

Applications System Development

❖ *DigitalSherlock.com | SafeHack.com*

❖ *Date: 2005/05/27*

❖ *Document Name: applications_system_development.pdf*

❖ *GNU Free Documentation License*

❖ *Version 1.00, 2005-05-27*

❖ *Copyright © 2005 Adonis, MSc, Eng, CISSP, Security+, CEH, GSec, MCSE, etc.*

❖ *Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation.*

❖ Databases and Data Warehousing

➤ Types of databases

- - Hierarchical
- - Mesh
- - Object-oriented
- - Relational

➤ DBMS / Database Management System

- A suite of programs used to manage large sets of structured data with ad hoc query capabilities for many types of users

➤ Database

- A collection of data stored in a meaningful way that enables multiple users and applications to access, view and modify data as needed.

- **Database terms/jargon**

- - Record: Collection of related data items
- - File: Collection of record of the same type
- - Database: Cross-referenced collection of files
- - DBMS: Manages and controls the database
- - Base relation: A table stored in a database
- - Tuple: A row in a database
- - Attribute: A column in a database
- - Primary key: Columns that make each row unique
- - View: Virtual relation defined by the database to control subjects view
- - Foreign key: Attribute of one table that is the primary key of another table
- - Cell: Intersection of a row and column
- - Schema: Holds data that describes a database
- - Data dictionary: Central repository of data element and their relationships.
- - Cardinality: The number of rows in the relation.
- - Degree: The number of columns in the relation.
- - Domain: Is a set of allowable values that an attribute can take

- **Database models**

- Relational data model
 - ♦ Uses attributes (columns)
 - ♦ Uses tuples (rows)
 - ♦ Use A primary key
 - Is a unique identifier in the table that point to an individual tuple or row in the table
 - Is a field that links all the data within a record to a corresponding value
 - Is a subset of candidate keys within a table
 - Each table must have a UNIQUE Primary Key
 - ♦ This model is composed of two dimensional tables
 - ♦ Each table contains unique rows, columns, and cells

- ◆ This model is the most used today
- Hierarchical data model
 - ◆ Combines records and fields that are related in a logical tree structure
 - ◆ The tree structure contain branches and each branch contain leaves (Data fields)
 - ◆ The parents Can have one child, many children, no children.
 - ◆ Are useful for mapping one-to-many relationships
 - ◆ This model is commonly used today
- Distributed data model
 - ◆ Has data stored in more than one database, but it is logically connected
 - ◆ Enable different databases to be managed by different administrators
 - ◆ one person or group must manage the entire logical database
 - ◆ Are useful for mapping Many-to-Many relationships
- Relational database components
 - ◆ Include DDL / Data Definition Language
 - DLL defines the strcuture and schema of the database
 - Define the Schema
 - Describe the type of data that will be held and manipulated and their properties
 - It defines
 - The structure of the database
 - The access operations
 - The intergrity procedures
 - The Schema Contain
 - Tables
 - Views
 - Indexes
 - Procedures
 - Functions
 - Triggers
 - ◆ DML / Data Manipulation Language
 - Contain All the commands that enable a user to:
 - view
 - manipulate
 - User the database
 - Examines and manipulates the data within the database
 - ◆ Include DCL Dat Control Language
 - Which define the internal organization of the database and an ad hoc query
 - ◆ Include QL / Query Language
 - Enables users to make requests to the database
 - Allow users to make queries and access the data within the database
 - ◆ Report Generator
 - Produces printouts of data in a userdefined manner
- **Data dictionary**
 - Is a central repository of data elements and their relationships.
 - Is a collection of data elements, schema objects and reference keys
- **Keys**
 - Primary key
 - ◆ Is a unique identifier in the table that point to an individual tuple or row in the table
 - ◆ Is a field that links all the data within a record to a corresponding value
 - ◆ Is a subset of candidate keys within a table
 - ◆ Each table must have a UNIQUE Primary Key
 - Foreign key
 - ◆ An attribute (column) in one relation that has values matching the primary key in another

relation.

