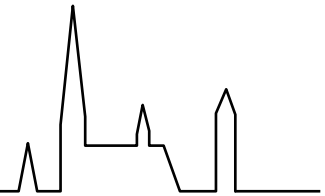
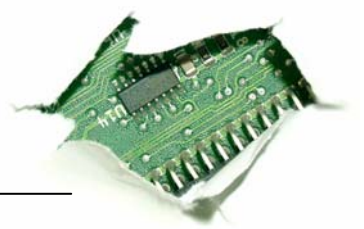
Vulnerability Analysis for Web Services-based Business Processes

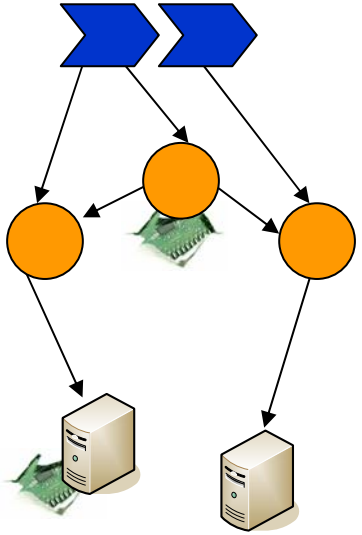
Lutz Lowis
Department of Telematics
Institute of Computer Science and Social Studies (IIG)
University of Freiburg, Germany
lowis@iig.uni-freiburg.de
http://www.telematik.uni-freiburg.de



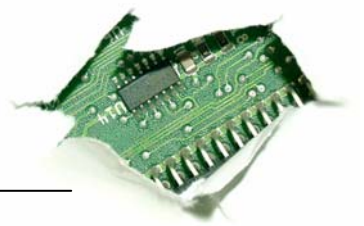
Vulnerability Example

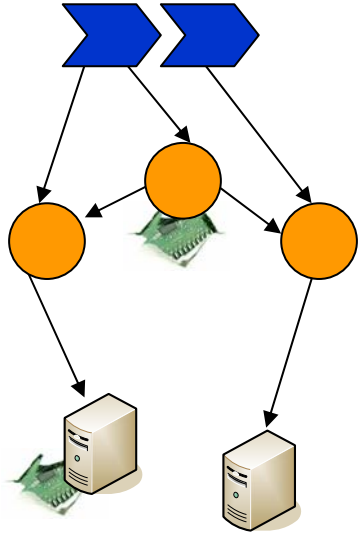


- XML injection vulnerability
- Business process:
  - order a book
  - claim travel expenses
  - order a travel expense claim
- Attacker collects bonus



Scenario for Analysis



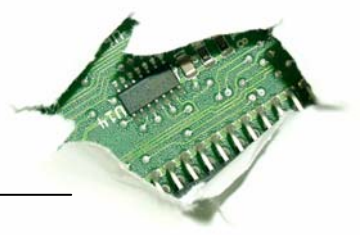


Business process as workflow

Web Services, some with vulnerabilities
(e.g., XML injection)

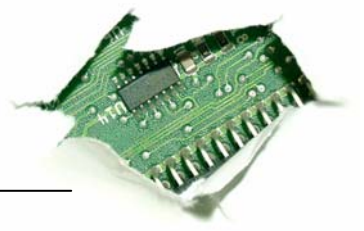
IT components, some with vulnerabilities
(e.g., buffer overflow)

Outline



- Vulnerability Analysis Methods
  - static and dynamic
  - bottom-up and top-down
- Analysis of Web Service Compositions
  - challenge
  - prototype design
- Use in Risk Management and Compliance

Static Analysis



- Analyze source code to reveal dangerous paths/actions (also used to answer information flow questions)

- Will find many vulnerable code sequences…

```
$input=$_GET[´id´];
$sql=´SELECT * FROM users WHERE id=´.$input;
mysql_query($sql);
```

- …but not all of them…

  - vulnerable patterns need to be known

  - false negatives are possible

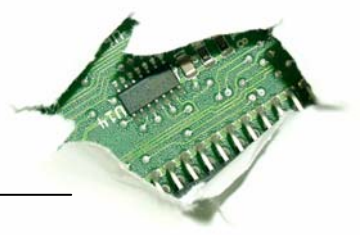
- …or more than there are

  - false positives are also possible

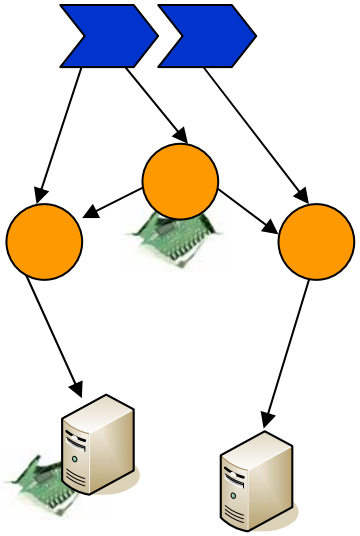
```
$input=$_GET[´id´];
$sql=somefunction($input);
mysql_query($sql);
```

order a book:
book title *might* be altered

Dynamic Analysis



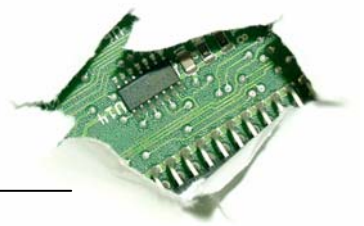
- Analyze program at runtime
- Less/no false alarms thanks to runtime information
- Utilizing static analysis results “is a must” (big picture)



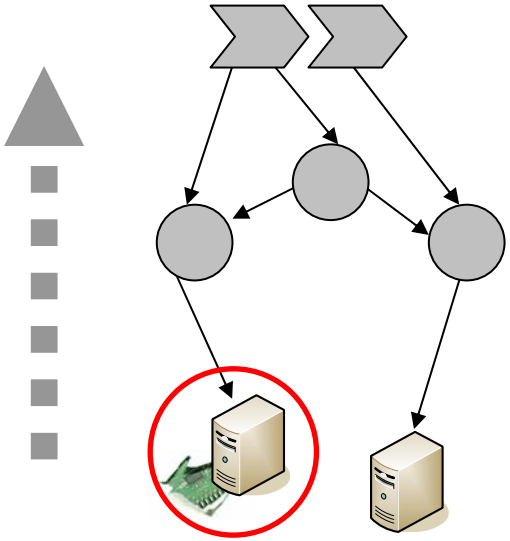
Static analysis shows possible violation
Dynamic analysis rings alarm

order a book:
book title *is about to be* altered

Bottom-up Analysis



- Failure Mode and Effects Analysis (FMEA)
- Given a failure (or an exploited vulnerability), what are the effects on the system?



…the workflow and how?

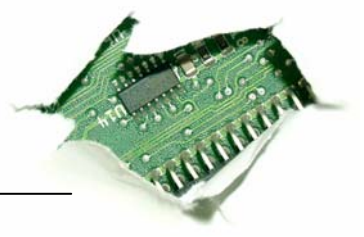
…which services will affect…

If a component fails…

- Highly complex, even for relatively small amounts of components and vulnerabilities

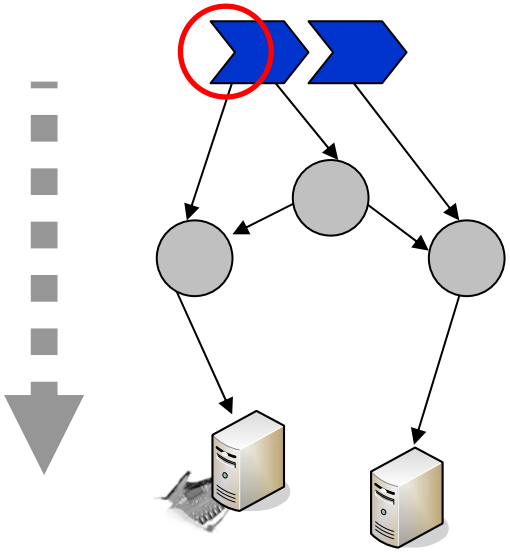
claim travel expenses:
sql injection *might* alter name

Top-down Analysis



- Attack Trees

- Given an attack (or an exploited vulnerability), what are the steps to get there?



How could an attacker…

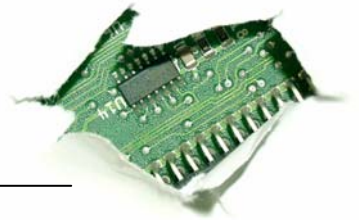
…violate the security of services…

…or other components?

