CSE508 Network Security

Stony Brook University

2024-03-19

Authentication

Michalis Polychronakis

Stony Brook University

Authentication

The process of verifying someone's identity or role

- User, device, service, request, …

What is identity?

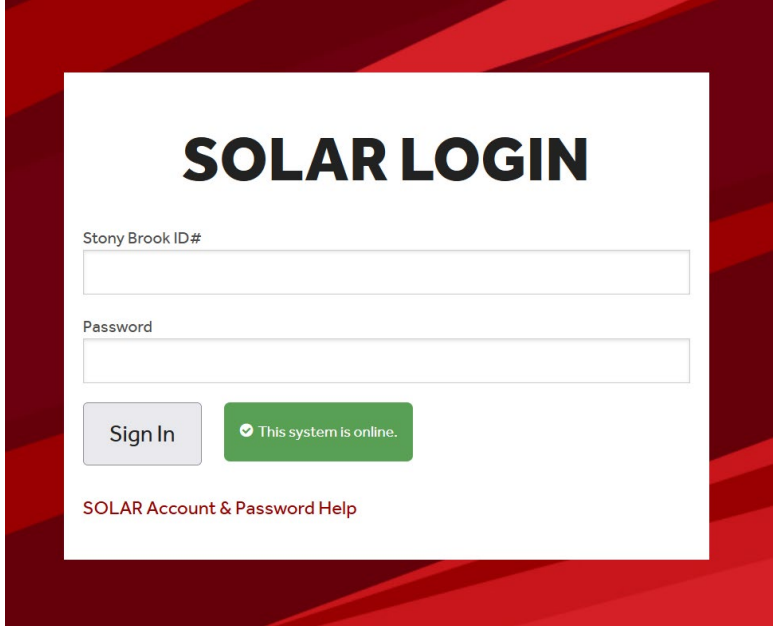
- Which characteristics uniquely identify an entity?

Authentication is a critical service

- Enables communicating parties to verify the identity of their peers
- Many other security mechanisms rely on it

Two main types

- Human to computer
- Computer to computer



Credentials

Evidence used to prove an identity

User Authentication: credentials supplied by a person

- *Something you know*
- *Something you have*
- *Something you are*

Computer authentication: cryptography, secret tokens, location, …

- Computers (in contrast to humans) can "remember" large secrets (keys or tokens) and perform complex cryptographic operations
- Location: evidence that an entity is at a specific place (IP, subnet, switch port, …)

Authentication can be delegated

- The verifying entity relies on a trusted third party to establish authentication ➔ Identity and Access Management (IAM) services (e.g., Okta, Duo, OneLogin)

Something You Know: Password-based Authentication

Passwords, passphrases, pins, key-phrases, access codes, …

Good passwords are easy to remember and hard to guess

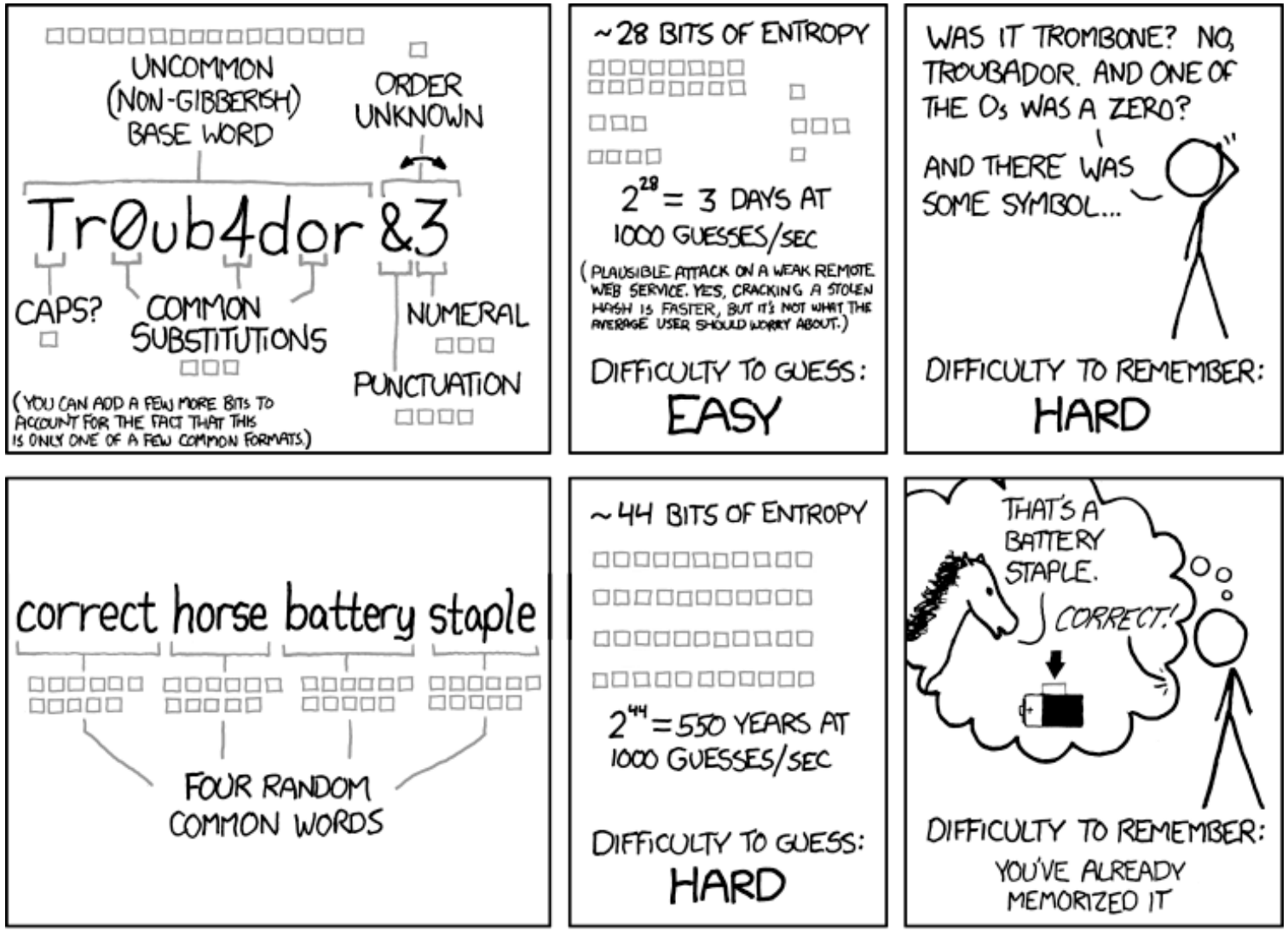
- Easy to remember ➔ easy to guess
- Hard to guess ➔ hard to remember
- Bad ideas: date of birth, SSN, zip code, favorite team name, …

Password space (bits) depends on:

- Password length
- Character set

Better way to think about strong passwords: **long passphrases**

- Can be combined with custom variations, symbols, numbers, capitalization, …



Password Meter - A visual asses

https://www.uic.edu/apps/strong-password/ 110%

Password Tr0ub4dor&3

Hide password

Complexity Very Strong

Score

Additions	Type	Rate	Count	Bonus
Number of characters	Flat	+(n*4)	11	+ 44
Uppercase letters	Cond/Incr	+((len-n)*2)	1	+ 20
Lowercase Letters	Cond/Incr	+((len-n)*2)	6	+ 10
Numbers	Cond	+(n*4)	3	+ 12
Symbols	Flat	+(n*6)	1	+ 6
Middle numbers or symbols	Flat	+(n*2)	3	+ 6
Requirements	Flat	+(n*2)	5	+ 10

Deductions	Type	Rate	Count	Bonus
Letters only	Flat	-n	0	0
Numbers only	Flat	-n	0	0
Repeat Characters (case insensitive)	Comp	-	2	- 1
Consecutive uppercase letters	Flat	-(n*2)	0	0
Consecutive lowercase letters	Flat	-(n*2)	3	- 6
Consecutive numbers	Flat	-(n*2)	0	0
Sequential letters (3+)	Flat	-(n*3)	0	0
Sequential numbers (3+)	Flat	-(n*3)	0	0
Sequential symbols (3+)	Flat	-(n*3)	0	0

```
mikepo@styx:~> zxcvbn
Password:
{
  "password": "Tr0ub4dor&3",
  "guesses": "100000000001",
  "guesses_log10": 11.000000000004341,
"crack_times_seconds": {
    "online_throttling_100_per_hour": "3600000000036.000199840144435",
    "online_no_throttling_10_per_second": "10000000000.1",
    "offline_slow_hashing_1e4_per_second": "10000000.0001",
    "offline_fast_hashing_1e10_per_second": "10.0000000001"
  },
  "crack_times_display": {
    "online_throttling_100_per_hour": "centuries",
    "online_no_throttling_10_per_second": "centuries",
    "offline_slow_hashing_1e4_per_second": "4 months",
    "offline_fast_hashing_1e10_per_second": "10 seconds"
  },
}
```

Password Meter - A visual asses

https://www.uic.edu/apps/strong-password/ 110%

Password correct horse battery staple

☐ Hide password

Complexity Good

Score

Additions	Type	Rate	Count	Bonus
Number of characters	Flat	+(n*4)	28	+ 112
Uppercase letters	Cond/Incr	+((len-n)*2)	0	0
Lowercase Letters	Cond/Incr	+((len-n)*2)	25	+ 6
Numbers	Cond	+(n*4)	0	0
Symbols	Flat	+(n*6)	0	0
Middle numbers or symbols	Flat	+(n*2)	0	0
Requirements	Flat	+(n*2)	2	0

Deductions	Type	Rate	Count	Bonus
Letters only	Flat	-n	28	- 28
Numbers only	Flat	-n	0	0
Repeat Characters (case insensitive)	Comp	-	20	- 2
Consecutive uppercase letters	Flat	-(n*2)	0	0
Consecutive lowercase letters	Flat	-(n*2)	24	- 48
Consecutive numbers	Flat	-(n*2)	0	0
Sequential letters (3+)	Flat	-(n*3)	0	0
Sequential numbers (3+)	Flat	-(n*3)	0	0
Sequential symbols (3+)	Flat	-(n*3)	0	0

```
mikepo@styx:~> zxcvbn
Password:
{
  "password": "correct horse battery staple",
  "guesses": "2138119689520000000000",
  "guesses_log10": 20.330032012866745,
"crack_times_seconds": {
    "online_throttling_100_per_hour": "7697230882272000427282.147568",
    "online_no_throttling_10_per_second": "21381196895200000000",
    "offline_slow_hashing_1e4_per_second": "213811968952000000",
    "offline_fast_hashing_1e10_per_second": "21381196895.2"
  },
  "crack_times_display": {
    "online_throttling_100_per_hour": "centuries",
    "online_no_throttling_10_per_second": "centuries",
    "offline_slow_hashing_1e4_per_second": "centuries",
    "offline_fast_hashing_1e10_per_second": "centuries"
  },
}
```

Password Policies (often have the opposite effect)

Password rules (miss the point)

"At least one special character," "Minimum/Maximum length of 8/12 characters," "Must contain at least one number," "Must contain at least one capital letter"

Makes passwords hard to remember! ➔ encourages password reuse

Better: encourage long passphrases, and evaluate strength on-the-fly

Periodic password changing (does more harm than good)

"You haven't changed your password in the last 90 days"

Probably too late anyway if password has already been stolen

Makes remembering passwords harder ➔ more password resets

Hinders the use of password managers (!)

What users do: `password1` ➔ `password2` ➔ `password3` ➔ …

NIST Special Publication 800-63B

Digital Identity Guidelines

Authentication and Lifecycle Management

Home

SP 800-63-3

SP 800-63A

SP 800-63B

SP 800-63C

Comment

If the chosen secret is found in the list, the CSP or verifier SHALL advise the subs
provide the reason for rejection, and SHALL require the subscriber to choose a di

Verifiers SHOULD offer guidance to the subscriber, such as a password-strength
memorized secret. This is particularly important following the rejection of a memo
modification of listed (and likely very weak) memorized secrets [Blacklists].

Verifiers SHALL implement a rate-limiting mechanism that effectively limits the number of failed authentication attempts that can be made on the subscriber's account as described in Section 5.2.2.

Verifiers SHOULD NOT impose other composition rules (e.g., requiring mixtures of different character types or prohibiting consecutively repeated characters) for memorized secrets. Verifiers SHOULD NOT require memorized secrets to be changed arbitrarily (e.g., periodically). However, verifiers SHALL force a change if there is evidence of compromise of the authenticator.

Verifiers SHOULD permit claimants to use "paste" functionality when entering a memorized secret. This facilitates the use of password managers, which are widely used and in many cases increase the likelihood that users will choose stronger memorized secrets.

In order to assist the claimant in successfully entering a memorized secret, the verifier SHOULD offer an option to display the secret — rather than a series of dots or asterisks — until it is entered. This allows the claimant to verify their entry if they are in a location where their screen is unlikely to be observed. The verifier MAY also permit the user's device to display individual entered characters for a short time after each character is typed to verify correct entry. This is particularly applicable on mobile devices.

The verifier SHALL use approved encryption and an authenticated protected channel when requesting memorized secrets in order to provide resistance to eavesdropping and MitM attacks.

