



# AUTHENTICATION

VITALY SHMATIKOV

Image: Lorrie Cranor's password quilt

# The Core Problem of Authentication

How do you prove to someone that you are who you claim to be?

Any system with access control must solve this problem

- **What you know:** password, PIN, answers to questions only you know
- **What you are:** biometrics
- **What you have:** phone number, mobile device, secure token
- **Where you are:** IP address, geolocation
- **Someone/something knows you:** single sign-on (Cornell NetID), PKI

# Authentication vs. Authorization & Access Control

- **Authentication:** is the user (or program) who they claim they are?
- **Authorization:** should user (or program) have access to a given resource?
  - Authorization decisions rely on correct authentication
- **Access control:** policy and enforcement mechanism



Who is your favorite member of The Roots?



Microsoft

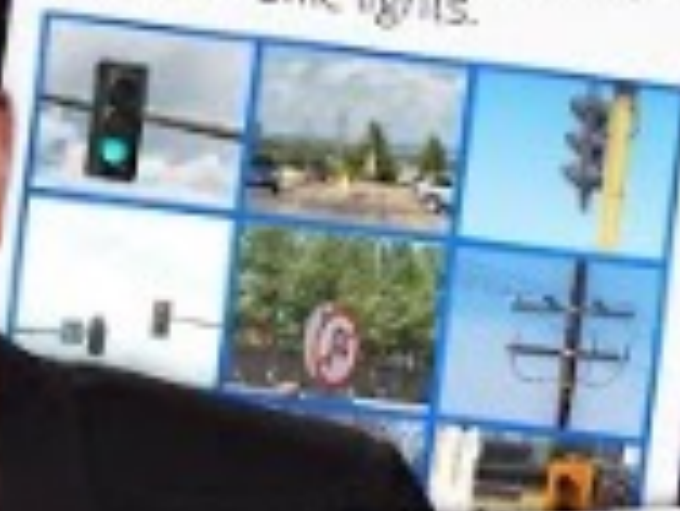


A verification code has been sent to your email.



Authenticator

In order to proceed, please identify which of these images contain traffic lights.



What is your tallest middle school teacher's middle name?

# Password-Based Authentication

User has a secret password.

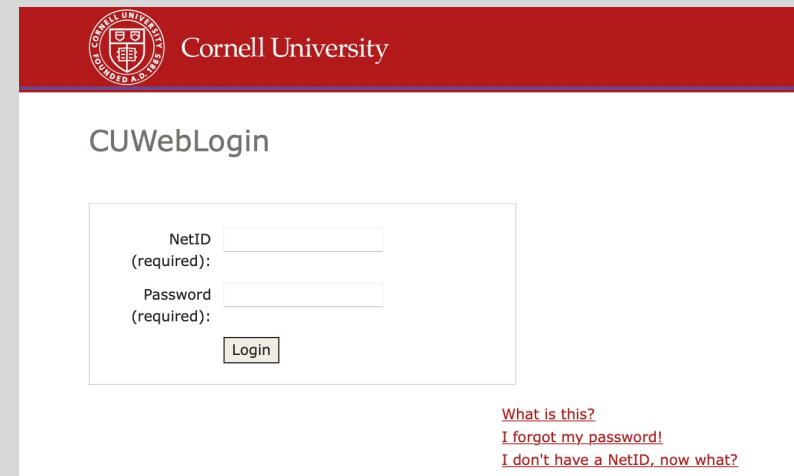
System checks it to authenticate the user.

- How is the password communicated?
- How is the password stored?
- How does the system check the password?
- How easy is it to guess the password?

Easy to deploy

Easy to use (nothing to carry, etc.)

No simple alternative



The screenshot shows the Cornell University CUWebLogin page. At the top is a red header with the Cornell University logo and name. Below the header, the title "CUWebLogin" is displayed. The main content area contains a login form with two input fields: "NetID (required):" and "Password (required):". A "Login" button is positioned below the password field. At the bottom right of the form area, there are three red links: "What is this?", "I forgot my password!", and "I don't have a NetID, now what?"



# Attacks on Passwords

## Online

- Try to guess passwords by logging to a live system

## Offline

- Try to guess passwords in the (typically stolen) password database
- **Pre-computation** can make offline attacks very fast

## Phishing

- Trick user into disclosing their password
- Spear-phishing: phishing a specific user with personalized attacks

# Passwords Are The Bane of Computer Security

Phishing and use of stolen credentials are the top two hacking techniques

- Source: Verizon Data Breach Investigations Report

First step after any successful intrusion: install sniffer or keylogger to steal more passwords

Then run cracking tools on password files

- Modern systems usually do not store passwords in the clear (how are they stored?)

In August 2021, a 21-year-old hacker scanned T-Mobile's known Internet addresses

Discovered an unprotected router, used it to gain access to a data center near East Wenatchee, WA

Login credentials stored in the data center provided access to 100 more servers

## ***T-Mobile Says Hack Exposed Personal Data of 40 Million People***

The company said that stolen files included the personal information of 7.8 million current customers and 40 million people who had applied for credit.

*NYT, August 18, 2021*



*Image: WSJ*



## Hackers breached Colonial Pipeline with one compromised password

*The password has since been discovered inside a batch of leaked passwords on the dark web.*



Hackers gained entry into the networks of Colonial Pipeline Co. on April 29 through a virtual private network account, which allowed employees to remotely access the company's computer network

...

The account's password has since been discovered inside a batch of leaked passwords on the dark web. That means a Colonial employee may have used the same password on another account that was previously hacked

...

The VPN account, which has since been deactivated, didn't use multifactor authentication, a basic cybersecurity tool, allowing the hackers to breach Colonial's network using just a compromised username and password.

## ***This Agency's Computers Hold Secrets. Hackers Got In With One Password.***

Hackers used one worker's login information to penetrate the Law Department's network after officials failed to implement a simple security measure.



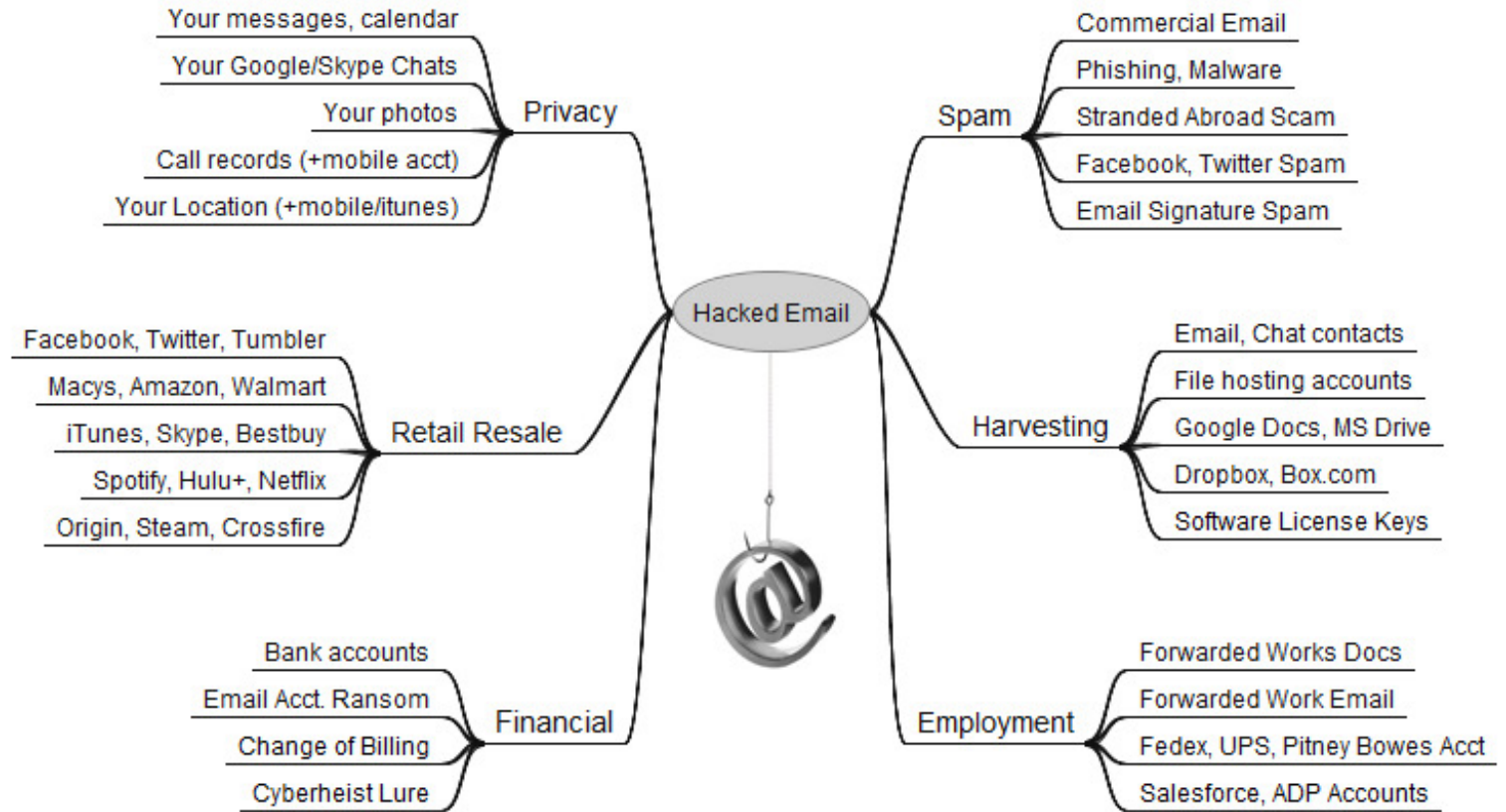
New York City's Law Department holds some of the city's most closely guarded secrets: evidence of police misconduct, the identities of young children charged with serious crimes, plaintiffs' medical records and personal data for thousands of city employees.

...

the hack was enabled by the Law Department's failure to implement a basic safeguard, known as multifactor authentication, more than two years after the city began requiring it

<https://www.nytimes.com/2021/06/18/nyregion/nyc-law-department-hack.html>

# From Here to Eternity



# Default Passwords

Examples from Mitnick's "Art of Intrusion"

- U.S. District Courthouse server: "public" / "public"
- NY Times employee database: pwd = last 4 SSN digits
- "Dixie bank": break into router (pwd="administrator"), then into bank server (pwd="administrator"), install keylogger to snarf other passwords
  - "99% of people there used password123 as their password"

Mirai botnet (2016)

- Used default passwords in IoT devices (Internet cameras, home routers, etc.) to stage a massive distributed denial-of-service flooding attack

# From Mirai's Source Code

Username	Password
666666	666666
888888	888888
admin	(none)
admin	1111
admin	1111111
admin	1234
admin	12345
admin	123456
admin	54321
admin	7ujMko0admin
admin	admin
admin	admin1234
admin	meinsm
admin	pass
admin	password
admin	smcadmin
admin1	password
administrator	1234
Administrator	admin
guest	12345

guest	guest
mother	fucker
root	(none)
root	00000000
root	1111
root	1234
root	12345
root	123456
root	54321
root	666666
root	7ujMko0admin
root	7ujMko0vizxv
root	888888
root	admin
root	anko
root	default
root	dreambox
root	hi3518
root	ikwb
root	juantech
root	jvzbz



## Hacker Group Says It Accessed Tesla's, Others' Internal Video-Surveillance Feeds

Exposed password to administrative account of security-camera vendor Verkada opened door to networks, hackers said

Tillie Kottmann, one of the hackers, said the group found a username and password for a Verkada administrative account on the internet, permitting them to obtain the footage. That included footage from 222 cameras placed inside various Tesla factories and warehouses, Kottmann said in a message. In all, the group could have accessed material from 150,000 Verkada cameras



<https://www.wsj.com/articles/hacker-group-says-it-accessed-teslas-others-internal-video-surveillance-feeds-11615396594>

# rockyou™ Hack (2009)

- “Social gaming” company
- Database with 32 million user passwords from partner social networks
- Passwords stored in the clear
- December 2009: entire database hacked using a **SQL injection attack** and posted on the Internet