- More likely to find combined vulnerabilities

claim travel expenses:
integrity *might* be harmed through sql injection

Attack Graphs



Oleg M. Sheyner: “Scenario Graphs and Attack Graphs“, PhD Thesis, CMU, 2004.

Input:
- attack(s)
- hosts and services
- connectivity
- vulnerabilities

Output:
- attack model
- attack graph

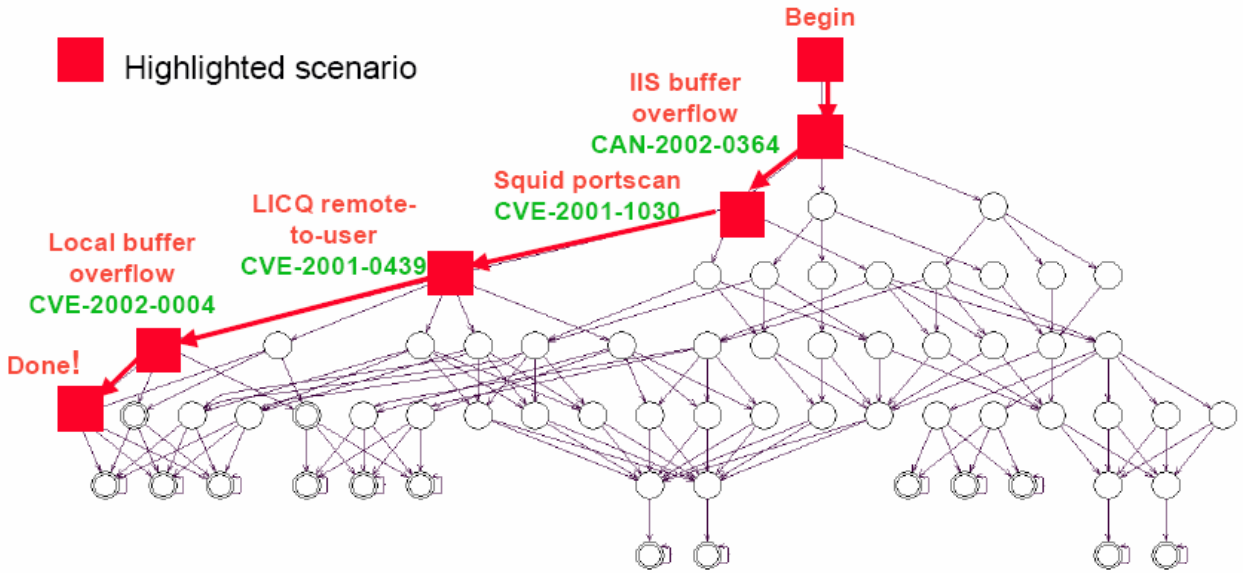
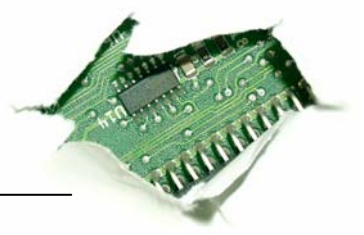


