DCML-CPS - Module 3

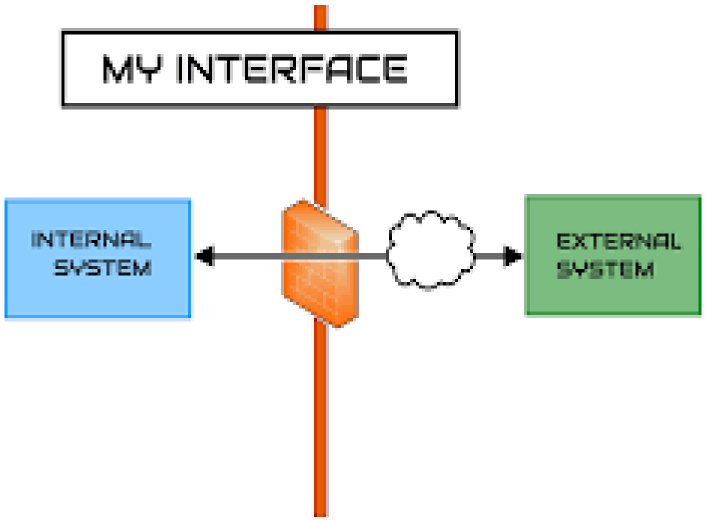
Testing Mechanisms

Tommaso Zoppi
University of Trento - Povo (IT)
tommaso.zoppi@unifi.it
tommaso.zoppi@unitn.it

Robustness Testing

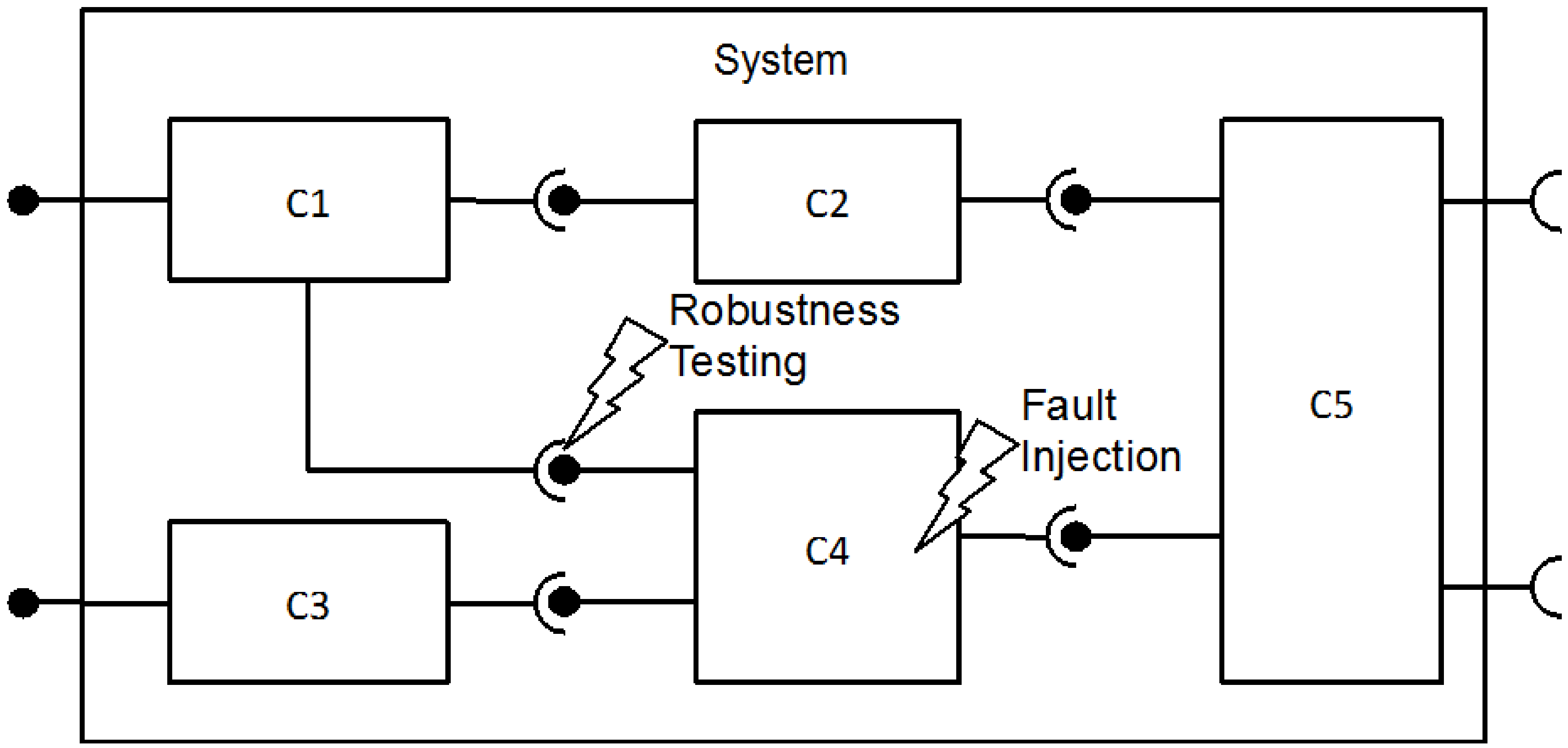
Robustness Testing

- In the case of systems based on components, the error injection can corrupt the values at the component interfaces (**Robustness Testing**)



It is the only way to test a black-box system / application!

