THE VERTEX PROJECT PRESENTS

THE APT1 SCAVENGER HUNT

Table of Contents

Table of Contents	2
Introduction	3
What's New for v1.11	3
About the APT1 Scavenger Hunt	3
Before You Begin	4
What's Included in My Demo Instance of Synapse?	5
Getting Started	8
Optic	8
The Storm Query Bar	8
The Storm Query Language	9
Challenges	10
Basic Challenges	12
Intermediate Challenges	16
Advanced Challenges	31
Hints	39
Basic Challenges	39
Intermediate Challenges	41
Advanced Challenges	49
Answer Key	53
Basic Challenges	54
Intermediate Challenges	59
Advanced Challenges	69
Answer Explanations	74
Basic Challenges	74
Intermediate Challenges	97
Advanced Challenges	211

Introduction

What's New for v1.11

The APT1 Scavenger Hunt (v1.0) has been revised to reflect:

- Additions and changes to the Synapse data model;
- New features and capabilities of the Synapse UI (Optic); and
- Revisions to The Vertex Project's tags and tag trees to better support analysis.

The majority of Scavenger Hunt questions remain the same, but screen captures, UI-based instructions and answers, and associated Storm queries have been updated to reflect these changes.

- **v1.1** has been updated to account for the Synapse data migration from deprecated it:av:sig and it:av:filehit nodes to the newer it:av:scan:result nodes (specifically, Questions 28 and 29 have been revised).
- **v1.11** has been updated to emphasize selecting the correct Workspace and View for the Scavenger Hunt.

Our commercial offering, Synapse Enterprise, is an on-premises solution that includes the <u>Synapse UI</u> (Optic) and a large suite of integrations called <u>Power-Ups</u>. **The APT1 Scavenger Hunt requires a cloud-hosted demo instance of Synapse Enterprise,** which comes pre-loaded with the Scavenger Hunt data. You can request a demo instance at no cost using the form on The Vertex Project's <u>website!</u>

IMPORTANT: The data for the Scavenger Hunt v1.1 resides in a dedicated view within your demo instance (the **APT1 Scavenger Hunt Data** view). If you have an older demo instance of Synapse and this view is not visible in your <u>View Selector</u>, you can <u>request a new demo instance</u>.

About the APT1 Scavenger Hunt

The Vertex Project's APT1 Scavenger Hunt is designed to help you:

- Test and explore the **Synapse** central intelligence system using **real world data**.
- Become familiar with the **Synapse UI (Optic)**, display modes, and analysis tools.

• Practice using the **Storm query language** to answer analytical questions.

We chose Mandiant's 2013 <u>APT1 report</u> (and associated data) for this Scavenger Hunt for a number of reasons:

- The report includes a large set of indicators, which provides a rich data set to work with.
- Many of the indicators (hashes / files, domains) referenced in the report have a wealth of enrichment data available, which further expands the available data.
- The APT1 report was the first public report to attempt to provide sufficient data to prove attribution of a set of threat activity beyond a reasonable doubt. This makes the report a good case study in the attribution process.

The Vertex Project's APT1 Scavenger Hunt includes:

- 32 questions or "challenges" to answer.
- Hints for each challenge, if you get stuck or just need a few pointers.
- The answers to each challenge. Each answer links to a detailed explanation of the solution.

Most of all, we hope this Scavenger Hunt provides a fun and useful way to get you started (and excited about) working with Synapse and Storm - so **have fun!**

Before You Begin

Select Your Workspace

In Synapse, you can <u>customize</u> many aspects of your user environment (**Workspace**). To make it easier for you to jump right into the Scavenger Hunt, your demo instance includes a **pre-configured Workspace** tailored to the Scavenger Hunt and its data. To use this Workspace, follow the instructions to <u>Select a Workspace</u> and choose the **APT1 Scavenger Hunt Workspace**:

APT1 Scavenger Hunt Workspace ✓

We recommend starting with this Workspace and further customizing it if necessary to meet your needs.

Select Your View

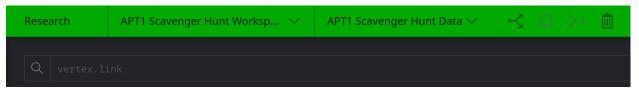
The Scavenger Hunt data is stored in its own **view** within your Synapse demo instance. Follow the instructions to **Select a View** and choose the **APT1 Scavenger Hunt Data** view to do the Scavenger Hunt:



Tip: You can optionally create a **fork** (fork a view) of the APT1 Scavenger Hunt Data view to preserve the original APT1 data set from any accidental modifications.

We strongly recommend creating a fork if you plan to add data to the APT1 Scavenger Hunt view, as adding data may affect the answers to some challenges.

Once you have made your selections, your **Top Bar** should look like this:



What's Included in My Demo Instance of Synapse?

Data

Your Synapse demo instance has been loaded with the Scavenger Hunt data, including:

- **Indicators** (IOCs) from the original <u>Mandiant APT1 report</u> and the associated <u>Digital Appendices</u> (tagged **rep.mandiant.apt1**).
- Other relevant objects, including organizations (ou:org), contacts (ps:contact), articles/reports (media:news), threat clusters (risk:threat), industries (ou:industry), etc.
- Enrichment data for many of the IOCs, ingested using Synapse Power-Ups.
 Enrichment data may include:

- IPv4 / IPv6 Autonomous System (AS), geolocation, and WHOIS (netblock registration) data.
- FQDN DNS data (A, AAAA, NS, CNAME, SOA, etc.).
- FQDN WHOIS data.
- Files (file:bytes) associated with reported hashes.
- o File metadata, antivirus / multiscanner data, and/or sandbox execution data.
- Additional data, objects, and reporting (not necessarily related to APT1). For example:
 - other articles (media:news nodes) and associated IOCs;
 - o the pre-loaded DNS public suffix list; and
 - nodes corresponding to the <u>MITRE ATT&CK</u> Enterprise, Mobile, and ICS matrixes (as loaded by the publicly available <u>synapse-mitre-attack</u> <u>Power-Up</u>).

Tags

Various **tags** (labels) have been applied to the demo data. Tags from third-party vendors (such as VirusTotal) may be applied automatically by Power-Ups. Tags have also been applied by Vertex analysts to record observations or assessments about the data. Tags are visible on individual nodes; you can also browse the current set of tags in your instance of Synapse using the **Tag Explorer** (available via the Synapse **Help Tool**). For more information on how tags are used in Synapse, see our **User Guide**.

Workspace

A Synapse **Workspace** is a customizable user environment. Your demo instance includes a Workspace (the **APT1 Scavenger Hunt Workspace**) that has been **pre-configured** with a number of useful options for displaying Synapse's data. You can learn more about Workspaces and available <u>customizations</u> in the <u>Synapse UI (Optic) User Guide</u>.

