The SPaCIoS Tool

property-driven and vulnerability-driven security testing for Web-based apps

Alessandro Armando
DIBRIS – University of Genova and Security & Trust – FBK, Trento
(on behalf of the SPaCIoS consortium)

STREP Project number: 257876
Objective ICT-2009.1.4 c: Technology and Tools for Trustworthy ICT
01.10.10 − 31.01.14
www.spacios.eu

Motivations

Problems:

- ★ In order to provide required guarantees to providers, intermediaries, and consumers of distributed services, rigorous security validation techniques must be applied.
- ★ State-of-the-art security validation technologies fail to realise their full potential because they are typically used in isolation.
- ★ Security validation in the Internet of Services must be performed not only at production but also at deployment and consumption times.

Goal

Project objectives and approach:

- ★ Improve security of the Internet of Services by laying the technological foundations of a new generation of security analysers for service deployment, provision and consumption.
- ★ Develop and combine state-of-the-art technologies for penetration testing, security testing, model checking, model extraction and automatic learning, resulting in the SPaCIoS Tool.
- ★ Assess SPaCIoS technology by running SPaCIoS Tool against a set of security testing problem cases drawn from industrial and open-source IoS application scenarios, as well as from an open-source cloud platform developed in another EU project.
- ★ Migrate SPaCIoS technology to industry (SAP and Siemens business units), as well as to industrial interest groups and open-source communities.

Model Checking vs Penetration Testing

	Model Checking	Penetration Testing	The SPaCIoS Ideal
Target of Verification	Abstraction of Actual System (the Model)	Actual System (the System Under Validation, SUV)	1. Use model to test system 2. Use system to discharge spurious attacks 3. Use system to build model
Scope	Design flaws	Implementation flaws	Design and Implementation flaws (and their interaction)
Input	Model + Spec of Sec. Goals & Assumptions	Vulnerabilities to seek (attack surface automatically discovered)	Partial model, sec. goals & assumptions, vulnerabilities (in user friendly notation)
Automation	High	Low	High

