SEMI-AUTOMATED CYBER THREAT INTELLIGENCE (ACT)

Training – FIRST Conference 2019 Edinburgh

Martin Eian, Geir Skjøtskift, Siri Bromander and Tom Spangebu
<table>
<thead>
<tr>
<th>When</th>
<th>What</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00 – 10:30</td>
<td>Introduction to ACT</td>
</tr>
<tr>
<td>10:30 – 10:45</td>
<td>Break</td>
</tr>
<tr>
<td>10:45 – 13:00</td>
<td>Assignments, case study</td>
</tr>
<tr>
<td>13:00 – 14:00</td>
<td>Lunch (not provided)</td>
</tr>
<tr>
<td>14:00 – 15:30</td>
<td>Recap, breakout (API/Graph queries)</td>
</tr>
<tr>
<td>15:30 – 15:45</td>
<td>Break</td>
</tr>
<tr>
<td>15:45 – 18:00</td>
<td>Practical work (build something)</td>
</tr>
<tr>
<td>18:30 – 19:00</td>
<td>Newbie reception</td>
</tr>
<tr>
<td>19:00 – 21:00</td>
<td>Icebreaker reception</td>
</tr>
</tbody>
</table>
Goal

To collect and organize our knowledge of threats to make it useful
Data and Information

Data

Information
Semi-Automated...

- Analysis
- Enrichment
- Information Sharing
- Countermeasures
Semi-Automated Cyber Threat Intelligence (ACT)

The main objective of the research project is to develop a platform for cyber threat intelligence to uncover cyberattacks, cyber espionage and sabotage.

The project will result in new methods for data enrichment and data analysis to enable identification of threat agents, their motives, resources and attack methodologies.

In addition, the project will develop new methods, work processes and mechanisms for the generation and distribution of threat intelligence and countermeasures, to stop ongoing and prevent future attacks.
DATA MODEL
Data Model

• Objects
  - Global
  - Example: IP address

• Facts
  - Connected to one or two objects
  - Immutable
  - Timestamped
  - Owner
  - Role-based and explicit access control
  - Backed by evidence and comments
Models, Taxonomies and Vocabularies

• MITRE ATT&CK
  - https://attack.mitre.org

• MITRE PRE-ATT&CK
  - https://attack.mitre.org/pre-attack/

• STIX 2.0 vocabularies
  - https://oasis-open.github.io/cti-documentation/

• Ryan Stillions’ DML model
Current OSINT Sources

Import:

• APTNotes
  - https://github.com/aptnotes/data
• APT & CyberCriminal Campaign Collection
  - https://github.com/CyberMonitor/APT_CyberCriminal_Campaign_Collections
• RSS Feeds
  - Infosec blogs
• MISP (circl.lu feed)
• MITRE ATT&CK

Enrich:

• mnemonic PassiveDNS
  - https://passivedns.mnemonic.no/
• Shadowserver IP-BGP
• VirusTotal
• Multiple ways to represent the same information
• Different names for the same thing
  - Threat actors
  - Malware
Example: Campaign targets sector
Example: Campaign targets sector
Example: Campaign targets sector

- Incident is attributed to a campaign.
- The incident targets the organization.
- The organization is a member of the sector.
Example: Threat actor uses tool (ATT&CK)
Example: Threat actor uses tool (ATT&CK)
Example: Threat actor uses tool (ATT&CK)
Example: md5sum connects to ipv4
Example: md5sum connects to ipv4
Example: md5sum connects to ipv4
Different names

- APT28
- Sofacy
- Sednit
- Fancy Bear
Different names

- APT28 alias Sednit
- Sofacy alias Fancy Bear
Data Model

• Objects
  - Global
  - Example: IP address

• Facts
  - Connected to one or two objects
  - Immutable
  - Timestamped
  - Owner
  - Role-based and explicit access control
  - Backed by evidence and comments

• Placeholders
THE ACT PLATFORM
Platform Architecture – Core technologies

- Cassandra
- elasticsearch
- Apache Nifi
- Apache TinkerPop
Platform Architecture – Workflow orchestration

• Originally developed by NSA

• Open sourced and transferred to the Apache Foundation in 2014

• Manage flows of data supporting a large number of inputs and outputs:
  - HTTP, FTP, SCP, Kafka, Elasticsearch, JMS, Syslog, MongoDB, Hadoop, Cassandra, SMTP, POP3, etc
Platform Architecture – Graph database

• Looked into existing graph databases, but they lacked proper fine granular permissions (and many of them had commercial licenses that could not be used in the research project)

• Apache Tinkerpop implemented on top of Cassandra/Elasticsearch

• Graph queries opens up a range of possibilities that is not possible on a flat data structure
ACT Core Backend