- **Integrity**
 - Concurrency problems
 - ♦ Making sure that different subjects receive the most up-to-date information
 - Database software performs two main types of integrity
 - ♦ Semantic
 - ♦ Referential
 - Semantic integrity
 - ♦ Makes sure that structural and semantic rules are enforced
 - ♦ These rules pertain to data types, logical values, uniqueness constraints and operations that could adversely affect the structure of the database
 - Referential integrity
 - ♦ Mechanism would ensure that no record would contain a reference to a primary key of a nonexisting record or a NULL value
 - ♦ This guarantees that the tuple is uniquely identified by primary key values
 - Entity integrity
 - ♦ If an attribute is NULL
 - Rollback
 - ♦ Is a statement that ends a current transaction and cancels all other changes to the database
 - Commit
 - ♦ Terminates a transaction and executes all changes that were just made by the user
 - Checkpoint
 - ♦ Are used to make sure that if a system failure occurs or if an error is detected, the user can always return to a point in time before the system crashed
 - ♦ Checkpoint are easy to implement within the database.
 - ♦ Checkpoint should be implemented in balanced way (Not too many not enough)
- **Database security issues**
 - Aggregation
 - ♦ Happen when a user does not have the clearance or permission to access specific information, but she does have the permission to access components of this information, then figure out the rest and obtain restricted information.
 - ♦ Aggregation is the act of combining information from separate sources
 - ♦ The combination of the information forms new information
 - ♦ The combined information may have a sensitivity greater than the individual parts
 - Inference
 - ♦ Happens when a subject deduces information that is restricted from data he has access to
 - ♦ This is seen when data at a lower security level indirectly portrays data at a higher level
 - ♦ Is the ability to derive information that is not explicitly available
 - ♦ Can be prevented
 - Cell suppression
 - Is a technique used to hide or not show specific cells that contain information that could be used in inference attacks
 - Partitioning
 - Involves dividing the database into different parts
 - makes it much harder for an unauthorized individual to find connecting pieces of data that can be brought together and other information that can be deduced or uncovered
 - Noise and perturbation
 - Is a technique of inserting bogus information in the hope of misdirecting an attacker or confusing the matter enough that the actual attack will not be fruitful
 - Content-dependents access control
 - ♦ Database security looks at the content of a file when it makes an access decision
 - ♦ This increases processing overhead, but it provides higher granular control
 - Database views

- ♦ Permit one group or a specific user to see certain information, while restricting another group from viewing it altogether
- ♦ Databases can employ (DAC) Discretionary Access Control
- ♦ Database can employ (MAC) Mandatory Access Control
- Polyinstantiation
 - ♦ Enables a relation to contain multiple tuples with the same primary keys with each instance distinguished by a security level
 - ♦ Is the process of interactively producing more detailed versions of objects by populating variables with values or other variables
 - ♦ Polyinstantiation creates two versions of the same object
- OLTP / On Line Transaction Processing
 - ♦ Provides mechanisms that watch for problems and deal with them appropriately when they do occur.
 - ♦ Is usually used when databases are clustered to provide fault tolerance
 - ♦ Will make sure that a transaction is not complete until all databases receive and reflect a change. This is a Two-phase commit service
 - ♦ OLTP will load balance incoming requests if it is necessary
- Data warehousing
 - ♦ Combines data from multiple databases into a large database
 - ♦ the purpose of a fuller extent of information retrieval and data analysis
- Data mining
 - ♦ Is the process of messaging the data held in the data warehouse into more useful information
 - ♦ Data mining tools are used to find an association in data to produce Metadata
 - ♦ Metadata
 - Data produced by data mining tools to find associations and correlations
- OODB / Object-Oriented Data Bases
 - ♦ Have the characteristics of ease of reusing code and analysis
 - ♦ reduced maintenance and an easier transition from analysis of the problem to design and implementation
 - ♦ Its main disadvantages are a steep learning curve and high overhead of hardware and software required for development and operation.
- Object-Relational Databases
 - ♦ Combines the attributes of object-oriented and relational technologies

❖ System Development Controls

➤ System Life Cycle Phases

- **Project initiation**
 - - Conception of project definition
 - - Proposal and initial study
- **Functional design analysis and planning**
 - - Requirements uncovered and defined
 - - System environment specification determined
- **System design specifications**
 - - Functional design review
 - - Functionality broken down
 - - Detailed planning put into place
 - - Code design
- **Software development**
 - - Developing and programming software
- **Installation / implementation**
 - - Product installation
 - - Testing and auditing

