Day 28

"CLOUD SECURITY"

Considerations for Designing Secure Cloud Applications

1. Multitenant Application Isolation
    a. Separate tenant data using container isolation and network isolation.
2. Application Security Management
    a. Use threat modeling, secure coding practices, secure design, and security testing.
    b. Follow SSDLC to ensure vulnerabilities are fixed before deployment.
3. Identity & Access Management (IAM)
    a. Use SAML or OpenID Connect for API authentication (partners, employees, customers).
    b. Use cloud directories for new customer authentication.
4. Web Application Protection
    a. Apply CSP-provided DDoS protection.
5. Application Runtime & Services Security
    a. Use RASP (Runtime Application Self-Protection) to detect and stop attacks in real time.
6. Visibility Across Environment
    a. Maintain logging, monitoring, and activity analysis to secure infrastructure and apps.

Best Practices for Secure Cloud Applications

1. Segregation by Default – Use isolated cloud environments (accounts/sub-accounts, virtual networks). Keep production restrictive and allow more rights only in dev/test.
2. Immutable Infrastructure – No remote logins, use immutable containers/servers, file integrity monitoring, and immutable recovery methods.
3. Microservices Security – Secure communication, discovery, routing, and scheduling between microservices.
4. PaaS & Serverless Architectures – Reduce attack surface; CSP secures the platform. Serverless prevents direct network attacks, limiting exposure to only API/HTTPS calls.
5. Software-Defined Security – Automate incident response, change management, remediation of unapproved changes, and dynamic entitlements.
6. Event-Driven Security – Use events (e.g., config changes, file uploads) to trigger automated security actions like assessment, notification, or remediation.

Vulnerability Assessment in CI/CD

- CI/CD Pipelines: Automate integration, testing, delivery, and deployment. Enable DevSecOps by embedding security checks.
- Cloud-Native Pipelines: Hosted in cloud, improve security + compliance for cloud apps.

Integration Patterns

1. Image Assessment – Run vulnerability scans on containers/images in a dedicated test segment. Approve only if passed.
2. Infrastructure Testing – Build a full test environment with IaC to scan the entire infrastructure for vulnerabilities.

Best Practice

- Keep production immutable (same as test), reducing need for live scans.
- Use host-based vulnerability tools inside VMs without requiring provider permissions.

DevOps vs DevSecOps:

Aspect	**DevOps**	**DevSecOps**
Focus	Speed, automation, and collaboration for faster software delivery.	Embeds security into every stage of DevOps pipeline.
Security Handling	Security checked at the end (after build/deploy).	Security integrated from the start (shift-left approach).
Responsibility	Security mainly handled by a separate security team.	Security is a shared responsibility of Dev, Sec, and Ops.
Tools & Practices	CI/CD, monitoring, automation tools.	Adds SAST, DAST, container scans, compliance checks to CI/CD.