# Revision Notes with Explanations: Designing Secure Architectures

## 1. Session Aims and Objectives

* Zero-trust approach: Ensure that no user or system is trusted by default. Always verify credentials and access.
* Advanced security techniques: Utilise encryption, multi-factor authentication (MFA), intrusion detection, and more.
* Visual communication: Use diagrams and flowcharts to clearly represent security components and data flow.

## 2. Key Concepts and Definitions

* Authentication vs Authorisation: Authentication confirms identity; authorisation grants access to resources.
* Service: A software function that performs a specific task (e.g., a web service).
* Service Account: An account used by applications/services to interact with the OS or other apps.
* Threat Modelling: Identifying threats, vulnerabilities, and attack vectors during the design process.
* Redundancy and Resilience: Designing systems to recover from failures or attacks.
* Security Controls: Safeguards or countermeasures to reduce risk and enforce security policies.

## 3. Secure by Design

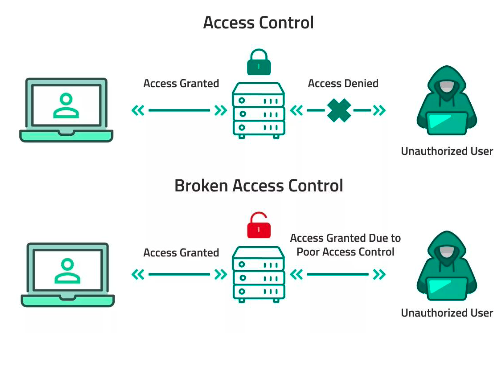
* Integrated security: Include security from the beginning of the design process.
* Proactive threat identification: Anticipate and mitigate risks before they occur.
* Cost efficiency: Fixing security issues during design is cheaper than post-deployment.
* Stakeholder trust: Users and stakeholders are more likely to trust secure systems.

## 4. Security Architecture Principles

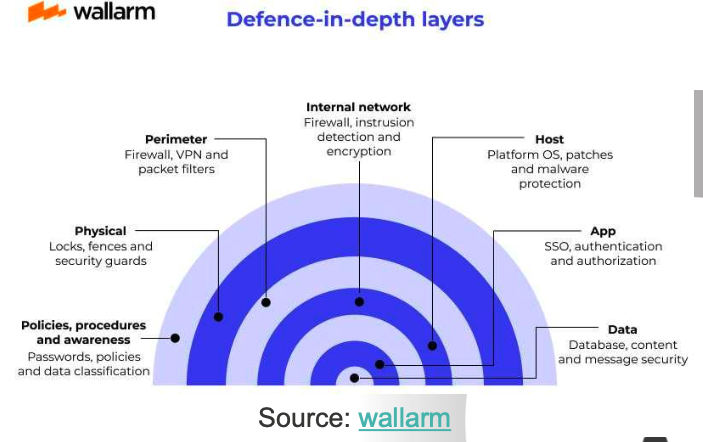
* Simplicity: Fewer components reduce the potential for vulnerabilities.
* Restrictiveness: Limit access to only what's needed to reduce risk.
* Economy of Mechanism: Keep systems simple and modular for easier auditing and maintenance.
* Fail-Safe Defaults: Default settings should deny access unless explicitly allowed.
* Complete Mediation: All access attempts should be checked for permissions.
* Open Design: Security should not rely on secrecy; the design should be robust even if known.
* Separation of Privilege: Require multiple approvals or factors to perform critical actions.
* Least Common Mechanism: Avoid shared mechanisms among users to limit communication paths.
* Psychological Acceptability: Security features should be user-friendly to encourage compliance.

## 5. Security Design Patterns

* Principle of Least Privilege: Grant users only the permissions they need to perform their tasks.



* Defence in Depth: Implement multiple layers of security to defend against various threats.



## 6. Practical Security Principles

* Secure Communication: Encrypt and authenticate data in transit using HTTPS, TLS.
* Secure Configuration: Disable unused features and secure default settings.
* Access Control: Define roles and use permissions and MFA to protect resources.
* Monitoring and Logging: Track activity and detect suspicious actions via logs.
* Patch Management: Apply updates promptly to fix known vulnerabilities.

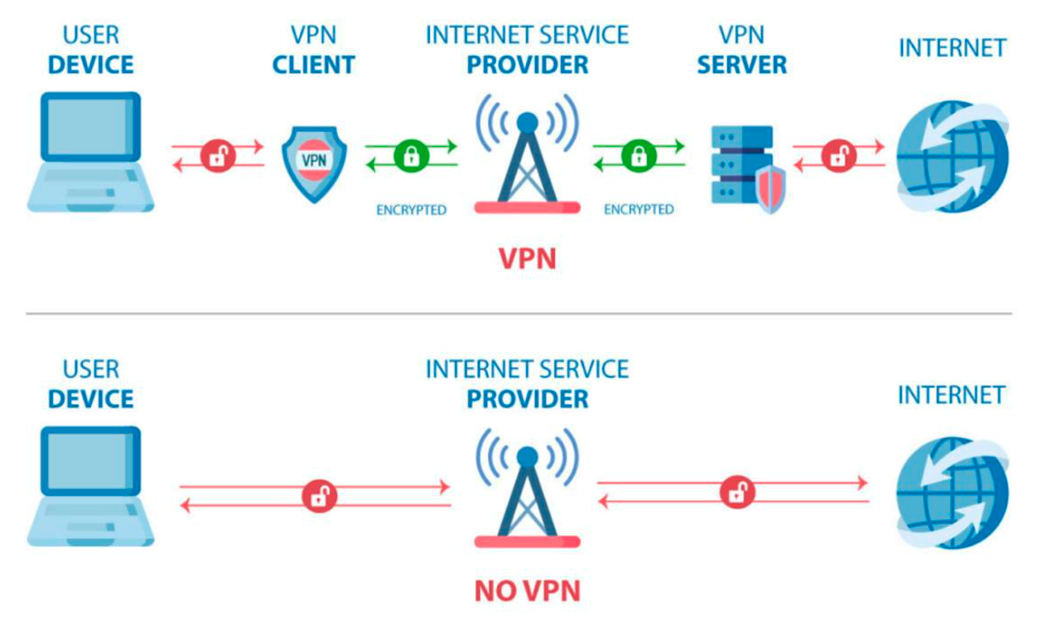
## 7. Anti-Patterns

* Security by Obscurity: Relying on secrecy of design or implementation instead of robust security.
* Unpatched Systems: Failing to apply updates can leave known vulnerabilities open to attackers.

## 8. Network and Communication Security

* Confidentiality: Ensure only authorised users can view sensitive data by encrypting it.
* SSL/TLS: Protocols that secure data in transit and prevent tampering or eavesdropping.
* HTTPS: HTTP over SSL/TLS, used for secure web communication.
* X.509 Certificates: Digital certificates used to authenticate identities and secure connections.
* Downgrade Attacks: Tricking systems into using less secure versions of protocols. Prevented by proper configurations.

VPN



## 9. OWASP Top 5 Vulnerabilities

* Broken Access Control: Weak permissions allow unauthorised actions.
* Cryptographic Failures: Insecure or outdated encryption methods expose data.
* Injection: Unsanitised input (e.g., SQL) can manipulate backend systems.
* Insecure Design: Design flaws that expose system information or enable attacks.
* Security Misconfiguration: Leaving default settings or services open to exploitation.

## 10. Best Practices

* Update Software: Regular updates patch known security issues.
* Secure Keys: Protect private keys used for encryption and authentication.
* Monitor Systems: Continuous monitoring helps detect and respond to threats early.