Top 10 Network Security Design Principles

Narudom Roongsiriwong

ThaiNOC 2024, May 31, 2024

About Me

- Information Security since 1995
- Web Application Development since 1998
- SVP, Global Architecture and Cyber Security,
  Banpu Public Company Limited
- Security and Risk Committee at National Digital ID Co.,Ltd.
- Cloud Security Alliance Fellow
- OWASP Bangkok Chapter Leader
- APAC Research Advisory Council Member, Cloud Security Alliance Asia Pacific
- CISO of the Year 2017, NetworkWorld Asia
- Contact: narudom@owasp.org

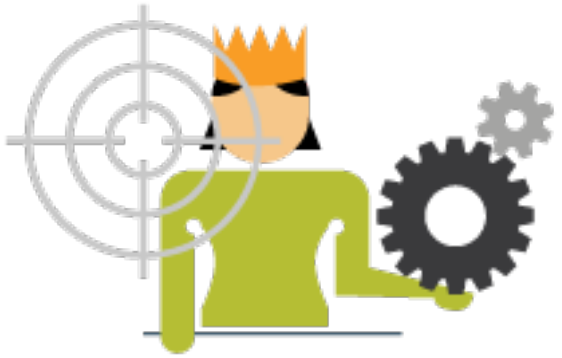
BANPU

- Least Privilege
- Separation of Duties
- Defense in Depth
- Fail Safe / Fail Secure
- Economy of Mechanisms
- Compartmentalization
- Complete Mediation
- Open Design
- Psychological Acceptability
- Weakest Link

Network Security Design Principles

Least Privilege

- When designing access controls, each user, network connection or system component should be allocated the minimum privilege required to perform an action for the minimum amount of time
- To limit damage that can be caused by an accident, error, or unauthorized act
- Benefits of the principle include:
  - Better system stability
  - Better system security
  - Ease of deployment



Privileged users have become the main **target for cyber criminals**.

Implementing Least Privilege

- Make "least privilege" the baseline for all new accounts and processes. Grant only the most basic permissions required for their function. Additional privileges should be granted on a case-by-case basis.
- Separate administrative accounts from standard user accounts. Avoid giving users with regular privileges the ability to perform administrative tasks.
- Whenever possible, restrict elevated privileges to only the moments they are strictly needed. This can involve using temporary permissions or one-time use credentials.

Cybersecurity Approaches That Rely on “Least Privilege”

- Zero Trust Security: Assumes constant verification and enforces strict access controls.
- Privilege Escalation: Allows users to have basic access for everyday tasks, but requires explicit authorization (escalation) for actions requiring higher privileges. Such as “sudo” in Unix/Linux
- Microsegmentation: Divides networks into smaller, isolated zones. Access controls are enforced at these micro-perimeters, granting access only to authorized users and devices for specific resources within the segment.
- Network Whitelisting: Configure firewall to deny access by default, and only allow specific exceptions. This makes it much harder for unauthorized access to occur.

Separation of Duties

- The concept of dividing critical tasks and responsibilities among multiple people or processes.
- The primary objective of SoD is the prevention of fraud, errors, or compromise on a person or process by distributing tasks