Research prototype

- model checking
- security testing
- penetration testing
- …

Complements state-of-the-art

Targets industrially-relevant Security Protocols & Web Apps

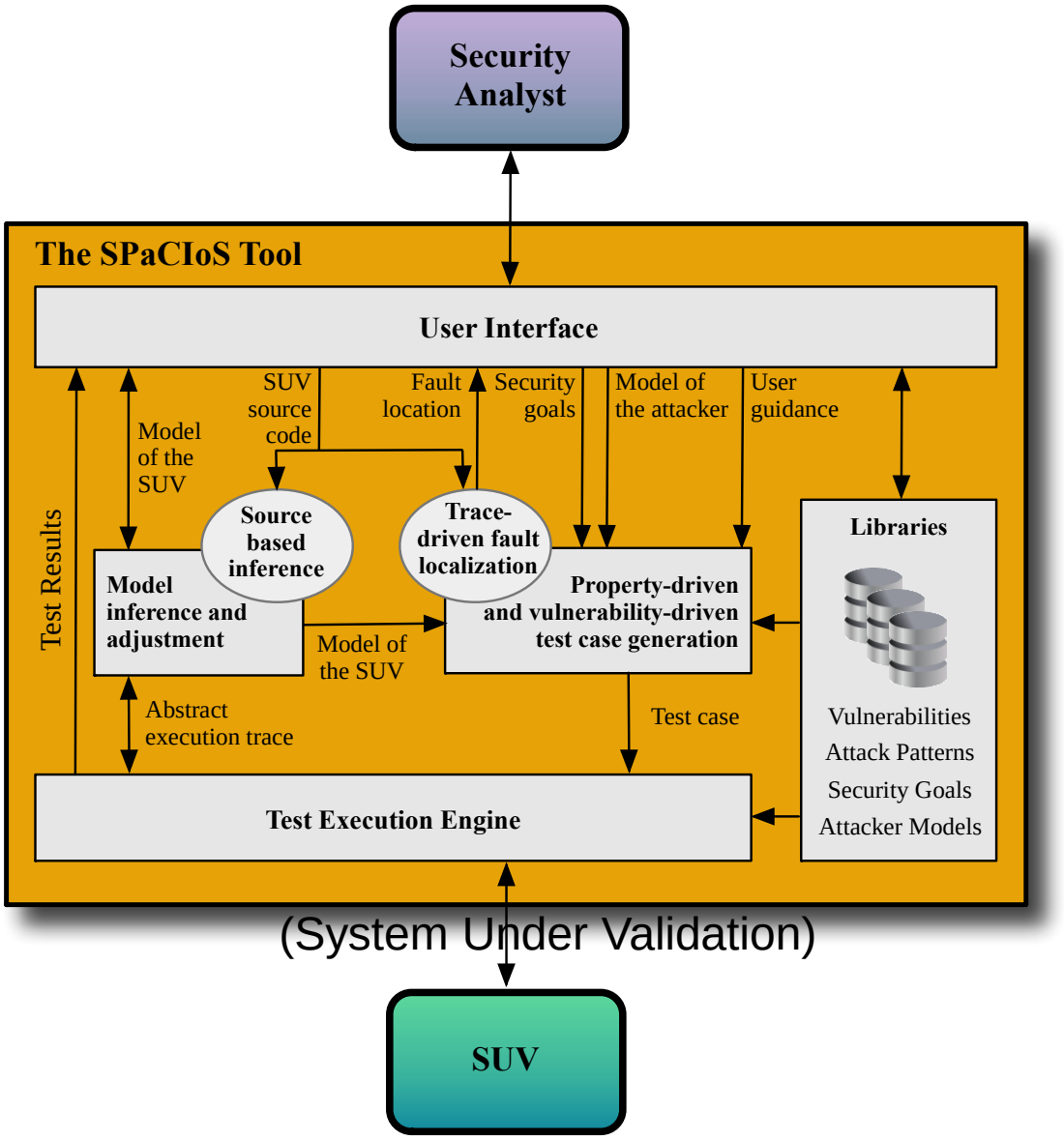
Broad security range

- logic-flaws, injections, AC, …
- good coverage of OWASP top 10

Promising results

- SAML SSO, OAuth2, ..
- WebGoat, Shopping Cart, ..

On-going transfers to SAP and SIEMENS



Technische Universität München

Research prototype

- model checking
- security testing
- penetration testing
- …

Complements state-of-the-art

Targets industrially-relevant Security Protocols & Web Apps

Broad security range

- logic-flaws, injections, AC, …
- good coverage of OWASP top 10

Promising results

- SAML SSO, OAuth2, ..
- WebGoat, Shopping Cart, ..

On-going transfers to SAP and SIEMENS

Pervasive Retail

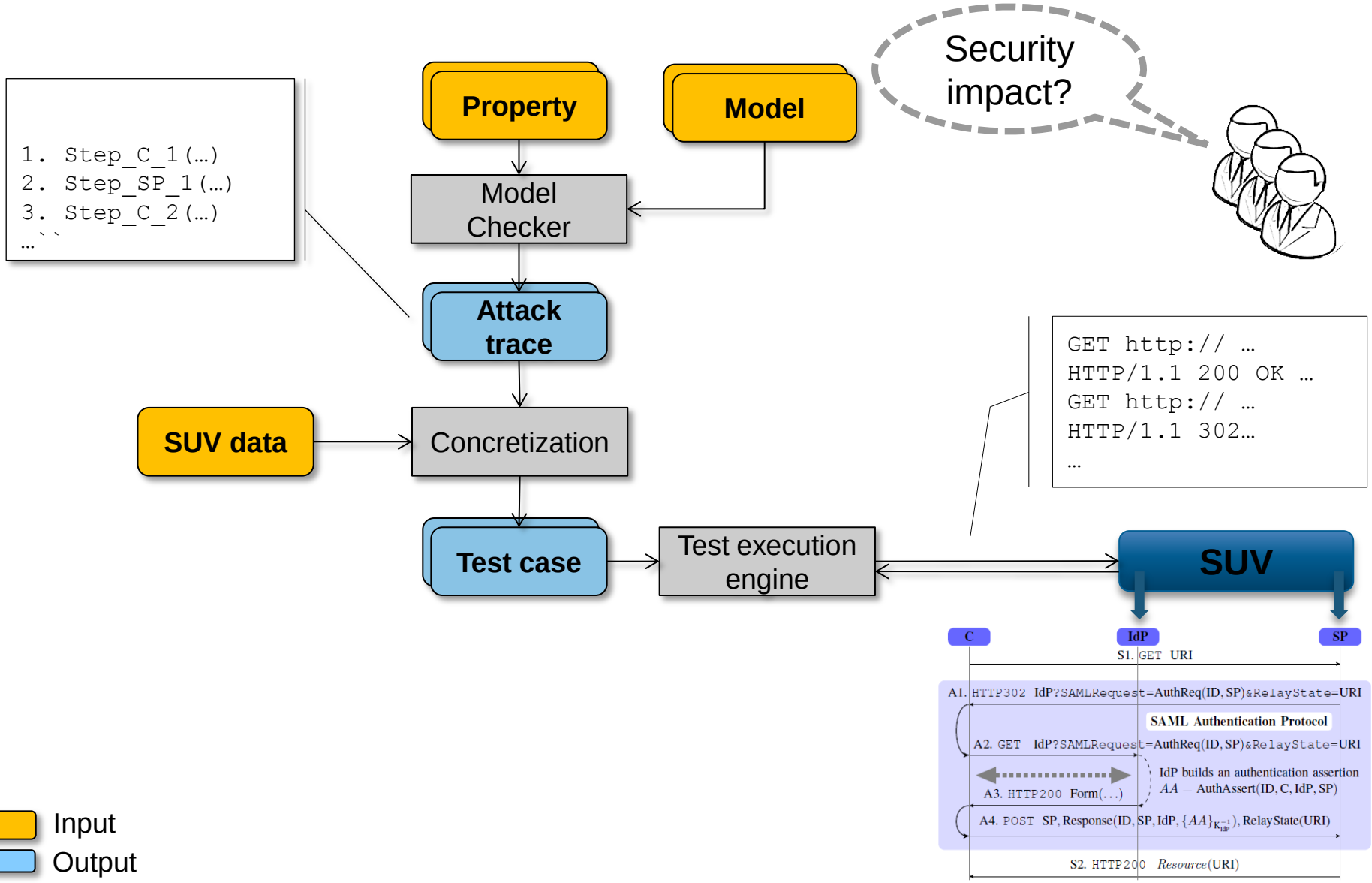
	OWASP Top 10	The SPaCloS Tool
A1	Injection	WebGoat lesson: String SQL Injection WebGoat lesson: Numeric SQL Injection SIEMENS InfoBase and eHealth
A2	Broken Authentication & Session Management	SAML, OpenID, OAuth: e.g., authentication logic-flaws Password brute-forcing on SIEMENS InfoBase and eHealth
A3	Cross-Site Scripting	WebGoat lesson: Stored XSS WebGoat lesson: Reflected XSS SIEMENS InfoCase and eHealth
A4	Insecure Direct Object References	SIEMENS InfoBase and eHealth: File Enumeration and Path Traversal
A5	Security Misconfiguration	WebGoat lesson: Forced Browsing (File Enumeration)
A6	Sensitive Data Exposure	SAML, OpenID, OAuth: data confidentiality logic flaws
A7	Missing Function Level Access Control	WebGoat lesson: Bypass Business Layer Access Control, WebGoat lesson: Bypass Data Layer Access Control WebGoat lesson: Role Based Access Control SIEMENS eHealth
A8	CSRF	SIEMENS InfoBase and eHealth
A9	Using Components with Known Vulnerabilities	
A10	Unvalidated Redirects and Forwards	

