Secure Software Design and Engineering
(CY-321)

Secure Design Requirements

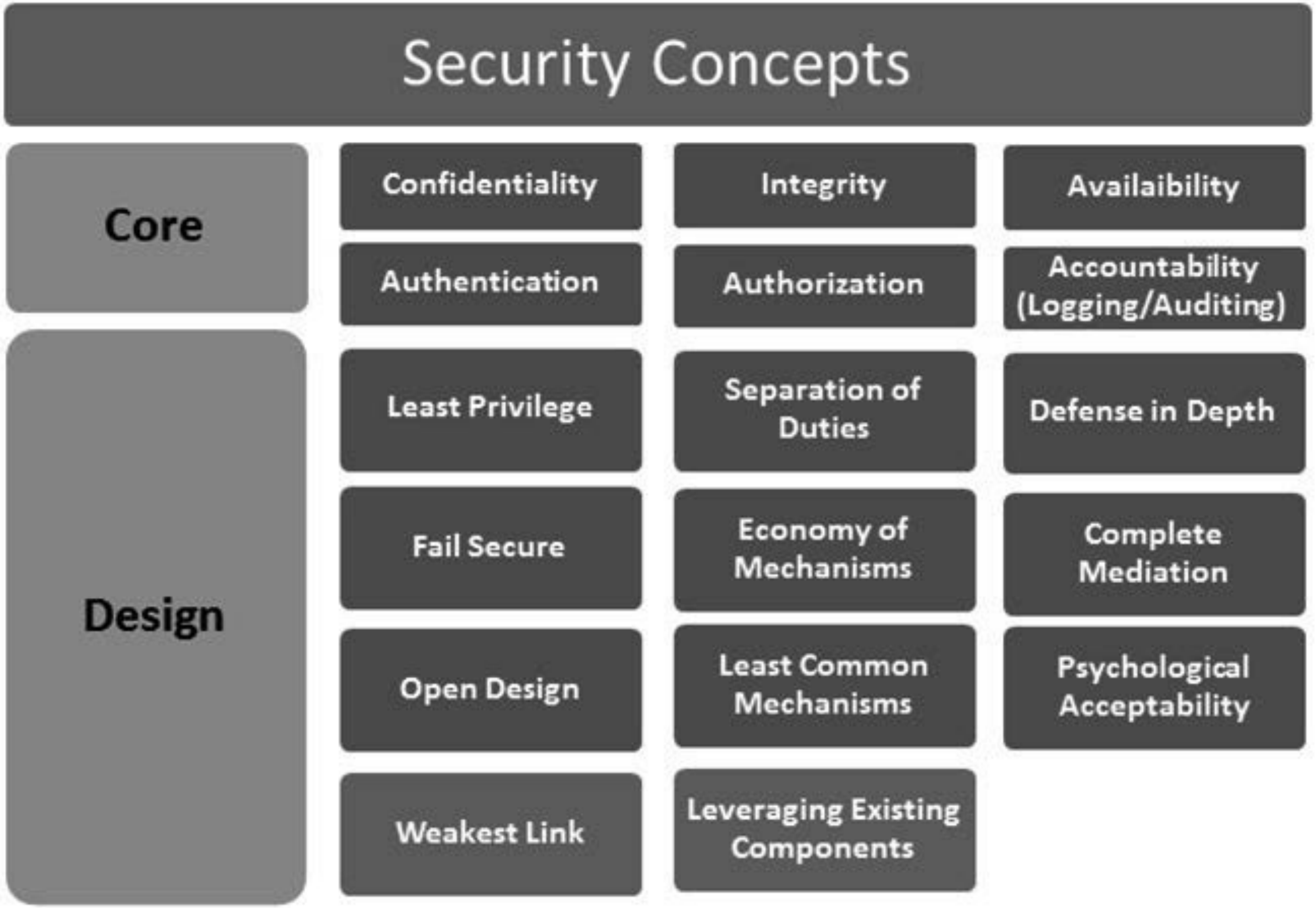
Dr. Zubair Ahmad

A kind Reminder

- **Attendance?**

  - Active Attendance
  - **Dead Bodies.**
  - **Active Minds**
  - Mobiles in hands -> Mark as absent
  - 80% mandatory

