

# Synapse Bootcamp - Module 14

## Modeling Data Manually - Answer Key

<b>Modeling Data Manually - Answer Key</b>	<b>1</b>
<b>Answer Key</b>	<b>2</b>
Modeling Data Manually	2
Exercise 1 Answer	2
Exercise 2 Answer	4
Exercise 3 Answer	6

---

# Answer Key

## Modeling Data Manually

### Exercise 1 Answer

**Objective:**

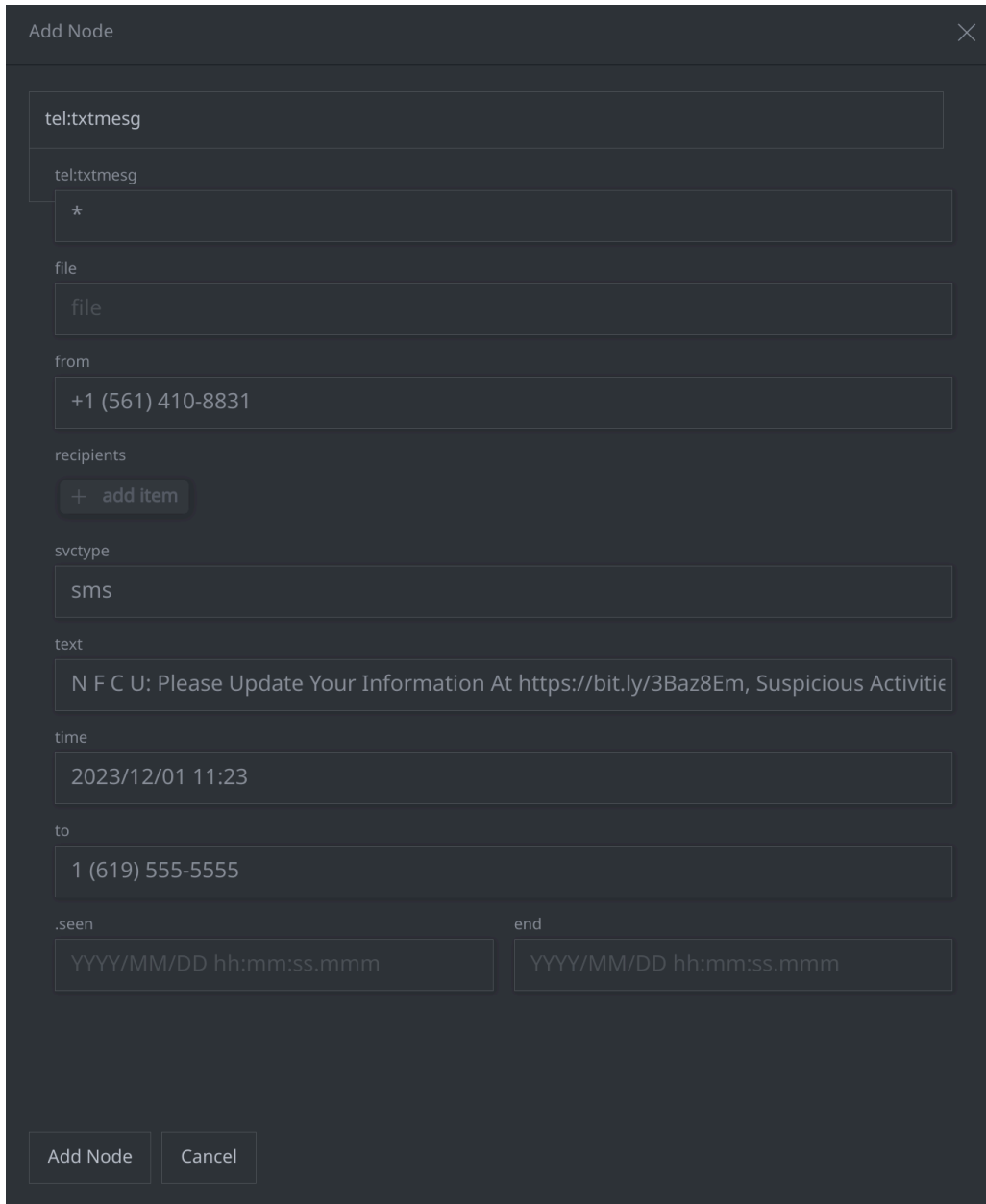
- Use real world data to model an SMS-based phishing attack.

**Question 1:** What does your new **tel:txtmesg** node look like?

- Your completed node should look similar to the following:

NODE		ALL TAGS	ALL PROPS	ANATOMY
tel:txtmesg		88d5ad15280b68c310e3ad814a3a3bc1		
:from		15614108831		
:svctype		sms		
:text		N F C U: Please Update Your Information At https://...		
:time		2024/05/20 13:01:00		
:to		16195555555		
.created		2024/05/21 14:41:08.378		

When you fill in the **Add Node** dialog, it should look similar to the following:



When creating nodes, the **Add Node** dialog helps by showing you available properties and providing hints (tooltips) for the type of data to fill in.

Because secondary properties are all **optional**, you can always create the node and add or edit properties later in the Research Tool.

## Exercise 2 Answer

### Objective:

- Use real world data to represent contact information for a company.

### Part 1

**Question 1:** What happened? Did the command create a **new** node, or lift an existing node? How can you tell?

- Synapse displays an **ou:org** node for Siemens:



The image shows a Synapse interface with a dark theme. At the top, there is a search bar with a magnifying glass icon and the text "gen.ou.org siemens". Below the search bar, there is a "Tabular" view icon. Underneath, it says "ou:org (1)". A table is displayed with the following columns: ":alias", ":name", ":names", and ":loc". The table has one row with the following values: "siemens", "siemens", "(siemens, siemens ag, siemens aktiengesellschaft)", and "de.munich".

