Looking for Similar: Helping Investigations

SANS Tactical Detection & Data Analytics Summit
Dec-2018

Gabriel Infante-Lopez| Principal Engineer
Map

1. process
2. two small examples

3. two more instances of the process

Finding similar endpoints

Finding similar investigation cases

4. a common tool

A 3-IN-1 SECURITY INCIDENT RESPONSE PLATFORM

Jaccard Similarity
The **what**, the **how**, and the **how-how**.

Analytics and the three levels of abstraction

**security realm**
- **what**
  - Business Problem
- **how**
  - ML/AI Problem
- **how-how**
  - Implementation Problem

**Example 1**
- is there abnormal behavior at this particular endpoint?
  - From a corpus of behavior, detect those entries that have low probability.
  - Depending on the distributions, use simple statistical methods or density-based anomaly detection

**Example 2**
- is this malware?
  - From a corpus of malware/clean software build a classifier.
  - Depending on the corpus size use either DNN, single class classification, or Bayes.

**ML realm**
A day in the life of a SOC analyst

When a system is reported as infected or is acting suspiciously:

- Do I have enough context to determine what to do next?
- What if the attacker can detect my response and change tactics?
- Is it possible that the system holds other malware that hasn’t been detected yet?
- How can I collect enough information from these systems (quickly enough) to determine the best containment and eradication strategy?
- Can they help me to profile the attacker’s techniques?
- How can I proactively search for Indicators of Compromise (IoC) across my endpoint?
- What other endpoints have a similar behavior?
- Did I already investigated a similar phenomenon?
Mining Endpoint Snapshots
Exploratory Analysis

what
In the context of an investigation, an analyst wants to find endpoints that have similar behavior.

how
Collect endpoint information and look for “similar” elements.

how-how
Use rastrea2r to collect endpoint snapshots, transform to set, use Jaccard similarity.
Mining Rastrea2r Endpoint Snapshots

Data Extraction
- Snapshot collection
- Json storage

Data Preprocessing
- Artifacts
- IOC
- Hashes

Mining

Jaccard Computation

Jaccard
Deploying Rastrea2r ON endpoints

1. Rastrea2r Deployment (SSCM, ePO, Ansible...)
2. Triage tools
3. DATA Write-only shared folder
4. YARA scan output

TOOLS
Read-only shared folder

Data Extraction

www.rastrea2r.io
Rastrea2r Output
Mining Rastrea2r Endpoint Snapshots

Data Extraction
- Snapshot collection
- Json storage

Data Preprocessing
- Artifacts
- IOC
- Hashes

Mining

Jaccard Computation

Data Extraction

Preprocessing

Jaccard
Elements in Rastrea2r

1. choose artifacts to use.
2. extract them from endpoints
3. store results on json files

Endpoint txt files → Preprocessing → Sets in json with selected artifacts

https://tomassetti.me/parsing-in-python/
https://realpython.com/python-json/

Process
- name
- path
- user
- time
- dlls

Services
- name
- owner
- time
- ports

DNS Query
- url
- frecuency
- reputation

Autostart entries
- command
- method
- path
- registryKey
Jaccard Similarity

Intuitive Definition

\[ Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \]

Source

SANS Tactical Detection & Data Analytics Summit
Computing Jaccard Similarity

List of Sets $\text{List}[(\text{Int, Set[String]})]$ → Jaccard $\rightarrow$ List of pairsof sets with similarity $\text{List}[(\text{Int, Int), Float})]$
Mining Rastrea2r Endpoint Snapshots

Data Extraction
- Rastrea2r collection
- Json storage

Data Preprocessing
- Artifacts
- IOC
- Hashes

Mining
- Jaccard Computation

Collection of json documents

Endpoints visiting similar websites.

URLs

Processes, DLLs

Hashes

Machines that have processes loading similar DLLs.

Endpoints having similar file hashes
The Hive

Tracking Investigations

A 3-IN-1 SECURITY INCIDENT RESPONSE PLATFORM

A scalable, open-source and free Security Incident Response Platform, tightly integrated with MISP (Malware Information Sharing Platform), designed to make life easier for SOCs, CSIRTs, CERTs and any information security practitioner dealing with security incidents that need to be investigated and acted upon swiftly.

Multiple SOC and CERT analysts can collaborate on investigations simultaneously. Thanks to the built-in live-stream, real-time information pertaining to new or existing cases, tasks, observables and IOCs is available to all team members. Special notifications allow them to handle or assign new tasks, preview new MISP events, SIEM alerts, email reports, import and investigate them right away.

Cases and associated tasks can be created using a simple yet powerful template engine. You may add metric and custom fields to your templates to drive your team's activity, identify the type of investigations that take significant time and seek to automate tedious tasks through dynamic dashboards. Analysts can record their progress, attach pieces of evidence or noteworthy files.

Add one, hundreds or thousands of observables to each case that you create or import them directly from a MISP event or any alert sent to the platform. Quickly triage and filter them. Harness the power of Cortex and its analyzers to gain precious insight and speed up your investigation. Leverage tags, flag IOCs, sightings and identify previously seen observables to feed your threat intelligence. Once investigations are completed, export IOCs to one or several MISP instances.

https://thehive-project.org
Mining The Hive Cases
Exploratory Analysis

what
In the context of TheHive investigation, an analyst needs similar cases that were already investigated.

how
Mine TheHive cases looking for similar ones

how-how
Extract data from Hive, transform to sets, and use jaccard to find similarities.
Mining Endpoint Snapshots

Data Extraction
- API calls
- Json storage

Data Preprocessing
- Artifacts
- NL Descriptions

Mining
- Jaccard Computation

Data Pushing
- API calls
Extracting Data from the Hive

1. choose artifacts to use.
2. restful api calls to TheHive
3. store results on json files

https://www.dataquest.io/blog/python-api-tutorial/
import nltk

for text in descriptions:
    sentences = nltk.sent_tokenize(text)
    for sentence in sentences:
        words = nltk.word_tokenize(sentence)
        tagged_words = nltk.pos_tag(words)
        ne_tagged_words = nltk.ne_chunk(tagged_words)
        print(ne_tagged_words)
Defining Sets

- **Case**
  - Title
  - Description
  - Tags

- **Observable**
  - MD5
  - Hash
  - Message
  - Tags

- **Task**
  - Description
  - Title
  - Tags

**Text**

- Tokenization
- Lemmatization
- Filtering
- Shingles

Standard NLP pipeline, most of them, dictionary based.

Investigation cases that have been assigned similar tags.
Defining Sets

- **Case**
  - Alerts
    - title
    - description
    - tags
- **Observable**
  - md5 hash
  - message
  - tags
- **Task**
  - description
  - title
  - tags

Text processing steps:
- tokenization
- lemmatization
- filtering

Standard NLP pipeline, most of them, dictionary based.

Investigation cases with similar descriptions given by the analysts.
Pushing results back to TheHive

1. Push back similarity relation to TheHive (use patch and case links)
2. Relations are shown in related cases.
3. Computation can be made off-line in batches.
4. All analysts see similarities.
Computing Jaccard on BigData
MinHash and LSH

- Webscale data
- Efficient algorithms
- Small memory footprint
- Approximate answers suffice

**Minhashing:** Convert large sets to short signatures, while preserving similarity

**Locality-sensitive hashing:** Focus on pairs of signatures likely to be from similar documents

Resources:
- Python implementation: [https://github.com/mattilyra/LSH](https://github.com/mattilyra/LSH)
Different Similarity Measures