Secure Design Requirements



Security Concepts

Core

Confidentiality

Integrity

Availaibility

Authentication

Authorization

Accountability (Logging/Auditing)

Design

Least Privilege

Separation of Duties

Defense in Depth

Fail Secure

Economy of Mechanisms

Complete Mediation

Open Design

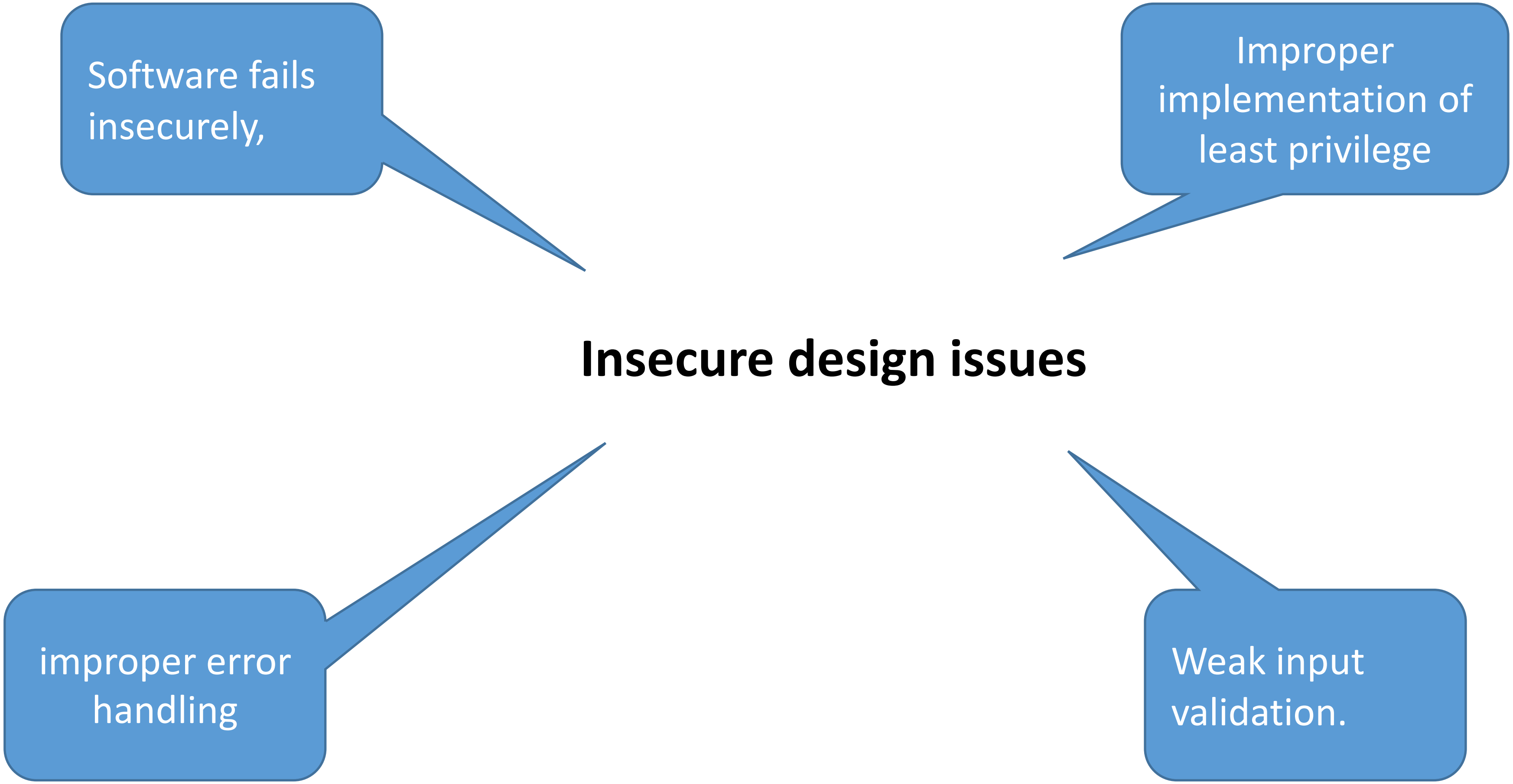
Least Common Mechanisms

Psychological Acceptability

Weakest Link

Leveraging Existing Components

Secure Design Requirements



Software fails insecurely,

Improper implementation of least privilege

Insecure design issues

improper error handling

Weak input validation.

