Projects I’m currently involved in…

Expedite (FFG; TU, WU, OYD)
- myData, GDPR right of access
- Vocabulary development (W3C DPCVG) and interoperability
- Goal: Roadmap and technical groundwork for personal data ecosystems

SEMCON (FFG; TU, OYD)
- Develop concept of “semantic containers”
- Context: DMA, OYD
- Goal: Interoperability, data mobility

CitySPIN (FFG; TU, WU, SWC, SWTW)
- City-wide infrastructures (district heating and public transport networks)
- Cyber-physical social systems
- Data infrastructure and integration challenges

SEPSES (FWF/netldee Science ; TU, SBA)
- SEmantic Processing of Security Event Streams
- Logging infrastructures + Semantic Technologies + Semantic exchange of threat intelligence + Abstract patterns and reasoning…
SEPSES
SEmantic Processing of Security Event Streams

This work was supported by the Austrian Science Fund (FWF) and netidee SCIENCE under grant P30437-N31, as well as the COMET K1 program by the Austrian Research Promotion Agency.
Just recently (pretty arbitrary example)...

BA customers’ credit card details ‘probably already for sale’

The credit card details of 380,000 British Airways customers could already be on sale on the internet after the airline suffered a "malicious" data breach, experts have warned.

The online theft saw details stolen including name, email address and credit card information, including the CVV code. BA has said that its encryption was not breached but that the hackers used other "very sophisticated" methods. Cybersecurity experts speculated that the inclusion of CVV its website and app over a two-week period between 21 August and 5 September. The airline said it would compensate passengers for any losses, priorising the potential for large payouts given the number of customers.

British Airways vows to compensate passengers after data breach

Customers express anger over hacking of their personal information

British Airways disclosed that hackers had stolen data relating to about 380,000 customers from its website. The airline said it would compensate passengers for any losses, priorising the potential for large payouts given the number of customers.
Security Challenges:
• Increasingly sophisticated attacks
• Indicators of compromise hidden in high-volume log data
• Difficult to connect disparate clues in scattered, heterogeneous logs

State-of-the-art approaches:
• SIEM systems: log aggregation, (rudimentary) correlation, alerting
• Intrusion detection/prevention (IDS/IPS) systems: statistical models or signatures

Not particularly good at:
– interpreting potential clues
– contextualizing them
– establishing causal links between clues
– detecting previously unseen attacks

Consequences
• Tedious manual analyses → Lack of situational awareness
• False positives → “Alert fatigue”
• Long detection and response times when incidents do occur
The good news...

Systems generate vast amounts of log data...

.. and there are tools to collect it.
The bad news..

- Logs are weakly structured, use varying formats, and inconsistent terminology
- Interpretation is difficult, requires background knowledge, and is highly context-dependent
- Connecting clues from disparate, heterogeneous logs is hard
- There is just too much of it for human experts to analyze …
Semantics to the rescue..

Characteristics of semantic technologies:

- **Graph-based**
  - flexible schemas
  - flexible querying
  - context-rich representation

- **Explicit Semantics**
  - terminological clarity
  - "machine-readability"
  - integration
  - reasoning

- **Decentralization**
  - linking
  - federation
  - alignment
  - reconciliation

- **Standards-based**
  - SPARQL
  - JSON-LD
  - RDF

**Semantic foundation for security analytics**
(monitoring, threat intelligence and detection, forensics, incident response etc.)
Semantic security log analysis: Architecture
Vocabulary design

- **Log level:**
  - Core vocabulary: basic logging concepts
  - Source-specific extensions
  - Largely literal representation for efficient transformation
  - Use existing vocabularies where directly possible

- **Event level:**
  - Enriched and extracted events
  - 1..n log entries → single log event
  - Vocabularies security and process knowledge

- **High-level event level**
  - events from multiple sources
  - patterns of related events

+ System vocabularies
+ Threat pattern vocabularies
Design goals:
- Preliminary harmonization, uniform representation
- Initial lifting for many sources in a scalable manner
- Large-scale, high-throughput log processing
- (near) real-time extraction
- Support for stream processing
- “Extension points” for subsequent enrichment, alignment, entity reconciliation and linking with background knowledge

Approach:
- Independent extraction for each log source
- Add semantics to raw JSON stream through JSON-LD @context injection (or mappings)
- Add nodes for subsequent linking (e.g., anonymous assets)
Example: Extracted log message (JSON-LD)

Apr 9 09:37:47 kabul-VirtualBox systemd[1]: Mounted Huge Pages File System.
- **System knowledge:**
  Capture organization-specific concepts, assets, policies..
  e.g., users, network components etc.

- **Event knowledge:**
  Event definitions and associated extraction patterns

- **Threat knowledge:**
  Threat intelligence (indicators of compromise, common attack patterns,..)

**Challenges:**
- Knowledge evolves dynamically
  e.g., IPs, devices, projects etc.
- Temporal context is important!
Declarative Linking Patterns (Silk)

Linking predicate: sameAs

Challenges:
- Inconsistent identifiers (e.g., IP, host name) → entity reconciliation
- Unstable identifiers (e.g., DHCP leases, PIDs) → temporal reasoning (?)
- Linking target may not exist in BGK → discovery mechanism

Example linking rule to establish equivalency based on host name:

```xml
<LinkageRule linkType="owl:sameAs">
  <Compare id="levenshteinDistance1" required="false" weight="1" metric="levenshteinDistance" threshold="0.0" indexing="true">
    <TransformInput id="lowerCase1" function="lowerCase">
      <Input id="sourcePath1" path="/syslog:hostName"/>
    </TransformInput>
    <TransformInput id="lowerCase2" function="lowerCase">
      <Input id="targetPath1" path="/bgk:hostName"/>
    </TransformInput>
    <Param name="minChar" value="0"/>
    <Param name="maxChar" value="z"/>
  </Compare>
</LinkageRule>
```
Example: Linking to Background Knowledge
- Virtual machine with Apache web server (syslog and Apache log)
- Log acquisition: filebeat
- Custom Logstash configuration to produce JSON-LD log entries
Example Query and Results

