Benchmarking the Security of Software Systems

**The past:** Performance & Dependability Benchmarking

**The present:** Security Benchmarking

Benchmarking the Security of Systems
- Approach: Qualification + Trustworthiness Assessment
- Example: Benchmarking Web Service Frameworks

Benchmarking Security Tools
- Approach: Vulnerability and Attack Injection
- Example: Benchmarking Intrusion Detection Systems

Challenges and Conclusions

Performance Benchmarking

Assessing and comparing computer systems and/or components in terms of performance

- Workload: Set of representative operations
- Metrics: Throughput, Response time, Latency, ...
TPC-C (1992)

- **Workload:**
  - Database transactions
  - Although some integrity tests are performed, it assumes that nothing fails
  - Transaction rate (tpmC)
  - Price per transaction ($/tpmC)

**Metrics:**
- Performance and/or dependability
- Both baseline and in the presence of faults
- Unconditional and/or direct

Dependability Benchmarking

- Assessing and comparing computer systems and/or components considering dependability attributes

DBench-OLTP (2005)

- **Workload:**
  - TPC-C transactions

- **Faultload:**
  - Operator faults + Software faults + HW component failures

- **Metrics:**
  - Performance: tpmC, $/tpmC, Tf, $/Tf
  - Dependability: Ne, AvtS, AvtC

DBench-OLTP (2005)

- **Faultload:** Operator faults

- Does not take into account malicious behaviors (faults = vulnerability + attack)
Security Benchmarking

Assessing and comparing computer systems and/or components considering security aspects

- Benchmarking the Security of Systems / Components
  - Systems that should implement security requirements
  - OS, middleware, server software, etc.

- Benchmarking Security Tools
  - Tools used to improve the security of systems
  - Penetration testers, static analyzers, IDS, etc.

A Different Approach...

Security Qualification

- Security Qualification:
  - Apply state-of-the-art techniques and tools to detect vulnerabilities
  - SUBs with vulnerabilities are:
    - Disqualified
    - Or vulnerabilities are fixed

A Different Approach...

Trustworthiness Assessment

- Security Qualification
  - Security = 0

- Trustworthiness Assessment:
  - Gather evidences on how much one can trust
  - e.g., best coding practices, development process, bad smells

A Different Approach...

Security Qualification

- Security Qualification
  - Security = 0

- Trustworthiness Assessment
  - Score

Example: Web Service Frameworks

- Qualification
  - DoS Attacks
  - Coercive Parsing, Malformed XML, Malicious Attachment, etc.

- Trustworthiness Assessment:
  - Quality model to compute a score

- Metrics:
  - Portray trust from a user perspective
  - Dynamic: may change over time
  - Depend on the type of evidences gathered
  - Different metrics for different attack vectors

- SUBs
Faultload: – Vulnerabilities are injected
  – Attacks target the injected vulnerabilities

Data can be collected for benchmarking security tools
  – Penetration testers, static analyzers, IDS, etc.

Faultload:

- Vulnerabilities are injected
- Attacks target the injected vulnerabilities
- Data can be collected for benchmarking security tools
  - Penetration testers, static analyzers, IDS, etc.
EVALUATION APPROACH

EXAMPLES OF VULNERABILITIES INJECTED

Original PHP code | Code with injected vulnerability | Operation performed
--- | --- | ---
$id=intval($_GET['id']); | $id=$_GET['id']; | Removed the `intval` function allowing also non-numeric values (i.e. SQL commands) in the "$id" variable
$pagno = urlencode($pagno); | $pagno = $pagno; | Removed the `urlencode` function allowing also alphanumeric values (i.e. SQL commands) in the "$pagno" variable

EXAMPLES OF ATTACKS

<table>
<thead>
<tr>
<th>Attack payload</th>
<th>Modified the structure of the query, usually results in an error</th>
<th>Expected result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$id</td>
<td>=1</td>
<td>Modifies the structure of the query. Usually results in an error</td>
</tr>
<tr>
<td>$id</td>
<td>=2</td>
<td>Modifies the structure of the query. Usually results in an error</td>
</tr>
<tr>
<td><code>$id</code></td>
<td>Modifies the query result to 0</td>
<td>0</td>
</tr>
<tr>
<td><code>$id</code></td>
<td>Modifies the query result to 0</td>
<td>0</td>
</tr>
<tr>
<td><code>$id</code></td>
<td>Modifies the query result to 0</td>
<td>0</td>
</tr>
</tbody>
</table>