Secure Design Requirements

Least Privilege

Need-to-know

limits the disclosure of sensitive information to only those who have been authorized to receive

Clearance level classification

Modular programming

Secure Design Requirements

Least Privilege

Modular programming

A software design approach that breaks a system into **smaller, reusable modules** instead of writing one large program

Each module performs a **specific function** and can be tested, updated, or replaced independently

High Cohesion

Loose Coupling

Secure Design Requirements

Modular programming

High Cohesion

Each module **does one thing well** and is **strongly related to its purpose**

Example??

A module for **user authentication** should handle login/logout but NOT database management.

Secure Design Requirements

Modular programming

Loose Coupling

Modules **interact with each other as little as possible** and communicate through well-defined interfaces

Example??

A payment processing module should work independently without needing to know how the shopping cart module is implemented

Secure Design Requirements

Separation of Duties

Software functionality into two or more conditions, all of which need to be satisfied before an operation can be completed

Software **Deployment**, **Operations**, **Maintenance**, and **Disposal** chapter.

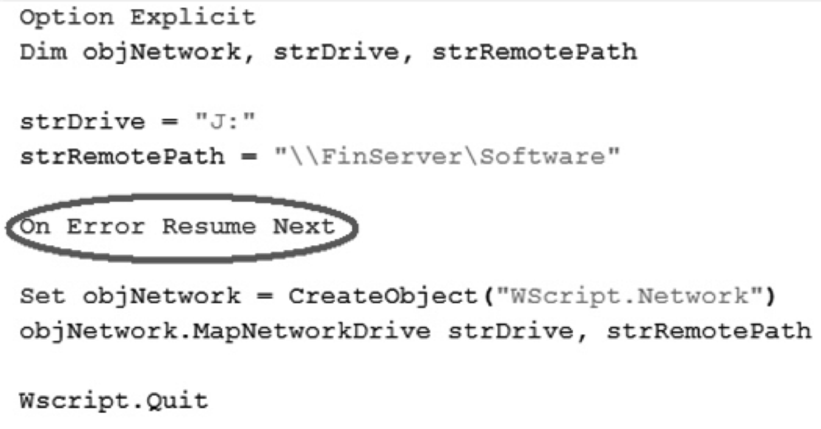
Secure Design Requirements

Defense in Depth → **Not putting all the eggs in one basket**

Use of Input Validation & Prepared Statements to Prevent Injection Attacks

Defending Against Cross-Site Scripting (XSS) with Output Encoding & Validation

Security Zones for Access Control



```
Option Explicit
Dim objNetwork, strDrive, strRemotePath

strDrive = "J:"
strRemotePath = "\\FinServer\Software"

On Error Resume Next

Set objNetwork = CreateObject("WScript.Network")
objNetwork.MapNetworkDrive strDrive, strRemotePath

Wscript.Quit
```

nts

when attacked and is rapidly *recoverable* into a normal business

The user is denied access by default and the account is locked out after the maximum number (clipping level) of access attempts is tried

Errors and exceptions are explicitly handled and the error messages are non-verbose in nature

Not designing the software to ignore the error and resume next operation

Secure Design Requirements

Fail Secure

```
Option Explicit
Dim objNetwork, strDrive,
strRemotePath

strDrive = "J:"
strRemotePath = "\\FinServer\Software"

On Error Resume Next

Set objNetwork =
CreateObject("WScript.Network")
objNetwork.MapNetworkDrive strDrive,
strRemotePath

Wscript.Quit
```

Secure Design Requirements

Fail Secure

```
Option Explicit
Dim objNetwork, strDrive, strRemotePath

strDrive = "J:"
strRemotePath = "\\FinServer\Software"

Set objNetwork =
CreateObject("WScript.Network")
objNetwork.MapNetworkDrive strDrive,
strRemotePath

If Err.Number <> 0 Then
    WScript.Echo "Error: " &
Err.Description
    Err.Clear
End If

Wscript.Quit
```

Secure Design Requirements

Economy of Mechanisms

→ “bells-and-whistles”

Unnecessary functionality or unneeded security mechanisms should be avoided.

