

CS 493 - Authentication & Access Control

Note Title

1/24/2011

Announcements

- First assignment (article summary) due Thursday.

length (300 words) - 10 points

Interest - up to 10 points

Grammar - 25 points

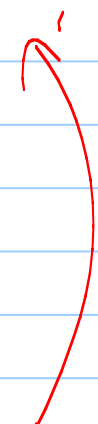
Thesis - 25 points

Facts to support your position, taken from article (30 points) - at least 4 facts

- Lab next week (Thursday in class)

Authentication (Access control)

4 basic strategies

- 1) Something you know ↳ passwords!
 - 2) Something you possess
 - 3) Something you are
 - 4) Something you do
- 

Which of these is the most common?

Passwords

Common Attacks:

- network sniffing
- brute force (dictionary attack)
- - getting file w/ password
- - physical access
- social engineering

Defenses against password attacks

- up to 5 login attempts
- automatic logout
- password policies
(user education)
- reactive password checking
(try to hack passwords)
- separate passwords from
user ids
(hashed passwords)
- salting the hash
- resetting passwords
- intentionally slow down
authentication

Hashed Passwords

In general, only hashed versions of passwords are saved.

Why?

Password files were irresistible.

One extra layer of protection.

In addition, there is generally a "random" piece of information.

Salt → This is used in the hashing in addition to the password.

Why? Makes sure repeat passwords hash to different values.

Usually this creation time ~~of~~ account.

$h(\text{password} + \text{salt})$

Unix Implementation

- User password of 8 digits
↳ 56-bit value
- 12 bit salt value, usually based on account creation time
← now, much larger
- Hash function (based on DES - more later) is run ~ 25 times.
- Resulting 64-bit value is converted to ~~11~~ character sequence

Sound impressive?

In 2003, a supercomputer managed over 50 million password guesses in 80 minutes.

(Back then, a regular machine could have done the same in about 1 month.)

(faster now)

Stronger variants essentially use stronger & slower hash algorithms.

(One even just runs a dummy for loop!)

Single Most Important Defense

user education

Choose Secure passwords!!

Use "random" phrases, &
take first letter of each
word.

Incorporate a letter or 2 from
website.

Password Checkers

Algorithms that allow or reject passwords based on how likely they are to be cracked.

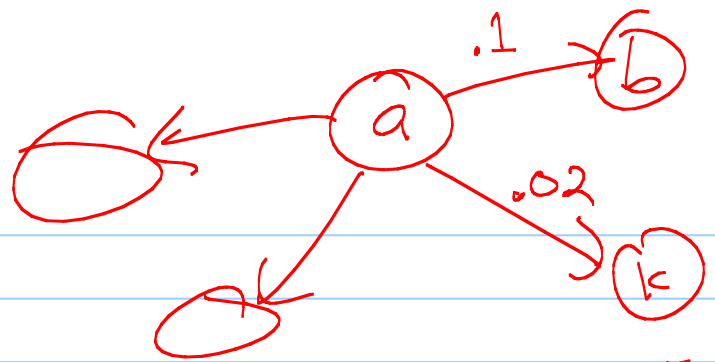
① Rule enforcement:

check 6-8 letters or digits

not your name, "password",
or dictionary word

at least 1 number...

② Markov model



Consider current letter, probability that next letter is any other letter.

③ Bloom filter

Token - Based Authenticators

(something you possess)

- ATM cards
 - key fobs w/ "random" # generators
 - RFID
 - key card
-

- Steal
- * - social engineering
 - keeping card after leaving
 - packet sniffing
 - forge (cloning)

Biometric Authentication

Something you are or do

Hard to steal

Expensive

People change

Difficult to make effective

At heart these are algorithms.
possible to fool them.

A Note about Remote Authentication

Goal: Give eavesdroppers as little information as possible.

Sample (+ simple) protocol:

- 1) user transmits identity
- 2) host sends a nonce (random #), r , plus 2 functions $h()$ + $f()$
- 3) user sends: $f(r, h(\text{password}))$

Defends against: