Synapse Bootcamp

Module 9

Pivoting and Traversal in Storm

v0.4 - May 2024

Objectives

- Understand what it means to pivot in Synapse
- Use Storm to pivot through data
- Leverage specialized pivots such as wildcard pivots and joins
- Understand what it means to traverse ("walk") edges in Synapse
- Use Storm to traverse through data
- Leverage combined "pivot and walk" operations
- Understand how pivoting and traversal relate to Explore in the UI

Pivoting

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

Pivoting

- Primary way to navigate data in Synapse
- Uses intuitive "arrow" symbol (**->**)
- Move between nodes that share a **property value**
  - If properties are also the same **type**...
  - ...Synapse **automatically** identifies these relationships
- A pivot represents an **implicit** connection / relationship between nodes
  - Don't need to create these connections - they just exist

Pivoting - Type Awareness

- Nodes with **properties** with the same **type** have an **implicit relationship**

Form / Property	Type
inet:fqdn	inet:fqdn
:domain	inet:fqdn
:host	str
:issuffix	bool
:iszone	bool
:zone	inet:fqdn

Form / Property	Type
inet:dns:a	inet:dns:a
:fqdn	inet:fqdn
:ipv4	inet:ipv4

Form / Property	Type
inet:ipv4	inet:ipv4
:asn	inet:asn
:dns:rev	inet:fqdn
:latlong	geo:latlong
:loc	loc
:place	geo:place
:type	str

Pivoting - Type Awareness

- Nodes with **properties** with the same **type** and **value** are **connected**

Form / Property	Value
inet:fqdn	work.viewdns.ml
:domain	viewdns.ml
:host	work
:issuffix	False
:iszone	False
:zone	viewdns.ml

Form / Property	Value
inet:dns:a	inet:dns:a
:fqdn	work.viewdns.ml
:ipv4	195.20.50.249

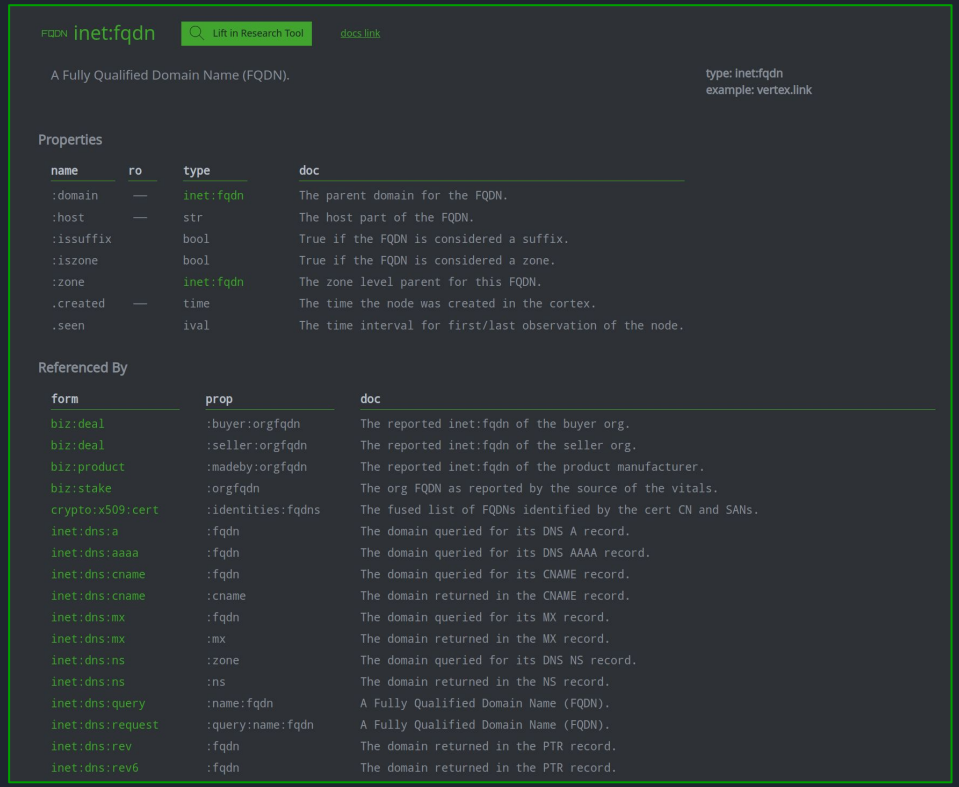
Form / Property	Value
inet:ipv4	195.20.50.249
:asn	31624
:latlong	52.3824,4.8995
:loc	nl
:type	unicast

Protip: Type awareness is how "Explore" in Synapse (or pivoting in Storm) works!

Data Model - Type Awareness

- "Properties"and "Referenced By" show all type-based interconnections
- Use to see:
  - What is this form connected to?
  - How is it connected?
  - How can I navigate between forms?