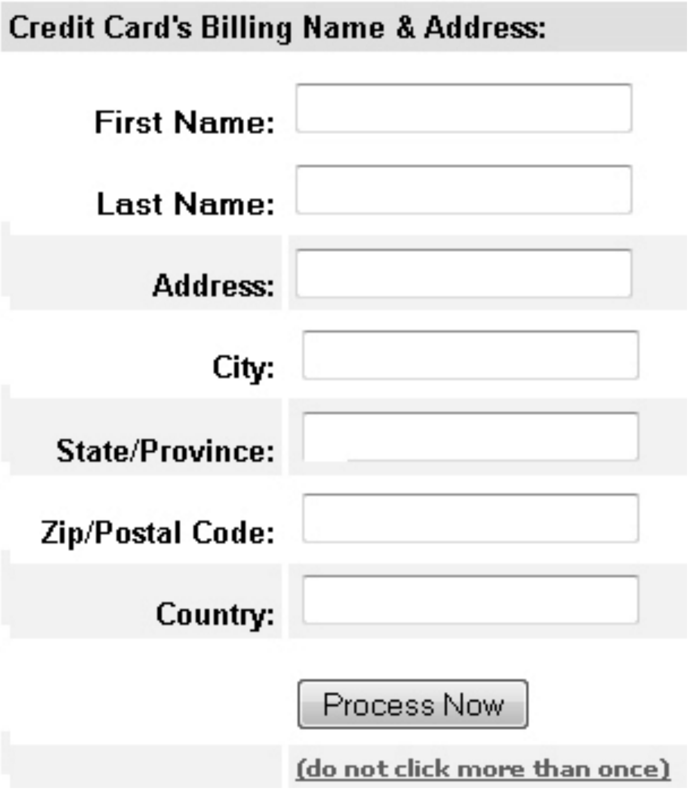
Strive for simplicity

Strive for operational ease of use

Secure Design Requirements

Complete Mediation

Whats wrong in this picture?



Credit Card's Billing Name & Address:

First Name:

Last Name:

Address:

City:

State/Province:

Zip/Postal Code:

Country:

(do not click more than once)

Secure Design Requirements

Complete Mediation

Not checking access rights each time a subject requests access to objects violates the principle of complete mediation

The complete mediation design principle also addresses the failure to protect alternate path vulnerability

Complete mediation also augments the protection against the *weakest link*

Secure Design Requirements

Open Design → The Opposite??

The security of your software should not be dependent on the *secrecy of the design*

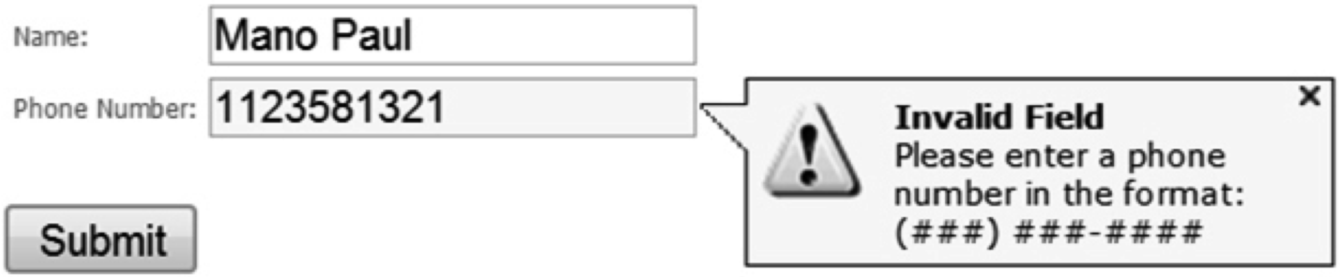
Security through obscurity should be avoided

The design of protection mechanisms should be open for scrutiny by members of the community

