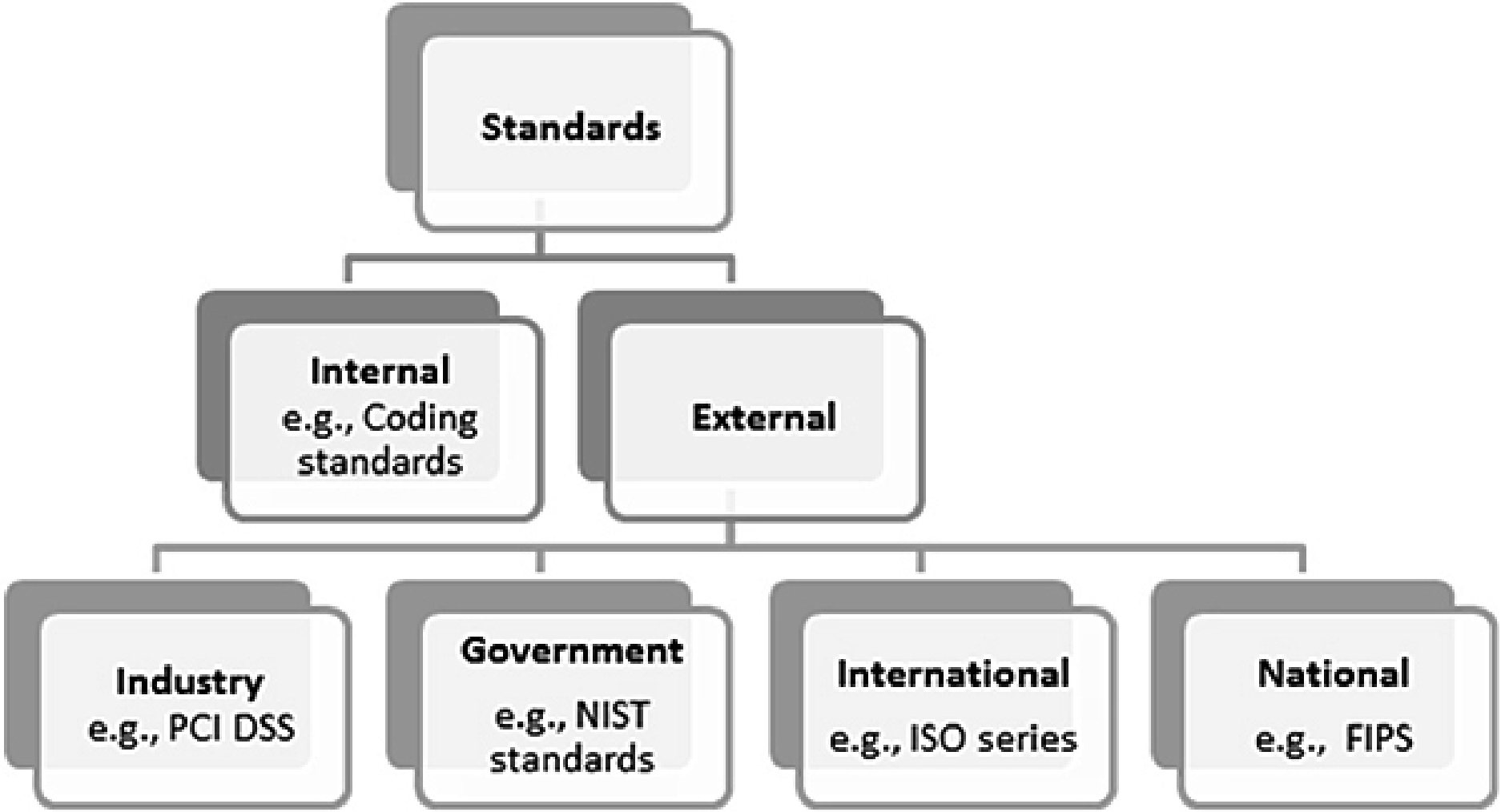
STANDARDS

ISO/IEC 15408 Evaluation Assurance Levels

- EAL1 Functionally tested
- EAL2 Structurally tested
- EAL3 Methodically tested and checked
- EAL4 Methodically designed, tested and reviewed
- EAL5 Semi-formally designed and tested
- EAL6 Semi-formally verified design and tested
- EAL7 Formally verified design and tested

