

Lecture Week 3

Semester 2 2023



UNIVERSITY OF
CANBERRA

SOFTWARE SYSTEMS ARCHITECTURE UG/G (11491/8746)

Introduction

Richa Awasthy

richa.awasthy@canberra.edu.au

ACKNOWLEDGEMENT OF COUNTRY

The University of Canberra acknowledges the Ngunnawal peoples as the traditional custodians of the land upon which the University's main campus sits. I pay my respect to all Elders past, present and emerging.



- 1. LOGIN TO MYUC**
- 2. GIVE A RATING**
- 3. LEAVE COMMENTS**

ISEQ is now open

We want to understand The Why behind your agree/disagree rating!

The comments section of each ISEQ survey is a useful tool for you to provide constructive feedback about your student experience. It only takes a couple of minutes to add in your comments.

Your de-identified comments are provided to the Faculty to help us understand what is working well and what needs to be improved in our units.

So go ahead...we are listening!

ISEQ closes on Sunday night.
Access the questionnaire through MyUC.



WEEK 3: AGENDA

1. Quality Attributes

- Understanding Quality Attributes
- Quality Attributes Scenario

2. Security

- Scenarios
- Tactics

1. Quality Attributes

Architecture and Requirements

System requirements can be categorized as:

Functional Requirements

- State what the system must do or how it must behave or react to run-time stimuli
- Functionality is the ability of a system to do the work for which it was intended
- Functionality does not determine architecture

Quality Attribute Requirements

- Annotate (qualify) functional requirements
- Qualification might be how fast the function must be performed, how resilient it must be to erroneous input, how easy the function is to learn, etc.

Constraints

- Design decisions that have already been made for you
- They cannot be changed, so architecture has to be built on top of the constraints

Main Quality Attributes



Quality Attribute Considerations

- Suppose that a functional requirement is "*when the user presses the 'Confirm' button the 'Options' dialog appears.*"
- A few quality attributes that could be related to this functional requirement are:
 - Performance:** how quickly the dialog will appear
 - Availability:** how often this function will fail; how quickly it will be repaired
 - Usability:** how easy it is to learn this function

Quality Attribute Scenarios

Specifying Quality Attributes Requirements

- Quality attributes requirements can be properly specified by building *quality attribute scenarios*
- Scenarios force the architect to specify well defined, testable and verifiable quality attributes
 - Avoiding overlaps, ambiguity
 - Avoiding meaningless definitions

Quality Attribute Scenarios

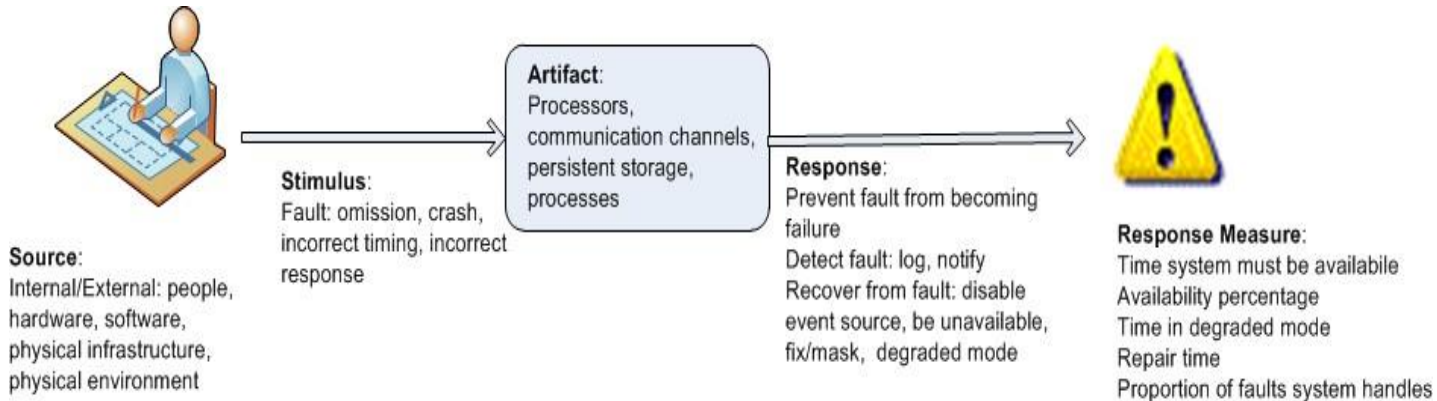
Source of Stimulus	<ul style="list-style-type: none">• Entity (human, computer system, or any other actuator) who generated the stimulus.
Stimulus	<ul style="list-style-type: none">• Condition that requires a response when it arrives at a system.
Environment	<ul style="list-style-type: none">• Conditions under which the stimulus occurs, e.g., system in normal operation or overload.
Artifact	<ul style="list-style-type: none">• Element being stimulated, e.g., a collection of systems, a system, or parts of a system.
Response	<ul style="list-style-type: none">• Activity undertaken as the result of the arrival of the stimulus.
Response Measure	<ul style="list-style-type: none">• The response should be measurable in some fashion so that the requirement can be tested.