How to Separate the Actors

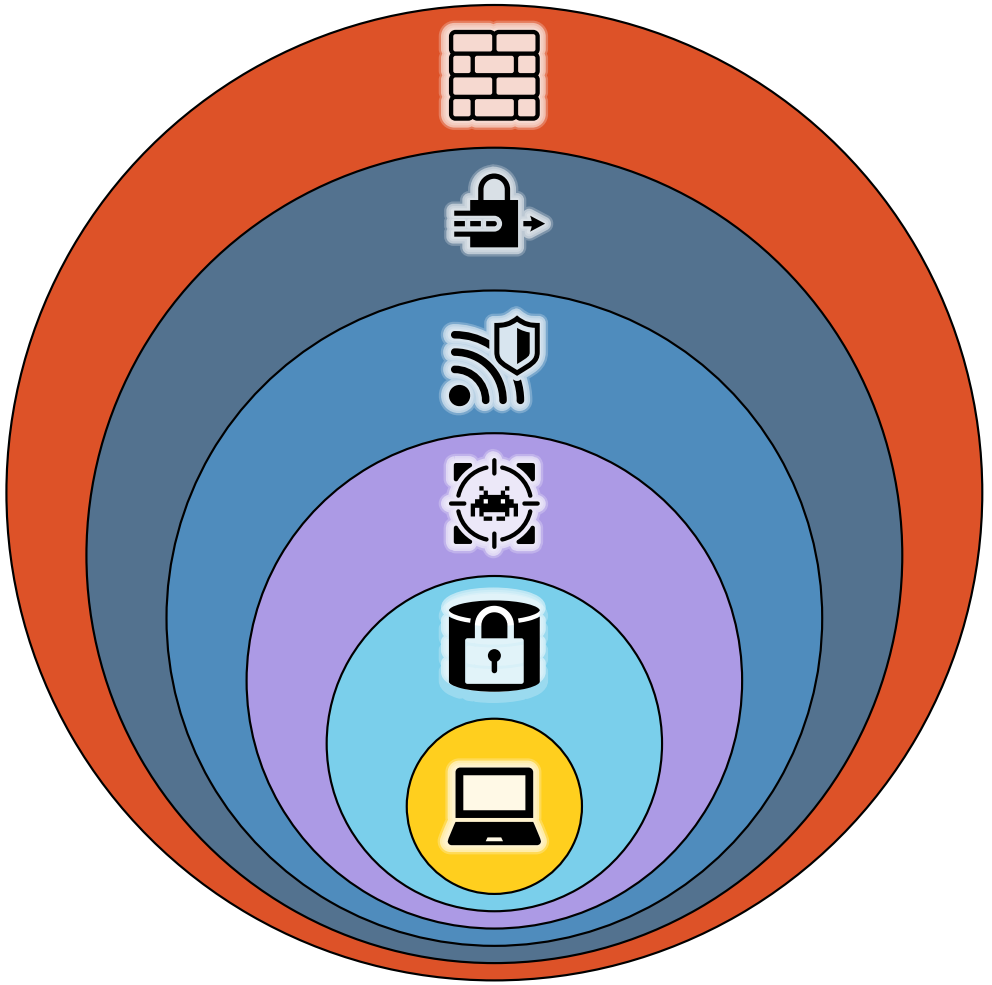
- By individuals—This is the traditional and most basic level of segregation. Either same or different duties will be performed by different individuals
- By functions or organizational units —At this level, different functions perform the separated duties. For example, the sales department might prepare an offering, which is then signed off by the operations department or the risk management function.
- By companies —At this level, operations must be performed by different legal entities. For example, investments made by a subsidiary might require authorization by the controlling company. Third-party audits may be viewed as an example of company-level separation of duties as well.

Examples Separation of Duties

- Access Control Granting and Defining: Separating the responsibilities of granting user access from those responsible for defining system roles and permissions. This prevents potential abuse of system privileges.
- Data Backup and Recovery: Splitting backup and recovery tasks among different roles to prevent one person from having complete control. For example, “Backup Administration”, “Monitoring and Verification”, and “Data Recovery Expertise”
- Network Administration: Separating the roles of “Network Configuration”, “System Administration”, and “Security Monitoring”

Defense in Depth

- The use of multiple computer security techniques to help mitigate the risk into network and system design
- To defend a system against any particular attack using several, varying methods.



Examples of Defense in Depth

- Network Perimeter Defenses: Firewalls + IDS/IPS + Web Application Firewall (WAF)
- Data Security: Data Encryption + Access Controls + Data Loss Prevention (DLP)
- Defend Cross-Site Scripting (XSS): Disallowing Active Scripting + Output Encoding + Input or Request Validation
- The use of security zones, which separates the different levels of access according to the zone that the software or person is authorized to access

Fail Safe / Fail Secure

- Fail Safe
  - Ensure that the system is expected to eventually fail but when it does it will be in a safe way
  - Identify a safe state and transition to that safe state upon failure
  - Aims to maintain system functionality even during partial failure.
- Fail Secure
  - Ensure that the system reliably functions when attacked and is rapidly recoverable into a normal business and secure state in the event of design or implementation failure.
  - Aim to maintain system resiliency (confidentiality, integrity and availability) of a system by defaulting to a secure state.

