IDAACS 2015, 24.-26.09.15

Intelligent monitoring with background knowledge

Kai-Oliver Detken · Carsten Elfers · Malte Humann
Thomas Rix · Stefan Edelkamp

DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen
https://www.decoit.de
detken@decoit.de

Outline

- Introduction
- Architecture of the System
- Knowledge Representation
- Knowledge Exchange Process
- Tolerant Pattern Matching
- Integration of Sensors
- Anomaly Detection
- Experimental Results
- Conclusion and Outlook

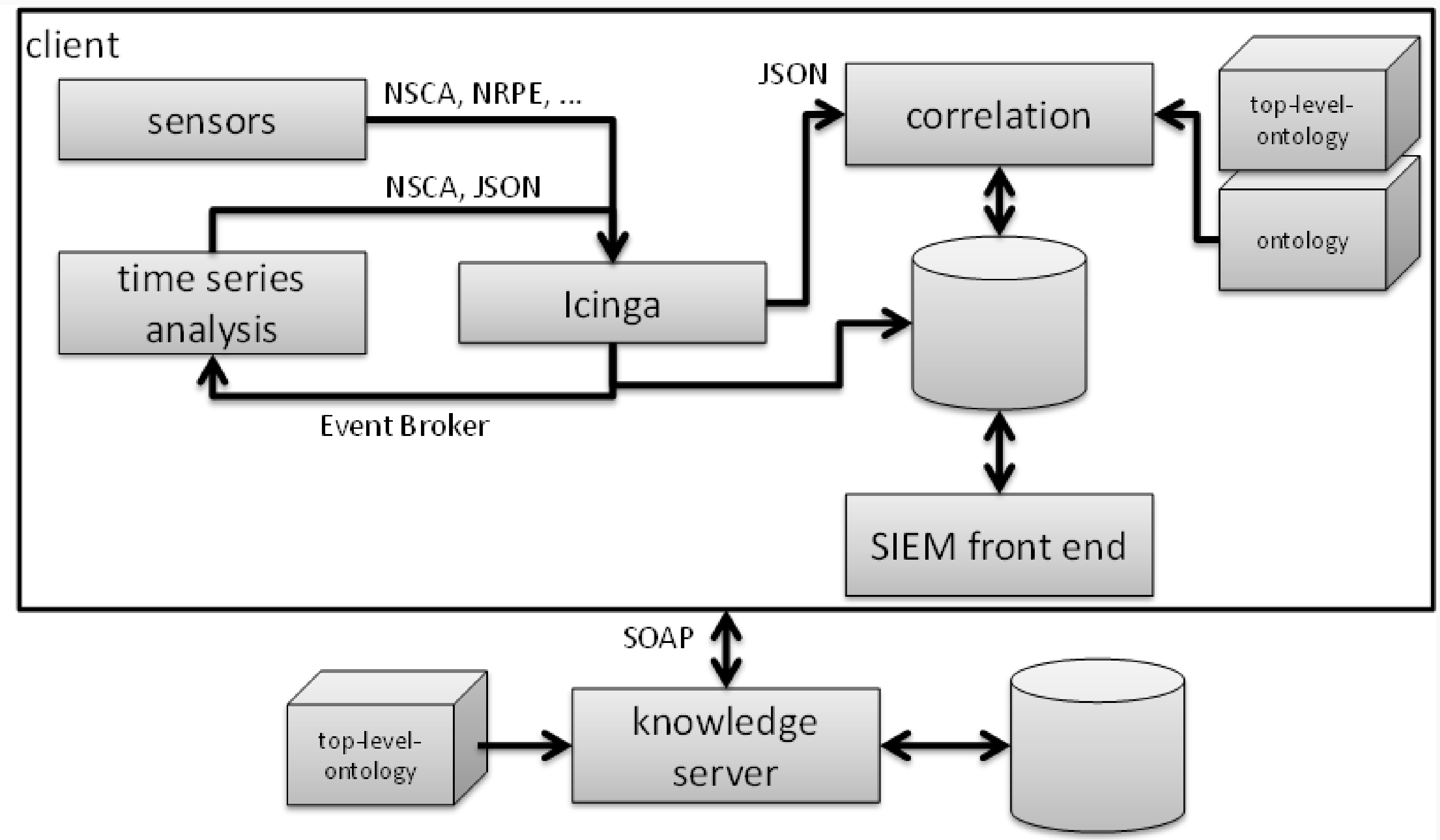
Introduction (1)

- The term SIEM is divided into:
  - Security Event Management (SEM)
  - Security Information Management (SIM)
- SEM security management includes:
  - Real-time monitoring
  - Event correlation
  - Event messaging
- Security management of SIM includes:
  - Long-term capturing
  - Analyze of log data
  - Reporting of log data
- Basically SIEM systems are able by collecting sensor information and events to recognize anomalies and prevent threats

