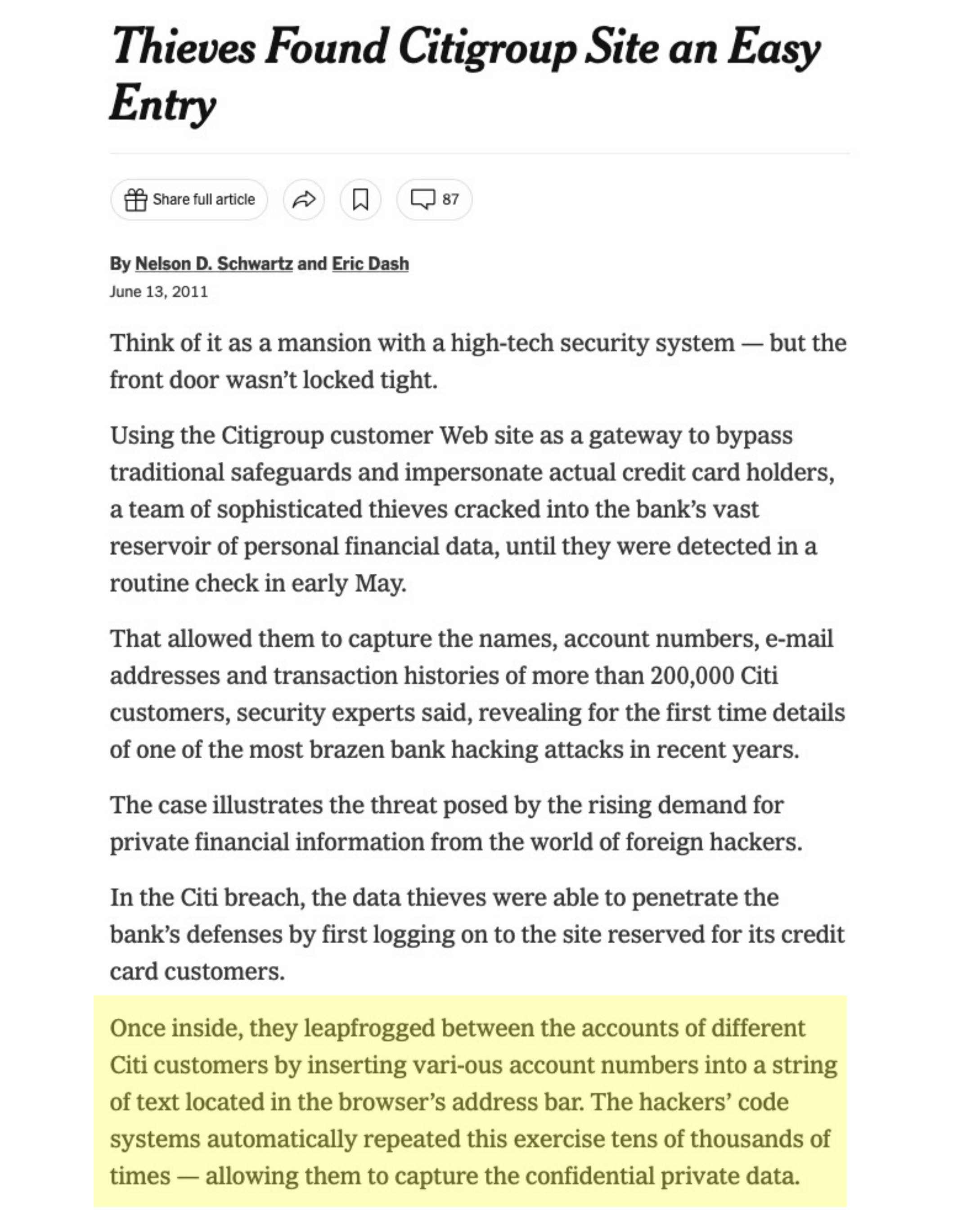
Case #2 Missing access control checks

1. Citigroup allowed credit card users to access accounts online
   - Login page asks for username and password
   - If username and password are valid, redirected to account info page

2. URL of the info page contained the user's account number

```
https://citigroup.com/acct?id=12345678
```

3. The server didn't check that you were logged into that account



Thieves Found Citigroup Site an Easy Entry

By Nelson D. Schwartz and Eric Dash
June 13, 2011

Think of it as a mansion with a high-tech security system — but the front door wasn't locked tight.

Using the Citigroup customer Web site as a gateway to bypass traditional safeguards and impersonate actual credit card holders, a team of sophisticated thieves cracked into the bank's vast reservoir of personal financial data, until they were detected in a routine check in early May.

That allowed them to capture the names, account numbers, e-mail addresses and transaction histories of more than 200,000 Citi customers, security experts said, revealing for the first time details of one of the most brazen bank hacking attacks in recent years.

The case illustrates the threat posed by the rising demand for private financial information from the world of foreign hackers.

In the Citi breach, the data thieves were able to penetrate the bank's defenses by first logging on to the site reserved for its credit card customers.

Once inside, they leapfrogged between the accounts of different Citi customers by inserting vari-ous account numbers into a string of text located in the browser's address bar. The hackers' code systems automatically repeated this exercise tens of thousands of times — allowing them to capture the confidential private data.

https://www.nytimes.com/2011/06/14/technology/14security.html

Case #3 SSL certificate name checking bug

1. The vulnerability exploited how browsers interpret null characters (`\0`) in certificate fields

2. Browsers incorrectly interpreted the domain name

3. For example: `amazon.com\0.attacker.com`

What can we trust?

What **code** can we trust?

Can we trust the “login” program on Linux?

No! The login program may contain a “backdoor”

For example, code that records passwords as they are typed in

Solution: recompile the login program from the source code

But can we trust the login source code?

No! But we can inspect the source code, and then recompile

Can we trust the **compiler**?

No! The compiler could contain malicious code:

```
compile(code)
{
     if (match(code, “login-program”))
     {
           compile(“login-program-with-backdoor”);
           return;
     }
     / * regular compilation */
}
```

What to do?

Solution: inspect the compiler source code
and then recompile the compiler

Problem: C compiler is itself written in C, and compiles itself

What if the compiler binary has a backdoor?

Ken Thompson’s clever backdoor

Attack step 1: modify the compiler source code

```
compile(code)
{
      if (match(code, “login-program”))
      {
            compile(“login-program-with-backdoor”);
            return;
      }
      if (match(code, “compiler-program”))
      {
            compile(“compiler-with-backdoor”);
            return;
      }
      / * regular compilation */
}
```

Ken Thompson’s clever backdoor

Attack step 2:

- Compile the modified compiler → compiler binary
- Restore the compiler source code to its original state

Now: inspecting the compiler source code reveals nothing unusual

… but compiling the compiler gives a corrupt binary

“The moral is obvious. You can’t trust code that you did not totally create yourself. No amount of source-level verification or scrutiny will protect you from using untrusted code.”

-- Ken Tompson **Reflections on Trusting Trust**

What can we trust?

If I order a laptop by mail. When it arrives, what can I trust on it?

Applications and/or operating system may be backdoored

Solution: reinstall OS and applications

But, how to reinstall?

Can’t trust OS to reinstall the OS

Boot a trusted version of Linux from a USB drive

What can we trust?

Need to trust pre-boot BIOS and Firmware code. Can we trust it?

No! (Supply chain attacks: ShadowHammer)

Can we trust the motherboard? Software updates?

So, what can we trust?

Sadly, nothing … anything can be compromised

but then we can't make progress

Trusted Computing Base (TCB)

- Assume some minimal part of the system is not compromised
- Then build a secure environment on top of that