ECE458/ECE750T27: Computer Security

Authentication

Dr. Kami Vaniea
Electrical and Computer Engineering
kami.vaniea@uwaterloo.ca

First, the news...

- First 5 minutes we talk about something interesting and recent
- You will not be tested on the news part of lecture
- You may use news as an example on tests
- Why do this?
    1. Some students show up late for various good reasons
    2. Reward students who show up on time
    3. Important to see real world examples

News...

Add a file to a GitHub comment

It gets auto uploaded and given a URL within the GitHub project

Send link that looks like legit repo link, but actually is malicious

Schneier on Security

Blog | Newsletter | Books | Essays | News | Talks | Academic | About M

Home > Blog

Using Legitimate GitHub URLs for Malware

Interesting social-engineering attack vector:

> McAfee released a report on a new LUA malware loader distributed through what appeared to be a legitimate Microsoft GitHub repository for the "C++ Library Manager for Windows, Linux, and MacOS," known as vcpkg.

The attacker is exploiting a property of GitHub: comments to a particular repo can contain files, and those files will be associated with the project in the URL.

What this means is that someone can upload malware and "attach" it to a legitimate and trusted project.

> As the file's URL contains the name of the repository the comment was created in, and as almost every software company uses GitHub, this flaw can allow threat actors to develop extraordinarily crafty and trustworthy lures.
>
> For example, a threat actor could upload a malware executable in NVIDIA's driver installer repo that pretends to be a new driver fixing issues in a popular game. Or a threat actor could upload a file in a comment to the Google Chromium source code and pretend it's a new test version of the web browser.
>
> These URLs would also appear to belong to the company's repositories, making them far more trustworthy.

AUTHENTICATION

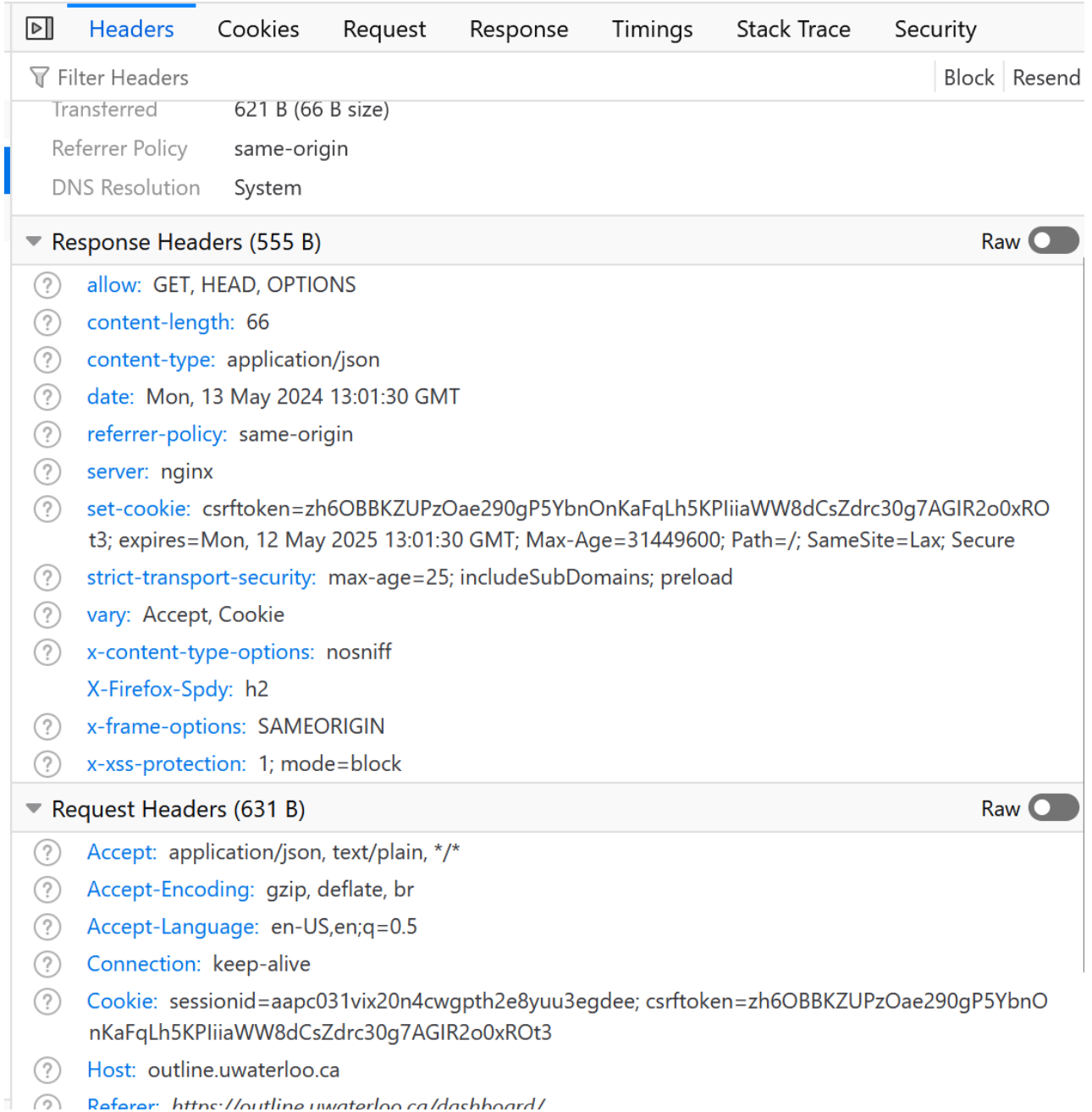
Security properties to ensure

Confidentiality	No improper information gathering
Integrity	Data has not been (maliciously) altered
Availability	Data/services can be accessed as desired
Accountability	Actions are traceable to those responsible
Authentication	User or data origin accurately identifiable

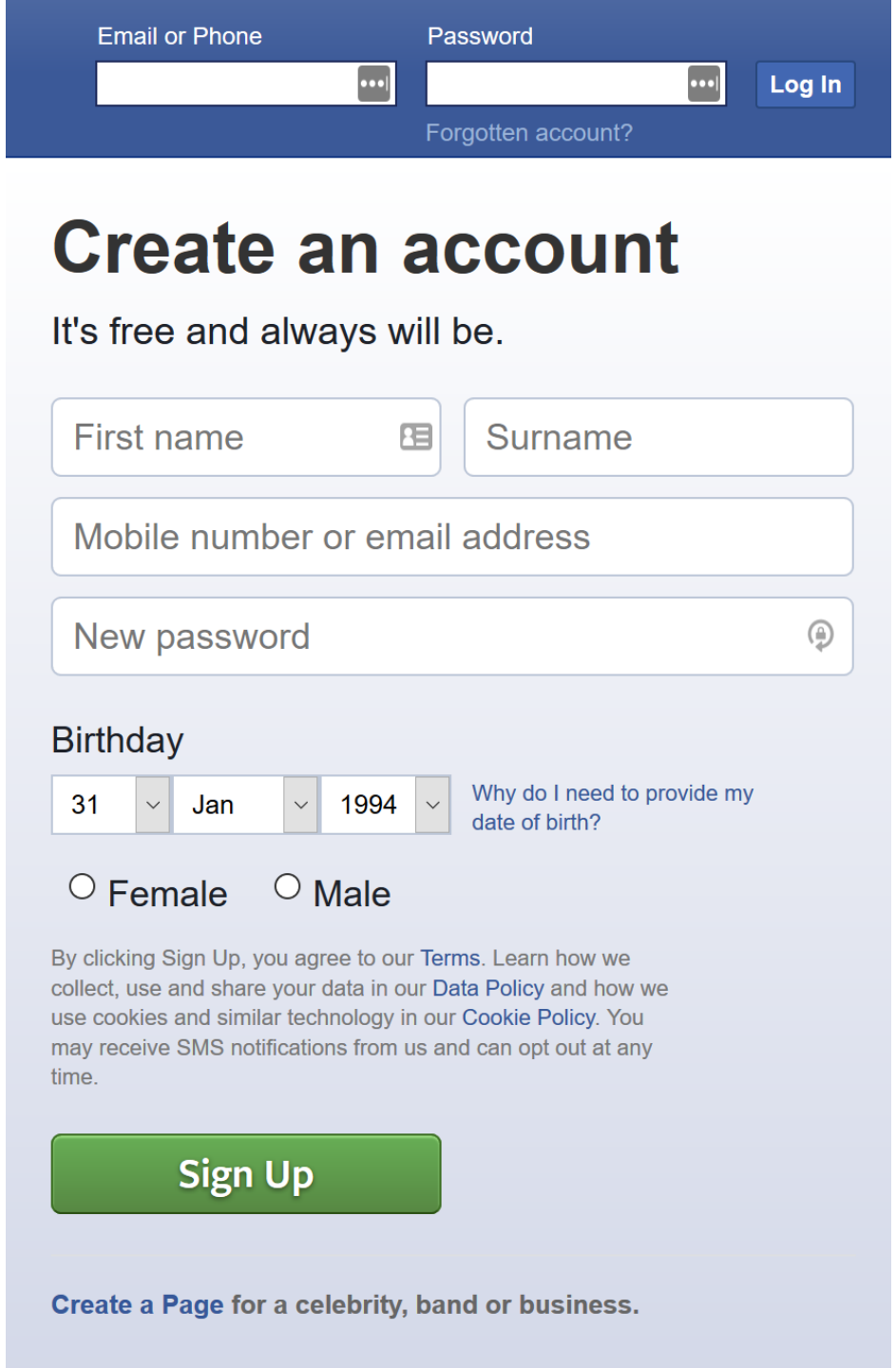
Authentication

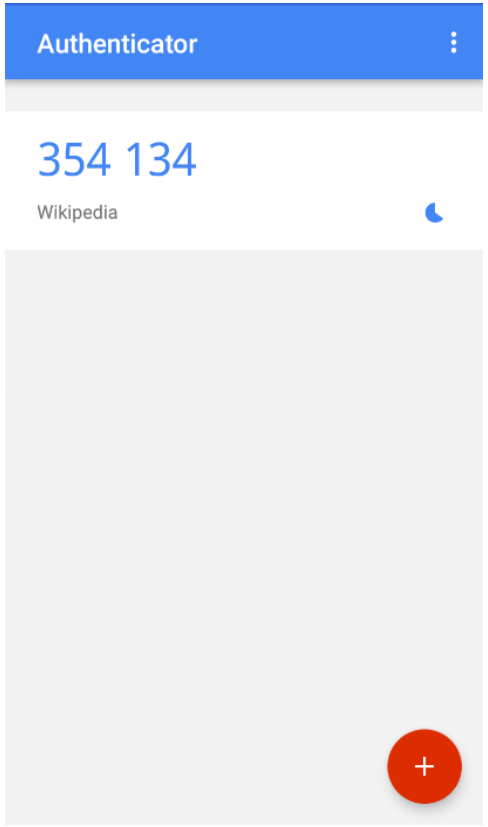
- Verifying a fact about an entity before allowing it/them to perform an action
  - Entity could be a person or a computer or even an animal (think dog doors)
  - Action can include viewing, reading, writing, or interacting in any way (see access control)
- Authentication should happen every time an action is taken and there is no way to be certain that the authenticated entity has not changed.
  - Authentications do not have to be the same
  - Initial authentication can be:
    - person -> computer
    - person -> web server
    - Followed by computer -> web server for future transactions

- Here I have logged into: https://outline.uwaterloo.ca
- The website set a cookie on login (csrftoken)
- My browser sends the token with each page change



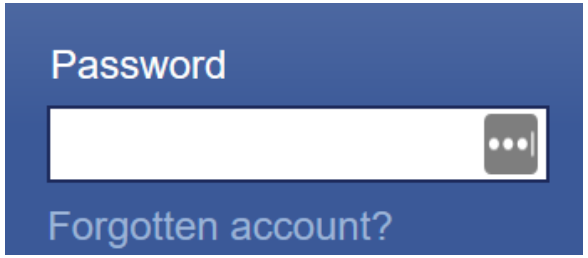
When you think of authentication you probably envision a password login like this one.





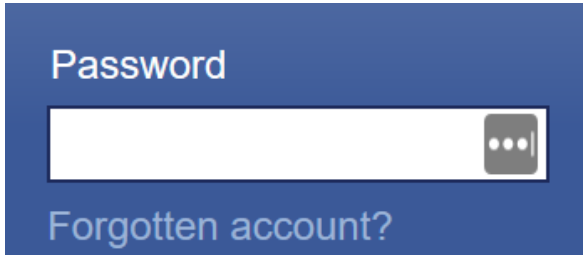
There are many forms of authentication

Authentication factors (for humans)



- Something you **know**
  - Password, mother's maiden name, your address
- Something you **have**
  - Student ID card, credit card chip, RSA key fob, Yubikey
- Something you **are**
  - Fingerprints, voice tones, iris, typing patterns

Also jokingly known as:



- Something you **can forget**
  - Password, mother's maiden name, your address

- Something you **can loose**
  - Student ID card, credit card chip, RSA key fob, Yubikey

- Something you **cannot change**
  - Fingerprints, voice tones, iris, typing patterns

Something you know

Something you know

- Passwords
- Birthdate
- Last ATM visited
- Last purchase made
- Where you lived in 2012
- Drivers license number
- SIN number
- Favorite song
- Make and model of first car

Something you have

Physical keys

- Keys are one of the most common examples of something you have
- Each key contains a “code” in the form of notches on the key
- Having one allows you to open physical locks
- Single factor authentication

RSA key fob

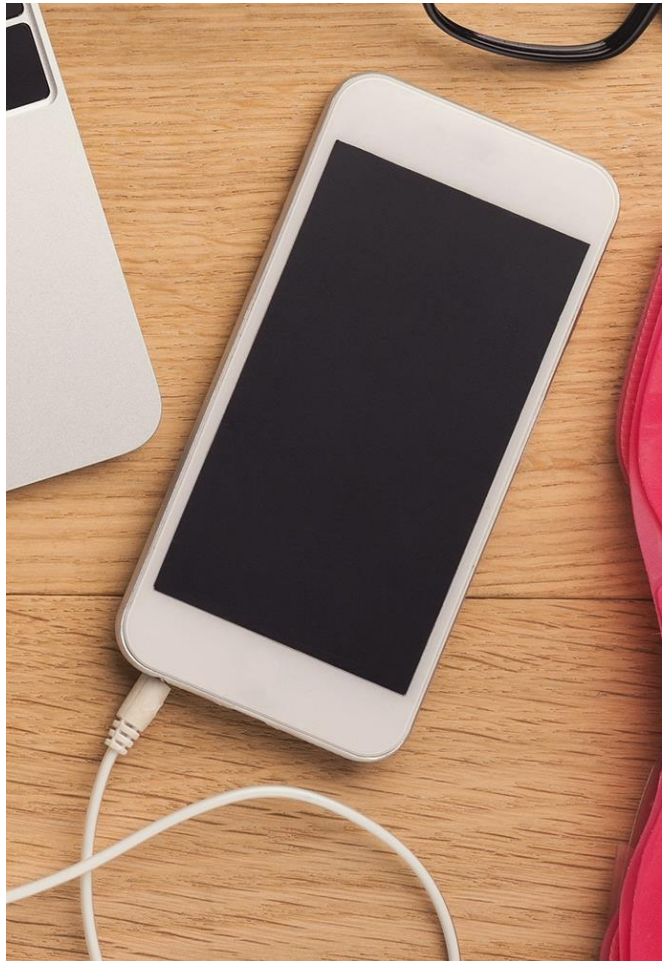
- When a button is pushed the fob prints out a number
- The number is generated securely using methods we will talk about later
- The number must be typed in along with a password
- Two factor authentication