SYSTEMS UNDER BENCHMARKING

<table>
<thead>
<tr>
<th>Tool</th>
<th>Architectural Level monitored</th>
<th>Detection Approach</th>
<th>Data Source</th>
<th>Source Technology Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACD</td>
<td>Application</td>
<td>Anomaly Based</td>
<td>Apache Log</td>
<td>Performance</td>
</tr>
<tr>
<td>Scalp</td>
<td>Application</td>
<td>Signature Based</td>
<td>Apache Log</td>
<td>Performance</td>
</tr>
<tr>
<td>ModSecurity</td>
<td>Network</td>
<td>Signature Based</td>
<td>Apache Log</td>
<td>Performance</td>
</tr>
<tr>
<td>GreenSQL</td>
<td>Database</td>
<td>Signature Based</td>
<td>MySQL data</td>
<td>Technology</td>
</tr>
<tr>
<td>DB IDS</td>
<td>Database</td>
<td>Anomaly Based</td>
<td>MySQL data</td>
<td>Technology</td>
</tr>
</tbody>
</table>

EXPERIMENTAL SETUP

MAIN RESULTS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ACD</td>
<td>0.52</td>
<td>171</td>
<td>171</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Scap</td>
<td>0.49</td>
<td>241</td>
<td>241</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>ModSecurity</td>
<td>0.41</td>
<td>157</td>
<td>157</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Net Snort 2.8</td>
<td>0.34</td>
<td>135</td>
<td>135</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>GreenSQL</td>
<td>0.47</td>
<td>244</td>
<td>244</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>DB IDS</td>
<td>0.45</td>
<td>365</td>
<td>365</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Snort 2.9</td>
<td>0.41</td>
<td>175</td>
<td>175</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
WHAT IS WRONG?

Established benchmarks are mostly for marketing!
- Strict benchmarking conditions
  - Fixed workload & faultload + Small set of metrics
- Workload & faultload:
  - May not be representative of the user scenario
- Metrics:
  - Fixed! May not satisfy the user needs
  - Decision based on several metrics is difficult!
  
No security benchmark endorsed by any organization or industry

FIXED!

Example:
- Benchmarking vulnerability detection tools
- Typical metric: F-Measure
- Is this good in all scenarios?
  - Business critical: recall
  - Best effort: F-Measure
  - Minimum effort: Markedness

A POTENTIAL APPROACH...

Benchmarking conditions adaptable to the user needs
- Include multiple usage scenarios:
  - Metrics depend on the scenario
  - Adaptable workload and faultload
- Use quality models instead of independent metrics
  - Quality models should also adapt to the scenario

SCENARIOS AND QUALITY MODELS

How to define scenarios? How to define quality models? How to adapt workloads and faultloads to the scenarios?

CHALLENGES

Satisfy industry requirements
- Representativeness, portability, scalability, non-intrusiveness, low cost, ...
- Prevent “gaming”

Satisfy user requirements
- Representativeness, usefulness, simplicity of use...
- Adaptable – allow “gaming”

Endorsement by TPC, SPEC, ...
  - How to?

IS THERE A FUTURE?

Resilience Benchmarking
- Assess and compare the behavior of components and computer systems when subjected to changes
  - Which resilience metrics?
    - Comparable, consistent, understandable, meaningful, ...
  - Changeloads:
    - Representative, practical, portable, ...

Trustworthiness Benchmarking
- What evidences to collect?
- What metrics?
  - Dynamicity of perception... social trust...
CONCLUSIONS

- The benchmarking concept is well established!
- Acceptance by “big” industry depends on perceived utility for marketing
- Acceptance by users requires “adaptability”
- From a research perspective, performance and dependability benchmarking are well known
- Security benchmarking approaches are weak
- New types of benchmarks will bring additional challenges!

QUESTIONS?

Marco Vieira
Department of Informatics Engineering
University of Coimbra
mvieira@dei.uc.pt
http://eden.dei.uc.pt/~mvieira