Tip: you can also use the Explore button to see "what's connected".



FQDN inet:fqdn — Lift in Research Tool — docs link

A Fully Qualified Domain Name (FQDN).

type: inet:fqdn
example: vertex.link

Properties

name	ro	type	doc
:domain	—	inet:fqdn	The parent domain for the FQDN.
:host	—	str	The host part of the FQDN.
:issuffix		bool	True if the FQDN is considered a suffix.
:iszone		bool	True if the FQDN is considered a zone.
:zone		inet:fqdn	The zone level parent for this FQDN.
.created	—	time	The time the node was created in the cortex.
.seen		ival	The time interval for first/last observation of the node.

Referenced By

form	prop	doc
biz:deal	:buyer:orgfqdn	The reported inet:fqdn of the buyer org.
biz:deal	:seller:orgfqdn	The reported inet:fqdn of the seller org.
biz:product	:madeby:orgfqdn	The reported inet:fqdn of the product manufacturer.
biz:stake	:orgfqdn	The org FQDN as reported by the source of the vitals.
crypto:x509:cert	:identities:fqdns	The fused list of FQDNs identified by the cert CN and SANs.
inet:dns:a	:fqdn	The domain queried for its DNS A record.
inet:dns:aaaa	:fqdn	The domain queried for its DNS AAAA record.
inet:dns:cname	:fqdn	The domain queried for its CNAME record.
inet:dns:cname	:cname	The domain returned in the CNAME record.
inet:dns:mx	:fqdn	The domain queried for its MX record.
inet:dns:mx	:mx	The domain returned in the MX record.
inet:dns:ns	:zone	The domain queried for its DNS NS record.
inet:dns:ns	:ns	The domain returned in the NS record.
inet:dns:query	:name:fqdn	A Fully Qualified Domain Name (FQDN).
inet:dns:request	:query:name:fqdn	A Fully Qualified Domain Name (FQDN).
inet:dns:rev	:fqdn	The domain returned in the PTR record.
inet:dns:rev6	:fqdn	The domain returned in the PTR record.

Pivoting in Storm

- A pivot requires:
  - The **source** (nodes or properties you're coming from)
  - A pivot **operator** (**->** is the most common)
  - The **target** of the pivot (nodes or properties you're pivoting to)
- Pivoting navigates **away** from the source **to** the target

```
inet:fqdn=vertex.link -> inet:dns:a:fqdn
```

Source:
- the FQDN 'vertex.link'

Target:
- `inet:dns:a` nodes with matching `:fqdn` property

Source and Target

```
inet:dns:a:fqdn=vertex.link :ipv4 -> inet:ipv4
```

- **Source:** `:ipv4` property (**relative** property name) of `inet:dns:a` nodes
- **Target:** any matching `inet:ipv4` nodes

```
inet:fqdn=hugesoft.org :zone -> inet:fqdn:zone
```

- **Source:** `:zone` property of FQDN `hugesoft.org`
- **Target:** any `inet:fqdn` with the same `:zone` property value

```
file:bytes#rep.mandiant.apt1 -> *
```

- **Source:** `file:bytes` nodes tagged `#rep.mandiant.apt1`
- **Target:** wildcard - **any** node matching **any** property of **any** source `file:bytes`

Explicit vs. Implicit Syntax

- **Explicit:** specifies the source and target forms and **properties**
  - Tell Synapse the **exact** navigation you want to perform

```
inet:fqdn=vertex.link -> inet:dns:request:query:name:fqdn :exe -> file:bytes
```

- **Implicit:** specify source and target **forms only**
  - Synapse figures out which properties you mean (type awareness)

```
inet:fqdn=vertex.link -> inet:dns:request -> file:bytes
```

- When using a **wildcard** explicit vs. implicit does not apply

```
file:bytes#rep.mandiant.apt1 -> *
```

Pivot Examples

Kind of Pivot	Example	Question
Wildcard pivot out	`inet:ipv4=1.2.3.4 -> *`	Show me everything **this node points to**
Wildcard pivot in	`inet:ipv4=1.2.3.4 <- *`	Show me everything **that points to this node**
Pivot out	`inet:fqdn=woot.com -> inet:dns:a:fqdn`	Show me the DNS A records for this FQDN
Pivot out	`inet:fqdn=woot.com -> inet:dns:a*`	Show me the DNS A and AAAA records for this FQDN
Pivot out	`inet:fqdn=woot.com -> inet:fqdn:zone`	Show me the FQDN records where this FQDN is the zone
Pivot out	`inet:fqdn=woot.com -> (inet:dns:a, inet:dns:ns)`	Show me the DNS A and NS records for this FQDN
Pivot out	`file:bytes:md5=6de25e21cfda939dda1a41a326f5de10 -> it:host:activity`	Show me all the execution activity associated with this file

