Vertex

Synapse Bootcamp

Module 14

Modeling Data Manually

v0.4 - May 2024

Objectives

- Discuss use cases for creating nodes manually
- Understand how to map real-world data and information into Synapse's data model
- Gain familiarity with some forms that may require manual modeling
- Describe how you can use Synapse to represent strategic data

Modeling Data Manually

Modeling Manually

- Capture information that cannot be ingested automatically
- May include:
  - One-off indicators or information
  - Unstructured or badly-formatted data
  - Non-traditional indicators
- Often records:
  - Supporting evidence
  - Strategic data
- Capture more abstract or complex concepts
  - Classes or categories (`ou:industry`)
  - Objectives (`ou:goal`, `ou:campaign`)
  - Attacks or compromises (`risk:attack`, `risk:compromise`)

Common 'One-Off' Data

Form	Object	Purpose
`ou:org`	Company, government ministry, group…	Represent a real organized entity
`ps:contact`	A collection of data associated with a purported individual or organization	Capture identifying data for entity resolution
`ps:person`	A known person	Represent a real individual
`media:news`	Article, blog, whitepaper, report…	Source for other data (indicators, victims, real world events)

Common 'One-Off' Data

Form	Object	Purpose
`inet:service:account` `inet:web:acct`	User or group account (e.g., social media, forum user, etc.)	Account owned by threat actor Account used maliciously
`inet:service:message` `inet:web:post`	Message posted to a service	Social media post, forum post
`inet:service:message` `inet:web:mesg`	Private message between accounts	Social media phishing, threat actor communications...
`tel:txtmesg`	Mobile device message (SMS, iMessage...)	Smishing, threat actor communications...

`inet:service:*` forms were introduced in June 2024 and will replace `inet:web:*` forms.

Node Deconfliction

- Many nodes we create manually are **guid** nodes
  - Easily create an arbitrary guid using '*'
  - `[ ou:org=* :name='the vertex project' ]`
- Problem: how to avoid duplicate nodes?
  - Two users create two different **arbitrary** `ou:org` nodes for The Vertex Project
- Solution: check whether the node already exists
  - Use a "meaningful" secondary property
  - `ou:org:name='the vertex project'`
  - `ou:name='the vertex project' -> ou:org`
  - Use Lookup mode to search (e.g., 'vertex')

Manually checking for duplicates works...but is an imperfect solution.

Storm gen.* Commands

- Storm gen.* ("generate") commands can deconflict for you!
- Commands for many common guid forms
  - Orgs, threats, vulnerabilities...
- Takes property value(s) as input
- Checks for an existing node, based on the value
  - **Lifts** (returns) the existing node if found
  - **Creates** (generates) a new node (based on the provided value(s)) if not

```
gen.ou.org 'the vertex project'
```

Modeling Data Demo

Strategic Data

Strategic Data

- Synapse can represent a broad range of data
- Much of this data is **tactical**
  - Atomic indicators or objects
    - Files, URLs, organizations
  - Concrete observables or relationships
    - A file connected to an IP, a domain was registered through a specific company
- Strategic analysis commonly done "the long way"
  - Review and consolidate multiple reports
  - Write another report

Strategic Data

- Synapse allows you to:
  - Record more abstract concepts ('goals')
  - Represent concise data for events ('attacks', 'compromises')
  - Link events to concrete indicators
  - Link events to their objectives
- Structurally representing things like:
  - IOCs linked to...
  - ...individual attacks that are part of...
  - ...specific compromises meant to achieve...
  - ...particular objectives
- ...allows you to **answer questions** about strategic data as well

Many "strategic" forms are used for Synapse's **threat intel** data model.

Sample Strategic Forms

Events	
Form	**Purpose**
`risk:attack`	Specific / atomic malicious event
`risk:attacktype`	User-defined set of categories
`risk:compromise`	Broader event; comprised of one or more attacks

Objectives	
Form	**Purpose**
`ou:goal`	Objective (broad or tactical)
`ou:camptype`	User-defined set of categories
`ou:campaign`	Broad objective; comprised of one or more goals

These are not mutually exclusive - for example, attacks can have goals and be part of campaigns.

risk:attack

Category	Questions	Model Element

Who	Who/what was attacked (host, location)?
	Who carried out the attack?
	Who is reporting on the attack?
		`-(targets)>`
		`:attacker / #rep.mandiant.apt41`
		`:reporter:name / :reporter`

What	What kind of attack was this?
	Was the attack targeted?
	Was the attack successful?
	Was the attack part of a larger compromise? A broader campaign?
		`:type`
		`:targeted`
		`:success`
		`:compromise, :campaign`

When	When did the attack occur? When was it detected?	`:time` `:detected`
Why	What was the purpose of the attack?	`:goal`

How	What did the attacker use to carry out the attack?
	How sophisticated was the attack?
	How bad was any impact from the attack?
		`-(uses)>`
		`:sophistication`
		`:severity`

risk:compromise

Category	Questions	Properties

Who	Who/what was the target/victim?
	Who was the perpetrator?
	Who is reporting on the compromise?
		`:target`
		`:attacker / #rep.mandiant.apt41`
		`:reporter:name / :reporter`

What	What kind of compromise was this? Was the compromise part of a broader campaign?	`:type` `:campaign`
When	When did the compromise start? End? How long did the compromise last?	`:time / :lasttime` `:duration`
Why	What was the purpose of the compromise?	`:goal` `:goals`
How	How did the compromise impact the victim? How did the attacker carry out the compromise? How significant was the impact of the compromise?	`:loss:* / :ransom:* / :theft:* / :response:cost -(uses)>` `:severity`

Modeling Strategic Data - Demo

Summary

- There are many cases where you will need to **manually** model data
  - **One-off** objects or occurrences
  - Non-standard or less common **indicators**
  - Capturing supporting **evidence**
  - Representing more **strategic** or **abstract** information
- Modeling entails:
  - Identify the appropriate **form** to use
  - Map real-world data to the form **properties**