Verifiers SHALL store memorized secrets in a form that is resistant to offline attacks. Memorized secrets SHALL be salted and hashed using a

Attacking Passwords

Offline cracking

Online guessing

} *Brute force attacks*

Eavesdropping

Capturing

Password Storage

Storing passwords as plaintext is disastrous

- Better way: store a cryptographic hash of the password

Even better: store the hash of a "salted" version of the password

- Defend against *dictionary attacks*: prevent *precomputation* of hash values (wordlists of popular passwords, rainbow tables, …)
- Unique salt per user (no need to be secret): even if two users happen to have the same password, their hash values will be different ➔ need to be cracked separately
- Salting *does not* make brute-force guessing a given password harder!

Username	Salt	Password hash
Bobbie	4238	h(4238, $uperman)
Tony	2918	h(2918, 63%TaeFF)
Mitsos	6902	h(6902, zour1da)
Mark	1694	h(1694, Rockybrook#1)

Password databases are still getting leaked…

Password Cracking

Exhaustive search ➔ infeasible for large password spaces

Dictionary attacks (words, real user passwords from previous leaks, …)

Variations, common patterns, structure rules

- Prepend/append symbols/numbers/dates, weird capitalization, l33tspeak, visually similar characters, intended misspellings, …

Target-specific information

- DOB, family names, favorite team, pets, hobbies, anniversaries, language, slang, …
- Easy to acquire from social networking services and other public sites
- **Particularly effective against "security questions"**

Advanced techniques

- Probabilistic context-free grammars, Markov models, …

example_hashes [hashcat

Secure | https://hashcat.net/wiki/doku.php?id=example_hashes

hashcat
advanced password recovery

hashcat Forums Wiki Tools Events

Recent changes Log In Sitemap

Example hashes

Table of Contents

If you get a "line length exception" error in hashcat, it is often because the hash mode that you have requested does not match the hash. To verify, you can test your commands against example hashes.

Unless otherwise noted, the password for all example hashes is **hashcat**.

Generic hash types

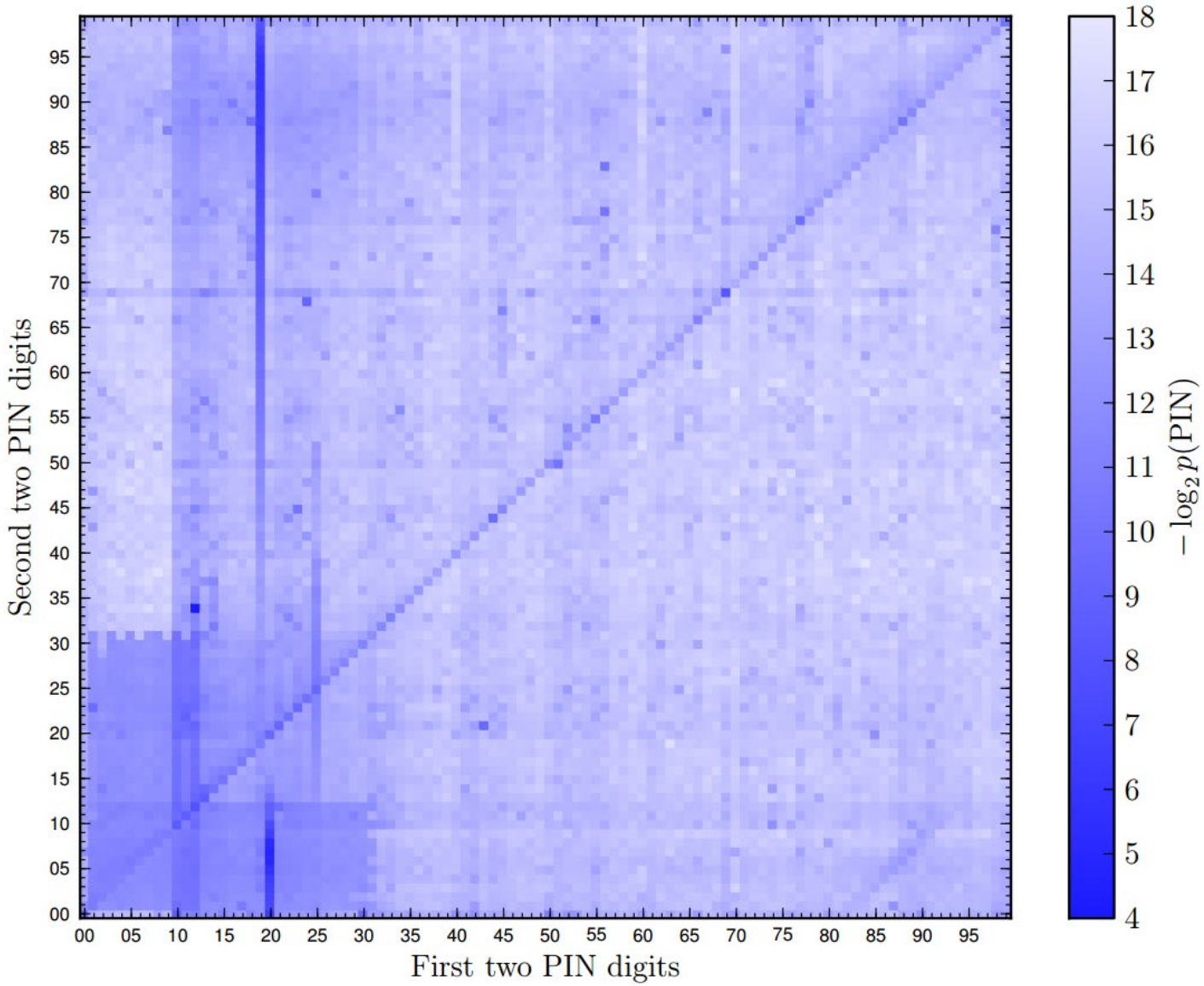
Hash-Mode	Hash-Name	Example
0	MD5	8743b52063cd84097a65d1633f5c74f5
10	md5($pass.$salt)	01dfae6e5d4d90d9892622325959afbe:7050461
20	md5($salt.$pass)	f0fda58630310a6dd91a7d8f0a4ceda2:4225637426
30	md5(utf16le($pass).$salt)	b31d032cfdcf47a399990a71e43c5d2a:144816
40	md5($salt.utf16le($pass))	d63d0e21fdc05f618d55ef306c54af82:13288442151473
50	HMAC-MD5 (key = $pass)	fc741db0a2968c39d9c2a5cc75b05370:1234
60	HMAC-MD5 (key = $salt)	bfd280436f45fa38eaacac3b00518f29:1234
100	SHA1	b89eaac7e61417341b710b727768294d0e6a277b
110	sha1($pass.$salt)	2fc5a684737ce1bf7b3b239df432416e0dd07357:2014
120	sha1($salt.$pass)	cac35ec206d868b7d7cb0b55f31d9425b075082b:5363620024
130	sha1(utf16le($pass).$salt)	c57f6ac1b71f45a07dbd91a59fa47c23abcd87c2:631225
140	sha1($salt.utf16le($pass))	5db61e4cd8776c7969cfd62456da639a4c87683a:8763434884872
150	HMAC-SHA1 (key = $pass)	c898896f3f70f61bc3fb19bef222aa860e5ea717:1234
160	HMAC-SHA1 (key = $salt)	d89c92b4400b15c39e462a8caa939ab40c3aeeea:1234
200	MySQL323	7196759210defdc0
300	MySQL4.1/MySQL5	fcf7c1b8749cf99d88e5f34271d636178fb5d130

50 Most-used (Worse) Passwords

123456	1234567	123	ashley	evite
123456789	qwerty	omgpop	987654321	123abc
picture1	abc123	123321	unknown	123qwe
password	Million2	654321	zxcvbnm	sunshine
12345678	000000	qwertyuiop	112233	121212
111111	1234	qwer123456	chatbooks	dragon
123123	iloveyou	123456a	20100728	1q2w3e4r
12345	aaron431	a123456	123123123	5201314
1234567890	password1	666666	princess	159753
senha	qqww1122	asdfghjkl	jacket025	0123456789

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

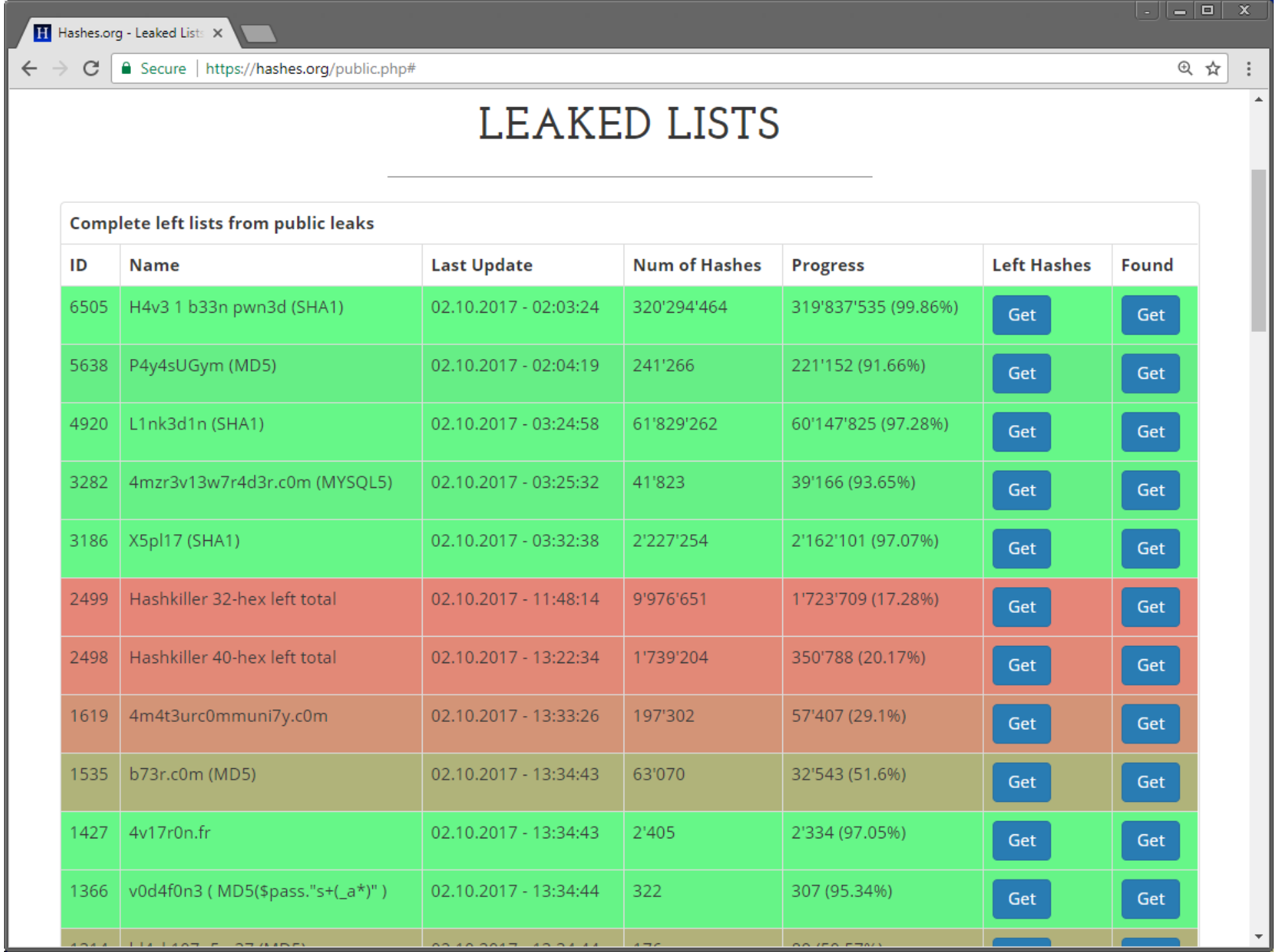
Distribution of 4-digit sequences within RockYou passwords



A birthday present every eleven wallets? The security of customer-chosen banking PINs. Joseph Bonneau, Sören Preibusch and Ross Anderson – FC '12

