Certification-Based Authentication

Certification authorities issue signed certificates
- Banks, companies, & organizations like Verisign act as CA's
- Certificates bind a public key to the name of a user
- Public key of CA certified by higher-level CA's
- Root CA public keys configured in browsers & other software
- Certificates provide key distribution

Certificate-Based Authentication (2)

Authentication steps
- Verifier provides nonce, or a timestamp is used instead.
- Principal selects session key and sends it to verifier with nonce, encrypted with principal’s private key and verifier’s public key, and possibly with principal’s certificate
- Verifier checks signature on nonce, and validates certificate.

Secure Sockets Layer (and TLS)

Encryption support provided between browser and web server - below HTTP layer
Client checks server certificate
Works as long as client starts with the correct URL
Key distribution supported through cert steps
Authentication provided by verify steps

Trust models for certification

- X.509 Hierarchical
  - Single root (original plan)
  - Multi-root (better accepted)
  - SET has banks as CA's and common SET root
- PGP Model
  - "Friends and Family approach" - S. Kent
- Other representations for certifications
- No certificates at all
  - Out of band key distribution
  - SSH
Global Authentication Service

- Pair-wise trust in hierarchy
  - Name is derived from path followed
  - Shortcuts allowed, but changes name
  - Exposure of path is important for security
- Compared to Kerberos
  - Transited field in Kerberos - doesn't change name
- Compared with X.509
  - X.509 has single path from root
  - X.509 is for public key systems
- Compared with PGP
  - PGP evaluates path at end, but may have name conflicts

Proxies

- A proxy allows a second principal to operate with the rights and privileges of the principal that issued the proxy
  - Existing authentication credentials
  - Too much privilege and too easily propagated
- Restricted Proxies
  - By placing conditions on the use of proxies, they form the basis of a flexible authorization mechanism

Restricted Proxies

- Two Kinds of proxies
  - Proxy key needed to exercise bearer proxy
  - Restrictions limit use of a delegate proxy
- Restrictions limit authorized operations
  - Individual objects
  - Additional conditions

Generic Security Services API

Standard interface for choosing among authentication methods

- Once an application uses GSS-API, it can be changed to use a different authentication method easily.
- Calls
  - Acquire and release cred
  - Manage security context
  - Init, accept, and process tokens
  - Wrap and unwrap

Authentication in Applications

- Unix login
- Telnet
- RSH
- SSH
- HTTP (Web browsing)
- FTP
- Windows login
- SMTP (Email)
- NFS
- Network Access

Unix Login (review)

- One way encryption of password
  - Salted as defense against pre-computed dictionary attacks
  - To validate, encrypt and compare with stored encrypted password
  - May use shadow password file
### Telnet

**A remote login application**
- Normally just an unencrypted channel over which plaintext password is sent.
- Supports encryption option and authentication options using protocols like Kerberos.

### RSH (Remote Shell/Remote Login)

**Usually IP address and asserted account name.**
- Privileged port means accept asserted identity.
- If not trusted, request unix password in clear.
- Kerberos based options available
  - Kerberos based authentication and optional encryption

### Secure Shell (SSH)

**Encrypted channel with Unix login**
- Establish encrypted channel, using public key presented by server
- Send password of user over channel
- Unix login to validate password.
- Public key stored on target machine
- User generates Public Private key pair, and uploads the public key to directory on target host.
- Target host validates that corresponding private key is known.

### Web Browsing (HTTP)

**Connect in the clear, Unix Password**
- Connect through SSL, Unix password
- Digest authentication (RFC 2617)
- Server sends nonce
- Responds is MD5 checksum of Username, password, nonce URI
- User certificate, strong authentication

### File Transfer Protocol

**Password based authentication or**
- GSS-API based authentication
  - Including use of Kerberos
  - Authentication occurs and stream is encrypted

### Windows Network Login

**In Win2K and later uses Kerberos**
- In Win NT
  - Challenge response
  - Server generates 8 byte nonce
  - Prompts for password and hashes it
  - Uses hash to DES encrypt nonce 3 times
Email

SMTP – To send mail
  Usually network address based
  Can use password
  Can be SSL protected
  SMTP after POP

Post Office Protocol
  Plaintext Password
  Can be SSL protected
  Eudora supports Kerberos authent
  IMAP
  Password authentication
  Can also support Kerberos

File System Authentication

Sun’s Network File System
  Typically address based
  Athena Kerberized version
  Maps authenticated UID’s to addresses
  NFS built on ONC RPC
  ONC RPC has stronger Kerberos/GSSAPI support

Andrew File System
  Based on Andrew RPC
  Uses Kerberos authentication
  OSF’s DCE File System (DFS)
  Based on DCE RPC
  Uses Kerberos authenciation

Network Access Servers

Radius
  Problem: Not connected to network until connection established
  Need for indirect authentication
  Network access server must validate login with radius server.
  Password sent to radius server encrypted using key between agent and radius server

Delegated Authentication

Usually an authorization problem
  How to allow an intermediary to perform operations on your behalf.
  Pass credentials needed to authenticate yourself
  Apply restrictions on what they may be used for.