

Synapse Bootcamp - Module 1

Introduction and Overview - Answer Key

Introduction and Overview - Answer Key	1
Answer Key	2
Your Synapse Environment	2
Exercise 1 Answer	2
Help Tool - Data Model Explorer / Tag Explorer	2
Exercise 2 Answer	2
Exercise 3 Answer	4
Workspaces Tool	8
Exercise 4 Answer	8
Research Tool	9
Exercise 5 Answer	9
Exercise 6 Answer	11
Part 1 - Use the Details Panel to view nodes	11
Console Help	14
Exercise 7 Answer	14

Answer Key

Your Synapse Environment

Exercise 1 Answer

Objective:

- **Set the Workspace and View to use for Synapse Bootcamp.**

This exercise ensures Synapse is set up correctly for Synapse Bootcamp. Your **Top Bar** should look like this:



Help Tool - Data Model Explorer / Tag Explorer

Exercise 2 Answer

Objective:

- **Use Data Model Explorer to search, view, and lift sample forms.**

Question 1: What information can Synapse record about an email address?

- An **inet:email** form in Synapse can record:
 - The **domain** (fully qualified domain name, or FQDN) from the email address (**:fqdn** property).
 - The **username** from the email address (**:user** property).
 - **When** the email address was **added to Synapse** (**.created** property).
 - An **optional** date/time range when the email address was **observed** (**.seen** property).

Properties

<u>name</u>	<u>ro</u>	<u>type</u>	<u>doc</u>
:fqdn	—	inet:fqdn	The domain of the email address.
:user	—	inet:user	The username of the email address.
.created	—	time	The time the node was created in the cortex.
.seen		ival	The time interval for first/last observation of the node.

Question 2: How many email address properties are associated with an `inet:email:message` object?

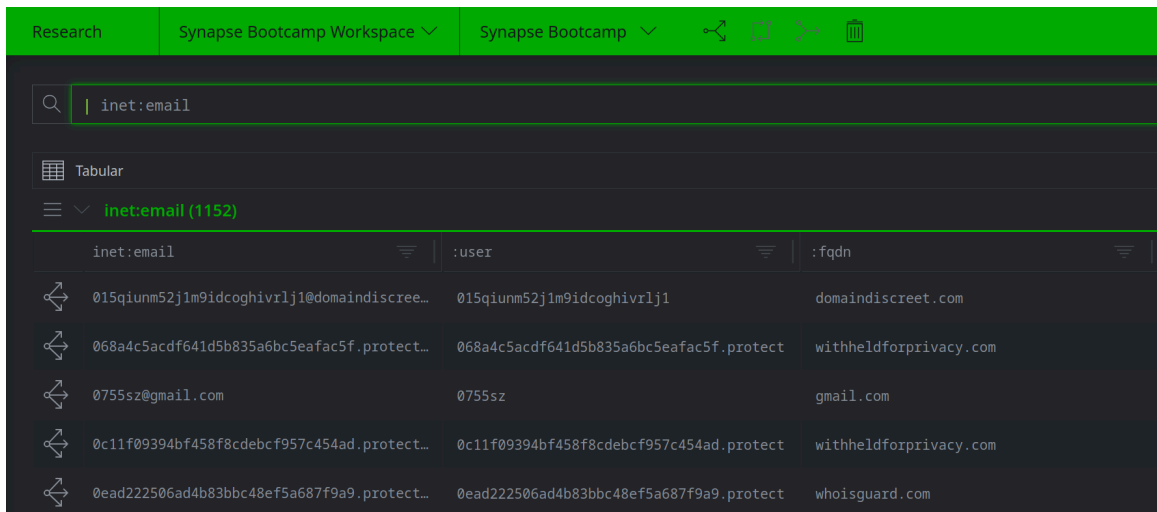
- Based on the **Referenced By** information, an `inet:email:message` node has **four** properties that can contain email addresses:

```
inet:email:message :to           The email address of the recipient.
inet:email:message :from        The email address of the sender.
inet:email:message :replyto     The email address parsed from the "reply-to" header.
inet:email:message :cc          Email addresses parsed from the "cc" header.
```

- **:to** is the recipient address (from the "to" header)
- **:from** is the sender address (from the "from" header)
- **:replyto** is the address where replies are sent (from the "reply-to" header)
- **:cc** is any additional recipients (from the "cc" header)

Question 3: What happens when you click the **Lift in Research Tool** button?