Storm Commands

Your Synapse demo instance includes all <u>Storm commands</u> that are built into Synapse, as well as commands associated with any installed Synapse **Power-Ups.** Some built-in Storm commands may be useful to solve certain challenges; these commands are referenced in the Hints (and in the Answers / Explanations).

Synapse Power-Ups

Synapse Power-Ups are **not required** to solve the Scavenger Hunt. However, Power-Ups are available for you to use, install (from the <u>Power-Ups Tool</u>), and test. Note that some Power-Ups require that you install and configure an API key, which may be free or paid, depending on the Power-Up and associated vendor. See the <u>Power-Up documentation</u> for details; documentation is also available through the **Power-Ups Tool** or <u>Help Tool</u> in Synapse.

Getting Started

If you're brand new to Synapse, Optic, and / or Storm, this section provides some brief background and links to resources to point you in the right direction. If you're already comfortable with Synapse (or just like adventure), you can go straight to the **Challenges**!

Optic

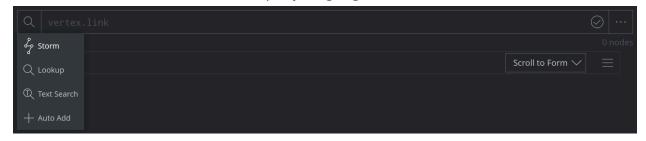
The **Optic** UI is designed to make using Synapse **easy and intuitive**, especially if you are new to Synapse or still learning the Storm query language. Optic menus, display options, buttons, and tools allow you to perform actions by selecting an option or clicking an icon, making it easy to navigate the data in Synapse while running Storm "under the hood".

If you are not familiar with Optic, the <u>Quick Tour</u> describes key UI features, menus, and navigation, while the <u>User Guide</u> provides more detailed information on several of Optic's features.

The User Guide sections on <u>Customizing Your Environment</u> and <u>Getting Help</u> (which includes information about the <u>Data Model Explorer</u> and <u>Tag Explorer</u>) are also very useful.

The Storm Query Bar

The <u>Storm Query Bar</u> is located at the top of the <u>Research Tool</u>. Most interactions with Synapse start with the query bar. You can use the query bar in different modes, including <u>Lookup mode</u> (a simple way to ask about indicators when you're just getting started) and <u>Storm mode</u> (which uses the Storm query language).



To start your investigation, you need to "ask" Synapse about the data you're interested in. "Asking about" (selecting) a set of data from Synapse's data store is known as **lifting** the data (or performing a **lift** operation).

The Storm Query Language

Synapse is built around the **Storm** query language, which gives you a **powerful and flexible** way to interact with data in Synapse. With Storm, you can ask and answer any analytical question!

Most importantly, Storm is **intuitive** and **easy to learn** because you can start with basic queries and build from there.

The Scavenger Hunt challenges will help teach you Storm and let you practice writing Storm queries, slowly introducing additional concepts. At its most basic level, Storm consists of the following operations:

- **Select** ("ask about") data (known as a **lift** you "lift" an object or set of objects from Synapse's data store to start your investigation);
- Filter a set of results to view the subset of data you're interested in;
- Navigate from your current results to adjacent / "connected" nodes. Technically, you navigate by:
 - o **Pivoting** between objects that share a property value, or
 - **Traversing** between objects that are connected by an edge.

You can also use Storm to **run commands** that act on the data in various ways; we'll introduce some useful commands as you progress through the Scavenger Hunt.

The <u>Storm Quick Reference</u> materials include several "cheat sheets" that provide at-a-glance syntax for the most commonly used Storm operations.

For more detailed information, see the <u>Storm Reference Guide</u> (part of the <u>Synapse User Guide</u>), which provides background on Storm and detailed explanations and examples of the various Storm operations / operators. (Links to the Synapse and Storm Guides can also be found under the <u>Documentation</u> tab in the Synapse **Help Tool**.)

Challenges

The Scavenger Hunt consists of 32 challenges - questions for you to answer about APT1 based on the data in Synapse.

The challenges are divided into sections (Basic, Intermediate, and Advanced), based on the type of question and how you might solve the challenge. They progress from simple challenges to more complex ones.

Each challenge question links to a set of **Hints** to point you in the right direction if you're not sure how to proceed, and to an **Answer** section that provides the solution.

Each **Answer** links to a detailed explanation of the solution (**Answer Explanations**), and describes how to solve the challenge by either:

- using Optic's UI features (buttons, menus, etc.) to navigate the data and find the solution; or
- using the Storm query language to directly ask and answer the challenge questions.

The answer is presented as "either / or" solely for the sake of simplicity! Realistically, analysts work with Synapse using a combination of Storm and UI features based on their needs and preferences, so "real world" solutions to each challenge would almost always include some combination of Storm and UI navigation.

There is no "right" way to solve the Scavenger Hunt challenges! Some paths to the solution may be more efficient than others, but how you arrive at the solution - using the UI, Storm, or a combination of both - is entirely up to you.

Tip: The **Answers** section may include either screen captures of relevant queries **or** a text box containing the query (for readability). The **Answer Explanations** will always use a text box so you can copy/paste any query from this PDF into your Synapse query bar.

Some answers also include **Bonus Notes** as part of the Answer Explanations. These sections briefly dive into a more detailed explanation of some part of the solution. Read them if you want to geek out with us about a particular aspect of Synapse, but they are not required.

Finally, we have included a few **Analytical Challenges** throughout the Scavenger Hunt. These challenges don't have specific answers, but are meant to encourage you to think about a particular analytical problem and how you might solve it (ideally using Synapse!).

Note: We are constantly updating Synapse! While we do our best to ensure that our documents are up-to-date, you may notice small discrepancies (such as between a screen capture in the documents and the appearance of your current instance of Synapse).

If something is unclear or if you identify an outright error, please reach out to us so we can assist!

Basic Challenges

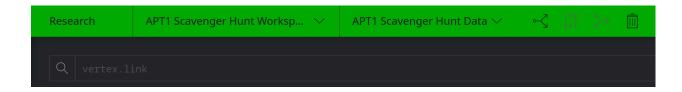
The Basic Challenges are meant primarily as a "warm up" to get you started with Synapse, Optic navigation, and a bit of Storm!

The Basic Challenges can be solved using:

- The Storm query bar in **Lookup mode** or **Storm mode**.
- Basic / simple lift operations in Storm.
 - See the Storm Quick Reference quide to get started with lifts.
- Using Storm to lift by <u>tag</u> (or to <u>lift forms by tag</u>).
- Optic navigation (e.g., using the **Explore button**).
- Optic Research Tool display modes.

Question 0

Check that you have selected the correct **Workspace** and **View** for the Scavenger Hunt. Your <u>Top Bar</u> should look like this:



See the **Before You Begin** section for help making these selections.

