Vertex

Synapse Bootcamp

Module 20
Automation in Synapse

v0.4 - May 2024

Objectives

- Define automation in Synapse
- Identify how automation can accelerate common analyst workflows
- Describe Synapse automation components
- Understand cron job and trigger use cases
- Understand use cases for macros
- Create, manage, and inspect cron jobs and triggers

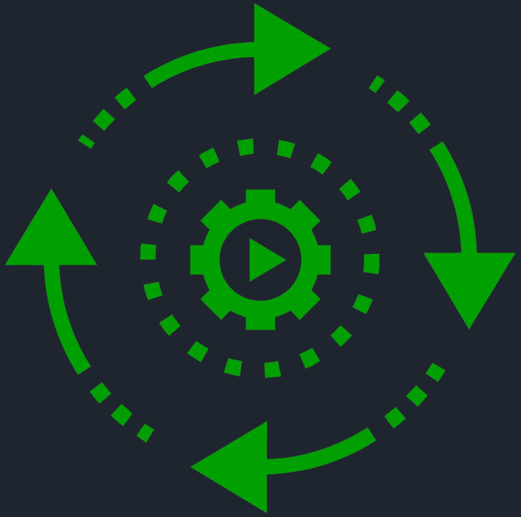
Simplifying Storm

- Synapse allows users to create, save, retrieve, and run Storm
  - Node Actions
  - Bookmarks
  - Queries (Storm Editor)
- Run commonly used Storm commands
  - Variations on default Node Actions
  - Combine commands often used together
- Store frequently used queries for easy access
  - "Hunt" style queries
  - That cool thing you wrote that you don't want to lose
  - "Daily tasking" queries
  - Gather data

What's **better** than being able to save and easily run Storm?

Synapse Automation

- Storm that is automatically invoked
  - By system events
  - At a scheduled time
  - On demand
- Executed with little or no human interaction
- Ideal for routine, pre-defined, and codifiable tasks
- Ensures tasks are executed regularly and consistently



Let Synapse run the Storm **for** you!

Example Use Cases

- Data collection:
  - Periodically retrieve data of interest (e.g., TOR, AlienVault, VirusTotal)
- Data enrichment:
  - Query available Power-Ups / data sources for IOCs of interest
- Threat hunting and detection:
  - Automate queries and tasks to search for new malware or threat activity
- Analysis:
  - Automate queries and tasks used to cluster malware families or threat groups
- Housekeeping:
  - Apply tags when specific conditions are met
  - Set tag definitions on newly created tags

Automation Components

Automation Components

- Three components used for automation
  - Cron jobs - time-based
  - Triggers - event-based
  - Macros - stored, callable Storm
- Components can be **combined** for power and flexibility
  - **Cron job** executes on a schedule, causes changes that…
  - …fire a **trigger** which…
  - …call a **macro** to perform a series of tasks…

Cron

- Time-based Storm execution
  - Frequency (hourly, weekly, twice a day…)
  - Once
- Ideal for:
  - Non-urgent tasks
  - Routine / periodic tasks
  - Housekeeping / maintenance

Cron Examples

Cron Job	Time Interval	Action
Set missing IPv4 data	Once	inet:ipv4:type=unicast -:asn | maxmind
Ingest AlienVault Pulses	Daily	alienvault.otx.pulses
Update MITRE ATT&CK data	Weekly	mitre.attack.sync
Attempt to download missing malware files	Daily at 18:00	hash:md5#rep hash:sha1#rep hash:sha256#rep -{ -> file:bytes +$lib.axon.has(:sha256) } | malshare.download
YARA retrohunt	Daily at 23:00	file:bytes -#cno | yara.match --rules ${ it:app:yara:rule.created@=(now,-24hours) }

Cron Demo

Triggers

- Event-driven Storm execution
  - Add / delete a node
  - Add / delete a tag
  - Add / delete an edge
  - Set a node property
- Ideal for:
  - Time-sensitive tasks
  - Encoding analysis logic

Trigger Examples

<table>
<tr><th>Trigger</th><th>Condition</th><th>Action</th></tr>
<tr><td>Populate IPv4 AS / geolocation data</td><td>cond = node:add
form = inet:ipv4</td><td>| maxmind</td></tr>
<tr><td>Enrich indicators</td><td>cond = tag:add
form = <any>
tag = cno.mal</td><td>| macro.exec enrich</td></tr>
<tr><td>Push tags from file to associated hashes</td><td>cond = tag:add
form = file:bytes
tag = cno.mal</td><td>| tee { :md5 -> hash:md5 }
{ :sha1 -> hash:sha1 } { :sha256 ->
hash:sha256 } | [+#cno.mal]</td></tr>
<tr><td>Tag sinkholed domains</td><td>cond = prop:set
prop =
inet:whois:email:email</td><td>+:email=domains@virustracker.info
-> inet:fqdn
[+#cno.infra.sink.holed.kleissner]</td></tr>
</table>

Trigger Demo

Macros

- Saved Storm queries that can be called:
  - On demand
  - By cron jobs
  - By triggers
- Ideal for:
  - Flexibility - call "from anywhere"
  - Longer queries
  - Queries shared across users / teams
    - Re-use
    - Consistency

Macro Examples

Example	Description
Enrich indicators	Use a single macro to specify which Power-Ups to call based on the type of indicator
"Hunt" queries	Perform a set of actions to search for potentially related malware or threat activity
Set tag definitions	Build and set tag (`syn:tag`) definitions (`:title`, `:doc`) for newly created tags
Run cron or trigger Storm	Queries executed by cron jobs or triggers can be stored in a macro, with the cron / trigger simply calling the macro

Macro Demo

Permissions and Scope

- You must have **permissions** to work with triggers and cron jobs
  - ...except in a forked view where you are **admin**
- **All users** are able to create macros
  - Author is **admin** of the macro
  - Other users can see and **run** the macro...but the macro runs as them
  - Admin can modify permissions to restrict (or grant) access

Element	Runs	Resides	Runs In	Need Permissions?	Runs As
Trigger	On event	View	View	Y	Author
Cron	On schedule	Cortex	View	Y	Author
Macro	On demand	Cortex	View	N	User who calls it

Summary

- Synapse supports **automation** for speed, efficiency, and consistency
- Automation uses **Storm**
  - Anything you can do in Storm you can automate
- **Triggers** are event-driven
  - Execute immediately - time-sensitive tasks
- **Cron** jobs run on a defined schedule
  - Non-urgent, repetitive, routine
- **Macros** allow you to compose and leverage longer queries
  - Call by trigger / cron job
  - Access via Node Action
  - Call from Storm query: `macro.exec <name_of_macro>`