5. SECURITY MISCONFIGURATION

📘 MODULE 5 — A05: SECURITY MISCONFIGURATION

🔥 TABLE OF CONTENTS

🔷 1. What is Security Misconfiguration?

Security misconfiguration occurs when:

- Security settings are **incorrect**
- Defaults are left enabled

- Excessive permissions are provided
- Debugging endpoints are exposed
- Application is deployed in an insecure state
- Services/ports are left open
- Error messages leak sensitive data

This is one of the **most common vulnerabilities** exploited by hackers.

Pentesters love this vulnerability because it often results in:

- Server takeover
- Database leakage
- Unauthorized access
- Cloud resource compromise
- Privilege escalation

🔷 2. Why Misconfigurations Happen

Because developers:

- Trust defaults
- Forget to disable dev/debug mode
- Misconfigure cloud IAM roles
- Leave sample files & backup files
- Use weak permissions (chmod 777)
- Misconfigure CORS
- Expose unnecessary services

🔷 3. Real-World High-Impact Misconfigurations

3.1 Tesla AWS S3 Bucket Leak

Developer left an S3 bucket public → internal code leaked.

3.2 Facebook exposed API debug endpoints

Allowed access to internal logs and tokens.

3.3 Jenkins server without authentication

Gave attackers **RCE** using script console.

🔷 4. Pentester Mindset for Misconfiguration Testing

A pentester must look for:

✔️ Default Credentials

- admin:admin
- root:root
- tomcat:s3cret

✔️ Debug / Admin Panels

- `/admin`
- `/env`
- `/debug`
- `/phpinfo.php`
- `/.git/`
- `/server-status`

✔️ Exposed Services

- Redis without password
- MongoDB open to internet
- Elasticsearch open

✔️ Directory listing

- `/uploads/`
- `/backup/`

- `/logs/`

✔️ Cloud Misconfigurations

- AWS IAM roles too permissive
- Public S3 buckets
- Exposed environment variables

🔷 5. Pentesting Methodology

Step 1 — Enumerate All Services

Ports, services, versions.

Step 2 — Check for Default Services

Tomcat, phpMyAdmin, Jenkins, Kibana.

Step 3 — Test for Weak Permissions

File permissions, folder traversal.

Step 4 — Check Leaked Files

- `.env`
- `.git`
- `backup.sql`
- `config.old.php`

Step 5 — Misconfigured Headers

- Missing security headers
- CORS misconfig
- Clickjacking

Step 6 — Cloud Misconfigurations

- IAM roles
- S3 access

- Storage bucket permissions

🔷 6. Manual Testing (Step-by-Step)

⭐ 6.1 Directory Enumeration

Tool: **Gobuster**, **FFUF**, **Dirsearch**

Command:

```
gobuster dir -u https://target.com -w /usr/share/wordlists/dirb/common.txt
```

What this does:

This brute-forces directory names to find hidden or sensitive folders.

⭐ 6.2 Check Header Misconfiguration

Tool: **curl**

Command:

```
curl -I https://target.com
```

Look for missing:

- X-Frame-Options
- X-Content-Type-Options
- Strict-Transport-Security
- Content-Security-Policy

⭐ 6.3 Check for Debug Mode

Known debug endpoints:

- `/debug`
- `/graphql-explorer`
- `/phpinfo.php`
- `/console`
- `/h2-console` (Java)

Try:

```
curl https://target.com/phpinfo.php
```

⭐ 6.4 Checking Backup Files

Common extensions:

- `.bak`
- `.zip`
- `.old`
- `.tar.gz`

Command:

```
ffuf -u https://target.com/FUZZ -w wordlist.txt -e .bak,.old,.zip
```

⭐ 6.5 Server Status Page Exposure (Apache)

Test:

```
curl https://target.com/server-status
```

If accessible → reveals internal server metrics.

⭐ 6.6 Exposed Git Repository

Test:

```
curl https://target.com/.git/
```

If index is open → full source code leak.

To dump repo:

```
git-dumper https://target.com/.git/ dump/
```

⭐ 6.7 Checking Cloud Misconfiguration (AWS)

Tool: **Pacu**, **ScoutSuite**, **CloudEnum**

Example Command (ScoutSuite):

```
scoutsuite aws --profile default
```

🔷 7. Tools with Full Commands (Detailed)

⭐ 7.1 Nmap (Service Enumeration)

Command:

```
nmap -sV -sC -p- target.com
```

What it does:

- `sV` → detect versions
- `sC` → run default NSE scripts
- `p-` → scan all ports

Detect misconfigurations like:

- Anonymous FTP login
- Open databases (MongoDB, Redis)
- Outdated services
- SNMP misconfig

⭐ 7.2 Nikto (Web Server Misconfig)

Command:

```
nikto -h https://target.com
```

Finds:

- Dangerous files
- Misconfig headers
- Sample scripts

⭐ 7.3 FFUF (Bruteforce Hidden Files)

Command:

```
ffuf -u https://target.com/FUZZ -w /usr/share/wordlists/dirb/big.txt
```

⭐ 7.4 Dirsearch (Fast Directory Scanning)

Command:

```
python3 dirsearch.py -u https://target.com -e php,txt,bak,zip
```

⭐ 7.5 CloudEnum (Check public storage buckets)

Command:

```
cloudenum -k target
```

🔷 8. Attack Scenarios (Realistic)

Scenario 1 — Exposed phpinfo.php

Shows:

- File paths
- OS
- PHP version
- Installed modules
- Environmental variables

Attacker → uses this for targeted RCE exploit.

Scenario 2 — Exposed .env File

Contains:

```
DB_HOST=localhost
DB_USER=root
DB_PASS=password123
SECRET_KEY=XYZ
```

Attacker gains instant DB access.

Scenario 3 — Open MongoDB

If MongoDB is exposed (common):

Test:

```
mongo mongodb://target.com:27017
```

If no authentication → DB dump.

Scenario 4 — Misconfigured CORS

Response:

```
Access-Control-Allow-Origin: *
```

Allows:

- Cookie theft (if credentials enabled)
- Cross-site unauthorized requests

Scenario 5 — Jenkins without Auth

Access:

```
https://target.com/jenkins/script
```

Run:

```
println "Hacked"
```

RCE possible.

🔷 9. Payload Examples

Directory listing check:

```
/uploads/
/backup/
/logs/
```

Access hidden config:

```
/config.php.bak
```

Check admin panels:

```
/admin
/administrator
/cpanel
```

🔷 10. Reporting Template (Professional)

Title: Misconfigured Apache Server Exposing server-status

Severity: High

Impact: Information Disclosure

Description:

The URL `/server-status` provides internal metrics including active connections, IPs, requests.

Steps to Reproduce:

1. Navigate to `/server-status`
2. Page loads with server details

Evidence:

Screenshot + raw HTTP response.

Recommendation:

Restrict using:

```
<Location /server-status>
   Require ip 127.0.0.1
</Location>
```

🔷 11. Mitigation & Hardening

✔️ Disable debug mode in production

✔️ Restrict admin panels

✔️ Apply least privilege

✔️ Remove unused features

✔️ Secure cloud resources

✔️ Disable directory listing

✔️ Set strong security headers

✔️ Use WAF

✔️ Regular configuration auditing

🔷 12. Hands-On Labs

Lab 1: Enumerate directories with Gobuster

Lab 2: Check missing security headers

Lab 3: Exploit exposed .git directory

Lab 4: Dump a misconfigured MongoDB instance

Lab 5: Find backup files using FFUF