Question 1

One of the first things people learn to do with Synapse is look up basic indicators of compromise (IOCs). Synapse provides a quick and easy way to ask "what do we know about" a hash, domain, URL, or other common IOC.

"Show me the MD5 hash 5e0df5b28a349d46ac8cc7d9e5e61a96."

- How can you look up the MD5 hash 5e0df5b28a349d46ac8cc7d9e5e61a96 in Synapse?
 - A hash:md5 object (node) represents an MD5 hash.

- What tags are on this node, and what do the tags tell you about the hash?
- Does this hash have an associated file in Synapse?
 - A file:bytes node represents a file.
- Has the hash been published in any reports or articles? If so, how many?
 - o A media: news node represents an article or report.

Tip: We encourage copy / paste into Synapse from this PDF when it is helpful (nobody wants to type that MD5 hash)!

<u>Question 1 - Hints</u> <u>Question 1 - Answer</u>

Question 2

"Show me the FQDN globalowa.com."

- How can you look up the Fully Qualified Domain Name (FQDN) globalowa.com in Synapse?
 - o An inet:fqdn node represents an FQDN.
- How many DNS A records are present in Synapse for this FQDN?
 - o Ainet:dns:a node represents a DNS A record.

Question 2 - Hints

Question 2 - Answer

Question 3

"Show me the email address 1fengg@163.com."

- How can you look up the email address lfengg@163.com in Synapse?
 - An inet:email node represents an email address.
- How many FQDNs were registered using this email address?
 - An inet:whois:email node represents the association between a domain whois record and an email address that appears in the whois record.
- How many of the FQDNs registered with this email address were reported by Mandiant as associated with APT1?

• The tag rep.mandiant.apt1 is used to note that Mandiant attributes an indicator or other object to APT1.

Question 3 - Hints
Question 3 - Answer

Question 4

Synapse uses labels called **tags** to provide context to objects (nodes) and to group related objects. With Synapse you can ask about objects that are labeled with the same tag.

"Show me all the APT1 indicators reported by Mandiant."

- How can you look up (lift) all the indicators Mandiant reported as associated with APT1?
 - The tag rep.mandiant.apt1 is used to note that Mandiant attributes an indicator or other object to APT1.
 - In Storm, the hashtag (#) symbol is used when we want to refer to a tag applied to a node (i.e., #rep.mandiant.apt1).
- How many indicators are there in total?
- What type of APT1 indicator (i.e., form) was **most** reported by Mandiant? How many were reported?

Question 4 - Hints
Question 4 - Answer

Question 5

"Show me all the APT1 MD5 hashes reported by Mandiant."

- How can you lift only the MD5 hashes Mandiant reported as associated with APT1?
- For the MD5 hashes, which malware family was **most** reported (i.e., had the largest number of associated hashes)?
 - Various rep.mandiant.* tags are used to note that Mandiant associates an indicator with a malware family.

Question 5 - Hints Question 5 - Answer

Intermediate Challenges

The Intermediate Challenges are designed to get you more comfortable with Optic, and with performing common Storm operations such as **lifts**, basic **filters**, and **pivots**.

The Intermediate Challenges can be solved using:

- The elements from the Basic Challenges (Optic navigation, Research Tool display modes, etc.)
- Stom lift operations using additional <u>standard comparison operators</u>.
- Storm queries that <u>chain</u> multiple Storm operations (lift, filter, pivot, etc.) together.
- Basic filters by <u>form</u> or <u>property</u>, or filters using <u>standard comparison</u> <u>operators</u> in Storm.
- Pivoting in Storm using the standard <u>pivot out</u> operator.
- <u>Traversal</u> of light edges in Storm.
- Storm commands such as **count**, **limit**, **max**, **min**, and **uniq**.
- Storm lifts using <u>extended comparison operators</u>.
- Storm filters using <u>extended comparison operators</u>.

Many intermediate challenges can still be solved using lifts and UI navigation. However, many are **easier** to solve using Storm. This is because Storm allows you to more narrowly target your question and ask **specifically** about the data you're interested in. While the UI may eventually get you to the answer, you may have to navigate through a lot of extra data first.

Tip: If you prefer using the UI to navigate, keep in mind that even when using the UI, every "question" in Synapse starts with a **lift** operation. You can lift nodes using either Lookup mode or Storm mode, but Storm mode gives you the greatest flexibility in telling Synapse "where you want to start". At minimum, we encourage you to learn to initially **lift** (ask about) nodes using Storm, even if you use the UI from then on!

Working with Tags

With Synapse, you can "compare" tags (labels) - that is, ask which nodes have a particular combination of tags. One common use case is to compare indicators from public reporting (where tags are used to represent different organizations' reporting on threats or malware). With these types of queries, you can see where public reports about threats overlap (or not).

Question 6

"Show me the Mandiant APT1 indicators that are <u>also</u> associated with Comment Crew by Symantec."

- How many APT1 indicators reported by Mandiant were also reported by Symantec as associated with the "Comment Crew" threat group?
 - The tag rep.symantec.commentecrew is used to note that Symantec attributes an indicator or other object to the Comment Crew.
- What types of indicators (forms) were reported by both organizations?

Question 6 - Hints
Question 6 - Answer

Question 7

"Show me the Symantec Comment Crew indicators that were <u>not</u> reported by Mandiant."

- How many indicators did Symantec report for Comment Crew that were not reported by Mandiant for APT1?
- What types of indicators did Symantec report?

<u>Question 7 - Hints</u> <u>Question 7 - Answer</u>

Question 8

"Show me the Symantec Comment Crew indicators that are <u>also</u> associated with Putter Panda by Crowdstrike."

- How many Comment Crew indicators reported by Symantec were **also** reported by Crowdstrike as associated with the threat group Putter Panda?
 - The tag rep.crowdstrike.putterpanda is used to note that Crowdstrike attributes an indicator or other object to Putter Panda.
- What types of indicators were reported by both organizations?

<u>Question 8 - Hints</u> <u>Question 8 - Answer</u>

Analytical Challenge 1

- In Question 6 and Question 7, we investigated where there was overlap (or not) between Mandiant's APT1 indicators and Symantec's Comment Crew indicators. Based on the results, do you think Mandiant's APT1 and Symantec's Comment Crew are the same threat group? Why or why not? If not, what additional data might help you decide one way or the other?
- Similarly, in <u>Question 8</u>, we investigated the overlap between Symantec's Comment Crew and Crowdstrike's Putter Panda. Based on the results, do you think Symantec's Comment Crew is the same threat group as Crowdstrike's Putter Panda? Why or why not? If not, what additional data might help you decide one way or the other?

Analytical Challenge 1 Notes

Working with Hashes

With Synapse, you can easily see how objects are "connected" to other objects. Both the Optic UI and the Storm query language allow you to navigate and explore the data to find interconnections and related objects. In the next sections, you'll see different examples of questions and queries to navigate common types of threat intel data.