*More about SQL injection later*

# Passwords in the RockYou Database

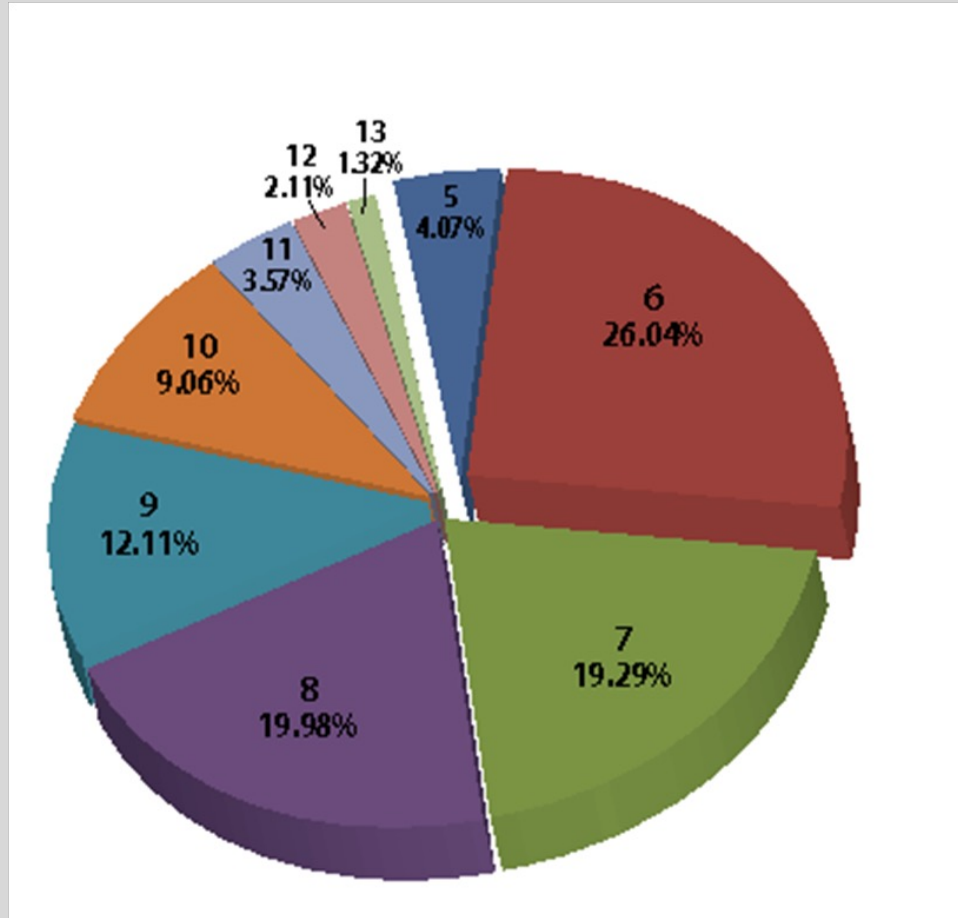
## Password Popularity – Top 20

Rank	Password	Number of Users with Password (absolute)
1	123456	290731
2	12345	79078
3	123456789	76790
4	Password	61958
5	iloveyou	51622
6	princess	35231
7	rockyou	22588
8	1234567	21726
9	12345678	20553
10	abc123	17542

Rank	Password	Number of Users with Password (absolute)
11	Nicole	17168
12	Daniel	16409
13	babygirl	16094
14	monkey	15294
15	Jessica	15162
16	Lovely	14950
17	michael	14898
18	Ashley	14329
19	654321	13984
20	Qwerty	13856

*Source: Imperva*

# Password Length Distribution

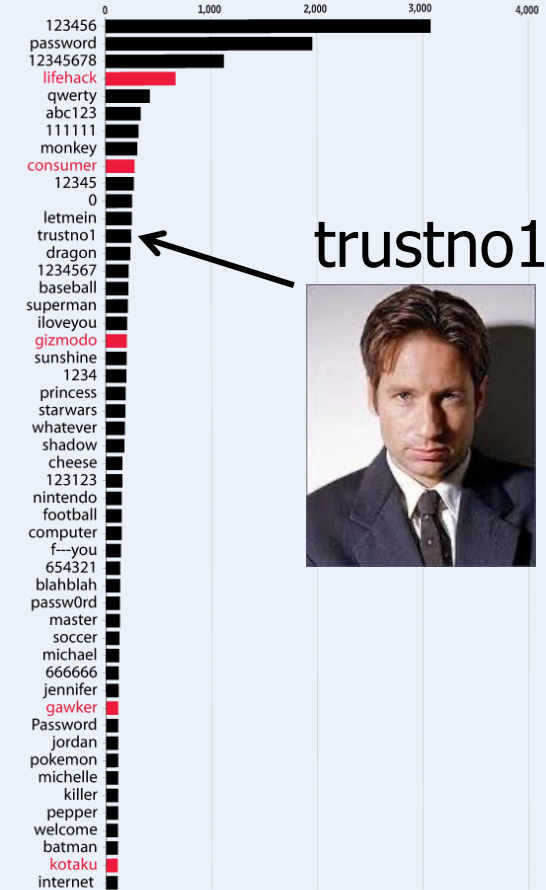


*Source: Imperva*

# Gawker Passwords (2010)

## Bet You Can Guess These

The most popular among 188,279 Gawker Media passwords that leaked online.



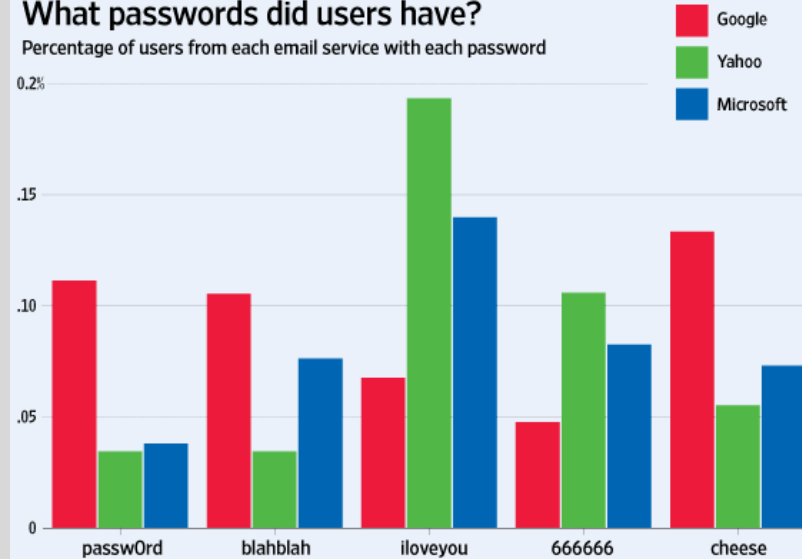
trustno1



Source: Anonymized set of 188,279 leaked Gawker Media passwords. Current and former Gawker Media sites are highlighted in red.

## What passwords did users have?

Percentage of users from each email service with each password



Source: Anonymized set of 188,279 leaked Gawker Media passwords

Source: WSJ





2020: major attack on US government and several companies via a compromised update to SolarWinds network management software

The company used the password **solarwinds123** for a GitHub server



Company denied this was the source of the breach



SolarWinds CEO

"I believe that was a password that an intern used on one of his Github servers back in 2017"

*Image: Insider*

# Adobe Passwords (2013)

- 153 million account passwords
  - 56 million of them unique
- Encrypted using 3DES in ECB mode rather than hashed (why is this important?)

```
79985232-|-|-.a@fbi.gov-|-+ujciL90fBnioXG6CatHBw==|-anniversary|--  
105009730-|-|-.gon@ic.fbi.gov-|-9nCgb38RHw==|-band|--  
108684532-|-|-.burn@ic.fbi.gov-|-EQ7fIpT7i/Q==|-numbers|--  
63041670-|-|-.iv-|-hRwtmq98mKzioXG6CatHBw==|-|--  
94038395-|-|-.n@ic.fbi.gov-|-MreVpEovY17ioXG6CatHBw==|-eod da  
116097938-|-|-.-|-Tur7Wt2zH5CwIIHfjvcHKQ==|-SH?|--  
83310434-|-|-.c.fbi.gov-|-NLupdfyYrsM==|-ATP MIDDLE|--  
113389790-|-|-.iv-|-lMhaearHXjPioXG6CatHBw==|-w|--  
113931981-|-|-.@ic.fbi.gov-|-lTmosXxYnP3ioXG6CatHBw==|-See MSDW  
114081741-|-|-.lom@ic.fbi.gov-|-ZcDbllvCad0==|-fuzzy boy 20|--  
106145242-|-|-.@ic.fbi.gov-|-xc2KumNGzYfioXG6CatHBw==|-4s|--  
106437837-|-|-.i.gov-|-adIewKvmJEsFqx0HFoFrXg==|-|--  
96649467-|-|-.ius@ic.fbi.gov-|-lsYW5KRKNT/ioxG6CatHBw==|-gl  
96670195-|-|-.fbi.gov-|-X4+k4uhyDh/ioxG6CatHBw==|-|--  
105095956-|-|-.earthlink.net-|-ZU2tTTFIZq/ioxG6CatHBw==|-socialsecurity#|--  
108260815-|-|-.r@genext.net-|-MuKnZ7KtsiHioXG6CatHBw==|-socialsecurity|--  
83508352-|-|-.@hotmail.com-|-ADEcoaN2oUM==|-socialsecurityno.--  
83023162-|-|-.k590@aol.com-|-9HT+kVHQfs4==|-socialsecurity name|--  
90331688-|-|-.b.edu-|-nNiwEcoZTBmXrIXpAZiRHQ==|-ssn#|--
```

Password hints

# “Collection #1” (2018-2019)

## Mother of All Breaches Exposes 773 Million Emails, 21 Million Passwords

... just a subset of the seller's offerings



<https://krebsonsecurity.com/2019/01/773m-password-megabreach-is-years-old>

# How About PINs?

- In 2012, Nick Berry analyzed all four-digit passwords from previous leaks

	PIN	Freq
#1	1234	10.713%
#2	1111	6.016%
#3	0000	1.881%
#4	1212	1.197%
#5	7777	0.745%
#6	1004	0.616%
#7	2000	0.613%
#8	4444	0.526%
#9	2222	0.516%
#10	6969	0.512%
#11	9999	0.451%
#12	3333	0.419%
#13	5555	0.395%
#14	6666	0.391%
#15	1122	0.366%
#16	1313	0.304%
#17	8888	0.303%
#18	4321	0.293%
#19	2001	0.290%
#20	1010	0.285%

	PIN	Freq
#9980	8557	0.001191%
#9981	9047	0.001161%
#9982	8438	0.001161%
#9983	0439	0.001161%
#9984	9539	0.001161%
#9985	8196	0.001131%
#9986	7063	0.001131%
#9987	6093	0.001131%
#9988	6827	0.001101%
#9989	7394	0.001101%
#9990	0859	0.001072%
#9991	8957	0.001042%
#9992	9480	0.001042%
#9993	6793	0.001012%
#9994	8398	0.000982%
#9995	0738	0.000982%
#9996	7637	0.000953%
#9997	6835	0.000953%
#9998	9629	0.000953%
#9999	8093	0.000893%
#10000	8068	0.000744%

# Memorability vs. Security

One bank's idea for making PINs "memorable"

- If PIN is 2256, write your favorite word in the grid

1	2	3	4	5	6	7	8	9	0
	b								
	l								
				u					
					e				

Normally 9,999 choices for PIN –  
hard(er) to guess

Now only a few dozen possible  
English words – easy to guess!

- Fill the rest with random letters

*Source: Ross Anderson*



# Cracking Techniques

## Wordlists

- 20-500 million words and leaked passwords in publicly available lists

## Mangling rules to generate variants

- Dozens of thousands of rules  
(Korelogic, Megatron, Generated2)

## Cracking tools

- Example: John the Ripper, Hashcat



# Password Mangling and Generation

Dictionary with  
words spelled  
backwards

First and last  
names, streets,  
cities

Same with upper-  
case initials

All valid license  
plate numbers in  
your state

Room numbers,  
telephone  
numbers, etc.

Letter substitutions  
and other tricks

*If you can think of it, attacker will, too!*

# Social Engineering

## Univ. of Sydney study (1996)

- 336 CS students emailed asking for their passwords
  - Pretext: “validate” password database after suspected break-in
- 138 returned their passwords; 30 returned invalid passwords; 200 reset passwords (not disjoint)

## Treasury Dept. report (2005)

- Auditors pose as IT personnel attempting to correct a “network problem”
- 35 of 100 IRS managers and employees provide their usernames and change passwords to a known value

# Bezos, Musk, Gates, Obama and others target of cryptocurrency hack on Twitter

**Jefferson Graham, Emre Kelly and Mike Snider** USA TODAY

Published 5:42 p.m. ET Jul. 15, 2020

The Twitter accounts of prominent figures from the worlds of tech and money, celebrities, a presidential candidate and a former president were all hacked Wednesday in what was the largest breach in the company's history.

