

CSE 484 / CSE M 584: Computer Security and Privacy

# Authentication

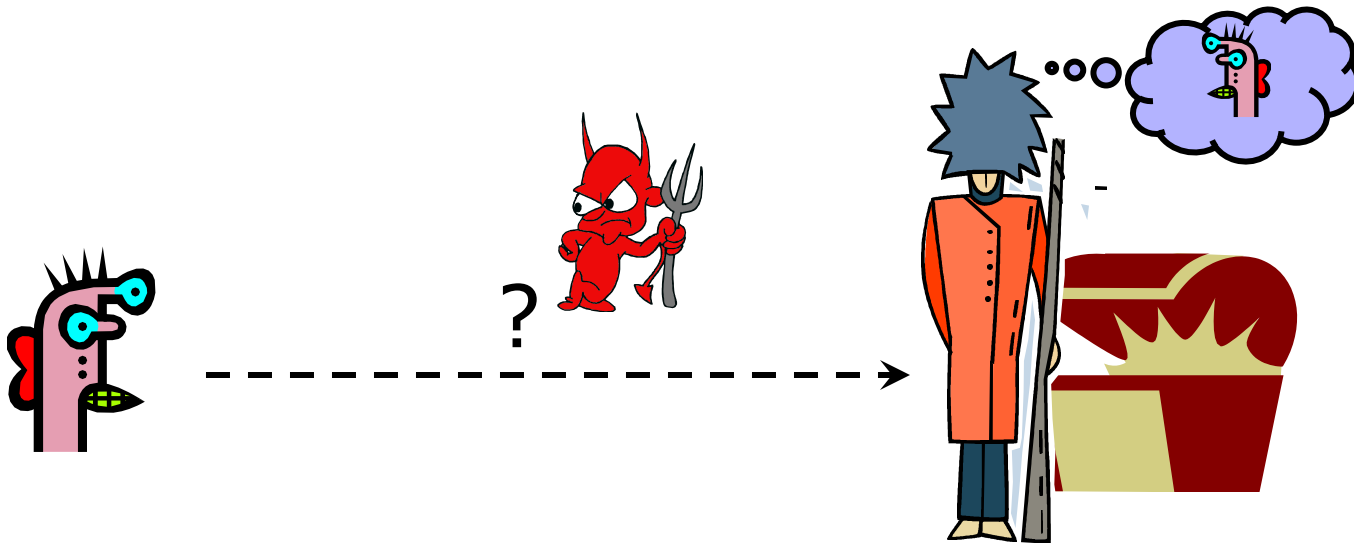
Autumn 2020

Franziska (Franzi) Roesner

[franzi@cs.washington.edu](mailto:franzi@cs.washington.edu)

Thanks to Dan Boneh, Dieter Gollmann, Dan Halperin, Yoshi Kohno, John Manferdelli, John Mitchell, Vitaly Shmatikov, Bennet Yee, and many others for sample slides and materials ...

# Basic Problem



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

Any system with access control must solve this problem.

# Many Ways to Prove Who You Are

- What you know
  - Passwords
  - Answers to questions that only you know
- Where you are
  - IP address, geolocation
- What you are
  - Biometrics
- What you have
  - Secure tokens, mobile devices