Wordlists

```
ce#ebc.dk
goddess5
20071002
271075711
zs3cu7za
scoopn
frygas1411
SL123456sl
12345687ee123
xuexi2010
daigoro
12345614
DICK4080
567891234
tilg80
6z08c861
:zark:
ravishsneha
150571611369
661189
passme
trolovinasveta
abdulkhaleque
007816
xLDSX
Florida2011
037037
WestC0untry
hitsugaiya
955998126
3n3rmax
4637324
bugger825
marmaris
jinjin111
170383gp
3484427
fl33321
zwqrfg
67070857
432106969
6856
704870704870
pv041886
20060814
512881535
milanimilani
472619
dbyxw888
85717221
cc841215
ariana19321
bbbnnn
ang34hehiu
wj112358
Brenda85
786525pb
shi461988
pingu
yeybozip
71477nak
stokurew
gea8mw4yz
kukumbike
260888
jordi10
lexusis
kj011a039
c84bwlrb
priyanka05
loveneverdies
u8Aqebj576
FGYfgy77
659397
327296
74748585
19720919
050769585
nicopa
2232566
bearss
n0tpublic
isitreal00
ashraf19760
48144
22471015
antyzhou115
0167005246
ec13kag
226226226226
6767537/33
mimilebrock
gueis8850
fujinshan
counter
N8mr0n
520057
adc123
bmaster
qbjh04zg
ueldaa79
EMANUELLI
yanjing
assynt
62157173
0704224950753
6903293
axaaxa
hilall
30091983
2510618981
soukuokpan
tosecondlife
p4os8m6q
015614117
acw71790
lsyljm2
2xgialdl
gaybar9
88203009
MKltyh87
quiggle
2063775206
fr3iH3it
masich
pengaiwei
coalesce
56402768
thesis
aabbcc894
marion&maxime
614850
ydz220105
584521584521
txudecp
84410545
pietro.chiara
jman1514
heryarma
39joinmam
timelapse
mwinkar
251422
willrock
YHrtfgDK
xys96exq
mercadotecnia
8s5sBEx7
0125040344
margitka
omaopa
dfTi6nh
1314520521
pixma760
pearpear
gothpunksk8er
rftaeo48
8d7R0K
5172032
aics07
34mariah
dongqinwei
samarica
cap10l4
0167387943
AE86Trueno
19700913
mcsuap
bu56mpbu
danbee
passw<>
money521
conan83
nxfjpl
rateg143
kojyihen
058336257
sarah4444
7363437
freindship
JytmvWO848
sb inbau
30907891
0515043111
1973@ati
wlxgjf
20081010
leelou44
8UfjeGbO
200358808
dellede
liang123.
captainettekt
kwiki-mart
mdovydas
tigmys2001
denial
678ad5251
woaiwuai
1591591591212
hNbDGN
cardcap
13985039393
001104
desare11
412724198
nibh1kab
asferg
hqb555
xgames7
muckerlee
choqui67
12130911
lierwei120
skytdvn
milena1995
kambala11
```



LEAKED LISTS

Complete left lists from public leaks

ID	Name	Last Update	Num of Hashes	Progress	Left Hashes	Found
6505	H4v3 1 b33n pwn3d (SHA1)	02.10.2017 - 02:03:24	320'294'464	319'837'535 (99.86%)	Get	Get
5638	P4y4sUGym (MD5)	02.10.2017 - 02:04:19	241'266	221'152 (91.66%)	Get	Get
4920	L1nk3d1n (SHA1)	02.10.2017 - 03:24:58	61'829'262	60'147'825 (97.28%)	Get	Get
3282	4mzr3v13w7r4d3r.c0m (MYSQL5)	02.10.2017 - 03:25:32	41'823	39'166 (93.65%)	Get	Get
3186	X5pl17 (SHA1)	02.10.2017 - 03:32:38	2'227'254	2'162'101 (97.07%)	Get	Get
2499	Hashkiller 32-hex left total	02.10.2017 - 11:48:14	9'976'651	1'723'709 (17.28%)	Get	Get
2498	Hashkiller 40-hex left total	02.10.2017 - 13:22:34	1'739'204	350'788 (20.17%)	Get	Get
1619	4m4t3urc0mmuni7y.c0m	02.10.2017 - 13:33:26	197'302	57'407 (29.1%)	Get	Get
1535	b73r.c0m (MD5)	02.10.2017 - 13:34:43	63'070	32'543 (51.6%)	Get	Get
1427	4v17r0n.fr	02.10.2017 - 13:34:43	2'405	2'334 (97.05%)	Get	Get
1366	v0d4f0n3 (MD5($pass."s+(_a*)")	02.10.2017 - 13:34:44	322	307 (95.34%)	Get	Get

Have I Been Pwned: Check if yo

https://haveibeenpwned.com

140%

755	13,044,161,748	115,769	228,884,627
pwned websites	pwned accounts	pastes	paste accounts

Largest breaches

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

Recently added breaches

49,102,176	Alleged AT&T accounts
3,262,980	ClickASnap accounts
552,094	Flipkart accounts
3,517,679	Habib's accounts
2,451,197	APK.TW accounts
3,805,265	Online Trade (Онлайн Трейд) accounts
21,994	WoTLabs accounts
27,123	Mr. Green Gaming accounts
19,972,829	Cutout.Pro accounts
243,462	Tangerine accounts

Password Hashing Functions

Hash functions are very fast to evaluate ➔ facilitate fast password cracking

Solution: slow down the guessing process (password *"stretching"*)

- Benefit: cracking becomes very inefficient (e.g., 10-100ms per check)
- Drawback: increased cost for the server if it must authenticate many users

Make heavy use of available resources

- Fast enough computation to validate honest users, but render password guessing infeasible
- Adaptable: flexible cost (time/memory complexity) parameters

Bcrypt [Provos and Mazières, 1999]

- Cost-parameterized, modified version of the Blowfish encryption algorithm
- Tunable cost parameter (exponential number of loop iterations)

Alternatives: Scrypt (memory-hard), PBKDF2 (PKCS standard)

Online Guessing

Similar strategy to offline guessing, but rate-limited

- Connect, try a few passwords, get disconnected, repeat…

Prerequisite: *know a valid user name*

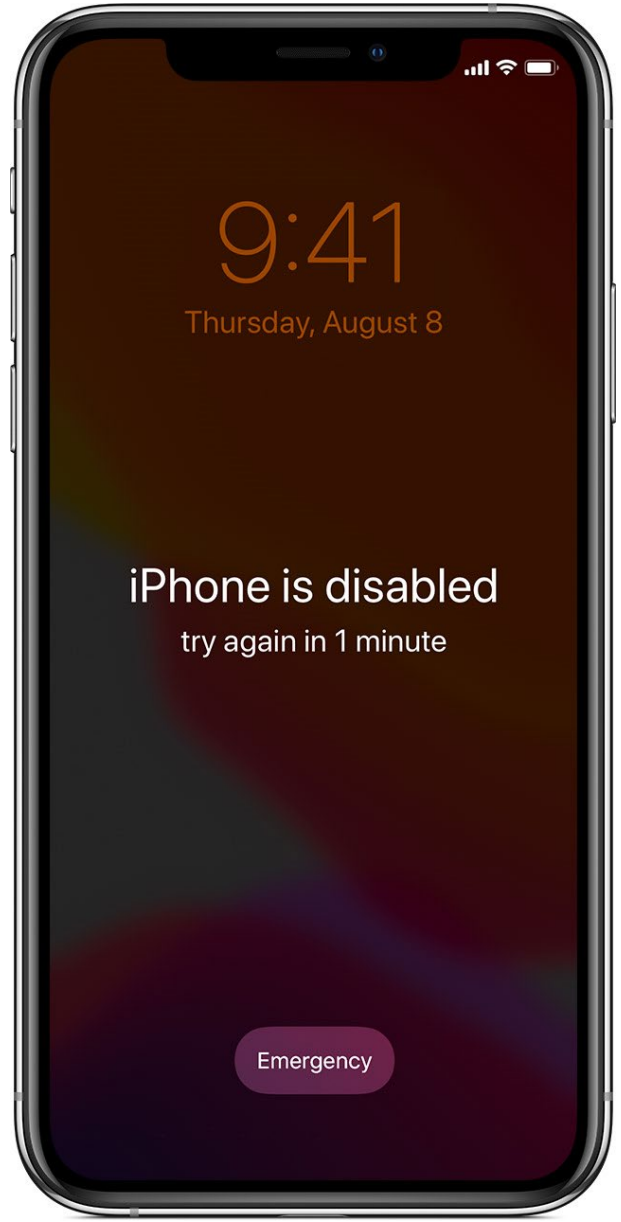
- **Credential stuffing**: try username + password combinations from previous breaches

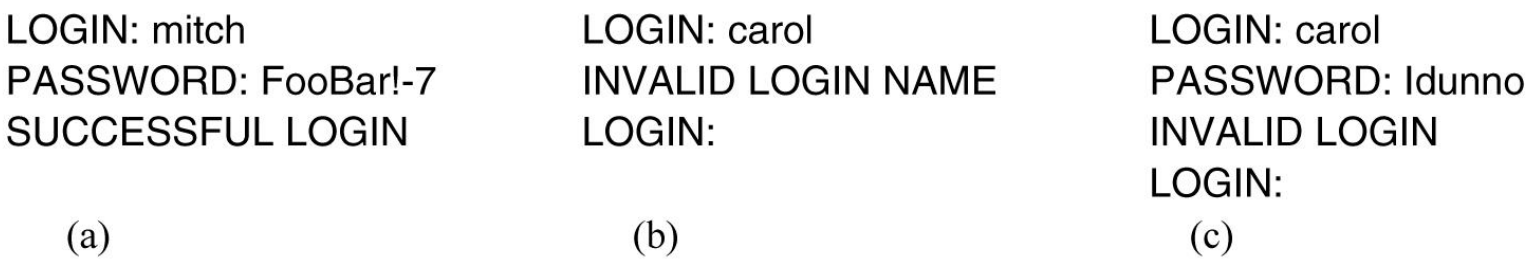
Many failed attempts can lead to a system reaction

- Introduce delay before accepting future attempts (exponential backoff)
- Shut off completely (e.g., ATM capturing/disabling the card after 3 tries)
- Ask user to solve a CAPTCHA

Very common against publicly accessible SSH, VPN, RDP, and other servers

- Main reason people move `sshd` to a non-default port
- Fail2Ban: block IP after many failed attempts ➔ attackers may now be able to lock you out
- Better: disable password authentication altogether and use a key pair ➔ cumbersome if having to log in from several devices or others' computers

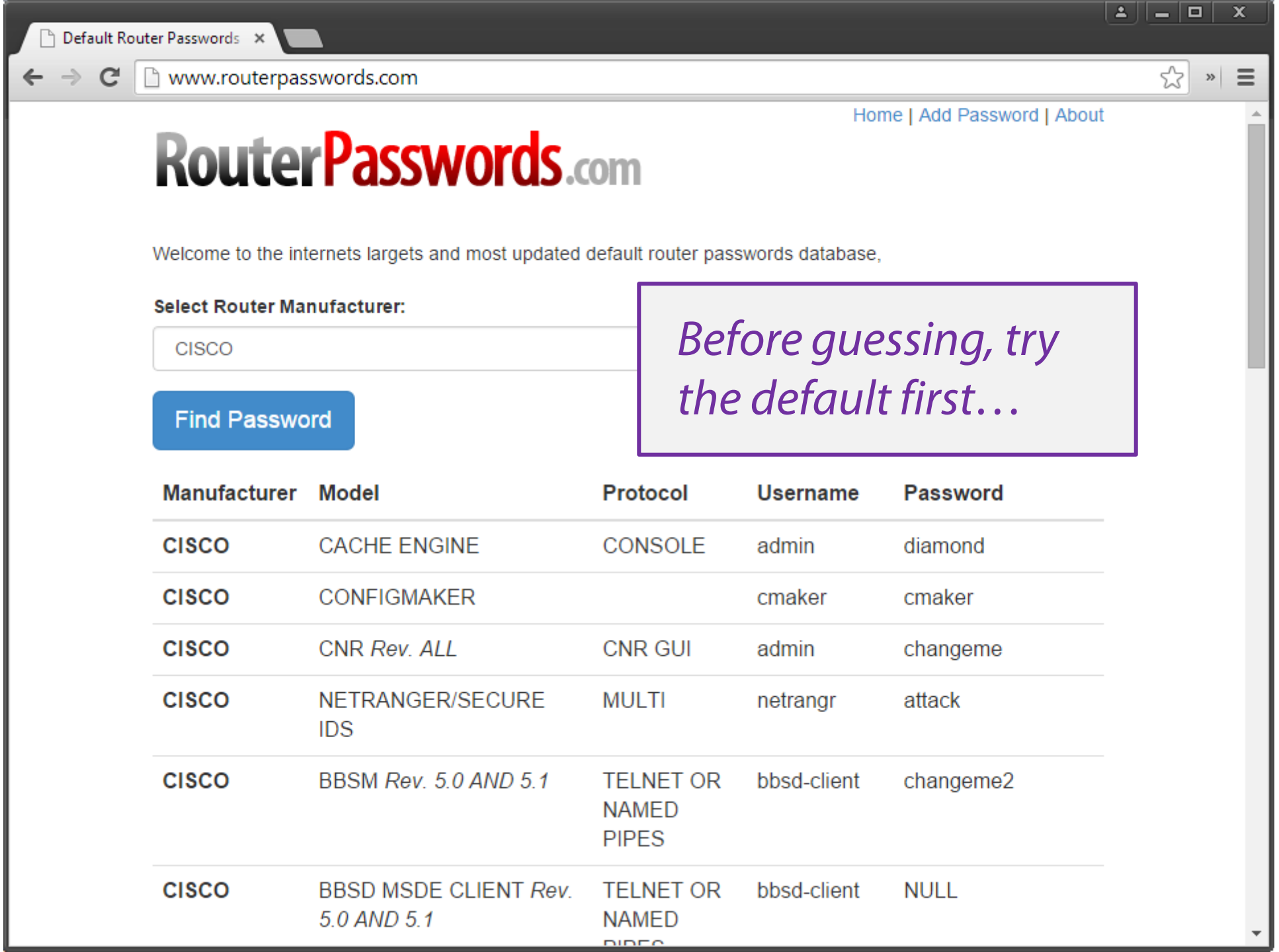

9:41
Thursday, August 8
iPhone is disabled
try again in 1 minute
Emergency



(a) Successful login

(b) Login rejected after name is entered

(c) Login rejected after name and password are typed ➔ less information makes guessing harder