Bogus messages soliciting bitcoin appeared on the Twitter accounts for Tesla CEO Elon Musk, Microsoft co-founder Bill Gates, Amazon CEO and founder Jeff Bezos, Berkshire Hathaway CEO and president Warren Buffett, former President Barack Obama, presumptive Democratic candidate Joe Biden, former New York mayor Michael Bloomberg, Israeli Prime Minister Benjamin Netanyahu and the corporate accounts for Apple and Uber.

Celebrities were also targeted in the bitcoin scam including rapper Kanye West and his wife Kim Kardashian and rapper Wiz Khalifa.

Twitter said late Wednesday that it detected what it believes was a "coordinated social engineering attack by people who successfully targeted some of our employees with access to internal systems and tools."

# July 2020 Twitter Hack

The social engineering that occurred on July 15, 2020, targeted a small number of employees through a phone spear phishing attack. A successful attack required the attackers to obtain access to both our internal network as well as specific employee credentials that granted them access to our internal support tools. Not all of the employees that were initially targeted had permissions to use account management tools, but the attackers used their credentials to access our internal systems and gain information about our processes. This knowledge then enabled them to target additional employees who did have access to our account support tools. Using the credentials of employees with access to these tools, the attackers targeted 130 Twitter accounts, ultimately Tweeting from 45, accessing the DM inbox of 36, and downloading the Twitter Data of 7.

*[https://blog.twitter.com/en\\_us/topics/company/2020/an-update-on-our-security-incident.html](https://blog.twitter.com/en_us/topics/company/2020/an-update-on-our-security-incident.html)*



# How People Use Passwords

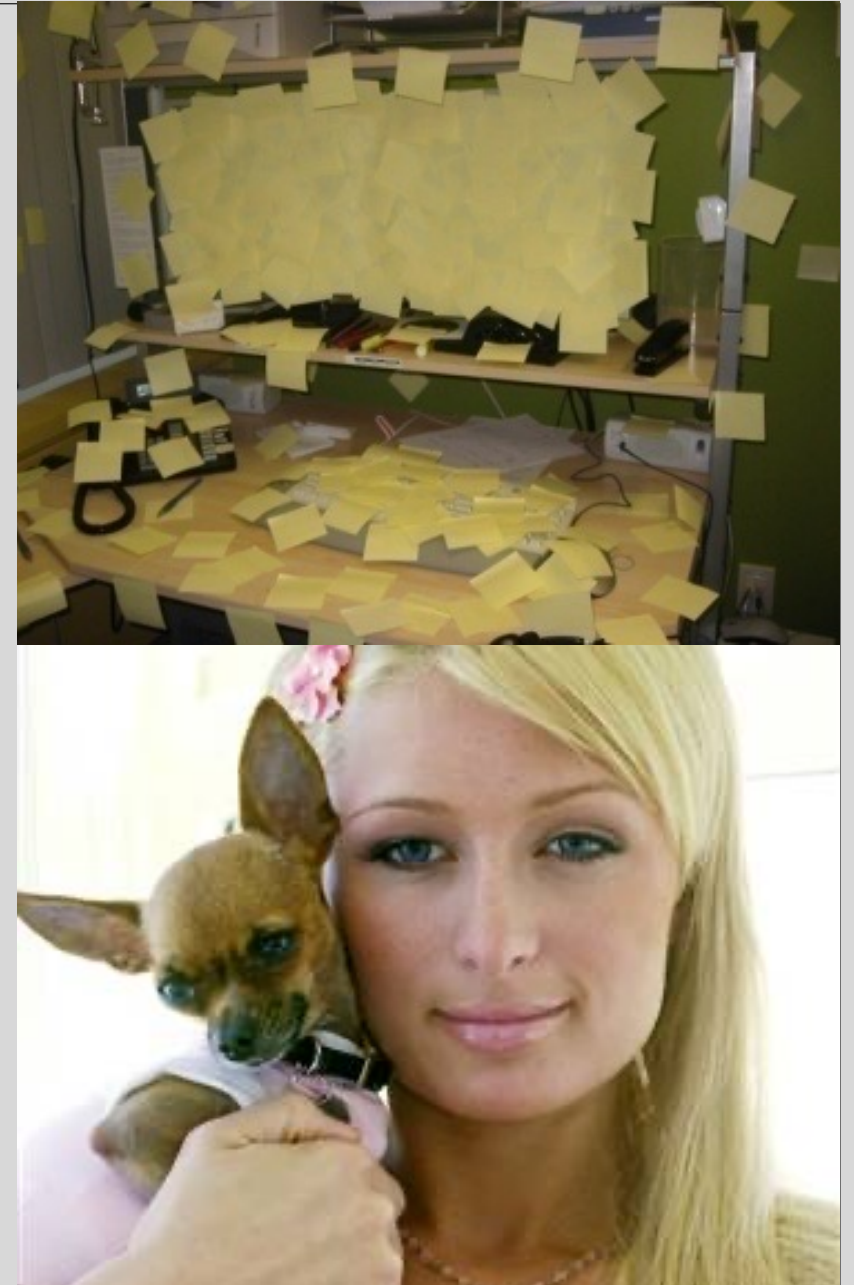
Write them down

Use a single password at multiple sites

- Do you use the same password for Amazon and your bank account? NetID? Do you remember them all?

Forget them... many services use “security questions” to reset passwords

- “What is your favorite pet’s name?”
- Paris Hilton’s T-Mobile cellphone hack



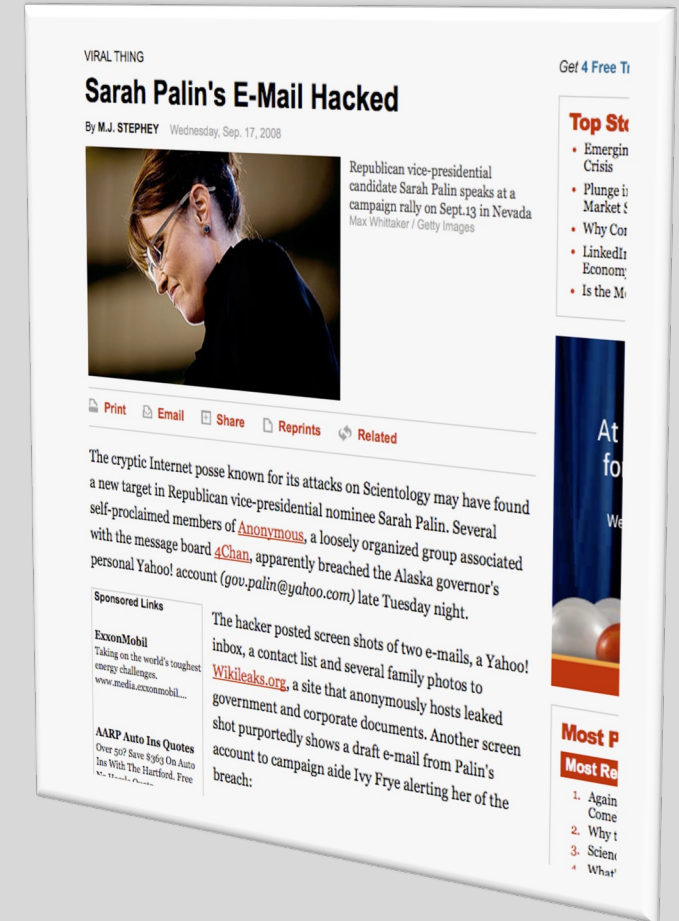
# Sara Palin's Email Hack

Reset password for gov.palin@yahoo.com

- No secondary email needed
- Date of birth? [Wikipedia](#)
- ZIP code? [Wasilla has 2](#)
- Where did you meet your spouse? [Wikipedia](#), [Google](#), ...

Changed pwd to "popcorn"

Hacker sentenced to 1 year in prison +  
3 yrs of supervised release



slide: Gustav Rydstedt

# Problems with Security Questions

## Inapplicable

- What high school did your spouse attend?

## Not memorable

- Name of kindergarten teacher? Price of your first car?

## Ambiguous

- Name of college you applied to but did not attend?

## Easily guessable

- Age when you married? Year you met your spouse? Favorite president? Favorite color?

## Automatically attackable (using public records!)

# Answers Are Easy to Find Out... Or Easy to Forget

- Make of your first car?
  - Until 1998, Ford had >25% of market
- First name of your best friend?
  - 10% of males: James/Jim, John, Robert/Bob/Rob
- Name of your first / favorite pet?
  - Max, Jake, Buddy, Bear...
  - Top 500 (covers 65% of names) available online
- Information available from Facebook, etc.
  - Where you went to school, college athletic rivals, favorite book/movie/pastime, high school mascot
- Name of the street, etc.
  - More than one
- Name of your best friend?
  - Friends change
- City where you were born?
  - NYC? New York? Manhattan? New York City? Big Apple?
- People lie to increase security... then forget the answers

# HealthCare.gov

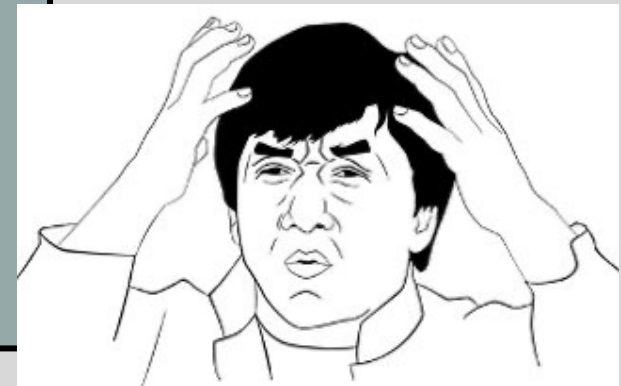


## Federal:

- What is a relative's telephone number that is not your own?
- Type a significant date in your life?
- What is the name of the manager at your first job?

## Individual states:

- What is your youngest child's birth weight?
- What color was your first bicycle?
- If you needed a new first name, what would it be?
- What band poster did you have on your wall in high school?
- How many bones have you broken?

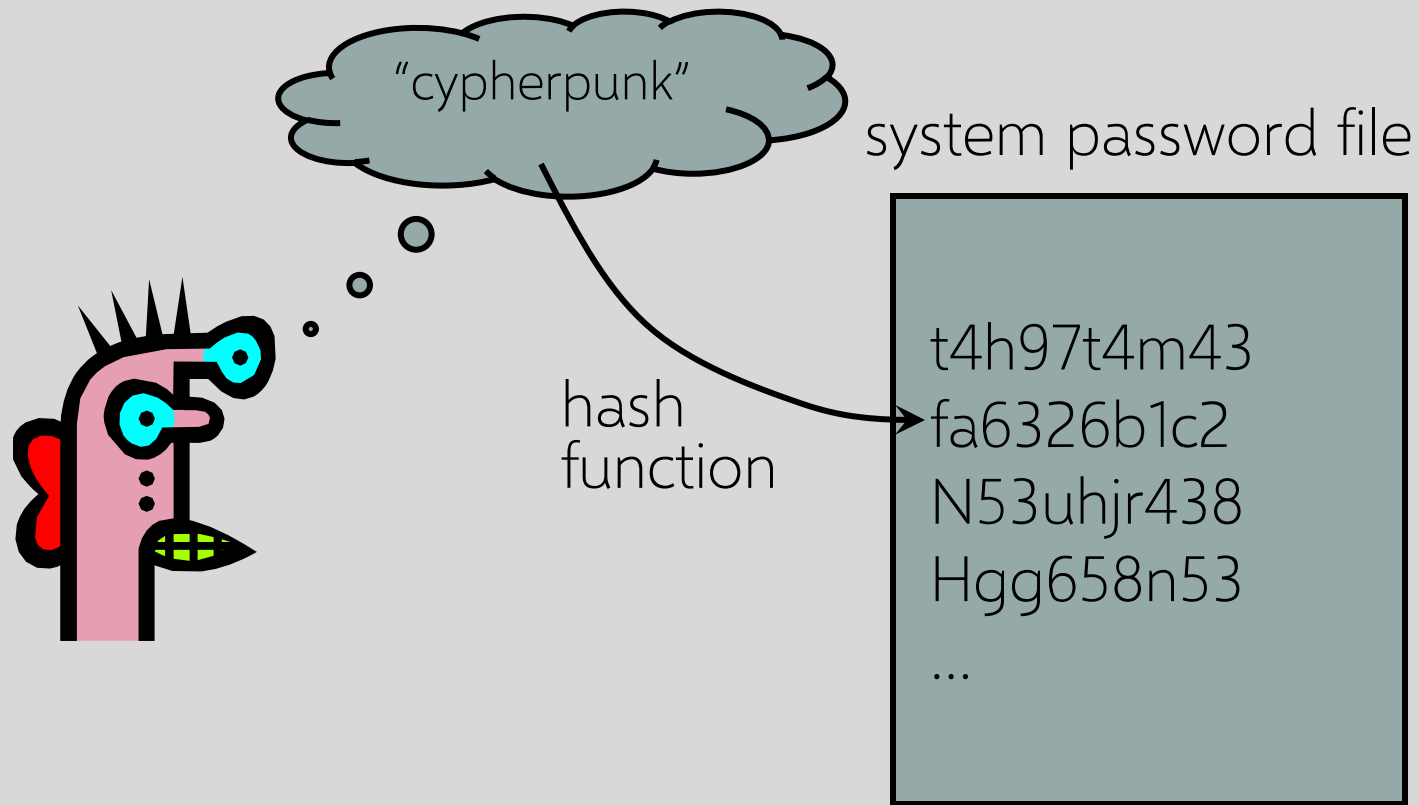


# Password Management

Countermeasure	Purpose
Password hashing	Database leak doesn't immediately reveal user passwords; slows <b>offline guessing attacks</b>
Strength meters	Nudge / force users to pick stronger passwords to mitigate guessing attacks
Lockout after N failed attempts	Prevent remote guessing attacks (X typically 10, 100, 1000); slows down / prevents <b>online guessing attacks</b>
Compromised credential checks	Check if password is in known breaches