Technische Universität München

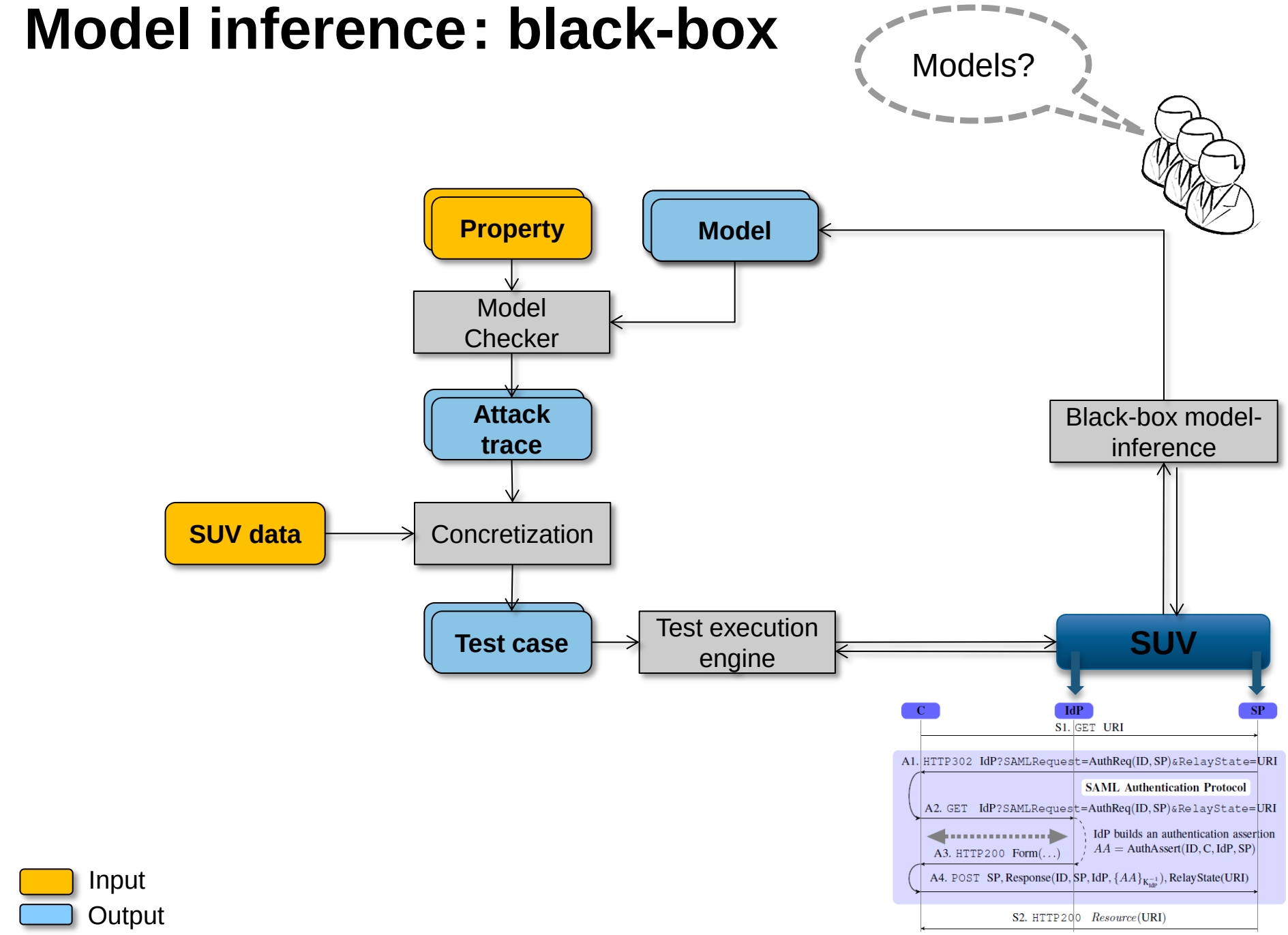
The SPaCloS Tool

- Property-driven Security Testing
- Model Inference
- Mutation-based Testing
- Vulnerabilty-driven Testing