# Passwords and Computer Security

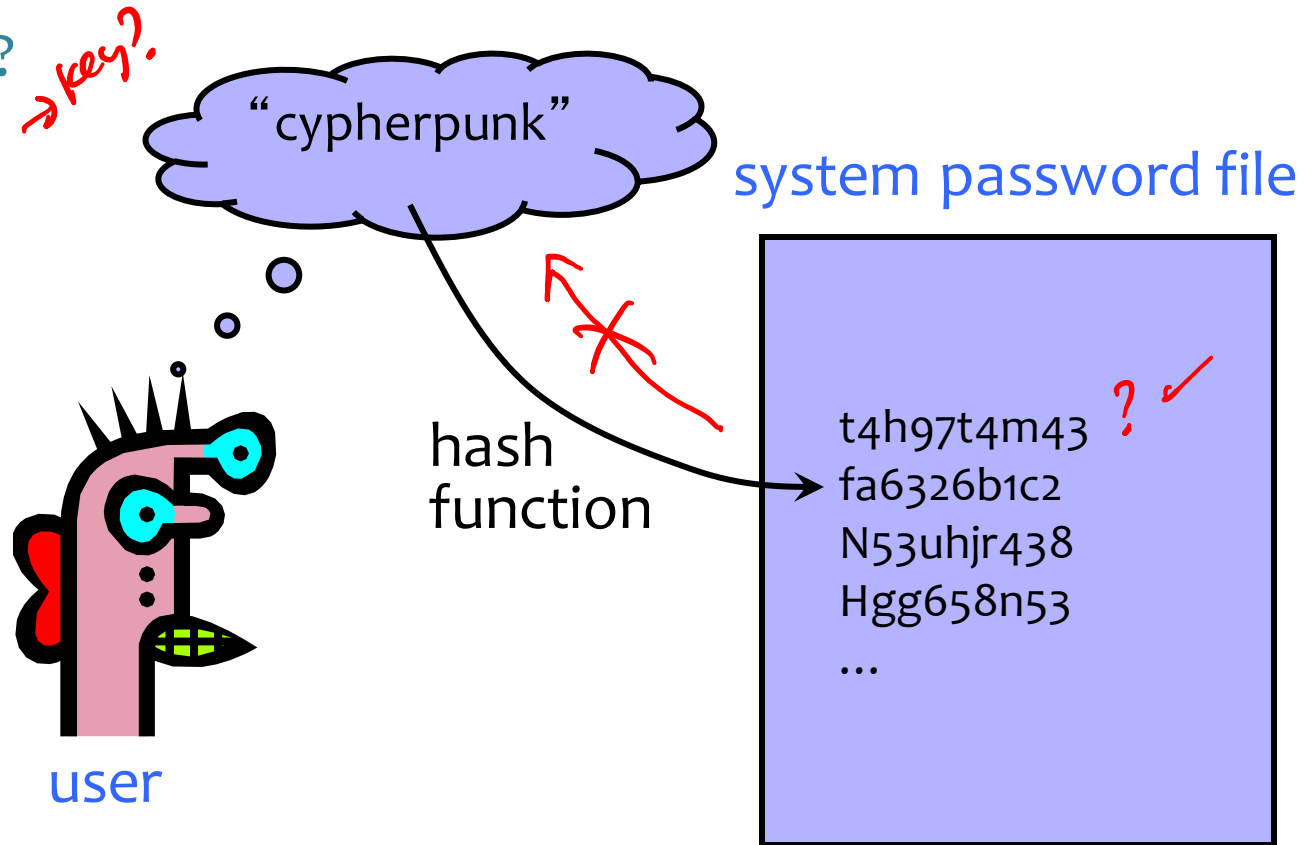
- In 2012, **76%** of network intrusions exploited weak or **stolen credentials** (username/password)
  - Source: Verizon Data Breach Investigations Report
- In Mitnick's "Art of Intrusion" **8 out of 9 exploits involve password stealing and/or cracking**
- First step after any successful intrusion: install sniffer or keylogger to steal more passwords
- Second step: run cracking tools on password files
  - Cracking needed because modern systems usually do not store passwords in the clear

HW3

# UNIX-Style Passwords

- How should we store passwords on a server?

- ~~X~~ In cleartext?
- ~~X~~ Encrypted?
- ✓ Hashed?



# Password Hashing

- Instead of user password, store  $H(\text{password})$
- When user enters password, compute its hash and compare with entry in password file
  - System does not store actual passwords!
  - System itself can't easily go from hash to password
    - Which would be possible if the passwords were encrypted
- Hash function  $H$  must have some properties
  - One-way: given  $H(\text{password})$ , hard to find password
    - No known algorithm better than trial and error
  - “Slow” to compute

# UNIX Password System

- Approach: Hash passwords
- Problem: passwords are not truly random
  - With 52 upper- and lower-case letters, 10 digits and 32 punctuation symbols, there are  $94^8$  H
  - 6 quadrillion possible 8-character passwords ( $\sim 2^{52}$ )
  - **BUT:** Humans like to use dictionary words, human and pet names H 1 million common passwords