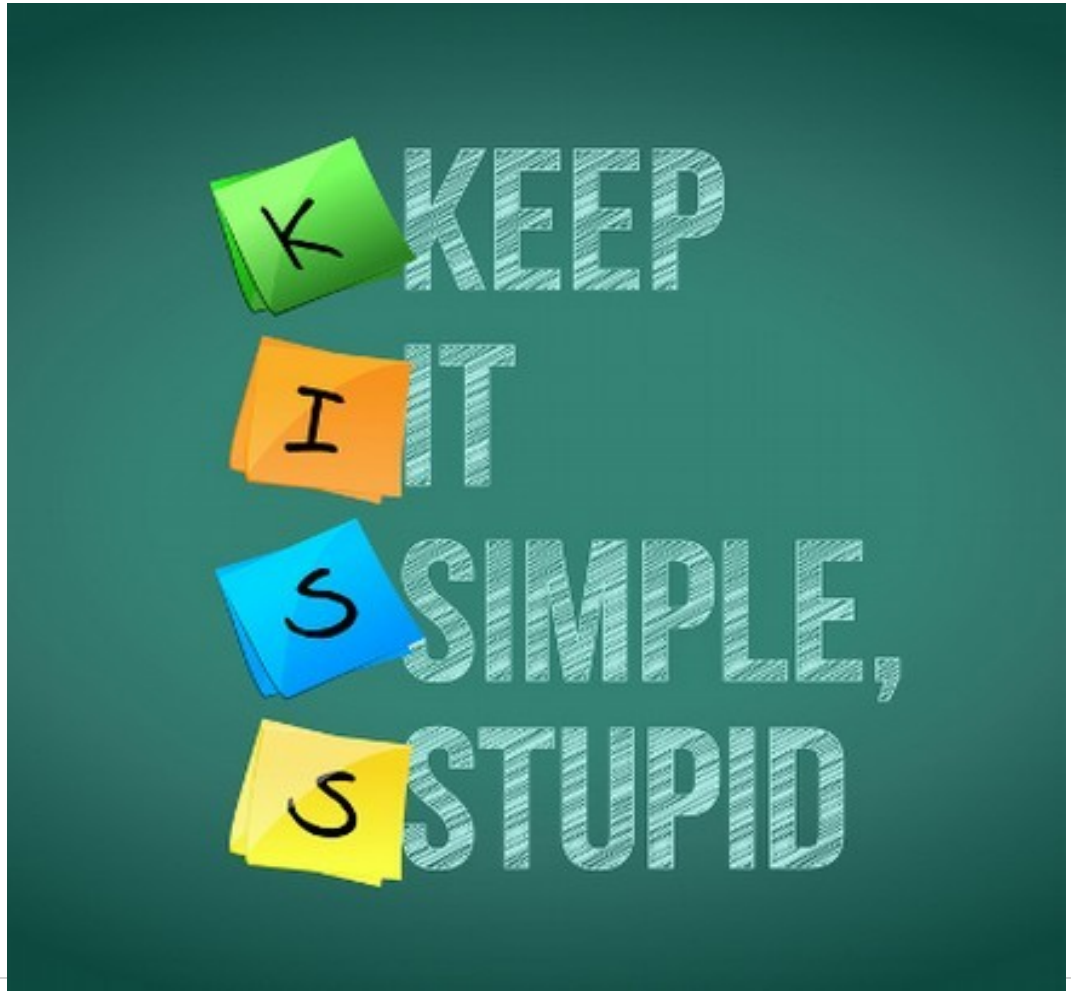
Examples of Fail Safe and Fail Secure

- Fail Safe
  - Redundant Network Paths: A network might have two internet connections. If one connection fails, the system automatically switches to the other, ensuring continued internet access.
  - Routing Protocols: Routing protocols dynamically adjust network traffic flow if a router or path malfunctions. This ensures data reaches its destination despite the failure.
- Fail Secure
  - Firewall Configuration: When a firewall is malfunction, network segments are isolated.
  - Limited Functionalities in Recovery Mode: By limiting available features, recovery mode reduces the attack surface and potential for further damage during troubleshooting. With fewer functionalities running, there are less entry points for malicious actors to exploit.

Economy of Mechanisms (Minimize Attack Surface)

- Core Idea:
  - Do more with less: Strive for the simplest approach that effectively achieves the desired security outcome.
  - Complexity is the enemy: Complex mechanisms are harder to understand, design, implement, and maintain. This increases the risk of vulnerabilities and errors.
- Benefits:
  - Improved Security: Simpler mechanisms are easier to analyze and test, leading to a more secure system overall. Fewer moving parts mean fewer potential points of failure.
  - Reduced Maintenance: Simpler systems require less effort to maintain and troubleshoot. This saves time and resources.
  - Enhanced Clarity: Straightforward mechanisms are easier to understand for everyone involved, from developers to users. This promotes better communication and reduces confusion.

