

CSE331: Introduction to Networks and Security

Lecture 32
Fall 2006

Announcements

- Project 4 is due Dec. 8th at midnight.
- Homework 3 is out:
 - Due. Dec. 8th at the start of class
- There was a bug in solutions for HW 2 problem 1(e)
 - Check out the new solutions on the web
 - If you were docked points for a correct answer, submit your HW to me for a regrade.
- Final Exam:
 - Friday Dec. 15th 12:00--2:00
 - CHEM B13

The “Gold” Standard

- *Authentication*
 - Identify which principals take which actions
 - Talked about this considerably
- *Authorization*
 - Determine what actions are permissible
 - Policy
- *Audit*
 - Recording the security relevant actions
 - Allows for post-mortem when problems occur



Authenticating Humans: Foundations

- Authentication is based on one or more of the following:
 - **Something you know**
 - password
 - **Something you have**
 - driver's license, Penn Card
 - **Something inherent about you**
 - Biometrics, location
- What's the most common method of authentication?



Guessing

- The "no such user" mistake
- The "here's who we are" mistake
- Common words, phrases for passwords
- Null passwords, "password", username, backwards, etc.
- Dictionary attacks

- How bad is it?



1979 Survey of 3,289 Passwords

- With no constraints on choice of password, Morris and Thompson got the following results:
 - 15 were a single ASCII letter.
 - 72 were strings of two ASCII letters.
 - 464 were strings of three ASCII letters.
 - 47 were strings of four alphanumerics.
 - 706 were five letters, all upper-case or all lower-case.
 - 605 were six letters, all lower case.



1990s Surveys of 15K Passwords

- Klein (1990) and Spafford (1992)
 - 2.7% guessed in 15 minutes
 - 21% in a week
 - Sounds ok? Not if the passwords last 30 days
- Tricks
 - Letter substitutions, words backwards, common names, patterns, etc.
 - Anything you can think of off the top of your head, a hacker can think of too
- Lazy users!
 - Weakest link is always the way of the attack



Heuristics for Guessing Attacks

- The dictionary with the words spelled backwards
- A list of first names (best obtained from some mailing list). Last names, street names, and city names also work well.
- The above with initial upper-case letters.
- All valid license plate numbers in your state. (About 5 hours work in 1979 for New Jersey.)
- Room numbers, social security numbers, telephone numbers, and the like.

What makes a good password?

- Password Length
 - 64 bits of randomness is hard to crack
 - 64 bits is roughly 20 “common” ASCII characters
 - But... People can't remember random strings
 - Longer not necessarily better: people write the passwords down
- Pass phrases
 - English Text has roughly 1.3 random bits/char.
 - Thus about 50 letters of English text
 - Hard to type without making mistakes!
- In practice
 - Non-dictionary, mixed case, mixed alphanumeric
 - Not too short (or too long)



Hacks on plaintext password file

- Is the password file readable by the OS?
 - Then if I break the OS
- Can privileged users see the file?
 - ... and make copies
- Is the file backed up somewhere
 - ... insecure?
- Is the file in plaintext somewhere in memory?
 - Core dump
- Fool the user
 - A program that masquerades as the authentication program

Counter-hacks

- Control-Alt-Del for logging in
 - For windows only
- Slow down
 - Make guessing take too long
- Encrypt the password file
 - “Salt” - to prevent duplicates
 - Use one way hashes or encryptions on the passwords
- Password rules
 - Min length, upper and lower case, no common words
 - Use letters and numbers and symbols
 - Change often
 - Keep a password history
 - Don't write it down!

Add Salt

- “Salt” the passwords by adding random bits.
 - Decreases the likelihood that two identical passwords will appear as identical entries in the password file.
- 12 bit salt results in 4,096 versions of each password.
- /etc/passwd entry:

user_id	salt _u	Hash(salt _u + passwd _u)	...
---------	-------------------	--	-----

- Actually most modern implementations use so-called *shadow* password files /etc/shadow that aren't world readable.



One Time Passwords

- Shared lists.
- Sequentially updated.
- One-time password sequences based on a one-way (hash) function.

