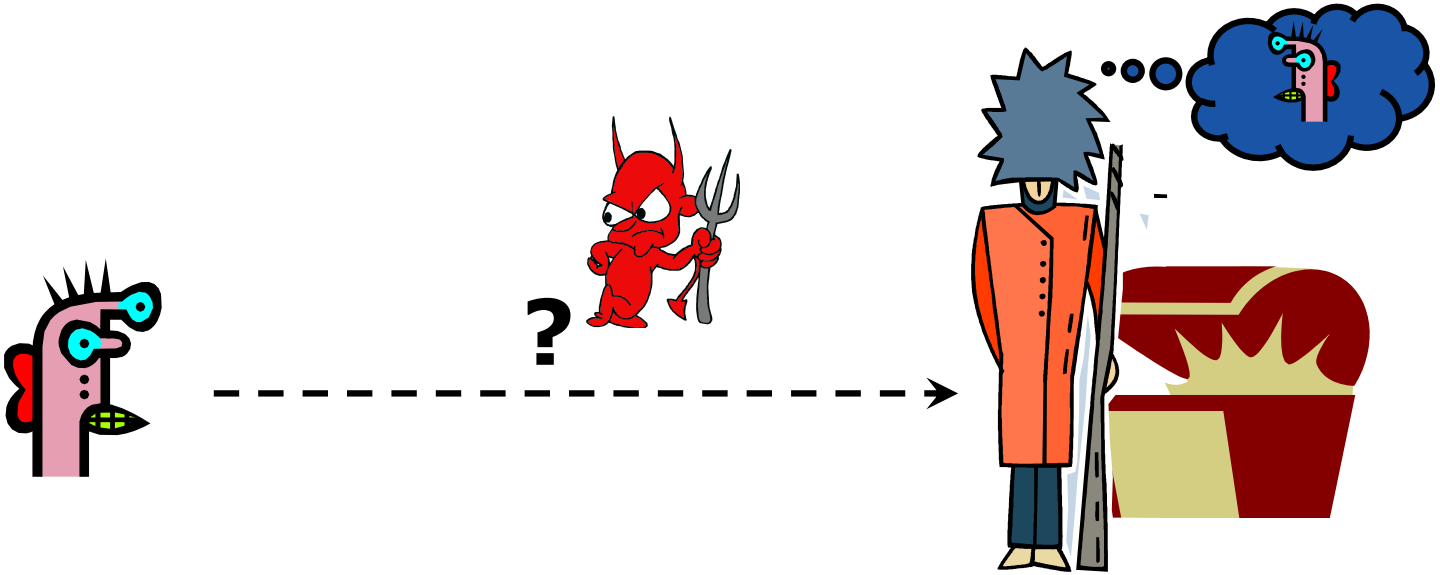
Authentication

IV1013

Peter Sjödin

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
  - Secret key
- What you have
  - Secure tokens
- What you are
  - Biometrics

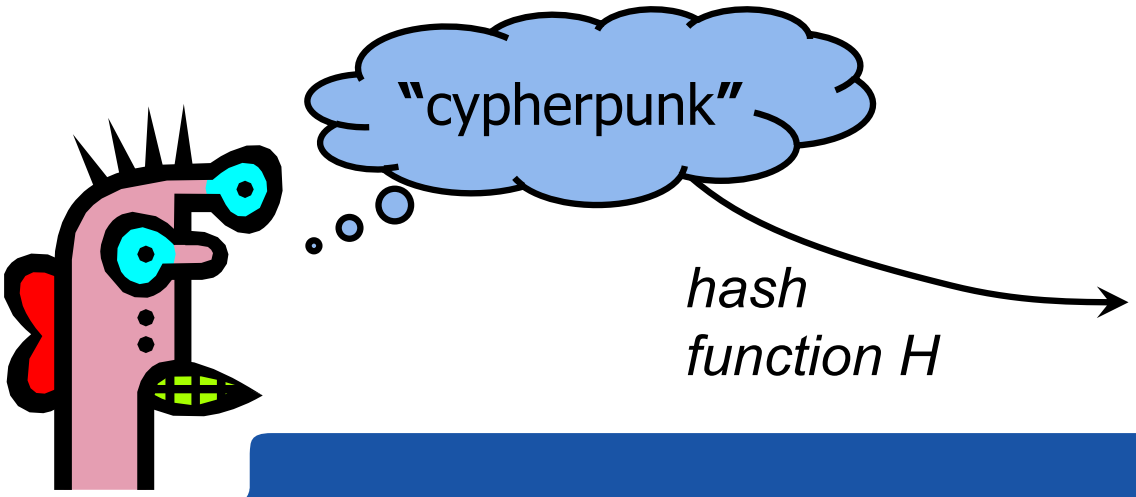
What You Know – Password-Based Authentication