- Synapse takes you to the **Research Tool (Tabular display mode)** and runs a Storm query to select (**lift**) all of the email addresses (**inet:email** nodes) in Synapse:

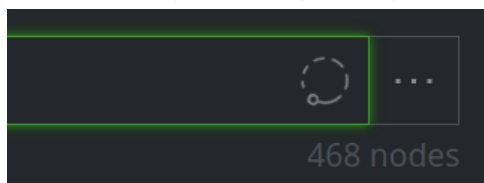


Lift in Research Tool is an easy way to see real examples (**nodes**) of a particular object (**form**) in Synapse.

Synapse will load:

- As many nodes as it can, up to the **Load increment** specified for **Tabular** display mode (as configured in the **Workspaces** tool), or
- **All** of the nodes (if the total number in Synapse is less than the Load increment).

You can stop the query by clicking the query status icon at the far right of the **Storm Query Bar**. A spinning circle indicates that a query is currently running:



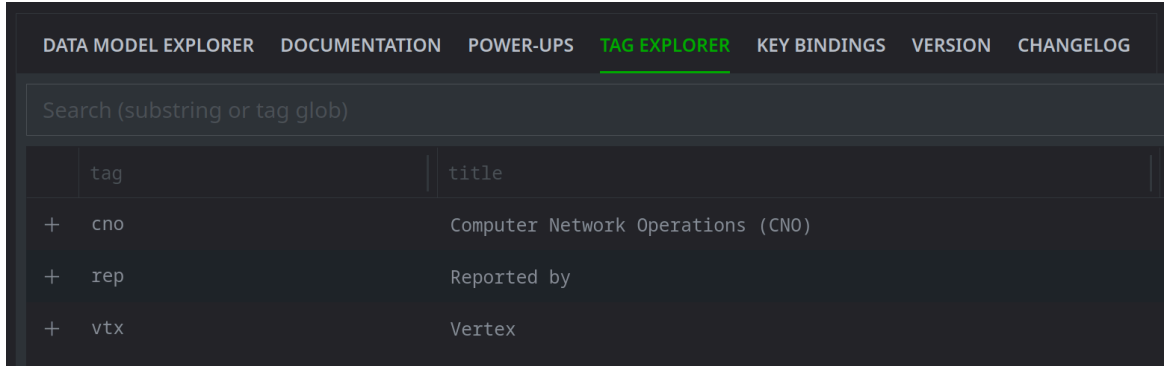
Exercise 3 Answer

Objective:

- Use Tag Explorer to:
 - view and explore tags,
 - find or set tag definitions, and
 - lift tags and / or tagged nodes.

Question 1: How many top-level tags have been created in your instance of Synapse?

- There are **three** top-level tags: **cno**, and **rep**, and **vtx**:



The screenshot shows the Synapse TAG EXPLORER interface. At the top, there are navigation tabs: DATA MODEL EXPLORER, DOCUMENTATION, POWER-UPS, TAG EXPLORER (highlighted), KEY BINDINGS, VERSION, and CHANGELOG. Below the tabs is a search bar with the placeholder text "Search (substring or tag glob)". Below the search bar is a table with two columns: "tag" and "title". The table contains three rows of data:

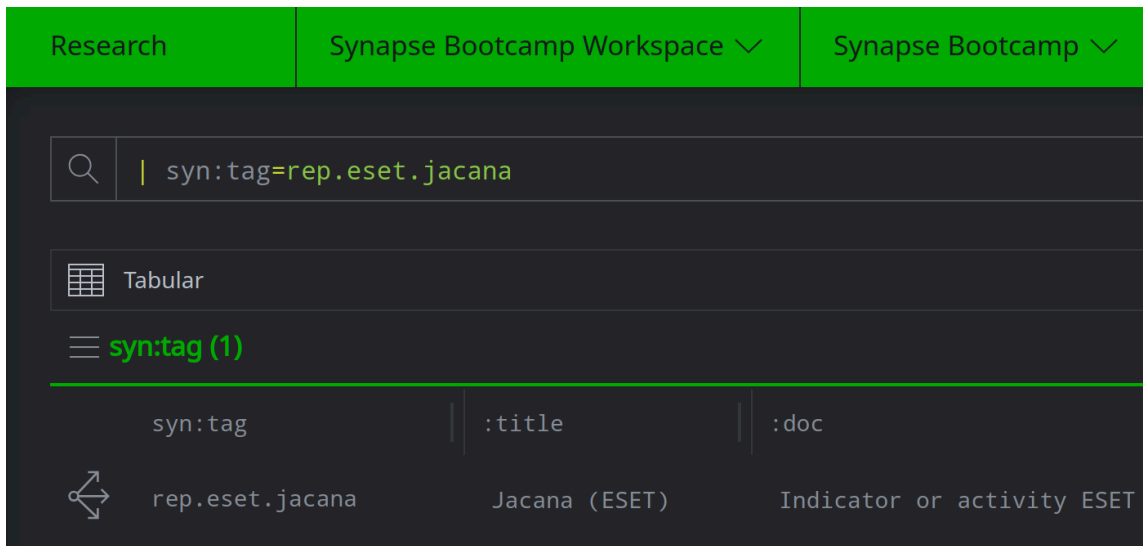
tag	title
+ cno	Computer Network Operations (CNO)
+ rep	Reported by
+ vtx	Vertex

Question 2: What do these tags represent, based on their definitions?

- The tags represent:
 - **cno**: tags related to computer network operations.
 - **rep**: tags for information reported by third-party organizations.
 - **vtx**: tags internal to The Vertex Project.

Question 3: What nodes (objects) are displayed when you select **research query > selected node** ?

- Synapse takes you to the **Research Tool (Tabular** display mode) and displays the **node** representing the tag (the **syn:tag** node):



The screenshot shows the Synapse interface with a search bar containing the query `syn:tag=rep.eset.jacana`. Below the search bar, the view is set to 'Tabular'. The results are displayed in a table with the following columns and data:

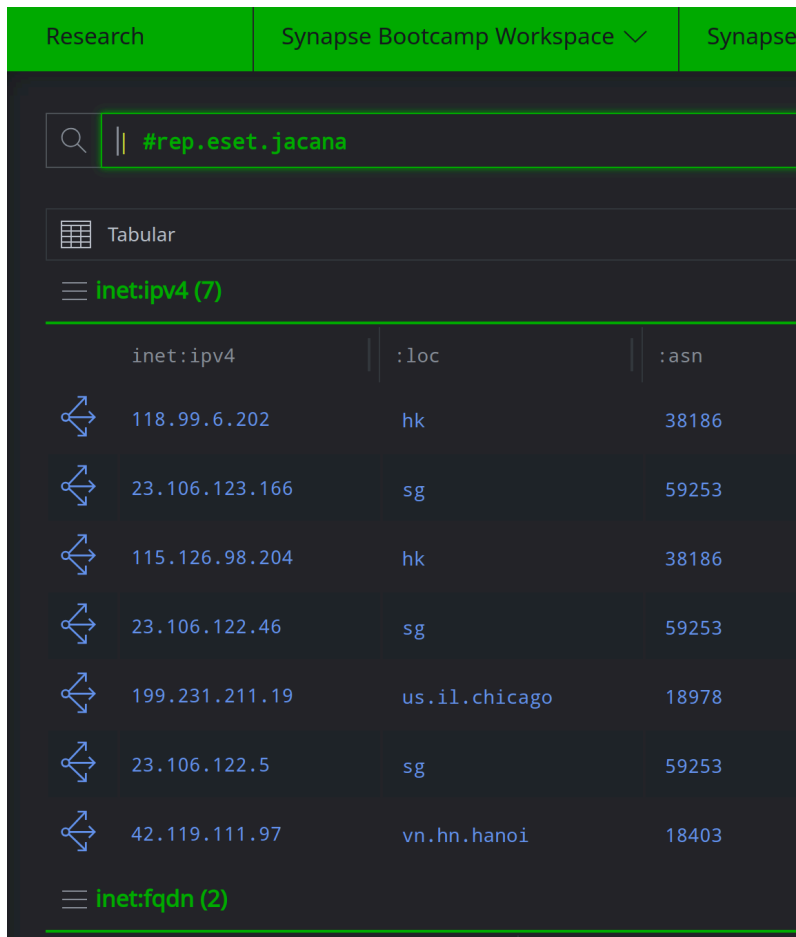
syn:tag	:title	:doc
rep.eset.jacana	Jacana (ESET)	Indicator or activity ESET

