

**18MCA44E SOFTWARE TESTING**  
**UNIT IV – ACCESSING CLIENT-SERVER AND LAN RISKS**

FACULTY

Dr. K. ARTHI MCA, M.Phil., Ph.D.,

Assistant Professor,

Postgraduate Department of Computer Applications,

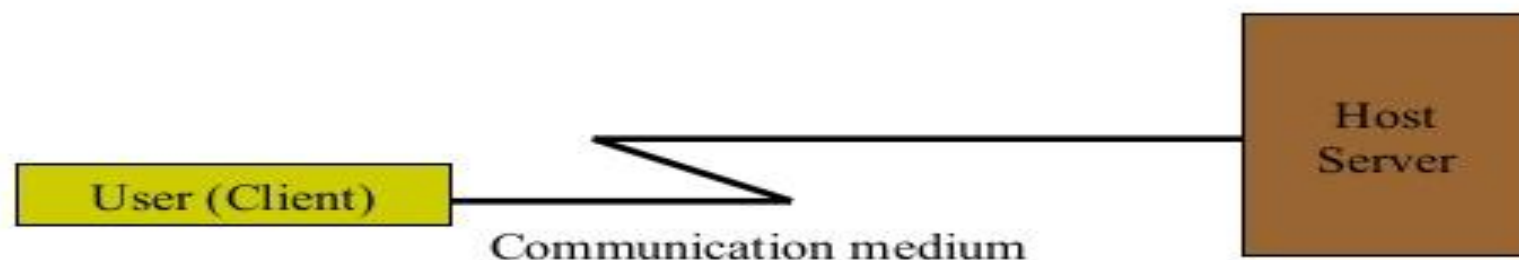
Government Arts College (Autonomous),

Coimbatore-641018.

# INTRODUCTION

---

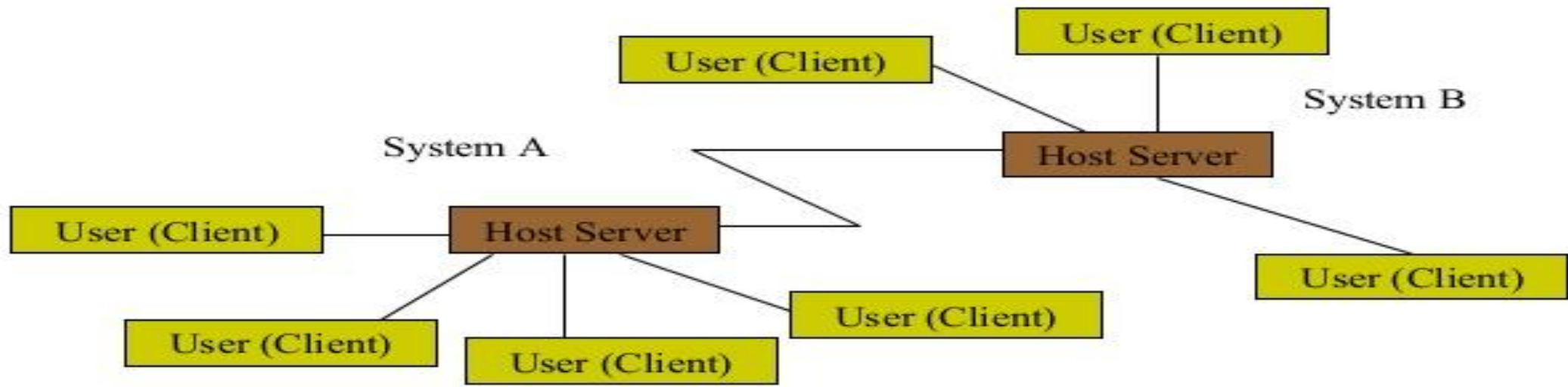
- Network is two devices connected across some medium by hardware and software that complete the communications (simple definition of network).



**Simple View of Network**

# Introduction

- A network is normally not just single client to a single server; typically many clients interact with many servers.





# Network Security Issues

---

Network have security problems for the following reasons:

- **Sharing** – resources and workload sharing
- **Complexity of system**
- **Unknown parameter** – expandability of a network also implies uncertainty about the network boundary
- **Many points of attack** – file may past through many host before reaching the destination
- **Anonymity** – attacker can mount an attack with touching the system
- **Unknown path** – there may be many path from one host to another.

