ONE PAGE SUMMARY

IT & Software › Software Architecture › Secure Architecture

Introduction to Secure Architecture

Building secure software starts with a solid architecture foundation.

DEFINITION

> Secure architecture involves designing software systems with security as a primary focus. It ensures that applications are resilient against threats and vulnerabilities.

KEY CONCEPTS

Threat Modeling

Identifying potential security threats to a system and planning defenses against them.

Defense in Depth

Using multiple layers of security controls to protect data and systems.

Least Privilege

Granting users and systems only the access necessary to perform their tasks.

Secure Coding Practices

Following coding guidelines that minimize vulnerabilities in software development.

Regular Audits

Conducting periodic reviews of the architecture to identify and fix security issues.

EXAMPLES

→ Implementing firewalls to protect network boundaries.

→ Using encryption for sensitive data storage and transmission.

→ Conducting penetration testing to find vulnerabilities.

MEMORY TIPS

★ Remember 'T-D-L-S-A' for Threat, Defense, Least privilege, Secure coding, Audits.

★ Visualize a castle with multiple walls for Defense in Depth.

★ Think of a key that only fits one lock for Least Privilege.

COMMON MISTAKES

✗ Neglecting to update security measures regularly.

✗ Assuming security is only an IT issue, not a design issue.

✗ Overlooking third-party software vulnerabilities.

QUICK RECAP

Secure architecture is essential for protecting software systems from threats. Key practices include threat modeling, defense in depth, and secure coding. Regular audits help maintain security over time.