Robustness Testing VS Fault Injection



Robustness Testing (2)

- Robustness Testing is a form of Fault Injection which aims to assess the robustness of a system

- It artificially introduces non-compliant input or limit conditions, which represent **external faults**
  - e.g., inputs that simulate invalid values from malicious users, external disturbances or malfunctioning components
- The Robustness Testing aims to identify **vulnerabilities** in the target in properly managing external faults
  - e.g., Missing checks on input values

Definition of Invalid Input

- Some approaches for the definition of robustness tests:
  - **Bit-flip**: it corrupts the input parameters of the target by inverting the value of a bit in the parameter
  - **Fuzzing**: random generation of input
  - **Data-based type**: of each input variable, it identifies a subset of values of its domain that typically are not valid for the data type
    - e.g.,0-value for memory addresses
- They differ in the number of tests, complexity of their implementation, effectiveness in finding vulnerabilities

The CRASH Scale

- Robustness uses the CRASH scale:
  - Catastrophic
    - App becomes corrupted or the machine crashes or reboots
  - Restart
    - Application hangs and must be force-closed
  - Abort
    - Abnormal termination of the application
  - Silent
    - No error is indicated on an operation
      cannot be performed
  - Hindering
    - The error code returned is not correct

Services robustness testing

- ▸ Erroneous call parameters
  - Generated using a set of predefined rules
  - Based on the **data types** of each parameter
  - Injected during execution

GetWeather (city, day) → **GetWeather ("Florence", null)**

- ▸ Key components needed:
  - Workload
  - Robustness tests
  - Failure modes classification

Key components

- ► Workload
  - Actions that the service must perform during the benchmark
- ► Robustness tests
  - Faultload consisting of a set of **invalid call parameters**
  - Applied to the target services to expose robustness problems
- ► Failure modes classification
  - Characterize the behavior of the service while executing the workload in the presence of the robustness tests

Testing procedure

1. Tests preparation
   1.1. Analysis of the services to gather informati
   1.2. Workload generation
2. Tests execution
   2.1. Execution of the workload
   2.2. Execution of the tests to trigger faulty behaviors
3. Analysis
   3.1. Analysis of the output of the tests
   3.2. Robustness problems identification/classification

Rules for Tests Generation

- Null and empty values
  - e.g., null string, empty string
- Valid values with special characteristics
  - e.g., nonprintable characters in strings, valid dates by the end of the millennium
- Invalid values with special characteristics
  - e.g., invalid dates using different formats
- Maximum and minimum valid values in the domain
  - e.g., maximum valid value for the parameter, minimum valid value for the parameter

Mutation rules

Parameter type	Description
String	Set a null value
	Set an empty string
	Set a predefined string
	Set a string with nonprintable characters
	Set an alphanumeric string
	Add characters to overflow maximum size
Number	Set 0
	Set +1
	Set -1
	Set maximum number valid for the type
	Set minimum number valid for the type
	Set maximum number valid for the domain
	Set minimum number valid for the domain
	Set maximum number in the domain plus one
	Set minimum number in the domain minus one
...	...
Object	Set a null object
	Set a non serializable object
	Set a correct target class empty object
	Set an objectified primitive datatype (Boolean, Byte, Short,Iint, Float, Double and String)
	Set common dataype (List, Map and Date)

In a Nutshell…

- Faulty services are frequently deployed
  - Unacceptable situation for providers and also for clients who are seeking for trustable provision of service
- Robustness problems may lead to security issues
- Robustness testing is essential when developing a services infrastructure
  - Test and fix services code
  - Select alternative services
  - Build wrappers to mitigate robustness problems

Security vulnerabilities

- Robustness testing can also be used to detect security vulnerabilities, e.g. using **SQL injection**
  - Using these vulnerabilities, corrupted input values can be used to modify the SQL query made by the application to the DBMS
    - e.g., authentication will pass even if the passwords are not known