Chip in a credit card

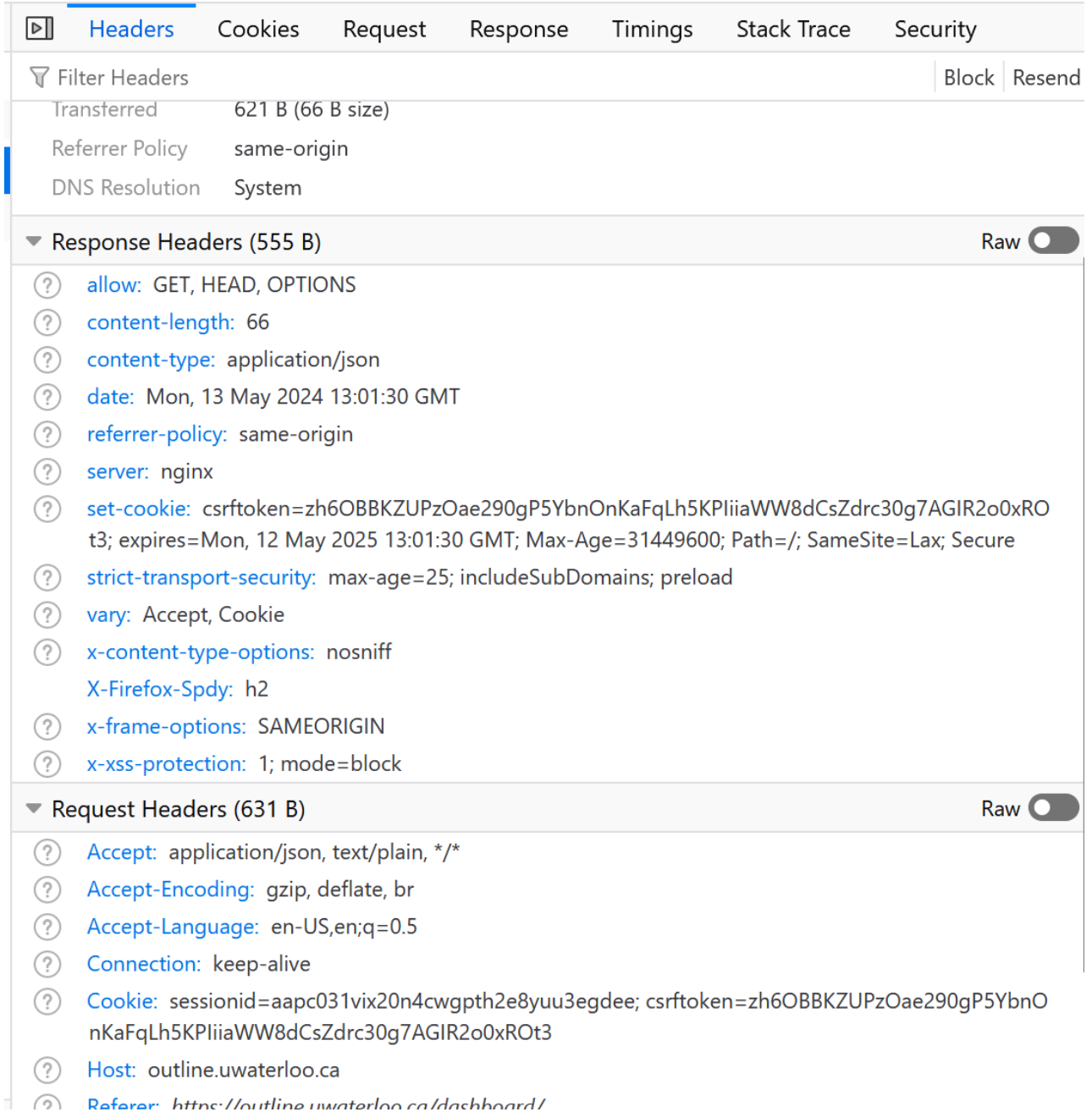
- Similar to RSA fob, the chip generates a unique code
- The user

Access to information sent to your phone number or email

- Having access to something else can be proof of something you have
- Messages sent to your phone number
- Messages sent to your email
- Information in your bank account (how much was deposited)



Your browser or computer may have things on your behalf



A public/private digital key

- We will discuss these in more detail later in the course
- Simply: A public key can unlock what a private key locks, and vice versa
- A PGP key is something you have which authenticates you
- For example, if a file is encrypted using the key on the right only I can decrypt it using the matching private key which only I possess

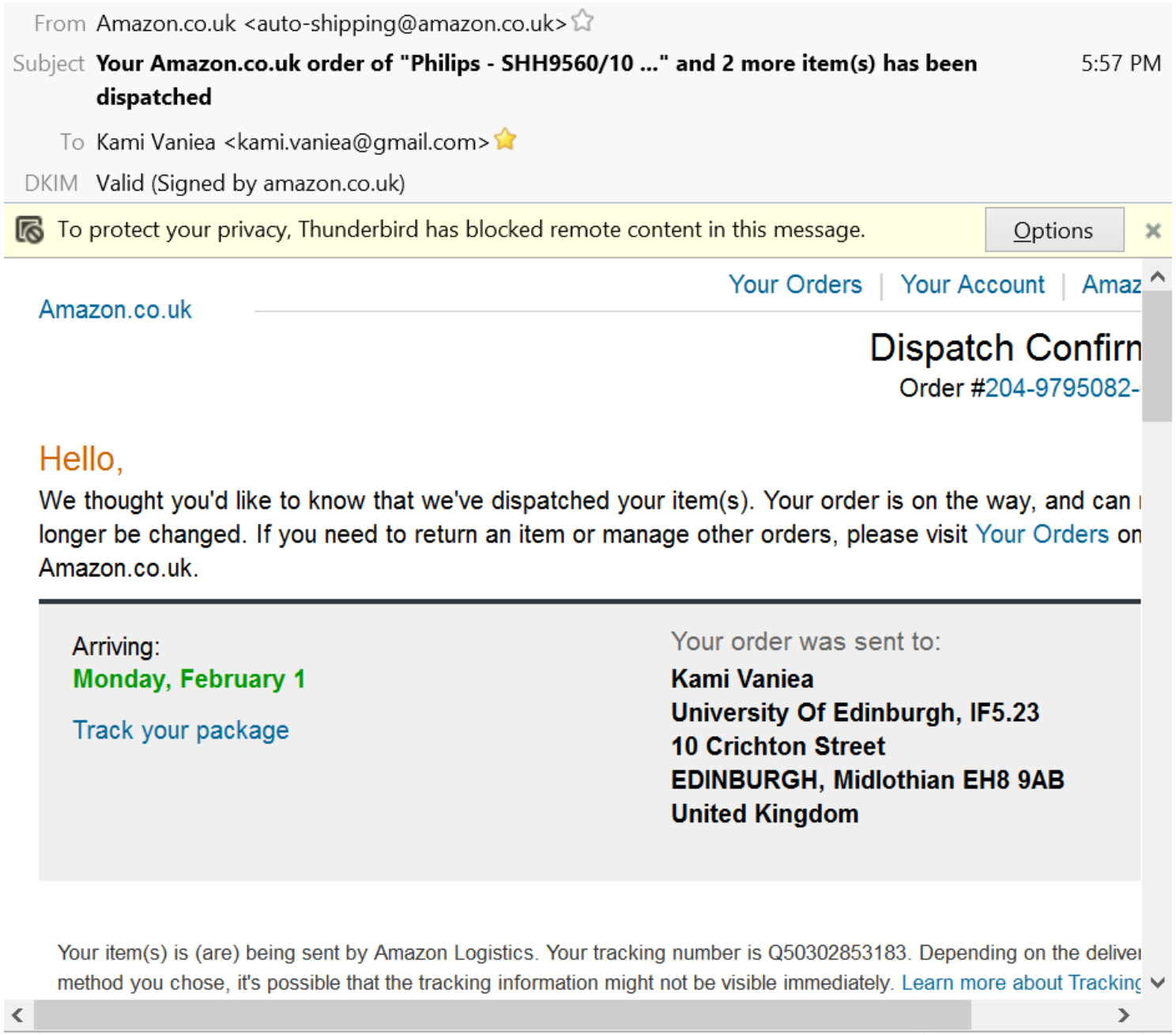
My public key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQENBFHMcgABCAC9WrYDO6K2L3VHyi4eHN6suHLqMpJ+SO+IUTuLEVn
UzIoXAUXH
KozHejfV/9XoG8j933ZtszXKCog3aMESeoE0z6fNGfolvaCe5B4jwq0Jt8NHwb5L
B2dnq0CplgXcN2GJxfEHHUaf27COSobCJxPMeshUh4ZHke+g6DatmiEtBpVp4
1Ot
1zgxdMQkgb2H2xw28RYfYkdDoueteIkOrFLrCy9ZF9KdMhA1eBH94KnwIQshdi
ZR
QYEX25+M8cKCb++Rc9H6an7EG9WHOFRW40UsY52OfveOyfQPzkkRto7u233
9hvHo
B/h+7xLM6FQbOUZQ9BD5w7IQHgYtXJVsUj0dABEBAAG0IkthbWkgVmFuaW
VhIDxr
dmFuaWVhQGluZi5lZC5hYy51az6JAT8EEwEIACkFAlYKYvECGyMFCQlmAYA
HCwkI
BwMCAQYVCAIJCgsEFgIDAQIeAQIXgAAKCRCTdsxl9/HZffG+CACShuKxje3Q
Aqew
GWh8K4gCdiY0xDqJwq3PHxmyhZmQeN/1a1KcOrIjI2b+Q75/5t+EgXOHpR0P
IxfG
lZ6zOEpf6A18iFXx3JgQZdwPD0jtBiWNpOyMeBGTgIvEYG3so2VueQoeXcq3db
Yp
5vstVxtD+TKHQ5CioIT75P2bzYq/XLT5aIbNQhQDPcTo0DgbRH+FvqsRXr7yea
ef
JaPnxX0+1L33t2QY9zctiGyebwrvHMrIPBJ2VYCDzQkJ7uQ5eFh4ZhsMgOmzL
QD4
YiGr5weIMFwAvxZOaRxEa9Vf48jiWvrxuJ8YfHWS0hEScNOcYC2P8q20lJwwE
26T
lpdtrwCqtB1LYW1pIFZhbmllYSA8a2FtaUB2YW5pZWEuY29tPokBQgQTAQIAL
AIb
IwUJCWYBgAcLCQgHAwIBBhUIAgkKCwQWAgMBAh4BAheABQJWCmMeAhk
BAA0JEJN2
zGX38dl9JJAIAIW0rxrlYsrmKS6CbW8MgTxxTDOXaCt1b7F0W0QZHskIUQhE
cE+a
XBYib1A5uHaatLfyjeXaD3qME0ZnQHoYMGE0GKu00wWsbhfoQzHPgwzRLkD
1i75M
BIbaww0KW0VB9e4AkMakXJCnF5BXeo6AHRL2v15V205DikVnlCRXocKtu8b7
LnkM
cLn70Lobr1de1uyKoNzbSnO/vpKDJp0/EY5yUeV90IypZy/6wFQBehg1sXye6zn
O
9wb9uUsu9+/P8pz4JILMDSevjfT7zSRSl/YP3fOfZ6N4bc+KOdwPM7u5Iyoeu9z
h
pzibv3ge7VhH2xIWz8vYZ/2xT1345tWRRMOJAhwEEwECAAYFAlTnSpEACgkQ
jyxM
p99tBt2B8A/+OpIzOsQbQJB8yxti4I7PpD1weJDf3a81Vhm7JyXE/Xy66ypfdt3w
XmFRUuIrwezY1NebWNCRQHzQvRv/VJwjbTUx+Q3HsjIkKlHbE7iCiQXXtTRk
0Eny
2nudcjGI2v03C3B2JCucEw6esF1x79PI/lPv2+6tgUBKmDfOpsB2vbtqrHnmAYK