# Introduction

- Although client/server architecture is the most popular and widely used computing environment, it is the most vulnerable to breaches in security.
- Why?
  - Distribution of services between client server which render to damage, fraud and misuse.
- Security considerations must be in place.

# Client and User Security

- Clients connect to servers and these connections, if left open or not secured, provide entry points for hackers and other intruders that may use data for evil purposes.
- The distribution of services in client/ server increases the susceptibility of these systems to damage from viruses, fraud, physical damage and misuse than in any centralised computer system.

# Security Considerations

- Security consideration must include:
  - the host systems
  - personal computers (PCs)
  - local area networks (LANs)
  - global wide area networks (WANs)
  - Users.

# The client

- PC desktop as the client machine can easily connect to server and hence pose a great threat to organisational data.
- Located in an open place for users and intruders.
- Require physical protection.
  - disk drive locks,
  - diskless workstations to prevent the loading of unauthorised software and viruses



# The Network

- Intruders can use network to break into computer systems and their various resources
  - Can sniff the network to read packet of information such as password, company details, etc.
- Use data encryption

# Servers

- The should be protected with the level of password security applicable to the business.
- Encrypt database contents cryptogram or advanced DES (Data Encryption Standard)
- To avoid 'trial and error' login attacks to database, servers impose long delay to respond to user.

# Users

- The easiest way to gain illegal entry to the system is by obtaining a valid users ID and password. Password can be discovered when:
  - User chose short password or can easily be guessed.
  - User keeps a list of password in a desk drawer.
  - Share password with another users.
  - Attacker pretend company IT staff and request password to fix unnamed problem.

