Joining Efforts Towards Dependability & Security Assessment

Jean Arlat
(jean.arlat@laas.fr)
Assessment

Modeling & Simulation, Controlled Experimentation, Field Measurement

- **Accidental faults:** HW, SW, Operator
  - Fault/Error models (stuck-at, bit-flip, ODC, etc.)
  - Probabilistic modeling (simulation, CTMC, SPN, SW reliability growth, etc.)
    - Integration into the main design thread (UML, AADL)
  - Experimentation (field measurement, fault injection, dependability benchmarking, etc.)

- **Malicious faults:** Insiders, Outsiders
  - Threats and vulnerabilities
  - Evaluation criteria (TCSEC, ITSEC, CC) ≈ qualitative assessment
    - quantitative assessment of operation security?
  - Experimentation (testing scripts, fault injection, honeypots, etc.)


1999->… — IFIP WG 10.4 SIGDEB (CMU, Critical SW, HP, IBM, Intel, LAAS-CNRS, Sun, U. Coimbra, UIUC, U. Valencia, etc.) [www.laas.fr/~kanoun/ifip_wg_10_4_sigdeb]
IBM Orthogonal Defect Classification

A SW fault is characterized by the change in the code that is necessary to correct it

- **Fault trigger**: Conditions that make the fault to become an error
- **Fault type**: Type of mistake in the code
  - **Assignment**: values assigned incorrectly or not assigned
  - **Checking**: missing or incorrect validation of data, or incorrect loop, or incorrect conditional statement
  - **Timing/serialization**: missing or incorrect serialization of shared resources
  - **Algorithm**: incorrect or missing implementation that can be fixed without the need of design change
  - **Function**: incorrect or missing implementation that requires a design change to be corrected

**Typical Data Table**

<table>
<thead>
<tr>
<th>ID</th>
<th>open date</th>
<th>closed date</th>
<th>activity</th>
<th>trigger</th>
<th>impact</th>
<th>type</th>
<th>qualifier</th>
<th>source</th>
<th>age</th>
<th>severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>3/1/97</td>
<td>3/8/97</td>
<td>des/rev</td>
<td>conformance</td>
<td>capability</td>
<td>assign</td>
<td>miss</td>
<td>in-house</td>
<td>new</td>
<td>2</td>
</tr>
<tr>
<td>12377</td>
<td>6/1/97</td>
<td>6/15/97</td>
<td>unit test</td>
<td>simple</td>
<td>usability</td>
<td>checking</td>
<td>miss</td>
<td>in-house</td>
<td>new</td>
<td>2</td>
</tr>
<tr>
<td>12470</td>
<td>6/5/97</td>
<td>7/15/97</td>
<td>function test</td>
<td>coverage</td>
<td>integrity security</td>
<td>algorithm</td>
<td>incorrect</td>
<td>outsourced</td>
<td>refixed</td>
<td>1</td>
</tr>
<tr>
<td>12543</td>
<td>8/4/97</td>
<td>8/30/97</td>
<td>system test</td>
<td>soft config</td>
<td>reliability</td>
<td>function</td>
<td>miss</td>
<td>ported</td>
<td>rewritten</td>
<td>1</td>
</tr>
</tbody>
</table>
From Software Faults to Faultload (2/2)

- **U. Coimbra**: Study Failure Reports from 9 OSS Programs: (text editors, Linux kernel, game, etc.)

<table>
<thead>
<tr>
<th>ODC Type</th>
<th># of faults</th>
<th>ODC distribution (U. Coimbra)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>118</td>
<td>22.1 %</td>
</tr>
<tr>
<td>Checking</td>
<td>137</td>
<td>25.7 %</td>
</tr>
<tr>
<td>Interface</td>
<td>43</td>
<td>8.0 %</td>
</tr>
<tr>
<td>Algorithm</td>
<td>198</td>
<td>37.2 %</td>
</tr>
<tr>
<td>Function</td>
<td>36</td>
<td>6.7 %</td>
</tr>
</tbody>
</table>

- **Alternative Fault Classification**

  Faults considered as language constructs that are:
  - **Missing** (e.g., missing part of a logical expression)
  - **Wrong** (e.g., wrong value used in assignment)
  - **Extraneous** (e.g., extra condition in a test)

  → **Propose a Mutation Strategy for machine-code level**

---

Quantitative Assessment of Security

Vulnerabilities Modeling
“privilege graph”

Node = set of privileges
Arc = vulnerability class
Path = sequence of vulnerabilities that could be exploited by an attacker to defeat a security objective
Arc weight = effort to exploit the vulnerability

-> Questions?

- Is such a model valid in the real world?
- Considered behaviors are two extreme ones, but, what would be a “real” attacker behavior?
- Weight parameters are assessed arbitrarily (subjective?)

-> Wanted! Real Data

CADHo project: “Collection and analysis of Attack Data based on Honeypots (Eurecom, LAAS-CNRS, Renater)

- Both low- (35 worldwide) and high-interaction honeypots
  
  Typical behavior:

1- Dictionary attack
Automated scripts
IPs dedicated to dictionary attacks
IPs dedicated to intrusion attacks

2- Share information?
knowledge base of attacks?

3- Get information?

4- Intrusion attack
Humans

High-interaction honeypot

Internet
Firewall
ssh + weak passwords
Debian
Debian

Application (LAAS Network)

# paths

Date

06/04 08/04 09/04 11/04 12/04 02/05 04/06 05/05 07/05

8

0

1

10

100

1000

0,1

1

10

100

1000
Some Concluding Remarks

- **Academia/Industry Cooperation: Some successful stories...**
    (LAAS-CNRS, Airbus, Astrium, EdF, Technicatome, Thales)
  - ReSIST Resilience for Survivability in IST (NoE) [www.resist-noe.org] 2006-2008

- **With few exceptions, industry reluctant to disclose fault/threat data (including contextual information)**
  - OSS community
  - Deployment of honeypots

- **Security assessment compatible with quantitative approaches?**

- **Some Additional Challenges Ahead:**
  - **System Features:**
    - Ubiquity, Evolvability, Openess, Scalability, Diversity
    - Network of Mobile Entities (Hidenets) [www.hidenets.aau.dk]
    - Interdependencies in Critical Infrastructures (Crutial) [crutial.cesiricerca.it]
  - **Assessment Techniques:**
    - Analytical Evaluation and Experimentation
    - Formal methods (Proving, Model Checking) and Testing
    - Accidental and Malicious Faults