Question 9

"Show me the files associated with the APT1 MD5 hashes reported by Mandiant."

• How many APT1 MD5 hashes reported by Mandiant have associated files?

Question 9 - Hints
Question 9 - Answer

Working with FQDNs and Infrastructure

Question 10

"Show me all of the APT1 FQDNs reported by Mandiant."

- How many APT1 FQDNs were reported by Mandiant?
- Of all the APT1 FQDNs reported, what are the **top six** most commonly used subdomains / host names?
 - The host portion of an FQDN (inet:fqdn) is recorded in its:host property.

"Show me all of the APT1 FQDNs that have a hostname of 'e'."

How many APT1 FQDNs use the subdomain / hostname 'e'?

Question 10 - Hints
Question 10 - Answer

Question 11

"Show me all of the IPv4 addresses that the FQDN jobsadvanced.com has resolved to."

- How many IPv4 addresses has the APT1 FQDN jobsadvanced.com resolved to?
 - An FQDN resolving to an IPv4 address is represented by a DNS A record (inet:dns:a node).
 - An inet:ipv4 node represents an IPv4 address.
- How many of those IPv4 addresses represent domain registrar parking infrastructure, based on annotations (tags) in Synapse?
- Which IPv4 address(es) is / are confirmed threat actor infrastructure, based on annotations (tags) in Synapse?
- When did the threat actor use the IPv4 address(es), based on annotations in Synapse?

Question 11- Hints
Question 11 - Answer

Question 12

"Show me the IPv4 addresses that jobsadvanced.com and <u>any of its subdomains</u> have resolved to."

- How many IPv4 addresses have the APT1 FQDN jobsandvanced.com and any of its subdomains resolved to?
 - Subdomains of an FQDN are part of the FQDN's zone of control and have the FQDN as their inet: fqdn:zone property value.
- For all of the IPv4 addresses, what Autonomous System(s) hosted the **most** IPs?
 - The Autonomous System (AS) number associated with an IPv4 address is stored in the IPv4's :asn property.
- For all of the IPv4 addresses, what country (or countries) hosted the **most** IPs?
 - The geolocation information for an IPv4 address is stored in the IPv4's :loc property.

Question 12 - Hints Question 12 - Answer

Question 13

"Show me all of the APT1 FQDNs that are zones."

- How many of the APT1 FQDNs reported by Mandiant are zones?
 - Whether an FQDN is a zone is stored in its:iszone property.

Question 13 - Hints
Question 13 - Answer

Question 14

"Show me all of the zones associated with all of the APT1 FQDNs."

- How many FQDN zones, in total, are associated with all of the APT1 FQDNs reported by Mandiant?
 - An FQDN's zone is stored in its: zone property.

Question 14 - Hints
Question 14 - Answer

Analytical Challenge 2

- In Question 13, we saw that Mandiant reported 117 FQDNs that are also zones. But in Question 14, we saw that all of the FQDNs reported by Mandiant were associated with a total of 149 FQDN zones. This means that in some cases, Mandiant reported a **subdomain** as an APT1 indicator (e.g., www.coachmotor.com) but did not report the associated **zone** (e.g., coachmotor.com).
- Why do you think these zones were not included in Mandiant's list of APT1 FQDNs?
- How might you use data and / or tags in Synapse to distinguish among things like:
 - An FQDN registered and controlled by a threat group?
 - An FQDN associated with a compromised website?
 - A legitimate FQDN or Internet service used maliciously?

Analytical Challenge 2 Notes

Working with Domain Whois Data

Legacy whois data (where registrant information is more readily available) is useful for correlating historical threat activity. While most modern domain whois data masks the registrant data for privacy purposes, whois records are still useful for determining when domains were registered or changed hands, or for identifying operational patterns (such as threat actors' preferred domain registrars, name servers, or other infrastructure).

Question 15

"Show me all of the domain whois records for the FQDN youipcam.com."

- How many domain whois records are available for the FQDN youipcam.com?
 - A domain whois record is represented by an inet:whois:rec node.
- What is the date of the **earliest registration** of this FQDN?
 - The registration date for an FQDN is captured in the :created property of its whois records (inet:whois:rec:created).
- What is the date of the **most recent capture** of whois data for this FQDN?
 - The "capture" date for a domain whois record is recorded in its :asof property (inet:whois:rec:asof).
- Based on whois records and annotations (tags) in Synapse, when was this FQDN registered by sinkhole organization Kleissner & Associates?

Question 15 - Hints
Question 15 - Answer

Question 16

"Show me all of the whois email addresses associated with the FQDN youipcam.com."

- How many email addresses are associated with domain whois data for FQDN youipcam.com?
 - An inet:whois:email node is used as a convenient method to link an FQDN with any email address used in the FQDN's domain registration data.
- Based on the results, what email address do you think was used as the registrant email address for the FQDN while it was controlled by APT1?
- Was this email address used to register any other FQDNs?

Question 16 - Hints

Question 17

"Show me the FQDNs registered using the email address issn.bgkit@yahoo.com."

- How many FQDNs were registered using this email address?
 - An inet:whois:email node is used as a convenient method to link an FQDN with any email address used in the FQDN's domain registration data.
- How many domain whois records are there for the FQDN(s)?
 - A domain whois record is represented by an inet:whois:rec node.
- How many domain whois records for the FQDN(s) contain the email address issn.bgkit@yahoo.com in the raw record?
 - The text of the raw whois record is stored in the record's :text property.
- What registrant name was used along with this email address to register the FQDN(s)?
 - The registrant name is stored in the whois record's :registrant property.
 Registrant names are also represented as their own inet:whois:reg nodes.
- Was this registrant name used to register any other FQDNs?

Question 17 - Hints
Question 17 - Answer

Analytical Challenge 3

- In <u>Question 17</u>, you identified that the email address issn.bgkit@yahoo.com was used to register only one FQDN (progammerli.com). However the registrant name "william michael" (used for progammerli.com) was also used to register a second FQDN (comrepair.net).
 - Is the use of the same registrant name enough evidence to decide that the two FQDNs are related / part of a single threat cluster? Why or why not?
 - o If not, what additional information might convince you?

Analytical Challenge 3 Notes

Working with Files

Files (whether exploits, scripts, documents, backdoors, or tools) and their behavior are a key component of identifying and correlating threat activity. With Synapse you can easily ask about files, file metadata, and file behavior. Not only can you understand the behavior of known / publicly reported files, but you can also use Synapse to identify unknown or unreported files based on similar behavior or metadata.

Question 18

"Show me the FQDNs queried by the file associated with APT1 MD5 hash 3107de21e480ab1f2d67725f419b28d0."