DKIM

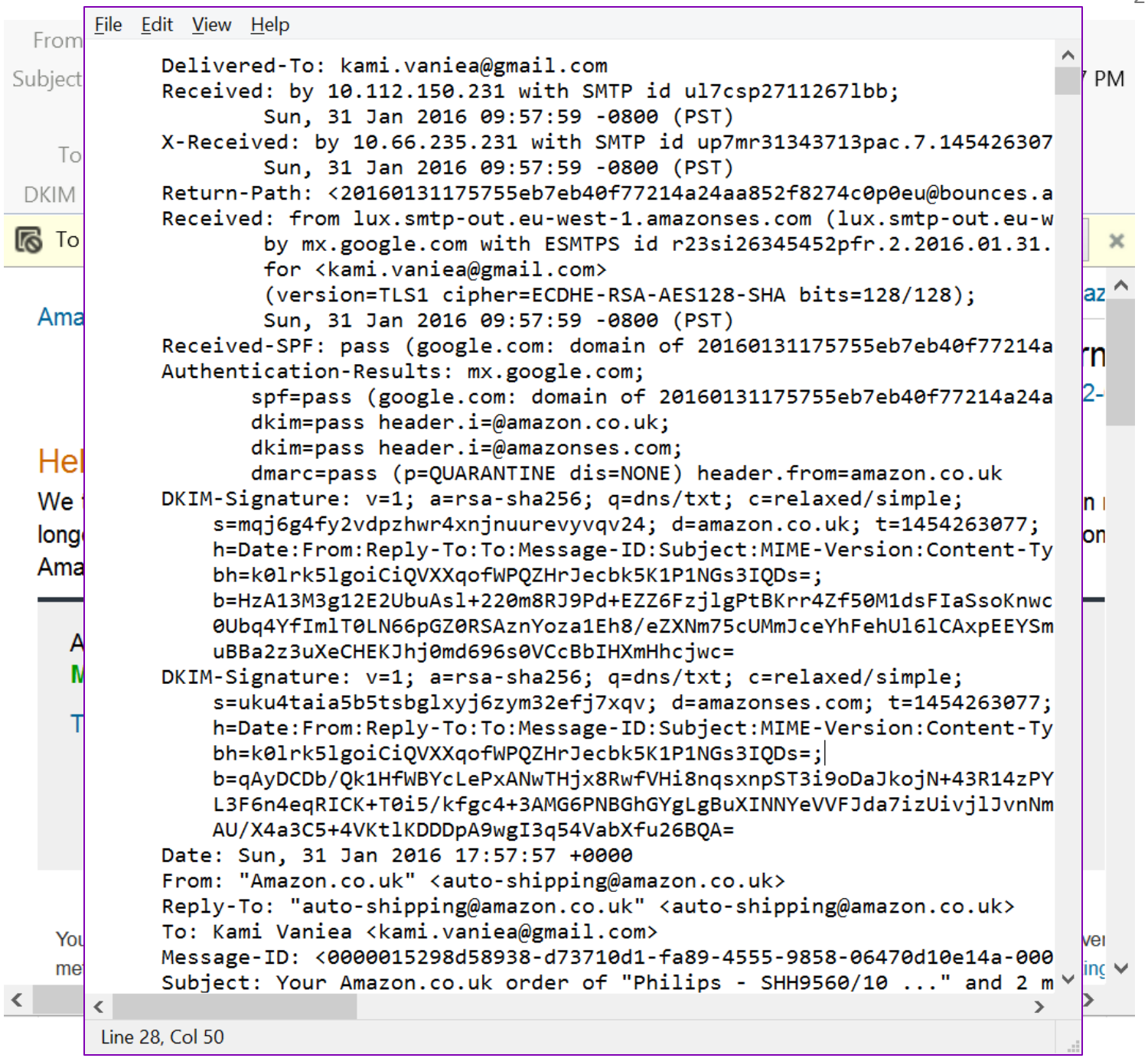
- People are not the only ones who authenticate
- Servers also need to authenticate to each other
- One of the most visible is DKIM signatures in email



DKIM

- Problem: Spam
- Solution:

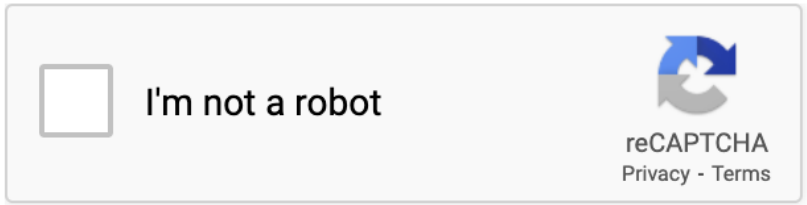
1. Sending email server signs the email using a private key
2. Receiving email server checks the key to authenticate the sending server

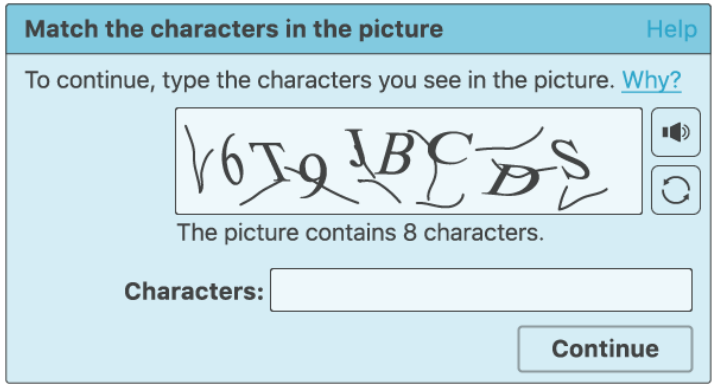


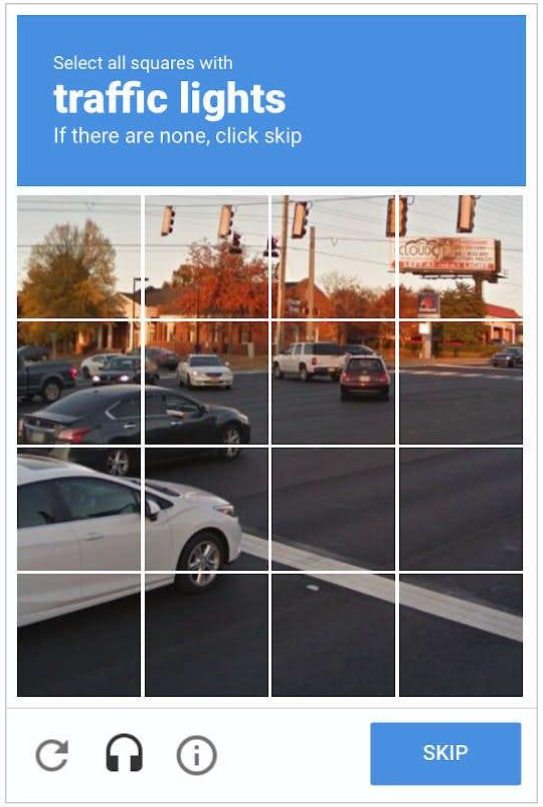
Something you are

Something you are

- A property about the person (or device)
- Fingerprints
- Iris scan
- Voice recognition
- Facial recognition
- The way you move your mouse







Fingerprint readers

- Fingerprints are nearly unique so they seem like a good authenticator
- Not all people have fingerprints
  - Some professions destroy fingerprints
  - Some fingerprints are too faint to read
  - Dehydration (e.g. airplane flight) reduces fingerprint ridges
- Fingerprints can never be changed
- You leave fingerprints everywhere

Fingerprint readers

- Fingerprints are nearly unique so they seem like a good authenticator
- Not all people have fingerprints
  - Some professions destroy fingerprints
  - Some fingerprints are too faint to read
  - Dehydration (e.g. airplane flight) reduces fingerprint ridges
- Fingerprints can never be changed
- You leave fingerprints everywhere

Most biometric readers have similar problems

Some aspects of “something you are” can be stolen or compelled

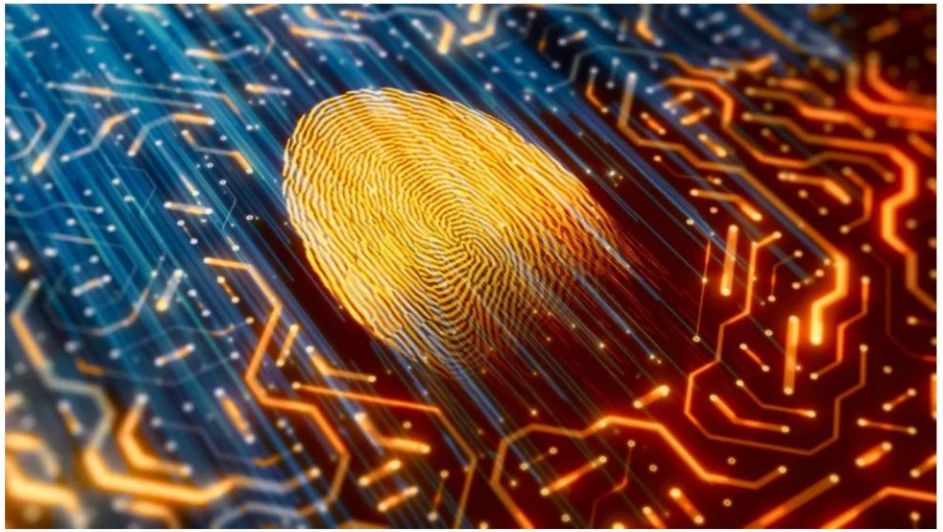
POLICE 1

Investigations

CHP using detainee's fingerprint to unlock phone not a Fifth Amendment violation, appeals court rules

The court stated that using the suspect's thumbprint required no mental exertion on his part and fell into the same category as a blood test taken at booking

April 19, 2024 01:04 PM



Payne argued that using biometrics is akin to providing a physical key to a safe but still a testimonial act because it confirms ownership and authentication of the phone's contents, according to the report. The U.S. Court of Appeals for the 9th Circuit dismissed this, stating, "Payne was never compelled to acknowledge the existence of any incriminating

ars TECHNICA SECTIONS FORUM SIGN IN

YIKES

We have reached the “severed fingers and abductions” stage of the crypto revolution

Wave of crypto kidnappings hits Europe.

NATE ANDERSON – MAY 7, 2025 4:01 PM 131

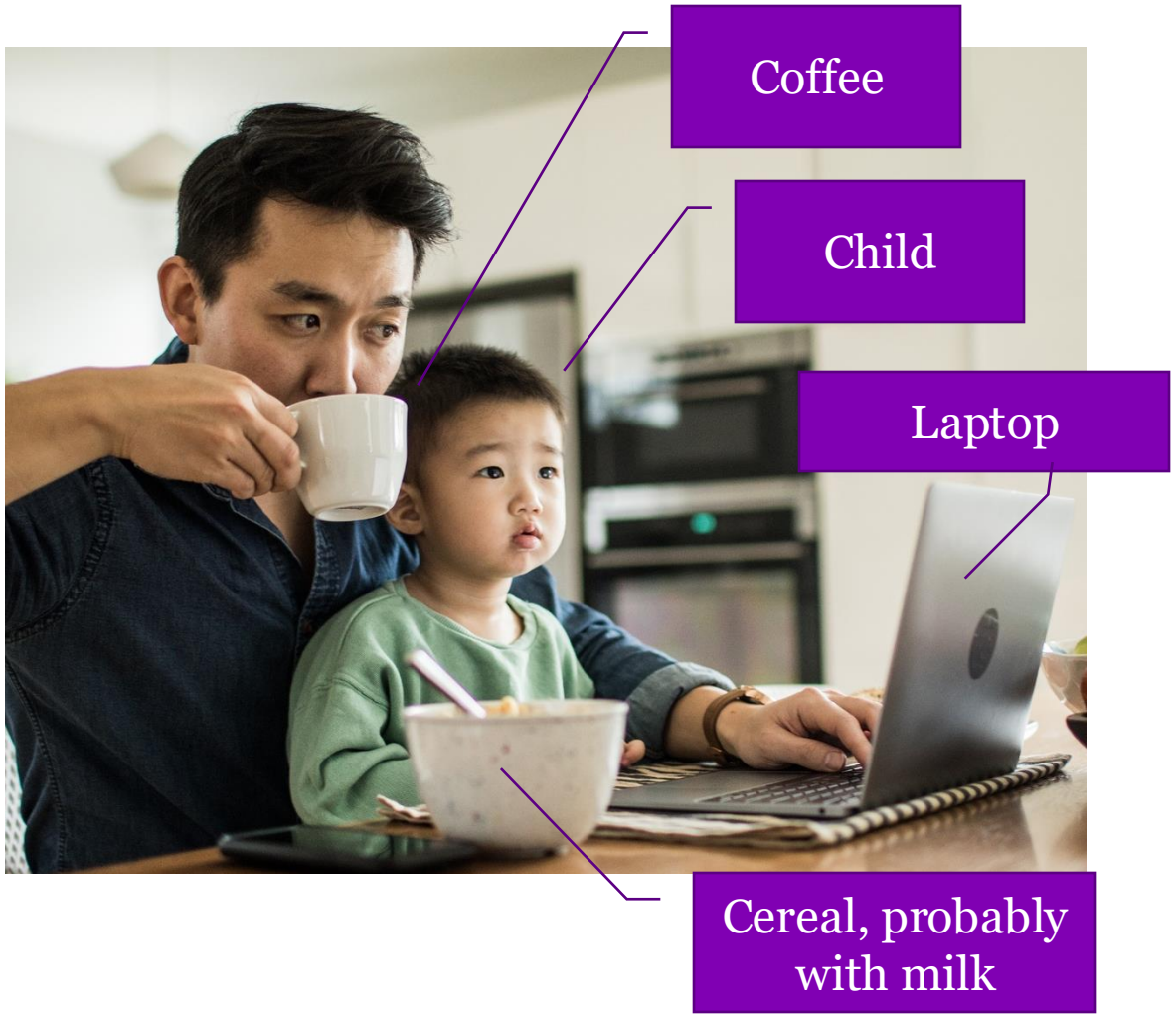
Credit: Getty Images

TEXT SETTINGS

French *gendarmes* have been busy policing crypto crimes, but these aren't the usual financial schemes, cons, and HODL! shenanigans one usually reads about. No, these crimes involve abductions, (multiple) severed fingers, and (multiple) people rescued from

Continuous authentication

- Your interaction with a computer is unique and we can measure it
  - Mouse movements
  - Keyboard typing patterns
- Nearly impossible to duplicate a real user's typing patterns
- Easy to lose access if the user hurts their hand, or is doing something non-standard
- Repetitive Stress Injury (RSI) patients trigger continuous authentication warnings regularly while healing



Privacy

- Users have a right to privacy, that is, a right to keep aspects of themselves hidden that are not necessary to expose
- Authentication mechanisms need to take privacy into account and not ask for more than they need
- Identifying a user using a Facebook, Google, or Apple account may be easy, but it gives away large amounts of data
- Similarly, requiring a validated ID such as drivers or passport information also exposes quite a bit of information

Multi-factor authentication

- Combine two of the earlier factors. For example:
  - **Having** a credit card and **knowing** the pin

  - **Knowing** a passcode and **being** the person with the correct fingerprints

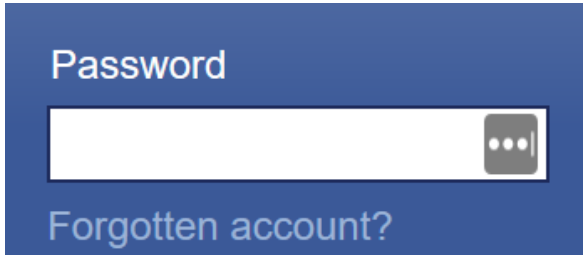
Multi factor authentication

- Authentication that requires two or more of the factors.
- Two-factor
  - Chip and pin in a credit card. Something you have (chip) something you know (pin).
  - Chip and signature credit card. Something you have (chip) something you are (signature pattern).
- Three-factor
  - Security guard that check's your ID against what you look like and then requires a code.
  - Secure finger print reading fob that gives you a code after it reads your fingerprint, then you use the code and a password to log in.

Authentication is not about verifying your identity.

Authentication verifies that you possess a property.

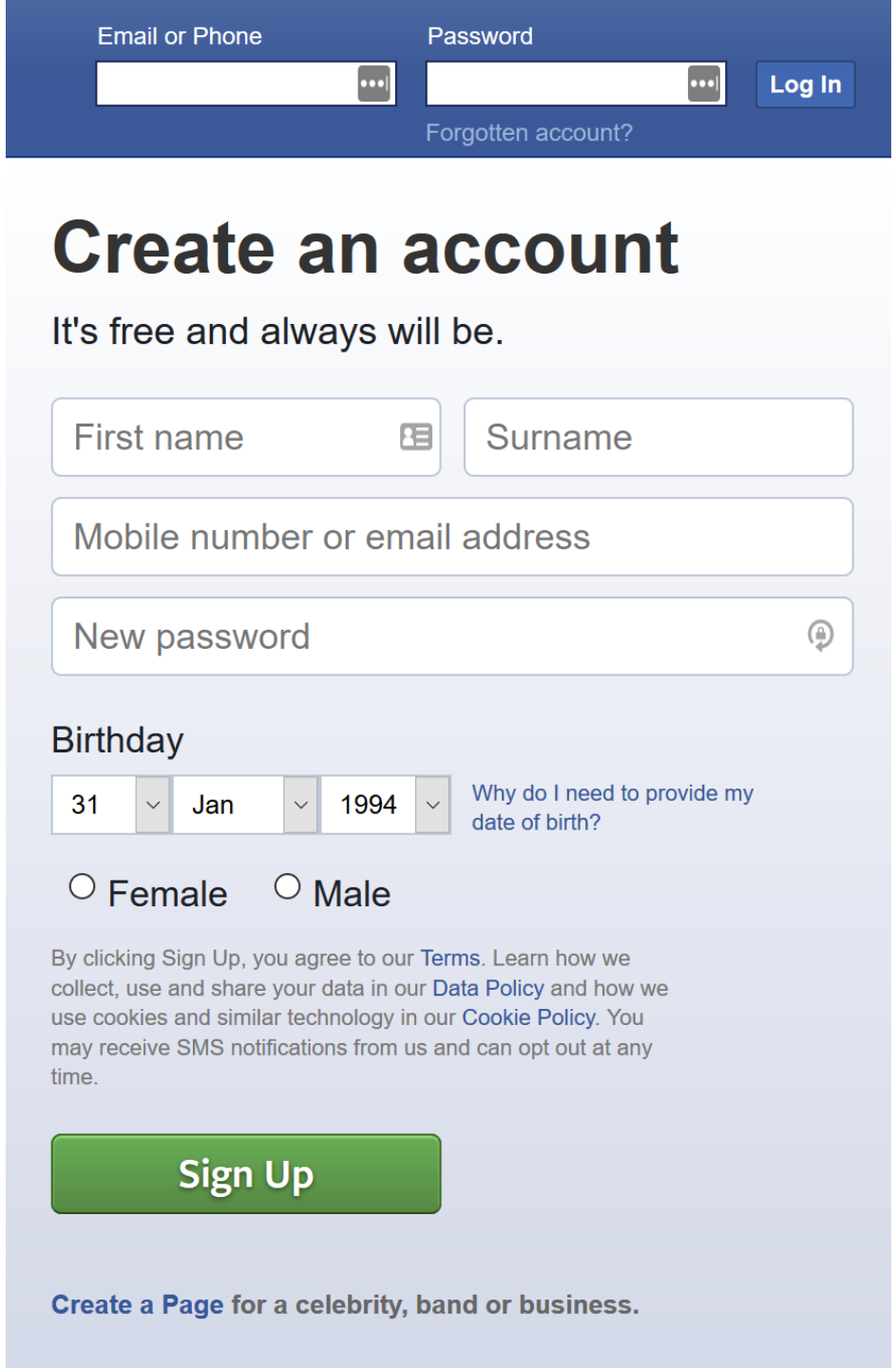
Most of these verify that the entity is the same as last time



- Something you **know**
  - Password, mother's maiden name, your address
- Something you **have**
  - Student ID card, credit card chip, RSA key fob, Yubikey
- Something you **are**
  - Fingerprints, voice tones, iris, typing patterns

The “create an account” is collecting information on the first interaction.

Then using the password to verify that you are the same person next time you log in.



Not all authentication proves identity

Money is available

Card number & Pin

Invisible continuing authentication

1. You log into a website using a password (something you know).
2. Website sets a cookie with a secret and a timestamp.
3. Every time you visit a new page your computer sends the cookie and the server verifies it.
4. When you log out the cookie is destroyed. Or when you don't use it for too long, the cookie is destroyed.

How bank websites do (mostly) invisible 2-factor authentication

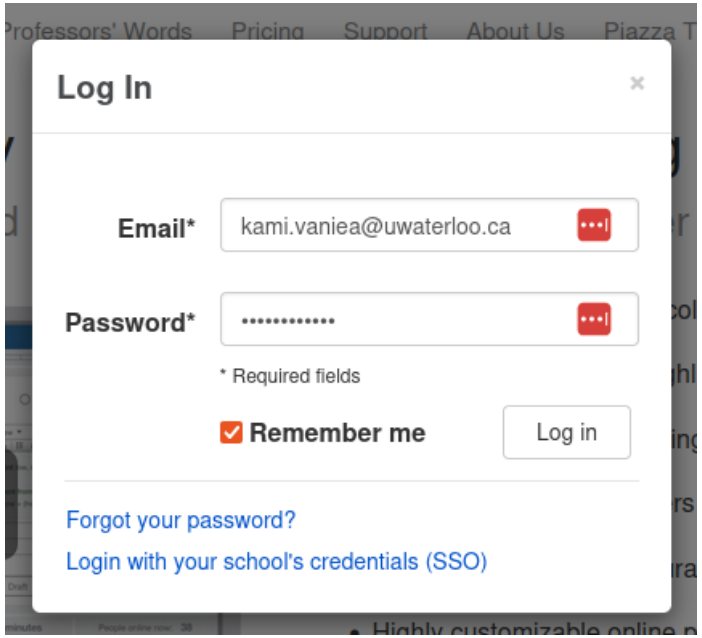
1. You log into a website using a password (something you know).
2. The website is also sent the cookie from the last time you logged in (something you have).
3. If the password and the cookie both match you get to log in.
4. If the cookie is missing, or wrong, the bank will ask you to prove that you have something else by calling you (phone) or emailing you a code (email).

How credit cards do (mostly) invisible 2-factor authentication

1. Build a fingerprint of the person's activities
    - Where are they normally physically located
    - What stores do they normally shop at
    - Amount they normally spend
2. If a transaction is "normal" allow lower-security card tapping
3. If a transaction does not match the fingerprint: require the pin be used
4. If transaction very odd – like purchasing things in Vancover BC and Waterloo ON within minutes – call the customer and possibly lock the card

Think-pair-share

- Why is multi-factor authentication so important?
  - What is the "multi" part protecting users against?
- Why is requiring two of the same factor not enough?



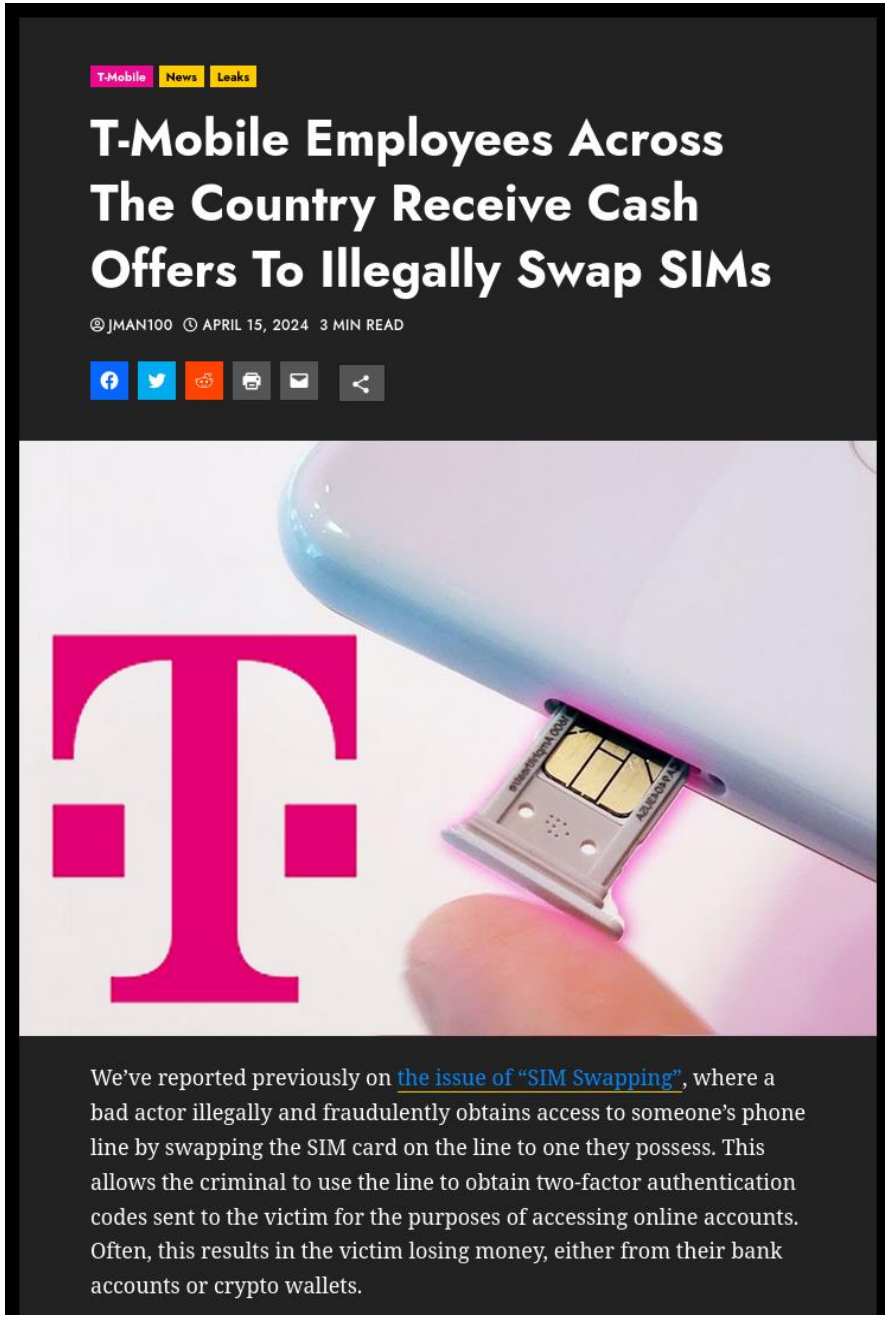
Think about what you are authenticating

- Actual identity of the person
- That they are the same entity who setup the account
- They have a specific property
  - Above the legal drinking age
  - Student at a university
  - Facebook user
- That another authenticator thinks they are the same entity (federated identity)
  - Authenticate using Facebook

SMS AUTHENTICATION

Think about who/what verifies the second factor

- Phone number destinations can be altered by a large number of people



T-Mobile News Leaks

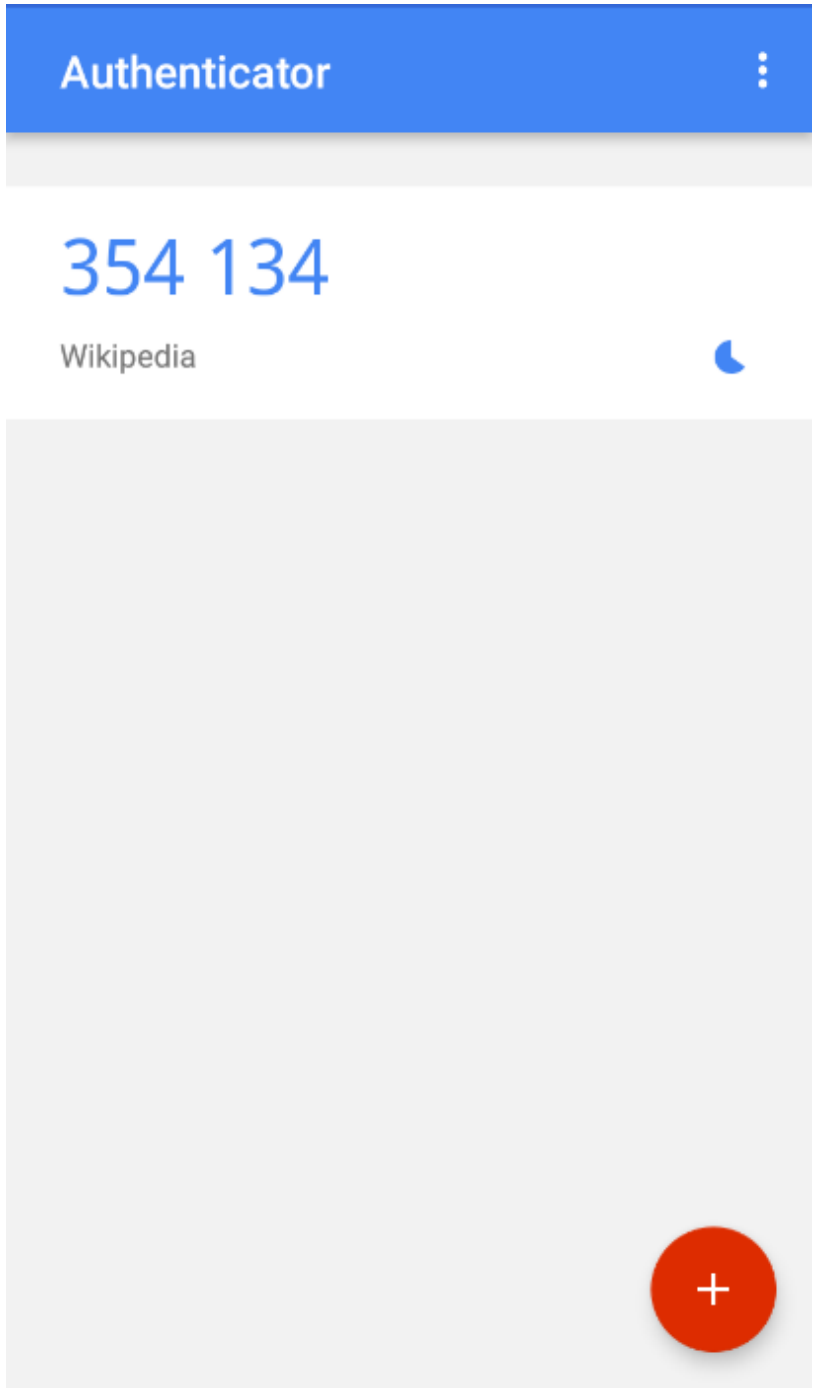
T-Mobile Employees Across The Country Receive Cash Offers To Illegally Swap SIMs