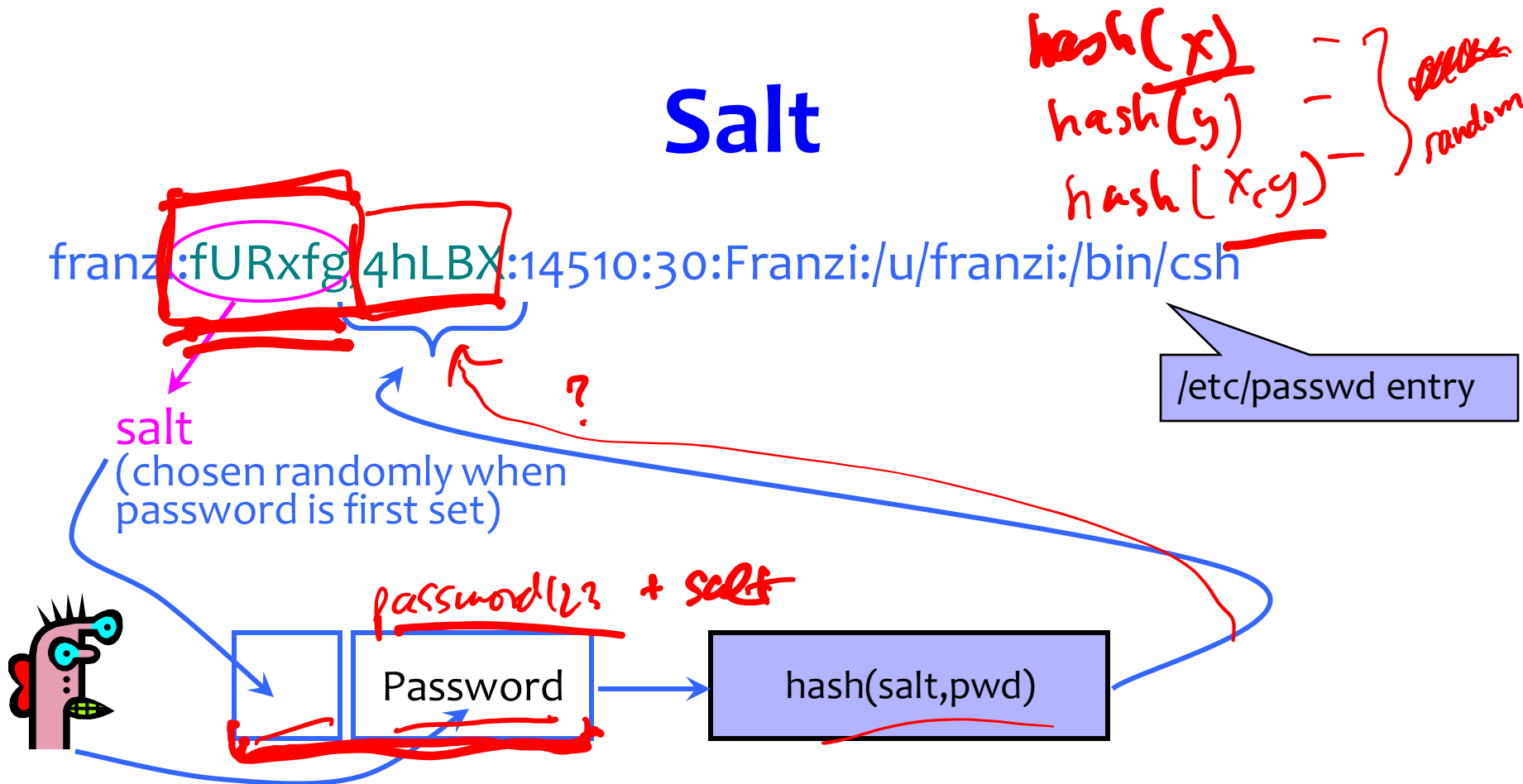
~~GA~~  
hash(pwd) → X  
hash(pwd') → X'

# Dictionary Attack

- Dictionary attack is possible because many passwords come from a small dictionary
  - Attacker can pre-compute  $H(\text{word})$  for every word in the dictionary – this only needs to be done once!
    - This is an offline attack
    - Once password file is obtained, cracking is instantaneous
  - Sophisticated password guessing tools are available
    - Take into account freq. of letters, password patterns, etc.



