

Agenda

- **Last time (Tues Feb 27)**
 - Security (chapt 7)
- **This time**
 - Security finish (chapt 7)
- **NEXT WEEK: Spring BREAK!!**
- **Next time (Tues Mar 13)**
 - Dis File systems (chpt 8)
 - Assignment #3 out
- **Thurs Mar 15: Name services (chpt 9)**
- **Tues Mar 20: Midterm**

CS451: Distributed Systems (Spring 2007)

Before we start: Assignment #2

- **New due date: Tues Mar 6 12:30pm (mid Spring break)**
 - We'll print things out for you

•"

CS451: Distributed Systems (Spring 2007)

Wrapping up Security....

	PRO	CON
Secret algorithms	<ol style="list-style-type: none">1. They actually work2. Simple	<ol style="list-style-type: none">1. Useless when stolen2. Not subject to public scrutiny3. Algorithm distribution
Symmetric algorithms	<ol style="list-style-type: none">1. Faster than "comparable" asymmetric	<ol style="list-style-type: none">1. Key distribution
Asymmetric algorithms	<ol style="list-style-type: none">1. Public key distribution (CAs)	<ol style="list-style-type: none">1. Public key distribution (CAs)

CS451: Distributed Systems (Spring 2007)

SSL – secure channels

- **SSL: Secure Socket Layer**
- **De facto standard protocol for use in applications requiring secure channels.**
- **A secure channel ensures:**
 - That each of the processes knows reliably the identity of the principal on whose behalf the other process is executing.
 - The privacy and integrity of the data transmitted across it.
 - Each message includes a physical or logical time stamp to prevent messages from being replayed or reordered.

CS451: Distributed Systems (Spring 2007)

SSL – main features

- **Negotiable encryption and authentication algorithms:**
 - No assumption that client and server use a particular encryption algorithm.
 - Algorithms are negotiated.
 - Connections may fail.
- **Bootstrapped secure communication:**
 - Unencrypted communication for the initial exchange.
 - Then public-key cryptography.
 - Finally secret-key cryptography.

CS451: Distributed Systems (Spring 2007)

SSL Handshake

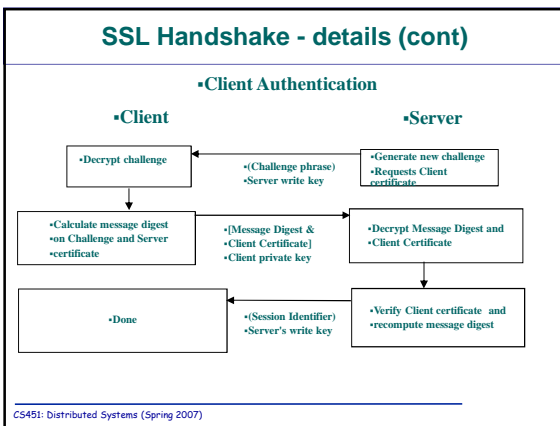
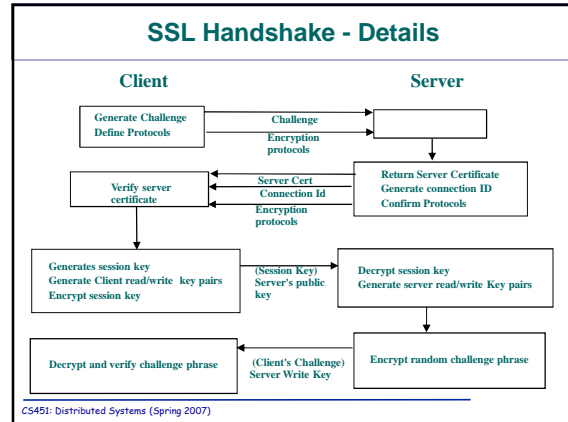
- **Negotiate the cipher suite**
- **Establish a shared session key**
- **Authenticate the server (optional)**
- **Authenticate the client (optional)**
- **Authenticate previously exchanged data**

CS451: Distributed Systems (Spring 2007)

SSL Handshake Details

- **Client hello:**
 - Client challenge
 - Available cipher suites (e.g. RSA + RC4/40 + MD5)
- **Server hello:**
 - Server certificate
 - Connection ID
 - Selected cipher suite
- **Server adapts to client capabilities**
- **Optional certificate exchange to authenticate server/client (Commercial sites: only server authentication)**

CS451: Distributed Systems (Spring 2007)



Logging into a UNIX Box Non-Remotely

- **/etc/passwd was world-readable. Why?**
 - Used by other subsystems of the OS (Contained other information besides just password)
- **Contains f(password), not password itself**
 - Assumption: cannot go from f(password) → password
 - Problem: offline hacking/cracking ("dictionary attack")
- **Now**
 - /etc/shadow only readable by superuser
 - Setuid bit for reading and writing

CS451: Distributed Systems (Spring 2007)

Logging into a UNIX Box Remotely

- **Problem with telnet (or ftp) is cleartext passwords**
 - "sniffing"
- **Solution: SSH**
 - Client communicates with a well-known port (22)
 - Server replies with its public key
 - After challenge-response (for the client to make sure that the machine indeed possesses the private key that corresponds to the public key), public key is used to transmit the password in cyphertext
 - **Problem:** public key in this protocol is not certified (!)

CS451: Distributed Systems (Spring 2007)

How "secure" is the service?

- It's a function of:
 - Crypto
 - Defaults / config
 - Human factors
- **Anything else?**

CS451: Distributed Systems (Spring 2007)