Measuring Security Investment Benefit and Threat Modeling for Off The Shelf Software Systems

- A Stakeholder Value Driven Approach

Yue Chen

CII Forum Workshop Tutorial 2007

Oct. 31, 2007
Agenda

Trends, challenges and Motivations

• Related Work
• T-MAP Framework
• The Tiramisu Tool
• Results and Validation
• Limitations
• Contributions and Future Work
Trends: Increasing Concerns on COTS Security

- Increasing COTS usage
  
  *Data source: [Boehm et al 2003][Standish]*

- Increasing number of COTS vulnerabilities published
  
  *Data source: [CERT Statistics]*

---

**Increasing Trend of COTS Based Applications**

<table>
<thead>
<tr>
<th>Year</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>20</td>
</tr>
<tr>
<td>1998</td>
<td>30</td>
</tr>
<tr>
<td>1999</td>
<td>40</td>
</tr>
<tr>
<td><strong>2000</strong></td>
<td><strong>50</strong></td>
</tr>
<tr>
<td>2001</td>
<td>60</td>
</tr>
<tr>
<td>2002</td>
<td>70</td>
</tr>
</tbody>
</table>

**COTS Product Vulnerability Published by CERT Annually**

<table>
<thead>
<tr>
<th>Year</th>
<th>Number of Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td>1000</td>
</tr>
<tr>
<td>1998</td>
<td>2000</td>
</tr>
<tr>
<td>1999</td>
<td>3000</td>
</tr>
<tr>
<td>2000</td>
<td>4000</td>
</tr>
<tr>
<td>2001</td>
<td>5000</td>
</tr>
<tr>
<td>2002</td>
<td>6000</td>
</tr>
<tr>
<td>2003</td>
<td>7000</td>
</tr>
<tr>
<td>2004</td>
<td>8000</td>
</tr>
<tr>
<td>2005</td>
<td>9000</td>
</tr>
</tbody>
</table>
Challenges

• About 47% of organizations spend only 2% of their IT budget on security
  [Source: CSI/FBI 2006 Survey by Gordon & Leob]

• Challenges: (1) How to allocate limited amount of security resources to maximize stakeholder utilities/values? (2) What are the top 20% vuln that causes 80% risks?

Current Practice

• Off-The-Shelf (OTS) software vulnerability rankings are value neutral and static

• Three ~ five levels of rankings are used to differentiate 27,000 published OTS vulnerabilities
  (e.g. Symantec, Microsoft & NIST)

• Decisions made based on
  – Best knowledge
  – Individual experience
  – Ad hoc
  [Butler, 2002]
Proposal: Threat Modeling framework based on Attack Path analysis (T-MAP)

A threat modeling framework for COTS based systems that is sensitive to system stakeholder value context, dynamic, and tool-automated

Current Approaches:

- COTS Vulnerability
- Value Neutral Assessment
- COTS Vulnerability Rankings

T- MAP:

- COTS Vulnerability
- Value Neutral Assessment
- Scenario Evaluation
- COTS Vulnerability Rankings

Evaluation criteria based on stakeholder value propositions

©USC-CSSE
Hypothesis

“No value driven COTS vulnerability prioritization system can be devised so that its performance will significantly differ from the existing mainstream value neutral approaches.”

- It is a null hypothesis
Agenda

• Trends, challenges and Motivations
• Related Work
  • T-MAP Framework
  • The Tiramisu Tool
  • Results and Evaluation
  • Limitations
  • Contributions and Future Work
Related work

• **Value Based Software Engineering & Security Economics**
  – Security should be at a level that makes stakeholders winners
  – Figure of Merit [Boehm,1981][Dowkont,1967]; AHP[Saaty,1980]; large body of Security Economics researches

• **COTS vulnerability studies**
  – CERT, NIST CVSS, [Arora 2004], [Martin 2002], Symantec, Microsoft, ISS, BugTraq, NVD

• **Utility Analysis**: How security may impact what stakeholder values?
  – Analytical Hierarchy Process (AHP) [Saaty,1980][Bodin, 2005]

• **Dependency Analysis**: How stakeholder values depend on security?
  – Result Chain [Thorpe,1998];
  – Data flow analysis [Howard and LeBlanc, 2002][Torr, 2005]

• **Tools**
  – Kuang, COPS, Nessus, SATAN, ISS, SkyBox
Agenda

• Trends, challenges and Motivations
• Related Work
• **T-MAP Framework**
  • The Tiramisu Tool
  • Results and Evaluation
  • Limitations
• Contributions and Future Work
Nature of The Problem

Permitted Ports

Firewall Wrapper

Software Applications, COTS
- e.g. Windows Server 2003
- e.g. SQL Server 2000
- e.g. IIS 6.0

IT Servers
- e.g. Web Server
- e.g. CRM Server

Org. Values
- Productivity
- Reputation

Nature of The Problem

Org. Values
- Productivity
- Reputation

Vulnerabilities impacting confidentiality, availability, integrity

Attacking Paths
- Unblocked vulnerabilities
- Blocked vulnerabilities
Threat Modeling with Attack Path Analysis

**Analog:** Measure the security of a castle by the value of treasures (values) in the castle, the number of holes on the walls (attack paths), as well as the size of the holes (attack path severity).