Figure 10.2: Example Attack Graph

Attack Graphs and Web Services

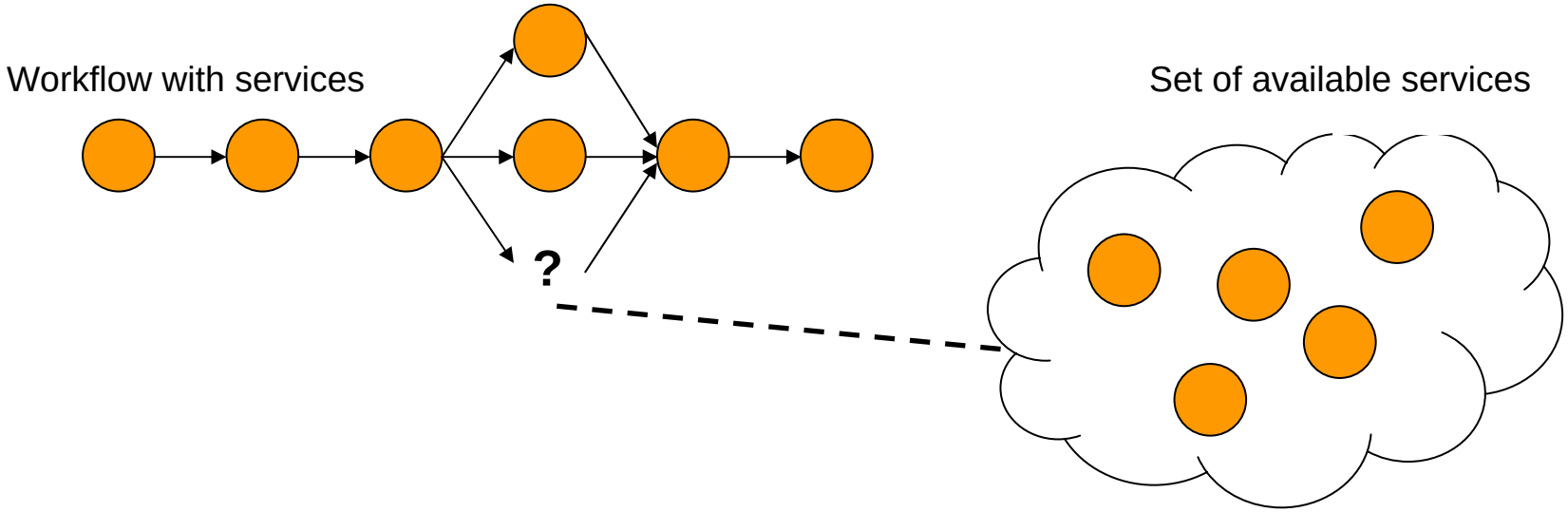


Web Services-based business processes with
- WSDL service descriptions,
- BPMN/BPEL workflow descriptions,
- CVE/CVSS vulnerability descriptions.

Input:
- attack(s)
- hosts and services — WSDL
- connectivity
- vulnerabilities — CVE/CVSS

Output:
- attack model
- attack graph

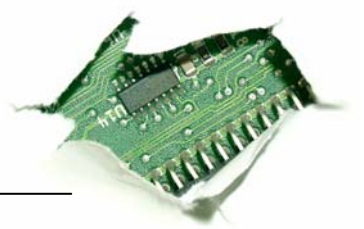
Challenge: Analyzing Service Compositions

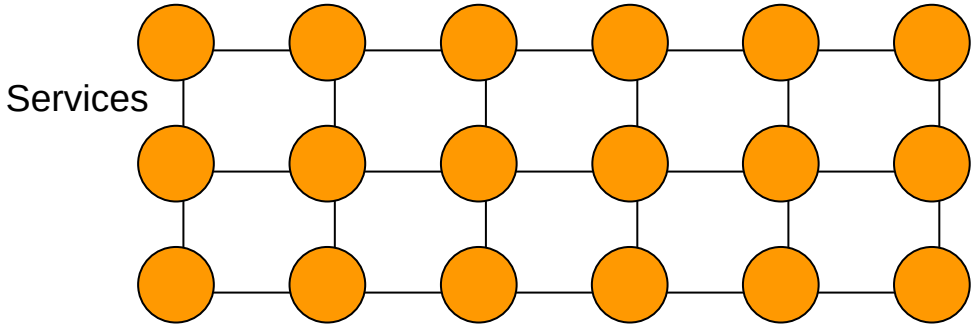


- “x protocols, y ports” boils down to HTTP on port 80
- huge graphs need to be created after each change
- service selection requires quick decision

order travel expense claim:
can an attacker steal the bonus/harm integrity?