# Storing Passwords



# Password Hashing

Instead of user's password, store

`Hash(password)`

When user enters a password, compute its hash and compare with the entry in the password file

- System does not store actual passwords
- Cannot go from hash to password (except by guessing the password)



Hash function  $H$  must have some properties

# Cryptographic Hash Functions

Cryptographic hash function  $H$  maps any message to a short digest (e.g., 256-bit string)

- One-way:

Given  $y = H(M)$ , hard to compute  $M$

- Collision-resistant:

Can't find  $M, M'$  s.t.  $H(M) = H(M')$

Good hash functions: SHA256, SHA512, SHA-3, bcrypt, scrypt, PBKDF2 ...

specifically for passwords

~~◦ Deprecated hash functions: MD5, SHA-1~~

# Dictionary Attacks



## Passwords are not random



- With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are  $94^8 \approx 6$  quadrillion possible 8-character passwords
- Humans like to use dictionary words, human and pet names  $\approx 1$  million common passwords

Attacker can pre-compute  $H(\text{word})$  for every word in the dictionary – do this once offline

- Once password file is obtained, cracking is instantaneous
- Sophisticated password guessing tools are available (take into account frequency of letters, password patterns, etc.)

# Brute-Force Password Cracking

```
[DaleGribble% openssl speed sha256  
Doing sha256 for 3s on 16 size blocks: 16553803 sha256's in 3.00s  
Doing sha256 for 3s on 64 size blocks: 9314565 sha256's in 3.00s  
Doing sha256 for 3s on 256 size blocks: 4382195 sha256's in 3.00s  
Doing sha256 for 3s on 1024 size blocks: 1382599 sha256's in 3.00s  
Doing sha256 for 3s on 8192 size blocks: 187044 sha256's in 3.00s  
Doing sha256 for 3s on 16384 size blocks: 94277 sha256's in 3.00s
```

~450,000 hashes per second

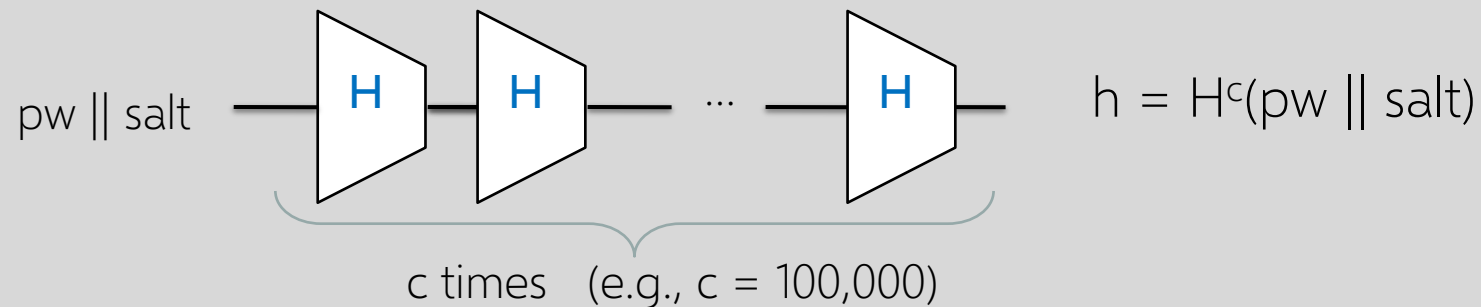
How many guesses / hashes needed to crack a password?

Also **rainbow tables**:

precompute huge number of hashes to make a quick-lookup table

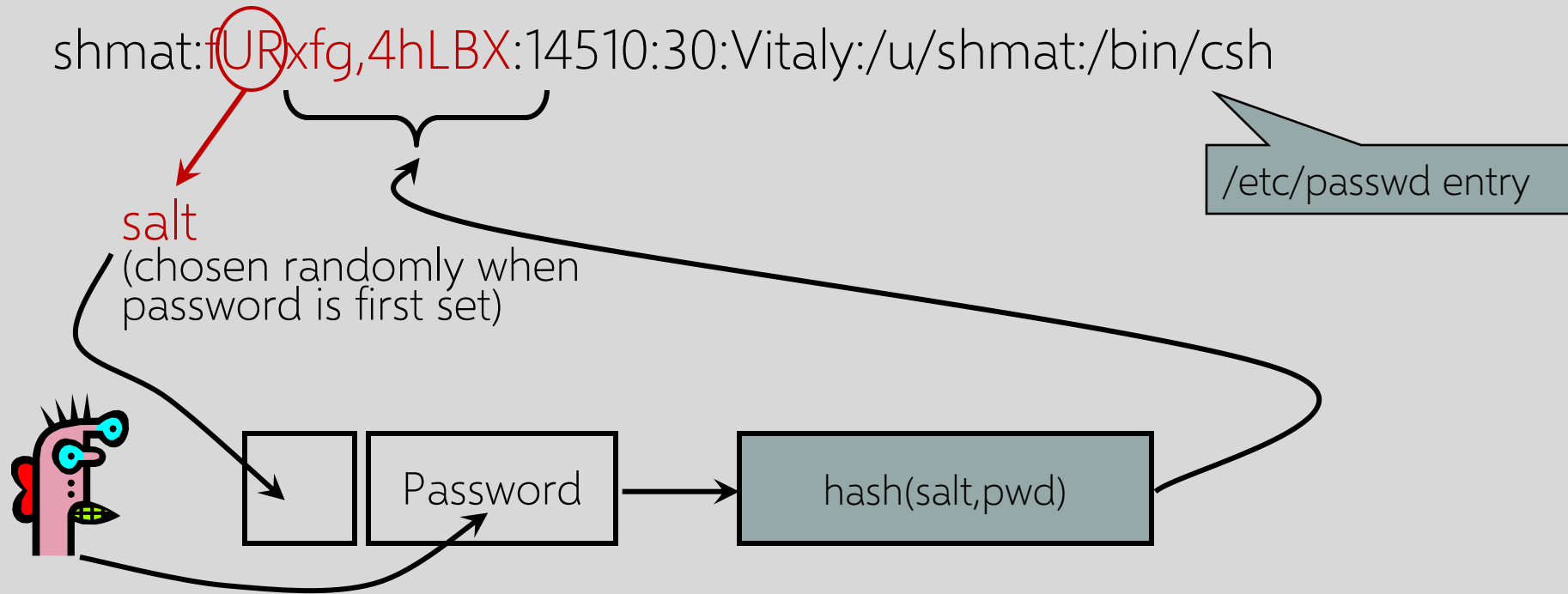
# Making Cracking Harder

- Make hashing slower to slow down cracking attacks
- Use random per-user **salts** to prevent use of rainbow tables
- PKCS#5 approach:



- Memory-hard hashing: Scrypt and argon2 require lots of memory to compute as well as time

# Salt



- Users with the same password have different entries in the password file
- Offline dictionary attack becomes much harder



# Advantages of Salting

Without salt, attacker can pre-compute hashes of all common passwords once

- Same hash function on all UNIX machines; identical passwords hash to identical values
- One table of hash values works for all password files

With salt, attacker must compute hashes of all common passwords for each possible salt value

- With 12-bit random salt, the same password can hash to 4096 different hash values

# Modern Hash Cracking

Hash type	Hashes / second	Passwords / month for 10M set <sup>3</sup>	Brute force equivalent <sup>4</sup>
MD5 unsalted	~50G	~130,000,000G	~8-9 characters
MD5 salted <sup>5</sup>	~50G	~13G	~5 characters
MD5crypt (= salted, 1,000 x MD5)	~22M	~5.6M	~3-4 characters
Bcrypt (= salted, work factor 8)	~3500	~900	~1-2 characters

... with custom GPU and FPGA hardware

IBM X-Force "Cracken"  
(circa 2017)



<https://medium.com/@ScatteredSecrets/how-to-crack-billions-of-passwords-6773af298172>

# Measuring Password Strength

Hashing slows down but does not prevent guessing attacks

Deprecated approaches for measuring password strength

- NIST entropy estimate
- Shannon entropy

Today: strength meters based on **guess ranks**

# Shannon Entropy

- Let  $\mathcal{X}$  be password distribution
- Passwords are drawn iid from  $\mathcal{X}$
- $N$  is size of support of  $\mathcal{X}$
- $p_1, p_2, \dots, p_N$  are probabilities of passwords in decreasing order

Shannon entropy:

$$H_1(\mathcal{X}) = \sum_{i=1}^N -p_i \log p_i$$

# Poor Measure of Guessability

$$N = 1,000,000$$

$$p_1 = 1 / 100$$

$$p_2 = (1 - 1/100)/999,999 \approx 1 / 2^{20}$$

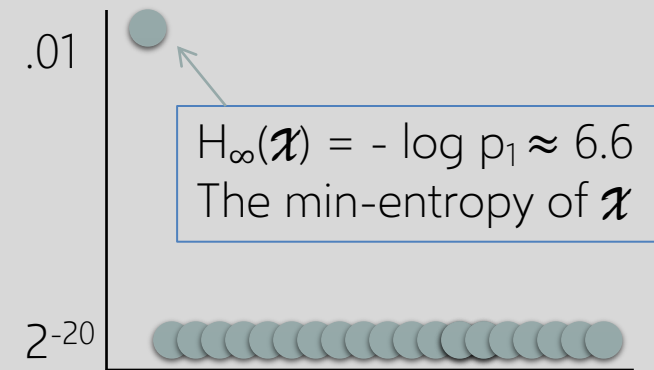
...