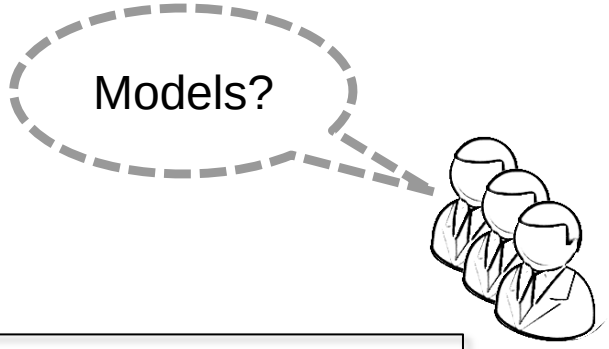
property-driven security testing



Model inference: black-box



Model inference: white-box



Property

Model

Model Checker

Attack trace

SUV data

Concretization

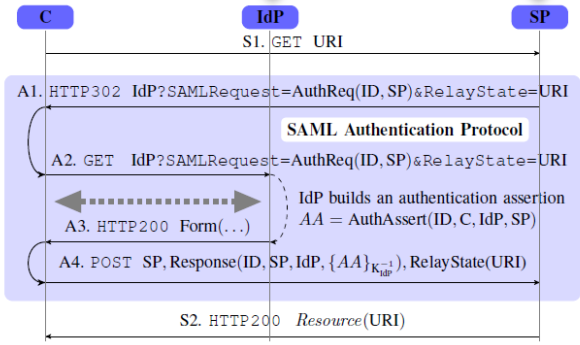
Test case

Test execution engine

SUV

White-box model-inference

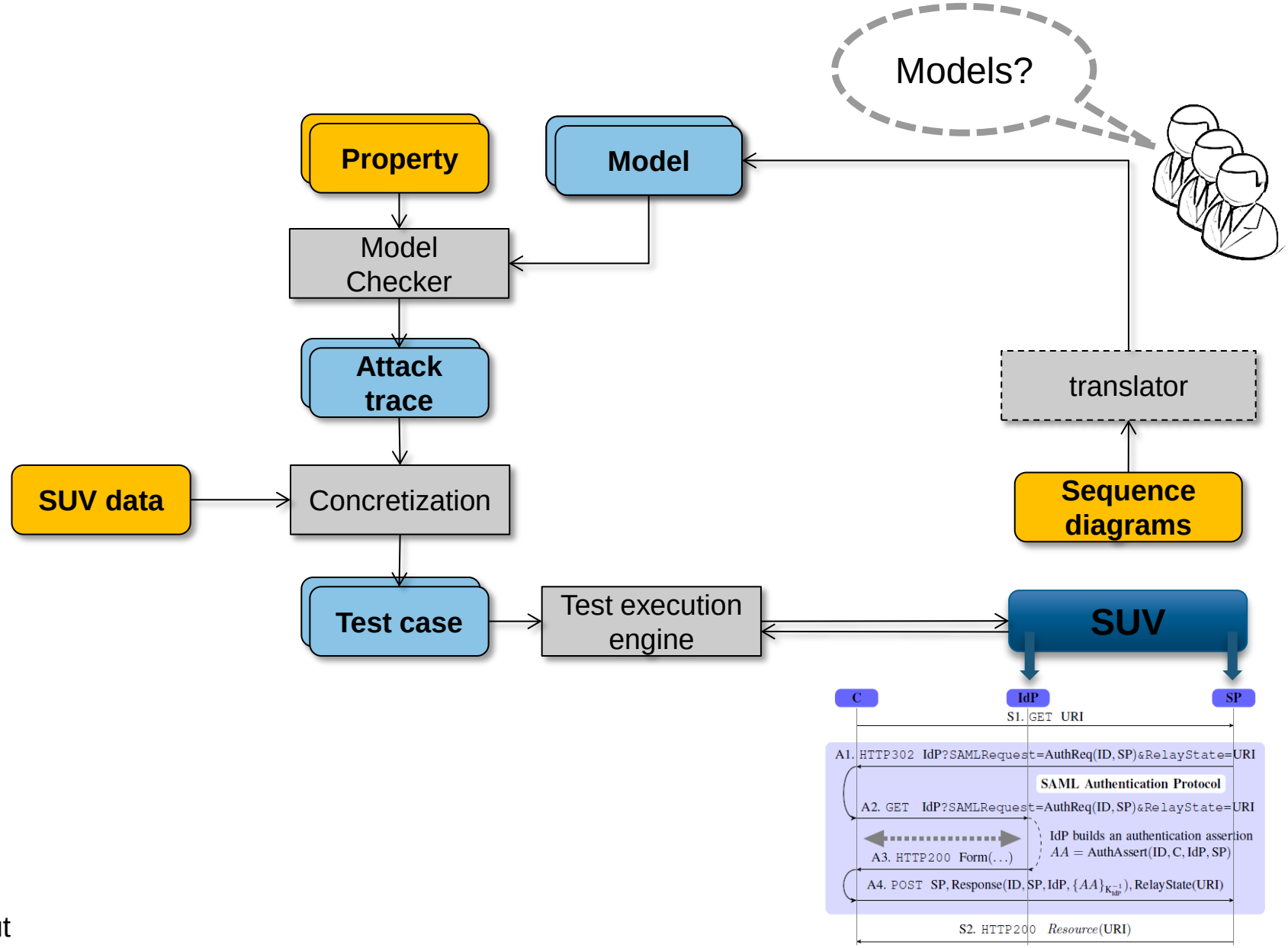
source code of system



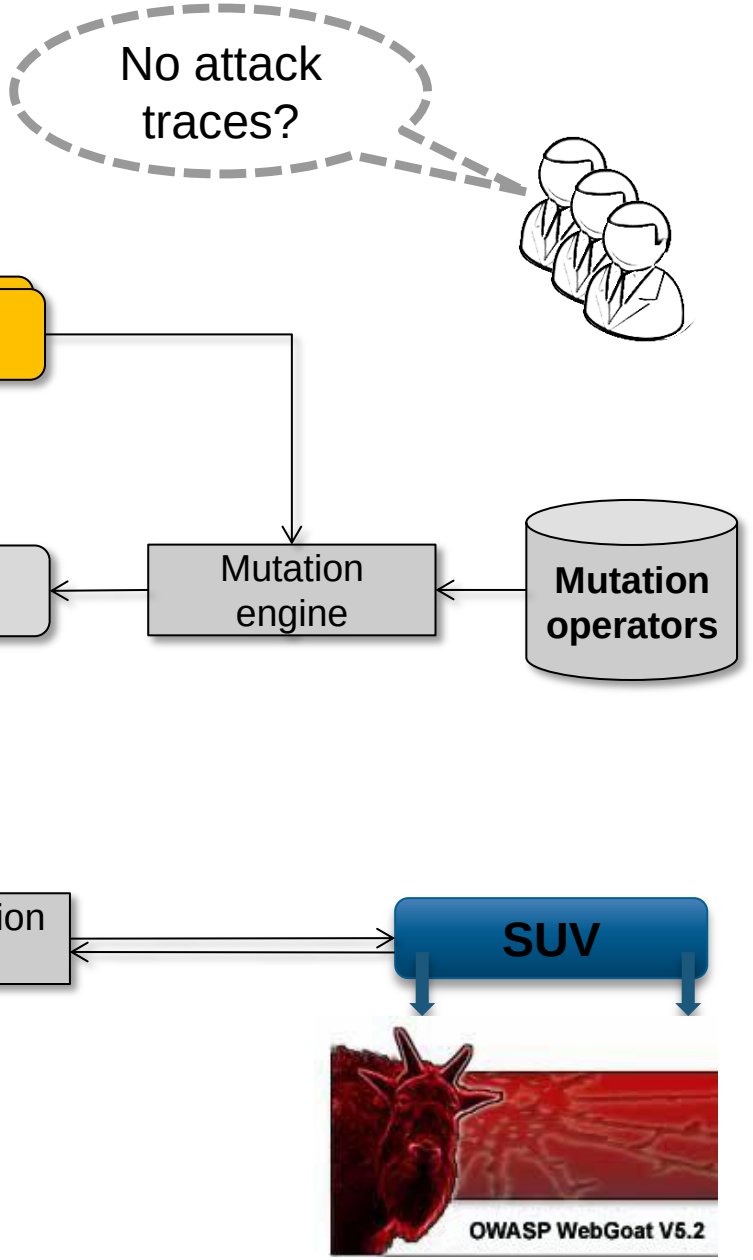
Input

Output

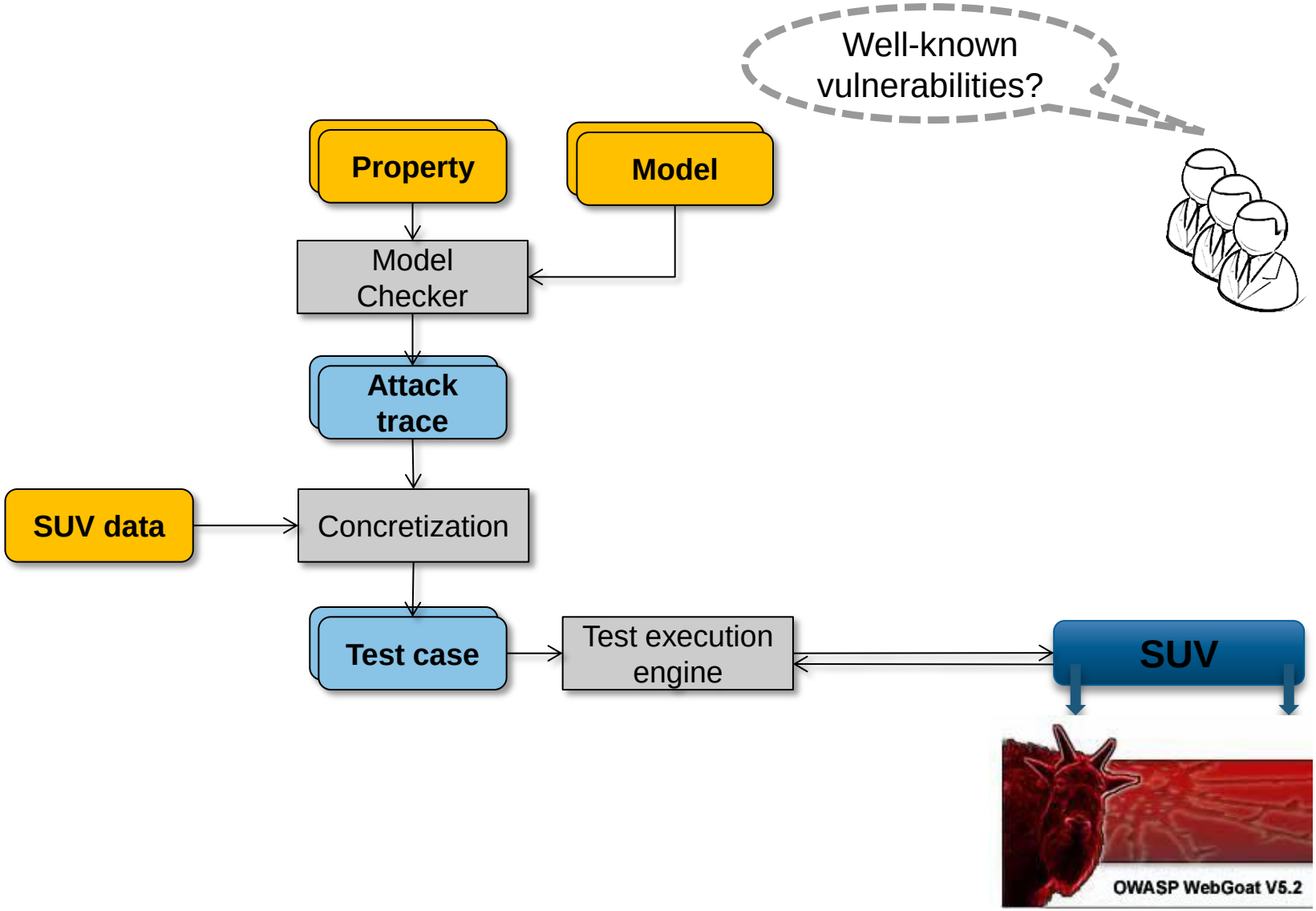
Model inference: sequence diagrams

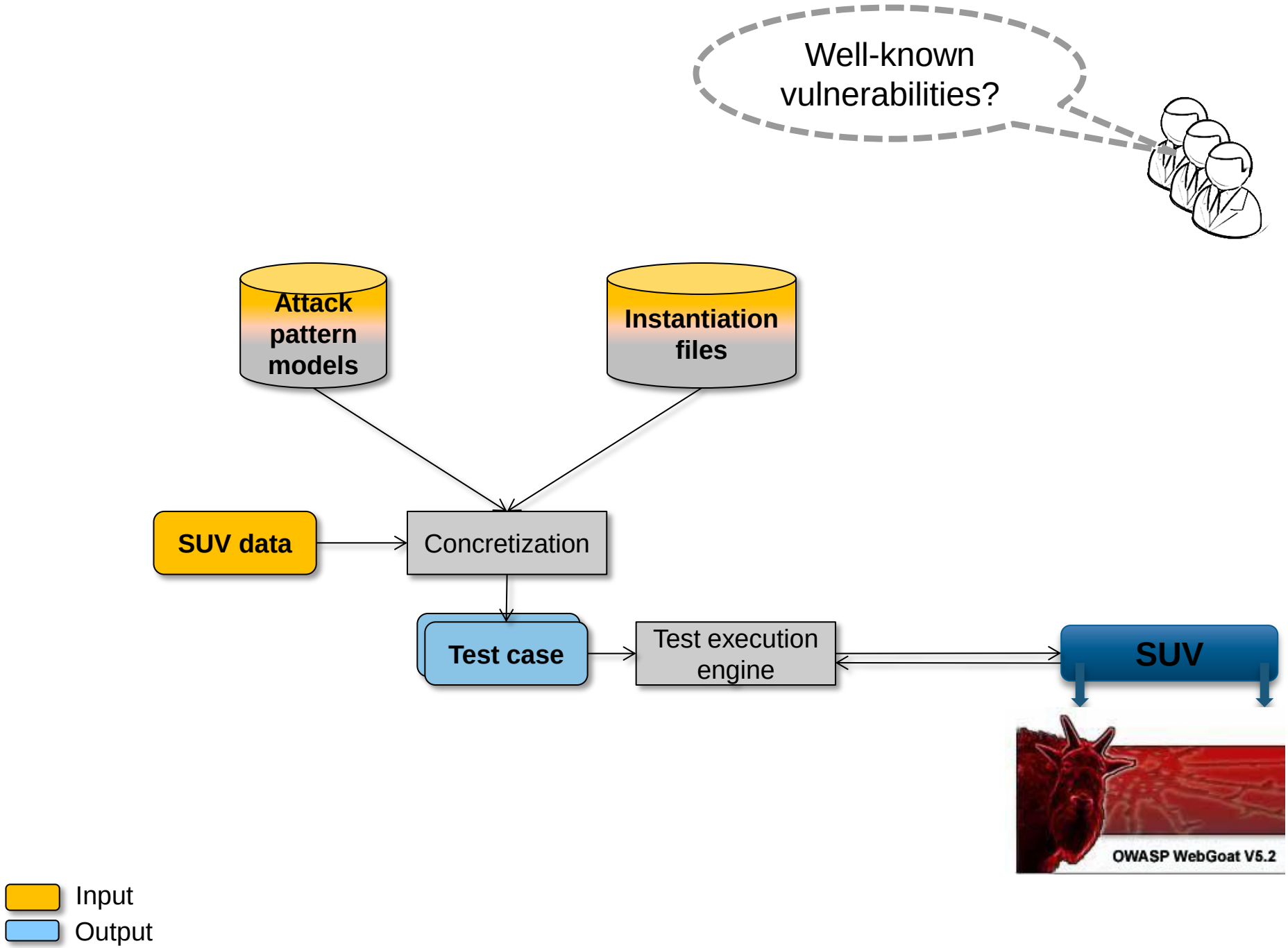


Mutation-based Testing

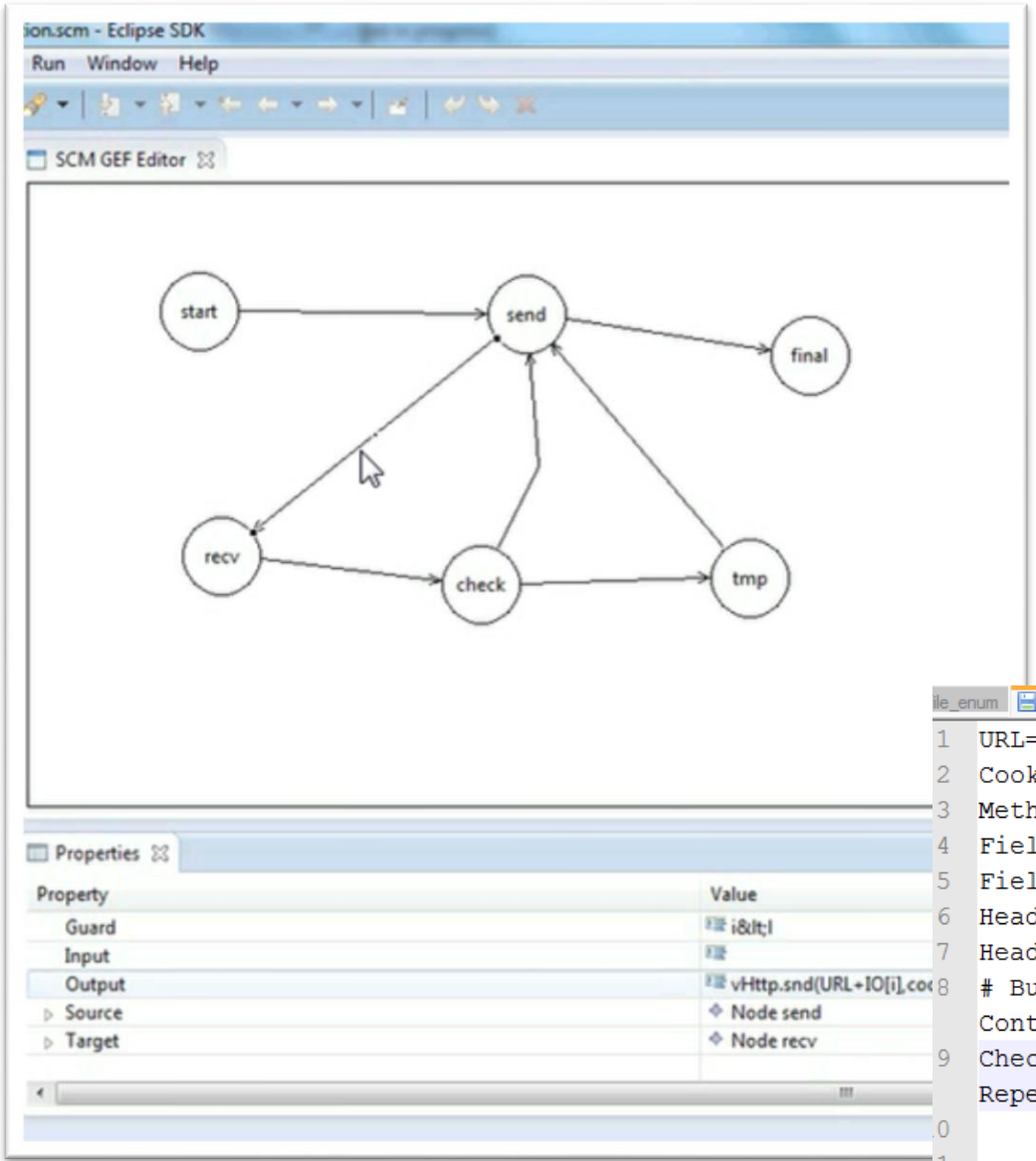


Vulnerability-driven Testing




Well-known vulnerabilities?
Attack pattern models
Instantiation files
SUV data
Concretization
Test case
Test execution engine
SUV
OWASP WebGoat V5.2
Input
Output