# Salt



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

# Advantages of Salting

- Without salt, attacker can pre-compute hashes of all dictionary words once for all password entries
  - Same hash function on all UNIX machines
  - Identical passwords hash to identical values; one table of hash values can be used for all password files
- With salt, attacker must compute hashes of all dictionary words once for each password entry
  - With 12-bit random salt, same password can hash to  $2^{12}$  different hash values
  - Attacker must try all dictionary words **for each salt value** in the password file
- Pepper: Secret salt (not stored in password file)

# Other Password Security Risks

- Keystroke loggers
  - Hardware
  - Software (spyware)
- Shoulder surfing
- Same password at multiple sites
- Broken implementations
  - Recall TENEX timing attack
- Social engineering


*phishing  
face to face*



# Default Passwords

- Examples from Mitnick's "Art of Intrusion"
  - U.S. District Courthouse server: "public" / "public"
  - NY Times employee database: pwd = last 4 SSN digits
- Mirai IoT botnet
  - Weak and default passwords on routers and other devices

# Weak Passwords

- RockYou hack
  - “Social gaming” company
  - Database with 32 million user passwords from partner social networks
  - Passwords stored in the clear
  - December 2009: entire database hacked using an **SQL injection attack** and posted on the Internet
  - One of many such examples!

# Weak Passwords

- RockYou hack



## Password Popularity – Top 20

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

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

# Password Policies

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

# Password Usability

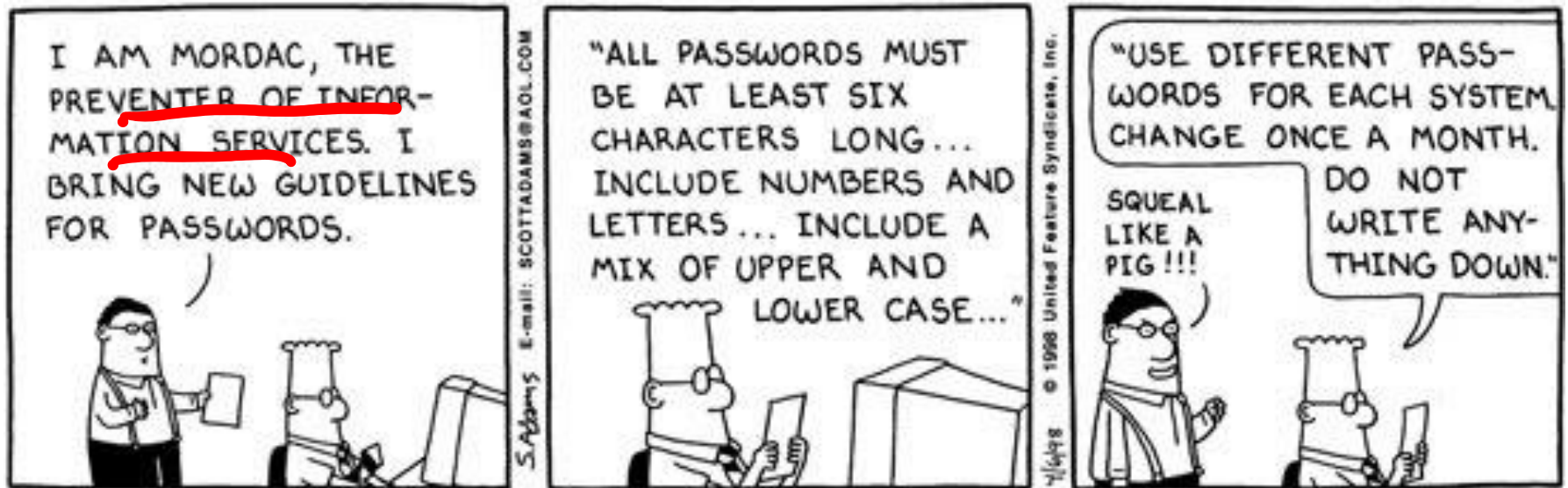







Image from [http://www.interactivetools.com/staff/dave/damons\\_office/](http://www.interactivetools.com/staff/dave/damons_office/)