Secure Design Requirements

Psychological Acceptability

When user feel that security is usually very complex



Name: Mano Paul

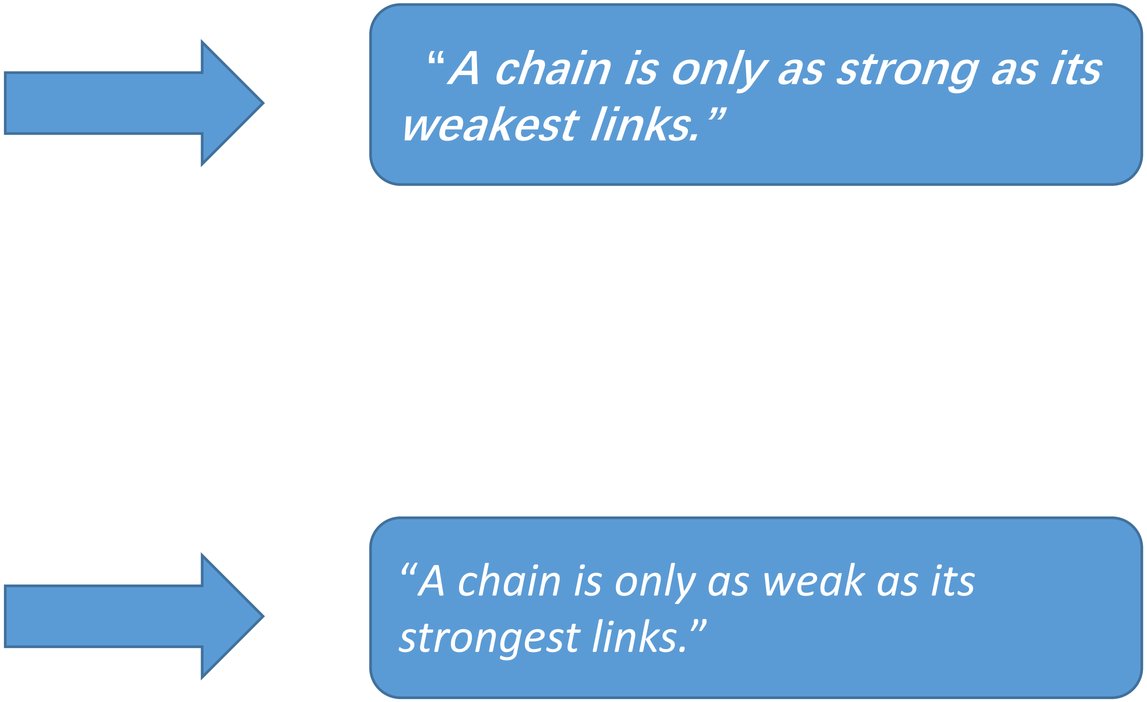
Phone Number: 1123581321

Invalid Field
Please enter a phone number in the format: (###) ###-####

- **are easy to use,**
- **do not affect accessibility, and**
- **are transparent to the user**

Secure Design Requirements

Weakest Link



“A chain is only as strong as its weakest links.”

“A chain is only as weak as its strongest links.”

“Single Point of Failure”

Software must be architected so that there is no single source of complete compromise

Secure Design Requirements

Leveraging Existing Components

Service Oriented Architecture (SOA) is prevalent in today’s computing environment

Avoid custom implementations of cryptographic functionality are also determined often to be the weakest link.

the attack surface is not increased, and no newer vulnerabilities are introduced

Secure Design Requirements

Interface Design

User Interface

Abstractions using user interfaces are also a good defense against insider threats

Application Programming Interfaces (API)