What Is SQL Injection_.mp4

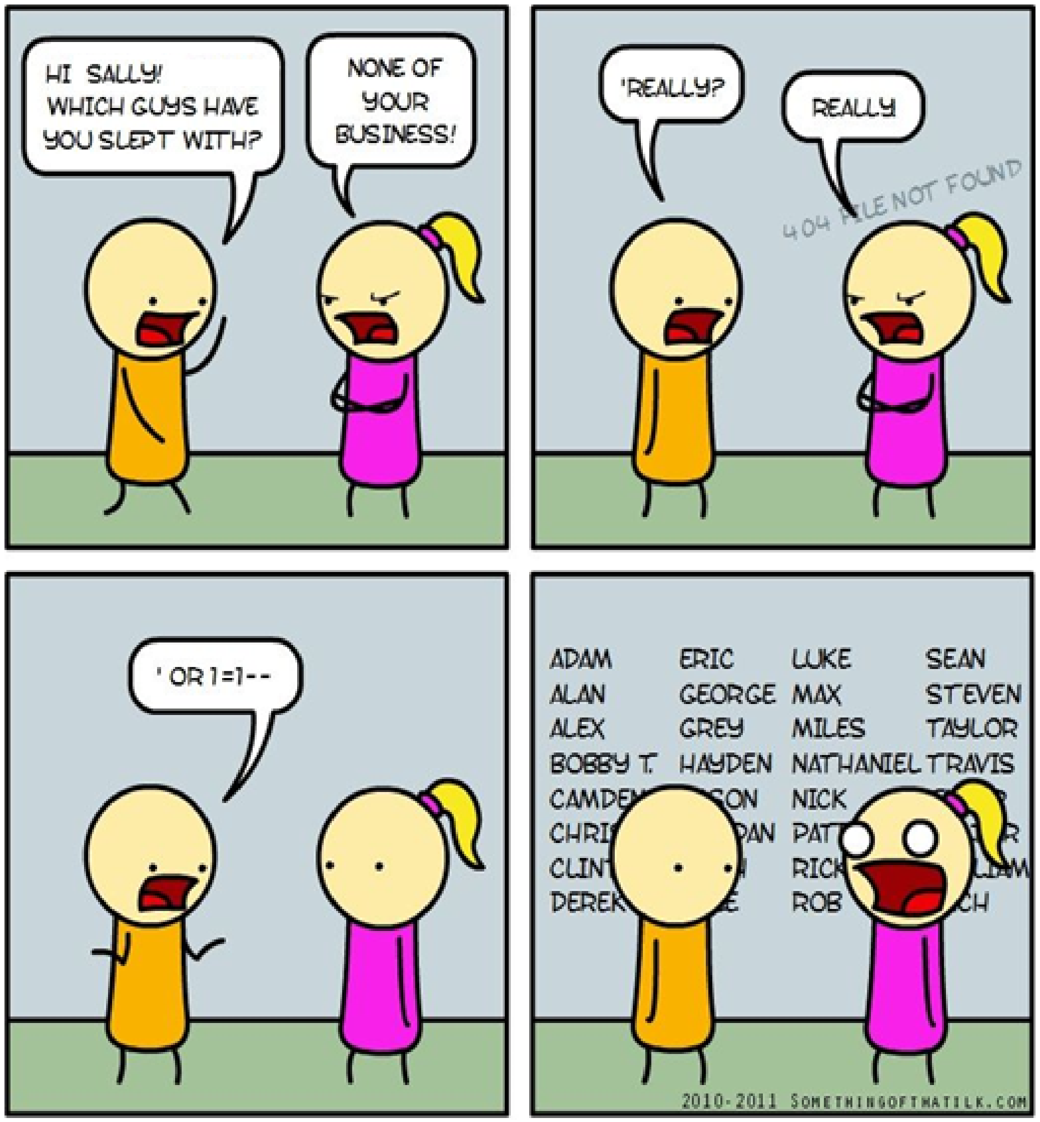
https://www.youtube.com/watch?v=wcaiKgQU6VE

- This vulnerability can be detected by injecting the corrupted input values, and by monitoring the SQL query generated by the application to the DBMS

Testing SQL Injection Vulnerabilities

Codice SQL iniettato
" or 1=1 --
" or 1=1 or ""="
' or (EXISTS)
' or uname like '%
' or userid like '%
' or username like '%
' UNION ALL SELECT
' UNION SELECT
char%2839%29%2b%28SELECT
&quot; or 1=1 or &quot;&quot;=&quot;
&apos; or &apos;&apos;=&apos;

Testing SQL Injection Vulnerabilities


HI SALLY! WHICH GUYS HAVE YOU SLEPT WITH?
NONE OF YOUR BUSINESS!
'REALLY?
REALLY
404 FILE NOT FOUND
' OR 1=1--
ADAM ALAN ALEX BOBBY T. CAMDEN CHRIS CLINT DEREK
ERIC GEORGE GREY HAYDEN
LUKE MAX MILES NATHANIEL NICK PAT RICK ROB
SEAN STEVEN TAYLOR TRAVIS
2010-2011 SOMETHINGOFTHATILK.COM