Authentication
CS 236
On-Line MS Program
Networks and Systems Security
Peter Reiher

Outline

- Introduction
- Basic authentication mechanisms

Introduction

- Much of security is based on good access control
- Access control only works if you have good authentication
- What is authentication?

Authentication

- Determining the identity of some entity
  - Process
  - Machine
  - Human user
- Requires notion of identity
- And some degree of proof of identity

Authentication Vs. Authorization

- *Authentication* is determining who you are
- *Authorization* is determining what someone is allowed to do
- Can't authorize properly without authentication
- Purpose of authentication is usually to make authorization decisions

Proving Identity in the Physical World

- Most frequently done by physical recognition
  - I recognize your face, your voice, your body
- What about identifying those we don't already know?

Other Physical Identification Methods

- Identification by recommendation
  - You introduce me to someone
- Identification by credentials
  - You show me your driver's license
- Identification by knowledge
  - You tell me something only you know
- Identification by location
  - You're behind the counter at the DMV
- These all have cyber analogs

Differences in Cyber Identification

- Usually the identifying entity isn't human
- Often the identified entity isn't human, either
- Often no physical presence required
- Often no later rechecks of identity

Identifying With a Computer

- Not as smart as a human
  - Steps to prove identity must be well defined
- Can't do certain things as well
  - E.g., face recognition
- But lightning fast on computations and less prone to simple errors
  - Mathematical methods are acceptable

Identifying Computers and Programs

- No physical characteristics
  - Faces, fingerprints, voices, etc.
- Generally easy to duplicate programs
- Not smart enough to be flexible
  - Must use methods they will understand
- Again, good at computations

Physical Presence Optional

- Often authentication required over a network or cable
- Even if the party to be identified is human
- So authentication mechanism must work in face of network characteristics
  - Active wiretapping
  - Everything is converted to digital signal

Identity Might Not Be Rechecked

- Human beings can make identification mistakes
- But they often recover from them
  - Often quite easily
- Based on observing behavior that suggests identification was wrong
- Computers and programs rarely have that capability
  - If they identify something, they believe it

Authentication Mechanisms

- Something you know
  - E.g., passwords
- Something you have
  - E.g., smart cards or tokens
- Something you are
  - Biometrics
- Somewhere you are
  - Usually identifying a role

Passwords

- Authentication by what you know
- One of the oldest and most commonly used security mechanisms
- Authenticate the user by requiring him to produce a secret
  - Usually known only to him and to the authenticator

Problems With Passwords

- They have to be unguessable
  - Yet easy for people to remember
- If networks connect remote devices to computers, susceptible to password sniffers
- Unless quite long, brute force attacks often work on them

Proper Use of Passwords

- Passwords should be sufficiently long
- Passwords should contain non-alphabetic characters
- Passwords should be unguessable
- Passwords should be changed often
- Passwords should never be written down
- Passwords should never be shared
- Hard to achieve all this simultaneously

Passwords and Single Sign-On

- Many systems ask for password once
  - Resulting authentication lasts for an entire "session"
- Used on its own, complete mediation definitely not achieved
- Trading security for convenience
- Especially if others can use the authenticated machine