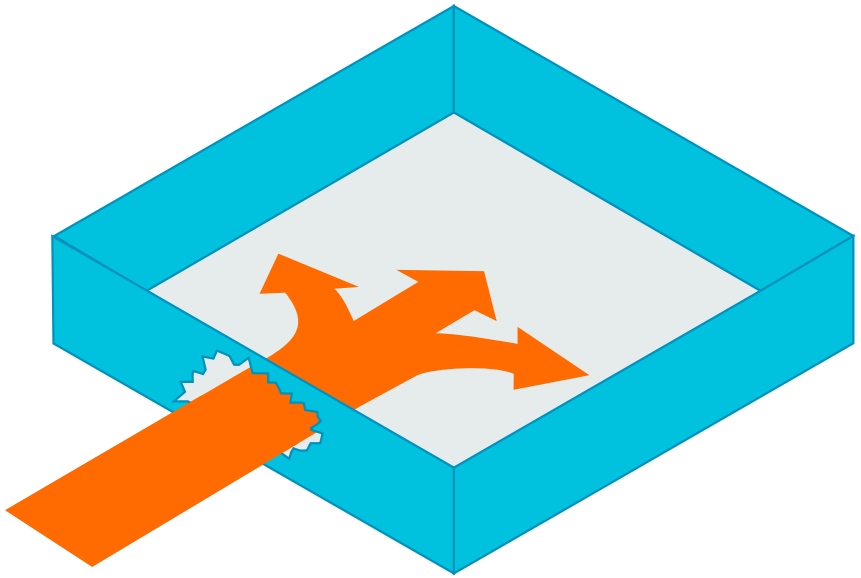
Economy of Mechanisms - Keep It Simple Stupid



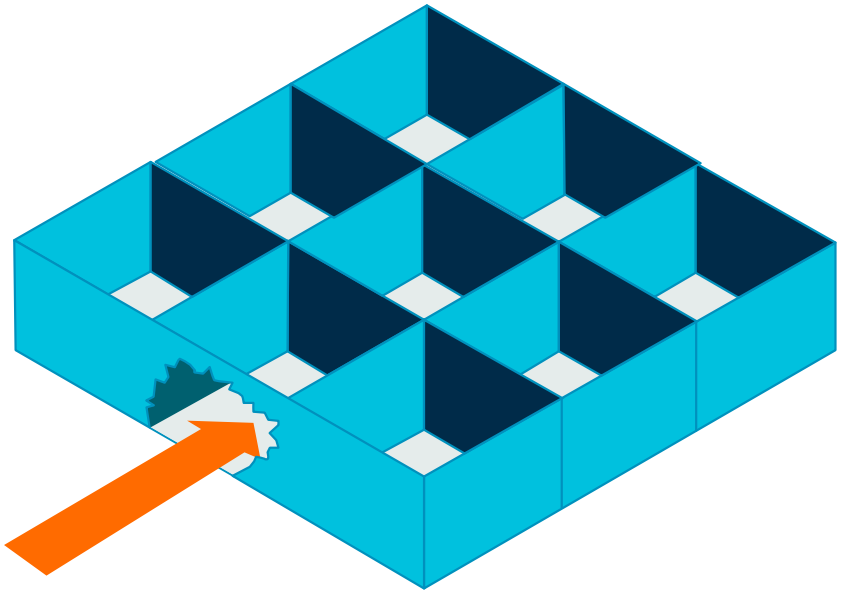
- Unnecessary functionality or unneeded security mechanisms should be avoided.
- Strive for simplicity
- Strive for operational ease of use

Compartmentalization

- Divide the network into segments or zones with different security levels. This limits the lateral movement of attackers within the network if they gain access to one segment.