Specifying Quality Attributes Requirements

- General quality attribute scenarios (general scenarios):
 - Independent of the system, and could, potentially, describe requirements of any system
- Concrete quality attribute scenarios (concrete scenarios):
 - Specific to the particular system under consideration

Specifying Quality Attribute Requirements

Example: general scenario for availability



Achieving Quality Attributes Through Tactics



Tactics:

- Collection of primitive design techniques that an architect can use to achieve a specified quality attribute response.
- Tactics, like design patterns, are techniques that architects have been using for years.
- We do not invent tactics. We simply use them to solve our problems.

Guiding Quality Design Decisions



- Architecture design is a systematic approach to making design decisions.
- There are several categories of decisions to make
 - ✓ Allocation of Responsibilities
 - ✓ Coordination Model
 - ✓ Data Model
 - ✓ Management of Resources
 - ✓ Mapping Among Architectural Elements
 - ✓ Choice of technology decisions

2. Security

What is Security?

Security is a measure of the system's ability to protect data and information from unauthorized access while still providing access to people and systems that are authorized.

An action taken against a computer system with the intention of doing harm is called an attack.

Scenarios

Characteristics

Confidentiality

- Data or services are protected from unauthorized access.

Integrity

- Data or services are not subject to unauthorized manipulation.

Availability

- The system is available for legitimate use.

Authentication

- Verifies the identities of the parties to a transaction (check if they are truly who they claim to be).




Non-repudiation

- Guarantees that the sender of a message cannot later deny having sent it and that the recipient cannot deny having received the message.

Authorisation

- Grants a user the privileges to perform a task.

Security General Scenario

Source	Human or another system which may have been previously identified (either correctly or incorrectly) or may be currently unknown. A human attacker may be from outside the organization or from inside the organization.
Stimulus	Unauthorized attempt is made to display data, change or delete data, access system services, change the system's behaviour, or reduce availability.
Artifact	System services; data within the system; a component or resources of the system; data produced or consumed by the system.
Environment	The system is either online or offline, connected to or disconnected from a network, behind a firewall or open to a network, fully operational, partially operational, or not operational.
Response	
Response Measure	 

Sample Concrete Security Scenario UNIVERSITY OF CANBERRA

A disgruntled employee from a remote location attempts to modify the pay rate table during normal operations. The attack is detected under a minute. The data is not compromised. The system maintains an audit trail and the admin is informed within a day.

Source	Human attacker from outside the organization (disgruntled employee from a remote location).
Stimulus	Unauthorized attempt is made to modify data and change the system's behaviour.
Artifact	Data consumed by the system.
Environment	Online system during normal operations. The system is connected to a network, behind a firewall and fully operational.
Response	Data is protected from unauthorized access. Attempts to access data are recorded and the system maintains an audit trail. The admin is informed.
Response Measure	The attack is detected under 1 minute. The admin is informed within a day.