- User has a secret password
- System checks it to authenticate the user

- Vulnerable to eavesdropping when password is communicated from user to system
- How is the password stored?
- How does the system check the password?
- How easy is it to guess the password?
  - Easy-to-remember passwords tend to be easy to guess
- Password file is difficult to keep secret

UNIX-Style Passwords

- Passwords stored in a file
  - Storing passwords in cleartext not a good idea
  - Why isn't read protection enough?
- Take advantage of properties of cryptographic hash
  - One-way and collision resistance
  - Store *H(password)*, where *H* is cryptographic hash



system password file

```
Alice: t4h97t4m43
Bob:   fa6326b1c2
…
```

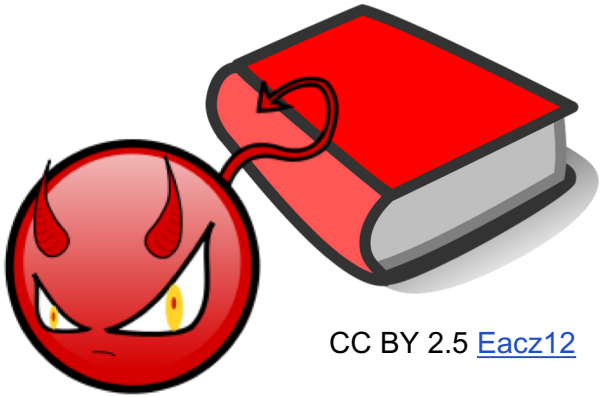
UNIX Password File

- /etc/passwd
- Stores password hash together with user account information
- Records separated by colon ':'
  - `username:hash:userid:groupid:full name:directory:shell`
- Password validation:
  - User inputs user name and password
  - System hashes password, and compares result to user's entry in password file

Hash

```
arabella:N83cm8.ujMPTk:501:501:Arabella Rittenhouse:/home/arabella:/bin/zsh
rufus:LPX5bFcaTmBPQ:502:502:Rufus T. Firefly:/home/rufus:/bin/bash
ronald:diZ9WSRZYh43M:503:503:Ronald Kornblow:/home/ronald:/bin/bash
suzanna:ZNF2bLyPtwhGg:504:504:Suzanna Dukesbury:/home/suzanna:/bin/csh
frank:efXFcx3vJ9mtY:505:505:Frank Wagstaff:/home/frank:/bin/bash
```

Password Guessing Attacks



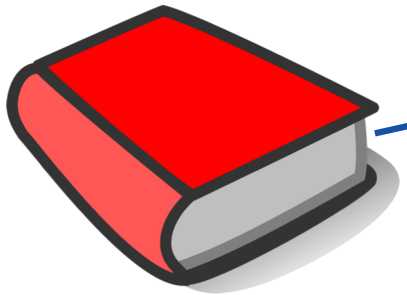
CC BY 2.5 Eacz12

- Try a large number of passwords
  - Taken from a dictionary, for instance
- Permutate dictionary words to generate guesses
- For each guess:
  - Compute the hash
  - Compare it to password file entry
- Offline attack
  - Attacker gets access to password hash (steals password file, for instance)
  - Does not need to communicate with target