$$p_N = (1 - 1/100)/999,999 \approx 1 / 2^{20}$$

$$H_1(\mathcal{X}) \approx 19$$

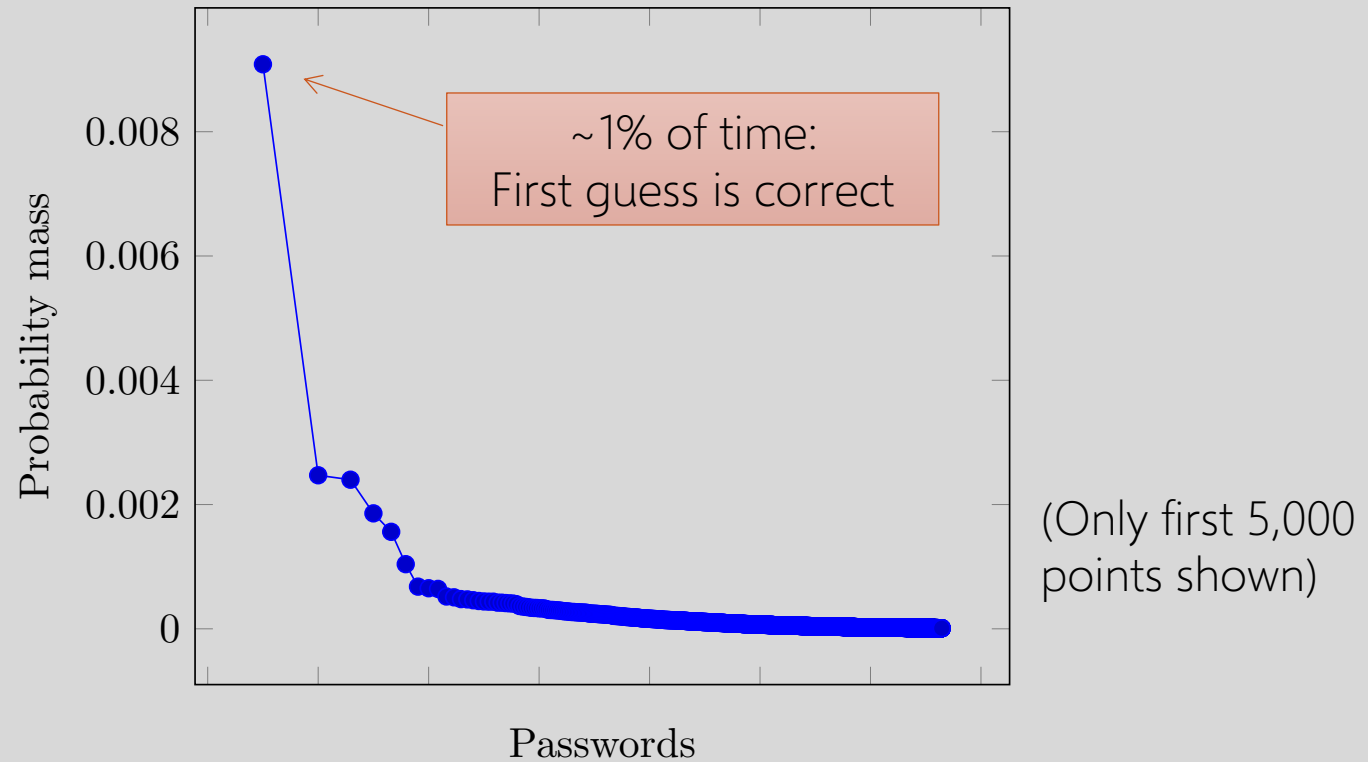
19 bits of “unpredictability”. Probability of success about  $1/2^{19}$ ?

What is probability of success if attacker makes one guess?



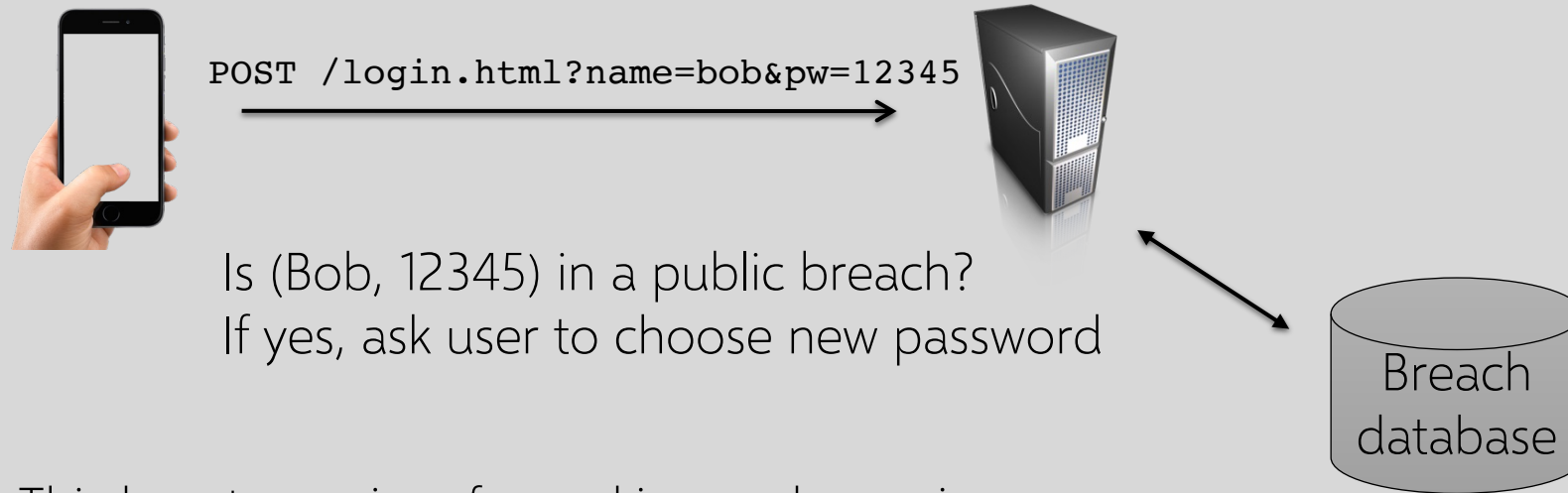
**Shannon entropy is almost never useful measure for security**

# RockYou Empirical Probability



# Credential Stuffing (Password Spraying)

Attacker tries multiple credentials from known breaches



Third-party services for making such queries:

- HavelBeenPwned
- Google password checker



# Have I Been Pwned

556

pwned websites

11,454,726,823

pwned accounts









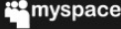

114,131

pastes











207,749,076

paste accounts

## Largest breaches

	772,904,991 <a href="#">Collection #1 accounts</a>
 <small>verifications.io</small>	763,117,241 <a href="#">Verifications.io accounts</a>
	711,477,622 <a href="#">Onliner Spambot accounts</a>
	622,161,052 <a href="#">Data Enrichment Exposure From PDL Customer accounts</a>
	593,427,119 <a href="#">Exploit.In accounts</a>
	509,458,528 <a href="#">Facebook accounts</a>
	457,962,538 <a href="#">Anti Public Combo List accounts</a>
	393,430,309 <a href="#">River City Media Spam List accounts</a>
	359,420,698 <a href="#">MySpace accounts</a>
	268,765,495 <a href="#">Wattpad accounts</a>

## Recently added breaches

	20,154,583 <a href="#">IndiaMART accounts</a>
	878,209 <a href="#">Imavex accounts</a>
	6,137,666 <a href="#">SubaGames accounts</a>
	2,789,609 <a href="#">Eatigo accounts</a>
	1,304,447 <a href="#">OrderSnapp accounts</a>
	2,660,295 <a href="#">MMG Fusion accounts</a>
	2,743,539 <a href="#">Audi accounts</a>
	112,031 <a href="#">Guntrader accounts</a>
	505,466 <a href="#">Short Édition accounts</a>
	30,433 <a href="#">Vastaamo accounts</a>

# STRONTIUM Attacks

aka Fancy Bear, aka APT28, aka Sofacy  
(GRU, Russian military intelligence)



Microsoft has tied STRONTIUM to a newly uncovered pattern of Office365 credential harvesting activity aimed at US and UK organizations directly involved in political elections

*target: elections*

...

*spear-phishing to harvest credentials*

*password spraying*

STRONTIUM relied heavily upon spear phishing in its credential harvesting efforts leading up to the 2016 US presidential election. In 2016, spear-phishing was the most common tactic for stealing credentials from targeted accounts. This time around, STRONTIUM appears to be taking a different approach, namely, brute-force/password-spray tooling. This shift in tactics, also made by several other nation-state actors, allows them to execute large-scale credential harvesting operations in a more anonymized manner. The tooling STRONTIUM is using routes its authentication attempts through a pool of approximately 1,100 IPs, the majority associated with the Tor anonymizing service.

*cover tracks using Tor*

<https://www.microsoft.com/security/blog/2020/09/10/strontium-detecting-new-patterns-credential-harvesting/>

# Credential Tweaking Attacks

Suppose user changes password to 123456

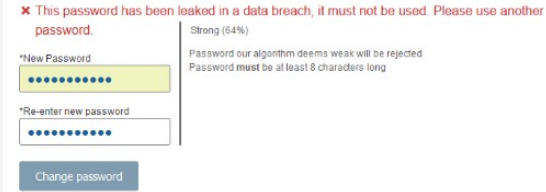
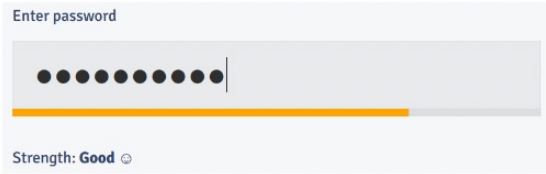
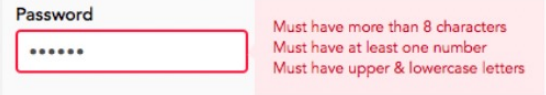
Credential stuffing no longer works, but guessing attacker could try variants of 12345

Deep learning techniques to learn conditional probability distribution

- $p(\text{pw}' \mid \text{pw})$  where pw is leaked password, pw' is variant
- Trained from leak data to capture typical password variants

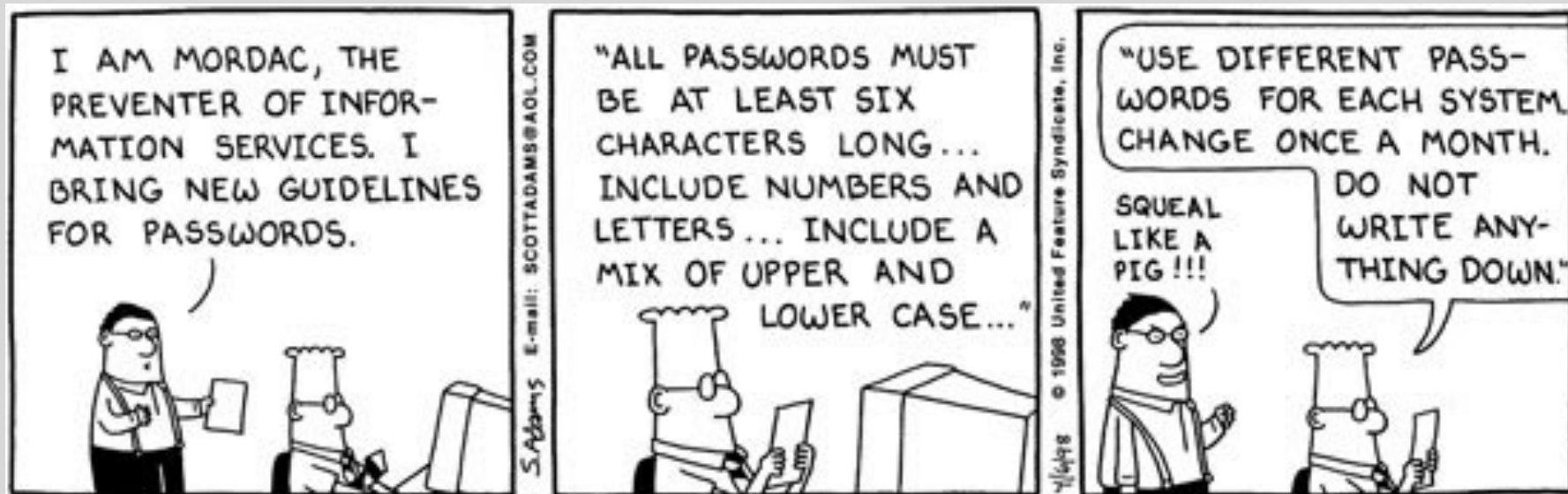
Experiments showed that 1,316 Cornell accounts vulnerable (Pal et al. 2019)

# Password policies of the 120 most popular English-language websites

Interventions	Best practices from prior research	Our key findings
<b>Blocklists</b> 	<ul style="list-style-type: none"> <li>Do check users' passwords against lists of leaked and easily-guessed passwords [1, 2, 3, 4].</li> <li>Do reject the password if it appears on a blocklist, prompt the user to select a different password [1, 4].</li> </ul>	<ul style="list-style-type: none"> <li>More than half (71 / 120) of websites do not check passwords at all, allowing all 40 of the most common passwords we tested (e.g., "12345678", "rockyou").</li> <li>19 more websites block less than half of the most common passwords we tested.</li> </ul>
<b>Strength meters and minimum-strength requirements</b> 	<ul style="list-style-type: none"> <li>Do provide real-time password strength estimates [5, 6, 7].</li> <li>Do set minimum-strength requirements by estimating guessability (the number of guesses it would take for an adversary to crack the password) [3, 8, 9, 10, 11].</li> </ul>	<ul style="list-style-type: none"> <li>Only 23 / 120 websites used password strength meters.</li> <li>Of those 23, 10 websites misuse meters as nudges toward specific types of characters and do not incorporate any notion of guessability.</li> </ul>
<b>Composition policies</b> 	<ul style="list-style-type: none"> <li>Do not require specific character classes; let users freely construct passwords [2, 3, 7, 12].</li> <li>NIST: Do set a minimum-length of at least 8 characters.</li> </ul>	<ul style="list-style-type: none"> <li>54 / 120 sites still require specific character classes such as digits or special characters.</li> <li>We devised a new method to measure the security and usability of composition policies. Based on our method, we found that all 120 policies performed poorly: none provided <math>\geq 60\%</math> security and usability simultaneously.</li> </ul>

Lee et al. "Password policies of most top websites fail to follow best practices" (2022)

# Password Policies



# Restrictive Password Policies Don't Help

Overly restrictive password policies...

