CySER The Complexities of Hierarchical Software Quality Assurance Models

March 4, 2024
Dr. Clemente Izurieta
Professor of Computer Science
Software Engineering and Cybersecurity Laboratory (SECL)
Montana State University
Participants:

**Institutional PI:** Dr. Clemente Izurieta
**ROTC Air Force:** Lieutenant Colonel Zachary A. Hegedish
**ROTC Army:** Lieutenant Colonel Christopher L’Heureux
**Graduate Research Assistant:** Yvette Hastings

2021-2022 Academic year: 4 Air Force cadets
2022-2023 Academic year: 2 Air Force and 2 Army cadets
2023-2024 Academic year: 2 Air Force, 1 Army, 1 civilian
ISO 25K

SOFTWARE PRODUCT QUALITY

Functional Suitability
- Functional Completeness
- Functional Correctness
- Functional Appropriateness

Performance Efficiency
- Time Behaviour
- Resource Utilization
- Capacity

Compatibility
- Co-existence
- Interoperability

Usability
- Appropriateness
- Recognizability
- Learnability
- Operability
- User Error Protection
- User Interface Aesthetics
- Accessibility

Reliability
- Maturity
- Availability
- Fault Tolerance
- Recoverability

Security
- Confidentiality
- Integrity
- Non-repudiation
- Authenticity
- Accountability

Maintainability
- Modularity
- Reusability
- Analysability
- Modifiability
- Testability

Portability
- Adaptability
- Installability
- Replaceability
Hierarchical Software QA Modeling

Standards
ISO/IEC 9126:2001
ISO/IEC 25010:2011
NIST 800-53/82
RMF (Risk Management Framework)

Quamoco (2012 Wagner et al.)
Qatch (2017 Miltiades et al.)
PIQUE (2020 SEL MSU)
CWE-699 View Structure
PIQUE Models

- Pique-Bin (INL, DHS)
- Pique-C# (CERL Army, Air Force)
- Pique-C#-Sec (CERL Army, Air Force, DHS)
- Pique-Azure (DHS)
- Pique-C++ (DHS)
- Pique-Cloud (DHS)
- Pique-ICS (DHS)
Diversity of Sources

• Variability associated with diverse sources of information is problematic:
  • data from multiple sources leads to the propagation of inconsistencies and errors
  • Accuracy and trustworthiness is hampered
  • We acknowledge that the variability inherent in vendors, tool versions, third party software, and host environments significantly influences the outcomes of security assessments
  • How do we normalize data?
  • How do we aggregate data?
Diversity of Sources

G1: Report on the high variability of SATs.
G2: Report on techniques used to aggregate results from multiple sources.
<table>
<thead>
<tr>
<th>Variability Source</th>
<th>Binaries</th>
<th>Source Code</th>
<th>Docker Containers</th>
<th>SBOMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor</td>
<td>Unexplored</td>
<td>Published [4], [7], [12]</td>
<td>WiP</td>
<td>Published [15]</td>
</tr>
<tr>
<td>Configuration</td>
<td>Expected</td>
<td>Published [4], [12]</td>
<td>Expected</td>
<td>Expected</td>
</tr>
<tr>
<td>Failures</td>
<td>WiP</td>
<td>Expected</td>
<td>WiP</td>
<td>WiP</td>
</tr>
<tr>
<td>Outputs</td>
<td>Expected</td>
<td>Expected</td>
<td>WiP</td>
<td>WiP</td>
</tr>
<tr>
<td>Dependencies</td>
<td>Published [9]</td>
<td>Expected</td>
<td>WiP</td>
<td>WiP</td>
</tr>
<tr>
<td>Environment</td>
<td>Suspect</td>
<td>Expected</td>
<td>Suspect</td>
<td>Suspect</td>
</tr>
</tbody>
</table>
Diversity of Sources

**G1** focuses on delineating the problem of reliance on one version of a SAT (e.g., the most recent version of the tool).

**G2** offers an unbiased, tool-agnostic solution that we have developed to facilitate aggregating tool findings from multiple sources.
Experimental Methods

**G1:** We focus on experimentation done on:

1. **Binary analysis tools CVE Binary Tool and CWE Checker**
   We evaluated 660 publicly accessible binaries sourced from a Kali Linux distribution with multiple versions of CWE Checker and CVE Binary Tool.