- What file is associated with MD5 hash 3107de21e480ab1f2d67725f419b28d0?
- What malware family is the hash / file associated with?
 - Various rep.mandiant.* tags are used to indicate Mandiant's reported malware families.
- Which FQDNs does this file make DNS requests for?
 - A DNS request is represented by an inet:dns:request node.
- Were these FQDN(s) reported by Mandiant?
- Were they reported by any other organization?

Question 18 - Hints Question 18 - Answer

Question 19

"Show me the files that make DNS queries for the APT1 FQDN earthsolution.org."

- How many files make DNS queries for that FQDN?
 - A DNS request is represented by an inet:dns:request node.

"Show me the files that make DNS queries for the APT1 FQDN earthsolution.org <u>or any</u> of its subdomains."

- How many files make DNS queries for earthsolution.org or any of its subdomains (i.e., any domains that have earthsolution.org as their inet:fqdn:zone property)?
- How many of these files were reported by Mandiant?
- What specific earthsolution.org subdomains do the files query?
- What other FQDNs do the files query?

Question 19 - Hints
Question 19 - Answer

Question 20

"Show me the APT1 files that have embedded PDB paths."

- How many APT1 files have PDB paths?
 - The PDB (program debug) path for a PE executable file is stored in the file:bytes:mime:pe:pdbpath property.
- How many unique PDB paths are there?
 - A PDB path is a file path, so each PDB path will have its own associated file:path node.
- How many of the PDB paths were **not** reported by Mandiant?
- Are there any other files in Synapse with the same PDB paths? How many?
- How many of those files were **not** reported by Mandiant?
- How many files **not** reported by Mandiant were reported by Symantec as "Comment Crew" rep.symantec.commentcrew)?
- How many files **not** reported by Mandiant were reported by Symantec as part of any activity (rep.symantec)?

Question 20 - Hints
Question 20 - Answer

Question 21

"Show me all the APT1 files that are signed with a code-signing certificate."

• How many APT1 files are signed with a code signing certificate?

- A crypto:x509:signedfile node links a file (file:bytes) with the primary certificate (crypto:x509:cert) extracted from the file.
- How many unique certificates were used to sign these files?
 - A crypto:x509:cert node represents a certificate.
- How many total files in Synapse were signed with the "Microsoft" certificate (i.e., the crypto:x509:cert with :subject='CN=Microsoft')?
 - Were any of those files **not** reported by Mandiant or Symantec?
 - Are the files associated with any malware families?

Question 21 - Hints Question 21 - Answer

Working with Light Edges

Lightweight (light) edges are used to represent some relationships in Synapse, such as "uses" or "targets". One of the most common light edges is "refs" (references).

Question 22

"Show me the articles that reference the FQDN update8.firefoxupdata.com."

- How many articles reference or report on the FQDN update8.firefoxupdata.com?
 - An article is represented by a media: news node.
 - Objects (such as indicators) that are reported or referenced in an article are linked to the article with a refs light edge.

Question 22 - Hints
Question 22 - Answer

Working with Threat Clusters

With Synapse's tags, you can track (and compare) your own firsthand analysis with activity reported by other organizations. This allows you to see where your analysis and reporting overlaps (or doesn't) with others, while still tracking different sets of reporting independently.

Question 23

"Show me the Vertex threat cluster(s) that overlap with Mandiant's APT1 indicators."

- How many Vertex threat clusters are associated with Mandiant's APT1 indicators, based on annotations in Synapse?
 - o Threat clusters tracked by Vertex use cno.threat.* tags.
- What are the clusters / cluster names?

Question 23 - Hints
Question 23 - Answer

Question 24

"Show me all of the indicators that Vertex asserts are <u>owned or controlled</u> by threat cluster T15."

- How many indicators are there?
 - The tag cno.threat.t15.own indicates objects that Vertex asserts are owned or controlled by threat cluster T15.
- What indicator was the "starting point" for threat cluster T15?
 - The tag cno.threat.t15.own.seed indicates the first indicator(s) that were associated with threat cluster T15 and used to build out the cluster.

"Show me all of the indicators that Vertex asserts have been <u>used by</u> threat cluster T15."

- How many indicators are there?
 - The tag cno.threat.t15.use indicates objects that Vertex asserts are used by T15, but not necessarily controlled by the cluster or unique to that cluster.

"Show me <u>all</u> of the indicators that Vertex associates with T15 (whether controlled or used by that group) (cno.threat.t15)."

- Based on the data annotated in Synapse, does T15 appear to be a coherent (fully connected) threat cluster?
- Based on the data annotated in Synapse, does Mandiant's APT1 appear to be a coherent (fully connected) threat cluster?

Question 24 - Hints Question 24 - Answer

Question 25

"Show me the domain whois records associated with T15."

- How many domain whois records are associated with T15?
 - o The tag cno.threat.t15.own indicates that T15 owns or controls a resource.
 - o A domain whois record is represented by an inet:whois:rec node.
- How many FQDNs did T15 register?
- Which T15 FQDN was the earliest/first FQDN registered by T15? When was it registered?
 - A domain's registration date is stored in the inet:whois:rec:created property.
- How many other FQDNs in Synapse were registered on the same date?
 - **Note:** "same date" does not mean at the **exact** same time.
- Which FQDN registered on that date is associated with APT1 but not T15?

Question 25 - Hints
Question 25 - Answer

Analytical Challenge 4

- In <u>Question 25</u>, you identified that the FQDN reutersnewsonline.com was an APT1 FQDN (according to Mandiant) and was registered on the same date as six other APT1 FQDNs that Vertex had added to their internal threat cluster T15. However, Vertex did **not** include reutersnewsonline.com in T15.
 - o Do you think reutersnewsonline.com should be added to T15? Why or why not?
 - If yes, what data / evidence supports adding the FQDN? Would this evidence make a coherent cluster in conjunction with existing T15 nodes?
 - o If not, what additional data / evidence, if available, might convince you?

Analytical Challenge 4 Notes

Advanced Challenges

The Advanced Challenges are meant to challenge you! These challenges make use of all the Synapse skills you've learned so far. In addition, they may:

- ask "bigger" questions over larger data sets;
- leverage more specialized parts of the Synapse data model;
- use lift / filter / pivot operations that are more use-case specific; or
- prompt you to analyze the data a bit in addition to using your Synapse skills.

These challenges are meant to show you some of the more detailed - but powerful - use cases and possibilities for Synapse!

Question 26

"Show me the files in Synapse that have PE metadata where the OriginalFilename is JpgAsp.exe."

Tip: Windows Portable Executable (PE) files may contain a broad range of metadata, including <u>VERSIONINFO</u> resources. These resources may contain string values describing the file, such as a CompanyName for the file's publisher or a ProductBuild with a build or version number. The Synapse-FileParser Power-Up can extract and model VERSIONINFO data from PE files.

In the IOC appendices of the APT1 report, Mandiant noted the NEWSREELS malware family used the value JpgAsp.exe for the OriginalFilename in the PE VERSIONINFO.