- 7 or 8 characters, at least 3 out of {digits, upper-case, lower-case, non-alphanumeric}, no dictionary words, change every 4 months, password may not be similar to previous 12 passwords...

... result in frustrated users and less security

- Burdens of devising, learning, forgetting passwords
- Users construct passwords insecurely, write them down
  - Can't use their favorite password construction techniques (small changes to old passwords, etc.)
  - "An item on my desk, then add a number to it"
- Heavy password re-use across systems

*Inglesant and Sasse, "The True Cost of Unusable Password Policies"*

# Password Management

5 minutes to brainstorm ideas for how to improve password-based authentication



# Managing Credentials

VP and Distinguished Engineer at AWS  
Works on identity and cryptography services



**Colm MacCárthaigh**  
@colmmacc



A quick rage-thread about credentials. When security auditors just say things like "Critical credentials need to be rotated every 90 days" you need to fire them into the sun with urgency. Here's what you actually need ...

1:57 PM · Jun 1, 2022 · Twitter Web App

<https://twitter.com/colmmacc/status/1532058883908198401>

1. Rotation does nothing. It's revocation that matters. You always need a well-tested mechanism to make sure that you can remove or invalidate a credential that has been compromised.
2. Have closed loops. Deactivated credentials are a common source of outages. When introducing a new credential you see it everywhere it needs to be before using it. When you remove one, you need to see it gone from use before deactivating.
3. Logging and detective controls are key. You need to be able to see when and where a credential is being used. This is important for operational safety and security. How would you even detect a stolen credential without this?
4. Be INCREDIBLY wary of time-based expiry. Use only when there is no other option (e.g. public SSL certificates). There's really no way to win with time-based expiry. If your expiry time is something like a year, you don't get much security. Are you ok with an attacker using that cred for a year? So you still need revocation. If your expiry time is very short, like hours, are you *\*always\** going to beat that renewal deadline? got good clocks?
5. Store credentials only where they are needed. This seems obvious but is rarely done. In particular it's common to see "treasure trove" secret-distribution control-planes that know all of the credentials.

*<https://twitter.com/colmmacc/status/1532058883908198401>*

6. If there is no reason to suspect credential disclosure or misuse, leave it alone. Replacing credentials usually exposes them to more systems, at least temporarily.
7. Asymmetric cryptography when you can, if not then choose between either memory-hard compute-hard hashing or derived-key symmetric auth depending on what fits your use-case. Avoid storing valuable secrets server side..
8. Keep credentials inside one-way enclaves like TPMs, TEEs, HSMs, when you can. Best line of defense is to keep credentials inaccessible.
9. If you can't write down a common password comparison side-channel from memory, do not implement your own authentication.
10. Check for all-zeroes creds, and repeated values. You can do this with hashing, you don't need to record the secrets. Coding errors, failures of entropy systems, and erasure mistakes are common enough to make this check worth doing.

***All attacks on authentication we'll see in this course violate one or more of these rules***

*<https://twitter.com/colmmacc/status/1532058883908198401>*

# Multi-Factor Authentication

1. Sign in with your Google Account

Email: hikingfan@gmail.com  
ex: pat@example.com

Password:

☒ Stay signed in

[Can't access your account?](#)

2. Google accounts

**Enter verification code**

To verify your identity on this computer, enter the verification code generated by your mobile application.

Enter code:

☒ Remember verification for this computer for 30 days.

[Other ways to get a verification code »](#)

Google Authenticator

966286  
wileyc@acme.com

001232

Cornell University

CUWebLogin

W4115:

PASSWORD:

Cornell University

Two-Step Login

Access to this site requires [Two-Step Login](#).

If you have not yet enrolled any devices, please visit [twostep.netid.cornell.edu](#).

Don't have any of your devices or a passcode with you? Contact the IT Service Desk: [it.cornell.edu/support](#)

Device:

Choose an authentication method

☒ Duo PUSH notifications ☐ Send Me a Push

☐ Call Me ☐ Enter a Passcode

☐ Remember me for 24 hours



Turn on Login Approvals

**What is Login Approvals?**

Login Approvals is a security feature that requires you to enter a code that we text to your phone when you log in from an unrecognized computer. You can enable this feature in a few simple steps.

If you ever lose access to your phone, you can always return to a previously-recognized computer to regain access to your account.

**Note:** You'll need to have your mobile phone with you to complete this process.

# Factors for 2FA

Combine passwords with another way to authenticate user

Second factor is usually proof of ownership of ...

- Email address
- Telephone number (via SMS)
- Device (via authenticator app)
- Hardware token (one-time-password token, universal second factor U2F token)



## Effectiveness of 2FA

Microsoft: **99.9%** of compromised accounts did not use multi-factor authentication

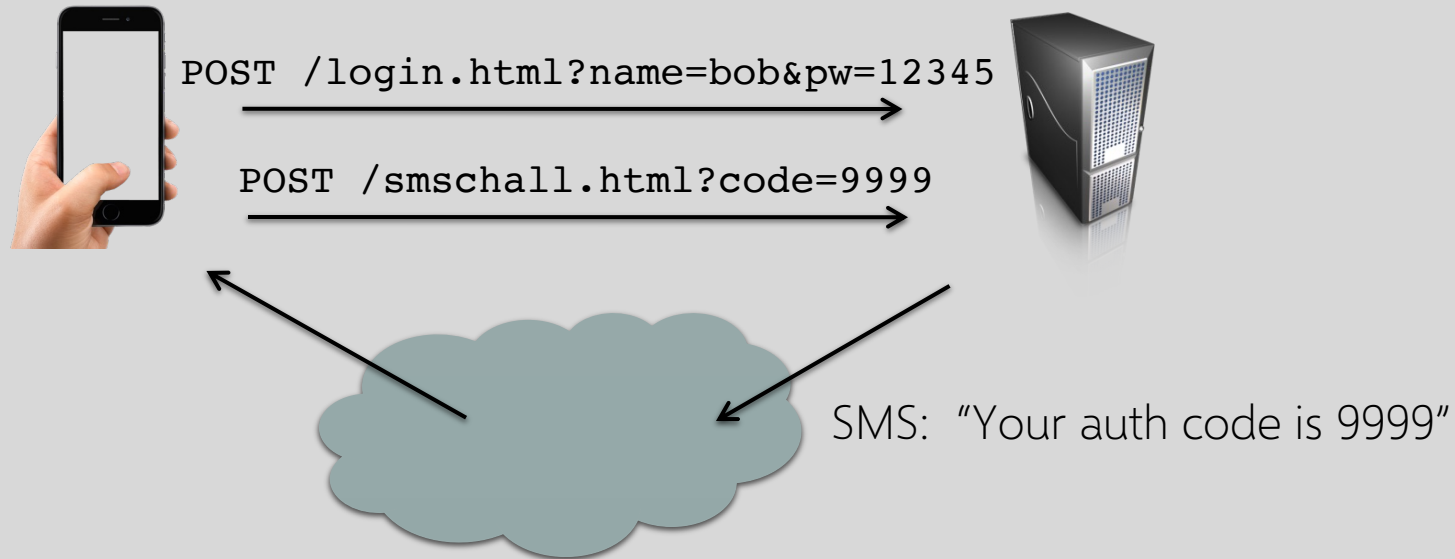
Only 11% of all enterprise accounts use a MFA solution overall.

Microsoft report, Mar 2020

successfully auto-enabled 2SV for over 150 million people, and we've also required it for over 2 million of our YouTube creators. As a result of this effort, we have seen a **50% decrease in accounts being compromised** among those users.

Google report, Feb 2022

# SMS Authentication



Suppose you know someone's password (e.g., due to breach) but their account is protected by SMS-based 2FA. **What can you do as an attacker?**



# Circumventing SMS-Based 2FA

- Have physical access to device that receives SMS
- Phishing attacks: confuse or trick user into disclosing SMS to you
- SIM swap: trick phone company into registering victim's phone # to your device
- SMS hijacking: exploit vulnerabilities in cellular network
  - <https://berlin.ccc.de/~tobias/31c3-ss7-locate-track-manipulate.pdf>
  - [Doerfler et al. 2019]: SMS 2FA circumvented in ~4% of phishing attacks, ~26% of targeted attacks
- Better practice: authenticator app or hardware token

# SMS Rerouting

## **A Hacker Got All My Texts for \$16**

A gaping flaw in SMS lets hackers take over phone numbers in minutes by simply paying a company to reroute text messages.



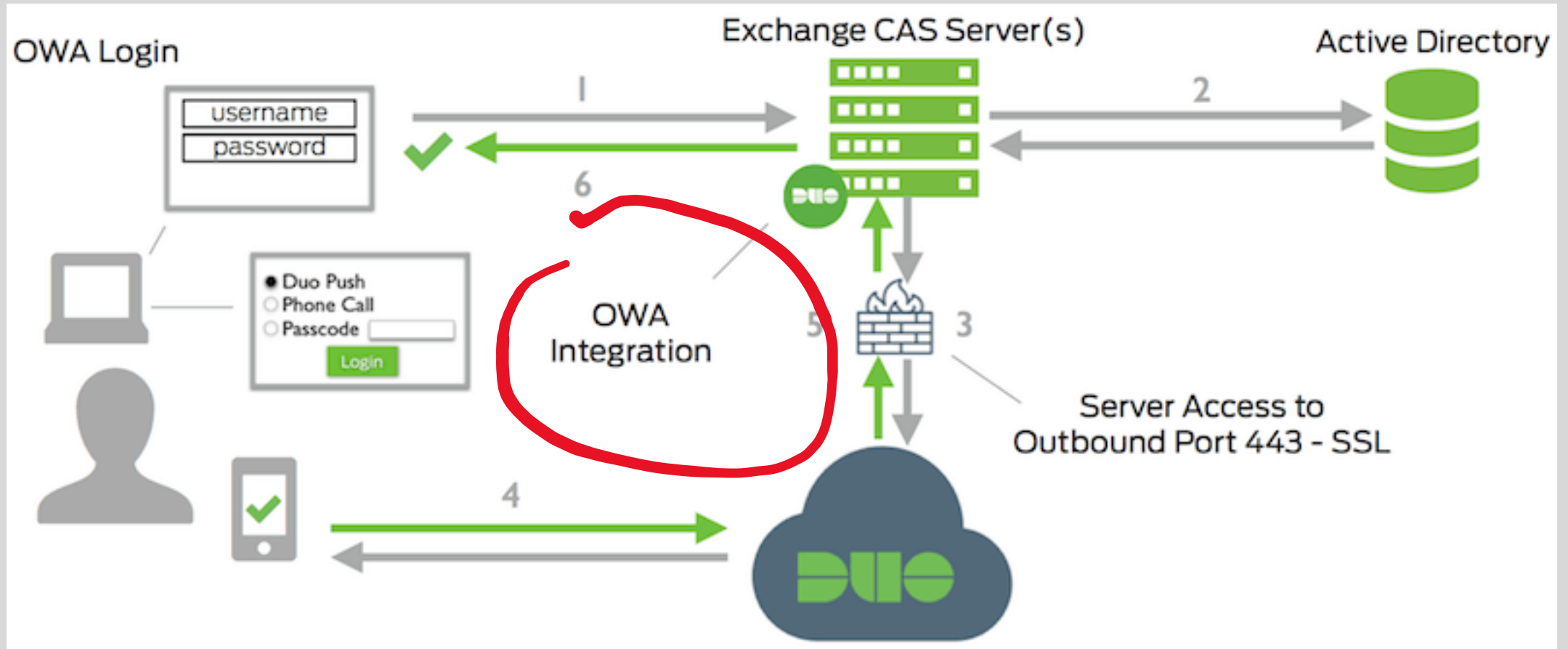
"I used a prepaid card to buy their \$16 per month plan and then after that was done it let me steal numbers just by filling out LOA info with fake info," Lucky225 added, referring to a Letter of Authorization, a document saying that the signer has authority to switch telephone numbers.

Not SIM swapping or phone number hijacking  
The victim doesn't notice (phone service not interrupted)

The OSR database is a core component of the message routing infrastructure in North America ... It is known as an "override registry" as it enables an individual subscriber to receive messaging services from a different provider than voice services.