Attack Pattern + Instantiation file + SUV data



file_enum

```
IO=[
  ".htaccess",
  ".htaccess.bak",
  ".htpasswd",
  ".meta",
  ".web",
  "conf",
  "apache/logs/access.log ",
  "apache/logs/access_log",
  "apache/logs/error.log ",
  "apache/logs/error_log",
  "httpd/logs/access.log ",
  "httpd/logs/access_log",
  "httpd/logs/error.log ",
  "httpd/logs/error_log",
  "logs/access.log",
  "logs/access.log ",
```

webgoat.conf

```
URL="http://localhost:8086/WebGoat/attack?Screen=37&menu=20
Cookie="JSESSIONID=E6E32C706E8CC910AB0962DDE7FCC1FD"
Method="POST"
Fields1={'employee_id':'105', 'password':'tom', 'action':'L
Fields2={'action':'ViewProfile'}
Header1={'Content-Type': 'application/x-www-form-urlencoded
Header2={'Content-Type': 'application/x-www-form-urlencoded
# Bug in WebGoat: After complete the exercise, the success
Control." will not be always showed, just when one load the
Check_Info=["Stage 4: Add Data Layer Access Control.", "Imp
Repeat stage 3.  Verify that access to other employee's pro
```

	OWASP Top 10	The SPaCIoS Tool
A1	Injection	WebGoat lesson: String SQL Injection WebGoat lesson: Numeric SQL Injection SIEMENS InfoBase and eHealth
A2	Broken Authentication & Session Management	SAML, OpenID, OAuth: e.g., authentication logic-flaws Password brute-forcing on SIEMENS InfoBase and eHealth
A3	Cross-Site Scripting	WebGoat lesson: Stored XSS WebGoat lesson: Reflected XSS SIEMENS InfoCase and eHealth
A4	Insecure Direct Object References	SIEMENS InfoBase and eHealth: File Enumeration and Path Traversal
A5	Security Misconfiguration	WebGoat lesson: Forced Browsing (File Enumeration)
A6	Sensitive Data Exposure	SAML, OpenID, OAuth: data confidentiality logic flaws