- -Accreditation is the final phase in system dev life cycle
- **Operational/maintenance**
 - - Product changes, fixes and minor modifications
- **Disposal / Revision and replacement**
 - - Modifying the product with revisions or replacing it altogether
- **The Waterfall Model**
 - - System requirements
 - - Software requirements
 - - Analysis
 - - Program design
 - - Coding
 - - Testing
 - - Operations & Maintenance
- **Modified Waterfall Model incorporating V&V**
 - - System feasibility -> validation
 - - Software plans & requirements -> validation
 - - Product design -> verification
 - - Detailed design -> verification
 - - Coding -> unit test
 - - Integration Product -> verification
 - - Implementation -> system test
 - - Operations & Maintenance -> revalidation
- **Security concerns**
 - Security should be addressed in each phase of system development
 - Security should not be addressed at the end of development because of the added cost, time, effort and lack of functionality.
 - Separation of duties should be practiced in roles, environments and functionality pertaining to development of a product
 - A programmer should not have direct access to code in production
 - Certification deals with testing and assessing the security mechanism in a system
 - Accreditation pertains to the management formally accepting the system and its security level
 - Changes must be authorized, tested and recorded
 - The changes must not affect the security level of the system or its capability to enforce the security policy.
- **Change control sub-phases**
 - - Request control
 - - Change control
 - - Release control
 - - Report change to management
- **Change control process**
 - - Make a formal request of change
 - - Analyze the request
 - - Develop the implementation strategy
 - - Calculate the costs of this implementation
 - - Review any security implications

- - **Record the change request**
- - **Submit the change request for approval**
- - **Develop the change**
 - - Recode segments of the product and add or subtract functionality.
 - - Link these changes in the code to the formal change control request
 - - Submit software for testing and quality approval
 - - Repeat until quality is adequate
 - - Make version changes
- **Configuration management**
 - - **Configuration identification**
 - - **Configuration control**
 - - **Configuration status accounting**
 - - **Configuration audit**
 - **CMM / Software Capability Maturity Model**
 - - Level 1
 - ◆ Initiating - Competent people and heroics; processes are informal and ad hoc
 - - Level 2
 - ◆ Repeatable - Project management processes; project management practices are institutionalized
 - - Level 3
 - ◆ Defined - Engineering processes and organizational support; technical practices are integrated with management practices institutionalized
 - - Level 4
 - ◆ Managed - Product and process improvement; product and process are quantitatively controlled
 - - Level 5
 - ◆ Optimized - Continuous process improvement; process improvement is institutionalized

❖ **Application Development Methodology**

- **Structured analysis approach**
 - **Looks at all objects and subjects of an application and maps the interrelationships, communication paths and inheritance properties**
- **Types of languages**
 - **Machine Code language (MCL)**
 - Is in a form that the computer and processor can understand and work with directly
 - **Assembly language**
 - Cannot be understood directly by the system and must be processed, results MCL
 - **High-level language**
 - Cannot be understood directly by the system and must be processed results MCL
- **Programs**
 - **There are different types of programs**
 - **Interpreters**
 - Translate one command at a time during execution
 - Interpreted programs - Have instructions that are read and interpreted by a program one instruction at a time
 - **Compilers**
 - Translate large sections of code at a time
 - Compiled programs - Are written in a high-level language and turned into machinereadable format by a program called compiler

- **Assembler**
 - Translate assembly language into machine language

➤ **OOP / Object-Oriented Programming**

- **OOP provides functional independence**
- **OOP is a class of programming languages and techniques**
- **In OOP each object can be reused**
- **Works with classes and objects within those classes**
- **Emphasizes the employment of objects and methods rather than types or transformations as in other programming approaches**
- **Once the class is defined, the attributes can be reused for each new member**
- **The object encapsulate the attribute values**
- **An object can have a shared portion**
 - The interface that enables it to interact with other components
- **An object can have a private portion**
 - How it actually works and performs the requested operations
- **Messages enter through the interface to specify the requested operation or method to be performed.**
- **Information hiding**
 - There is no need for other components to know how each object works internally.
 - Data hiding is provided by Encapsulation that protect the object's private data
- **Abstraction**
 - Is the capability to suppress unnecessary details so that the important, inherent properties can be examined and reviewed
- **Messaging**
 - Can happen in several different ways
 - Two objects can have
 - ♦ A single connection One-to-One
 - ♦ Multiple connection (One-to-Many)
 - ♦ Mandatory connection
 - ♦ Optional Connection
 - This will ensure that sensitive data cannot pass to objects of a lower security level
- **What should be developed for every object ?**
 - Object name
 - Attribute descriptions
 - ♦ Attribute name
 - ♦ Attribute content
 - ♦ Attribute data type
 - External input object
 - External output from object
 - Operation descriptions
 - ♦ Operation name
 - ♦ Operation interface description
 - ♦ Operation processing description
 - ♦ Performance issues
 - ♦ Restrictions and limitations
 - Instance connections
 - Message connections
- **Phases of object-orientation**
 - OORA / Object-Oriented Requirements Analysis