- Used in practice: SKey mechanism

Hash-based 1-time Passwords

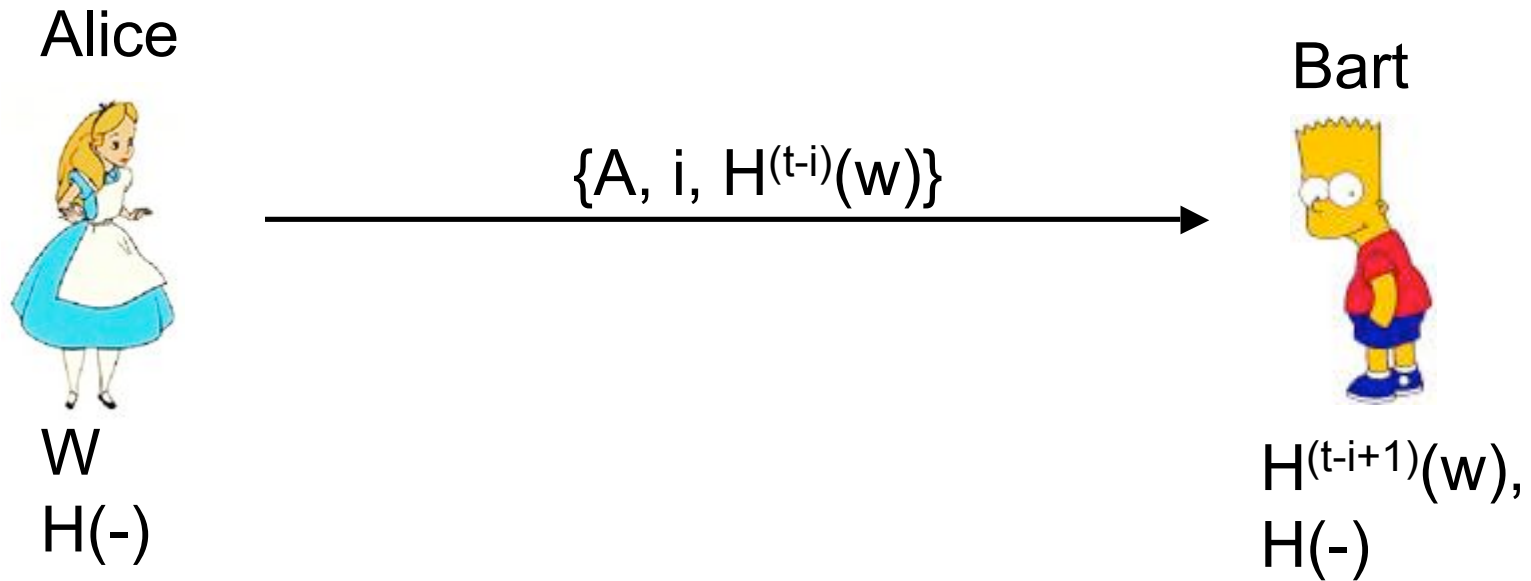
- Alice identifies herself to verifier Bart using a well-known one-way hash function H .
- One-time setup.
 - Alice chooses a secret w .
 - Fixes a constant t for the number of times the authentication can be done.
 - Alice securely transfers $H^t(w)$ to Bart

$$\underbrace{H(H(H\dots(H(w))\dots))}_{t \text{ times}}$$

Hash-based 1-time Passwords

- Protocol actions. For session i , claimant A does the following to identify itself:
 - A computes $w' = H^{(t-i)}(w)$ and transmits the value to B .
 - B checks that i is the correct session (i.e. that the previous session was $i-1$) and checks to see if $H(v) = w'$ where v was the last value provided by A (as part of session $i-1$).
 - B saves w' and i for use in the next session.

One-time passwords: i^{th} authentication



- Alice does the following to identify herself:
 - A computes $w' = H^{(t-i)}(w)$ and transmits the value to B.
 - B checks that i is the correct session (i.e., that the previous session was $i-1$) and checks to see if $H(w') = v$ where v was the last value provided by A (as part of session $i-1$).
 - B saves w' and i for use in the next session.



Why This 1-time Password Works

- It's hard to compute x from $H(x)$.
 - Even though attacker gets to see $H^{(t-i)}(x)$, they can't guess then next message $H^{(t-(i+1))}(x)$.