JMAN100 APRIL 15, 2024 3 MIN READ

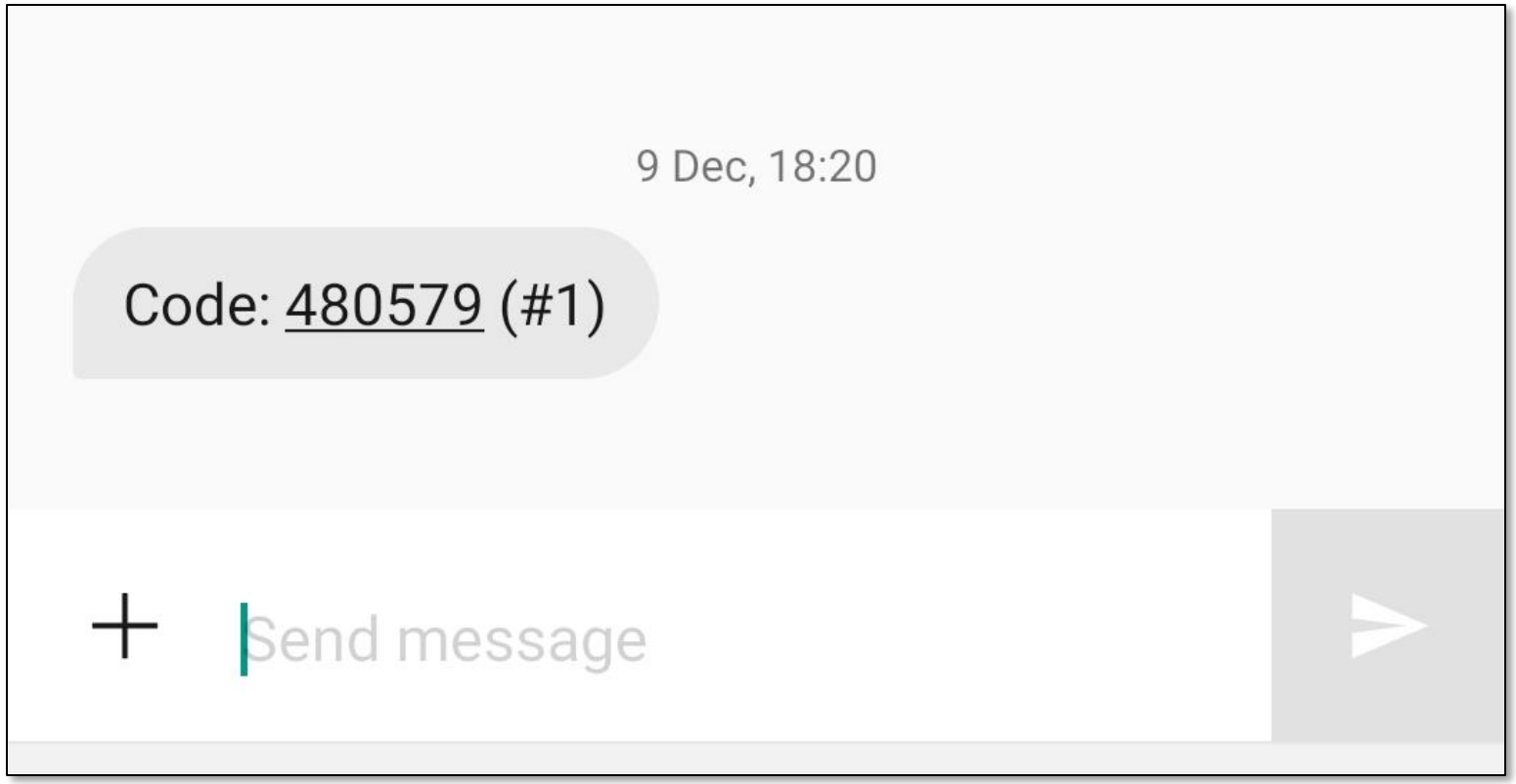
We've reported previously on the issue of "SIM Swapping", where a bad actor illegally and fraudulently obtains access to someone's phone line by swapping the SIM card on the line to one they possess. This allows the criminal to use the line to obtain two-factor authentication codes sent to the victim for the purposes of accessing online accounts. Often, this results in the victim losing money, either from their bank accounts or crypto wallets.

Variation on “something you have” which is your phone.

Google Authenticator uses the app as a secondary secure channel to send a code



Weaker version of “something you have” which is access to messages sent to your cell phone provider.



But is your cell phone provider doing authentication?

We Were Warned About Flaws in the Mobile Data Backbone for Years. Now 2FA Is Screwed.

Financially-motivated hackers are using SS7 attacks to break into bank accounts.

SHARE TWEET

It has finally happened.

For years, researchers, hackers, and even some politicians have warned about stark vulnerabilities in a mobile data network called SS7. These flaws allow attackers to listen to calls, intercept text messages, and pinpoint a device's location armed with just the target's phone number. Taking advantage of these issues has typically been reserved for governments or surveillance contractors.

But on Wednesday, German newspaper *The Süddeutsche Zeitung* reported that financially-motivated hackers had used those flaws to help drain bank accounts.

Malicious actors target the weakest link in the authentication chain.

The Register
Biting the hand that feeds IT

Security

Bitcoin backer sues AT&T for $240m over stolen cryptocurrency

Michael Terpin not happy about funds-draining SIM swap fraud

By Kieren McCarthy in San Francisco 15 Aug 2018 at 19:12 42 SHARE ▼

A bitcoin investor is suing AT&T for $240m after it allegedly ported his phone number to a hacker, allowing the criminal to steal $24m in cryptocurrency.

Assumptions are also made between one software and another.

Such as: the address book content is accurate

Exclusive: how the Atlantic's Jeffrey Goldberg got added to the White House Signal group chat

Internal investigation cleared the national security adviser Mike Waltz, but the mistake was months in the making

Mike Waltz (left) and Jeffrey Goldberg. Composite: AP/Reuters

Hugo Lowell *in Washington*

Sun 6 Apr 2025 14.54 BST

Share

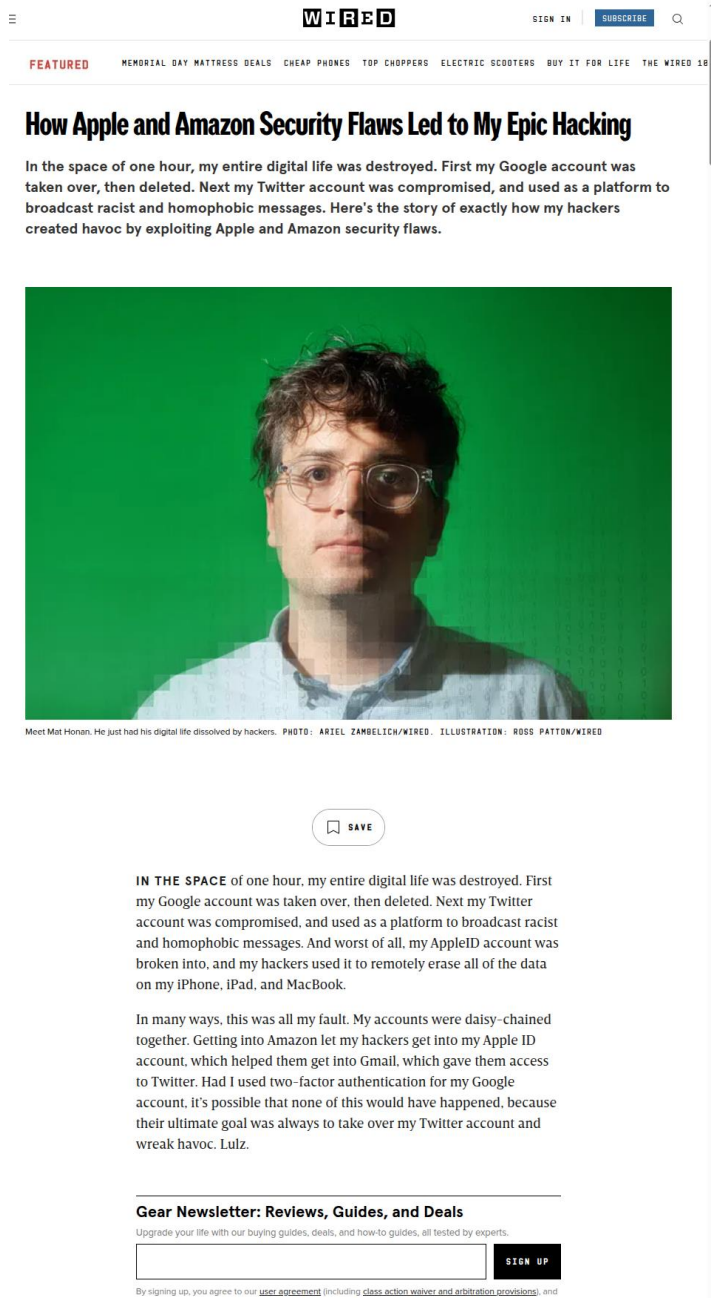
Donald Trump's national security adviser Mike Waltz included a journalist in the Signal group chat about plans for US strikes in Yemen after he mistakenly saved his number months before under the contact of someone else he intended to add, according to three people briefed on the matter.

The mistake was one of several missteps that came to light in the White House's internal investigation, which showed a series of compounding slips that started during the 2024 campaign and went unnoticed until Waltz created the group chat last month.

Assumptions are also made between companies.

Such as: what data is secret and what is public.

Apple tech support gave the hackers access to my iCloud account. Amazon tech support gave them the ability to see a piece of information – a partial credit card number – that Apple used to release information. In short, the **very four digits that Amazon considers unimportant** enough to display in the clear on the web are precisely the same ones that **Apple considers secure enough to perform identity verification**.



WIRED SIGN IN SUBSCRIBE

FEATURED MEMORIAL DAY MATTRESS DEALS CHEAP PHONES TOP CHOPPERS ELECTRIC SCOOTERS BUY IT FOR LIFE THE WIRED 18

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

SAVE

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. And worst of all, my AppleID account was broken into, and my hackers used it to remotely erase all of the data on my iPhone, iPad, and MacBook.

In many ways, this was all my fault. My accounts were daisy-chained together. Getting into Amazon let my hackers get into my Apple ID account, which helped them get into Gmail, which gave them access to Twitter. Had I used two-factor authentication for my Google account, it's possible that none of this would have happened, because their ultimate goal was always to take over my Twitter account and wreak havoc. Lulz.

Gear Newsletter: Reviews, Guides, and Deals

Upgrade your life with our buying guides, deals, and how-to guides, all tested by experts.

SIGN UP

By signing up, you agree to our user agreement (including class action waiver and arbitration provisions), and

PASSWORDS

Threats – fiction vs reality

Envisioned by security specialists, Hollywood, journalists and fiction writers

Real serious everyday threats normal people face

Entropy

- Roughly password entropy is a calculation of how big the space of possible passwords is.
- In theory, if the space is bigger it will be harder for an attacker to guess a password.
- But that is only true if passwords are evenly divided over the space….

Generate Passwords.org

Menu

Entropy Formula

L = *Password Length; Number of symbols in the password*

S = *Size of the pool of unique possible symbols (character set).*

For example:

- *Numbers (0-9): 10*
- *Lower Case Latin Alphabet (a-z): 26*
- *Lower Case & Upper Case Latin Alphabet (a-z, A-Z): 52*
- *ASCII Printable Character Set (a-z, A-Z, symbols, space): 95*

Number of Possible Combinations = S^L

Entropy = $\log_2$(Number of Possible Combinations)

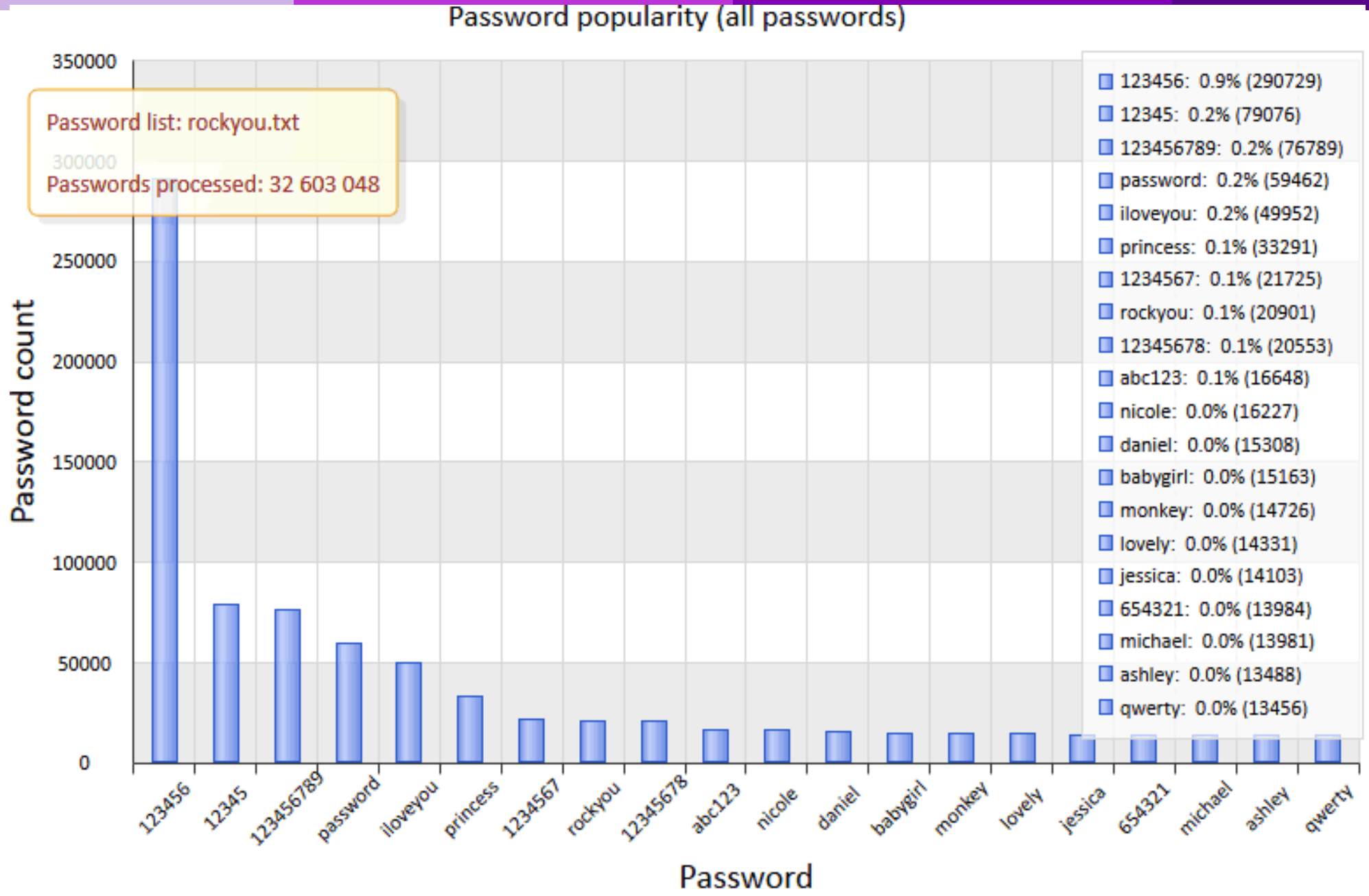
It is important to note that statistically, a brute force attack will not require guessing **ALL** of the possible combinations to eventually hit the right permutation. We therefore tend to look at the *expected number of guesses required* which can be rephrased as *how many guesses it takes to have a 50% chance of guessing the password.*