TYPES OF STANDARDS



STANDARDS

- National Institute of Standards and Technology (NIST)
  - Special Publications
  - Federal Information Processing Standards (FIPS)
- International Organization for Standardization (ISO)
- Payment Card Industry (PCI)
- Organization for the Advancement of Structured Information Standards (OASIS)

STANDARDS

Important NIST Publications

FIPS 200 Minimum Security Requirements for Federal Information and Information Systems

FIPS 199 Standards for Security Categorization of Federal Information and Information Systems

FIPS 197 Advanced Encryption Standard

FIPS 186-3 Digital Signature Standard (DSS)

FIPS 190- 4 Secure Hash Standard (SHS)

FIPS 140 series Security Requirements for Cryptographic Modules

SP 800- 152 A Profile for U.S. Federal Cryptographic Key Management Systems (CKMS)

SP 800- 107 Recommendation for Applications Using Approved Hash Algorithms

SP 800- 100 Information Security Handbook: A Guide for Managers

Continued in next slide…

STANDARDS

Continued from previous slide…

SP 800-64 Security Considerations in the System Development Life Cycle

SP 800-63 Electronic Authentication Guideline

SP 800-53 Security and Privacy Controls for Federal Information Systems and Organizations

SP 800-30 Guide for Conducting Risk Assessments

SP 800-14 Generally Accepted Principles and Practices for Securing Information Technology Systems

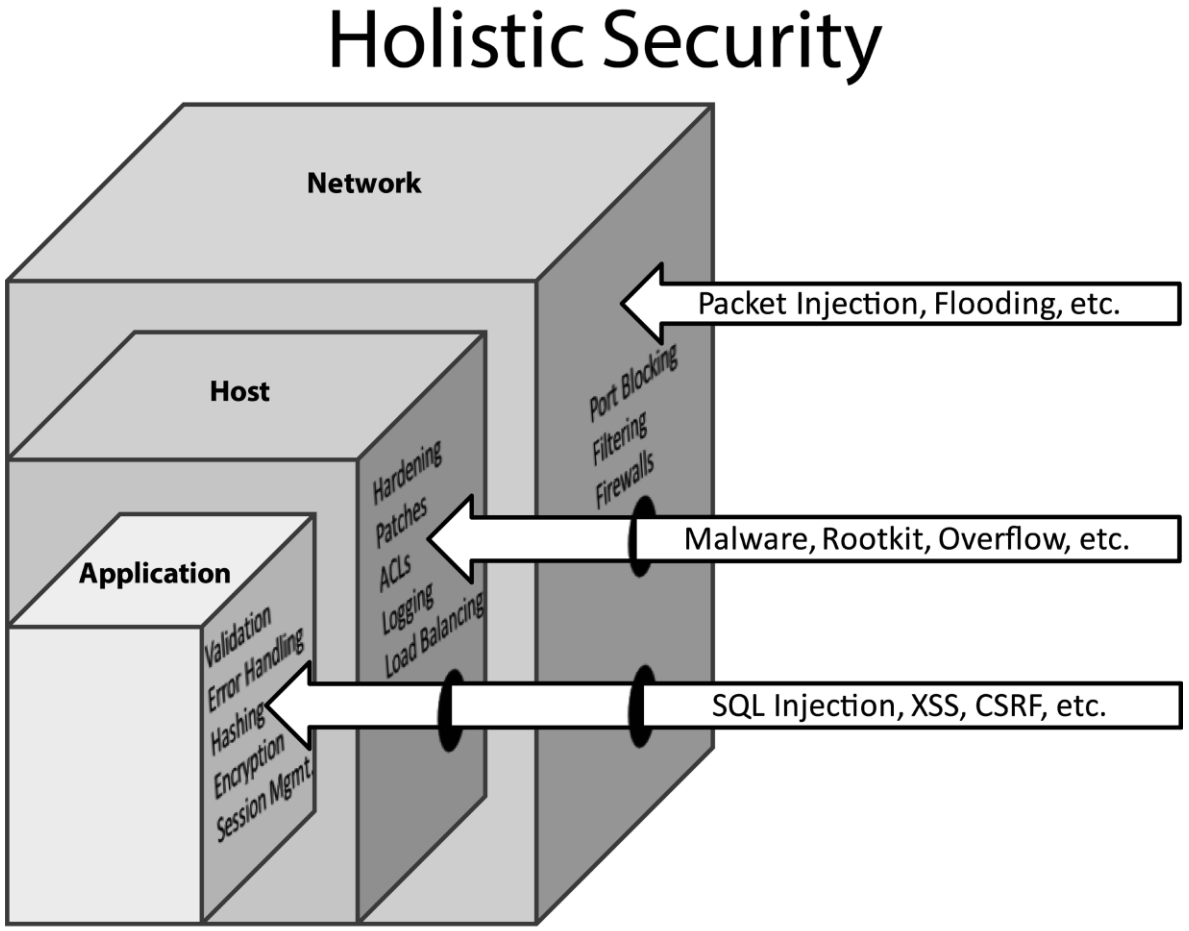
SP 800-12 An Introduction to Computer Security: The NIST Handbook