Synapse allows you to represent (model) and query very specific details to locate known data (in this case, files) and identify similar or related data.

- How many files in Synapse have PE file version information where the OriginalFilename is JpgAsp.exe?
 - An individual VERSIONINFO name / value pair is represented by a file:mime:pe:vsvers:keyval node.
 - A file:mime:pe:vsvers:info node links a name/value pair (file:mime:pe:vsvers:keyval) with a file (file:bytes) that uses the name/value.

 How many of these files were **not** publicly reported by Mandiant (rep.mandiant) or associated with a threat cluster by Vertex (cno.threat)?

Question 26 - Hints Question 26 - Answer

Question 27

With Synapse, you can easily ask questions that encompass thousands (or even hundreds of thousands) of nodes. You can quickly get answers about large data sets (such as the collective activity of a threat group).

"Show me all the files that make DNS queries for any APT1 or Comment Crew FQDNs."

- How many files in Synapse make DNS queries for any of the APT1 / Comment Crew FQDNs (as reported by Mandiant and Symantec)?
 - A DNS query is represented by an inet:dns:request node.
 - FQDNs reported by Mandiant and Symantec are tagged rep.mandiant.apt1 and rep.symantec.commentcrew respectively.
- How many of these files are unreported (e.g., not part of internal Vertex or public reporting)?

"Show me all the FQDNs that are queried by <u>any</u> APT1 or Comment Crew files."

- How many FQDNs are queried by **any** of the APT1 / Comment Crew files reported by Mandiant and Symantec?
- How many of these FQDNs were **not** publicly reported by a security organization (such as Mandiant or Symantec e.g., rep.symantec) or associated with a threat cluster by Vertex (cno.threat)?

Question 27 - Hints Question 27 - Answer

Question 28

Synapse can record both detection signatures (AV signatures, YARA rules, Snort signatures, other generic detection rules) and "matches" or hits between those signatures and the detected object (such as a file).

"Show me the antivirus / antimalware signatures that reference the malware name 'barkiofork'."

- How many antivirus / antimalware signature names in Synapse reference the malware name 'barkiofork'?
 - An antivirus signature name is represented by an it:av:signame node.
- How many unique scanners / vendors have a 'barkiofork'-related signature?
 - A signature name (it:av:signame) is associated with one or more scan results (it:av:scan:result nodes).
 - An it:av:scan:result has an associated scanner (:scanner:name) that represents a scanning engine / scanning software.
 - An it:prod:softname node represents the name of a software package or product.
- How many files are detected by any of the 'barkiofork' signatures?
 - An it:av:scan:result node represents the verdict (result) of a file scanned by a particular scanner.
 - Note that a verdict can represent a variety of outcomes (e.g., malicious, suspicious, unknown, etc.).
- Which malware families are represented by the files detected by these signatures?
 - An association with a malware family is commonly represented by a tag on a node.
 - A tag representing a malware family may be associated with a tool (risk:tool:software node, via the node's:tag property).

Question 28 - Hints
Question 28 - Answer

Analytical Challenge 5

In <u>Question 28</u>, you identified files associated with multiple malware families (according to Mandiant) that were detected by various AV signatures containing the name 'barkiofork'.

- Based on the data available, do the 'barkiofork' signatures detect / correspond to a specific malware family?
- If an unidentified file was detected by a 'barkiofork' signature, would it be enough to convince you to classify that file as part of a particular malware family? Why or why not?
- If not, what additional information might change your mind?

Analytical Challenge 5 Notes

Question 29

Because Synapse records detection information, you can also ask about and identify files that "have" detection as well as files that have low (or no) detection.

"Show me the APT1 files that have any "<u>malicious</u>" verdicts from antivirus / antimalware detection data."

- How many APT1 files in Synapse have antivirus / antimalware detection data associated with them?
 - An it:av:scan:result node represents a file that has been scanned by a detection engine / product.
 - The it:av:scan:result:verdict property contains the vendor's assessment
 / verdict for the scanned file.
 - **Note** that not all verdicts are "malicious".

"Show me the APT1 files that are detected as malicious by <u>ten or fewer</u> antimalware vendors."

• How many APT1 files in Synapse are detected by ten or fewer antimalware vendors (i.e., have low detection rates, possibly including NO detection)?

"Show me the APT1 files that are detected as malicious by <u>ten or fewer</u> antimalware vendors, but <u>at least two</u> antimalware vendors."

 How many files are detected by ten or fewer vendors, but by at least one vendor (i.e., have low detection rates, but have SOME detection)?

Question 29 - Hints
Question 29 - Answer

Question 30

While analysts typically focus on finding "interesting" data - malicious indicators or evidence of threat activity - it can be equally important to identify those things that can safely be ignored. Identifying common indicators or behavior (and annotating or tagging them in Synapse) can save time in future investigations by providing the context that a particular indicator is innocuous or otherwise "safe" in most circumstances.

In <u>Question 19</u> you identified, based on sandbox execution data, that the FQDN tom-pc was queried by one or more APT1 files. This FQDN is unusual and does not seem to be a "normal" C2 FQDN. How can we find out more about these queries?

"Show me the files that make DNS queries for the FQDN 'tom-pc'."

- How many files in Synapse query this FQDN?
 - An instance of a DNS query is represented by an inet:dns:request node.
- What malware family / families are the files associated with, based on annotations in Synapse?
- How many hosts are associated with the DNS requests for FQDN 'tom-pc'?
 - A system or device is represented by an it:host node. (This includes virtual hosts or notional hosts, such as a malware execution sandbox). The host / sandbox associated with an observed action (such as an inet:dns:request) is stored in that node's :host property (inet:dns:request:host).
- What sandbox(es) are associated with the hosts?
 - Information about a host is stored in its :desc property (it:host:desc).

Question 30 - Hints
Question 30 - Answer

Analytical Challenge 6

In <u>Question 19</u> you identified that, based on sandbox execution data, the unusual FQDN tom-pc was queried by one or more APT1 files. In <u>Question 30</u>, you learned a bit more about similar queries for the same FQDN.

Synapse allows you to dig further into things like:

- Whether the query was an anomaly seen only once, or something observed on multiple occasions.
- Whether the queries have anything in common that might point to a source or cause.
- Whether the queries are meaningful or something that can potentially be ignored for future analysis.

Based on the available data...

- Does it appear that DNS queries for FQDN tom-pc are due to malware behavior or a sandbox artifact? What makes you think so?
- If you assess that the queries are a sandbox artifact, how might you annotate this in Synapse for future reference?

Analytical Challenge 6 Notes

Question 31

The term "decoy document" is used to refer to a non-malicious document that may be dropped (extracted) and opened on the victim's system by a malicious file to disguise the fact that the system is being infected in the background. The non-malicious decoy is displayed to the victim to make them believe they simply opened a legitimate file.