- ◆ Defines classes of objects and their interactions
- OOA / Object-Oriented Analysis
 - ◆ Structured analysis approach
 - ◆ Is the process of classifying objects that will be appropriate for a solution
 - ◆ In terms of object-oriented concepts, understanding and modelling a particular problem within a problem domain.
- DA / Domain Analysis
 - ◆ Seeks to identify the classes and objects that are common to all applications within a given domain
- OOD / Object-Oriented Design
 - ◆ OOB is a design method where a system is modeled as a collection of objects
 - ◆ OOD enable the developer to indicate the objects that will derive from each class and how these objects will work together
 - ◆ OOD creates a representation of a real-world problem and maps it to a software solution
 - ◆ The results of OOD is a design that modularizes data and procedures
 - ◆ The design interconnects data objects and processing operations
 - ◆ OOD allow for
 - Abstraction
 - Information hiding
 - Modularity
 - Is accomplished by using objects, applets and agents
- **Features of OOP**
 - Encapsulation
 - ◆ Hides internal data and operations.
 - ◆ which means that this information is packaged under one name and can be reused as one entity by other objects.
 - Polymorphism
 - ◆ Makes copies of objects and makes changes to those copies.
 - ◆ Different objects respond to the same command or message in different ways
 - Polyinstantiation
 - ◆ Multiple distinct differences between data within objects to discourage lower-level subjects from learning information at a higher-level of security.
 - Inheritance
 - ◆ Shares properties and attributes.
 - ◆ Objects derive functionality and data automatically from another object
 - Multiple inheritance
 - ◆ A class inherits the behavioural characteristics of more than one parent class.
 - Delegation
 - ◆ Forwarding of a request by an object to another object or delegate.
 - ◆ This forwarding is necessitated because the object receiving the request does not have a method to service the request

❖ Data Modelling

➤ Data modelling

- **Considers data independently of the way that the data is processed and the components that process the data**

➤ Data Structures

- **Data Structure**
 - Is a representation of the logical relationship between elements of data
- **Cohesive**
 - A cohesive module can perform a single task with little or no help from other modules
 - - Low Cohesion: Scatter brained, does several tasks.

- - High Cohesion: Focused on one task.
 - The best programming uses the most cohesive modules possible, but because different modules need to pass data and communicate, they usually cannot be totally cohesive
- **Coupling**
 - Is a measure of interconnection among modules in an application
 - - Low Coupling: Promotes module independence.
 - - High Coupling: Depend on other modules
 - The lower the coupling, the better the software design, because it promote module independence
 - The more independent a component is, the less complex the application is and the easier it is to modify and troubleshoot.
- **Modular Code**
 - Modules should be self-contained and perform a single logical function which is cohison
 - Modules should not drastically affect each other, which is LOW COUPLING
- **OMA / Object Management Architecture**
 - **OMA provides a high-level overview of a complete distributed environment**
 - **OMA contain two main parts**
 - ORB
 - Application oriented components
 - **ORB / Object Request Brokers**
 - Manages all communication between components and enables them to interact in a heterogeneous and distributed environment
 - ORB is a platform independant that provide interoperability
 - ORB is the middleware that establishes client/server relationship between objects
 - **CORBA / Common Object Request Broker Architecture**
 - Provides interoperability among the vast array of different software, platforms and hardware in environments
 - Enables applications to communicate with one another no matter where the application is located or who developed it
 - To implement this compatible interchange, a user develops a small amount of initial code and an Interface Definition Language (IDL) file.
 - Provides standard interface definitions between OMG compliant objects
 - OMG Object Management Group developed a model for the use of these different services in an environment
 - **COM / Common Object Model**
 - Supports the exchange of objects among programs
 - Defines how components interact and provides an architecture for (IPC) Interprocess Communication
 - Enables applications to use components on the same system
 - **DCOM / Distributed Common Object Model**
 - Defines the standard for sharing objects in a networked environment
 - Uses a globally unique identifier, GUID, to uniquely identify users, resources and components within an environment
 - Defines how components interact and provides an architecture for DISTRIBUTED (IPC) Interprocess Communication
 - Enables applications to use components on reside in different system
 - DCOM support process-to-process communication across networks
 - DCOM work as Middleware that enable distributed processing
 - **Type of Middleware**