- Cain & Abel (Windows password hashes), John the Ripper (Windows, Linux, Mac OS X password hashes), Crack, …

Pre-Computed Dictionary Attack

- Pre-compute and store hashes for likely passwords
  - Taken from a dictionary, for instance



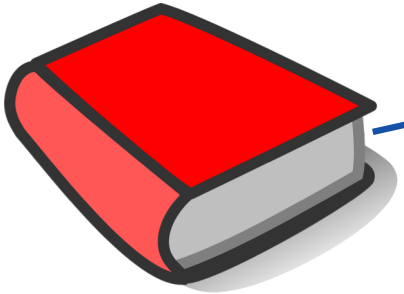
Hash

Hash	*Password*
YXJhYmVsbGE6T	football
aW4vYmFzaApyb	sunshine
ebi9iYXNoCnN1	monkey
LPX5bFcaTmBPQ	swordfish
iaW4vYmFzaApy	master
3N1emFubmE6L2	princess

Table with pre-computed hashes for words in dictionary.

Could be very large – but disk space is cheap.

Offline Dictionary Attack



Hash

Hash	Password
YXJhYmVsbGE6T	football
aW4vYmFzaApyb	sunshine
ebi9iYXNoCnN1	monkey
LPX5bFcaTmBPQ	swordfish
iaW4vYmFzaApy	master
3N1emFubmE6L2	princess

```
arabella:N83cm8.ujMPTk:501:501:Arabella Rittenhouse:/home/arabella:/bin/zsh
rufus:LPX5bFcaTmBPQ:502:502:Rufus T. Firefly:/home/rufus:/bin/bash
ronald:diZ9WSRZYh43M:503:503:Ronald Kornblow:/home/ronald:/bin/bash
suzanna:ZNF2bLyPtwhGg:504:504:Suzanna Dukesbury:/home/suzanna:/bin/csh
frank:efXFcx3vJ9mtY:505:505:Frank Wagstaff:/home/frank:/bin/bash
```

- Get password file from victim host
- Compare entries in password file to table of pre-computed hashes
- Offline attack – attacker does not communicate with victim host
- Can process large number of hashes in short period of time

Password Salt

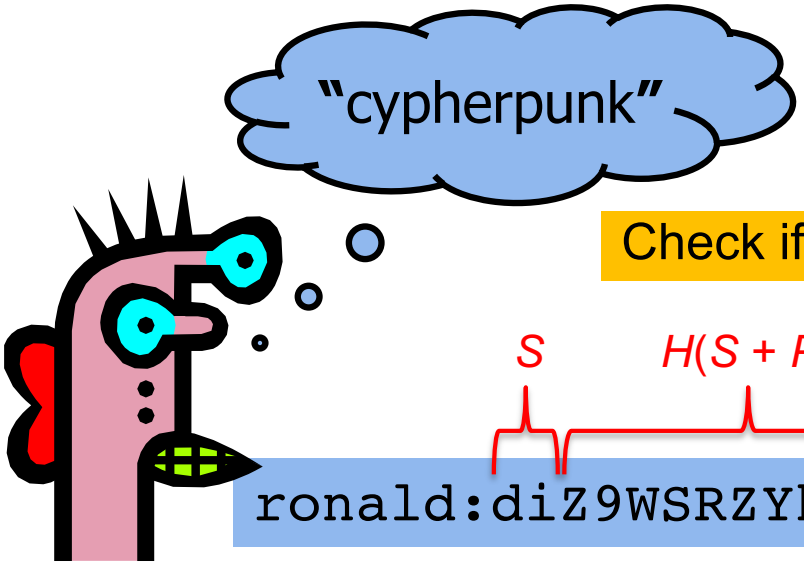
- Introduce randomness in password hashing
  - Same password will be hashed differently, depending on salt
- Generate a salt, as a random value S, for each password entry
- Hash password P together with salt: $H(S + P)$
- Store S and $H(S + P)$ in password file
  - ASCII-encoded