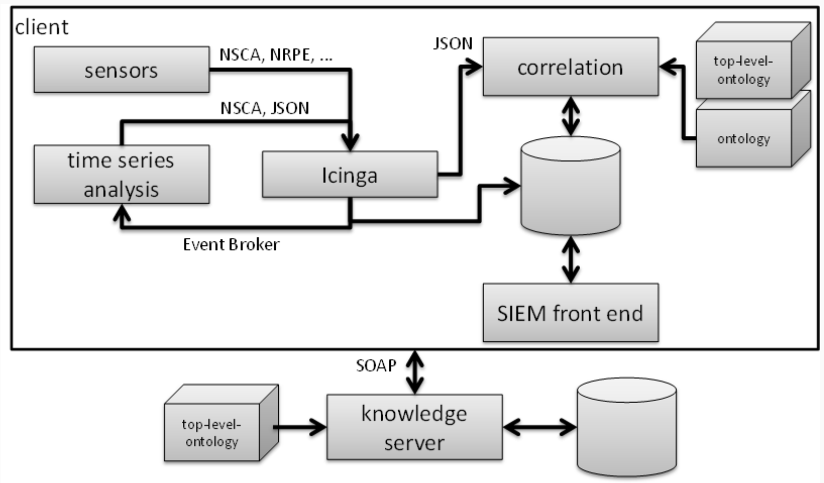
Introduction (2)

- iMonitor project (www.imonitor-project.de) of BMWi started in July 2013 and ended in June 2015 successfully
- Partner of the „Bremer" project were:
  - DECOIT GmbH (coordination, development, exploitation)
  - University of Bremen, TZI (development)
  - neusta GmbH (development, exploitation)
- The project developed a new form of event correlation, which recognize new attack variants automatically (with artificial intelligence)
- Exchange rules through a central knowledge server
- An event overview is presented in one SIEM-GUI centrally

Architecture of the System (1)



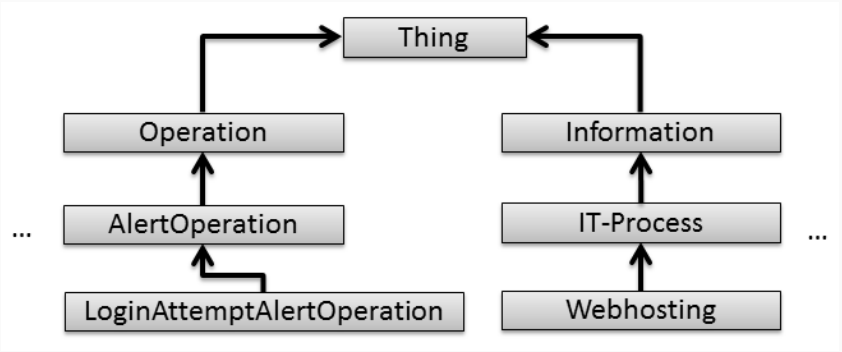
Architecture of the System (2)



- **Sensors:** Analyzing tools as Snort, Nmap, and OpenVAS for collecting data of the environment of a company regarding recognition of the normal behavior
- **Long-term analyzing:** Analyze of the usual behavior and recognition of anomalies without typical pattern detection
- **Time series analysis:** Detect anomalies from different data sources provided through Icinga
- **GUI:** Graphical view of the SIEM modules to recognize processes, events, and tickets for a definition of a risk analyzes
- **Correlation:** Processing of the sensor event data and creation of a report with a proposal of recommendations for action
- **Knowledge server:** External source for exchanging correlation rules

Knowledge Representation

- One advantage of using ontology based correlation is the flexibility of background-knowledge that can be used to detect incidents
- The background-knowledge is well structured by the T-Box part of an ontology defining a common language for all objects that are needed for the correlation
- To allow time based correlation the ontology is structured into two parts: static *information* and dynamic *operations*
  - Information part includes background-knowledge (e.g. asset information)
  - Operation part holds information that may be included and used by rules



Knowledge Exchange Process

- Data first needs to be committed to a central server (called knowledge server) prior to being downloaded by any client
- Before the rule is committed to the server, the committing client checks that the rule only uses concepts from the top-level ontology, i.e. the T-Box part that is used by all clients
- The knowledge server checks again if *no individual knowledge* has been conveyed to guarantee that the rule can be integrated into each client using the same top-level ontology
- New rules need to be verified by a neutral moderator who accepts the given rule

Tolerant Pattern Matching

- Since the available rules won't cover all possible incidents a tolerant pattern matching approach is used to find similar rules (instead of suppressing the event)
- The rule definitions are specifically annotated by SPARQL functions to allow abstraction and specify how the abstraction is performed

