Review

- Chapter 1: Basic Concepts and Terminology
- Chapter 2: Basic Cryptographic Tools
- Chapter 3 – User Authentication
- Chapter 4 – Access Control Lists
- Chapter 5 – Database Security (skipped)
- Chapter 6 – Malicious Software
- Networking Basics (not in book)
- Chapter 7 – Denial of Service
- Chapter 8 – Intrusion Detection
- Chapter 9 – Firewalls and Intrusion Prevention
- Chapter 10 – Buffer Overflow
- Chapter 11 – Software Security
- Chapter 12 – OS Security
- Chapter 22 – Internet Security Protocols
- Chapter 23 – Internet Authentication Applications
Chapter 23

Internet Authentication Applications
Kerberos Overview

- initially developed at MIT
- software utility available in both the public domain and in commercially supported versions
- issued as an Internet standard and is the defacto standard for remote authentication
- overall scheme is that of a trusted third party authentication service
- requires that a user prove his or her identity for each service invoked and requires servers to prove their identity to clients
Kerberos Protocol

- designed to counter a variety of threats to the security of a client/server dialogue
- obvious security risk is impersonation
- servers must be able to confirm the identities of clients who request service

involves clients, application servers, and a Kerberos server

- user initially negotiates with AS for identity verification
- AS verifies identity and then passes information on to an application server which will then accept service requests from the client

use an Authentication Server (AS)

need to find a way to do this in a secure way

- if client sends user’s password to the AS over the network an opponent could observe the password
- an opponent could impersonate the AS and send a false validation
Kerberos Overview

1. User logs on to workstation and requests service on host.

2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.

3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.

4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.

5. Workstation sends ticket and authenticator to server.

6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

Figure 23.1 Overview of Kerberos
Kerberos Realms

- a Kerberos environment consists of:
  - a Kerberos server
  - a number of clients, all registered with server
  - a number of application servers, sharing keys with server
- this is referred to as a realm
  - networks of clients and servers under different administrative organizations generally constitute different realms
- if multiple realms:
  - their Kerberos servers must share a secret key and trust the Kerberos server in the other realm to authenticate its users
  - participating servers in the second realm must also be willing to trust the Kerberos server in the first realm
Kerberos Realms

Figure 23.2 Request for Service in Another Realm
Kerberos Versions 4 and 5

• Kerberos v4 is most widely used version
• improvements found in version 5:
  – an encrypted message is tagged with an encryption algorithm identifier
    • this enables users to configure Kerberos to use an algorithm other than DES
  – supports authentication forwarding
    • enables a client to access a server and have that server access another server on behalf of the client
    • supports a method for interrealm authentication that requires fewer secure key exchanges than in version 4
Kerberos Performance Issues

- see larger client-server installations

  environment:
  - very little if system is properly configured
  - tickets are reusable which reduces traffic

- Kerberos performance impact in a large-scale Kerberos security is best assured by placing the Kerberos server on a separate, isolated machine

- motivation for multiple realms is administrative, not performance related
Certificate Authority (CA)

certificate consists of:

• a public key plus a User ID of the key owner
• signed by a trusted third party
• typically the third party is a CA that is trusted by the user community (such as a government agency or a financial institution)

user can present his or her public key to the authority in a secure manner and obtain a certificate

• user can then publish the certificate
• anyone needing this user’s public key can obtain the certificate and verify that it is valid by way of the attached trusted signature
### X.509 Authentication Service

<table>
<thead>
<tr>
<th>Universally accepted standard for formatting public-key certificates</th>
<th>Widely used in network security applications, including IPsec, SSL, SET, and S/MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part of CCITT X.500 directory service standards</td>
<td>Uses public-key crypto &amp; digital signatures</td>
</tr>
<tr>
<td></td>
<td>Algorithms not standardized, but RSA recommended</td>
</tr>
</tbody>
</table>
X.509 Certificates

(a) X.509 Certificate

(b) Certificate Revocation List

Figure 23.3 X.509 Formats
Figure 23.4 PKIX Architectural Model
PKIX Management Functions

registration

initialization

certification

key pair recovery

key pair update

revocation request

cross certification
Federated Identity Management

• use of common identity management scheme
  – across multiple enterprises and numerous applications
  – supporting many thousands, even millions of users

• principal elements are:
  – authentication, authorization, accounting, provisioning, workflow automation, delegated administration, password synchronization, self-service password reset, federation
Identity Management

Principals provide attributes

Administrators provide attributes

Data consumers apply references to obtain attribute data

Data consumers obtain identifiers, attribute references

Identity Provider
- Identity control interface
- Attribute locator
- Principal authentication
- Identifier translation

Figure 23.5 Generic Identity Management Architecture
Standards Used

- **Extensible Markup Language (XML)** characterizes text elements in a document on appearance, function, meaning, or context.
- **Simple Object Access Protocol (SOAP)** for invoking code using XML over HTTP.
- **WS Security** set of SOAP extensions for implementing message integrity and confidentiality in Web services.
- **Security Assertion Markup Language (SAML)** XML-based language for the exchange of security information between online business partners.
Federated Identity Management

(b) Federation based on roles

(b) Chained Web Services

Figure 23.6 Federated Identity Scenarios
Summary

- Kerberos
- Kerberos protocol
- Kerberos realms
- Kerberos versions 4 and 5
- Kerberos performance issues
- X.509

- public-key infrastructure
- PKIX management functions
- PKIX management protocols
- federated identity management