Default Router Passwords
www.routerpasswords.com
Home | Add Password | About
RouterPasswords.com
Welcome to the internets largets and most updated default router passwords database,
Select Router Manufacturer:
CISCO
Find Password
Before guessing, try the default first…
Manufacturer Model Protocol Username Password
CISCO CACHE ENGINE CONSOLE admin diamond
CISCO CONFIGMAKER cmaker cmaker
CISCO CNR Rev. ALL CNR GUI admin changeme
CISCO NETRANGER/SECURE IDS MULTI netrangr attack
CISCO BBSM Rev. 5.0 AND 5.1 TELNET OR NAMED PIPES bbsd-client changeme2
CISCO BBSD MSDE CLIENT Rev. 5.0 AND 5.1 TELNET OR NAMED bbsd-client NULL

Eavesdropping and Replay

Physical world

- Post-it notes, notebooks, …
- Lift fingerprints (e.g., Apple Touch ID)

Network

- Sniffing (LAN, WiFi, …)
- Man-in-the-Middle attacks

Defenses

- Encryption
- One-time password schemes

Kerberos Network Authentication Protocol

Most widely used (non-web) single sign-on system

- Originally developed at MIT, now used in Unix, Windows, …

Long-lived vs. session keys

- Use long-lived key for authentication and negotiating session keys
- Use "fresh," ephemeral session keys for encrypted communication, MACs, …
- Prevent replay, cryptanalysis, old compromised keys

Authenticate users to services: using their password as the initial key, without having to retype it for every interaction

- A Key Distribution Center (KDC) acts as a trusted third party for key distribution
- Online authentication: variant of Needham-Schroeder protocol
- Assumes a non-trusted network: prevents eavesdropping
- Assumes that the Kerberos server and user workstations are secure…

Use cases: workstation login, remote share access, printers, …

https://web.mit.edu/kerberos/

Password Capture

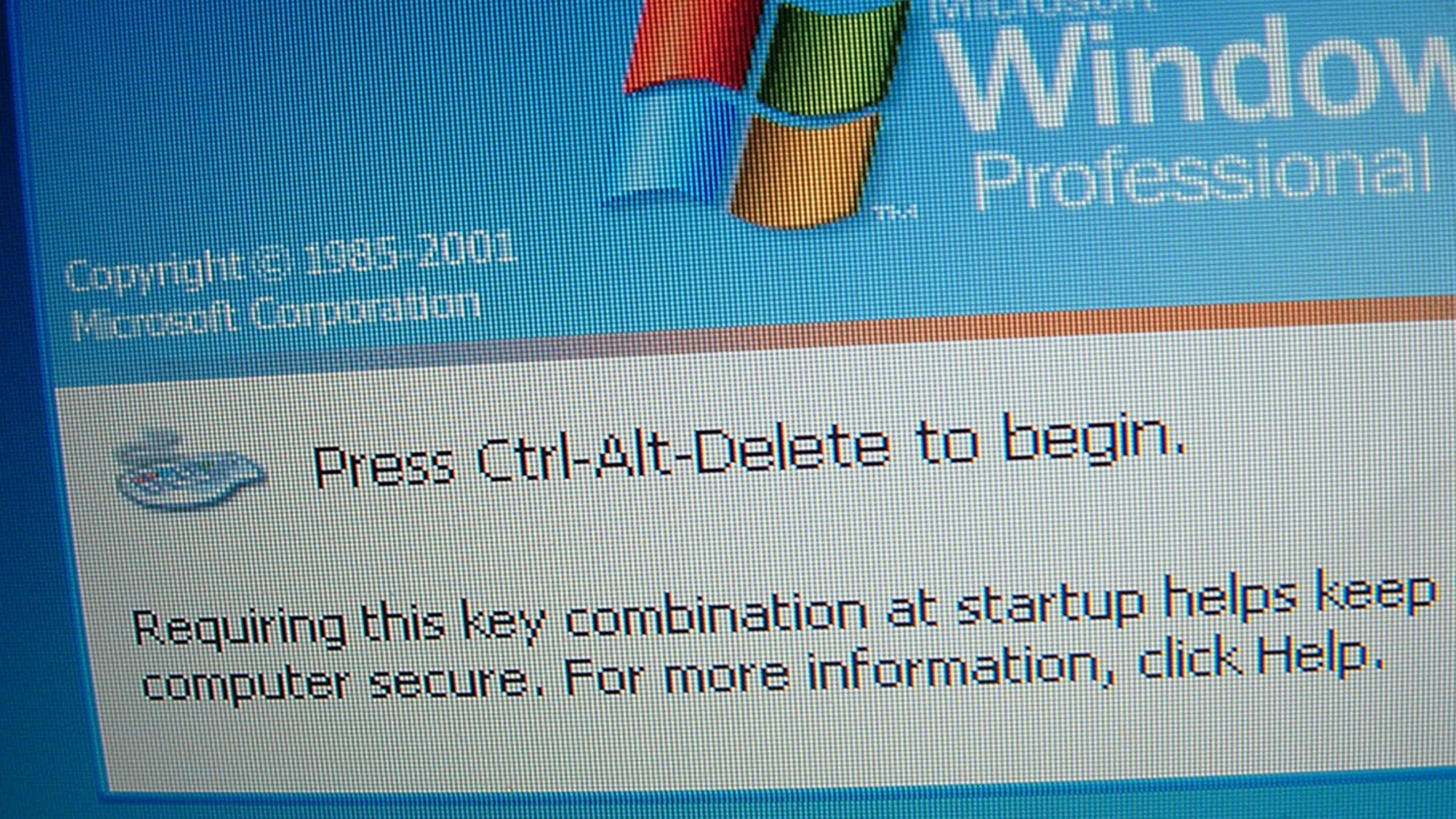
Hardware bugs/keyloggers

Software keyloggers/malware

Shoulder surfing

Cameras (e.g., ATM skimmers)

Social engineering


Windows
Professional
Copyright © 1985-2001
Microsoft Corporation
Press Ctrl-Alt-Delete to begin.
Requiring this key combination at startup helps keep your computer secure. For more information, click Help.

(a) Correct login screen

(b) Phony login screen

Something You Have: Authentication Tokens

One-time passcode tokens

- Time-based or counter-based

Various other authentication tokens

- Store certificates, encryption keys, challenge–response, …

Smartcards (contact or contactless)

- Identification, authentication, data storage, limited processing
- Magnetic stripe cards, EMV (chip-n-pin credit cards), SIM cards, RFID tags, …

USB/BLE/NFC tokens, mobile phones, watches, …

- Can be used as authentication devices

Something You Are: Biometrics

Fingerprint reader

Face recognition

- Depth sensing, infrared cameras, …
- Liveness detection (pulse, thermal) to foil simple picture attack

Retina/iris scanner

~~Voice recognition~~ → broken

…

Related concept: continuous authentication

- Keystroke timing, usage patterns, …

“The probability that a random person the population [sic] could look at your iPhone X and unlock it using Face ID is approximately 1 in 1,000,000 (versus 1 in 50,000 for Touch ID).

For additional protection, Face ID allows only five unsuccessful match attempts before a passcode is required to obtain access to your iPhone.

The probability of a false match is different for twins and siblings that look like you as well as among children under the age of 13, because their distinct facial features may not have fully developed. If you're concerned about this, we recommend using a passcode to authenticate.”

https://www.apple.com/business-docs/FaceID_Security_Guide.pdf

How I Broke Into a Bank Account With an AI-Generated Voice

Banks in the U.S. and Europe tout voice ID as a secure way to log into your account. I proved it's possible to trick such systems with free or cheap AI-generated voices.

By Joseph Cox

February 23, 2023, 11:44am

The bank thought it was talking to me; the AI-generated voice certainly sounded the same.

Multi-factor Authentication

Must provide several separate credentials of different types

Most common: *two-factor authentication (2FA)*

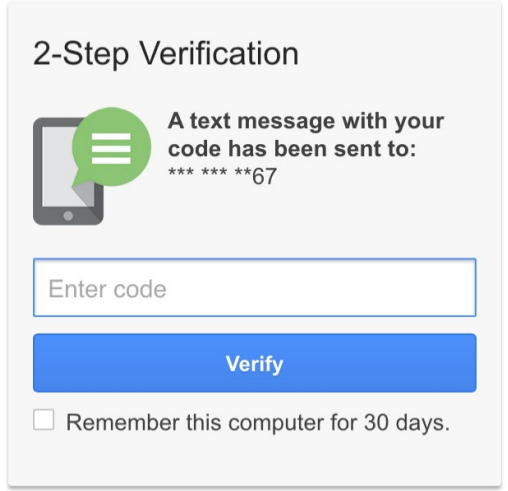
- Example: Password + hardware token/SMS message/authenticator app, …
- Example: ATM card + PIN

Motivation: a captured/cracked password is now not enough to compromise a victim's account ➔ **not always true**

- *Man-in-the-Middle:* set up fake banking website, relay password to real website, let the user deal with the second factor…
- *Man-in-the-Browser:* hijack/manipulate an established web session after authentication has been completed (malware, e.g., banking trojans)
- *Dual infection:* compromise both PC and mobile device (rare)
- **More importantly: the most commonly used 2nd factor (SMS) is the least secure**

SMS Is Not a Secure 2nd Factor

(but still better than no 2nd factor)



Social engineering

Call victim's mobile operator and hijack the phone number

SIM swaping, message/call forwarding, …

Message interception

Rogue cell towers: IMSI catchers, StingRays,…

Some phones even display text messages on the lock screen (!)

SS7 attacks

The protocol used for inter-provider signaling is severely outdated and vulnerable

Allows attackers to spoof change requests to users' phone numbers and intercept calls or text messages

'Sim swap' gives fraudste

Secure | https://www.theguardian.com/money/2015/sep/26/sim-swap-fraud-mobile-phone-vodafone-customer

sign in | become a supporter | subscribe | search | jobs | US edition

theguardian

US | politics | world | opinion | sports | soccer | tech | arts | lifestyle | fashion | business | travel | environment | all sections

home

Scams

'Sim swap' gives fraudsters access-all-areas via your mobile phone

There's a new, little-known scam designed to empty your bank account, as one Vodafone customer found to her cost

1908 15

Anna Tims

Saturday 26 September 2015 02.00 EDT

Most popular in US

Las Vegas shooting: death toll rises to 58 as police name suspect – latest updates

Confusion follows reports of Tom Petty death after heart attack

Las Vegas gunman may have used special device to fire faster, expert says

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.

By Joseph Cox

March 15, 2021, 1:10pm Share Tweet Snap

I hadn't been SIM swapped, where hackers trick or bribe telecom employees to port a target's phone number to their own SIM card. Instead, the hacker used a service by a company called Sakari, which helps businesses do SMS marketing and mass messaging, to reroute my messages to him. This overlooked attack vector shows not only how unregulated commercial SMS tools are but also how there are gaping holes in our telecommunications infrastructure, with a hacker sometimes just having to pinky swear they

Jack Schickler / CoinDesk:

Hackers access some customer data at FTX, Genesis, and BlockFi by SIM swapping an employee of Kroll, which manages creditor claims for the bankrupt companies — A 'cybersecurity incident' affected Kroll, which gathers customer claim data on behalf of bankrupt companies. — Register Now

Aug 25, 2023, 7:40 PM – In context

Iain Martin / Forbes:

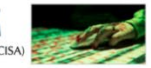
Blockchain Capital co-founder Bart Stephens sues a hacker who stole $6.3M in crypto via a SIM-swap attack; FBI: $72M was stolen via SIM swaps in 2022, up 6% YoY — Bart Stephens, cofounder and managing partner of crypto fund Blockchain Capital who was an early and prominent evangelist for cryptocurrencies …

Aug 21, 2023, 1:06 PM – In context

Emma Roth / The Verge:

CISA releases a report detailing Lapsus$'s key techniques, calls for passwordless logins, and asks the FTC and the FCC for stricter SIM swapping protections — The US Cybersecurity and Infrastructure Security Agency (CISA) is calling for stricter SIM swapping protections and the transition …

Aug 10, 2023, 12:10 PM – In context



Bloomberg:

How members of the Community, a group of teenage SIM swappers who met on the forum OGUsers, stole millions in crypto in 2018 before turning on each other — Michael Terpin lost a fortune to a phone "SIM swap." When he went to war to get it back, he found some surprising allies.

Aug 5, 2023, 9:45 AM – In context

David Canellis / The Next Web:

Federal judge refuses to dismiss $224M lawsuit against AT&T for allegedly letting a customer be SIM-swapped twice, leading to the loss of $24M in cryptocurrency — THIS INVESTOR LOST $24M IN CRYPTOCURRENCY AFTER HE WAS SIM-SWAPPED, TWICE — A US federal judge has rejected AT&T's request …

Jul 23, 2019, 12:06 PM – In context

Andy Greenberg / Wired:

While many foreign phone carriers are sharing real-time SIM swap data with banks to stop financial fraud, US carriers are dragging their feet — AROUND A YEAR ago, André Tenreiro was called into a meeting between the chief technology officer of the phone carrier he worked for …

Apr 27, 2019, 12:00 PM – In context

Joseph Cox / VICE:

A look at so-called Russian, encrypted, or "white" SIMs, used by criminals to spoof phone numbers, add voice manipulation to calls in real-time, and more — Criminals use so-called Russian, encrypted, or white SIMs to change their phone number, add voice manipulation to their calls …

Aug 12, 2020, 11:15 AM – In context

Catalin Cimpanu / ZDNet:

Europol, working with US, UK, and others, says 10 people have been arrested for allegedly stealing $100M in cryptocurrency from celebrities via SIM-swap attacks — Eight men were arrested in England and Scotland as part of an investigation into a series of SIM swapping attacks targeting US celebrities.