A7	Missing Function Level Access Control	WebGoat lesson: Bypass Business Layer Access Control, WebGoat lesson: Bypass Data Layer Access Control WebGoat lesson: Role Based Access Control SIEMENS eHealth
A8	CSRF	SIEMENS InfoBase and eHealth
A9	Using Components with Known Vulnerabilities	
A10	Unvalidated Redirects and Forwards	

Research prototype

- model checking
- security testing
- penetration testing
- …

Complements state-of-the-art

Targets industrially-relevant Security Protocols & Web Apps

Broad security range

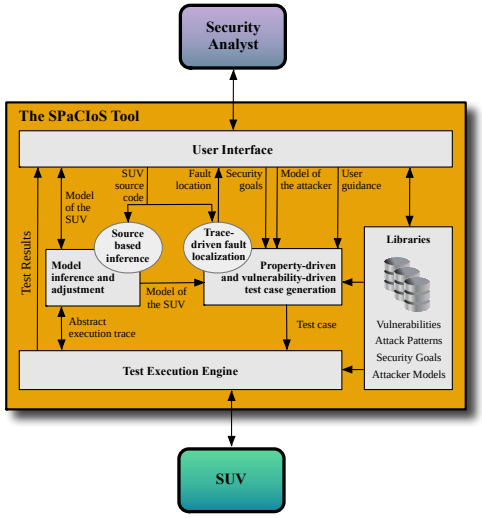
- logic-flaws, injections, AC, …
- good coverage of OWASP top 10

Promising results

- SAML SSO, OAuth2, ..
- WebGoat, Shopping Cart, ..

On-going transfers to SAP and SIEMENS

Thank you!



The SPaCIoS Tool is available for public download at http://www.spacios.eu