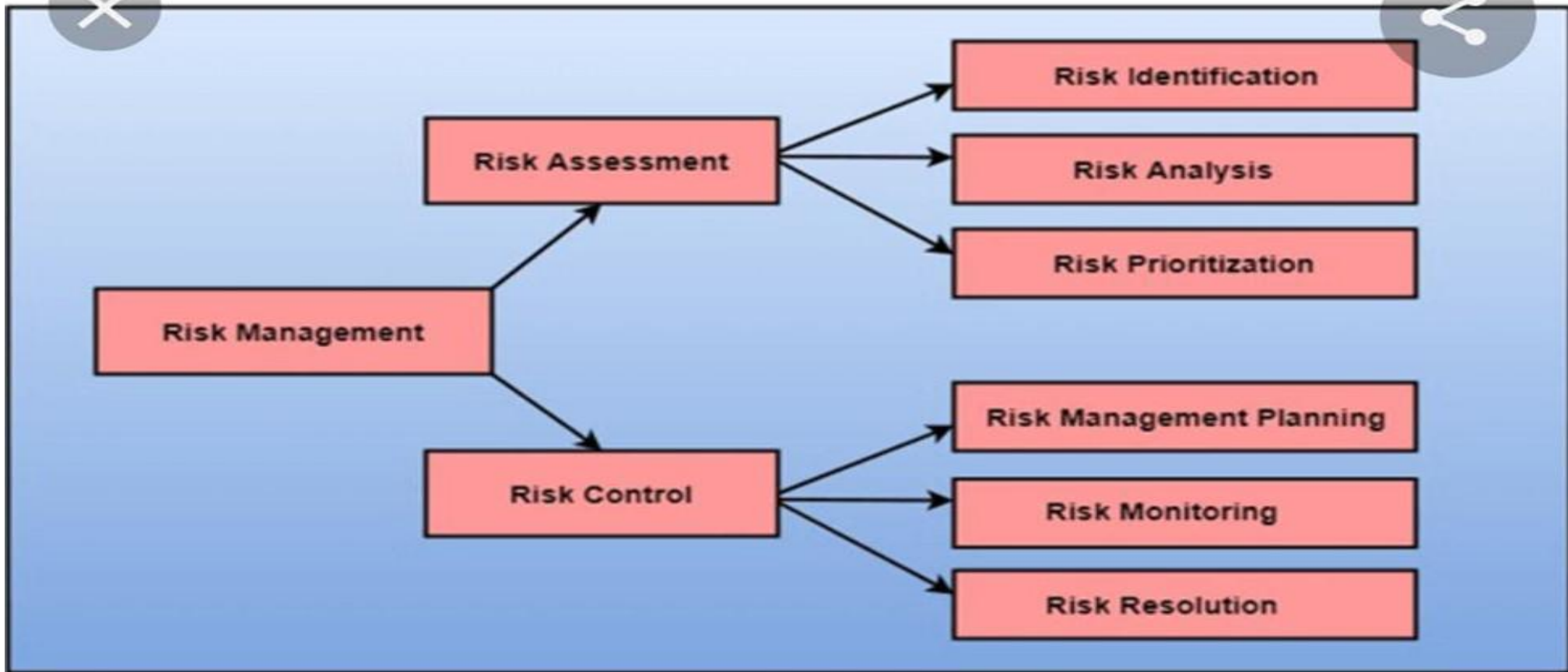
# Client/Server Risks

- Client/Server Development Risks
  - Skills of C/Server Development Risks
  - Rapid Application development Risk.
- Work Station Risk
  - Insecurity of PC on the network
    - Add third party security products.
    - PC should be used to work on select application records for few period of time.
- The Network Wire Risk
  - Packet data can be read during the communication process.
    - Use encryption methods to encrypt data (Userids and password)

# Client/Server Risks

- The DBMS Risk
  - C/S computing is based largely on the Database Management Software that supports the applications.
    - Security professional needs to identify System Manager for DBMS “System” or “Sys” for Oracle.
  - More read on the handout (CS\_Security.doc) available on web.

## Risk Management Activities



# A TESTING STRATEGY FOR A RAPID PROTOTYPING

## Definition:

- ❖ Rapid prototyping is an agile strategy used throughout the product development process.
- ❖ With this approach, 3-dimensional prototypes of a product or feature are created and tested in an attempt to optimize characteristics like shape, size, and overall usability.

## The Rapid Prototyping Process:

- ❖ The rapid prototyping process involves three simple steps:

### 1. Prototyping :

- ❖ The team creates an initial prototype. This is a visual representation of the design specifications as set out in the requirements document.
- ❖ The prototype may be either low-fidelity or high-fidelity (more to come on high- and low-fidelity prototypes) and may be interactive or non-interactive.

### 2. Feedback :

The creators share the prototype with other team members, stakeholders, and focus groups made up the intended end-users. Everyone evaluates both the design and usability before submitting feedback.



### **3. Improvement :**

- ❖ The feedback is used to create a new iteration of the prototype.
- ❖ The process then cycles round to Step 2 for further feedback.
- ❖ This continues until there are no more changes or a specified cut-off is reached (either a date or number of iterations).

## **Benefits of Rapid Prototyping:**

- ❖ Quick Iterations Generate Progress.
- ❖ Focus on Improving Core Features.
- ❖ Focuses Design on End-User.
- ❖ Encourages Collaboration and Feedback.
- ❖ Reduces Project Risk.
- ❖ Reduced Time to Market.

# TESTING TECHNIQUES

## **SOFTWARE TESTING:**

- ❖ Testing is a process of executing a program with the aim of finding error.
- ❖ To make our software perform well it should be error free.
- ❖ If testing is done successfully it will remove all the errors from the software.

## **TESTING TECHNIQUES:**

- ❖ Unit Testing
- ❖ Integration Testing
- ❖ Regression Testing
- ❖ Smoke Testing
- ❖ Alpha Testing
- ❖ Beta Testing
- ❖ System Testing
- ❖ Performance Testing

## Unit Testing:

- ❖ It focuses on smallest unit of software design.
- ❖ In this we test an individual unit or group of inter related units.
- ❖ It is often done by programmer by using sample input and observing its corresponding outputs.

## Integration Testing:

- ❖ The objective is to take unit tested components and build a program structure that has been dictated by design.
- ❖ Integration testing is testing in which a group of components are combined to produce output.
- ❖ The four common types of **Integration testing** are:
  - ❖ Top down
  - ❖ Sandwich
  - ❖ Bottom up
  - ❖ Big-Bang

## **Regression Testing**

- ❖ Every time new module is added leads to changes in program.
- ❖ This type of testing make sure that whole component works properly even after adding components to the complete program.

## **Smoke Testing:**

- ❖ This test is done to make sure that software under testing is ready or stable for further testing.
- ❖ It is called smoke test as testing initial pass is done to check if it did not catch the fire or smoked in the initial switch on.