Tactics

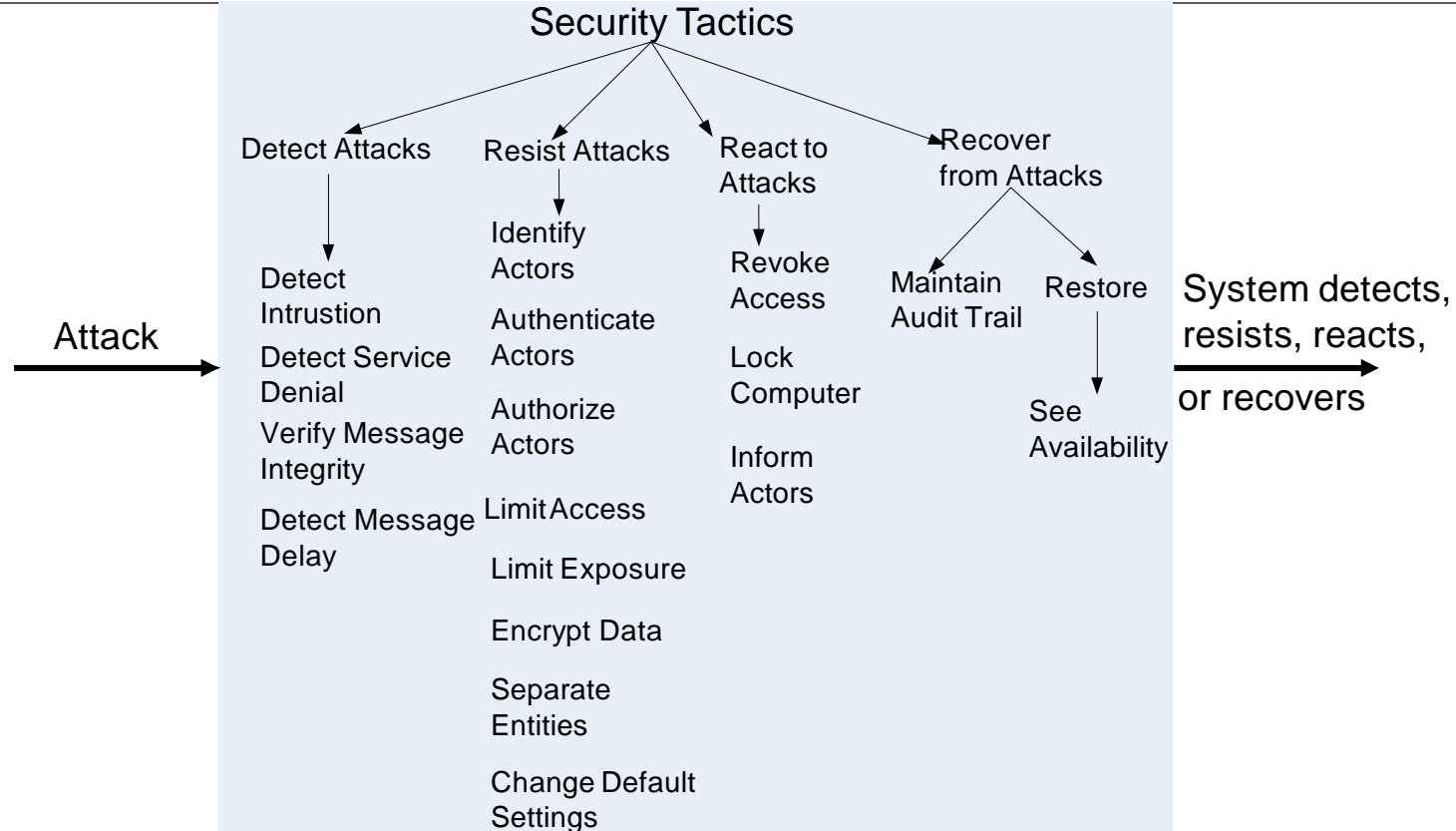
Goal of Security Tactics

Security:

- Secure installations have limited access to them (e.g., by using security checkpoints),
- have means of detecting intruders (e.g., by requiring legitimate visitors to wear badges),
- have deterrence mechanisms such as armed guards,
- have reaction mechanisms such as automatic locking of doors, and
- have recovery mechanisms such as off-site back up.

This leads to four categories of tactics: detect, resist, react, and recover.

Security Tactics



Detect Attacks

Detect Intrusion

Compare network traffic or service request patterns within a system to a set of signatures or known patterns of malicious behavior

Detect Service Denial

Compare the pattern or signature of network traffic coming into a system to historic profiles of known Denial of Service (DoS) attacks

Verify Message Integrity

Use techniques such as checksums or hash values to verify the integrity of messages, resource files, deployment files, and configuration files

Detect Message Delay

Checking the time that it takes to deliver a message, it is possible to detect suspicious timing behavior

Resist Attacks

Identify Actors

Identify the source of any external input to the system

Authenticate Actors

Ensure that an actor (user or a remote computer) is actually who or what it purports to be

Authorize Actors

Ensure that an authenticated actor has the rights to access and modify either data or services

Limit Access

Limit access to resources such as memory, network connections, or access points

Resist Attacks

Limit Exposure

Minimize the attack surface of a system by having the fewest possible number of access points

Encrypt Data

Apply some form of encryption to data and to communication

Separate Entities

Physical separation on different servers & networks, virtual machines, or “air gap”

Change Default Settings

Force the user to change settings assigned by default

React to Attacks

Revoke Access

Limit access to sensitive resources, even for normally legitimate users and uses, if an attack is suspected

Lock Computer

Limit access to a resource if there are repeated failed attempts to access it

Inform Actors

Notify operators, other personnel, or cooperating systems when an attack is suspected or detected

Recover from Attacks

Audit

Keep a record of user and system actions and their effects, to help trace the actions of, and to identify, an attacker

In addition to Audit, check the *Availability* tactics for recovery of failed resources

References:

- Len, Bass, Clements Paul, and Kazman Rick. (2013) "Software architecture in practice." Boston, Massachusetts Addison. 3rd Edition.
 - CHAPTER 4 - Understanding Quality Attributes
 - CHAPTER 9 - Security

References

- Len, Bass, Clements Paul, and Kazman Rick. (2013) "Software architecture in practice." Boston, Massachusetts Addison. 3rd Edition.
 - CHAPTER 4 - Understanding QualityAttributes
 - CHAPTER 9 - Security