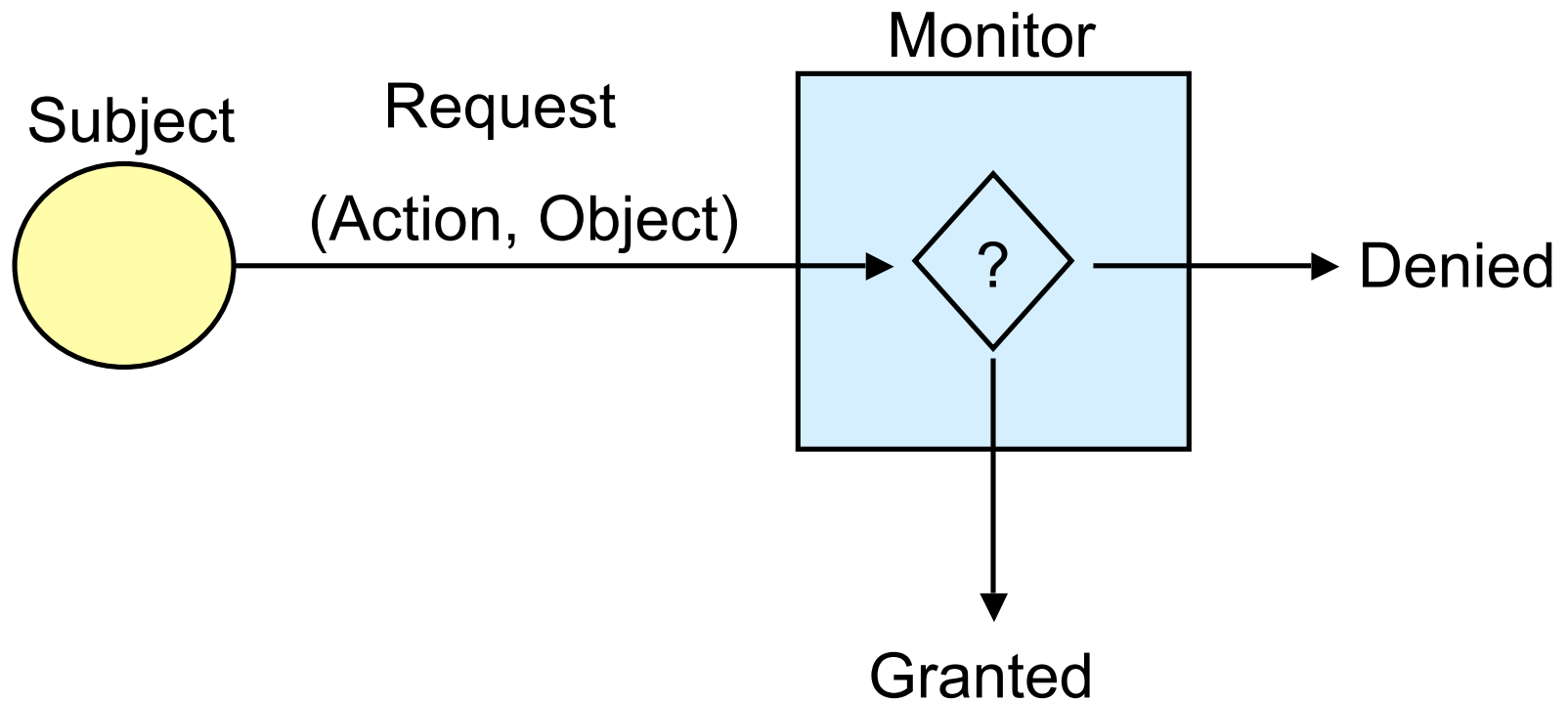
Authorization

- Authorization is the process of determining whether a principal is permitted to perform a particular action.
- Access control
 - Example: Read/Write/Execute permissions for a file system.
 - Example: Java applets have restricted authorization to perform network & disk I/O.

Policy vs. Mechanism

- Access control policy is a *specification*
 - Given in terms of a model of the system
 - Subjects: do things (i.e. a process writes to files)
 - Objects: are passive (i.e. the file itself)
 - Actions: what the subjects do (i.e. read a string from a file)
 - Rights: describe authority (i.e. read or write permission)
- Mechanisms are used to *implement* a policy
 - Example: access control bits in Unix file system & OS checks
 - Mechanism should be general; ideally should not constrain the possible policies.
 - Complete mediation: every access must be checked

Reference Monitors





Example Reference Monitors

- Operating Systems
 - File system
 - Memory (virtual memory, separate address spaces)
- Firewalls
 - Regulate network access
- Java Virtual Machine
 - Regulates Java programs' resource usage
- Operate at different levels of abstraction
 - Interface (Subjects, Objects, Actions) varies