Feb 10, 2021, 2:45 PM – In context

Stefanie Marotta / Bloomberg:

Canada arrests a teenager for allegedly stealing $36.5M in crypto from a US victim using SIM swapping, the largest reported single-person crypto theft — - Case marks the biggest crypto theft reported by one person — Police identified the alleged thief through a gaming username

Nov 18, 2021, 10:35 AM – In context

Lorenzo Franceschi-Bicchierai / VICE:

Europol, alongside Italian and Spanish police, arrest 106 people accused of working for the Italian Mafia and laundering over €10M made through cybercrimes — European police accused several people of SIM swapping, phishing, and hacking in support of Italian organized crime.

Sep 20, 2021, 11:00 AM – In context

Results 1 - 10 of about 477:

Ashley Belanger / Ars Technica:

The US indicts a Chicago man who allegedly led a SIM-swap gang; members stole millions and posed as other people in Apple, T-Mobile, AT&T, and Verizon stores — Scheme allegedly targeted Apple, AT&T, Verizon, and T-Mobile stores in 13 states. — The US may have uncovered the nation's largest …

Jan 30, 2024, 6:55 PM – In context

MacKenzie Sigalos / CNBC:

The US SEC says the January 9 hack of its X account was via a SIM swap attack to reset its password; it had disabled 2FA in July 2023 over account access issues — - The U.S. Securities and Exchange Commission said on Monday that a SIM swap attack was to blame for the breach of its official account on X …

Jan 22, 2024, 4:35 PM – In context

Gary Miller / The Citizen Lab:

Research details how vulnerabilities in signaling protocols used by mobile network operators for international roaming can be exploited to geolocate devices — Table of Contents — 2. Geolocation Attacks Against Telecommunications Networks — 4. Incentives Enabling Geolocation Attacks

Oct 29, 2023, 2:35 PM – In context

Lorenzo Franceschi-Bicchierai / Motherboard:

A college student who stole $5M+ in cryptocurrency via SIM hijacking gets 10 years in prison and is the first person in the US to be sentenced for such a crime — A 20-year-old college student who was accused of stealing more than $5 million in cryptocurrency in a slew of SIM hijacking attacks …

Feb 3, 2019, 7:55 PM – In context

Brian Krebs / Krebs on Security:

A detailed look at SIM swapping, a complex form of mobile phone fraud often used to steal cryptocurrency and other items of value — KrebsOnSecurity recently had a chance to interview members of the REACT Task Force, a team of law enforcement officers and prosecutors based in Santa Clara …

Nov 10, 2018, 8:00 AM – In context

Catalin Cimpanu / ZDNet:

Researchers: AT&T, T-Mobile, Tracfone, US Mobile, and Verizon use vulnerable procedures for customer support that put users at risk of SIM swapping attacks — Researchers find that 17 of 140 major online services are vulnerable to SIM swapping attacks.

Jan 13, 2020, 3:20 PM – In context

New York Times:

Profile of Twitter hack mastermind Graham Clark, a troubled teen who allegedly went from scamming on Minecraft to joining hacker forum OGUsers and SIM swapping — The teenage "mastermind" of the recent Twitter breach, who had a difficult family life, poured his energy into video games and cryptocurrency.

Aug 2, 2020, 8:11 PM – In context

Brian Krebs / Krebs on Security:

The Twitter attack may have been perpetrated by Joseph James Connor, a 21-year-old English SIM swapper linked to a group that hijacked @jack's account last year — Twitter was thrown into chaos on Wednesday after accounts for some of the world's most recognizable public figures …

Jul 16, 2020, 7:00 PM – In context

Sergiu Gatlan / BleepingComputer:

T-Mobile confirms reports of a data breach caused by SIM swap attacks on a "very small number of customers", following six other data breaches since 2018 — T-Mobile confirmed that recent reports of a new data breach are linked to notifications sent to a "very small number of customers" who fell victim to SIM swap attacks.

Dec 29, 2021, 12:50 PM – In context

Brian Krebs / Krebs on Security:

Three SIM-swapping gangs separately claimed multiple times on Telegram to have phished staff at T-Mobile throughout 2022, far more often than other US carriers — Three different cybercriminal groups claimed access to internal networks at communications giant T-Mobile in more than 100 …

Feb 28, 2023, 12:35 PM – In context

https://keydiscussions.com/2024/02/05/sim-swap-attacks-can-be-blamed-on-companies-embracing-sms-based-password-resets/

SMS as 2nd Factor vs. SMS for Account Recovery

Despite its shortcomings, SMS as a *2nd factor* is better than nothing

- Data point (Google): prevented 100% of 3.3B automated password stuffing attacks, 96% of 12M bulk phishing, and even 76% of <10k targeted attacks seen over a year

Unfortunately, the convenience of phone numbers has led many services to overload SMS as the *sole authentication factor*

- SMS-based onboarding
- SMS-based authentication (login with phone number)
- SMS-based password reset/account recovery

These are disastrous: a simple SIM-swap attack can take over an account **without knowing the password**

- Password reset via email is much more secure

Better Alternative: Authenticator Apps

Time-based one-time password (TOTP)

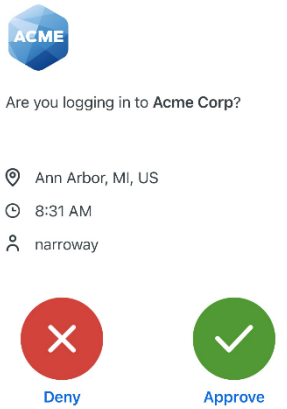
- Six/eight digit code provided after password validation
- HMAC of a shared secret key and the current time
- The key is negotiated during registration

Requires "rough" client–server synchronization

- Code constantly changes in 30-second intervals

User-friendly alternative: push notification (e.g., Duo Push)

- MFA "fatigue" attacks: flood a user with push notifications



More importantly: **Phishing is still possible!**

- The attacker just needs to proxy the captured credentials in real time (rather than collecting them for later use)

MFA fatigue attacks: Users tricked into allowing device access due to overload of push notifications

Jessica Haworth 16 February 2022 at 15:40 UTC
Updated: 18 February 2022 at 14:24 UTC

2FA Research Social Engineering

Social engineering technique confuses victims to gain entry to their accounts

Malicious hackers are targeting Office 365 users with a spare of 'MFA fatigue attacks', bombarding victims with 2FA push notifications to trick them into authenticating their login attempts.

This is according to researchers from GoSecure, who have warned that there is an increase in attacks that are exploiting human behavior to gain access to devices.

Multi-factor authentication (MFA) fatigue is the name given to a technique used by adversaries to flood a user's authentication app with push notifications in the hope they will accept and therefore enable an attacker to gain entry to an account or device.

In a blog posted earlier this week, GoSecure described the attack as "simple", given that "it only requires the

Latest Posts

We're going teetotal – It's goodbye to The Daily Swig
02 March 2023

Uber: Lapsus$ Targeted External Contractor With MFA Bombing Attack

The ride-sharing giant says a member of the notorious Lapsus$ hacking group started the attack by compromising an external contractor's credentials, as researchers parse the incident for takeaways.

Jai Vijayan, Contributing Writer
September 19, 2022

Uber has attributed last week's massive breach at U
Lapsus$ hacking group and released additional deta
Researchers say the incident has highlighted the risk
trusting too much in multifactor authentication (MFA
risk around cloud-service adoption.

The attacker then repeatedly tried to log in to the Uber account using the illegally obtained credentials, prompting a two-factor login approval request each time. After the contractor initially blocked those requests, the attacker contacted the target on WhatsApp posing as tech support, telling the person to accept the MFA prompt — thus allowing the attacker to log in.

In an update on Monday, Uber laid out the attribution: "We believe that this attacker (or attackers) are affiliated with a hacking group called Lapsus$, which has been increasingly active over the last year or so." Uber's announcement pointed to other companies that had been targeted by the notorious gang via similar techniques, including Cisco, Microsoft, Nvidia, Okta, and Samsung,

Lapsus$ has attracted considerable attention in recent months for its brazen attacks on some of the world's largest and well-known companies. One well-known tactic that the group has been known to use is co-opt MFA-

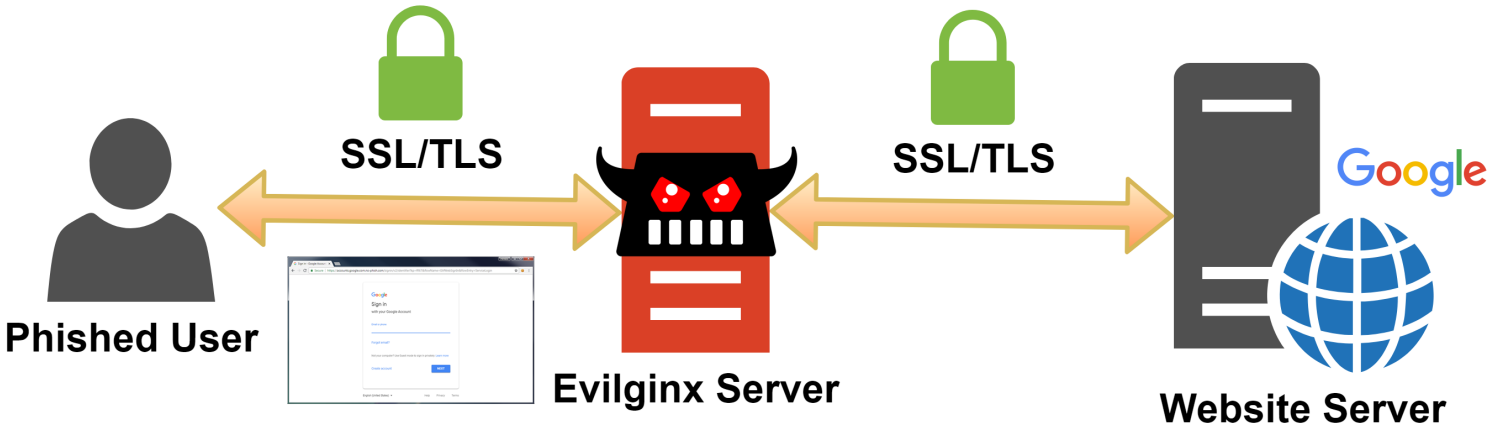
Evilginx2 https://github.com/kgretzky/evilginx2

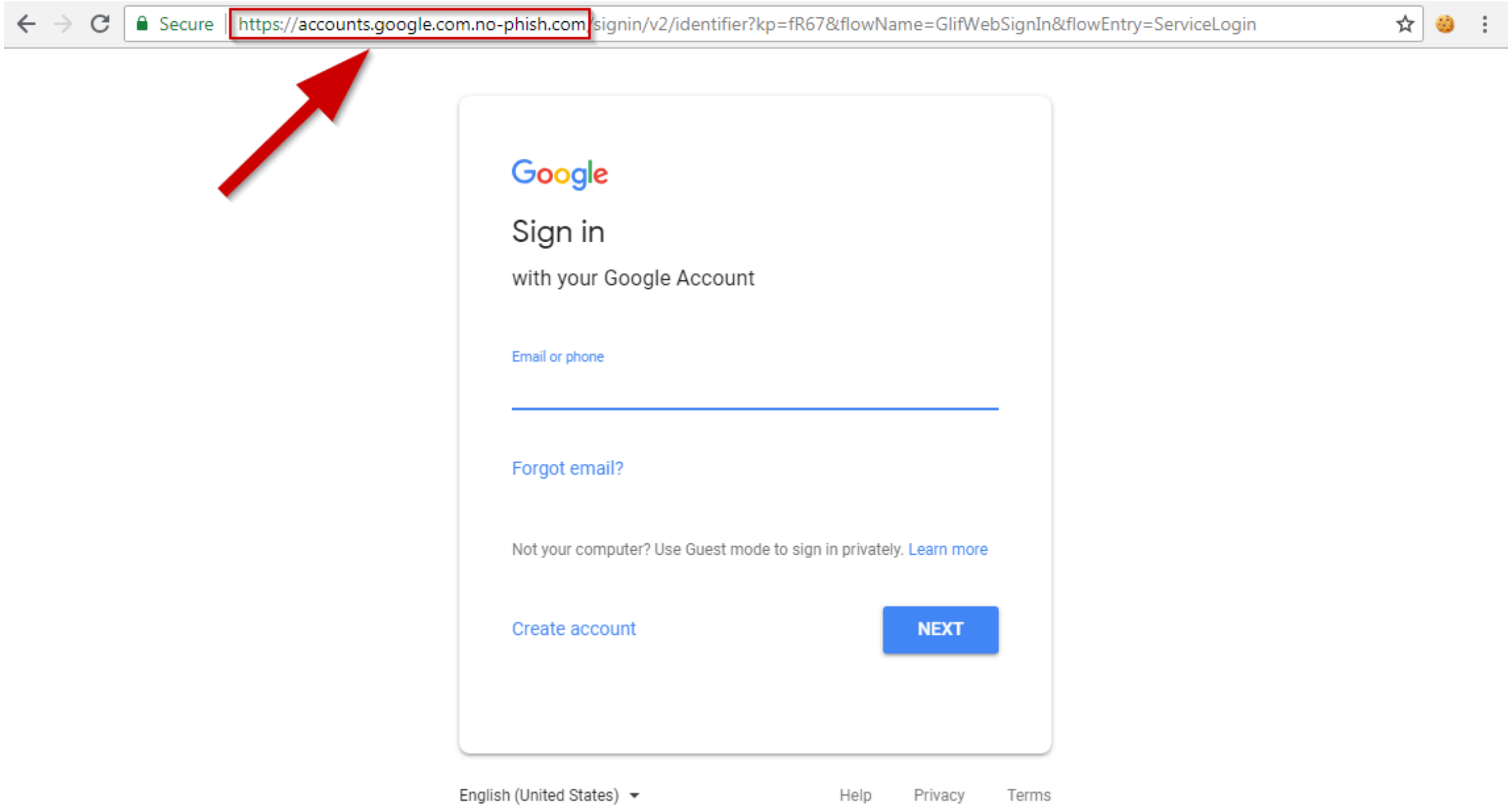
Man-in-the-middle attack framework for phishing login credentials along with session cookies