**Structured-Attack-Graph** (an attack result chain graph)
T-MAP Framework

- **Step 1:** Identify key stakeholders and value propositions (the treasures in the castle);
- **Step 2:** Establish a set of security evaluation criteria based on stakeholder value propositions;
- **Step 3:** Use tool to enumerate and analyze attack paths based on a comprehensive COTS vulnerability database containing 27,400 vulnerability information (the holes);
- **Step 4:** Evaluate the severity of each scenario in terms of numeric ratings against the evaluation criteria established in Step 2 (the size of the holes);
- **Step 5:** System total threat is quantified with the total severity ratings of all attack paths;
- **Step 6:** The security threat of each vulnerability is quantified with the total severity ratings of all attack paths that go through this vulnerability;

[Note] Step 3 to 6 are tool automated by the *Tiramisu* Tool
Injecting Stakeholder Utilities through AHP – An Example

Example (from USC ISD Server X Case Study)

Weights derived through AHP pair-wise comparisons

Possible Breach Scenarios

Value Centric Criteria

<table>
<thead>
<tr>
<th>Table</th>
<th>Security Breach Scenario Evaluation Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>Criteria</td>
</tr>
<tr>
<td>0.095</td>
<td>1. Productivity</td>
</tr>
<tr>
<td>0.738</td>
<td>2. Regulation</td>
</tr>
<tr>
<td>0.167</td>
<td>3. Privacy</td>
</tr>
<tr>
<td>0.648</td>
<td>a. Student</td>
</tr>
<tr>
<td>0.230</td>
<td>b. Faculty</td>
</tr>
<tr>
<td>0.122</td>
<td>c. Staff</td>
</tr>
</tbody>
</table>

©USC-CSSE
T-MAP Threat Rating System

• **Severity Weight of Attack Path** $P$:

\[
\text{Weight}(P) = \prod_i \text{Rating}(P.\text{Attribute}_i)
\]

• **Overall Security Threat Score** of COTS System $G$:

\[
\text{TotalThreat}(G) = \sum_i \text{Weight}(\text{AttackPath}_i)
\]

• **ThreatKey of elements in Attack Graph**:

For a given node $N$ in a Structured Attack Graph $G$, define:

\[
\text{ThreatKey}(N) = \sum_i \text{Weight}(\text{AttackPath}_i)
\]

where $i$ varies from 1 to the total number of attacking paths that go through node $N$, and $\text{AttackPath}_i$ enumerates all the Attack Paths that go through $N$.

• **Effectiveness of Security Practice**:

For a given security practice $SP$, 

\[
\text{Effectiveness}(SP) = 1 - \frac{\text{TotalThreat}(\text{AfterSP})}{\text{TotalThreat}(\text{BeforeSP})}
\]
Example: The Effect of Patching *Vuln 2*
Example: The Effect of Patching Vuln 2
Security Investment Effectiveness Estimation

- How much security threats can be avoided by implementing Firewall, Software hardening (patching), user account control, or file system encryption?
- Results as well depends on the total value of the protected system

<table>
<thead>
<tr>
<th>Attack Path Attributes</th>
<th>Properties</th>
<th>Security Investment Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Firewall</td>
</tr>
<tr>
<td>Fix Availability Level</td>
<td>Official Fix</td>
<td>80%</td>
</tr>
<tr>
<td>Temporal Fix</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>Work around</td>
<td>10%</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attacker needs user account to attack? Required</td>
<td></td>
<td>100%</td>
</tr>
<tr>
<td>Type of compromise</td>
<td>Confidentiality</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Integrity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Availability</td>
<td></td>
</tr>
<tr>
<td>AG1</td>
<td>-</td>
<td>90%</td>
</tr>
<tr>
<td>AG2</td>
<td>-</td>
<td>90%</td>
</tr>
<tr>
<td>AG3</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

AG1: Attacker Group who knows escalate his user privilege on victim host
AG2: Attacker Group who do not know how to escalate his user privilege
AG3: Insiders