**[Q1] All log messages in a particular time window**

```sql
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX scl: <http://purl.org/sepses/vocab/log/coreLog#>
PREFIX sysbg: <http://purl.org/sepses/bg/system#>

SELECT ?time ?logType ?hostName ?ipAddress ?hostType ?message 
WHERE { ?logEntry a scl:LogEntry;
    scl:originatesFrom ?host;
    scl:hasLogType ?logType;
    scl:logMessage ?message;
    scl:timestamp ?time.
    ?host sysbg:hostType ?hostType;
    scl:hostName ?hostName;
    scl:ipAddress ?ipAddress.
FILTER(?time > "2018-04-09T07:29:00+00:00"^^xsd:dateTime \&\&
?time < "2018-04-09T07:34:00+00:00"^^xsd:dateTime) }
```

<table>
<thead>
<tr>
<th>time (xsd:dateTime)</th>
<th>logType</th>
<th>hostName</th>
<th>ipAddress</th>
<th>hostType</th>
<th>message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018-04-09T07:29:15+00:00</td>
<td>scl:syslog</td>
<td>kabul-VirtualBox</td>
<td>192.168.0.164</td>
<td>DatabaseServer</td>
<td>&quot;org.debian.apt[683]: …&quot;</td>
</tr>
<tr>
<td>2018-04-09T07:31:45+00:00</td>
<td>scl:apache</td>
<td>linux-Machine</td>
<td>192.145.0.124</td>
<td>WebServer</td>
<td>&quot;GET /presentations/ &quot;</td>
</tr>
<tr>
<td>2018-04-09T07:31:45+00:00</td>
<td>scl:apache</td>
<td>linux-Machine</td>
<td>192.145.0.124</td>
<td>WebServer</td>
<td>&quot;GET /presentations/ …&quot;</td>
</tr>
<tr>
<td>2018-04-09T07:31:45+00:00</td>
<td>scl:syssyslog</td>
<td>kabul-VirtualBox</td>
<td>192.168.0.164</td>
<td>DatabaseServer</td>
<td>&quot;systemd-tmpfiles[3572]: …&quot;</td>
</tr>
<tr>
<td>2018-04-09T07:31:45+00:00</td>
<td>scl:syssyslog</td>
<td>kabul-VirtualBox</td>
<td>192.168.0.164</td>
<td>DatabaseServer</td>
<td>&quot;systemd[1]: Started …&quot;</td>
</tr>
</tbody>
</table>
Example: reconciliation with external threat intelligence

CVE: Common Vulnerabilities and Exposures
- Semi-structured public vulnerabilities list published by MITRE
- XML-based
- Attributes:
  - CVE-ID
  - Date, Description, References, ...

Common Platform Enumeration
- Vulnerable software and versions
- CVE-PDE mapping available

Query example:
- Match CVEs to affected assets based on background knowledge
- Link to relevant log stream
- ..
Example Query with background knowledge

**[Q2] Vulnerability information from CVE linked to affected server**

```
PREFIX cve_ex: <http://example.org/sepses/cve#>
PREFIX win: <http://purl.org/sepses/vocab/log/winEventLog#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX cve: <http://purl.org/sepses/vocab/cve#>

SELECT ?host ?programName ?programVersion ?CVE ?CVEDesc
WHERE {   ?logEntry a win:WindowsHighLevelEvent ;
    win:sourceHost ?host ;
    win:programName ?programName ;
    cve:hasVulnerability ?CVE .
    ?CVE cve:CVEDesc ?CVEDesc ;
    cve:ProgramVersion ?programVersion
}
```

<table>
<thead>
<tr>
<th>host</th>
<th>programName</th>
<th>programVersion</th>
<th>CVE</th>
<th>CVEDesc</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;DESKTOP-MAC610T&quot;</td>
<td>&quot;notepad++.exe&quot;</td>
<td>&quot;7.3.3 (32-bit)&quot;</td>
<td>cve_ex:CVE-2017-8803</td>
<td>&quot;might allow user-assisted attackers to execute code via a crafted file&quot;</td>
</tr>
<tr>
<td>&quot;DESKTOP-MAC610T&quot;</td>
<td>&quot;notepad++.exe&quot;</td>
<td>&quot;6.6.9&quot;</td>
<td>cve_ex:CVE-2014-9456</td>
<td>&quot;Buffer overflow in NotePad++ 6.6.9 allows remote attackers to have unspecified impact via a long Time attribute in an Event element in an XML file&quot;</td>
</tr>
</tbody>
</table>
Conclusions

Semantic security logs create a foundation for tools that could:
- improve situational awareness
- facilitate better security decision-making
- create better understanding
- enable faster forensics
- make more effective incident response and containment possible

What we have accomplished so far:
- Architecture
- Domain-specific Vocabularies
- Log acquisition and knowledge extraction
- Alignment and entity reconciliation
- Linking to background knowledge
- Prototype
Current/Future Work

- Scalable storage and query federation
- Linked data and LDF query interfaces
- RDF Stream processing
- Event linking
- Event Archive
  - Event Stream Monitoring
  - SPARQL
  - TPF
  - APIs
  - Auditing
  - Forensics
  - Monitoring
  - Incidence Response
- Process mining
- Threat intelligence sharing