Pivoting - Demo

Tag Pivots

Tag Pivots

Kind of Pivot	Example	Question
Pivot from tags (from `syn:tag` nodes to tagged nodes)	`syn:tag=cno.mal -> *`	Show me the nodes that have this tag
Pivot to tags (from tagged nodes to `syn:tag` nodes)	`inet:ipv4=1.2.3.4 -> #`	Show me the **leaf** syn:tag nodes for the tags on this node
Pivot to tags (from tagged nodes to `syn:tag` nodes)	`inet:ipv4=1.2.3.4 -> #*`	Show me **all** the syn:tag nodes for the tags on this node

Tip: Because pivoting to/from `syn:tag` nodes uses special behavior by default, you must use explicit syntax when pivoting to/from tag **properties.**

Tag Pivots - Demo

Traversal

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

Edge Traversal

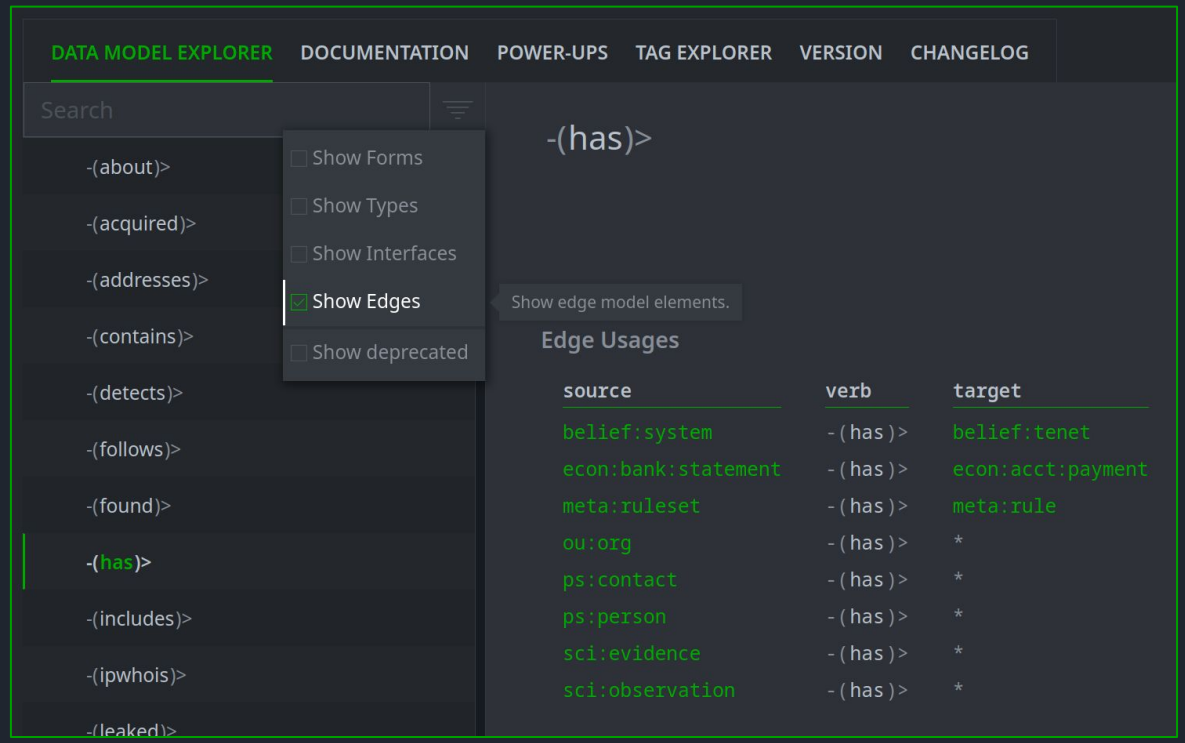
- Lightweight (light) edges connect nodes that do not have properties in common
  - Often a generic relationship ("references", "seen by")
- Traversal navigates ("walks") between nodes joined by a light edge
- A light edge is an **explicit** connection / relationship between nodes
  - Must explicitly create or remove edges between nodes

Traversal in Storm

- Edge traversal in Storm requires:
  - The **source** (nodes you're coming from)
  - The traversal **"arrow"** operator: -( )> or <( )-
    - Correct "direction" for the edge relationship
    - Edge name/names (or wildcard *)
  - The **target** (nodes you're traversing to)
- Traversal navigates **from** the source **to** the target

Data Model - Edges

- View in Data Model Explorer
- Recommended edges / edge use



Source and Target

```
media:news:publisher:name=eset -(refs)> *
```

- **Source:** All `media:news` nodes whose `:publisher:name` property is 'eset'
- **Edge:** `refs` ("references"); direction is "forward"
- **Target:** wildcard - **any** nodes linked to the article(s) by a `refs` edge

```
media:news:publisher:name=eset -(refs)> inet:fqdn
```