SECURITY – CORE CONCEPTS

Confidentiality (C)	Confidentiality is the security concept that has to do with protection against unauthorized information disclosure
Integrity (I)	1. Ensure that the data that is transmitted, processed and stored is as accurate as the originator intended 2. The software performs reliably as it was intended to.
Availability (A)	1. The software or the data it processes must be accessible by only those who are authorized (who) 2. The software or the data it processes must be accessible only at the time (when) that it is required. Data must not be available to the wrong people or at the wrong time.

INTRODUCTION - SECURITY CONCEPTS

What to protect?

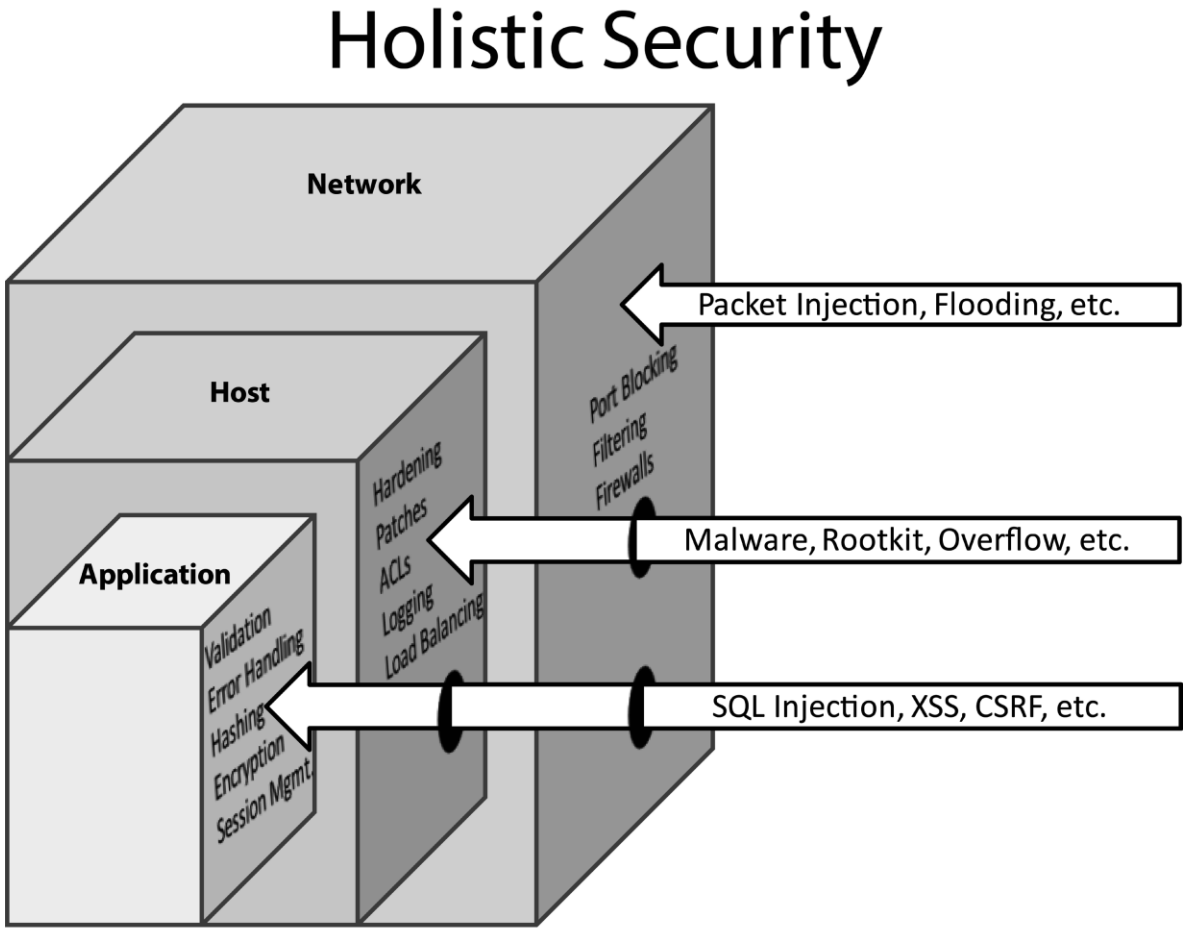


Implementation Challenges

- Project constraints – Time (Schedule), resources (scope) and cost (budget)
- Security as an Afterthought
- Security vs Usability

INTRODUCTION - SECURITY CONCEPTS

What to protect?



Implementation Challenges

- Project constraints – Time (Schedule), resources (scope) and cost (budget)
- Security as an Afterthought
- Security vs Usability

SECURITY – CORE CONCEPTS

Authentication

Identify Human Users, Processes etc

Identity

- Something you know – username and password
- Something you have – Physical lock/key, token
- Something about you (something that you are) – biometrics (Finger prints, voice, retinal scan)

SECURITY – CORE CONCEPTS

Authorization

- **Subject** access **Object**
- Determines access levels of the (**Subject**)
  - Example - User, Process
- **Object** is any resource that is protected
  - Example - a file, a program, an item of data, or any other resource, the authorization system makes the access determination as to grant or deny access.

SECURITY – CORE CONCEPTS

Accountability

- Audit logs (Audit Trials): Build historical record of user actions
- Auditing is a detective and deterrent control

Auditing requirements at the bare minimum must include the following elements

- the identity of the subject (user or process) performing an action (who)
- the action (what)
- the object on which the action was performed (where)
- the timestamp of the action (when)

Repudiation

Denial of a previous action performed

- Accountability with audit trails helps in Non-Repudiation.

Challenges

- Performance impact
- Information overload
- Capacity limitation
- Configuration interfaces protection
- Audit log protection

SECURITY – DESIGN CONCEPTS

