InfoSec Tutorial: Access Control

Tony Kenyon, CEO.

Revision 1.01.

Updated: Jan 5th 2006

Ref: CNXT0004

Access Control Systems

- Necessary for preserving C.I.A
  - **Protection Domain**: A group of processes that share access to the same resources
- Integrity
  - **Prevent modification** of info by unauthorised users
  - **Prevent unauthorised/unintentional modification** of info by authorised users
  - **Preserving internal and external consistency**

Access Control Systems

- Controls
  - Used to mitigate risk or reduce potential loss
  - **Preventative**, **Detective** or **Corrective**
  - Implemented as:
    - **Administrative**: policies, procedures, training, background check, work habit checks, vacation history, increased supervision
    - **Logical/Technical**: encryption, smart cards, ACLs, transmission protocols, firewalls, IDS.
    - **Physical**: door locks, secure server rooms, cable protection, separation of duties, backups

Access Control Models

- Mandatory
  - **Subject-object labels** (clearance, classification, sensitivity).
  - Still Need-to-know even for clearance at same level. Rule-based
  - SSP (cannot read up), Star Properties (cannot write down)
- Discretionary
  - Subject has some authority to specify what objects are accessible. E.g. using **ACLs**.
  - **Access Control Triple** (user, program, file).
  - **User** or **Identity** based, or hybrid.
  - Used in **local dynamic situations** where some local discretion is required.
- Non-Discretionary
  - A **central Authority** determines access rights, based on security policy
  - **Role-based**: job title, group etc. or **Task based** (function)
  - Used where **frequent changes in personal** are made (access rights stay with the role or task)

Control Sets

- Preventative/Administrative
- Preventative/Technical
- Preventative/Physical
- Detective/Administrative
- Detective /Technical
- Detective /Physical

ID and Authentication

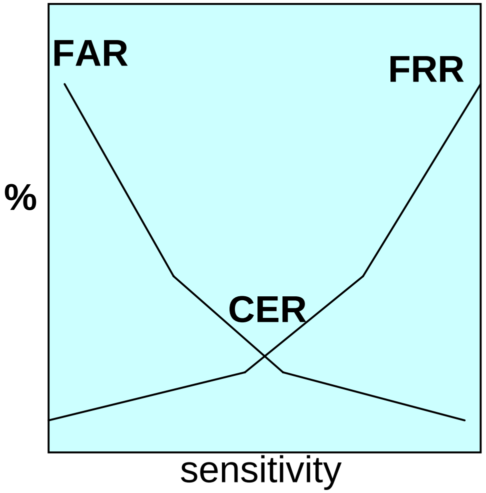
- Three factors
  - Type 1: Something you know (e.g. a PIN number)
  - Type 2: Something you Have (e.g. a smart card)
  - Type 3: Something you are (e.g. a fingerprint)
  - And also possibly something you do.
- Components
  - Type 1: Password (one-time, static, dynamic, pass-phrase)
  - Type 2: Token (smart cards – supply both static and dynamic passwords)

Smart Cards

- Main Types
  - Static Password Tokens
  - Synchronous Dynamic Password Tokens
  - Asynchronous Dynamic Password Tokens
  - Challenge-Response Tokens

Biometrics

- Type 3 factor authentication system
- Performance Measures:
  - Type 1 Error: False Rejection Rate (FRR): % of valid subjects falsely rejected
  - Type II Error: False Acceptance Rate (FAR): % of invalid subject falsely accepted
  - Crossover Error Rate (CER): % in which FRR=FAR. Measures performance.
    - If sensitivity is increased get a higher FRR
    - Conversely desensitising the system gets a higher FAR.
- Other key factors
  - Enrolment time (2 mins considered acceptable)
  - Throughput rate (10 subject per minute considered acceptable)
  - Acceptability (privacy, invasiveness, comfort)



Biometrics

- Typical Biometrics
  - Fingerprints, Retina Scans, Iris Scans, Facial Scans, Palm Scans, Hand Geometry, Voice
  - Handwritten Signature dynamics
- ‘Feature-Extraction’
- Fingerprint
  - High quality image requires approx 250KB per finger.
    - Used for one-to-many searches in very large databases.
  - Finger-scan technology stores only attributes and requires approx 0.5KB or 1KB storage. Cannot reconstruct the image.
    - Used for one-to-one scans in smaller databases

Single Sign On (SSO)

- SSO addresses the issue of multiple sign-on/passwords
  - Better admin, stronger passwords
  - But, once a password available user is free to roam
- Open Group SSO Standard
  - Objectives
    - Interface is independent of the type of authentication
    - No predefined timing of secondary sign on operations
    - Support for default user profiles
  - Scope of service defs to support
    - Apps for common single end user signon for enterprises
    - Apps for ccordinated mgt of multiple user account mgt databases for enterprises
  - SSO can be implemented by:
    - scripts to reply user user logins
    - Authentication servers that provide encrypted tickets

Single Sign On (SSO)

- Enterprise Access Management (EAM)
  - Web SSO
  - Role-Based access control
  - Accommodates several authentication schemes
  - Implemented in a number of ways, e.g.
    - Non-Persistent, Encrypted cookies on clients, for web apps in the same domain on multiple servers. A cookie is provided to each application the user wishes to access.
    - Build a secure credential for each user on a reverse proxy in front of the web server. The credential is presented each time to user accesses protected web apps.
  - Does NOT provide interoperability amongst implementers.