Breach has full access



Breach is contained to a specific area

Examples of Compartmentalization

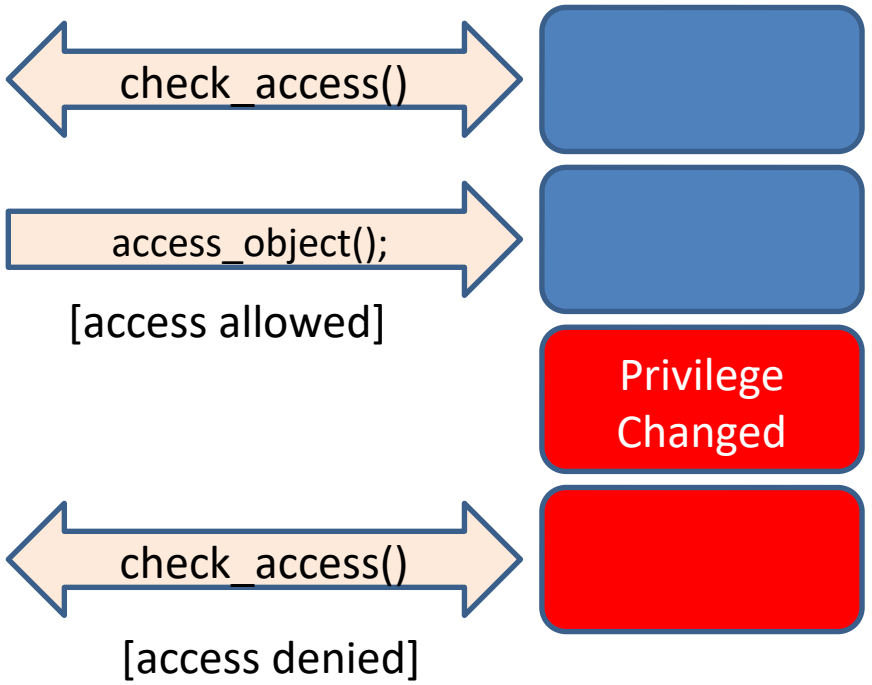
- Network Segmentation: An architecture that divides a network into smaller sections or subnets. Each network segment acts as its own network, which provides security teams with increased control over the traffic that flows into their systems.
- VLANs (Virtual Local Area Networks): VLANs are logical subdivisions within a physical network. They allow you to group devices based on function or department, even if they're physically connected to the same switch.
- Security Groups: Software Defined Network (SDN) firewalls are usually policy sets that define rules for inbound and outbound traffic to a single asset or a group of assets, independent of network location (within a given virtual network).

Complete Mediation

- Every access to every object should be checked

```
MyAccess – FALSE;
MyAccess = check_access();
If (MyAccess == TRUE) {
  access_object();
}

MyAccess = check_access();
If (MyAccess == TRUE) {
  access_object();
}
```



Implementing Complete Mediation in Network Security

- Avoid to rely solely on client-side, cookie-based caching of authentication credentials for access, if possible
- Implement mechanisms to mediate all access attempts to network resources such as Firewalls, IDS/IPS, and Web Application Firewall with continuous authorization checks.
- Leverage network devices like Client-to-Site VPNs, Proxy Servers, and API Gateways that enforce granular access control with continuous authorization checks.

Open Design

- Security should not depend on the secrecy of its design or implementation
- Avoid “security through obscurity” which relies on keeping the system hidden, believing attackers won't be able to exploit what they don't know.
- Protection mechanisms should be open for public to find a security vulnerability or flaw than it is for an attacker

Benefits of Open Design

- Enhanced Security Through Review: By openly sharing the design, you invite security experts and the community to analyze it for vulnerabilities. This collaborative approach helps identify and address weaknesses before they can be exploited.
- Faster Patching: When vulnerabilities are discovered in an open system, they can be patched more quickly. The open community can contribute to finding solutions and developing fixes.
- Promotes Collective Security Responsibility: An open design fosters a sense of shared responsibility for security. Everyone can contribute to identifying and mitigating threats.

Psychological Acceptability

- The system should not place an undue burden on its users
- Users frustrated with security may turn off security mechanisms or avoid use
- However, by default should be secure but the user can reduce security if they are allowed.
- The security protection mechanisms must
  - Be easy to use
  - Not affect accessibility
  - Be transparent to the user

Weakest Link

- The resiliency of your software against hacker attempts will depend heavily on the protection of its weakest components, be it the code, service or an interface.
- “A chain is only as strong as its weakest links.”
  → So is the network

Balancing Secure Design Principles

- Design with all security principles above may not be possible. Don’t worry.
- Careful design considerations need to be given to balance, based on the business needs and requirements, without reducing the security of the systems

BANPU