Architectural Solutions for Next Generation Software Systems

Presenter: Faheem Ullah & Nguyen K. Tran
PhD Students

Supervisor: M. Ali Babar

CREST – The Centre for Research on Engineering Software Technologies
The University of Adelaide, Australia
faheem.ullah@adelaide.edu.au, nguyen.tran@adelaide.edu.au
“Helping industries, governments, and societies to build human- as well as technological-based competencies in software systems engineering”

Faheem Ullah
Ph.D. Student – The University of Adelaide
“Engineering big data security analytics solutions”

Nguyen Tran
Ph.D. Student – The University of Adelaide
“Engineering search engine systems for Internet of Things”
Outline

• Architecting for Big Data Cybersecurity Analytics
  – How software architecture enables the achievement of the quality of service delivered by Big Data Cybersecurity Analytics Systems

• An Architectural Solution for Internet of Things Search Engines
  – How software architecture enables Search Engine Systems for the Future Internet and research on these systems.
How software architecture enables the achievement of the quality of service delivered by Big Data Cybersecurity Analytics Systems

ARCHITECTING FOR BIG DATA CYBERSECURITY ANALYTICS
Outline

▪ Introduction

▪ Architectural Tactics for Big Data Cybersecurity Analytics: A Systematic Literature Review

▪ Towards Evidence-Based Understanding of Architectural Tactics for Cybersecurity Analytics

▪ Architecture-Driven Self-Adaptation for Cybersecurity Analytics
Big Data Cybersecurity Analytics

A research domain that leverage big data technologies for analysing security events data to protect organizational networks, computers, and data from cyber attacks

Architectural Tactics for Big Data Cybersecurity Analytics: An SLR

Research Questions

**RQ1:** Which are the most important quality attributes for security analytic systems?

**RQ2:** What are the architectural tactics for addressing quality concerns in security analytic systems?
Quality Attributes

- Performance
  - Realtime response required to attacks
  - Size and speed of security event data hinders real-time response
- Accuracy
  - Catastrophic consequences of letting attack go undetected
  - Only detect attacks and not shield legitimate access
- Scalability
  - Challenging to estimate the speed and size of security data
  - Long period attacks such as Advanced Persistent Threats
- Reliability
  - Reliable data collection for ensuring attack detection
  - High speed security data input can crash the resources
- Usability
  - Unfriendly system can lead to delay in response to attack
  - Large number of alerts generated by the system
- Interoperability
  - Collaborate with other security systems i.e., security orchestration
  - Data collection from a variety of sources
- Adaptability
  - Adapt to comply with the required Quality of Service
  - Changing operating environments e.g., network topology
- Modifiability
  - Updating attacks signatures to detect new attacks
  - Incorporating new tools and technologies
- Privacy
  - Comply with the privacy laws while analysing the data
  - Avoid processing content of a packet

Faheem Ullah & Nguyen Tran

Architectural Solutions for Next Generation Software Systems

CREST

8
Architectural Tactics

Architectural Tactics for Security Analytics

Performance
- Parallel Processing
- Unnecessary Data Removal
- Feature Selection and Extraction
- Result Polling and Optimized Notification
- Data CutOff
- Algorithm Optimization

Accuracy
- Attack Detection Algorithm Selection
- Combining Multiple Detection Methods
- Combining Signature-based and Anomaly-based Detection

Scalability
- Dynamic Load Balancing
- MapReduce

Reliability
- Maintaining Multiple Copies
- Data Ingestion Monitoring
- Dropped NetFlow Detection

Security
- Secure Data Transmission

Usability
- Alert Ranking
Future Research Areas

- **Under addressed Quality Attributes**
  - Several quality attributes such as *interoperability, adaptivity, modifiability, generality, and stealthiness* requires further investigation from the architectural perspective.

- **Tactics Evaluation**
  - The codified tactics should be *evaluated both qualitatively and quantitatively* to investigate their impact on various quality attributes.