Utilizing BPEL for Composition Analysis

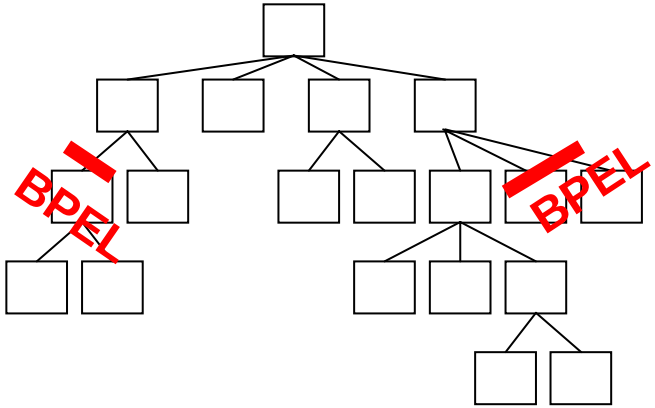




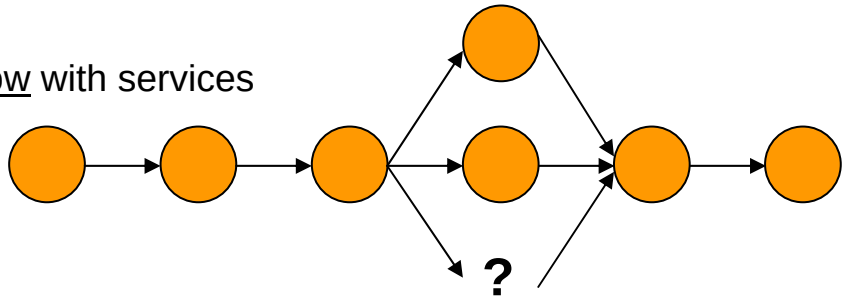
- From a workflow point of view, BPMN/BPEL describes service connectivity

- Pre-calculated graph can be pruned according to that connectivity

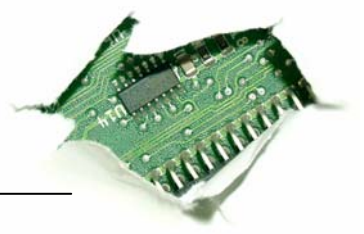
Business process attack graph (BAG)



Workflow with services

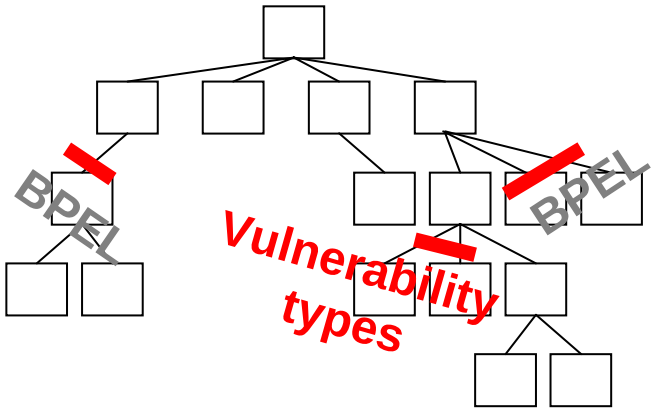


Vulnerability-based Graph Pruning



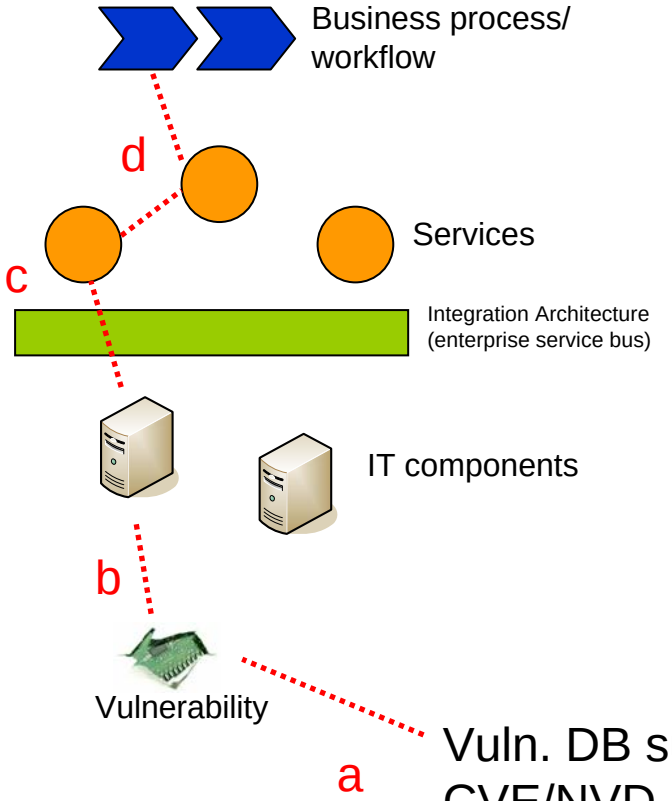
- Fewer vulnerability types mean a smaller attack graph
- Considering only **known and present** vulnerabilities further reduces the graph