This can be expressed by extending the formula above:

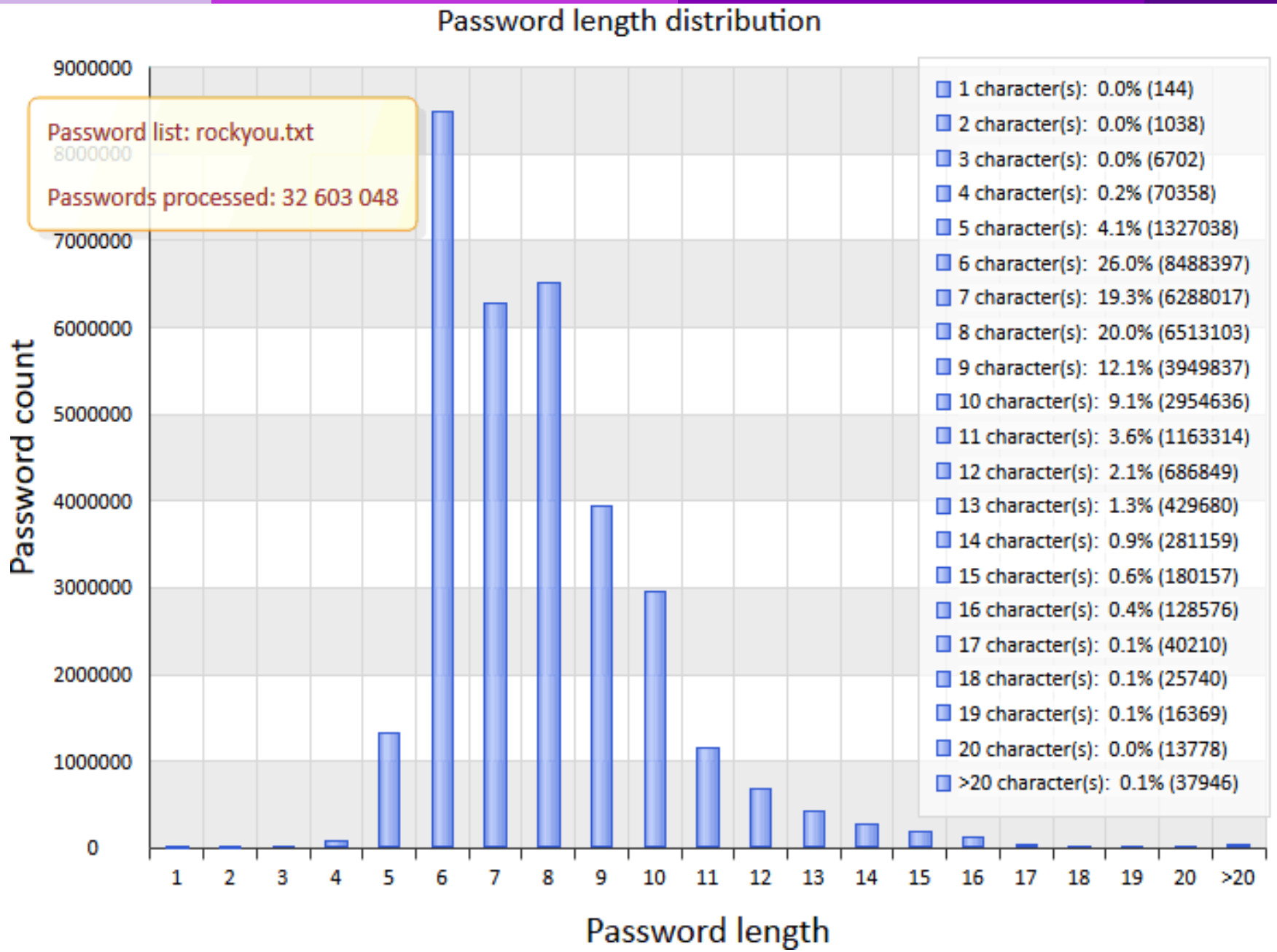
Expected Number of guesses (to have a 50% chance of guessing the password) = $2^{Entropy-1}$

Password space

- 1 character passwords made of only ASCII letters:
  - 26^1 possible passwords
- 8 characters passwords made of only ASCII letters:
  - 26^8 possible passwords
- 8 character password made of ASCII letters + 10 digits:
  - 36^8 possible passwords
- 16 character password made of ASCII letters + 10 digits + 10 symbols:
  - 46^{16} possible passwords

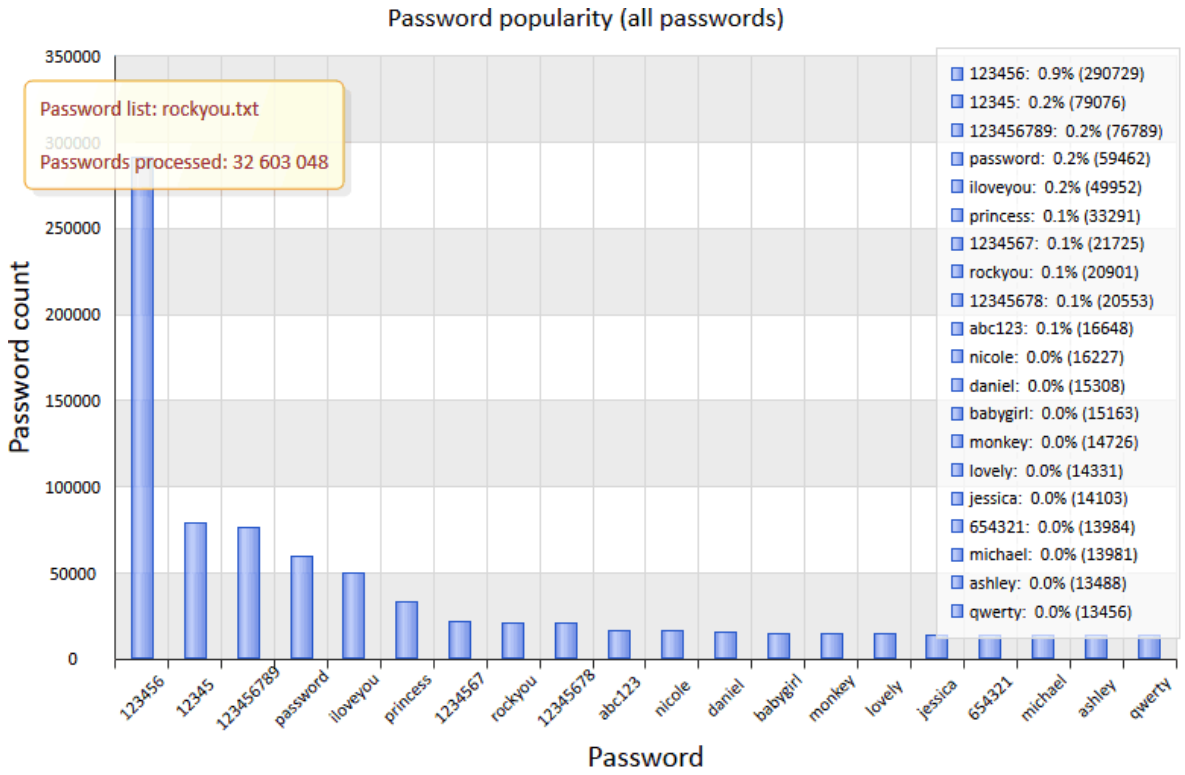


https://www.passcape.com/index.php?section=blog&cmd=details&id=17



https://www.passcape.com/index.php?section=blog&cmd=details&id=17

Entropy vs frequencies



Generate Passwords.org

Menu

Entropy Formula

L = *Password Length; Number of symbols in the password*

S = *Size of the pool of unique possible symbols (character set).*

For example:

- *Numbers (0-9): 10*
- *Lower Case Latin Alphabet (a-z): 26*
- *Lower Case & Upper Case Latin Alphabet (a-z, A-Z): 52*
- *ASCII Printable Character Set (a-z, A-Z, symbols, space): 95*

Number of Possible Combinations = S^L

Entropy = $\log_2$(Number of Possible Combinations)

It is important to note that statistically, a brute force attack will not require guessing **ALL** of the possible combinations to eventually hit the right permutation. We therefore tend to look at the *expected number of guesses required* which can be rephrased as *how many guesses it takes to have a 50% chance of guessing the password.*

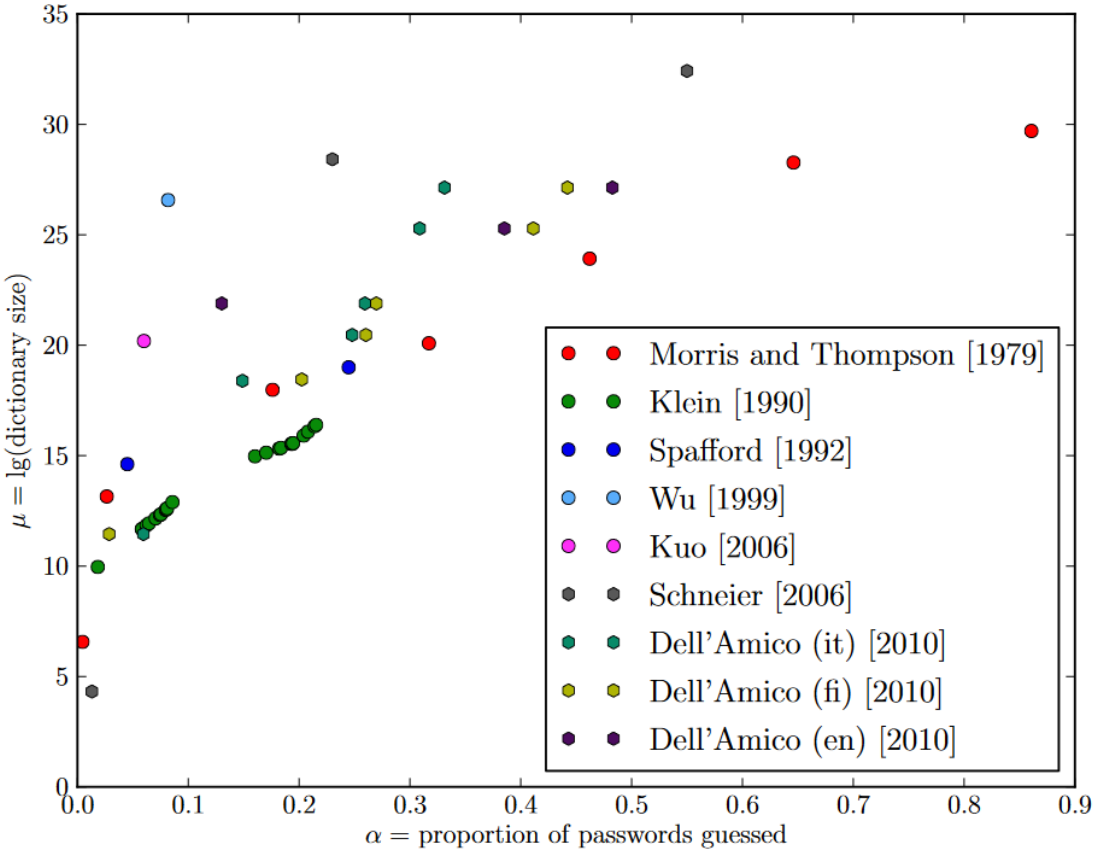
This can be expressed by extending the formula above:

Expected Number of guesses (to have a 50% chance of guessing the password) = $2^{Entropy-1}$

https://www.passcape.com/index.php?section=blog&cmd=details&id=17

Dictionaries

- Lists of common passwords
- Lists of commonly used words
- Mangeling strategies: common adjustments to dictionary words
  - Password -> P@$$word
  - o (char) -> 0 (num)
  - s -> $
- Theoretically dependent on user characteristics like language



(a) Historical cracking efficiency, raw dictionary size

Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In Proceedings of IEEE SP 2012.