The names of decoy documents can provide insight into the possible motives and targets of the threat actors behind the documents.

"Show me the full file paths for all the files that are added during the execution of any APT1 files."

- How many unique file paths are created from **file add** operations when any APT1 files are executed?
 - An it:exec:file:add node represents a file add operation.
 - o A file:path node represents a file path.
- How many unique file names are associated with the file paths?
 - A file:base node represents the base / final portion of a file:path (typically a file name).
- How many unique file names are there for Word documents and PDF files?
 - A file's extension (e.g., doc or pdf) is stored in the file's :ext property (file:base:ext).
- How many unique file names are there for Word documents and PDF files based on file write operations (vs. file add)?
 - An it:exec:file:write node represents a file write operation.

Tip: Files that are created during execution may be represented by either **file add** (it:exec:file:add) or **file write** (it:exec:file:write) nodes. Which form is used depends on a number of factors, including the data provided by a given data source (e.g., VirusTotal vs. Hybrid Analysis vs. Polyswarm), the file system API calls made during execution, and / or the instrumentation of a given sandbox.

Extra credit:

• Can you write a Storm query to ask about the files produced by **both** it:exec:file:add and it:exec:file:write operations at the same time?

Question 31 - Hints
Question 31 - Answer

BONUS: Question 32

'Uglygorilla' was a persona that Mandiant reported as associated with APT1. He was <u>identified</u> as Wang Dong, one of the five individuals <u>indicted</u> by the U.S. Department of Justice in 2014. Based on the information published by Mandiant that has been recorded in Synapse, what was uglygorilla's tie to APT1?

Hint: Try starting with his username / handle:
 inet:user=uglygorilla

Question 32 - Hints Question 32 - Answer

Hints

Basic Challenges

Question 1 - Hints

- Use the Storm query bar in **Lookup mode** to easily ask about basic indicators.
- Use the query bar in <u>Storm mode</u> to ask about the hash using your first Storm query!
- Select any object (node) in the <u>Results Panel</u> to view additional information about the object in the <u>Details Panel</u>.
- Use the <u>Explore button</u> next to any result to navigate data in the UI and view adjacent or "connected" nodes.

Return to <u>Question 1</u>
Go to <u>Question 1 - Answer</u>

Question 2 - Hints

- Use the Storm query bar in **Lookup mode** to ask about the FQDN, or in **Storm mode** to create a basic Storm query to **lift** the inet:fqdn node.
- What additional information can you find for this FQDN using the **Explore button**?

Return to <u>Question 2</u>
Go to <u>Question 2 - Answer</u>

Question 3 - Hints

- Use the Storm query bar in **Lookup mode** to ask about the email address, or in **Storm mode** to create a basic Storm query to lift the inet:email node.
- What additional information can you find for this email address using the **Explore** button?
- How can you view the tags associated with your results?

Return to <u>Question 3</u>
Go to <u>Question 3 - Answer</u>

Question 4 - Hints

- How can you use the query bar in **Storm mode** to <u>lift by tag</u> to show all the indicators with the tag #rep.mandiant.apt1?
- How can you use the Storm <u>count</u> command to obtain the total number of indicators?
- Is there a Research Tool <u>display mode</u> that would make it easier to see which form had the highest number of indicators reported?

Return to <u>Question 4</u>
Go to <u>Question 4 - Answer</u>

Question 5 - Hints

- How can you use the query bar in Storm mode to <u>lift forms by tag</u> to lift only the MD5 hashes (hash:md5) associated with APT1 (#rep.mandiant.apt1)?
- Is there a Research Tool **display mode** you can use that would make it easier to see which malware family tag was applied to the most MD5 hashes?
 - You may need to interact with the results (e.g., click or drill down) to get the final answer.

Return to <u>Question 5</u>
Go to <u>Question 5 - Answer</u>

Intermediate Challenges

Question 6 - Hints

- How can you use the query bar in **Storm mode** to <u>lift by tag</u> to retrieve the Mandiant APT1 indicators?
- How can you use a <u>filter</u> to only **include** indicators that Symantec also attributed to Comment Crew (rep.symantec.commentcrew)?

Return to <u>Question 6</u>
Go to <u>Question 6 - Answer</u>

Question 7 - Hints

- How can you use the query bar in **Storm mode** to <u>lift by tag</u> to retrieve the Comment Crew indicators (rep.symantec.commentcrew)?
- How can you use a <u>filter</u> to **exclude** indicators also reported by Mandiant (rep.mandiant.apt1)?

Return to <u>Question 7</u>
Go to <u>Question 7 - Answer</u>

Question 8 - Hints

- How can you use the query bar in **Storm mode** to <u>lift by tag</u> to retrieve Symantec's Comment Crew indicators (rep.symantec.commentcrew)?
- How can you use a <u>filter</u> to only **include** indicators that Crowdstrike also attributed to Putter Panda (rep.crowdstrike.putterpanda)?

Return to <u>Question 8</u>
Go to <u>Question 8 - Answer</u>

Question 9 - Hints

- How can you use the guery bar in **Storm mode** to <u>lift</u> the APT1 MD5 hashes?
- How can you **pivot** (or navigate) from the MD5 hashes to their associated files?

Return to <u>Question 9</u>
Go to <u>Question 9 - Answer</u>

Question 10 - Hints

- How can you use the query bar in **Storm mode** to <u>lift</u> the APT1 FQDNs reported by Mandiant (rep.mandiant.apt1)?
- Is there a Research Tool **display mode** that would more easily show you the top six hostnames?
 - You may need to modify the display to get your answer.
- Once you have lifted the APT1 FQDNs, how can you use Storm to filter the results to display only those with the hostname 'e'?

Return to <u>Question 10</u>
Go to <u>Question 10 - Answer</u>

Question 11 - Hints

- How can you <u>lift</u> the FQDN jobsadvanced.com and then <u>pivot</u> (or navigate) to its DNS A records?
 - o Or, how can you **lift** the inet:dns:a nodes for the FQDN directly?
- How can you **pivot** (or navigate) from the DNS A nodes to the associated IPv4 addresses?
- Looking at the resulting IPv4 nodes, what tag(s) does Vertex use to identify parking infrastructure?
- How can you identify the IPv4(s) that represent parking infrastructure?
- Looking at the resulting IPv4 nodes, what tag(s) does Vertex use to identify indicators associated with internally created threat clusters / threat groups?
- How can you identify the IPv4(s) associated with Vertex threat clusters / threat groups?

Tip: Select an individual node in the Results Panel to view its tags in the Details Panel (including any extended tag properties, such as timestamps). You can also:

- Add the timestamps associated with a tag to the Results Panel.
- <u>Display selected tags in the Results Panel</u> (use * to display all tags)

Return to <u>Question 11</u>
Go to <u>Question 11</u> - Answer

Question 12 - Hints

- How can you use the query bar in **Storm mode** to <u>lift</u> all of the FQDNs that are part of the the jobsadvanced.com DNS **zone?**
- How can you <u>pivot</u> (or navigate) from those FQDNs to their DNS A records and then to their associated IPv4s?
- Do you need to **deduplicate** any results? How can you use the Storm <u>uniq</u> command to do this?
- Is there a Research Tool **display mode** that would more easily allow you to identify the most used Autonomous System and / or the country with the most IPv4s?