- Bypasses 2-factor authentication

No need for HTML templates: just a web proxy

- Victim's traffic is forwarded to the real website
- TLS termination at the proxy (e.g., using a LetsEncrypt certificate)





Even Better Alternative: U2F Tokens (AKA Security Keys)

Universal Second Factor (U2F)

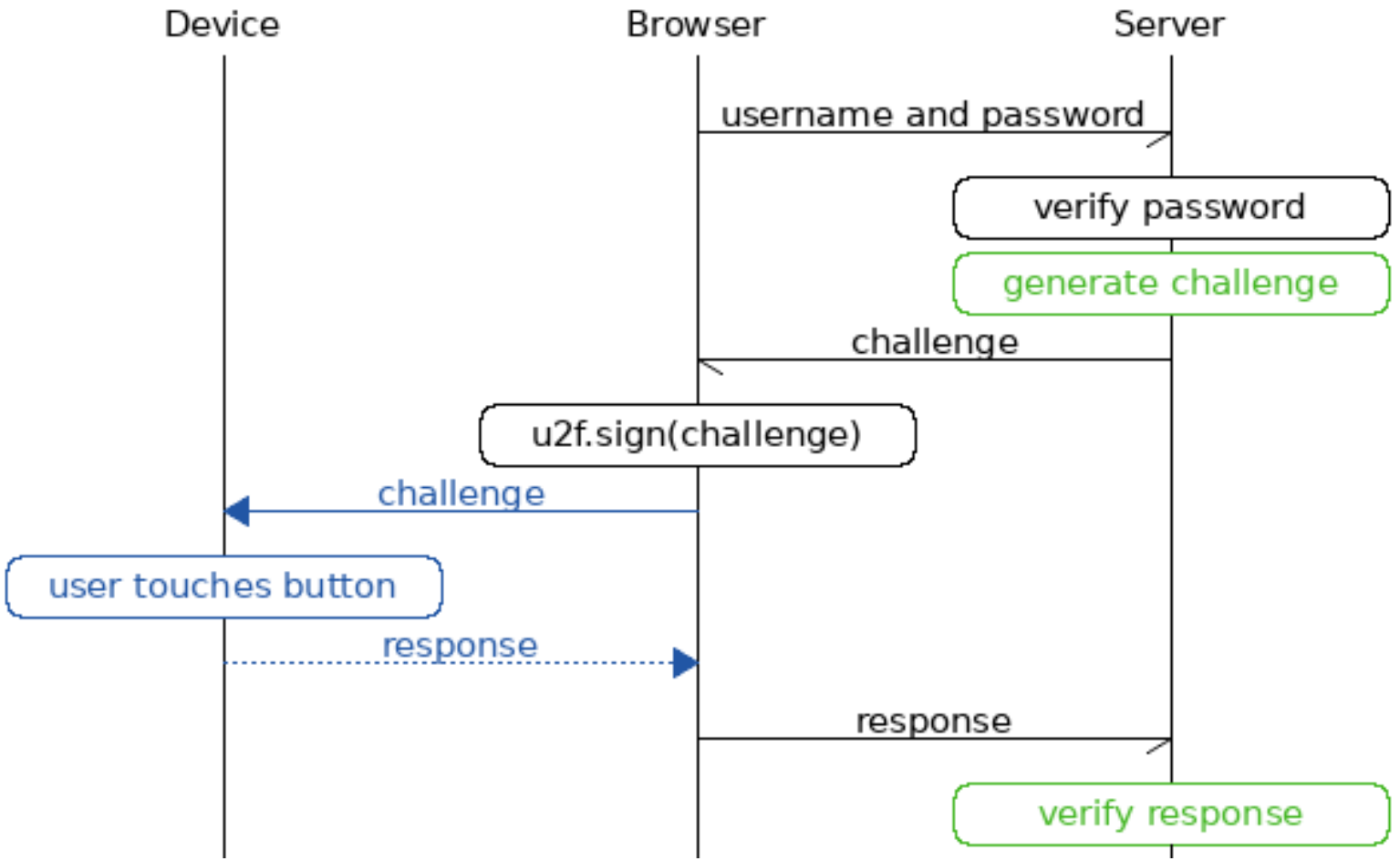
- FIDO (Fast IDentity Online) alliance: Google, Yubico, …
- Supported by all popular browsers and many online services

A different key pair is generated for each origin during registration

- Origin = <protocol, hostname, port>
- Private key ~~stored~~ re-generated on device
- Public key sent to server

Additions to the authentication flow:

- Origin (URI): ***prevents phishing***
- TLS Channel ID (optional): *prevents MitM*



Key Generation

Storing a private key + metadata per service would require a lot of storage

Alternative: store only a master symmetric key

- Generated on-device upon first startup, and never leaves the YubiKey in any form

Registration

- YubiKey generates a random key pair per credential
- YubiKey encrypts the private key + metadata with the master key ➔ *key handle*
- Key handle + public key sent to server

Authentication

- The server presents the key handle to the YubiKey, along with a challenge
- YubiKey decrypts the key handle and reveals the private key (authenticated encryption: ensures integrity, and that the credential is used with the correct AppID)
- YubiKey signs the challenge with the private key to complete the authentication

https://developers.yubico.com/U2F/Protocol_details/Key_generation.html

U2F tokens

Benefits

Easy: just tap the button (no typing)

Works out of the box (no drivers to install)

USB, NFC, Bluetooth communication

No shared secret between client and server

Origin checking ➔ prevents phishing!

Drawbacks

Can be lost ➔ need a fallback (backup codes, 2nd U2F token, authenticator app, …)

Cumbersome: have to pull keychain out and plug token in (or have an always pugged-in token, in which case though it can be stolen along with the device)

Cost ($10–$70)

Advanced Protection Program

https://landing.google.com/advancedprotection/

Google Advanced Protection Program

Overview FAQ

Get started

Google's strongest security helps keep your private information safe.

The Advanced Protection Program safeguards users with high visibility and sensitive information from targeted online attacks. New protections are automatically added to defend against today's wide range of threats.

Learn how to get started

Advanced Protection

https://myaccount.google.com/u/0/advanced-protection/enroll/details?pli=1

Google Account

Advanced Protection

1 Get security keys — 2 Register security keys — 3 Enroll

Get security keys

First, you need 2 security keys, one of them for backup. Your security key will be used in addition to your password to sign in to your account. You can use keys that you already own or buy new ones. Learn more

Ship to: United States

Titan Security Key
From Google

Make sure to get key types that are compatible with your devices.

Google Store

Buy now

Register security keys

Because you use a physical key instead of the six-digit code, security keys strengthen the two-factor authentication process and help prevent your second authentication factor from being intercepted or requested by an attacker.

You're responsible for maintaining access to your security keys. If you lose all of your trusted devices and security keys, you could be locked out of your account permanently.

Learn more about two-factor authentication ›

What's required for Security Keys for Apple ID

- At least two FIDO® Certified* security keys that work with the Apple devices that you use on a regular basis.
- iOS 16.3, iPadOS 16.3, or macOS Ventura 13.2, or later on all of the devices where you're signed in with your Apple ID.
- Two-factor authentication set up for your Apple ID.
- A modern web browser. If you can't use your security key to sign in on the web, update your browser to the latest version or try another browser.

2FA Recap – *What threats does it prevent?*

SMS: useful against two main threats

- Credential stuffing (people tend to reuse passwords across different services)
- Leaked passwords (post-it, hardware keyloggers, cameras, shoulder surfing, …)
- Introduces new security/privacy issues: SIM swapping, SMS account recovery, SMS spam…

Authenticator Apps/Push Auth: much better alternative than SMS

- Protects against the same threats without relying on phone numbers

U2F: **additional protection against phishing**

- Modern phishing toolkits bypass SMS/Authenticator/Push 2FA through MitM
- Humans fall for typosquatting, but U2F's origin check doesn't

None of the above protect against session hijacking and Man-in-the-Browser

- Game over anyway if the host is compromised after the user has successfully logged in

Password Managers

Have become indispensable

- Encourage the use of complex/non-memorable passwords
- Obviate the need for password reuse: unique passwords per site/service

Protection against phishing: *auto-fill won't work for incorrect domains*

- As long as users don't copy/paste passwords out of the password manager (!)

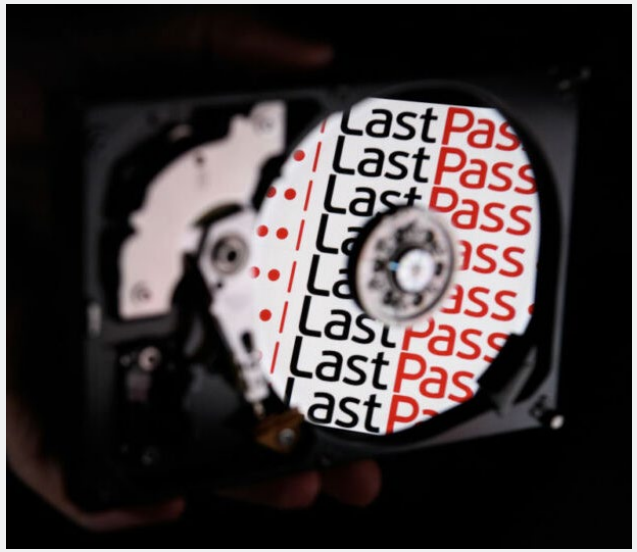
Various options: third-party applications, OS-level, in-browser

Password synchronization across devices

- Can the service provider access all my passwords or not?
- Preferable option: passwords should be encrypted locally with a master password never visible to the cloud service

Single point of failure (!)

LastPass says employee's home computer was hacked and corporate vault taken



Already smarting from a breach that stole customer vaults, LastPass has more bad news.

DAN GOODIN - 2/27/2023, 8:01 PM

284

Already smarting from a breach that put partially encrypted login data into a threat actor's hands, LastPass on Monday said that the same attacker hacked an employee's home computer and obtained a decrypted vault available to only a handful of company developers.

Although an initial intrusion into LastPass ended on August 12, officials with the leading password manager said the threat actor "was actively engaged in a new series of reconnaissance, enumeration, and exfiltration activity" from August 12 to August 26. In the process, the unknown threat actor was able to steal valid credentials from a senior DevOps engineer and access the contents of a LastPass data vault. Among other things, the vault gave access to a shared cloud-storage environment that contained the encryption keys for customer vault backups stored in Amazon S3 buckets.

Another bombshell drops

"This was accomplished by targeting the DevOps engineer's home computer and exploiting a vulnerable

Single Sign-on/Social Login

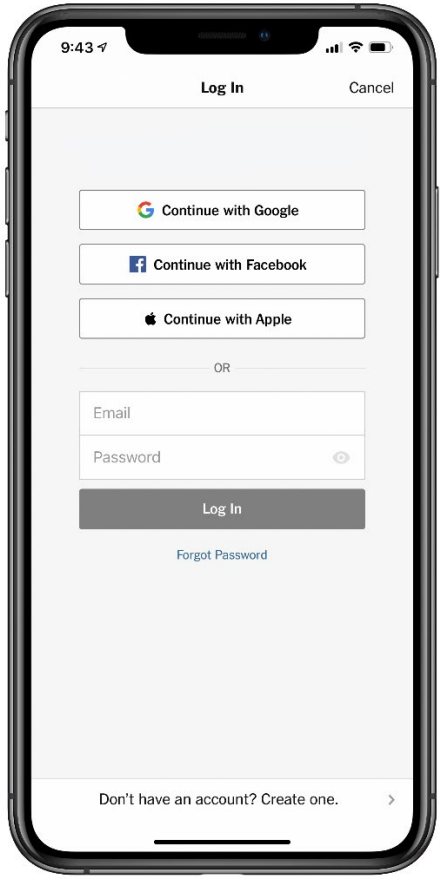
Use a central authentication service for multiple sites

Pros

- Convenience: fewer passwords to remember
- Easier development: outsource user registration/management
- Rich experience through social features

Cons

- Same credentials for multiple sites: single point of failure
- Third-parties gain access to users' profiles
- Provider can track users



WebAuthn

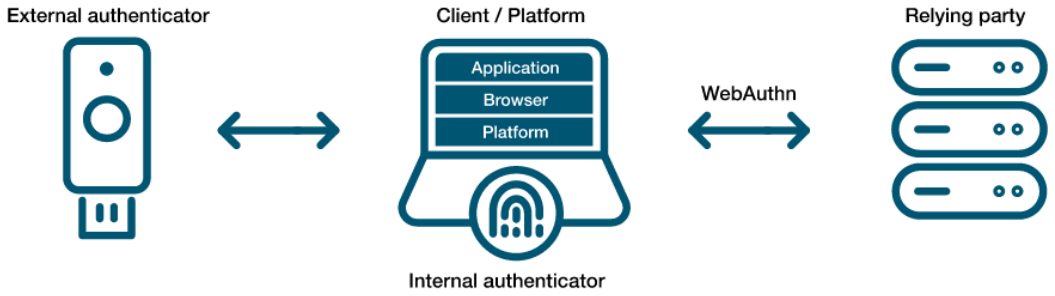
W3C Web Authentication standard (FIDO2): Successor of FIDO U2F

Use cases

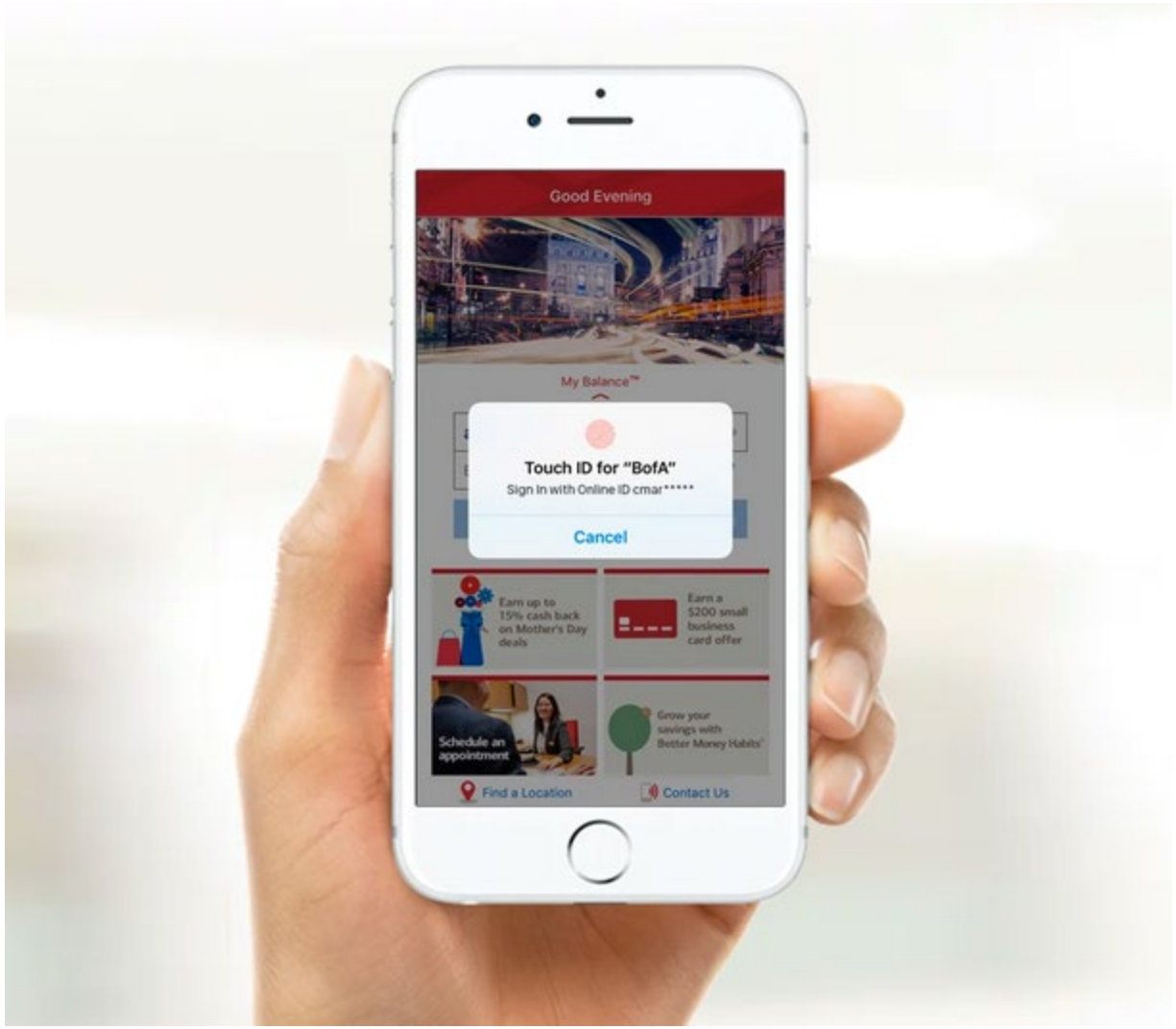
- Low friction and phishing-resistant 2FA (in conjunction with a password)
- Passwordless, biometrics-based *re-authorization*
- 2FA *without* a password (passwordless login)

Authenticators: devices that can generate private/public key pairs and gather consent (simple tap, fingerprint read, …)

- Roaming Authenticators:
  USB/BLE/NFC security keys
- Platform Authentications:
  Built-in fingerprint readers, cameras, …



https://www.yubico.com/authentication-standards/webauthn/


Good Evening
My Balance™
Touch ID for "BofA"
Sign In with Online ID cmar*****
Cancel
Earn up to 15% cash back on Mother's Day deals
Earn a $200 small business card offer
Schedule an appointment
Grow your savings with Better Money Habits®
Find a Location
Contact Us

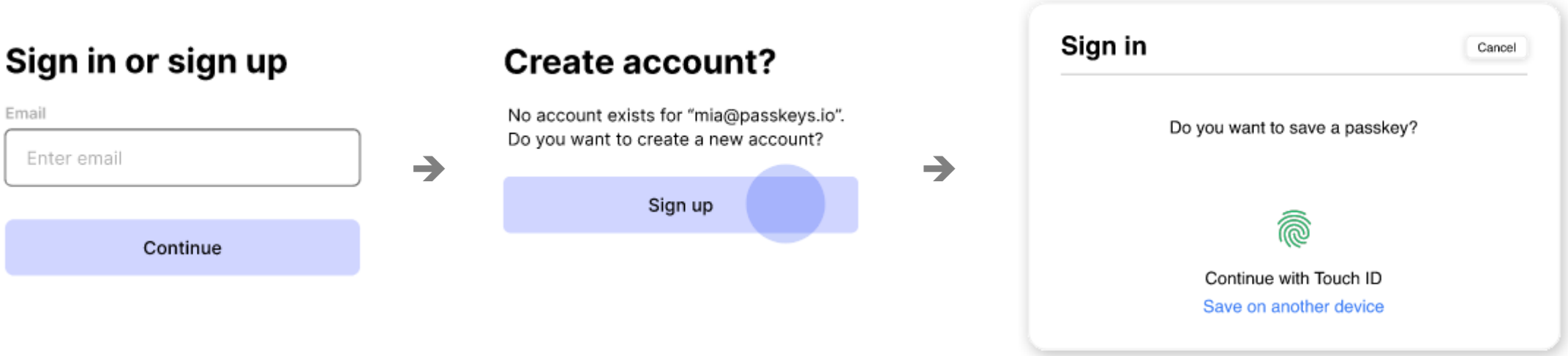
Passkeys

Completely replace passwords with cryptographic key pairs

- Server only keeps a user's public key
- Based on WebAuthn: rely on biometric identification (Face ID, Windows Hello, …)

Key enabler: identity providers (Apple, Google, …) who also sell devices

- The device becomes an authenticator: what if it gets lost? ➔ recovery through vendor
- Users have more than one device ➔ seamless syncing



https://www.passkeys.io/

SAFETY & SECURITY

Passwordless by default: Make the switch to passkeys

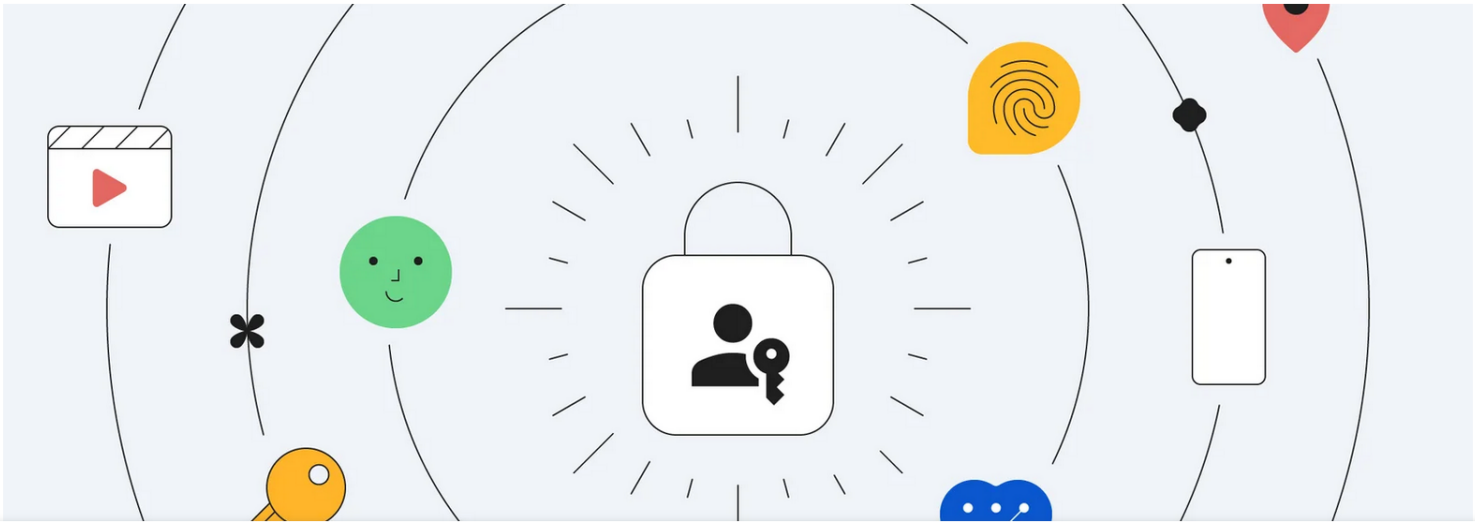
Oct 10, 2023
2 min read

For Cybersecurity Awareness Month we're making it even easier for users to get started with passkeys

S **Sriram Karra**
Senior Product Manager

C **Christiaan Brand**
Group Product Manager

Share



Passkeys.directory

https://passkeys.directory

Passkeys.directory

Passkeys.directory is a community-driven index of websites, apps, and services that offer signing in with passkeys.

Passkeys supported | Vote for passkeys support NEW

Search passkeys.directory | Viewing All listings | Sort by Name

NAME	SUPPORTED	CATEGORY	
Adobe adobe.com	Sign In	Information Technology	Details
Air New Zealand airnewzealand.co.nz	Sign In	Travel & Tourism	Details
Albert Heijn ah.nl	Sign In	eCommerce	Details
Amazon amazon.com	Sign In	eCommerce	Details
Apple apple.com	Sign In	Information Technology	Details
Arcalive arca.live	Sign In MFA	Social	Details
Arpari arpari.com	Sign In	Finance	Details
au	Sign In	Information Technology	Details

Multi-factor vs. Multi-step

Factor: something you know/have/are

Step: user-specific action

Type password, tap fingerprint reader, press security key, look at camera, …

Example: U2F flow with passwords

Type password + tap security key ➔ two factors, two steps ********* +

Example: FIDO2 passwordless flow

Tap biometric security key ➔ two factors, one step

Phone Face ID ➔ two factors, one step

https://medium.com/webauthnworks/its-single-step-not-factor-clarifying-more-fido-terminology-d06d9c31b4f2

Recap: Crypto-based Authentication

Rely on a cryptographic key to prove a user's identity

User performs a requested cryptographic operation on a value (challenge) that the verifier supplies

- Usually based on knowledge of a key (shared secret key or private key)
- Can use symmetric (e.g., Kerberos) or public key (e.g., U2F, passkeys) schemes

How can we trust a key? Why is it authentic?