the communication of one software component with another

Secure Design Requirements

Interface Design

Out-of-Band Interface

Log Interfaces

Secure Design Requirements

Interconnectivity

Upstream and downstream compatibility of software should be explicitly designed

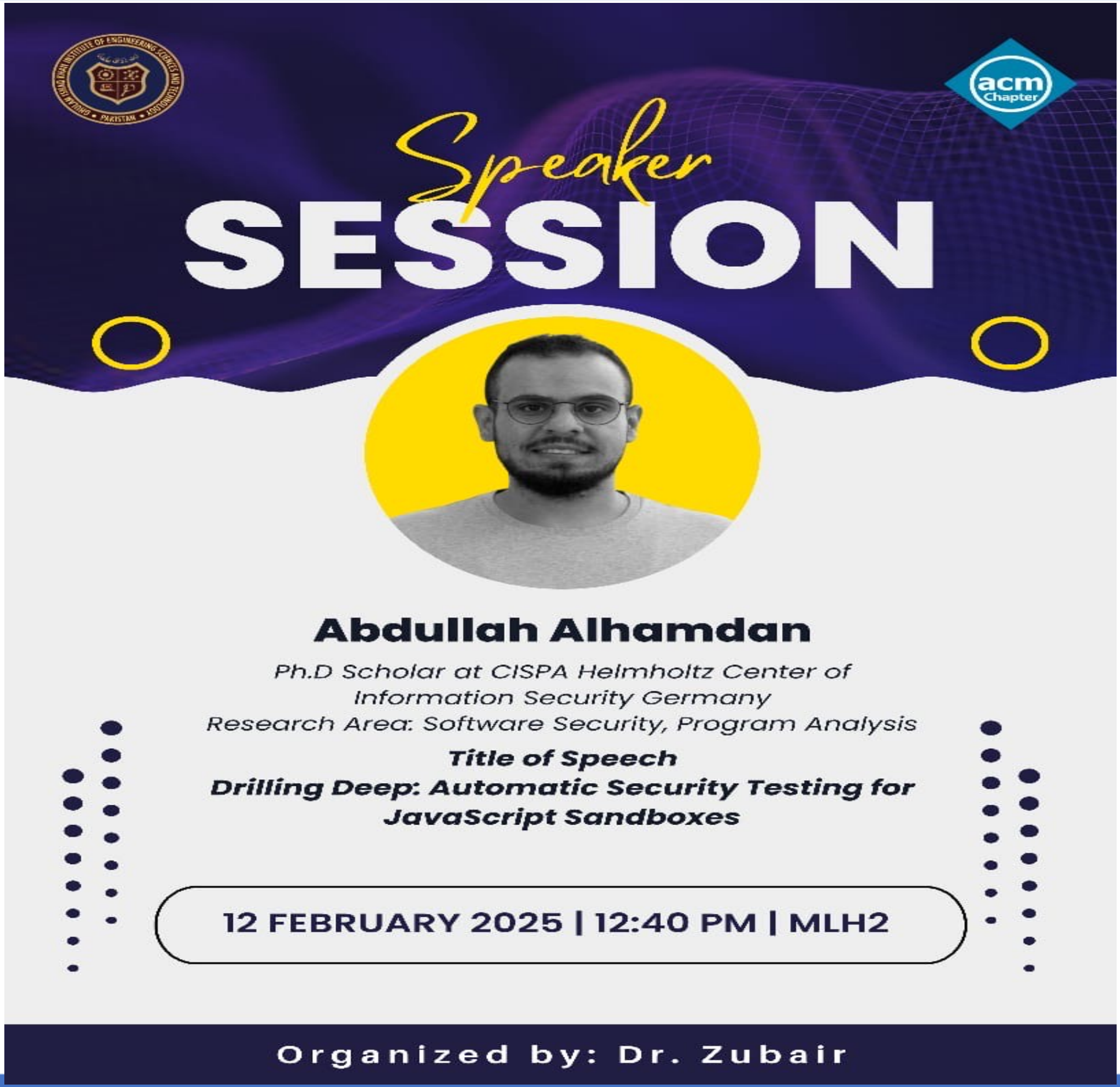
Interconnectivity is not only observed in software applications, but in devices as well.

Balancing Secure Design Principles

“SSO can heighten user experience and increase psychological acceptability, it contradicts the principle of complete mediation and so a business decision is necessary to determine the extent to which SSO is designed into the software or to determine that it is not even an option to consider "

Invited Talk!!

Lets move to MLH2 (NAB) for the invited talk



Speaker

SESSION

Abdullah Alhamdan

Ph.D Scholar at CISPA Helmholtz Center of Information Security Germany

Research Area: Software Security, Program Analysis

Title of Speech

Drilling Deep: Automatic Security Testing for JavaScript Sandboxes

12 FEBRUARY 2025 | 12:40 PM | MLH2

Organized by: Dr. Zubair

Questions??

zubair.ahmad@giki.edu.pk

Office: G14 FCSE lobby