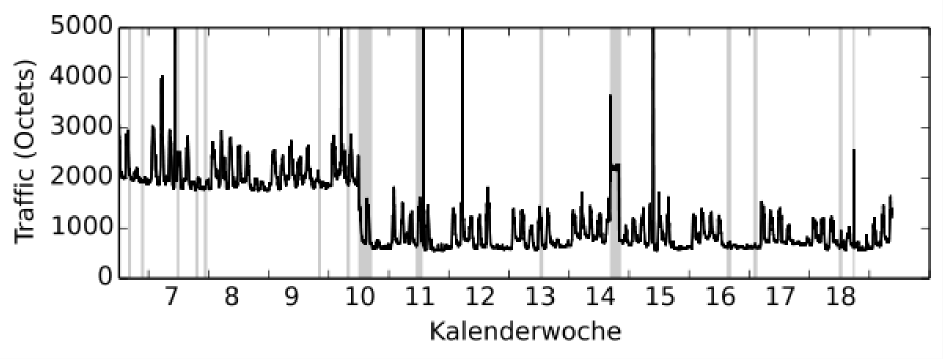
```
ASK { ?servicestate imonitor:hasName "$SERVICESTATE".
      ?servicestate a imonitor:ServicestateCritical.
      ?service a imonitor:Service.
      ?service imonitor:hasName "$SERVICEDESC".
      ?highcriticality a imonitor:CriticalityHigh.
      ?service imonitor:hasCriticality ?highcriticality.
      ?service imonitor:affects ?customers. }
```

Integration of Sensors

- To integrate new sensors (e.g. for Android), each correlation requires mapping of vendor specific event information to a common language used by the correlation
- This *mapping* can be modeled manually into the ontology or may already be given by the top-level-ontology; however, it limits the user in integrating its own individual sensors
- A tool called sensor mapper was developed to support the user
  - It reads the possible outcomes of a sensor from a file and compares it to known elements from the ontology
  - The user can optionally specify a top concept for the comparison to avoid that the tool tries to match all elements in the ontology

Anomaly Detection (1)

- A *time series* based anomaly detection was developed to further add to the correlation
- It is capable of finding anomalies in generic time series which consist of numerical data for different points in time
- This includes the optional performance data provided by Icinga for, e.g., general host and service checks via SNMP

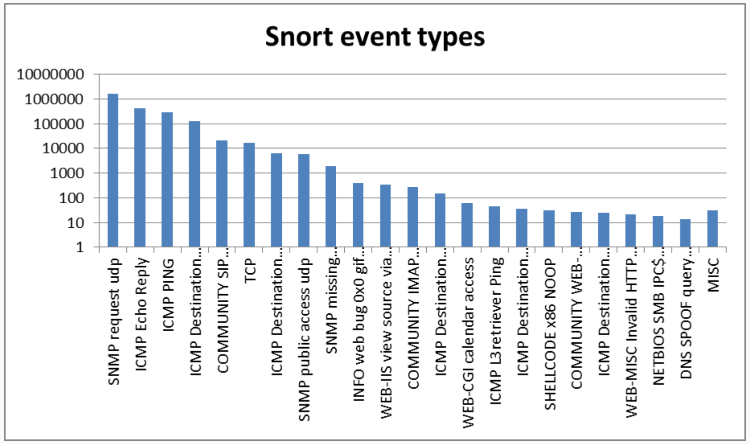


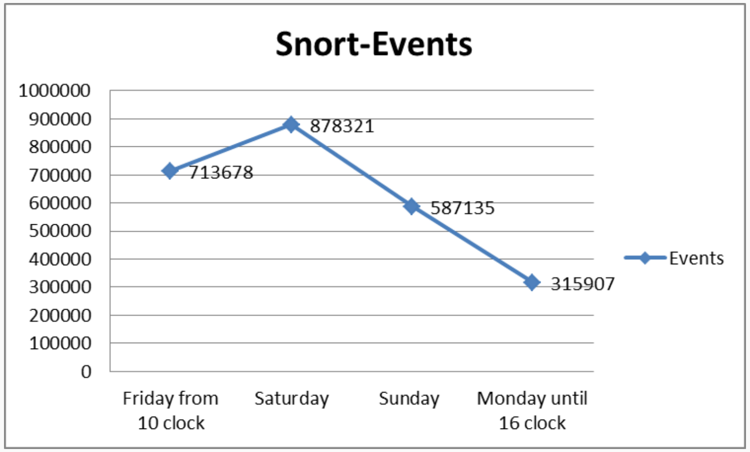
Anomaly Detection (2)

- Anomalies are predicted based on previous behavior
- To keep the configuration to a minimum the required *patterns of normal behavior* are extracted automatically…
  - …by combining the information provided by the periodical diagram and circular autocorrelation of the time series…
  - …and removing statistical outliers (to not include potential anomalies in the normal behavior)
- An incident is reported to Icinga if multiple anomalies occur in a short period of time (to prevent noise generating false positives)

Experimental Results

- Correlation performance depends on the complexity of the rules and background knowledge
- We made experimental results by analyze the traffic behavior in our company:
    - 406 events per second with one rule
    - 187 events per second with five rules
- Big data problem exists if all events are stored unfiltered in one database
- More sufficient since Icinga can be used for pre-filtering events





Demonstrator-Video

Conclusion and Outlook

- Improves monitoring to reduce the work load of system and security administrators generated by the maintenance of their computer infrastructure
- Correlate and condense events that are triggered by different sensor source
- Learn, generalize and exchange the knowledge that comes from particular observations
- Detect unknown incidents and handle them during the correlation process
- We plan to integrate an automated approach for the tight integration of an intelligent network scanner

Thank you!
...for your attention.

DECOIT GmbH
Fahrenheitstraße 9
D-28359 Bremen

https://www.decoit.de
info@decoit.de