- Need to establish a level of trust

Different approaches: **TOFU**, **PKI**, **Web of Trust**

Trust on First Use (aka Key Continuity)

Use case: SSH

- Performs *mutual authentication*

Server *always* authenticates the client

- password, key pair, …

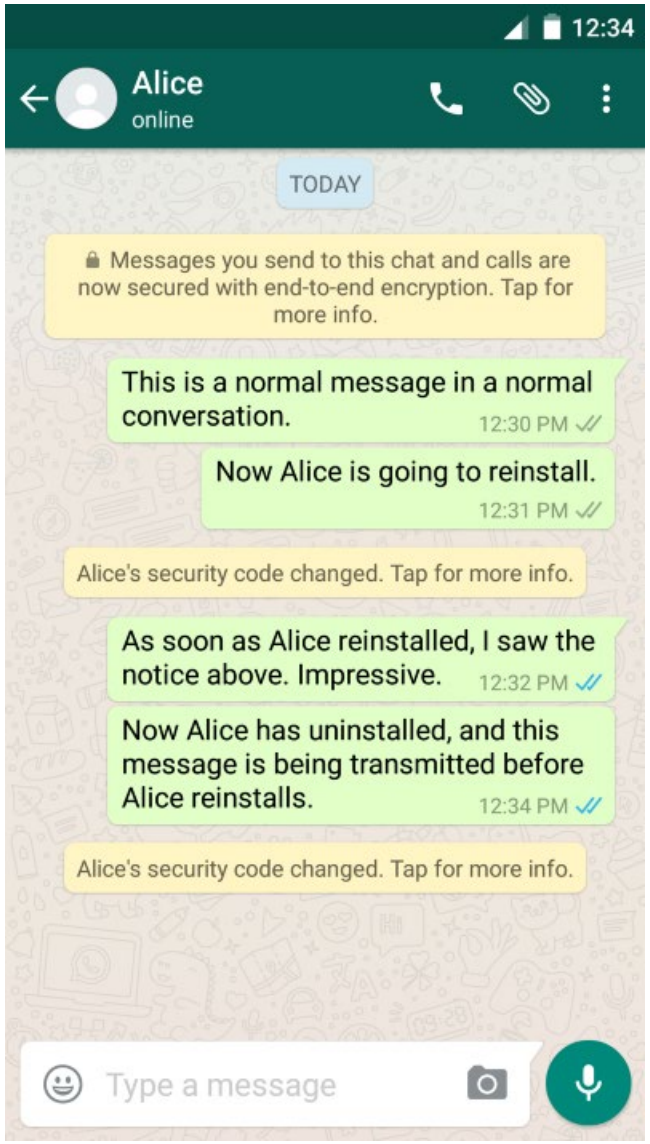
Client *almost* always authenticates the server – *except the first time!*

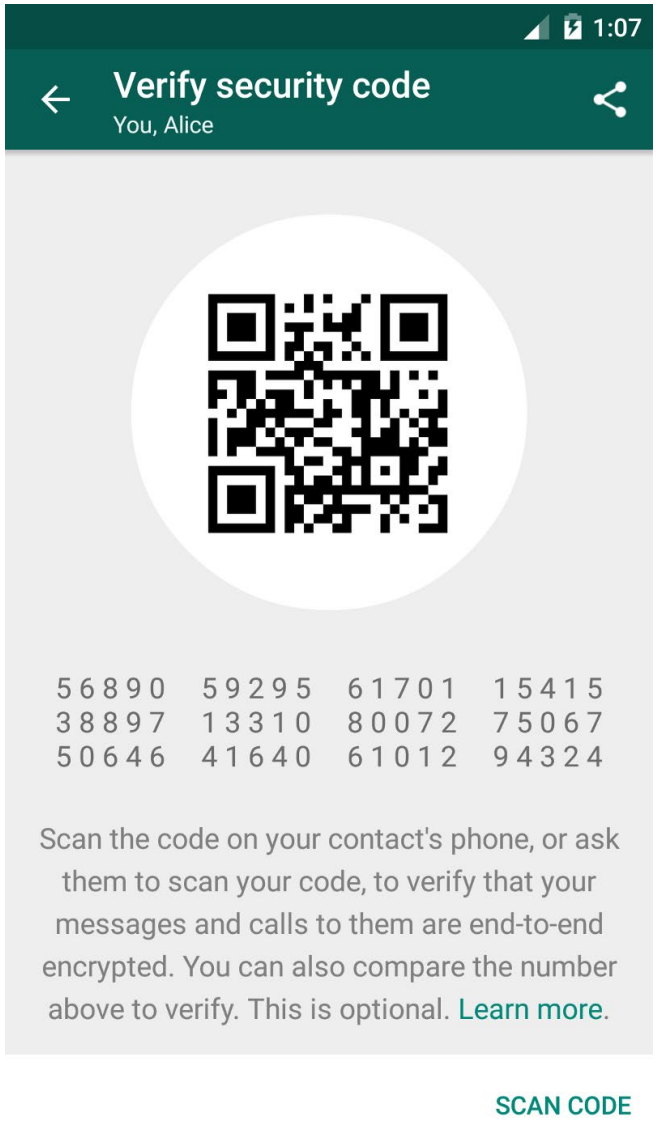
- First connection: server presents its public key
- No other option for the user but to accept it: MitM opportunity
- Subsequent connections: client remembers server's key, and triggers an alert on key mismatch

Pragmatic solution, but shifts the burden to users

- Users must determine the validity of the presented key
- Accepting a key change without verifying the new key offers no protection against MitM (unfortunately, that's what most users do)

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
df:c8:52:aa:cd:e3:da:8c:ec:50:46:db:4d:21:d9:c7.
Please contact your system administrator.
Add correct host key in /root/.ssh/known_hosts to get rid of this message.
Offending key in /root/.ssh/known_hosts:1
RSA host key for 192.168.2.5 has changed and you have requested strict checking.
Host key verification failed.
```





Certificates

How can we distribute "trusted" public keys?

- Public directory ➔ risk of forgery and tampering, scalability issues
- More practical solution: "certified" public keys

A certificate is a digitally signed message that contains an identity and a public key

- Makes an association between a user/entity and a private key
- Valid until a certain period
- Most common format: X.509

Why trust a certificate?

- Because it is signed by an "authority"
- Requiring a signature by a third party prevents straightforward tampering

Public Key Infrastructures (PKI)

Facilitate the authentication and distribution of public keys with the respective identities of entities

- People, organizations, devices, applications, …
- Set of roles, policies, hardware, software, and procedures to create, mange, distribute, use, store, and revoke digital certificates and manage public key encryption

An issuer signs certificates for subjects: *"Trust anchor"*

Methods of certification

- **Certificate authorities** (hierarchical structure – root of trust)
- **Web of trust** (decentralized, peer-to-peer structure)

Certificate Authorities

Trusted third-parties responsible for certifying public keys

- Most CAs are tree-structured

A public key for any website in the world will be accepted without a browser warning if it has been certified by a trusted CA

Why should we trust an authority?

- How do we know the public key of the Certificate Authority?

CA's public key (trust anchor) must somehow be provided out of band

- Operating systems and browsers are pre-configured with tens/hundreds of trusted root certificates (more on this in the TLS lecture)

Single point of failure: CAs can be compromised!

DigiNotar Files for Bankr

Secure | https://www.wired.com/2011/09/diginotar-bankruptcy/

WIRED

DigiNotar Files for Bankruptcy in Wake of Devastating Hack

SUBSCRIBE

SHARE

1

KIM ZETTER SECURITY 09.20.11 03:05 PM

DIGINOTAR FILES FOR BANKRUPTCY IN WAKE OF DEVASTATING HACK

A Dutch certificate authority that suffered a major hack attack this summer has been unable to recover from the blow and filed for bankruptcy this week.

MOST POPULAR

TRANSPORTATION
General Motors Announces an All-Electric Future
ALEX DAVIES

SECURITY
This "Ghost Gun" Machine Now Makes Untraceable Metal Handguns
ANDY GREENBERG

SECURITY
How the Las Vegas Shooter Could Have Gotten an Automatic Rifle
ANDY GREENBERG

MORE STORIES

Web of Trust (mainly used in PGP for encrypted email – future lecture)

Entirely decentralized authentication

- No need to buy certs from CAs: users create their own certificates

Users validate other users' certificates, forming a "web of trust"

- No trusted authorities: trust is established through friends *(yay! key signing parties!)*

Main problems

- Privacy issues: social graph metadata
- Bootstrapping: new users are not readily trusted by others
- When opinions vary, "stronger set" wins: impersonation through collusion/compromised keys
- Scalability: challenging to create a WoT for the whole world

WoT: Finding Public Keys

Public PGP key servers

- pgp.mit.edu
- keyserver.pgp.com

Cache certificates from received emails

Integration with user management systems (LDAP, IAM/IDP)

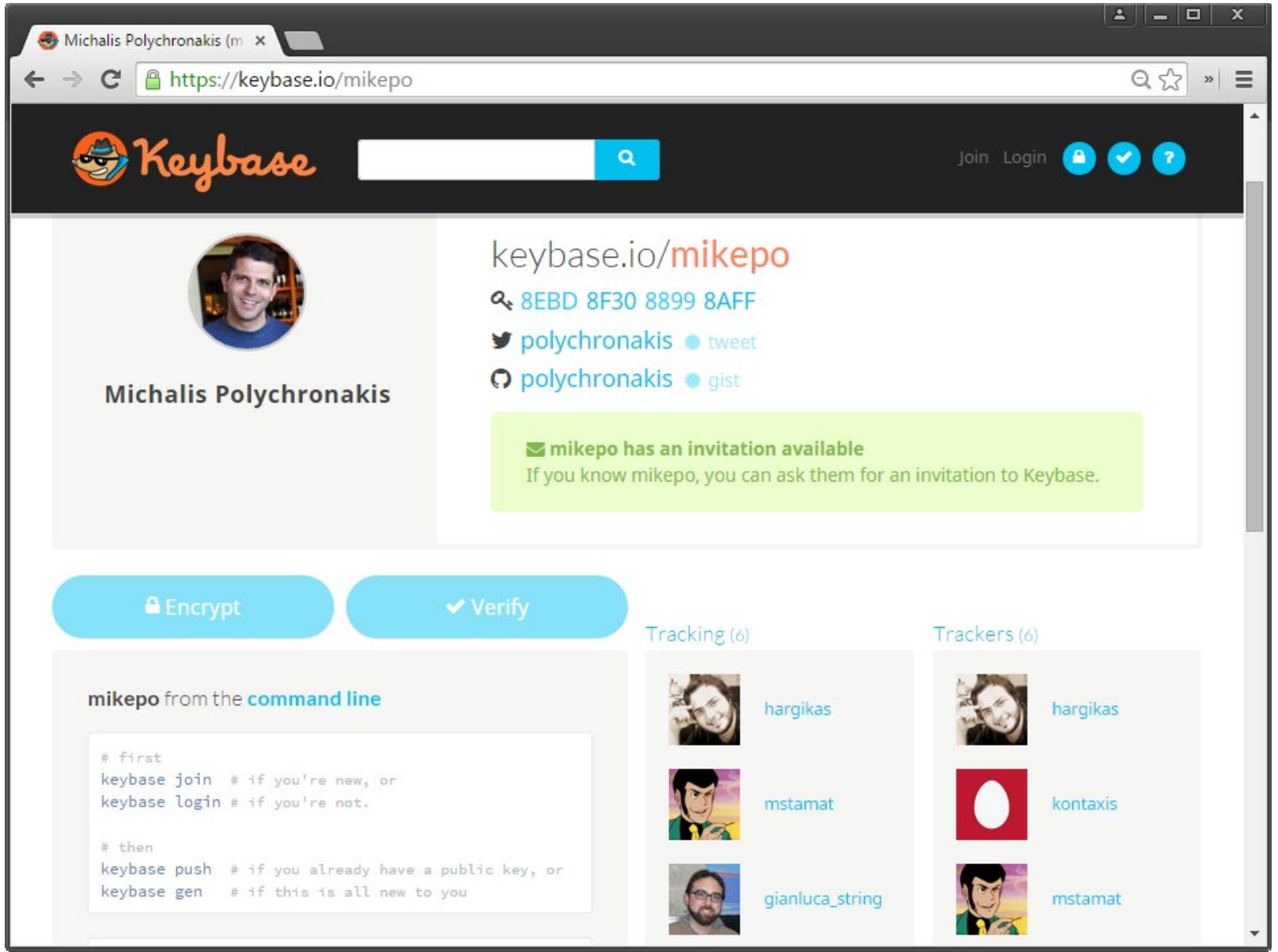
Ad-hoc approaches

- List public key on home page
- Print on business card
- Exchange through another medium on a case-by-case basis

Association with social profiles/identities

- keybase.io

Online Social "Tracking"



Keybase.io

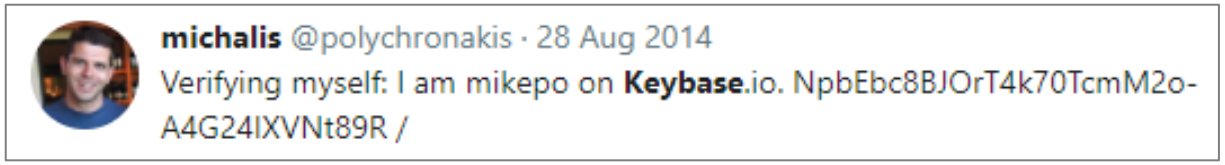
In essence, a directory associating public keys with names

Identity established through *public signatures*

- **Identity proofs**: *"I am Joe on Keybase and MrJoe on Twitter"*
- **Follower statements**: *"I am Joe on Keybase and I just looked at Chris's identity"*
- **Key ownership**: *"I am Joe on Keybase and here's my public key"*
- **Revocations**: *"I take back what I said earlier"*

Keybase identity = sum of public identities

Twitter, Facebook, Github, Reddit,
domain ownership, …



An attacker has to compromise all connected identities

The more connected identities, the harder to impersonate a user

Best Practices

Use long passphrases instead of passwords

Never reuse the same password on different services

Use two-factor authentication

Avoid SMS if possible! Use an authenticator app or even better **U2F (or passkeys)**

Remove phone number from account after authenticator/U2F setup

Store your backup codes/backup key in a safe location

Use a password manager

Pick non-memorable passwords and avoid copy/pasting them

Password auto-fill helps against phishing! (auto-fill will fail if the domain is wrong)

Use SSH keys instead of passwords