S $H(S + P)$

```
ronald:diZ9WSRZYh43M:503:503:...
```

Verification of Salted Password

- Compute hash of salt and user password
- Check if results matches password file



Check if *H*("di" + "cypherpunk") equals "Z9WSRZYh43M"

S *H*(*S* + *P*)

```
ronald:diZ9WSRZYh43M:503:503:...
```

Advantages of Salting (1)

- Without salt, attacker can pre-compute hashes of all dictionary words once for all password entries
  - Same hash function on all hosts
  - 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

Advantages of Salting (2)

- Duplicate passwords are not visible in password file
  - Two users with same password will have different hashes
- Salting makes it more difficult to detect if a user has same password on two or more systems

Shadow Password File

- Problem: we want to keep password file secret
  - But /etc/passwd contains various information about user accounts
    - Information needed for other purposes than authentication
  - For example, home directory is stored in /etc/passwd
  - Therefore, must be world-readable
- Solution: store hashed passwords in "/etc/shadow" file, only readable by system administrator (root)

/etc/passwd entry:

wagstaff:x:1002:1002:Quincy A. Wagstaff:/home/wagstaff:/bin/bash

Hashed password is *not* stored here

It is here!

/etc/shadow entry:

wagstaff:6WbvdRYUE$F7u6rrWKgidBz/W709mOyaKhM6FBYy/BXl0/cRtcQEcLNdmswbd26uMq43nIOShOK.jp8xlmlPRI3Nl8R6/8Y.:16026:0:99999:7:::

Passwords in Modern UNIX/Linux

- Longer passwords – unlimited, in effect
- More secure hash algorithms
  - MD5, Blowfish, SHA-256, SHA-512
- Longer salt – up to 16 characters
- Password stretching
  - By default 5000 iterations for SHA-256 and SHA-512
- /etc/shadow – Hash field with "$" delimiter

```
wagstaff:$6$WbvdRYUE$F7u6rrWKgidBz/W709mOyaKhM6FBYy/BXl0/cRt
cQEcLNdmswbd26uMq43nIOShOK.jp8xlmlPRI3Nl8R6/8Y.:16026:0:99999:7:::
```

Hash algorithm:

"6" means SHA-512

Salt

Hash value

Timing: expiration, date of last change, …

How are Passwords Selected

- Study performed by CentralNic UK, 1200 users
- family related: 47%
  - Children, pets, ...
- fans/cartoons: 32%
  - David Beckham, Homer Simpson, ...
- fantastists: 11%
  - goddess, hero, ...
- cryptics: 10%
  - Carefully selected passwords
- Vulnerability to dictionary attacks if one single user has an easy guessable password!

How Passwords Are Selected

- SplashData's "Worst Passwords of 2015"
  - Based on 2 million leaked passwords
- https://www.teamsid.com/worst-passwords-2015/

25 most frequent passwords

123456	111111
password	1qaz2wsx
12345678	dragon
qwerty	master
12345	monkey
123456789	letmein
football	login
1234	princess
1234567	qwertyuiop
baseball	solo
welcome	passw0rd
1234567890	starwars
abc123	

AshleyMadison Hack 2015

- "Ashley Madison account passwords are even dumber than you'd expect"
  - mashable.com, Sept. 8, 2015
    - http://mashable.com/2015/09/08/ashley-madison-dumb-passwords
- 36 million passwords stolen
- 1 million checked against "500 worst passwords"
  - www.whatsmypass.com
  - 17,000 cracked

1	123456	11	sparky
2	password	12	helpme
3	12345	13	bl*wj*b
4	12345678	14	nicole
5	qwerty	15	justin
6	p*ssy	16	camaro
7	secret	17	johson
8	dragon	18	yamaha
9	welcome	19	midnight
10	ginger	20	chris