:alias	:name	:names	:loc
siemens	siemens	(siemens, siemens ag, siemens aktiengesellschaft)	de.munich

- Synapse returned an **existing** node - the command used the name you provided (**siemens**) to find the node.

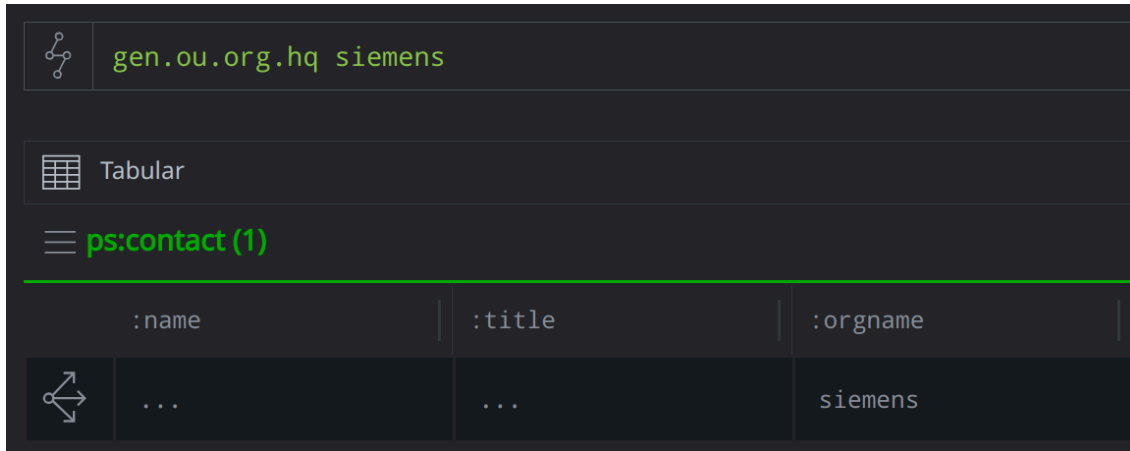
You can tell this is an existing node because:

- Many node properties are already set. If Synapse had created a **new** node, only the **:name** property would be set.
- The node has a **.created** timestamp of **2023/11/16 18:17:51.476**, which shows when it was created in Synapse.

## Part 2

**Question 2:** What happened? Did the command create a **new** node, or lift an existing node? How can you tell?

- Synapse displays a **ps:contact** node for Siemens:



The screenshot shows the Synapse interface. At the top, there is a search bar with the text "gen.ou.org.hq siemens". Below the search bar, there is a "Tabular" view icon. Underneath, a section titled "ps:contact (1)" is expanded, showing a table with three columns: ":name", ":title", and ":orgname". The table contains one row with the values "...", "...", and "siemens".

:name	:title	:orgname
...	...	siemens

- Synapse returned a **new** node. If the **ou:org** for the name **siemens** does not have its **:hq** property set, the command creates a new contact.

You can tell this is a new node because:

- Only the **:orgname** (and **:org**) properties are set. Synapse automatically links the contact and the organization (**ou:org**).
- The **.created** time for the node should be the current date and time. This shows the node was just added.

---

**Question 3:** What does your new **ps:contact** node look like?

- Your **ps:contact** should look similar to the following:

NODE		ALL TAGS	ALL PROPS	ANATOMY
▪ ps:contact		f44feac8dfbc3977d8015b5aca6b6681		
▪ :address		werner-von-siemens-straße 1 80333 mu...		
▪ :email		contact@siemens.com		
▪ :loc		de.munich		
▪ :org		8cf488734f435905ca2165ae64d14fce		
▪ :orgfqdn		siemens.com		
▪ :orgname		siemens		
▪ :phone		498938035491		
▪ :phone:fax		49697976664		
▪ :url		https://www.siemens.com/global/en/ge...		
▪ .created		2024/05/21 14:28:38.214		

---

## Exercise 3 Answer

### Objective:

- Use public reporting to represent basic information about a compromise.

**Question 1:** What does your **risk:compromise** node look like?

- Your **risk:compromise** node should look similar to the following:

NODE

ALL TAGS

ALL PROPS

ANATOMY

▪ risk:compromise

ec91b206c19e82820324130b697edb84

▪ :desc

Winnti compromise of German corpo...

▪ :name

siemens compromise (2016)

▪ :reporter:name

quointelligence

▪ :target

f44feac8dfbc3977d8015b5aca6b6681

▪ :time

2016/01/01 00:00:00

▪ .created

2024/05/21 14:18:40.346


**Question 2:** Can you use the **Explore** button to navigate from the Siemens **ou:org** node, to its **ps:contact(s)**, and then from the **ps:contact** to the **risk:compromise** node?

- **Yes.** First **Explore** from the **ou:org** node:

ou:org (1)

	:alias	:name	:names	:loc
<div><div></div><div></div></div>	siemens	siemens	(siemens, siemens ag, siemens aktiengesell schaft)	de.munich
	Explore			

Then locate the **ps:contact** for Siemens' headquarters and **Explore** from that node:

≡ ps:contact (1)		
:name	:title	:orgname
 Explore	...	siemens

Then locate the **risk:compromise** node in the Results Panel (use **Scroll to Form** if necessary):

≡ tel:phone (2)

tel:phone

498938035491

49697976664

≡ risk:compromise (1)

	:name	:time	:lasttime	:reporter::name
<div></div>	siemens compromise (2016)	2016/01/01 0...	...	...

If the nodes are **not** linked, the **risk:compromise:target** property may not be set correctly.

**Question 3:** Is there a Storm query you could use to do this?

- **Yes.** The following Storm query will also find the compromise:



```
ou:org:name=siemens -> ps:contact -> risk:compromise | uniq
```

---