Entropy vs other measures

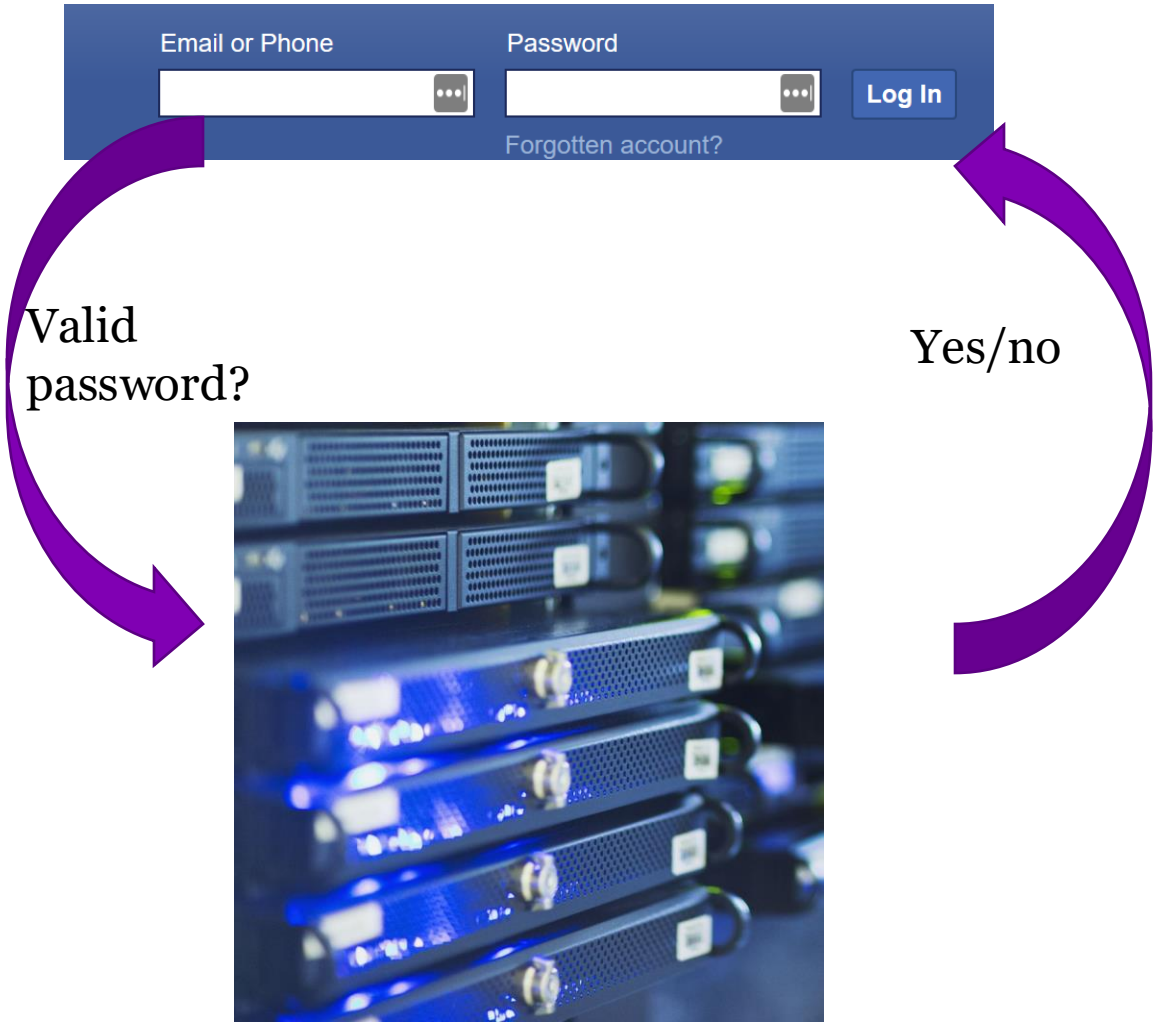
- Shannon entropy – originally intended to measure signal/noise
- Hartley entropy – how big is the distribution
- Min-entropy – what is the probability of guessing the most common password
- Guesswork: expected number of guesses to find the password
  - Sequential guessing?
  - Probabilistic guessing?
    - Where do the probabilities come from?

Joseph Bonneau. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In Proceedings of IEEE SP 2012.

Password Storage

To authenticate, servers need to be able to compare given and expected passwords

- Plain text – store the password as it is and hope your other security measures protect it VERY bad idea
- Salted hash – compute the salted hash of the password and store the hash Best Practice
- Lockout – stop guessing attacks by "locking" an account after N failed password guesses Best Practice



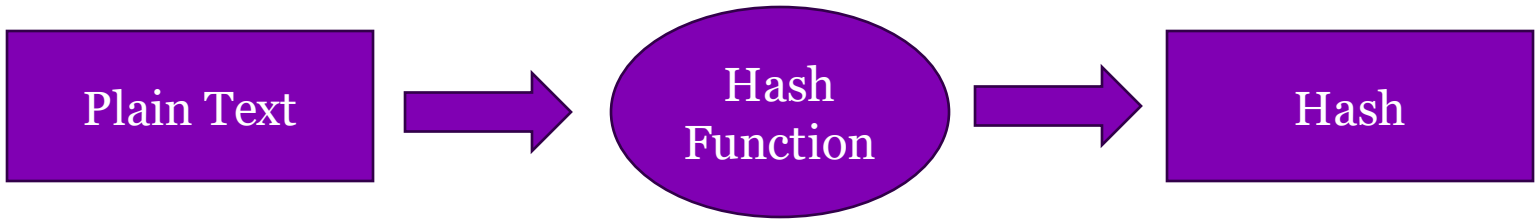
Hashing

A row from /etc/shadow

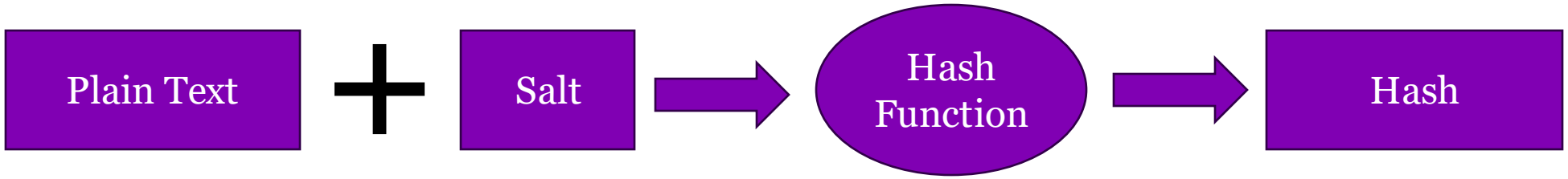
aychedee:6vb1tLY1qiY$M.1ZCqKtJBxBtZm1gRi8Bbkn39KU0YJW1cuMFzTRANcNKFKR4RmAQVk4rqQQCkaJT6wXqjUkFcA/qNxLyqW.U/:15405:0:99999:7:::

- There are two ways to protect a password on a server:
  - You can encrypt the password and keep the key in a *really really* safe place
  - You can hash the password. Hashing does not require a secret key so there is no secret key to lose

- A hash is a one way function. So the same password will always produce the same hash. But the hash cannot be used to produce the password.



Salted Hash



Hashing

A row from /etc/shadow

aychedee:6vb1tLY1qiY$M.1ZCqKtJBxBtZm1gRi8Bbkn39KU0YJW1cuMFzTRANcNKFKR4RmAQVk4rqQQCkaJT6wXqjUkFcA/qNxLyqW.U/:15405:0:99999:7:::

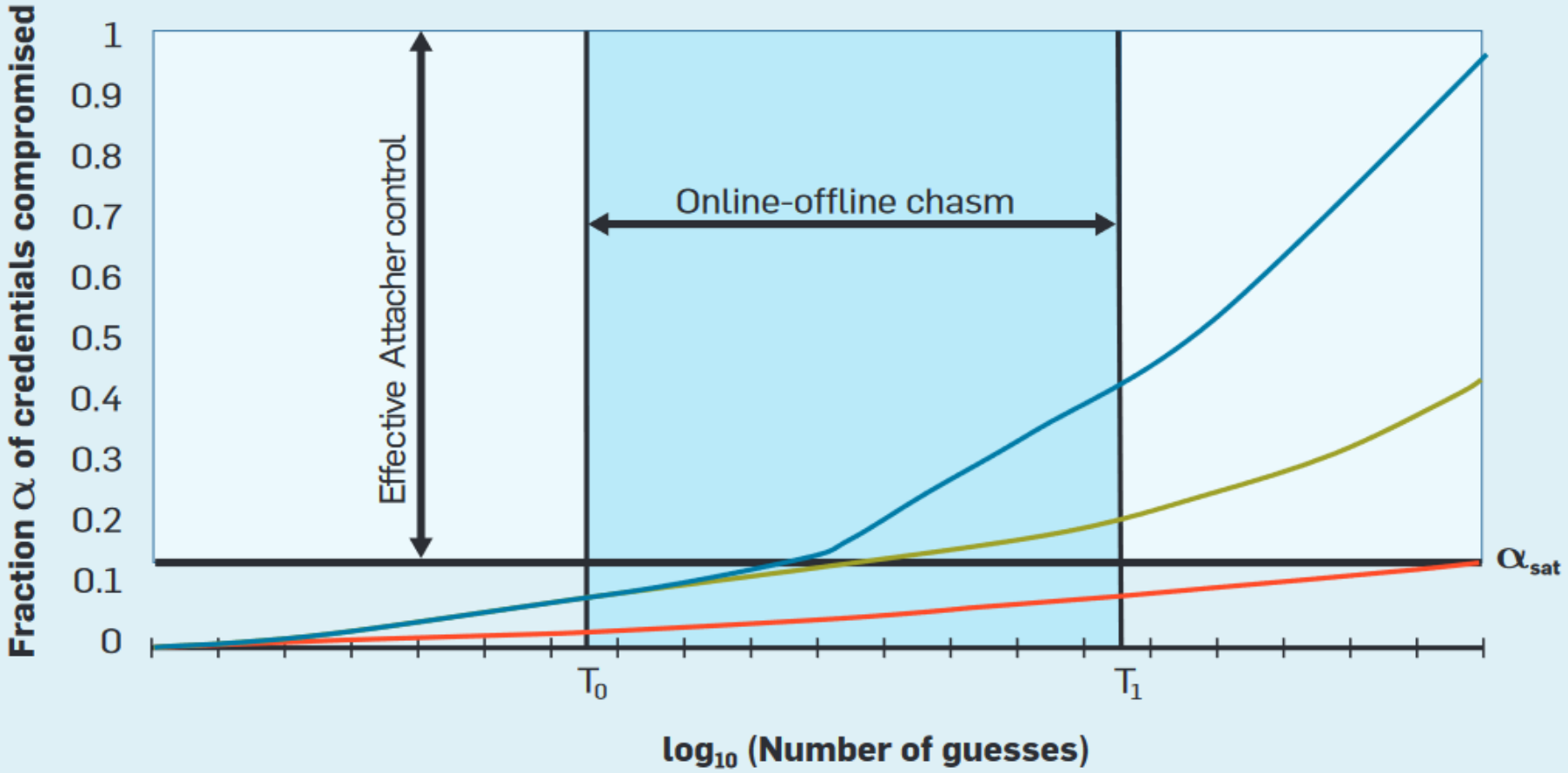
- What type of hash function was used
  - 6
- Salt
  - vb1tLY1qiY
- Hashed password
  - M.1ZCqKtJBxBtZm1gRi8Bbkn39KU0YJW1cuMFzTRANcNKFKR4RmAQVk4rqQQCkaJT6wXqjUkFcA/qNxLyqW.U/

Problem: what if an attacker gets the hashed password list?

- Worst case: passwords stored in plain text or hashed with no salt so the attacker can get them in a matter of days
- Best case: proper hashing with salt, attacker must try many possible combinations of passwords
  - Naïve assumption is attacker will brute force trying all combinations
  - Reality is attacker will start with lists of known common passwords

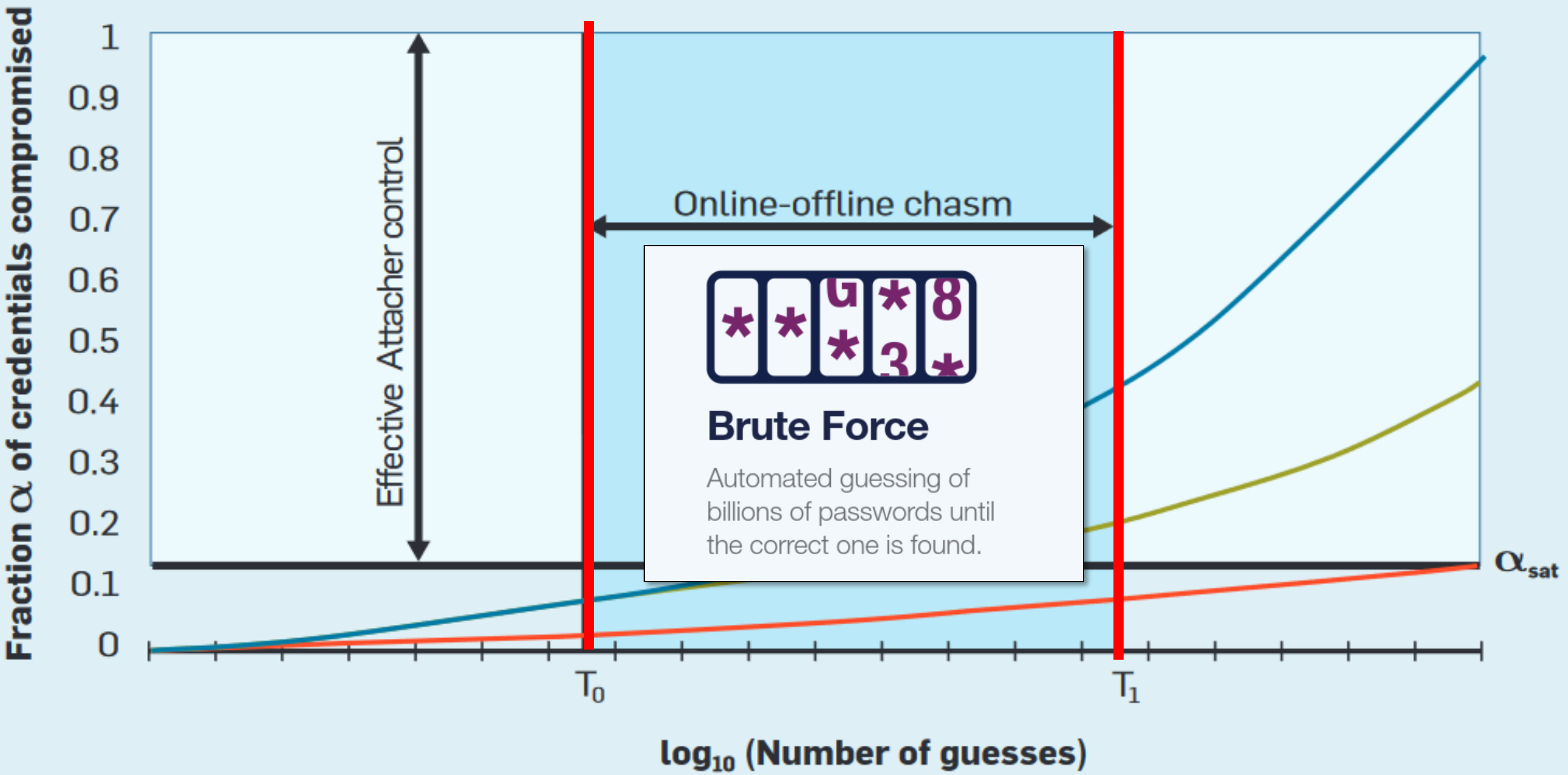
Figure 2. "Don't care" regions where there is no return for increasing effort.

T_0 is the threshold above which online attacks cease to be a threat.
T_1 is the threshold below which passwords almost surely will not survive credible offline attacks.
α_{sat} is the threshold fraction of compromised accounts at which an attacker effectively has control of system resources. Examples for these parameters might be $T_0 = 10^6$, $T_1 = 10^{14}$, and $\alpha_{sat} = 0.1$.



Pushing on string: The'don't care'region of password strength, D Florêncio, C Herley, PC Van Oorschot - Communications of the ACM, 2016

Figure 2. “Don’t care” regions where there is no return for increasing effort.