* Case study results estimated by professional security manager at USC-ITS

© USC-CSSE
A Quick Algorithm to Calculate ThreatKey

Complexity: $O(|V|+|E|) + O(n \log(n))$, $n$ is the number of vuln nodes

$$\text{ThreatKey}(V_{21}) = \sum \text{Weight(AttackPath}_i)$$
6 Attack Paths total

Define:

TopDownWeight = $W(V_{21}) \times [ W(V_{31})+W(V_{32}) ]$

BottomUpWeight = $W(V_{21}) \times [ W(V_{11})+W(V_{12})+W(V_{13}) ]$

$$\text{ThreatKey}(V_{21}) = \frac{\text{TopDownWeight} \times \text{BottomUpWeight}}{W(V_{21})}$$

CalculateThreatKey(AttackGraph G) {
    Inductively calculate BottomUpWeight for each node in G;
    Inductively calculate TopDownWeight for each node in G;
    Calculate ThreatKey for each node in G;
    QuickSort Vulnerability Nodes by ThreatKey;
}

©USC-CSSE 18
Agenda

- Trends, challenges and Motivations
- Related Work
- T-MAP Framework
- The Tiramisu Tool
- Results and Evaluation
- Limitations
- Contributions and Future Work
T-MAP tool: Tiramisu

System Capabilities

• Automated vulnerability collection from:
  – NIST NVD, Symantec BugTraq, Microsoft, ISS, FrSIRT, SANS
• Allow security professionals define their IT systems from a comprehensive COTS list
• Prioritize security vulnerabilities based on stakeholder value propositions
• Estimate the effectiveness of common security practices such as: firewall, patching, data encryption
• Export report into Excel
Tiramisu Tool Architecture

Data Collection Engine

- Contains more than 27,000 published vulnerability impacting more than 31,700 COTS software

Software Architecture

- Report Generator
- User Definable Security Practice Simulator
- Attacking Path Calculation and Management
- Data Storage and Management: Vulnerability XML DB, IT Asset XML DB, Business Value XML Definitions
- Automated Data Collecting Engine
Tiramisu Tool Screenshots(1)

Finding Attack Paths
Tiramisu Tool Screenshots(2)

Estimating Effectiveness of Popular Security Practices
COTS Security Economics – Finding Sweet Spots

- Economic curve of security patching
  (from USC Server X case study)
- “20% percent of vulnerabilities causes 80% of the security risks”, T-MAP tells what are the 20%

- Sweet spot to invest in security
- Also driven by the total value of system
  (from USC Server X case study)

Sweet spots to invest
Agenda

- Trends, challenges and Motivations
- Related Work
- T-MAP Framework
- The Tiramisu Tool
- Results and Evaluation
- Limitations
- Contributions and Future Work
Validation (1) – USC ITS Case Study
Validation (2) – MASH Case Study
Validation (3) – GreenBay Case Study

- CSSE Greenbag Server Case Study
- CSSE Greenbag Server Case Study
- CSSE Greenbag Server Case Study
- CSSE Greenbag Server Case Study

©USC-CSSE
Validation(4) Metrics: Inaccuracy

- **Clash**: prioritization mismatch between the vulnerability ranking system and the security manager
- **Metrics**: clash counting of over-estimates and under-estimates

\[ \text{Inaccuracy} = \frac{\# \text{ clashes}}{\# \text{ of comparisons}} \]

\[ \# \text{ of comparisons} = \binom{n}{2} = \frac{n(n-1)}{2} \]

\[ C_n^2 = \frac{n(n-1)}{2} \]
### Validation(5): Inaccuracy Comparisons between T-MAP and Value Neutral Approaches

<table>
<thead>
<tr>
<th></th>
<th>ServerX Case Study (Total Comparisons 28)</th>
<th>MASH Case Study (Total Comparisons 36)</th>
<th>GreenBay Case Study (Total Comparisons 66)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Clashes</td>
<td>Inaccuracy</td>
<td># of Clashes</td>
</tr>
<tr>
<td>T-MAP</td>
<td>2</td>
<td>7.1%</td>
<td>4</td>
</tr>
<tr>
<td>CVSS v2.0</td>
<td>5</td>
<td>17.9%</td>
<td>9</td>
</tr>
<tr>
<td>CVSS v1.0</td>
<td>6</td>
<td>21.4%</td>
<td>12</td>
</tr>
<tr>
<td>IBM ISS</td>
<td>9</td>
<td>32.1%</td>
<td>17</td>
</tr>
<tr>
<td>Microsoft</td>
<td>N/A</td>
<td>N/A</td>
<td>21</td>
</tr>
</tbody>
</table>
Validation(6): Inaccuracy Visualization by Box plotting
Validation(7) Top Reasons for Over and Under Ranking Estimates