*<https://www.vice.com/en/article/y3g8wb/hacker-got-my-texts-16-dollars-sakari-netnumber>*

# Duo 2FA Authentication



# 2020 SolarWinds Hack

Compromised computer networks across the US government

- Pentagon, State, Treasury, Energy, Justice, Commerce, Labor, DHS, NIH ...

Many state and local governments

Many corporations

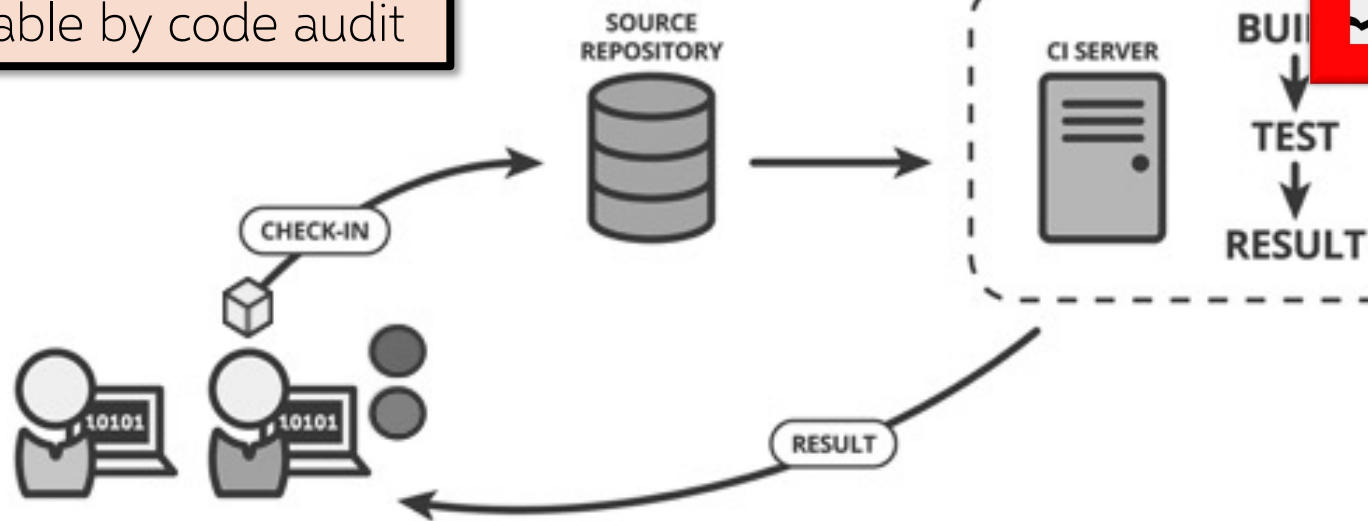
- Microsoft, Intel, Cisco, network security firms such as FireEye

18,000 customers in total



# Software Development: Continuous Integration

No modification to source code, thus attack not detectable by code audit

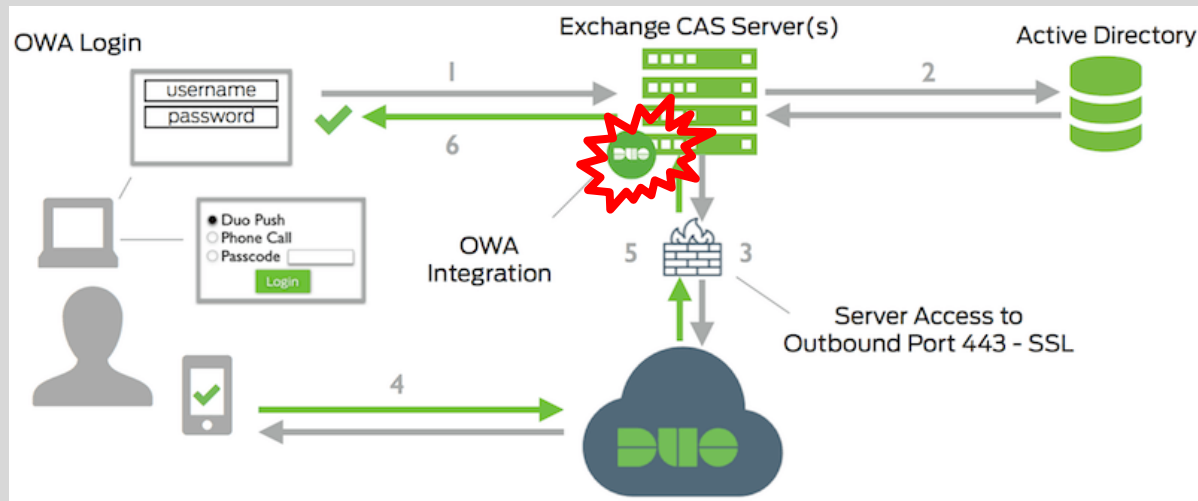


Malicious functionality introduced into the executable code during the build process

Deployed to all customers of Orion network management product

Image: <https://techbeacon.com/app-dev-testing/are-you-really-doing-continuous-integration-heres-how-tell>

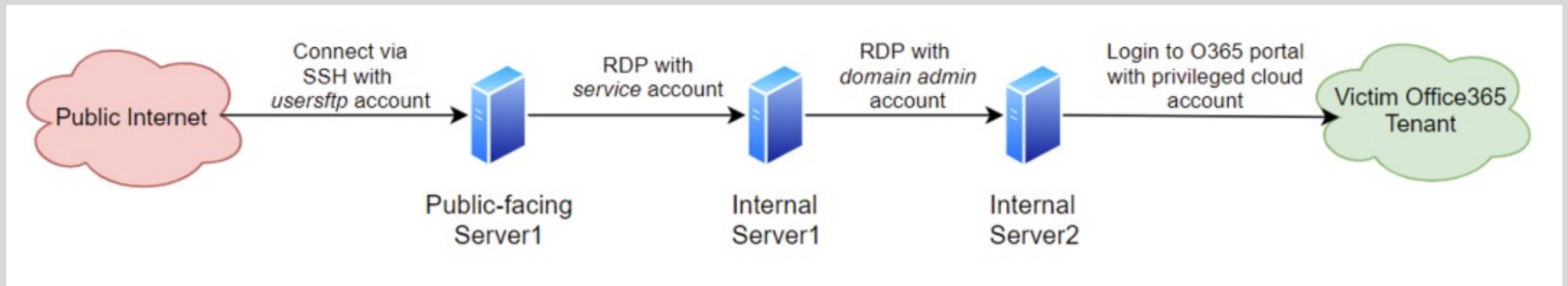
# SolarWinds hackers have a clever way to bypass multifactor authentication



... the attacker had accessed the Duo integration secret key (akey) from the OWA server. This key then allowed the attacker to derive a pre-computed value to be set in the duo-sid cookie. After successful password authentication, the server evaluated the duo-sid cookie and determined it to be valid. This allowed the attacker with knowledge of a user account and password to then completely bypass the MFA set on the account.

More about Web authentication later

# Credential Hopping (SolarWinds Hack)



<https://www.crowdstrike.com/blog/observations-from-the-stellarparticle-campaign/>



# Over 90 percent of Gmail users still don't use two-factor authentication

*The security tool adds another layer of security if your password has been stolen*

By [Thuy Ong](#) | [@ThuyOng](#) | Jan 23, 2018, 8:30am EST

Usability remains a key issue preventing adoption



# Other Authentication Signals

## Location-based authentication

- IP-based geolocation

## Device identification

- Cookies, device fingerprinting

## Behavioral cues

- Typical actions on platform (even after authenticated)

## Biometrics

- Fingerprints, etc

# User Authentication Is a Huge Pain

Simple typos in passwords cause 3% of Dropbox users to be unable to login in a 24-hour period [Chatterjee et al. 2016]

52% of users fail login challenges at Google, 3% don't get in within short period of time [Doerfler et al. 2019]

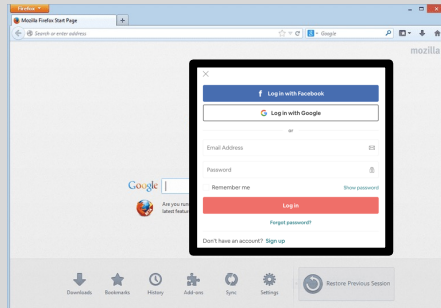
# Single Sign-On (SSO)

In-class 5-minute exercise:

How would you securely authorize browser to access `www.cornell.edu/account.html?user=vitaly` based on authentication to `identity.cornell.edu`?

Identity provider handles authentication

- Google, Facebook, proprietary services, etc.



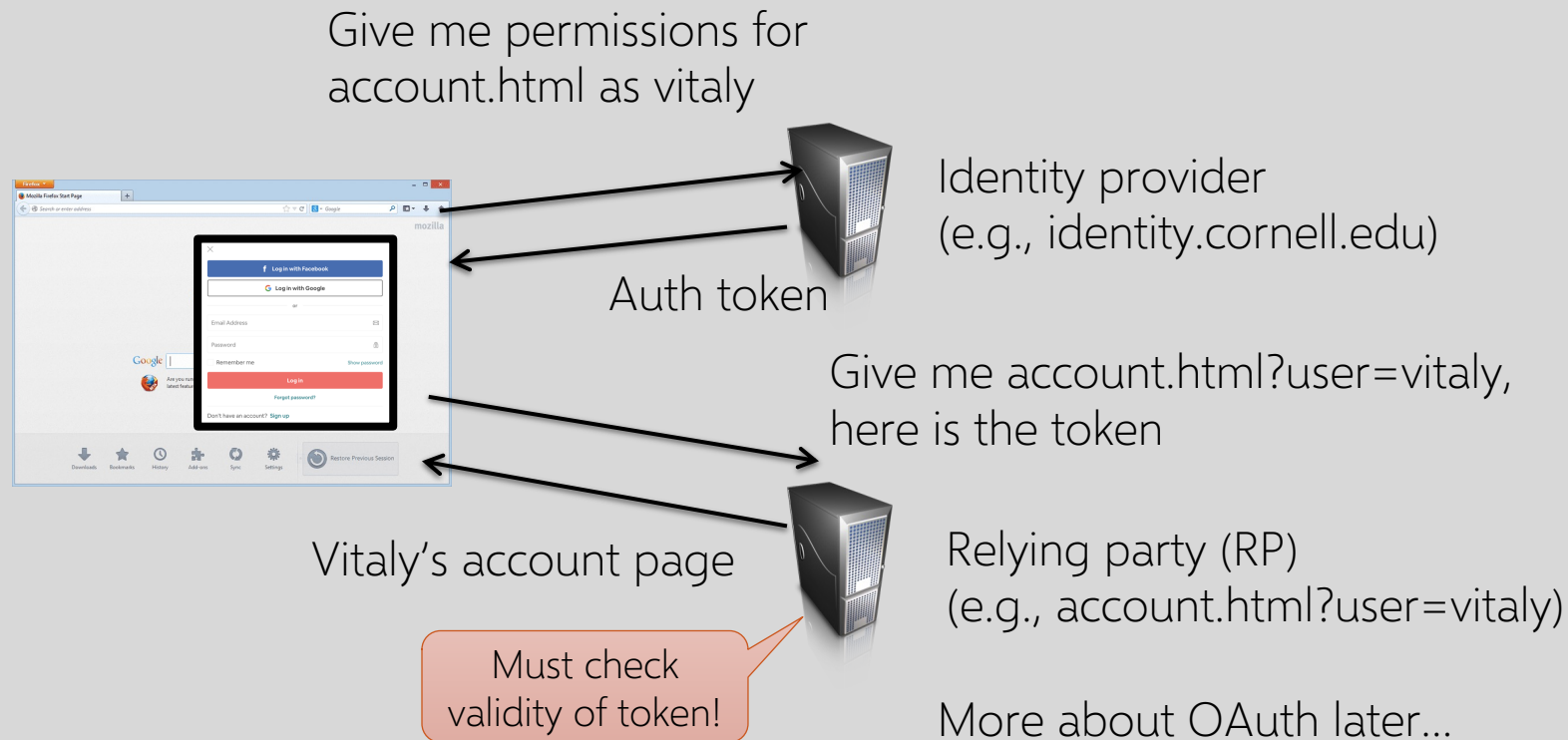
Identity provider  
(e.g., identity.cornell.edu)



Relying party (RP)  
(e.g., account.html?user=vitaly)

# Single Sign-On (SSO)

Many standards and systems: SAML, OpenID Connect + OAuth 2.0, ...