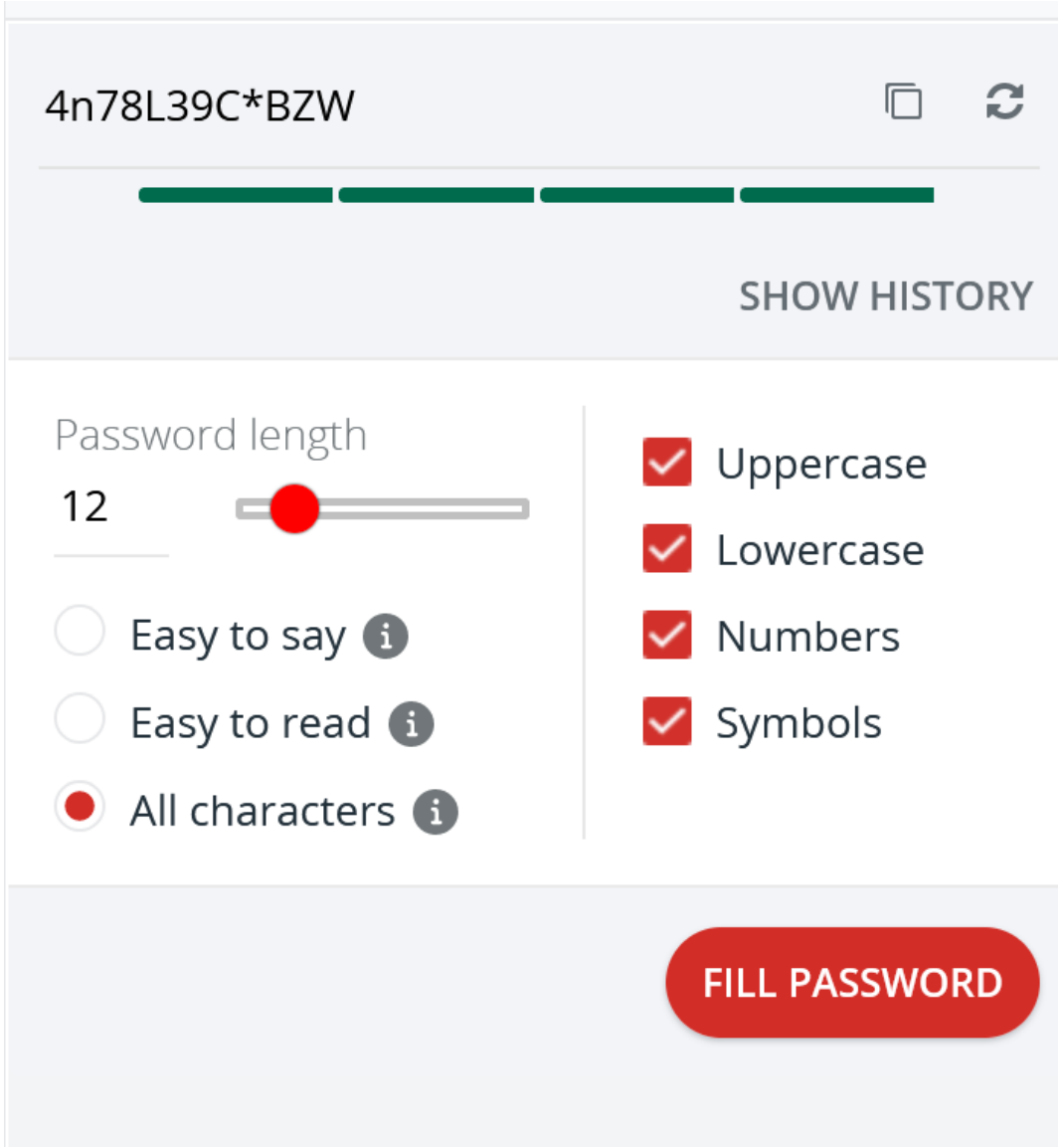
Pushing on string: The'don't care'region of password strength, D Florêncio, C Herley, PC Van Oorschot - Communications of the ACM, 2016

Lockout

- Password guessing attacks work because a computer can guess many times a second
- Humans don’t guess many times a second
- Idea: if a user can’t guess a password in 10 tries or less lock them out for a time period or require another factor

What does a password manager do?

- Generate passwords for you that are truly random (high entropy)
- Remember those passwords for you (no forgetting)
- Automatically insert the password into the website it goes to (computers are not fooled by phishing)
- Store the passwords somewhere outside your computer (safe from coffee spillage)
- Give anyone with the master password access to all your passwords (um…. Bad?)
- Allow you to have a unique password for every website (why is this important?)

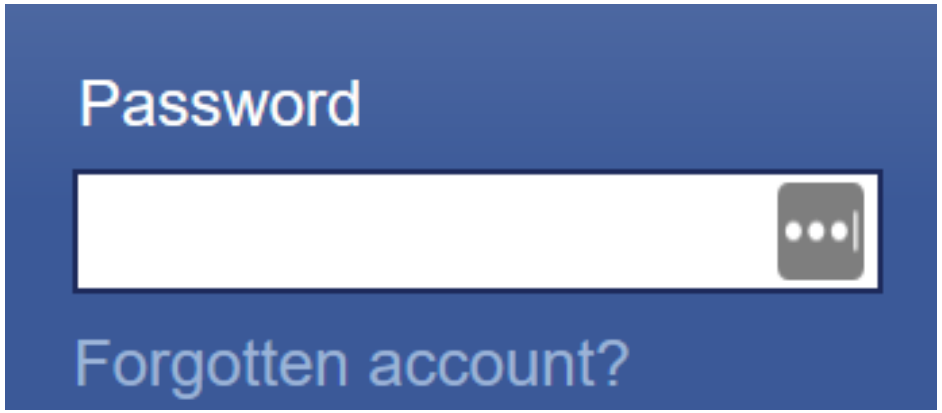


Graphical Passwords

Recognition vs Recall

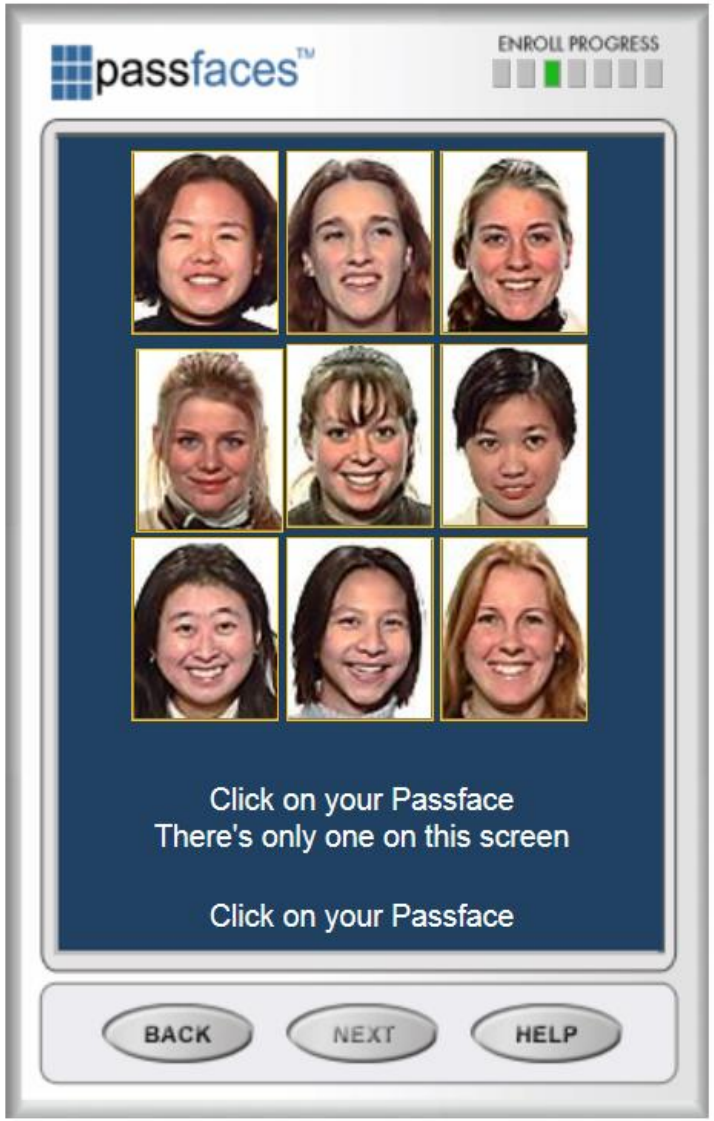
- Recognition – You are shown a set of things and asked to recognize the correct thing

- Recall – You must state the correct thing from memory



PassFaces

- Humans are better at recognizing things than they are at recalling information.
- High feature information, like faces, are easier to recognize
- Idea: Use high feature information as the pin, so humans can recognize their password
- Problem: People select faces that mean something to them. If you know basic characteristics about someone you can easily guess their PassFace.



PassFaces

- Password length = 4
- Each password selected from a set of 9 faces like what is shown on the right
- Theoretical password space = 6561
- What is the best way to break someone's password?
  - If the person is a white male, you can guess the correct password in about two guesses by selecting all the pretty white females.

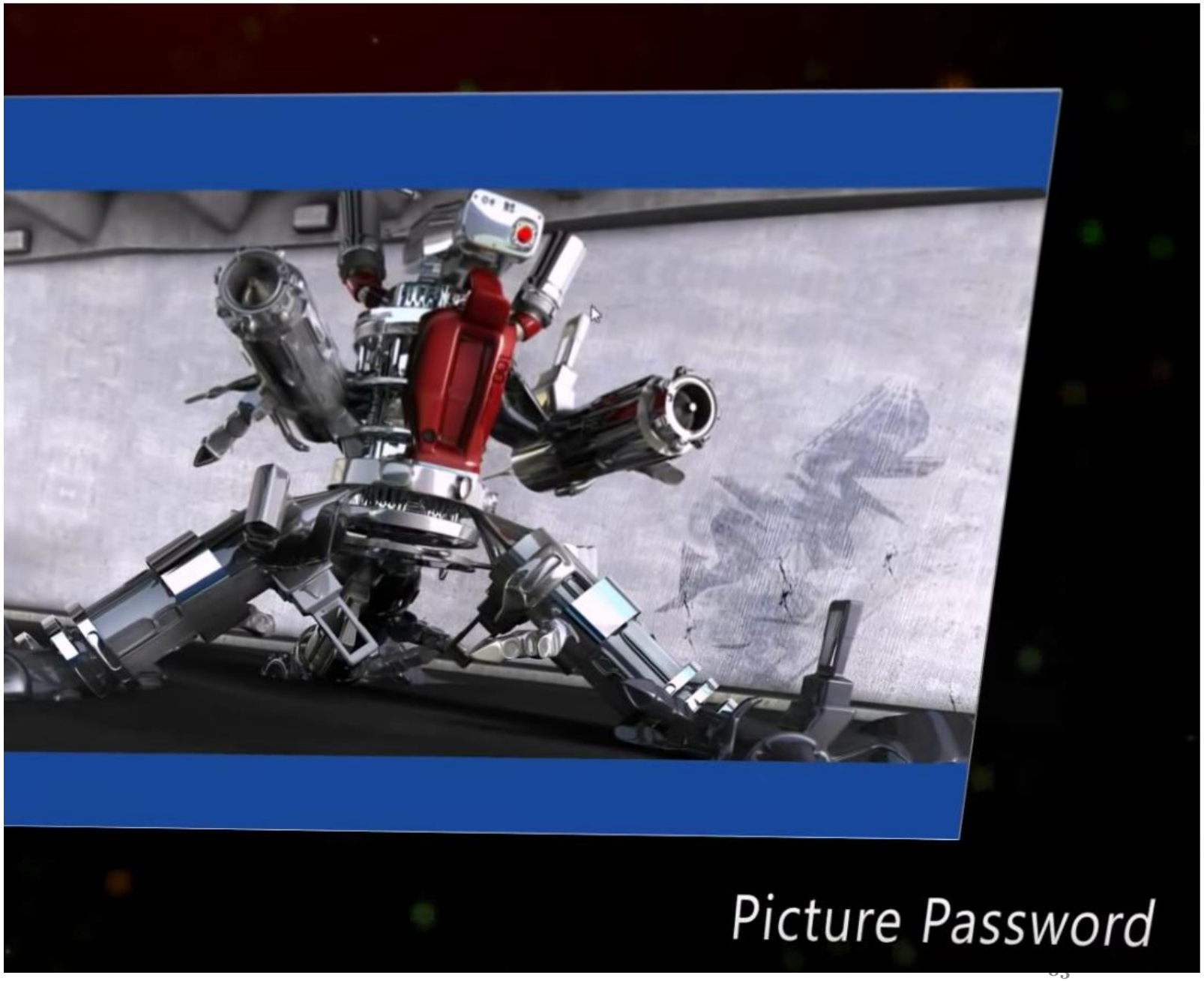
Authentication Using Graphical Passwords: Effects of Tolerance and Image Choice

Susan Wiedenbeck
Jim Waters
College of IST
Drexel University
Philadelphia, PA 19104
+1 215 895-2490
sw53@drexel.edu
jw65@drexel.edu

Jean-Camille Birget
Computer Science Department
Rutgers University
Camden, NJ
+1 856 225-6653
birget@camden.rutgers.edu

Alex Brodskiy
Nasir Memon
Computer Science Department
Polytechnic University
Brooklyn, NY 11201
+1 718 260-3970
abrods01@utopia.poly.edu
memon@poly.edu

Memorywise-Effortless	Usability
Scalable-for-Users	
Nothing-to-Carry	
Physically-Effortless	
Easy-to-Learn	
Efficient-to-Use	
Infrequent-Errors	
Easy-Recovery-from-Loss	
Accessible	Deployability
Negligible-Cost-per-User	
Server-Compatible	
Browser-Compatible	
Mature	
Non-Proprietary	
Resilient-to-Physical-Observation	Security
Resilient-to-Targeted-Impersonation	
Resilient-to-Throttled-Guessing	
Resilient-to-Unthrottled-Guessing	
Resilient-to-Internal-Observation	
Resilient-to-Leaks-from-Other-Verifiers	
Resilient-to-Phishing	
Resilient-to-Theft	
No-Trusted-Third-Party	
Requiring-Explicit-Consent	
Unlinkable	



Graphical passwords

Pros

- Easier to recall
- Theoretically a large password space
- Work well with touch screens

Cons

- Easier to guess
- Practically much smaller password space than theoretical
- Accessibility issues

Face ID

Apple FaceID

- FaceID is a convenience, it does not replace passwords
- Passwords are used as a backup and periodically required to be entered
- However, it does make it easier for someone else to login using your sleeping face, a photo, or the real you