A Multivocal Literature Review of Current Tools for Increasing the Degree of Automation in the Development of Secure and Privacy Compliant Applications

John Nguyen, 10. August 2020, Kick-off presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
www.matthes.in.tum.de
Motivation and Background

Challenge: More and more data is being stored and processed → Volume of stolen data increases

Issues:
- Violation of security principles
  - Least privilege
  - Defense in depth
  - Minimum exposure
- Lack of security training
  - Security awareness [5]
- Weak encryption
  - MD5 encryption used by Yahoo [6]
- Conflict of interests between stakeholders
  - Focus on functional requirements
  - Developers ↔ Architects ↔ Management

Note: Worldwide; April 2020; based on number of records lost
Further information regarding this statistic can be found on page 8.
Source(s): Information is Beautiful; Various sources (VizSweet), Thomson Reuters; ID 230525
Motivation and Background (cont’d)


- 81% of breaches were contained in days or less
- 72% of breaches involved large business victims
- 37% of breaches stole or used credentials
- Web applications were involved in 43% of breaches
- 86% of breaches were financially motivated
Motivation and Background (cont’d)
Not so different after all

Waterfall

Agile

DevOps

DevOps At Scale

Time
Motivation and Background (cont’d)

Need for Security and Privacy Automation Tools

Challenges of agile software development

- Neglection of non-functional requirements
- Rapid pace of development and deployment
- Knowledge boundaries (e.g. documentation) [3][4]

Benefits

- Improving scalability
- Reducing human error [1]
- Checking continuously instead of an interval basis [2]
- Avoid cost explosion due fixing at implementation stage [9]
- Lack of security experts [10]
- Parallel testing [13]

Security and Privacy Automation is not a Silver Bullet

- Wide range of skills required
- Integration of new tools
  - Conflict with legacy systems
  - Conflict with an established mindset [7]
- Faded boundaries between the security team and developers [8]
- Additional cost
  - Need for additional education and improved culture → tool
  - Acquiring new technologies (tools)
- Cross-team cooperation necessary [1]
- Isolated tools to solve specific problems [11]
  → we look into centralized solutions

Based on [12]. DevSecOps Pyramid
Outline

1. Motivation and Background

2. Research Goal and Methodology
   a. Methodology
   b. Research Questions
   c. Search Strategy

3. Initial Results

4. Timeline
Multivocal Literature Review:
- Form of Systematic Literature Review
- Covers both the state-of-the-art and practice
- Inclusion of a large body of grey literature

Benefits:
- Certain evidence is often based on experience and opinion
- Inclusion of real-world needs in industrial settings
- May avoid publication bias

Challenges:
- Quality assessment
- Large volumes of data → need for a termination criteria
- Bias and lack of quality \[14\]
Methodology (cont’d)

Research Questions:

- RQ1: What challenges can automation tools address when developing security and privacy compliant applications?
- RQ2: How can the identified technologies and tools be classified?
- RQ3: To what extent can security and privacy activities be automated with current available automation tools?
Search Strategy

Search String:
- SLR: (secur* OR priva* OR protect*) AND ("software development" OR "software engineering") AND (tool* OR control*) AND (automat* OR "continuous") AND ("DevOps" OR "agile")
- GLR: tool AND (security OR privacy) AND automation AND (DevOps OR Agile)

Inclusion criteria for SLR:
- Written in English and full test accessible
- Accessible with TUM rights or freely accessible
- Online available
- Paper must discuss security or privacy compliance automation
- Paper must include any kind of tool, framework demonstration/implementation, prototype or similar to the support software development lifecycle
- Within the Software Engineering domain
- Paper must be published in a journal or conference paper (workshops and tutorials are excluded)
- Published between 2015-2020 (initial search)

Exclusion criteria for SLR:
- Duplicates or repeated studies
- Lack of relevance for RQs
- Algorithms and concepts
Search Strategy (cont’d)