REST API

GUI

TinkerPop

Cassandra

Elasticsearch
# API - Swagger

<table>
<thead>
<tr>
<th>Method</th>
<th>Endpoint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>/v1/fact</td>
<td>Create a new Fact.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/fact/{fact}/access</td>
<td>Retrieve a Fact's ACL.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/fact/{fact}/access/{subject}</td>
<td>Grant a Subject access to a Fact.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/fact/{fact}/comments</td>
<td>Retrieve a Fact's comments.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/fact/{fact}/comments</td>
<td>Add a comment to a Fact.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/fact/{fact}/retract</td>
<td>Retract an existing Fact.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/fact/{id}</td>
<td>Retrieve a Fact by its UUID.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/factType</td>
<td>Create a new FactType.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/factType</td>
<td>List available FactTypes.</td>
</tr>
<tr>
<td>PUT</td>
<td>/v1/factType/{id}</td>
<td>Update an existing FactType.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/factType/{id}</td>
<td>Retrieve a FactType by its UUID.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/object/{type}/{value}</td>
<td>Retrieve an Object by its type and value.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/{type}/{value}/facts</td>
<td>Retrieve Facts bound to a specific Object.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/{type}/{value}/traverse</td>
<td>Traverse the Object/Fact graph starting at an Object identified by its type and value.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/search</td>
<td>Search for Objects.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/traverse</td>
<td>Traverse the Object/Fact graph after performing an Object search.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/object/{id}</td>
<td>Retrieve an Object by its UUID.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/{id}/facts</td>
<td>Retrieve Facts bound to a specific Object.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/object/{id}/traverse</td>
<td>Traverse the Object/Fact graph starting at an Object identified by its UUID.</td>
</tr>
<tr>
<td>GET</td>
<td>/v1/objectType</td>
<td>List available ObjectTypes.</td>
</tr>
<tr>
<td>POST</td>
<td>/v1/objectType</td>
<td>Create a new ObjectType.</td>
</tr>
</tbody>
</table>
**API – Python library (act-api on pypi)**

**Project description**

**python-act**

python-act is a library used to connect to the ACT platform.

The platform has a REST api, and the goal of this library is to expose all functionality in the API.

**Objects and Facts**

The act platform is built on two basic types, the object and fact.

Objects are universal elements that can be referenced uniquely by its value. An example of an object can be an IP address.

Facts are assertions or observations that ties objects together. A fact may or may not have a value describing further the fact.

Facts can be linked on or more objects. Below, the seenIn fact is linked to both an ipv4 object and report object, but the hasTitle fact is only linked to a report.

<table>
<thead>
<tr>
<th>Object type</th>
<th>Object value</th>
<th>Fact type</th>
<th>Fact value</th>
<th>Object type</th>
<th>Object value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ipv4</td>
<td>127.0.0.1</td>
<td>seenIn</td>
<td>report</td>
<td>report</td>
<td>cbc80bb5c0c0f8944bf73(...)</td>
</tr>
<tr>
<td>report</td>
<td>cbc80bb5c0c0f8944bf73(...)</td>
<td>hasTitle</td>
<td>Threat Intel Summary</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>
## Splunk Add-on - Queries

![Splunk interface showing search results for APT29](image)

<table>
<thead>
<tr>
<th>fact_value</th>
<th>fact_type</th>
<th>dest_object_type</th>
<th>source_object_value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>usesTechnique</td>
<td>technique</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>threatActorAlias</td>
<td>threatActor</td>
<td>APT29</td>
</tr>
<tr>
<td>apt29:hammer: stealthy-tactics define a pdf</td>
<td>hasTitle</td>
<td>eaa8f5aeb5a5d2627ceeb9b5ad6784b91d6323f1859abc7fa24d4629ab054c</td>
<td></td>
</tr>
<tr>
<td></td>
<td>usesTool</td>
<td>tool</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>usesTechnique</td>
<td>technique</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>usesTool</td>
<td>tool</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>usesTechnique</td>
<td>technique</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>threatActorAlias</td>
<td>threatActor</td>
<td>APT29</td>
</tr>
<tr>
<td></td>
<td>usesTool</td>
<td>tool</td>
<td>APT29</td>
</tr>
</tbody>
</table>
### Splunk Add-on – Annotate search results

```plaintext
1 source="carbanak csv" dest_ip=179.43.140.82 | acta dest_ip
2 | table dest_ip usesC2* seenIn*
```

<table>
<thead>
<tr>
<th>dest_ip</th>
<th>usesC2:ipv4</th>
<th>seenIn.report</th>
</tr>
</thead>
<tbody>
<tr>
<td>179.43.140.82</td>
<td>c6ec175692ea26c4ee27974237e592ff 188f261e5fca94bd1fc1edc1aaf08ec0 6e9408c338e98a8bc166a8d4f8264019</td>
<td>9c624e51ffab866aaa73c41f944f7ec6045ec6c04a99e24b37eadd518b74780c 2d460cb52158909dad07e6b0f9491339ce4ce1550f64832b0c5396c2f5b08f</td>
</tr>
<tr>
<td>179.43.140.82</td>
<td>c6ec175692ea26c4ee27974237e592ff 188f261e5fca94bd1fc1edc1aaf08ec0 6e9408c338e98a8bc166a8d4f8264019</td>
<td>9c624e51ffab866aaa73c41f944f7ec6045ec6c04a99e24b37eadd518b74780c 2d460cb52158909dad07e6b0f9491339ce4ce1550f64832b0c5396c2f5b08f</td>
</tr>
<tr>
<td>179.43.140.82</td>
<td>c6ec175692ea26c4ee27974237e592ff 188f261e5fca94bd1fc1edc1aaf08ec0 6e9408c338e98a8bc166a8d4f8264019</td>
<td>9c624e51ffab866aaa73c41f944f7ec6045ec6c04a99e24b37eadd518b74780c 2d460cb52158909dad07e6b0f9491339ce4ce1550f64832b0c5396c2f5b08f</td>
</tr>
</tbody>
</table>