- Disabled services/programs
- Privileges of running services/programs
- Security professional’s self-confidence on handling certain vulnerabilities
- Confidence on insiders and environment
## Validation (8): Threat to Validation

<table>
<thead>
<tr>
<th>Threat to Validation</th>
<th>Mitigation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using security expert’s vulnerability ranking as “truth” to determine clashes</td>
<td>Conduct case studies with experienced Security Managers: (Case studies conducted with 2 CISSP Holders, and a security manager with 18 yrs experience)</td>
</tr>
<tr>
<td>The number of case studies can be conducted is limited, also the number of vulnerability that a security manager can prioritize manually is limited</td>
<td>Explore more case studies with CSSE affiliates and other possible sources in the CSSE Annual Research Review (Promised Guidance Committee 3 Case Studies in my Qual Exam)</td>
</tr>
<tr>
<td>Need for comprehensive vulnerability database to generate meaningful output</td>
<td>Developed automated vulnerability information crawling/collecting engine</td>
</tr>
</tbody>
</table>
Agenda

• Trends, challenges and Motivations
• Related Work
• T-MAP Framework
• The Tiramisu Tool
• Results and Evaluation
• Limitations
• Contributions and Future Work
Limitations

• Only sensitive to known COTS Vulnerabilities
  – Empirical study by Arora shows that the average attacks per host per day jumped from 0.31 to 5.45 after vulnerability get published

• Not sensitive to nuance in local system configurations
  – Disabled services
  – Services running on different privileges, etc.

• Only cover “one-step-attacks” that exploiting COTS vulnerabilities

• Depends on comprehensive vulnerability database
  – Our database: 27,400 vulnerability published from 1999-2007 that resides in more than 31,300 Off-The-Shelf-Software

• OTS vulnerability is only a small island in the overall scope of security research world
Agenda

- Trends, challenges and Motivations
- Related Work
- T-MAP Framework
- The Tiramisu Tool
- Results and Evaluation
- Limitations
- Contributions and Future Work
Contributions (1)
(specific for the case studies conducted)

• Inject stakeholder utilities into security economic evaluation, bringing systems back to value context

• Prioritize COTS vulnerabilities better by injecting stakeholder value/utilities into ranking system

• Establish traceability and consistency between management-level value propositions and tech-level security threat mitigation strategies

• Enables reasoning the effectiveness of common security practices such as patching, data encryption, firewall, etc., with respect to stakeholder value/utilities

• Help communication between mgmt. and technical
Contributions (2)
(specific for the case studies conducted)

• Developed an $O(n)$ algorithm to calculate the associated threat weight (ThreatKey) of vulnerabilities

• The Tiramisu Tool and a comprehensive COTS vulnerability database

• Provided empirical evidence that value driven approach outperformed value neutral approaches in COTS vulnerability evaluation

• Designed a metric to evaluate the accuracy of COTS vulnerability ranking systems
Feedbacks

Applied for a patent

Client feedbacks

• “A valuable way of quantifying the very difficult tradeoffs that we have to make everyday.”
• “T-MAP results well fit my past experience.”

Citations

• Enlisted in the Software Security Assurance, State of The Art Report (Page 149) prepared by the Information Assurance Technology Analysis Center (IATAC) for DoD
• Commented as one of the most promising emerging methods for assessing software security risks by the Department of Homeland Security publication, Security in Software Lifecycle (Page 46) August 2006
Selected Publications

- Yue Chen, "Stakeholder Value Driven Threat Modeling for Off The Shelf Based Systems", The 29th International Conference on Software Engineering (ICSE) , Doctoral Symposium, Minneapolis, MN, May 2007 (11 out of 48 selected for long presentation, ratio 23%)
Acknowledgement

This research was sponsored by the NSF Grant CCF-0438931 and the Affiliates of the USC Center for Systems and Software Engineering. We are grateful to:

Dr. Barry W. Boehm (USC)
Dr. Ming-deh Huang (USC)
Dr. Neno Medvidovic (USC)
Dr. Clifford Neuman (USC)
Dr. Bert Steece (USC)
Dr. Mary Shaw (CMU)
Dr. Rick Kazman (CMU, SEI)
Mr. Robert C. Seacord (CMU, CERT)
Mr. Tyler Moore (Cambridge University)
Dr. Dan Port (University of Hawaii)
Ms. Elizabeth Fong (NIST)

for their very helpful discussions and insights on this research.
Thank you!