Non-Dictionary Passwords

- Combination of small and capital letters, digits, special symbols, etc
  - Ht!%Lkr9(A
  - Usually qualities of a “good” password
  - Enforced by some password systems

- More difficult to remember
  - Write down
  - Use same password on many systems
- Vulnerable to “shoulder-surfing” attacks
  - Slower to type

Passphrases

- Consider arguments for using passphrases instead of passwords
- Real sentences that we can remember
  - Easier to remember than non-dictionary passwords
  - Harder to break
  - Strength increases with length
- See The Usability of Passwords by Thomas Baekdal, August 11, 2007:
  - “J4fS>2” takes 219 years to crack with brute force
  - “this is fun” takes 2537 years
    - http://www.baekdal.com/insights/password-security-usability

Xcd comic strip on Password Strength: http://xkcd.com/936/

Password Cracking Scenarios

Attacker gets access to the password database, and runs a password-guessing program, such as John the Ripper.

Attacker attempts to login to a certain user account by guessing its password.

How can we increase protection against these attacks?

Attacker gets access to the password file, and compares it with a database of password hashes.

Attacker tries popular passwords against a large number of accounts.

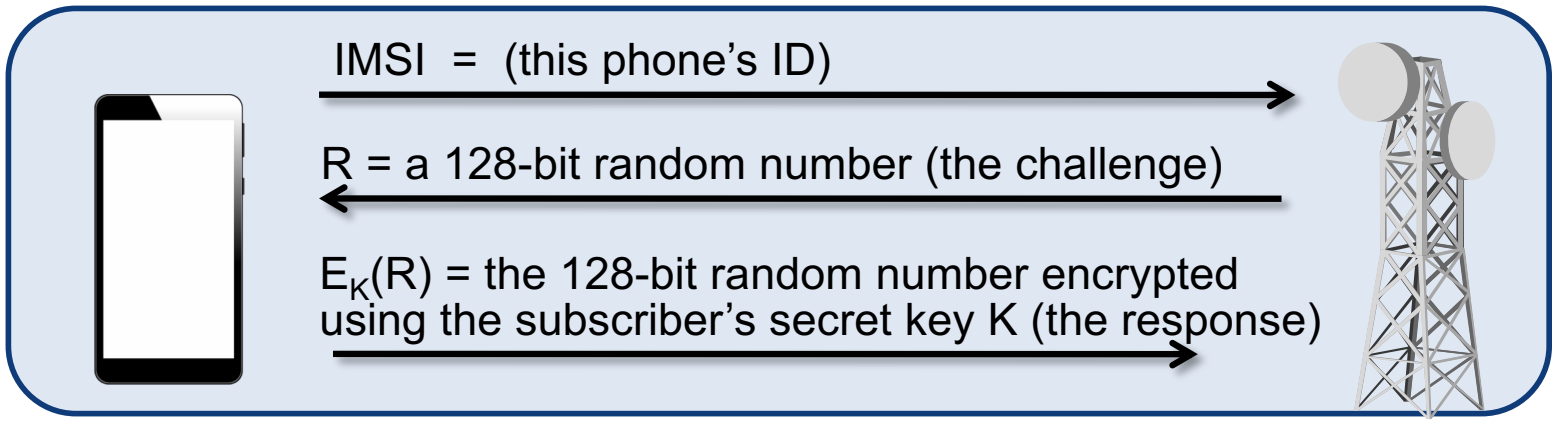
What you Have – Smart Cards

- Integrated circuit on card
  - Memory and processor
- Can provide authentication and data protection
  - “Electronic wallet”
    - “Reskassa” on SL Access card
- Communication through electrical interface or wireless (RFID)
- *SL byter ut korten – för att stoppa hackare*, Computer Sweden, 2015-03-27
  - http://computersweden.idg.se/2.2683/1.618301/sl-byter-ut-korten--for-att-stoppa-hackare

Public domain image from http://en.wikipedia.org/wiki/File:Carte_vitale_anonyme.jpg

What you Have – SIM Cards

- Subscriber Identity Module card
- Several means of identification/authentication
  - **Integrated Circuit Card ID (ICCID)**18-digit hardware identifier.
  - **International Mobile Subscriber Identity** (**IMSI)** identifying the owner.
  - A 128-bit **secret key** for authenticating a phone to a mobile network.
  - PIN to unlock SIM card in phone.