Note that Synapse loads and runs the following Storm query to show you the tag:

```
| syn:tag=rep.eset.jacana
```

Question 4: What nodes (objects) are displayed when you select **research query > selected tag** ?

- Synapse takes you to the **Research Tool (Tabular display mode)** and shows the **nodes that have the tag applied**:



The screenshot shows the Synapse interface with a search bar containing the query `#rep.eset.jacana`. Below the search bar, the display mode is set to 'Tabular'. A section titled 'inet:ipv4 (7)' contains a table of nodes. Each row includes a link icon, an IPv4 address, a location code, and an ASN number.

inet:ipv4	:loc	:asn
118.99.6.202	hk	38186
23.106.123.166	sg	59253
115.126.98.204	hk	38186
23.106.122.46	sg	59253
199.231.211.19	us.il.chicago	18978
23.106.122.5	sg	59253
42.119.111.97	vn.hn.hanoi	18403

Below the table, there is a section titled 'inet:fqdn (2)'.

Note that Synapse loads and runs the following Storm query to show you the tagged nodes:

```
| #rep.eset.jacana
```

Tip: The hashtag symbol (#) is used in Synapse's Storm query language to represent a tag **applied** to a node (as opposed to a `syn:tag` form).















Workspaces Tool

Exercise 4 Answer

Objective:

- **Customize your Synapse UI using the Workspaces tool.**

- Your full set of tag color rules should look like this:

TAG COLORS	NODE ACTIONS
	#cno.threat.*
	#cno.code.*
	#cno.mal.*
	#cno.mal
	#rep.mandiant.*
	#rep.symantec.*
	#rep.electiciq.*
	#rep.alienvault.*
	#rep.shodan.*
	#rep.vt.*
	#rep.*
	#cno.ttp.*
	#cno.infra.*
	#cno.common

Research Tool

Exercise 5 Answer

Objectives:


- Understand how to customize the layout and appearance of Tabular display in the Research tool.
- Know how to add, remove, and reset columns using:
 - standard controls from the Details Panel (Node tab), and
 - column / form menus.

Question 1: What **columns** are displayed in the **Results Panel** for the DNS A records?

- Synapse displays the domain (**:fqdn**) and IPv4 address (**:ipv4**) from the DNS A record:



The screenshot shows the Synapse interface with a search query `inet:dns:a | limit 10`. The results are displayed in a tabular view with columns `:fqdn` and `:ipv4`. The table contains three rows of DNS A records.

	<code>:fqdn</code>	<code>:ipv4</code>
	<code>mail.usnewssite.com</code>	<code>69.195.129.72</code>
	<code>dod.dnsweb.org</code>	<code>184.168.221.96</code>
	<code>ttl.tfxdccssl.net</code>	<code>217.174.156.100</code>

By default, Synapse displays the column(s) for the **primary property** of any object (**node**) in the Results Panel.