# Strengthening Passwords

## Add biometrics

- For example, keystroke dynamics or voiceprint
- **Revocation** is often a problem with biometrics

## Graphical passwords

- Goal: increase the size of memorable password space
- Dictionary attacks are believed to be difficult because images are very “random” - is this true?

# PixelPin



Upload a picture,  
use 3 or more points as the “password”

random?

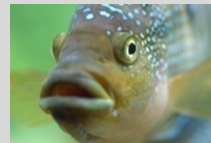
# Images + Story

Invent a story for an image  
or a sequence of images

*"We went for a walk  
in the park yesterday"*



*Fish-woman-girl-corn*



Need to remember the order!

# 'Person, woman, man, camera, TV': Trump insists cognitive test was difficult

**US president's pride in his own mental agility on display  
during interview in which he lists five things repeatedly**





# User Experiences

50% unable to invent a story, so try to pick four pleasing pictures and memorize their order

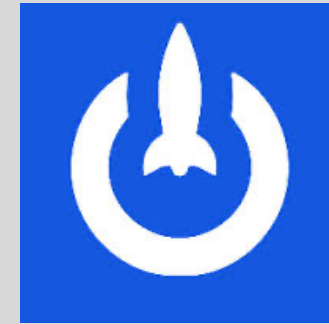
- "I had no problem remembering the four pictures, but I could not remember the original order"
- "... on the third try I found a sequence that I could remember, fish-woman-girl-corn. I would screw up the fish and corn order 50% of the time, but I knew they were the pictures"

Picture selection biases

- Males select nature and sports more than females
- Females select food images more often

# Alternatives to Passwords

Mobile phones,  
USB devices,  
special tokens,  
etc. etc.



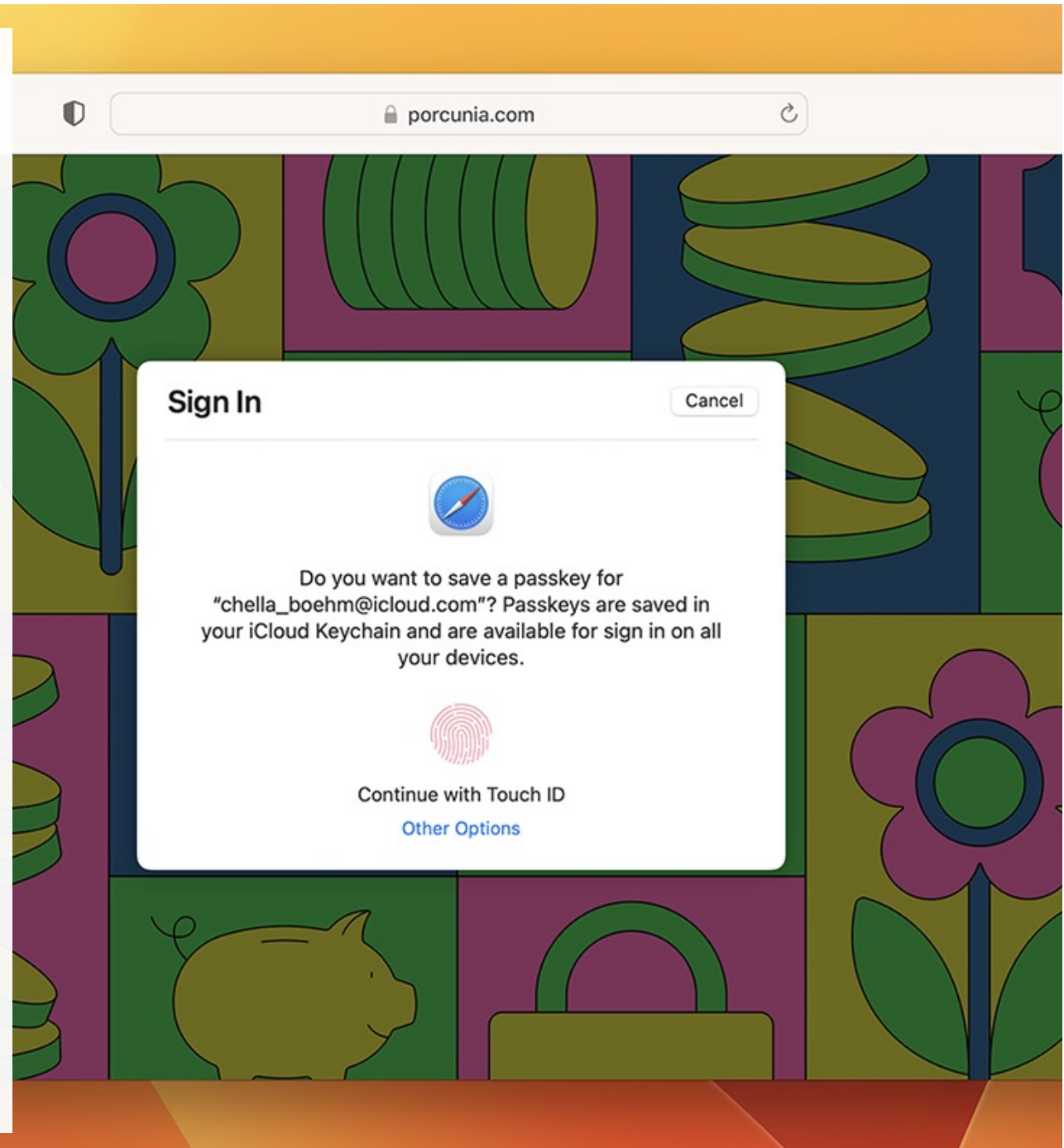
LaunchKey



*Coming in iOS 16 and  
MacOS Ventura (Fall 2022)*

# Passkeys

- Unique value for each site
- Generated by the client-side OS using biometrics (Face ID, Touch ID)
- Stored in the iCloud keychain, synchronized across devices
- Not stored on the server
- Open standards via FIDO alliance

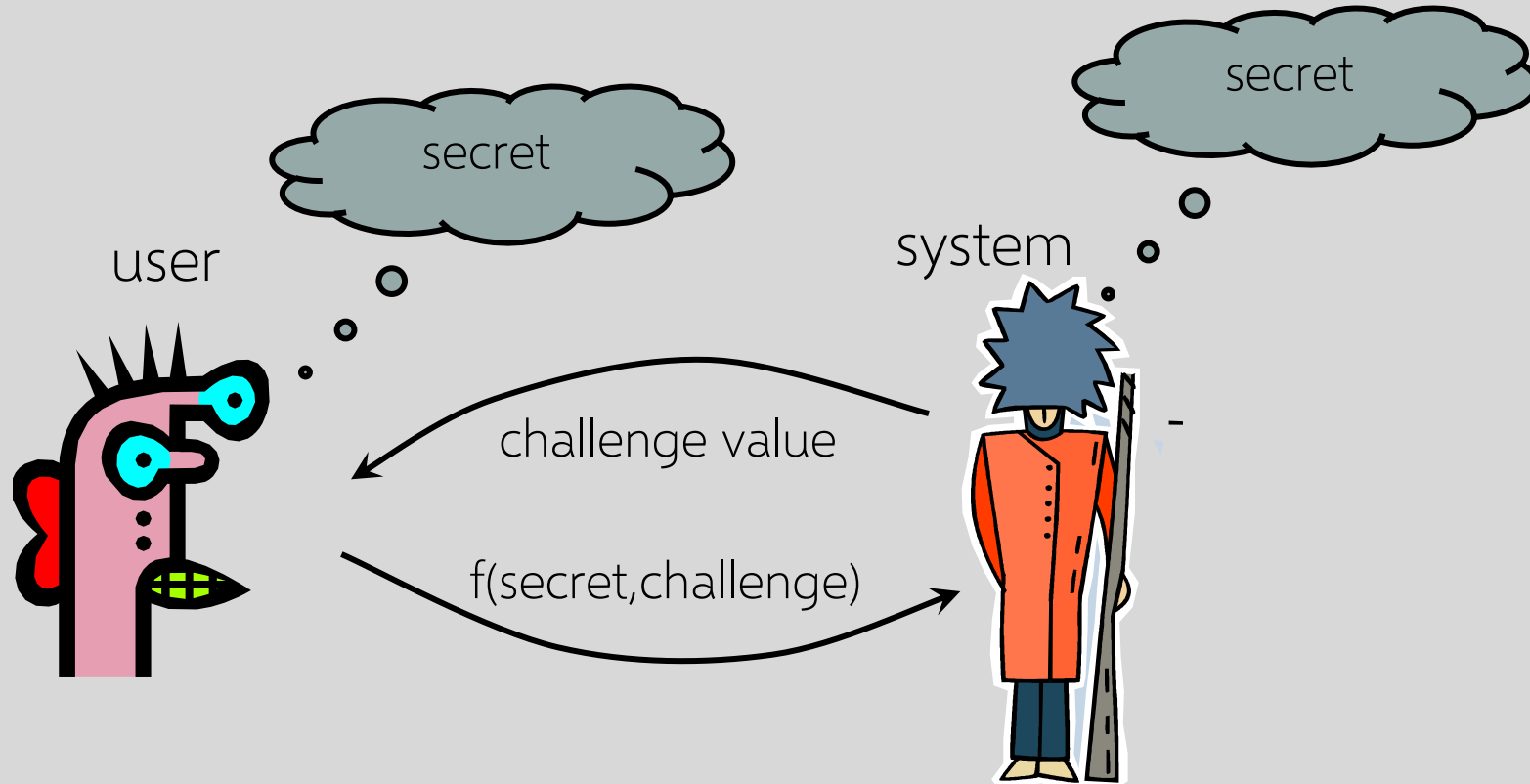


# One-Time Passwords

Idea: use a shared secret to derive a  
**one-time password**

If the attacker eavesdrops on the network, he'll learn this password but it will be useless for future logins

# Challenge-Response

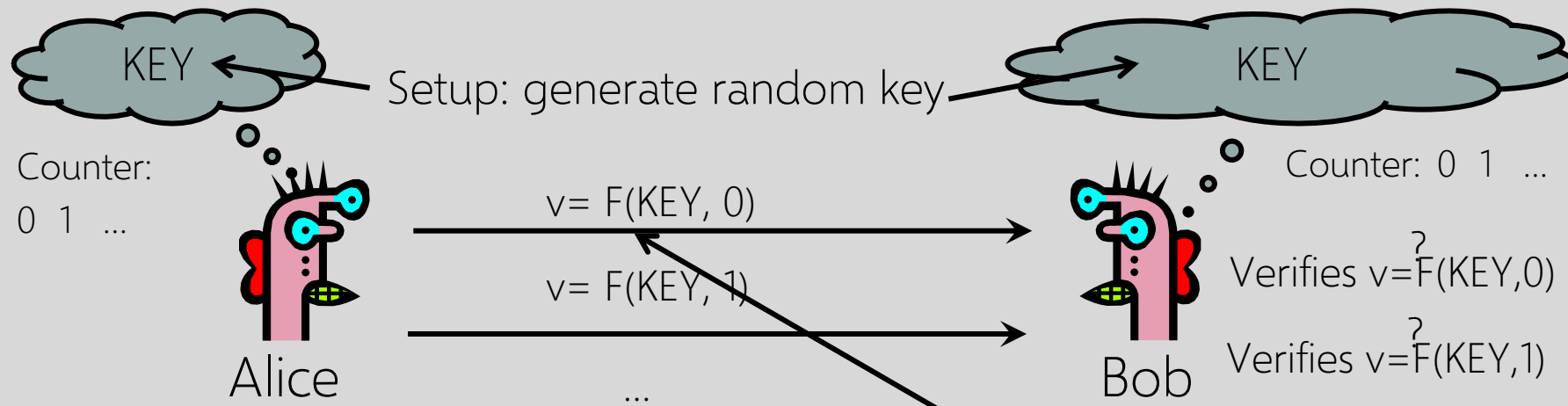


Why is this better than the password over a network?

# Challenge-Response Authentication

- User and system share a secret (key or password)
- **Challenge**: system presents user with some string
- **Response**: user computes the response based on the secret and the challenge
  - **Secrecy**: difficult to recover secret from response
    - Cryptographic hashing or symmetric encryption work well
  - **Freshness**: if the challenge is fresh, attacker on the network cannot replay an old response
    - Fresh random number, counter, timestamp....
- Good for systems with pre-installed secret keys
  - Car keys; military friend-or-foe identification

# SecurID



- Advancing the counter
  - Time-based (60 seconds) or every button press
- Allow for skew in the counter value
  - 5-minute clock skew by default

RSA uses a custom function  
Input: 64-bit key, 24-bit counter  
Output: 6-digit value