What You Are – Biometrics

- **Biometric** refers to any measure used to uniquely identify a person based on biological or physiological traits.
- Generally, biometric systems incorporate some sort of sensor or scanner to read in biometric information and then compare this information to stored templates of accepted users before granting access.

Image from http://commons.wikimedia.org/wiki/File:Fingerprint_scanner_in_Tel_Aviv.jpg used with permission under the Creative Commons Attribution 3.0 Unported license

Requirements for Biometrics

- **Universality**. Almost every person should have this characteristic.
- **Distinctiveness**. Each person should have noticeable differences in the characteristic.
- **Permanence**. The characteristic should not change significantly over time.
- **Collectability**. The characteristic should have the ability to be effectively determined and quantified.

Candidates for Biometric IDs

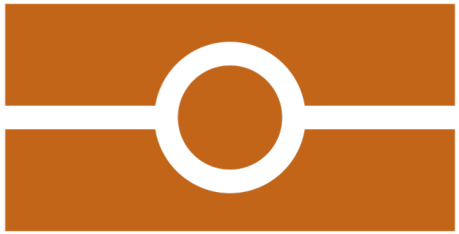
- Fingerprints
- Retinal/iris scans
- DNA
- “Blue-ink” signature
- Voice recognition
- Face recognition
- Gait recognition
- How do these score in
  - universality,
  - distinctiveness,
  - permanence,
  - and collectability…?

Public domain image from
http://commons.wikimedia.org/wiki/File:Fingerprint_Arch.jpg

Public domain image from
http://commons.wikimedia.org/wiki/File:Retinal_scan_securimetrics.jpg

Public domain image from
http://commons.wikimedia.org/wiki/File:CBP_chemist_reads_a_DNA_profile.jpg

Passports



- E-passports contain biometrics information
  - Standards for facial, fingerprint, and iris recognition
- Processor and RFID
- Public-key authentication
- Communication encrypted with secret key

- *Lätt förfalska EU-pass*, Ny Teknik, 2006-08-15.
  - http://www.nyteknik.se/digitalisering/latt-forfalska-eu-pass-6438260

Multi-factor Authentication

- Combine two (or more) factors
- Knowledge factors (what you know)
- Possession factors (what you have)
- Inherence factors (what you are)

- Example:
  - Smartcard (possession) plus password (knowledge)

Authentication Scenarios

Scenario 1	• You present a smartcard at the subway gate to prove that you have a valid ticket.
Scenario 2	• You authenticate yourself to your bank through Mobilt BankID, an app on your phone.
Scenario 3	• You present your passport at the border control to show that you are eligible to enter the country.
Scenario 4	• You use fingerprint recognition on your smartphone to authorize an online purchase. (For example, Touch ID and iTunes Store).

What factors are involved? What you know/have/are?

Project Assignment – Password Cracking

- You get a password file
  - Old UNIX format, with DES hashing and 12-bit salt
- Your task is to crack as many passwords as possible:
  - While not all passwords cracked:
    - Generate password guess
    - Hash it
    - Compare to entries in password file
  - Or, if you prefer, pre-compute a dictionary of hashes
- Report passwords you (think that you) cracked

Summary

- What you know
  - Password authentication
  - Password cracking attacks
  - Password-guessing
  - Dictionary attacks
  - Pre-computed hashes
- What you have
  - Smart cards
  - SIM
- What you are
  - Biometrics