For a DNS A record (**inet:dns:a** node), the primary property is the **combination** of the domain (**:fqdn**) and the IPv4 address (**:ipv4**) that the DNS A record points to.

Question 2: How does the **Results Panel** change when you toggle on the **.seen** property?

- Synapse adds two **date/time columns**, one for the "first seen" date/time (**.seen[min]**) and one for the "last seen" date/time (**.seen[max]**):

	:fqdn	:ipv4	.seen[min]	.seen[max]
↔	mail.usnewssite.com	69.195.129.72	2015/01/11 15:18:33	2016/11/09 03:48:31
↔	dod.dnsweb.org	184.168.221.96	2014/08/16 00:04:09	2014/08/16 00:04:09.001
↔	ttl.tfxdccssl.net	217.174.156.100	2016/09/27 09:16:20	2017/01/19 10:59:29

.seen ("dot seen") is a **universal property** - every form in Synapse has a **.seen** property that you can optionally use to record the dates/times when an object was "seen" (observed, known to exist, etc.). Because **.seen** consists of a **pair** of date/times, Synapse displays each in its own column.

Question 3: How does the **Results Panel** change when you toggle on the **cno.infra.dns.sink.hole.kleissner** tag?

- Synapse adds two **date/time columns** associated with the tag:

.seen[max]	cno.infra.dns.sink.hole.kleissner[min]	cno.infra.dns.sink.hole.kleissner[max]
2016/11/09...	null	null

Tags can have date/times associated with them. Tag date/times can be used to indicate "when" the assessment that the tag represents was observed, true, or valid.

The **cno.infra.dns.sink.hole.kleissner** tag on this node does not have any date/times, so the columns' values are **null**.

Tip: When adding **tags** to the **Tabular** mode display, Synapse's default behavior is to add the tag's **date/time** columns.

You can add a column to show the **tag itself** using the **Edit Columns** menu option (covered in a later exercise.)

Exercise 6 Answer

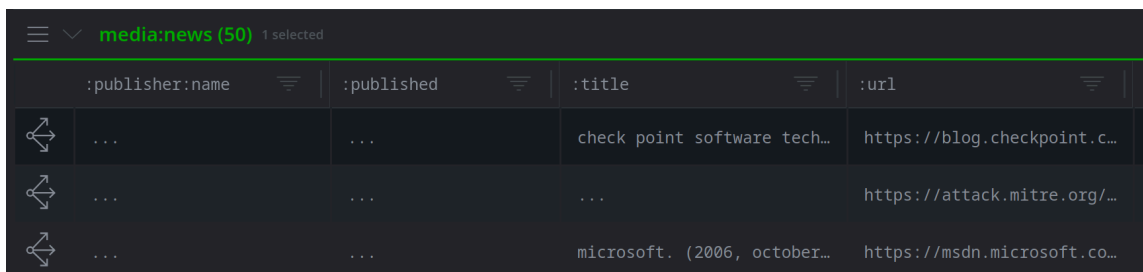
Objectives:

- Understand how to customize Tabular display in the Research tool.
- Know how to add and remove properties from the All Props tab of the Details Panel.
- Know how to modify columns using the Edit columns menu option.

Part 1 - Use the Details Panel to view nodes

Question 1: What columns are displayed in the **Results Panel** for the **media:news** nodes?

- Synapse displays the columns for:
 - the publisher name (**:publisher:name**)
 - the publication date (**:published**)
 - the title (**:title**), and
 - the location (**:url**):



	:publisher:name	:published	:title	:url
↻	check point software tech...	https://blog.checkpoint.c...
↻	https://attack.mitre.org/...
↻	microsoft. (2006, october...	https://msdn.microsoft.co...

Question 2: What properties are **set** for the **media:news** node you selected?