## **Alpha Testing:**

- ❖ This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers.
- ❖ It is typically done by QA people.

## **Beta Testing**

- ❖ The beta test is conducted at one or more customer sites by the end-user of the software.
- ❖ This version is released for the limited number of users for testing in real time environment.

## **System Testing:**

- ❖ In this software is tested such that it works fine for different operating system.
- ❖ It is covered under the black box testing technique.
- ❖ In this we just focus on required input and output without focusing on internal working.

## **Performance Testing:**

- ❖ It is designed to test the run-time performance of software within the context of an integrated system.
- ❖ It is used to test speed and effectiveness of program.

# TESTING TOOLS

## **Testing tools:**

- ❖ Tools from a software testing context can be defined as a product that supports one or more test activities right from planning, requirements, creating a built, test execution, defect and test analysis.

## **Classification of tools:**

- ❖ The purpose of the tools
- ❖ The activities that are support with in the tools
- ❖ The type/level of testing it support
- ❖ The kind of licensing (open source, freeware, commercial)
- ❖ The technology used



## Tool type:

- ❖ Test management tool
- ❖ Configuration management tool
- ❖ Static analysis tool
- ❖ Test data preparation tools
- ❖ Test execution tools
- ❖ Test comparators
- ❖ Coverage measurement tools
- ❖ Performance testing tools
- ❖ Project planning and tracking tools
- ❖ Incident management tools

## **Test management tools:**

- ❖ Management of test
- ❖ Scheduling of test to be executed
- ❖ Management of testing activities
- ❖ Traceability of tests
- ❖ Linking to incident reports
- ❖ Preparing progress report

## **Configuration management tools:**

- ❖ Used for control testing
- ❖ Storing information about versions and builds of the software and test ware
- ❖ Static testing:
- ❖ Storing and sorting comments
- ❖ Communication comments to be relevant tool
- ❖ Monitoring the review status

### **Static analysis test:**

- ❖ Mostly used by developers
- ❖ Enforce coding standards
- ❖ Code understanding

### **Modeling tools:**

- ❖ Mostly used by developers helps designing of the software
- ❖ Checking consistency
- ❖ Identify defects as early as possible

### **Test design tools:**

- ❖ Screen scraper
- ❖ Generating expected result. If an oracle is available to the tool

### **Test data preparation tools:**

- ❖ Extract selected data records from files or databases
- ❖ Construct a large number of similar records

## **Test execution tools:**

- ❖ Also known as 'captures/playback' tools. Capture/reply tools
- ❖ Use scripting language to drive the tool
- ❖ Best used for difficult to maintain
- ❖ Test harness/unit test framework tools:
- ❖ Mostly used by developers create stub and drivers
- ❖ Support for debugging(the framework tools)

## **Test comparators:**

- ❖ Test comparators help to automate aspects of that comparison
- ❖ Tools support dynamic comparison i.e. when execution is going on
- ❖ Comparing large files

## **Coverage measurement tools:**

- ❖ Identifying coverage items
- ❖ Calculating the percentage of coverage items
- ❖ Generating stubs and drivers

## **Security tools:**

- ❖ Identifying viruses
- ❖ Probing for open parts or other extremely visible point of attacks
- ❖ Identifying weakness in password files and passwords

## **Dynamic analysis tools:**

- ❖ Detecting memory leads
- ❖ Identifying pointer arithmetic errors such as null pointers
- ❖ Dead links present an code i.e. links

## Performance testing, load testing and stress testing tools:

- ❖ Performance testing: it is performed to evaluate the performance of components of a particular system in a specific situation
- ❖ Load testing : it is done by constantly increase the load on the application under test till the time it reaches the threshold limit
- ❖ Stress tools : it is done to evaluate the application behavior beyond normal peak load conditions
- ❖ Generating a load on the system to be tested
- ❖ Measuring average response times
- ❖ Producing graphs or charts of response over time

## Monitoring tools:

- ❖ Continuously keep track of the status of the system in use servers, databases, security , performance, website and internet usages and applications
- ❖ Identifying problems and sending an alert message to the network administrators
- ❖ Monitoring the number of users on a network
- ❖ Monitoring network traffic

**THANK YOU**

**This content is taken from the text books and reference books  
prescribed in the syllabus.**