Business process attack graph (BAG)



order travel expense claim:
bonus is secure, service does not introduce new attack possibilities

Prototype Design 1/4: Identifying Service Vulnerabilities



Bottom-up approach to keep the analysis up-to-date regarding published vulnerabilities:

a) Obtain vulnerability description from vuln. DB, e.g., via RSS feed

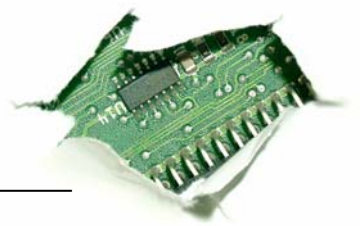
b) Identify affected IT components through, e.g., network scan

c) Identify affected services via enterprise service bus (ESB) info

d) Identify affected workflows from process models (e.g., BPMN/BPEL)

Vuln. DB such as
CVE/NVD, ISS X-Force, OSVDB

Prototype Design 2/4: Building and Pruning BAGs



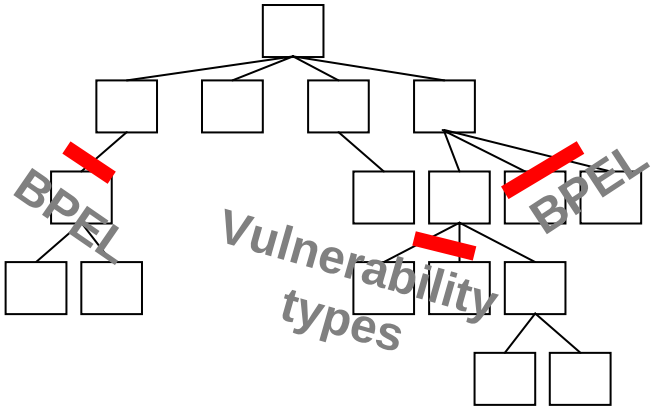
Input:

- attack(s) ?
- hosts and services WSDL
- connectivity BPMN/BPEL
- vulnerabilities CVE/CVSS

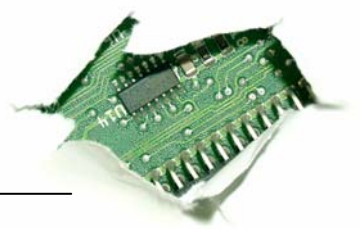
Output:

- business process attack model
- business process attack graph

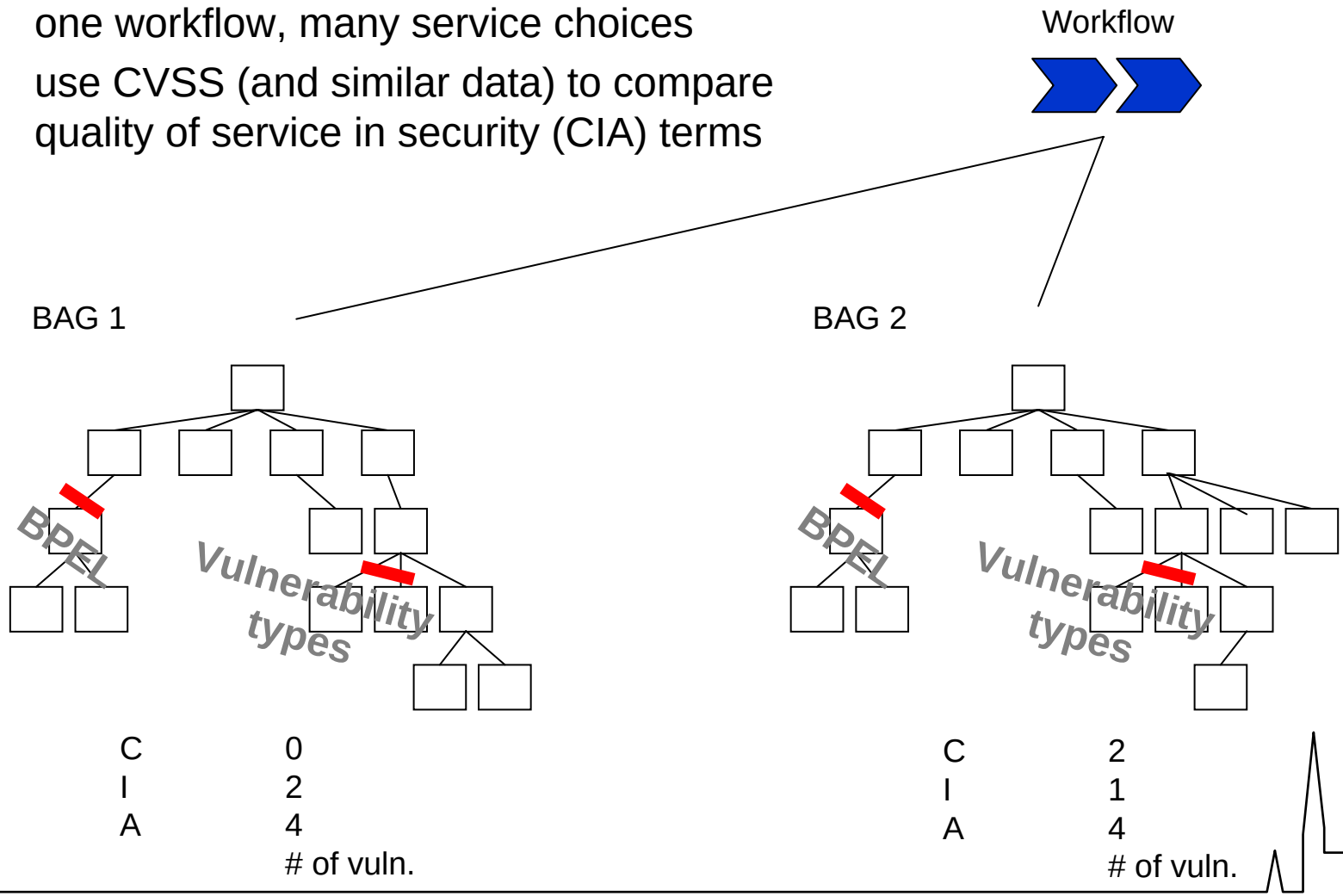
Business process attack graph (BAG)



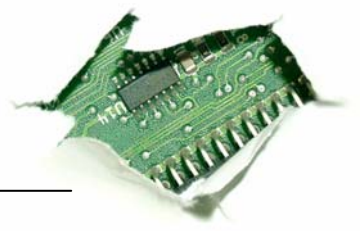
Prototype Design 3/4: Comparing Service Compositions



- one workflow, many service choices
- use CVSS (and similar data) to compare quality of service in security (CIA) terms



Prototype Design 4/4: Putting it all together



workflow:

1) Identify service vulnerabilities

which services are vulnerable?

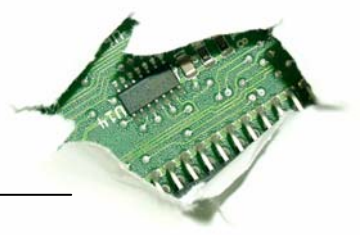
2) Build BAGs

which attacks are possible?

3) Compare service compositions

which composition better fits the security requirements?

Risk Management and BAG: current status



- **Identification**

  Which services are vulnerable? How can the workflow be attacked?
  Component-to-service identification, BAG

- **Quantification**

  How likely is an attack? How much damage would it cause?
  Expert estimates (CVSS), controlling, weighted BAG

- Control

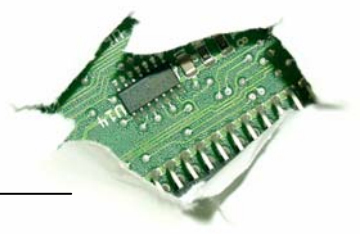
  Are there countermeasures? Where should they be placed?

  Expert analysis, attack graph analysis

- Monitoring

  Are the selected countermeasures effective? And efficient?
  BAG w/ dynamic analysis, expert analysis

Compliance and BAG: some thoughts



- show that services have been “carefully selected“
- create logs for audits and forensics
- help automate audits, guide human auditors

Vulnerability Analysis for Web Services-based Business Processes

Questions?

Comments?

Lutz Lowis
Department of Telematics
Institute of Computer Science and Social Studies (IIG)
University of Freiburg, Germany
lowis@iig.uni-freiburg.de
http://www.telematik.uni-freiburg.de