2. **Docker Images analysis tools Grype and Trivy.**
   We evaluated a single version of each of 163 Docker Official Images (i.e., containers) using the SATs Grype and Trivy. We collected these Official Images from Docker Hub.
Experimental Methods

**G2:** We report on a procedure we have developed to aggregate results from diverse SATs.

1. **Software Artifacts**
   - E.g., assemble collection of binaries

2. **SATs**
   - E.g., evaluate all binaries using cwe-checker and cve-bin-tool

3. **Aggregation File**
   - E.g., create file with counts of all CVEs and CWEs in each binary in collection

4. **Distribution of Findings**
   - E.g., evaluate distribution of counts of each CVE in all binaries in collection (histogram for one CVE plotted above)

5. **Evaluate New Artifact**
   - E.g., the count of one finding in an end user’s binary of interest

<table>
<thead>
<tr>
<th>Software Artifact Name</th>
<th>Finding Name</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Artifact of Interest</td>
<td>CVE-Unknown-Other Diagnostic</td>
<td>40</td>
</tr>
</tbody>
</table>

6. **PDF**
   - E.g., create a PDF from the counts of the CVE in all binaries in benchmark repository and score the count of a finding in an end user’s binary of interest relative to all files in the collection

7. **Score**
   - E.g., 0.66
G1 Results

(A) CVE Binary Tool

(B) CWE Checker

(C) Trivy

(D) Grype
G2 Results

Example Finding 1

Example Finding 2

Example Finding 3

Constant Value (CV)

Score ≤ CV → Score = 1

Score > CV → Score = 0
Lessons Learned and Futures

- Assessing these sources of variability simultaneously is too complex. Breaking each component down into more atomic components will facilitate understanding the nuances of each source of variability.
- We offer a primary classification for sources of variability as a first step towards developing a taxonomy for classifying variability sources (e.g., vendor, version, environment).
- The solution that we present for aggregation has the benefit of being applicable for information across a range of sources where no oracles exist.
- Sources of variability compound uncertainty. The compounding of uncertainty are inevitable side-effects of aggregation.
Quality Assurance Pipeline

Dr. Ann Marie Reinhold

Model Output

Model Inputs

Tool Outputs

Risk Communication

Converting Numeric Scores to Meaningful Risk Communication Messages
Quality Assurance Pipeline

Closing the perception gap

Risk Communication
Converting Numeric Scores to Meaningful Risk Communication Messages

Proportion of population

Perceived probability of breach

Lines represent median

Experts
Target population

0 1
Software Bill Of Materials (SBOMs)
Eric O'Donoghue (MS Student)
Software Bill Of Materials (SBOMs)

Eric O'Donoghue (MS Student)

A list of ingredients that make up software systems
Software Bill Of Materials (SBOMs)
Eric O’Donoghue (MS Student)

- CycloneDX
- SWID
- SPDX

- Structural Quality of SBOM
- Security Assessment of the Contents
The Use of AI and ML to Improve Models
Dr. Bradley Whitaker
Improving the confidence of machine learning models through improved software testing approaches
Decomposition of CWE-200

Identify security zones and sensitive sections of source code
Malware Detection Using Obfuscation of Opcodes in FPGAs

Dr. Brock LaMeres

- We control the entire hardware description of the processor, thus we control the CPU implementation.
- This means we can assign random instruction codes for each core, which prevents malware from ever infecting more than one of the redundant computers.
Near Future

- Goals: Detection, mitigation, guidance of Electrical Vehicle (EV) Infrastructure
  - Measurement of quality in EV infrastructure components
  - Develop repeatable and quantifiable processes for testing components
  - Identify test bed for solutions
  - Use hardware obfuscation techniques
Electrical Vehicle Threat Models

- Power Grid
  - Power plants
  - Transmission
  - Distribution

- EVCS: Electric Vehicle Charging Station
- EVCSMS: EVCS Management Station (Back Office)
- OCPP: Open Charge Point Protocol
- HMI: Human Machine Interface
Electrical Vehicle Threat Models

- EVCS: Electric Vehicle Charging Station
- EVCSMS: EVCS Management Station (Back Office)
- OCPP: Open Charge Point Protocol
- HMI: Human Machine Interface

Attacks on the power grid e.g., frequency instability attacks

- Gain access to HMI
- DoS attacks
- OCPP attacks
- Firmware updates

Adversary compromises public/private EVCS by attacking EVCSMS

- Hijack Accounts
- Steal PII
- Manipulate charging process
- Compromise payment gateway
• Book Club
• Independent study credit
• HackerCats club
Research Collaborations

https://www.montana.edu/cyber/
Education

• Associates degree in Cybersecurity (Gallatin College)
• MS in Cybersecurity
  • Board of Regents approved
  • Seeking CAE certification
• NSF REU program – Cybersecurity algorithms
• Griffiss/DoD program to train 4 ROTC cadets on a yearly basis before commissioning
Power of Collaboration