# Password Policies

- Old recommendation:
  - 7 or 8 characters, at least 3 out of {digits, upper-case, lower-case, non-alphanumeric}, no dictionary words, change every 4 months, password may not be similar to previous 12 passwords...
- **But ...** results in frustrated users and less security
  - Burdens of devising, learning, forgetting passwords
  - Users construct passwords insecurely, write them down
    - Can't use their favorite password construction techniques (small changes to old passwords, etc.)
  - Heavy password re-use across systems
  - (Password managers can help)


# New NIST Guidelines 😊

- Remove requirement to periodically change passwords
  - Screen for commonly used passwords
  - Allow copy-paste into password fields
  - Allow but don't require arbitrary special characters
  - Etc.
- 

<https://pages.nist.gov/800-63-3/sp800-63b.html>

# Recovering Passwords

## Palin E-Mail Hacker Says It Was Easy

By Kim Zetter  September 18, 2008 | 10:05 am | Categories: [Elections](#), [Hacks and Cracks](#)

A p  
obt  
priv  
sup  
rev  
too  
Re|

after the password recovery was reenabled, it took seriously 45 mins on wikipedia and google to find the info, Birthday? 15 seconds on wikipedia, zip code? well she had always been from wasilla, and it only has 2 zip codes (thanks online postal service!)

the second was somewhat harder, the question was "where did you meet your spouse?" did some research, and apparently she had eloped with mister palin after college, if youll look on some of the screenshits that I took and other fellow anon have so graciously put on photobucket you will see the google search for "palin eloped" or some such in one of the tabs.

I found out later though more research that they met at high school, so I did variations of that, high, high school, eventually hit on "Wasilla high" I promptly changed the password to popcorn and took a cold shower...

# Wired Cover Story (Dec 2012)



## Also in this issue

Kill the Password: Why a String of Characters Can't Protect Us Anymore


*“This summer, hackers destroyed my entire digital life in the span of an hour. My Apple, Twitter, and Gmail passwords were all robust—seven, 10, and 19 characters, respectively, all alphanumeric, some with symbols thrown in as well—but the three accounts were linked, so once the hackers had conned their way into one, they had them all. They really just wanted my Twitter handle: @mat.”*


# Improving(?) Passwords

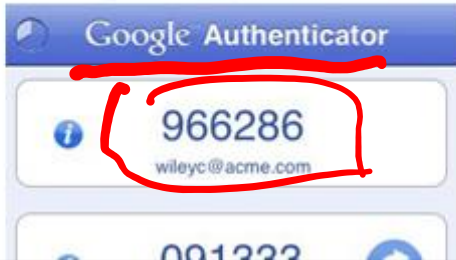
- Add biometrics
  - For example, keystroke dynamics or voiceprint
- Graphical passwords
  - Goal: easier to remember? no need to write down?
- Password managers
  - Examples: LastPass, KeePass, built into browsers
  - Can have security vulnerabilities...
- Two-factor authentication
  - Leverage phone (or other device) for authentication




# Multi-Factor Authentication

1.  Sign in with your Google Account  
Email: hikingfan@gmail.com  
ex: pat@example.com  
Password: .....  
☒ Stay signed in  
  
[Can't access your account?](#)

2.  Google accounts  
**Enter verification code**  
To verify your identity on this computer, enter the verification code generated by your mobile application.  
**Enter code:** 466453   
☒ Remember verification for this computer for 30 days.  
[Other ways to get a verification code »](#)

 Google Authenticator  
966286  
wileyc@acme.com

 Turn on Login Approvals  
What is Login Approvals?  
Login Approvals is a security feature that requires you to enter a code that we text to your phone when you log in from an unrecognized computer. You can enable this feature in a few simple steps.  
If you ever lose access to your phone, you can always return to a previously-recognized computer to regain access to your account.  
Note: You'll need to have your mobile phone with you to complete this process.

# FIDO + Hardware Two Factors





# Questions:

**Do you use 2-factor auth?**

**Do you use a password manager?**

**Why or why not?**