- Most nodes have a small number of properties configured:

```

NODE  ALL TAGS  ALL PROPS  ANATOMY
├── media:news
│   └── 000ef1294631f66bba7ef8f26c957f79
├── :title      cimpanu, c. (2016, april 26). malware shuts...
├── :url        https://news.softpedia.com/news/on-chernoby...
├── :url:fqdn   news.softpedia.com
├── .created    2022/09/05 12:00:02.695
└── + Add Tags

```

The properties may include:

- The title (**:title**);
- The URL where the article can be found (**:url**); and
- The FQDN of the URL (**:url:fqdn**).

Every node in Synapse has a **.created** property to show when it was added to Synapse.

The **NODE** tab displays details about the selected node. The tab shows **only** properties that are **set** (have a value) and any tags that have been applied.

Question 3: What properties are **available** for this **media:news** node (that is, what additional properties **could** be set for this node)?

- Several additional properties are **available**:

NODE	ALL TAGS	ALL PROPS	ANATOMY
▪	+author	...	
▪	:authors	...	
▪	:ext:id	...	
▪	:file	...	
▪	+org	...	
▪	:published	...	
▪	:publisher	...	
▪	:publisher:name	...	
▪	:rss:feed	...	
▪	:summary	...	
▪	:title	cimpanu, c. (2016, april 26). malware...	
▪	:topics	...	
▪	:type	...	
▪	:updated	...	
▪	:url	https://news.softpedia.com/news/on-ch...	
▪	:url:fqdn	news.softpedia.com	
▪	.created	2022/09/05 12:00:02.695	
▪	.seen	...	

The **ALL PROPS** tab displays the properties that are **available** for a node. This includes:

- properties that are **set** (like **:title** in the image), and
- properties that are **not set** - the three dots (. . .) mean the property is not configured and currently has no value.

Tip: In Synapse, most secondary properties are **optional**. You can create nodes even if you only have limited information available. You can always go back and add or update information later!

Note: A **line** through a property name means the property has been **deprecated**. This means we have made changes to the data model to improve Synapse; usually we have added a property (or a new form) to replace the deprecated one.

You can still use deprecated properties. The line tells you that the property will be removed from a future version of Synapse. This gives you time to change the way you model data and / or migrate any existing data if necessary.

You can view details on our [Data Model Deprecation Policy](#) in the Synapse online documentation.

Console Help

Exercise 7 Answer

Objective:

- Understand how to use the Console Tool to:
 - list available help,
 - search for specific commands, and
 - display help / options for individual commands.

Question 1: What commands / package(s) / Power-Up(s) are displayed?

- Synapse displays any installed command that contains the string **min**:

```
> help min
The following Storm commands are available:
package: synapse
min      : Consume nodes and yield only the one node with the lowest value for an expression.

service: maxmind (b6a8d6a7f1f8afd7be4814c13bf91787)
package: synapse-maxmind
maxmind: Enrich nodes using the MaxMind GeoIP2 metadata.

For detailed help on any command, use <cmd> --help
complete. 0 nodes in 1698 ms (0/sec).
```

There are two commands:

- The **min** command (part of the default Synapse package).
- The **maxmind** command (part of the Synapse-Maxmind Power-Up).

Question 2: What does the **min** command do?

- The **min** command takes a set of results (nodes) and finds a node with the lowest or smallest value for a property (such as a size or date):

```
> min --help

Consume nodes and yield only the one node with the lowest value for an expression.

Examples:

// Yield the file:bytes node with the lowest :size property
file:bytes#foo.bar | min :size

// Yield the file:bytes node with the lowest value for $tick
file:bytes#foo.bar +.seen ($tick, $stock) = .seen | min $tick

// Yield the it:dev:str node with the shortest length
it:dev:str | min $lib.len($node.value())

Usage: min [options] <valu>

Options:

  --help                : Display the command usage.

Arguments:

  <valu>                : The property or variable to use for comparison.
complete. 0 nodes in 3 ms (0/sec).
```

We will examine this and other useful commands later in the course!