- **Quality Trade-offs among tactics**
  - The *quality trade-offs among the tactics* should be established to help a software architect select the required set of tactics.

- **Dependencies among tactics**
  - Considering that the tactics cannot be applied in isolation, it is important to explore the possible *dependencies and collaborations among the codified set of tactics*.

- **Modelling the tactics**
  - To facilitate the software architect, the codified tactics need to be modelling using a *standard modelling language such as UML*.
Validation of Tactics

Motivation

- Establishing quality trade-offs among the tactics
- Developing evidence-based design space
- Quantification of the contribution of tactics to the intended quality attributes

Research Questions

What is the impact of the

**RQ1:** Removal of Duplicates Tactic
**RQ2:** Feature Selection and Extraction Tactic
**RQ3:** Alert Ranking Tactic

on the accuracy and response time of a security analytics system
## Empirical Findings

<table>
<thead>
<tr>
<th>Tactic</th>
<th>Quality Attribute</th>
<th>Without the Tactic</th>
<th>With the Tactic</th>
<th>Wilcoxon’s P</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Removal of Duplicates</strong></td>
<td>Accuracy</td>
<td>DR</td>
<td>91.532</td>
<td>91.533</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPR</td>
<td>29.632</td>
<td>29.621</td>
</tr>
<tr>
<td></td>
<td>Response Time</td>
<td>TrT</td>
<td>2090.2</td>
<td>2087</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PrT</td>
<td>19.6</td>
<td>19.0</td>
</tr>
<tr>
<td><strong>Feature Selection and Extraction</strong></td>
<td>Accuracy</td>
<td>DR</td>
<td>91.403</td>
<td>91.404</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPR</td>
<td>6.706</td>
<td>6.706</td>
</tr>
<tr>
<td></td>
<td>Response Time</td>
<td>TrT</td>
<td>632</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PrT</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td><strong>Alert Ranking</strong></td>
<td>Response Time</td>
<td>TrT</td>
<td>632</td>
<td>625</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PrT</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

**RQ1** Removal of Duplicates Tactic improves DR by **1.11%**, reduces FPR by **22.26%**, and improves TrT by **89.74%**.

**RQ2** Feature Selection and Extraction Tactic reduces DR by **12.86%**, increases FPR by **7.15%**, and improves TrT and PrT by **17.43%** and **1.93%**.

**RQ3** Alert Ranking Tactic improves usability, which leads to enhanced accuracy, but increases PrT by **98.11%**.

DR – Detection Rate  
FPR – False Positive Rate  
TrT – Training Time  
PrT – Prediction Time
Motivation

- Accuracy and response time are the most significant quality attributes
- Increasing accuracy reduces response time and vice versa
Architecture-Driven Self-Adaptation for Security Analytics

<table>
<thead>
<tr>
<th>C#</th>
<th>Component Name</th>
<th>Contributes to</th>
<th>Contr. Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Duplicates Removal</td>
<td>Accuracy</td>
<td>A=+0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response time</td>
<td>R=+0.6</td>
</tr>
<tr>
<td>C2</td>
<td>Data CutOff</td>
<td>Response time</td>
<td>A=0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.3</td>
</tr>
<tr>
<td>C3</td>
<td>Dynamic Feature Selection</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.2</td>
</tr>
<tr>
<td>C4</td>
<td>Static Feature Selection</td>
<td>Response time</td>
<td>A=+0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.2</td>
</tr>
<tr>
<td>C5</td>
<td>Result Polling</td>
<td>Response time</td>
<td>A=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.02</td>
</tr>
<tr>
<td>C6</td>
<td>ML Algorithm Application</td>
<td>Accuracy</td>
<td>A=+0.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response time</td>
<td>R=+0.5</td>
</tr>
<tr>
<td>C7</td>
<td>ML Algorithm Selector</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.1</td>
</tr>
<tr>
<td>C8</td>
<td>Signature-based Detection</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.8</td>
</tr>
<tr>
<td>C9</td>
<td>Alert Correlation</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.6</td>
</tr>
<tr>
<td>C10</td>
<td>False Positive Reduction</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.3</td>
</tr>
<tr>
<td>C11</td>
<td>Alert Ranking</td>
<td>Accuracy</td>
<td>A=+0.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.8</td>
</tr>
<tr>
<td>C12</td>
<td>Hadoop Booster</td>
<td>Response time</td>
<td>A=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.1</td>
</tr>
<tr>
<td>C13</td>
<td>Adjusting MapReduce Jobs</td>
<td>Response time</td>
<td>A=0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=+0.05</td>
</tr>
<tr>
<td>C14</td>
<td>MinMax Normalization</td>
<td>Accuracy</td>
<td>A=+0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R=0</td>
</tr>
</tbody>
</table>