### Outline

1. Motivation and Background
2. Research Goal and Methodology
3. Initial Results
4. Timeline
## Initial Results

<table>
<thead>
<tr>
<th>Tools</th>
<th>Source</th>
<th>Description</th>
<th>Taxonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonarqube</td>
<td><a href="https://www.sonarqube.org/">https://www.sonarqube.org/</a></td>
<td>Static code analysis</td>
<td>Development – testing – code</td>
</tr>
<tr>
<td>NMAP</td>
<td><a href="https://nmap.org/">https://nmap.org/</a></td>
<td>Network scanning tool</td>
<td>Infrastructure – scanning</td>
</tr>
<tr>
<td>Conpan</td>
<td><a href="https://github.com/neglectos/ConPan">https://github.com/neglectos/ConPan</a></td>
<td>Analysis of packages in container</td>
<td>Infrastructure – scanning</td>
</tr>
<tr>
<td>Data-driven Security Game (DdSG)</td>
<td><a href="https://github.com/dagerikhl/ddsght">https://github.com/dagerikhl/ddsght</a></td>
<td>Improve security awareness</td>
<td>Governance – training</td>
</tr>
</tbody>
</table>
1. Motivation and Background
2. Research Goal and Methodology
3. Initial Results
4. Timeline
Schedule

Analysis of Academic Literature → Analysis of Grey Literature → Tool Extraction and Synthesis → Developing Tool-Taxonomy → Mappint Tools to Challenges → Identification of non-automatable or non-tool-supported activities

Thesis Kickoff: 15 Jun
Submission: 29 Oct
Thesis completion: 15 Nov

2020

Jun

July 2020

Today

Preparation Phase

Research Phase

Aug

62 days

Writing Phase

50 days

Correction Phase

12 days

Buffer

13 days

28 Oct - 15 Nov

Nov

2020
References

Backup Slides
**About me**

Name: John Nguyen (john.nguyen@tum.de)  
Current Study: Information Systems (6\textsuperscript{th} Semester)  
Faculty: Informatics

**Thesis Organization**

- **Chair**: Software Engineering for Business Information Systems (sebis)
- **Type**: Bachelor Thesis
- **Supervisor**: Prof. Dr. Florian Matthes
- **Advisor**: Sascha Nägele
- **Timeframe**: 15.06.2020 – 15.11.2020

**Relevant Courses**

- SEBA Bachelor
- Seminar: Trends der Wirtschaftsinformatik - Agilität und DevOps
- Security Engineering (in progress)
- iLab1
Knowledge of developers (2018)

100 employees of a selected industrial company

"Which tools do you know and/or use?"

Response options:

- (DevOps tools) Jenkins; Kubernetes; TeamCity; Spinnaker; Travis; GoCD; Concourse CI; JFrog Artifactory;
- (static analysis tools) PMD; Checkstyle; FindBugs; FindBugs Security;
- (security tools) OWASP ZAP; BDD Security; JFrog Xray; Security Monkey; Black Duck; Snyk" [1]
Security Standards and Modell (2016)

- “What application security standards or models do you follow? Select all that apply.” [5]
- 435 respondents
- OWASP Top 10 leading standard

Shackleford, D. (2016). A DevSecOps Playbook. SANS Institute InfoSec Reading Room. A DevSecOps Playbook Figure 2 p. 11 [5]
OWASP

- International non-profit organization
- Focus on web application security
- Goal: Improve web application security
- OWASP Top 10: represents top 10 risks (critical) [3]

<table>
<thead>
<tr>
<th>OWASP Top 10 - 2013</th>
<th>OWASP Top 10 - 2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1 - Injection</td>
<td>A1:2017-Injection</td>
</tr>
<tr>
<td>A2 - Broken Authentication and Session Management</td>
<td>A2:2017-Broken Authentication</td>
</tr>
<tr>
<td>A3 - Cross-Site Scripting (XSS)</td>
<td>A3:2017-Sensitive Data Exposure</td>
</tr>
<tr>
<td>A4 - Insecure Direct Object References [Merged+A7]</td>
<td>A4:2017-XML External Entities (XXE) [NEW]</td>
</tr>
<tr>
<td>A5 - Security Misconfiguration</td>
<td>A5:2017-Broken Access Control [Merged]</td>
</tr>
<tr>
<td>A6 - Sensitive Data Exposure</td>
<td>A6:2017-Security Misconfiguration</td>
</tr>
<tr>
<td>A8 - Cross-Site Request Forgery (CSRF)</td>
<td>A6:2017-Insecure Deserialization [NEW, Community]</td>
</tr>
<tr>
<td>A9 - Using Components with Known Vulnerabilities</td>
<td>A9:2017-Using Components with Known Vulnerabilities</td>
</tr>
<tr>
<td>A10 - Unvalidated Redirects and Forwards</td>
<td>A10:2017-Insufficient Logging&amp;Monitoring [NEW, Comm.]</td>
</tr>
</tbody>
</table>

https://www.heise.de/developer/imgs/06/2/3/6/0/4/7/9/2013-2017-5068086774500.png [Date: 04.06.2020]
BSIMM