---

**mnemonic**
Threat Intelligence Platform - Summary

• Github repositories
  - https://github.com/mnemonic-no/act (installation docs)
  - https://github.com/mnemonic-no/act-api-python
  - https://github.com/mnemonic-no/act-bootstrap
  - https://github.com/mnemonic-no/act-frontend
  - https://github.com/mnemonic-no/act-platform
  - https://github.com/mnemonic-no/act-scio
  - https://github.com/mnemonic-no/act-splunk
  - https://github.com/mnemonic-no/act-triggers
  - https://github.com/mnemonic-no/act-workers
  - License: ISC (BSD compatible)

• Public AWS instance
  - https://act-eu1.mnemonic.no
TRAINING - INTRODUCTION
Before We Start

“Whatever can go wrong, will go wrong.”

A HISTORY OF MURPHY’S LAW

by Nick T. Spark

“MURPHY WAS NOT A MYTH. WHO KNEW?”

“CURIOUSLY MURPHYESQUE!”

www.historyofmurphyslaw.com

Periscope Film LLC [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0) or GFDL (http://www.gnu.org/copyleft/fdl.html)], via Wikimedia Commons
Accessing the read-only AWS instance

GUI:
https://act-eu1.mnemonic.no

Tasks:
https://act-eu1.mnemonic.no/examples/

API:
https://act-eu1.mnemonic.no/swagger/
Introduction 1
Introduction 1 – Click and Double-click
Introduction 1 – History, Layouts and Filtering
Introduction 1 – Fact Types
Introduction 1 – Graph Queries
Introduction 1 – Graph Queries
Try the following object queries and explore the graph:

- threatActor: APT3
- tactic: lateral-movement
- tool: foosace
- ipv4: 153.148.23.[.]118
Task 1

Try the following object query:

tool: remsec

Which threat actor is associated with this tool? Which techniques are associated with this threat actor? Can you find any reports that mention file hashes classified as remsec?
Task 2

Try the following object query:

ipv4: 188.116.32[.]164

Try to find reports, threat actors, tools and any other information related to this IP address.
Task 3

Explore Autonomous System Number 8048

- asn: 8048

What kind of malicious behaviour has been observed from this AS?

Where is the organization that owns AS8048 located?
Introduction 3 – Aliases
Introduction 3 – Aliases
Introduction 3 – Aliases
Introduction 3 – Aliases
Introduction 3 – Aliases


“The Webshell is named **TwoFace** as it is comprised by two components. The first is named TwoFace Loader, a basic and preliminary shell that extracts and installs the second component, a more advances tool named TwoFace Payload (identified by Microsoft as **Seasharpee**). These tools are written in #C, and run on Webservers that support ASP.NET.”
Introduction 3 – Aliases
Task 4

Try to find an alias for the tool ‘gulpix’. Then try to find a publically available, credible source that confirms your findings.
CASE STUDY
BREAKOUT: API, GRAPH QUERIES, EXPLORATION
Breakout: API/workers, graph queries, exploration

• API/workers - Geir
  - https://github.com/mnemonic-no/act-workshop-api
  - https://github.com/mnemonic-no/act-api-python
  - https://github.com/mnemonic-no/act-workers

• Graph queries - Martin

• Exploration
GRAPH QUERIES

With Great Power Comes Great Responsibility
Graph Query 2 – Show Edges
Graph Query 3 – 2 hops
Graph Query 4 – Filter Edges (Facts)
Graph Query 5 – Filter Nodes (Objects)
Graph Query 6 – Unique Tool Usage
Public Read-Only ACT Instance

https://act-eu1.mnemonnic.no/examples/
FURTHER WORK
New Information Sources

• Security events
• Incidents
• Reputation lists
• Malware analysis systems
• STIX feeds
• ...
Information Sharing

- Mechanism for sharing schema
- Format (STIX?)
- Trust models
Trust and Confidence

• Trust (source)
• Confidence (fact)
• Subjective Logic (quantify uncertainty)
GUI Improvements

- Timelines
- Share workspace
- Prune graph
GUI Improvements

Create Fact

Fact type
attributedTo

Source

Object Type
threatActor

Object Value
Axiom

Destination

Object Type
person

Object value
Ole Brumm

Access mode
Public

Comment
This is just a test