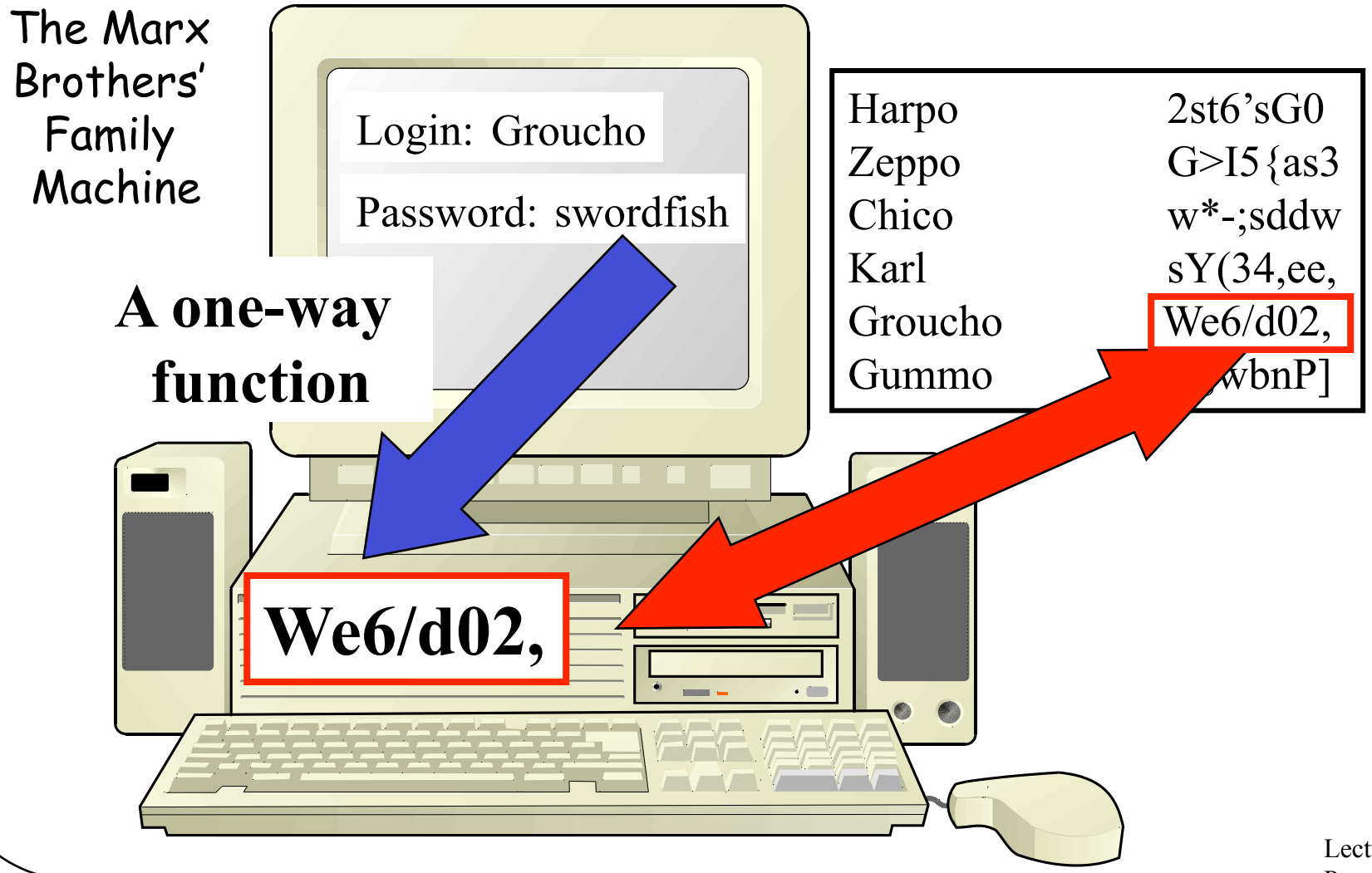
Handling Passwords

- The OS must be able to check passwords when users log in
- So must the OS store passwords?
- Not really
  - It can store an encrypted version
- Encrypt the offered password
  - Using a *one-way function*
- And compare it to the stored version

One Way Functions

- Functions that convert data A into data B
- But it's hard to convert data B back into data A
- Often done as a particular type of cryptographic operation
  - E.g., cryptographic hashing
- Depending on particular use, simple hashing might be enough

Standard Password Handling



Is Encrypting the Password File Enough?

- What if an attacker gets a copy of your password file?
- No problem, the passwords are encrypted
  - Right?
- Yes, but . . .

Dictionary Attacks on an Encrypted Password File

Harpo	2st6'sG0
Zeppo	G>I5{as3
Chico	
Karl	sY(34,ee
Groucho	
Gummo	3(;wbnP]

Now you can hack the Communist Manifesto!

sY(34,ee

Rats!!!!

Dictionaries

- Real dictionary attacks don't use Webster's
- Dictionary based on probability of words being used as passwords
- Partly set up as procedures
  - E.g., try user name backwards
- Checks common names, proper nouns, etc. early
- Tend to evolve to match user trends

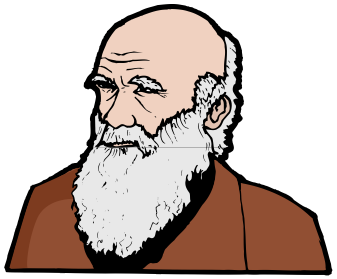
A Serious Issue

- All Linux machines use the same one-way function to encrypt passwords
- If someone runs the entire dictionary through that function,
  - Will they have a complete list of all encrypted dictionary passwords?
  - For all Linux systems?

Illustrating the Problem

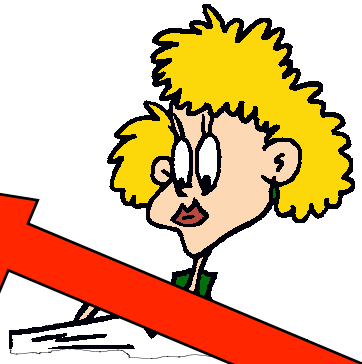
Karl Marx

^*eP6la-



Charles Darwin

^*eP6la-



aardvark
Aardwolf

340ja
K[ds
sY(34

beard

^*eP61a-

The Real Problem

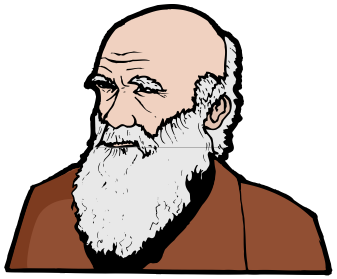
- Not just that Darwin and Marx chose the same password
- But that anyone who chose that password got the same encrypted result
- So the attacker need only encrypt every possible password once
- And then she has a complete dictionary usable against anyone

Salted Passwords

- Combine the plaintext password with a random number
  - Then run it through the one-way function
- The random number need not be secret
- It just has to be different for different users

Did It Fix Our Problem?

Karl Marx



Charles Darwin

D0Cls6&

aardvark
aardwolf

340jafg;
K[ds+3a,
sY(34,ee

beard ^*eP61a-

)#4,doa8

What Is This Salt, Really?

- An integer that is combined with the password before hashing
- How will you be able to check passwords by hashing them, then?
- By storing the salt integer with the password
  - Generally in plaintext
- Note the resemblance to nonces
- Why is it OK (or OK-ish) to leave this important information in plaintext?

Modern Dictionary Attacks

- Modern machines are very fast
- Even with salting, huge dictionaries can be checked against encrypted passwords quickly
- In 2012, Ars Technica challenged 3 hackers to crack 16,000 hashed, salted passwords
  - Using dictionary attacks, they got 90% of them in 20 hours
  - Why? Weak password choices