Least Privilege

- ✓ The minimum level of access rights (privileges) that is necessary for that person or process to complete an assigned operation.
- ✓ And right must be given only for a minimum amount of time that is necessary to complete the operation.

Separation of Duties or Compartmentalization Principle or Separation of Privilege

- ✓ The successful completion of a single task is dependent upon two or more conditions that need to be met and just one of the conditions will be insufficient in completing the task by itself.

Defense in Depth or Layered Defense

- ✓ Single points of complete compromise are eliminated or mitigated by the incorporation of a series or multiple layers of security safeguards and risk-mitigation countermeasures

Fail Secure or fail safe

- ✓ Aims to maintaining confidentiality, integrity and availability by defaulting to a secure state, rapid recovery of software resiliency upon design or implementation failure

SECURITY – DESIGN CONCEPTS

Economy of Mechanisms or Keep It Simple principle
The likelihood of a greater number of vulnerabilities increases with the complexity of the software architectural design and code. By keeping the software design and implementation details simple, the attackability or attack surface of the software is reduced.

Complete Mediation
A security principle that ensures that authority is not circumvented in subsequent requests of an object by a subject, by checking for authorization (rights and privileges) upon every request for the object. That means, the access requests by a subject for an object is completed mediated each time, every time.

Open Design
The implementation details of the design should be independent of the design itself, which can remain open.

Unlike in the case of security by obscurity wherein the security of the software is dependent upon the obscuring of the design itself.

When software is architected using the open design concept, the review of the design itself will not result in the compromise of the safeguards in the software

SECURITY – DESIGN CONCEPTS

Least Common Mechanisms

The sharing of mechanisms that are common to more than one user or process if the users and processes are at different levels of privilege.

For example, the use of the same function to retrieve the bonus amount of an exempt employee and a non-exempt employee will not be allowed. In this case the calculation of the bonus is the common mechanism.