- ORB / Object Request Brokers
- DCOM / Distributed Common Object Model
- ODBC / Open Database Connectivity
 - ♦ Provides a standard SQL dialect that can be used to access many types of relational db
 - ♦ Is the middleware between applications and databases
 - ♦ The ODBC driver manager decides which drivers need to be used for communication
- MOM Message Oriented Middleware
- The RPC function collects the arguments and commands from the sending program and prepares them for transmission over the network
- **DDE / Dynamic Data Exchange**
 - Enables different applications to share data by providing IPC
 - mechanism that enables direct conversation between two applications
 - Spreadsheet can use DDE link to another program that tracks stock prices
- **DCE / Distributed Computing Environment**
 - Is a set of management services with a communication layer based on RPC
 - Sits on top of the network layer and provides services to the applications above it.
 - Uses universal unique identifier, UUID, to uniquely identify users, resources
 - The DFS / Distributed File Services provides a single integrated file system that all DCE users can use to share files
- **OLE Object Linking and Embedding**
 - Provides a way for objects to be shared on a local computer
 - OLE uses COM as its foundation base
 - Linking - Is the capability for one program to call another program
 - Embedding - Is the capability to put data inside a foreign program or document
- **Expert systems / knowledge based systems**
 - Use artificial intelligence / emulate human knowledge to solve problems
 - Is a computer program containing a knowledge base and set of algorithm and rules used to infer new facts from knowledge and incoming data
 - - Rule-based programming: Is a common way of developing expert systems.
 - - Pattern matching: Based on if-then logic units.
 - - Inference engine: A mechanism that automatically matches facts against patterns and determines which rules are applicable
- **Artificial Neural Networks**
 - Is an electronic model based on the neural structure of the brain
 - Replicate the basic functions of neurons and their circuitry to solve problems

❖ Method of attack

- **Malicious Code**
 - Viruses, worms, trojan horses, logic bombs
 - Can be detected by
 - - File size increase
 - - Many unexpected disk accesses
 - - Change in update or modified timestamps
 - **Java**
 - Is a platform independent because it creates intermediate code, bytecode, which is not processor specific
 - The Java Virtual Machine then converts the bytecode to machine code
 - Java applets use a security scheme that employs a sandbox to limit the applet's access

to certain specific areas within the user's system and protects them from malicious or poorly written applets

- **ActiveX**
 - Practices security by informing the user where the program came from.
 - Uses authenticode that relies on digital certificates and trusting certificate authorities
- **Virus**
 - Searches out other programs and infects them by embedding a copy of itself
 - When the infected program executes, the embedded virus is executed which propagates the infection
 - Types
 - ♦ Boot sector infectors
 - Move data within the boot sector or overwrite the sector with new information
 - ♦ Stealth virus
 - Hides the modifications that it has made to files or boot records.
 - ♦ Polymorphic virus
 - Produces varied but operational copies of itself.
 - ♦ Multitpart virus
 - Infects both the boot sector of a hard drive and executable files.
 - ♦ Self-garbling virus
 - Attempts to hide from antivirus software by garbling its own code
 - As the virus spreads, it changes the way its code is encoded.
 - ♦ Macros (Word)
- **Worm**
 - They can reproduce on their own with no need for a host application
 - they are self-contained programs
- **Logic bomb**
 - when a certain event happens
- **Trojan horse**
 - Is a program disguised as another program

➤ Denial Of Service

- **An attack consuming the victim's bandwidth or resources**
- **cause the system to crash or stop processing other packet**
- **Smurf**
 - Requires three players: the attacker, the victim and the amplifying network
 - The attacker spoofs, or changes the source IP address in a packet header, to make an ICMP ECHO packet seem as though it originated at the victim's system.
 - This ICMP ECHO message is broadcasted to the amplifying network, which will reply to the message in full force
 - The victims system and victim's network is overwhelmed
- **Fraggle**
 - Uses UDP as its weapon of choice
 - The attacker broadcasts a spoofed UDP packet to the amplifying network, which in turn replies to the victim's system
- **SYN Flood**
 - Continually sending the victim SYN messages with spoofed packets
 - The victim will commit the necessary resources to set up this communication socket and it will send its SYN/ACK message waiting for the ACK message in return
- **Teardrop**
 - An attacker sending very small packets that would cause a system to freeze or reboot
 - Causes by the fact that some systems make sure that packets are not too large, but do

not check to see if a packet is too small.

- **DDoS / Distributed Denial of Service**
 - Is a logical extension of the DoS
 - The attacker creates master controllers that can in turn control slaves / zombie machines
- **DNS DoS Attacks**
 - A record at a DNS server is replaced with a new record pointing at a fake/false IP address
 - Cache poisoning - The attacker inserting data into the cache of the server instead of replacing the actual records