SSO – Authentication Servers

- Examples of Authentication Servers that can implement SSO include:
  - SESAME
  - KryptoKnight
  - NetSP

SSO – Kerberos

- Background
  - MIT Project Athena
  - Uses **Symmetric Key** Cryptography
  - Authenticates clients to entities on the network
  - Built into Windows 2000 as standard
  - Addresses confidentiality and Integrity of information

SSO – Kerberos

- Issues
  - Does not address availability and attacks (e.g. frequency analysis)
  - Both TGS and AS hold secret keys and are therefore vulnerable
  - Replay is possible if compromised tickets are available within the allotted time window
  - Since client password is used in initiating Kerberos requests password guessing can be used to impersonate a client
  - Keys are vulnerable because they are stored temporarily on machines (client secret key is stored locally, and session keys stored on both client and servers)

SSO – Kerberos

- Components
  - KDC: trusted **Key Distribution Centre**
  - TGS: **Ticket Granting Service**
  - AS: **Authentication Service**
- Operations
  - KDC holds all secret keys of clients and servers
  - KDC initially communicates with clients & servers using secret key
  - Kerberos authenticates clients to services (on a server) via TGS
    - Uses temporary symmetric session keys for client-KDC, server-KDC, and client-server communications
  - Client-Server communication then proceeds using the temporary session key

source: www.cyphernetix.co.uk

SSO – SESAME

- Background
  - Designed to address weaknesses in Kerberos
    - Uses public key cryptography for key distribution
    - Additional access control support
- Characteristics
  - **Needham-Schroeder protocol** and a **trusted Authentication Server** at each host to reduce key management requirements
  - Uses **MD5** and **crc32** one-way hashes
  - Incorporates 2 certificates (tickets): **Authentication** and **Access Privileges**
- Issues
  - **Authenticates** using only the **first block** of the message

SSO – KryptoKnight

- Background
  - IBM. Designed for mixed performance systems
  - Provides authentication, SSO, key distribution services
- Operations
  - Uses a **Trusted Key Distribution Centre** (KDC)
    - Knows the secret key of each party
  - **Peer-peer relationship** between parties and the KDC
  - Secret key is a one-way hash of the password
  - Client to KDC initiates with a user name, a value (nonce) and the password.
  - KDC returns ticket, encrypted with the user's secret key.
  - This ticket is used for authenticating to services
- NetSP is based on KryptoKnight, uses a workstation as an AS, and tickets are compatible with RACF and other access control servers

Access Control Methodologies

- Centralised
  - Dialup users can use RADIUS, Call Back, CHAP, PAP.
    - **Call Forwarding** is a dial-back attack
  - Networked Apps can use TACACS. TACACS+ is two-factor.
- Decentralised/Distributed
  - Typically via databases

Database Security

- Relational Database has 3 parts
  - Data structures (tables, relations)
  - Integrity Rules (allowable values)
  - Operators (on data in the tables)
- Overview
  - Database description is its **schema**, defined in **Data Description Language** (DDL)
  - **Database Management System** (DBMS) provides and maintains access to the database
  - **Relation**: represented by a 2-dimensional table
    - **Rows**: **records** (**tuples**)
    - **Columns** (**attributes**)
  - **Cardinality**: no of rows
  - **Degree**: no of columns
  - **Domain** of a relation is the set of allowable values for an attribute

RDBMS - Keys

- **Keys**
  - **Primary key**:
    - each table requires a unique identifier that unambiguously points to an individual tuple (record).
      - I.e. a column with unique entries (e.g. part number), that can be used to uniquely pull out a single record
    - A subset of the **candidate** keys within a table
      - I.e. where two columns may be potential primary keys
  - **Foreign Key**
    - A key in Table B that is used as the Primary key is Table A.
- Entity and Referential Integrity
  - Entity Integrity: Primary Key column cannot have NULL entries
  - Referential Integrity: tuple used by the foreign key must mach the primary key

RDBMS - Views

- A virtual table
  - Defined from operations Join, Project and Select.
  - Query Plan (optimal cost) and Binds
- Important for access control
  - Restrict access to data in a context or role dependent way
  - Implements **Least Privilege**
- Normalisation
  - Eliminating redundant data
  - Eliminating repeating groups
  - Eliminating attributes not dependent on the Primary Key

source: www.cyphernetix.co.uk

Object Databases

- Object Oriented Databases (OODB)
  - Suited where data is often non-text (images etc)
- Object-Relational Database
  - Marriage of RDBMS and OODB
  - Introduced in 1992 as UniSQL/X
  - HP later released OpenODB (later called Odapter)

Further Research

- IDS
- Access Control Matrix (rows are ACLs)
- Reference Monitor - Security Kernel
- Clipping Levels – Audit Logs
- GSM – uses symmetric key
- GPRS – uses IPSec

Questions?