- Software security framework
- 12 practices (in 4 domains) and 119 activities
- Domains:
  - Governance
  - Strategy & Metrics
  - Compliance & Policy
  - Training
  - Intelligence
  - Attack Models
  - Security Features & Design
  - Standards & Requirements
  - SSDL Touchpoints
  - Architecture Analysis
  - Code Review
  - Security Testing
- Deployment
  - Penetration Testing
  - Software Environment
  - Configuration Management & Vulnerability Management [4]

**TEN CORE ACTIVITIES “EVERYBODY” DOES**

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SM1.4]</td>
<td>Identify gate locations and gather necessary artifacts.</td>
</tr>
<tr>
<td>[CP1.2]</td>
<td>Identify PII obligations.</td>
</tr>
<tr>
<td>[T1.1]</td>
<td>Provide awareness training.</td>
</tr>
<tr>
<td>[AM1.2]</td>
<td>Create a data classification scheme and inventory.</td>
</tr>
<tr>
<td>[SFD1.1]</td>
<td>Build and publish security features.</td>
</tr>
<tr>
<td>[SR1.3]</td>
<td>Translate compliance constraints to requirements.</td>
</tr>
<tr>
<td>[AA1.1]</td>
<td>Perform security feature review.</td>
</tr>
<tr>
<td>[CR1.2]</td>
<td>Have SSG perform ad hoc review.</td>
</tr>
<tr>
<td>[ST1.1]</td>
<td>Ensure QA supports edge/boundary value condition testing.</td>
</tr>
<tr>
<td>[PT1.1]</td>
<td>Use external penetration testers to find problems.</td>
</tr>
<tr>
<td>[SE1.2]</td>
<td>Ensure host and network security basics are in place.</td>
</tr>
<tr>
<td>[CMVM1.2]</td>
<td>Identify software bugs found in operations monitoring and feed them back to development.</td>
</tr>
</tbody>
</table>

[10.06.2020]
Figure 11. Effective Appsec Security Practices

## Quality Assessment in MLRs

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Questions</th>
</tr>
</thead>
</table>
| Authority of the producer       | • Is the publishing organization reputable? E.g., the Software Engineering Institute (SEI)  
• Is an individual author associated with a reputable organization?  
• Has the author published other work in the field?  
• Does the author have expertise in the area? (e.g., job title principal software engineer) |
| Methodology                     | • Does the source have a clearly stated aim?  
• Does the source have a stated methodology?  
• Is the source supported by authoritative, contemporary references?  
• Are any limits clearly stated?  
• Does the work cover a specific question?  
• Does the work refer to a particular population or case? |
| Objectivity                     | • Does the work seem to be balanced in presentation?  
• Is the statement in the sources as objective as possible? Or, is the statement a subjective opinion?  
• Is there vested interest? E.g., a tool comparison by authors that are working for particular tool vendor  
• Are the conclusions supported by the data? |
| Date                            | • Does the item have a clearly stated date? |
| Position w.r.t. related sources | • Have key related GL or formal sources been linked to / discussed? |
| Novelty                         | • Does it enrich or add something unique to the research?  
• Does it strengthen or refute a current position? |
| Impact                          | • Normalize all the following impact metrics into a single aggregated impact metric (when data are available): Number of citations, Number of backlinks, Number of social media shares (the so-called “alt-metrics”), Number of comments posted for a specific online entries like a blog post or a video, Number of page or paper views |
| Outlet type                     | • 1st tier GL (measure=1): High outlet control / High credibility: Books, magazines, theses, government reports, white papers  
• 2nd tier GL (measure=0.5): Moderate outlet control/ Moderate credibility: Annual reports, news articles, presentations, videos, Q/A sites (such as StackOverflow), Wiki articles  
• 3rd tier GL (measure=0): Low outlet control/ Low credibility: Blogs, emails, tweets |

Garousi, V., Felderer, M., & Mäntylä, M. V. (2017). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Information and Software Technology, 106, 101-121; Table 7-Quality assessment checklist of grey literature for software engineering
References


