📘 Domain 5: Secure Systems Design (18%)

🔹 5.1 Architecture Models

🧱 Security Architecture Concepts

Model	Description
Least Privilege	Users/apps get only the access they need
Defense in Depth	Multiple security layers (firewall, AV, MFA, etc.)
Zero Trust	"Never trust, always verify" — assumes breach
Segmentation	Separate networks/services into zones
Air Gapping	Physically isolate system from network

> 🔐 Use **segmentation** to limit lateral movement. **Zero Trust** is now a major security trend.

🏢 Security Zones

- **DMZ (Demilitarized Zone)**: Exposes services like web/mail to public.
- **Intranet**: Internal corporate network.
- **Extranet**: Shared with trusted partners/vendors.
- **Cloud**: Public, private, hybrid models.

🔄 Architectural Approaches

- **On-prem**: Full control, full responsibility.
- **Cloud**: Shared responsibility.
- **IaaS / PaaS / SaaS**:
    - IaaS: Infra (e.g., VMs) → You secure OS and apps.
    - PaaS: Platform (e.g., App Engine) → You secure code.
    - SaaS: Apps (e.g., Gmail) → Mostly vendor responsibility.

🔹 5.2 System Hardening

🔧 Hardening Best Practices

- Remove unnecessary services/software
- Disable unused ports and protocols
- Rename or disable default accounts
- Apply security patches
- Use secure configurations (CIS Benchmarks)

✅ Harden **servers, endpoints, network devices, VMs**, and **containers**.

🛠️ Secure Configuration Tools

- **Group Policy Objects (GPO)** for Windows
- **Ansible / Chef / Puppet** for automation
- **SCAP / CIS Benchmarks** for security baselines

🔹 5.3 Security Implications of Embedded and Specialized Systems

🧩 Types of Embedded/OT Systems

System	Description
IoT Devices	Smart home, sensors, wearables
SCADA/ICS	Industrial control systems (power grid, water)
RTOS	Real-time operating systems (aircraft, robotics)
Medical Devices	Infusion pumps, pacemakers
Automotive Systems	ECU, infotainment, ADAS

> 🔥 Embedded devices often lack proper patching, encryption, or strong access control — they're **common attack targets**.

⚙️ Common Security Risks

- Default credentials
- Insecure firmware updates
- Lack of encryption
- Limited monitoring

> 🛡️ Use **network segmentation**, firmware validation, and **zero trust** for OT security.

🔹 5.4 Cloud and Virtualization Security

☁️ Cloud Security Considerations

- **Shared Responsibility Model**: You vs. cloud provider
- **Data Classification & Encryption** in cloud
- **IAM Controls** (roles, policies, MFA)
- **Logging & Monitoring** via cloud-native tools (e.g., CloudTrail, Azure Monitor)

📦 Virtualization Concepts

Term	Description
Hypervisor Type 1	Bare metal (e.g., ESXi)
Hypervisor Type 2	Runs on OS (e.g., VirtualBox)
Snapshots	Save VM state for rollback
Containerization	Lightweight app environments (Docker)
Orchestration	Manages containers (Kubernetes)

> 🔐 Secure virtual environments with **isolation**, **resource limits**, and **container hardening**.

🔹 5.5 Mobile Device Security

📱 Common Controls

- Screen locks, encryption, remote wipe
- Mobile Device Management (MDM)
- App vetting/whitelisting
- Blocking rooted/jailbroken devices
- Enforcing VPNs and secure Wi-Fi

📶 Mobile Threats

- Malicious apps
- Rogue Wi-Fi / man-in-the-middle
- Data leakage
- OS vulnerabilities

> ✅ Use **Mobile Application Management (MAM)** to secure work apps even on personal devices.

🔹 5.6 Secure Application Development

🔐 Secure Development Lifecycle (SDLC)

1. **Planning** – Define security requirements
2. **Design** – Use secure architecture
3. **Development** – Follow secure coding standards
4. **Testing** – Conduct static/dynamic analysis
5. **Deployment** – Harden servers, configs
6. **Maintenance** – Patch and monitor

🧪 Code Testing Types

Test	Description
Static Analysis	Analyzes source code (SAST)
Dynamic Analysis	Tests running apps (DAST)
Fuzzing	Sends random input to test stability
Regression Testing	Re-checks after changes

🛡️ Use **input validation**, **output encoding**, and **least privilege** in code.

☠️ Common App Vulnerabilities

- **Injection (SQL, command)**
- **Broken authentication**
- **Sensitive data exposure**
- **Cross-site scripting (XSS)**
- **Insecure deserialization**
- **Security misconfigurations**

> 🔍 Refer to the **OWASP Top 10** list to understand and prevent these issues.

🔹 5.7 Resiliency and Redundancy

🧱 High Availability Concepts

Term	Description
Load Balancer	Distributes traffic
Failover	Auto-switch to backup system
Clustering	Redundant nodes for availability
Redundant Power / NICs / Paths	Avoid single points of failure

---

🔄 Backup Strategies

- **Full** – All data
- **Incremental** – Since last backup
- **Differential** – Since last full backup

✅ Follow **3-2-1 Rule**: 3 copies, 2 types of media, 1 offsite.

⚙️ Fault Tolerance Tools

- RAID (disk redundancy)
- UPS (uninterruptible power supply)
- Generators
- Geographic redundancy (multi-site)

📌 Key Takeaways for Domain 5

- Understand **cloud models**, virtualization, and shared responsibility.
- Know how to **harden systems** and secure **mobile, embedded, and IoT** devices.
- Be familiar with **application development** risks and **secure coding practices**.
- Design for **resilience and availability** with redundancy and backups.
