



Vertex

Synapse Bootcamp

Module 8

Intro to Storm

v0.4 - May 2024



Objectives

- Understand the basic elements used in Storm queries
- Understand the key **operating concepts** behind Storm's behavior
- Know how to **lift** nodes using Storm
 - Using form / property names
 - Using form / property values
 - Using extended Storm operators



Why Storm?

- **Storm** is the query language used to interact with the data in Synapse
 - A "data language"
 - Designed to be concise, intuitive
- Using Storm can be as simple (or as powerful) as you like
- Storm allows you to "learn as you go"
 - Start simple
 - Learn additional syntax as needed

The Synapse UI simplifies many analysis tasks and often runs Storm "under the hood". However, Storm gives you the power and flexibility to tell Synapse **exactly** what you want. Storm can often get you an answer faster and more efficiently than the UI.



Storm Query Components

| Component | Example | Description |
|--------------------------------|---|--|
| Form | <code>inet:ipv4</code> | The objects to view / work with |
| Property (full or relative) | <code>inet:ipv4:asn</code> <code>:asn</code> | |
| Comparison Operator | <code>inet:ipv4 = 8.8.8.8</code> <code>file:bytes:size >= 102400</code> <code>inet:whois:rec:created @= (2023,2024)</code> <code>ou:org:name ^= ministry</code> | Used with values; how the value should be compared / evaluated |
| Value | <code>inet:ipv4 = 8.8.8.8</code> <code>ou:org:name ^= ministry</code> | Value used to select object(s) |
| Tag | <code>#cno.mal</code> <code>#cno.threat.t13</code> | Annotations placed on nodes |



Storm Query Components

| Component | Example | Description |
|--|---|--|
| Operation (lift, filter, pivot, traverse...) | <pre>inet:fqdn=vertex.link inet:ipv4#rep.eset +:asn=16276 inet:fqdn=woot.com -> inet:dns:a media:news -(refs)> hash:*</pre> | How Synapse should act on the data May include a symbol representing the operation to perform |
| Command | <pre>file:bytes max :mime:pe:compiled risk:threat limit 10 inet:ipv4=123.120.102.48 maxmind</pre> | Action or function to perform on the data |



Storm Operations

| Operation | Meaning | Common Storm Operator | UI Equivalent |
|---------------|--|-----------------------|---|
| Lift | Select data (nodes) from Synapse | Query bar - Storm | Query bar - Lookup / Text Search query and copy menu options |
| Pivot | Move between nodes that share the same property value | -> or <- * | Explore button pivot menu option |
| Traverse | Move between nodes that are linked by an edge | -(*)> or <(*)- | Explore button |
| Filter | Include / exclude a subset of nodes | + or - | n/a (column filters; query / select menu options) |
| Run | Execute a Storm command | <command> | Node Action |
| Modify / Edit | Modify or delete properties Add or remove tags Add nodes | [] or [()] | Inline property edit; delete menu option Add / remove tags menu options Lookup or Auto Add / Add Node |



Storm and Operation Chaining

- Operations are "building blocks" that you **chain** together

```
inet:fqdn=vertex.link -> inet:dns:a -> inet:ipv4 -#cno.infra.dns.sink | maxmind
```

Lift

Pivot

Pivot

Filter

Command



Lifting Data with Storm



Lifts in Storm

- **Select** one or more objects (nodes) from the Synapse data store
 - No "show me all the data" button
- Most lifts in Storm use one of the following:
 - **Form**
 - **Form** and **property**
 - **Form / property** and **value**
 - **Tag**



Lift by Property Value

- Equals (=) is the simplest comparison operator
 - o Specific node
 - o All nodes with a specific value

| Kind of Lift | Example | Question |
|-----------------------------|---|--|
| By primary property value | <code>inet:fqdn = vertex.link</code> | Show me the FQDN 'vertex.link' |
| By secondary property value | <code>inet:ipv4:asn = 9009</code> | Show me all the IPv4s on AS 9009 |
| By secondary property value | <code>ou:org:name = 'the vertex project'</code> | Show me the organization(s) named 'the vertex project' |



Lift by Tag

- Nodes that have a particular tag
 - o All nodes
 - o Specific nodes

| Kind of Lift | Example | Question |
|--------------|---|--|
| By tag | <code>#rep.mandiant.ap1</code> | Show me everything Mandiant associates with APT1 |
| Form by tag | <code>inet:fqdn#rep.mandiant.ap1</code> | Show me the FQDNs Mandiant associates with APT1 |



Lift by Form or Property

- All nodes of a particular kind
- All nodes that have a particular property

| Kind of Lift | Example | Question |
|----------------------------|--|---|
| By form | <code>inet:ipv4</code> | Show me all the IPv4s |
| By property (secondary) | <code>inet:ipv4:asn</code> | Show me the IPv4s that have an Autonomous System number |
| By property (extended) | <code>file:bytes:_virustotal:reputation</code> | Show me the files that have a VirusTotal reputation score |
| By property (universal) | <code>.created</code> | Show me all the nodes in Synapse |
| Form by universal property | <code>inet:dns:a.seen</code> | Show me all the DNS A records that have a <code>.seen</code> property |



Lift Demo - Basic Lifts



Additional Comparison Operators



Additional Comparison Operators

| Symbol(s) | Type of Operator | Example |
|--|-------------------------|--|
| <code>=, >, <, >=, <=</code> | Standard / Mathematical | <code>file:bytes:mime:pe:compiled >= 20130101</code> |
| <code>^=</code> | By prefix | <code>ou:org:name ^= ministry</code> |
| <code>~=</code> | By regular expression | <code>ou:org:alias ~= edu</code> |
| <code>@=</code> | By time / interval | <code>inet:dns:a.seen @= (20220121, now)</code> |
| <code>*[<operator>]</code> | Arrays | <code>crypto:x509:cert:identities:fqdns*[~=gov]</code> |



Lift Demo - Comparison Operators



Specialized Lifts



Specialized Lifts

- Lift forms by **wildcard**
 - Use the wildcard (*) to partially match form names
- Lift by **interface**
 - Specialized data model element
 - Defines set of properties that are copied to other forms
 - Lift all forms that use the interface
 - Lift all forms with a shared interface property / property value



Lift Demo - Specialized Lifts



Synapse UI and Storm

| Synapse UI (Lookup / Text Search Mode) | Storm (Storm Mode) |
|--|--|
| Lift common IOCs Search strings and try to find what you need | Directly lift any object in the data store |
| Use a subset of simple lifts | Lift using additional operators (mathematical, extended) |
| Explore to find desired data | Lift exactly what you need |



Summary

- **Storm** is Synapse's native query language
- Storm queries use:
 - **Data model** objects (forms, nodes, properties, tags)
 - **Operations** (lift, pivot, filter...)
- Operations are **chained** together to build longer queries
 - Storm functions as a **pipeline** from left to right
- A **lift** is the most basic operation in Storm
 - **Simple** lifts match specific forms, property values, or tags
 - Can also lift using **mathematical** operators (>, <=, etc.)
 - Storm-specific **extended** operators for custom lifts (by prefix, by time interval, etc.)