- **Source:** All `media:news` nodes whose `:publisher:name` property is 'eset'
- **Edge:** `refs` ("references"); direction is "forward"
- **Target:** any `inet:fqdn` nodes linked to the article(s) by a `refs` edge

```
inet:fqdn=music.todayusa.org <(*)- meta:source
```

- **Source:** the FQDN `music.todayusa.org`
- **Edge:** wildcard - **any** edge; direction is "backward"
- **Target:** any data source (`meta:source` node) linked to the FQDN by **any** edge

Common Edges - refs

- `refs` ("references") use cases:
  - Link articles to things "referenced" by an article:

```
media:news=<guid> -(refs)> *
```

  - Link a node with a text property to things scraped from the text:

```
inet:service:message=<guid> -(refs)> *
```

  - Find all the things that "reference" an MD5 hash:

```
hash:md5=<hash> <(refs)- *
```

```
hash:md5=<hash> <(refs)- media:news
```

Common Edges - seen

- seen use cases:
  - Link a data source to all nodes observed by the source:

```
meta:source=<guid> -(seen)> *
```

  - Find all the sources that "saw" an indicator

```
inet:dns:a=(woot.com,1.2.3.4) <(seen)- *
```

```
inet:dns:a=(woot.com,1.2.3.4) <(seen)- meta:source
```

Protip: Light edges are documented in **Data Model Explorer.** Light edges created by Power-Ups are documented in the Power-Up Help.

Traversal Examples

Example	Question
`inet:fqdn=vertex.link <(*)- *`	Show me every node connected to this FQDN by any edge
`inet:fqdn=vertex.link <(refs)- *`	Show me every node that references this FQDN
`inet:fqdn=vertex.link <(refs)- media:news`	Show me every article that references this FQDN
`inet:fqdn=vertex.link <( (refs, seen) )- *`	Show me every node that references or has seen this FQDN
`media:news=<guid> -(refs)> (inet:fqdn, inet:ipv4)`	Show me the FQDNs and IPv4s referenced by this article
`media:news=<guid> -(refs)> hash:*`	Show me all the hashes referenced by this article

Traversal - Demo

Additional Operations

Pivot and Traverse

- Pivots and traversals navigate different kinds of connections
  - Property-based vs edge-based
- Use a "double arrow" to perform both at once
  - Must use wildcard as a target

```
inet:ipv4=1.2.3.4 --> *
```

```
inet:fqdn=vertex.link <-- *
```

Tip: Using the wildcard as a target is an easy way to use Storm to explore / see what is connected to your source node(s).

Pivot / Traverse and Explore

Operation	Meaning	Common Storm Operator	UI Equivalent
Pivot	Move between nodes that share the same **property value**	-> or <- *	**Explore button**
Traverse	Move between nodes that are linked by an **edge**	-(*)> or <(*)-	**Explore button**

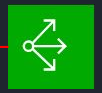
Pivot / Traverse and Explore

Operation	Meaning	Common Storm Operator	UI Equivalent
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**

- The **Explore** button does all of the above at once:
  - Wildcard pivot out (-> *)
  - Wildcard traverse (walk) out (-(*)>)
  - Wildcard pivot in (<- *)
  - Wildcard traverse (walk) in (<(*)-)

Pivot out and walk (--> *)

Pivot in and walk (<-- *)



Explore button

Join Operations

- Pivot and traversal both navigate away from the source to the target
  - The nodes you see (your "working set") changes
- Sometimes you want to see both
- Pivot / traversal operators can use an embedded plus sign (`+`) to keep the source nodes

```
inet:dns:a:fqdn=vertex.link -+> ( inet:fqdn, inet:ipv4 )
```

```
inet:ipv4=1.2.3.4 <+(seen)- meta:source
```

```
file:bytes=<sha256> <+-- *
```

Synapse UI and Storm

Synapse UI (Explore)	Storm Pivot / Traverse
Good for Exploring - "I don't know what's connected"	Good for targeted navigation - "I know exactly where I want to go"
No typing!	Typing...but ways to simplify
Supports standard pivots - between primary / secondary properties	Supports **all** pivot formats using explicit syntax
May need to navigate / display large numbers of nodes to get to where you want to go	Only navigate / display exactly what you need

Summary

- In Synapse, **pivoting** navigates between nodes that share a **property value**
  - Storm uses the "arrow" symbol for pivots (`->`)
- **Explicit syntax** tells Synapse **exactly** how to pivot
  - Source / target **forms** and **properties**
- Many pivots can use **implicit syntax**
  - Source / target **forms** only
- **Traversal** navigates between nodes connected by a **light edge**
  - Storm uses an arrow with an edge name (or wildcard) for traversal (`-(*)>`)
- The Synapse **Explore** button uses pivots and traversals to automatically navigate for you