Psychological Acceptability

- Maximizing the usage and adoption of the security functionality in the software by ensuring that the security functionality is easy to use and at the same time transparent to the user.
- Ease of use and transparency are essential requirements for this security principle to be effective.

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. (Achilles heel)

Leveraging Existing Components

Ensure that the attack surface is not increased and no new vulnerabilities are introduced by promoting the reuse of existing software components, code and functionality.

SECURITY – DESIGN CONCEPTS

Some principles are contradictory to others.
Example :

1. **Economy of Mechanism *vs* Complete Mediation**
2. **Leveraging Existing Components *vs* Leveraging Existing Components**

Architectural decisions must be taken to address these conflicts without compromising the security of the software.

SECURITY - RISK MANAGEMENT

- **Asset** - Assets are those items that are valuable to the organization, the loss of which can potentially cause disruptions in the organization's ability to accomplish its missions. (Tangible and Intangible assets)
- **Vulnerability** - A weakness or flaw that could be accidently triggered or intentionally exploited by an attacker, resulting in the breach or breakdown of the security policy is known as a vulnerability.
- **Threat** - Vulnerabilities pose threats to assets. A threat is merely the possibility of an unwanted, unintended or harmful event occurring. When the event occurs upon manifestation of the threat, it results in an incident.
- **Threat Source/Agent** - Anyone or anything that has the potential to make a threat materialize is known as the threat-source or threat-agent. Threat agents may be human or non-human.
- **Attack** - an intentional action attempting to cause harm is the simplest definition of an attack.
- **Exploit** - When an attack happens as a result of an attacker taking advantage of a known vulnerability, it is known as an 'exploit'. The attacker exploits a vulnerability causing the attacker (threat agent) to cause harm (materialize a threat).
- **Probability** - probability is the chance that a particular threat can happen.
- **Impact** - The extent of how serious the disruptions to the organization's ability to achieve its goal is referred to as the impact.
- **Exposure Factor** - Exposure Factor is defined as the opportunity for a threat to cause loss.

SECURITY - RISK MANAGEMENT

- **Controls** - Security controls are mechanisms by which threats to software and systems can be mitigated. Total Risk - Total risk is the likelihood of the occurrence of an unwanted, unintended or harmful event. This is traditionally computed using factors such as the asset value, threat, and vulnerability. This is the overall risk of the system, before any security controls are applied. This may be expressed qualitatively (e.g., High, Medium or Low) or quantitatively (using numbers or percentiles).
- **Residual Risk** - Residual risk is the risk that remains after the implementation of mitigating security controls (countermeasures or safeguards).
- **Calculation of Risk**
  - Single Loss Expectancy(SLE) = Asset Value ($) x Exposure Factor (%)
  - Annual Rate of Occurrence (ARO) = ALE = Single Loss Expectancy (SLE) x Annualized Rate of Occurrence (ARO)
- **Risk Handling**
  - Ignore the risk
  - Avoid the risk
  - Mitigate the risk
  - Accept the risk
  - Transfer the risk

STRIDE AND DREAD

STRIDE threats can be grouped and categorized into the following six categories.

- **Spoofing** – Impersonating another user or process
- **Tampering** – Unauthorized alterations that impact integrity
- **Repudiation** – Cannot prove the action; deniability of claim
- **Information Disclosure** – Exposure of information to unauthorized user or process that impact confidentiality
- **Denial of Service** – Service interruption that impacts availability
- **Elevation of privilege** – Unauthorized increase of user or process rights

DREAD is a risk calculation or rating methodology that is often used in conjunction with STRIDE.

- ***Damage potential*** – What will be the impact upon exploitability?
- ***Reproducibility*** – What is the ease of recreating the attack/exploit?
- ***Exploitability*** – What minimum skill level is necessary to launch the attack/exploit?
- ***Affected users*** – How many users will be potentially impacted upon a successful attack/exploit?
- ***Discoverability*** – What is the ease of finding the vulnerability that yields the threat?

Questions

THANK YOU

ENTERPRISE APPLICATION AND SECURITY FRAMEWORKS

- Zachman Framework
- Control Objectives for Information and related Technology (COBIT®)
- Committee of Sponsoring Organizations (COSO)
- Sherwood Applied Business Security Architecture (SABSA)