\[
F(wf) = \sum_{k=1}^{K} w^k \cdot V(k) + D(wf)
\]

Faheem Ullah & Nguyen Tran
Summary

Systematic Literature Review
- 12 Quality Attributes
- 17 Architectural Tactics
- 5 Future Research Areas

Empirical Validation of Architectural Tactics
- Removal of Duplicates
- Feature Selection and Extraction
- Alert Ranking
- Accuracy
- Response Time

Architecture-Driven Self-Adaptation
- Runtime architecture adaptation
- Utility function and workflow for optimizing QoS
- QoS
How software architecture enables Search Engine Systems for the Future Internet and research on these systems.

AN ARCHITECTURAL SOLUTION FOR INTERNET OF THINGS SEARCH ENGINES
Background – Internet of Things Search Engines

Find “meeting room which reporting abnormal energy consumption”

Find “Available parking bay nearest to the Uni”

On-campus IoT Infra.

Smart-city IoT Infra.

Room A1 and B2

Try bay 6 on Grenfell str.

Faheem Ullah & Nguyen Tran
Motivating Scenario

Architecting for reuse and composition?

Software Infrastructure

Which components? Which architecture patterns?

Internet of Things Search Engine 1
- Detector 1
- Storage 1
- Index 1
- Query 1

Internet of Things Search Engine 2
- Detector 2
- Storage 2
- Index 2
- Query 2

Internet of Things Search Engine 3

IoTSE Composition and Deployment Pattern

Faheem Ullah & Nguyen Tran
Research Problem

“How to enable the construction of Internet of Things Search Engine from independently developed and reusable components?”
Research Questions

“How to enable the construction of Internet of Things Search Engine from independently developed and reusable components?”

RQ1: What components and patterns constitute IoTSE?

Reference Architecture for IoTSE

RQ2: How to architect IoTSE to be supportive to composition and reuse?

Service-oriented Architecture (SOA) for IoTSE

RQ3: How to support the application of the architectural solution?

Software Platform for Development, Composition and Deployment of IoTSE
Guiding Principles:

P1: Independence of component developers is the top priority.

P2: Workflow logic is separated from the logic of IoTSE component.

P3: Deployment is separated from the logic of IoTSE component.
Results

IoTSE Reference Architecture

Published: CSUR
WIP & Submitted: CACM, TSE

Platform for Composition and Deployment of IoTSE

Published: WISE 2017
WIP & Submitted: ASE 2018

WIP & Submitted: WISE 2018
Unit of measurement: Celsius

Observed Property: Apparent Temperature

Sensor reading: 24.1044 degrees
Results – Impact of architecture

Parallelism actually **INCREASES THE RESPONSE TIME**!
Results

Finding available parking bay around a certain radius in Adelaide
(Simulated data generated from real sensor data)
Summary

• Architectural solution provides the glue to bring independent components together for next-gen software systems

• Potential for new research
Thank You!

Questions and Comments!

Faheem Ullah, Nguyen Tran
CREST – Centre for Research on Engineering Software Technologies
The University of Adelaide, Australia
faheem.ullah@adelaide.edu.au, nguyen.tran@adelaide.edu.au