Return to <u>Question 12</u>
Go to <u>Question 12 - Answer</u>

Question 13 - Hints

- How can you use the query bar in **Storm mode** to **lift** the APT1 FQDNs?
- Is there a Research Tool **display mode** that might make it easier to get a count of the FQDNs that are zones?
- Alternatively, can you use Storm to <u>filter</u> your results to only show FQDNs that are zones?

Return to <u>Question 13</u>
Go to <u>Question 13 - Answer</u>

Question 14 - Hints

Note: You'll need to use Storm to answer this question. There's not an easy way to get from your initial query to the answer using UI navigation.

- How can you use the query bar in **Storm mode** to **lift** the APT1 FQDNs?
- How can you <u>pivot</u> from the :zone property of the FQDNs (i.e., (inet:fqdn:zone) to the FQDNs that represent those zones?
 - You'll need to use <u>explicit pivot syntax</u> in Storm to specify the property you want to pivot **from** on your source nodes.
- Do you need to **deduplicate** any results? How can you use the Storm <u>uniq</u> command to do this?

Return to <u>Question 14</u>
Go to <u>Question 14</u> - <u>Answer</u>

Question 15 - Hints

- How can you <u>lift</u> the FQDN and <u>pivot</u> (or navigate) to the FQDN's whois records?
 - Or, how can you use the query bar in **Storm mode** to **lift** the whois records directly?
- Can you use the Storm <u>min</u> or <u>max</u> commands to help answer some of these questions?
- View all of the tags present on all of the whois records for this FQDN. Which tags indicate that a whois record is associated with sinkhole activity?

Return to <u>Question 15</u>
Go to <u>Question 15 - Answer</u>

Question 16 - Hints

- How can you <u>lift</u> the FQDN and <u>pivot</u> (or navigate) to the inet:whois:email nodes associated with the FQDN?
 - For Storm users, the <u>pivot and join</u> operator can be used to display both the source and target nodes of a pivot at the same time, which may be helpful for this question.
- Once you have retrieved the inet:whois:email nodes, can you identify the email address that is most likely the registrant email?

- The . seen universal property on an inet:whois:email node is used to show when the FQDN / email association was "seen" (i.e., the time period during which the email was known to be associated with the FQDN's registration data).
- Can you use a new query to retrieve any / all FQDNs registered with the same email address?

Return to <u>Question 16</u>
Go to <u>Question 16 - Answer</u>

Question 17 - Hints

- How can you <u>lift</u> the email address and <u>pivot</u> (or navigate) to its associated inet:whois:email nodes?
- Once you have identified the associated FQDN(s), how can you retrieve their whois records?
- An inet:whois:rec node does not have an :email property we can pivot to or filter on to retrieve only those records that have email address issn.bgkit@yahoo.com. Can you use **Storm** to <u>filter by regular expression</u> to return those records whose :text property contains the email address?
- Once you identify the registrant name associated with those records, how can you **pivot** (or navigate) to other whois records with the same :registrant property value?

Return to <u>Question 17</u>
Go to <u>Question 17 - Answer</u>

Question 18 - Hints

- How can you <u>lift</u> the MD5 hash and <u>pivot</u> (or navigate) from the hash to its associated file?
- How can you view the tags on the file?
- How can you **pivot** (or navigate) from a file to any DNS requests made by the file?
- How can you **pivot** (or navigate) from the DNS requests to the FQDN(s) associated with the requests?
- How can you use Storm to <u>filter</u> your results to exclude FQDNs that were already reported by Mandiant (rep.mandiant)?

Return to <u>Question 18</u>
Go to <u>Question 18 - Answer</u>

Question 19 - Hints

- How can you <u>lift</u> the FQDN and <u>pivot</u> (or navigate) to any associated DNS requests?
- How can you **pivot** (or navigate) from the DNS requests to any files that make those requests?
- How can you lift all of the FQDNs in a specific zone?
- How can you pivot (or navigate) from a set of FQDNs to any associated DNS requests, and then pivot (or navigate) to any files that make those requests?
- How can you determine who reported on a file?
- How can you **pivot** (or navigate) from a set of files to any DNS requests made by those files, and then **pivot** (or navigate) to the FQDNs queried?

Return to <u>Question 19</u>
Go to <u>Question 19 - Answer</u>

Question 20 - Hints

- How can you <u>lift</u> the files (file:bytes) associated with APT1 (#rep.mandiant.apt1)?
- How can you <u>filter</u> the APT1 files to only display those that have PDB paths (file:bytes:mime:pe:pdbpath property)?
- How can you <u>pivot</u> (or navigate) from files with PDB paths to the unique set of file paths (file:path nodes) representing those paths?
- How can you <u>filter</u> your results to only show file paths **not** reported by Mandiant?
- How can you **pivot** to other files with **any** of the same PDB paths (whether the PDB path was reported by Mandiant or not)?
- How can you leverage tags in Synapse to answer questions about who reported (or did not report) various indicators?

Return to <u>Question 20</u> Go to <u>Question 20 - Answer</u>

Question 21 - Hints

- How can you <u>lift</u> the APT1 files?
- How can you <u>pivot</u> (or navigate) to the crypto:x509:signedfile nodes that indicate which files are signed?
- How can you **pivot** (or navigate) from the "signed file" nodes to only those files that are signed?
- How can you pivot (or navigate) from the "signed file" nodes to the unique certificates used to sign the files?
- How can you lift a certificate (crypto:x509:cert) node?
- How can you **pivot** (or navigate) from a certificate to its "signed file" nodes, and then to the signed files?
- How can you **view all tags** on a set of nodes (or **filter by tag**) to determine who has (or has not) reported on a given indicator?

Return to <u>Question 21</u> Go to <u>Question 21 - Answer</u>

Question 22 - Hints

- How can you <u>lift</u> the FQDN update8.firefoxupdata.com?
- How can you <u>traverse</u> (or navigate) the 'refs' light edges from the FQDN to the articles that "reference" the FQDN?
 - Note: if using Storm, keep in mind that light edges have a particular "direction", though you can traverse them from either end.

Return to <u>Question 22</u> Go to <u>Question 22 - Answer</u>

Question 23 - Hints

- How can you use Storm to <u>lift</u> the APT1 indicators?
- How can you identify any Vertex threat clusters that overlap with the APT1 indicators, based on annotations in Synapse?

Return to <u>Question 23</u> Go to <u>Question 23 - Answer</u>

Question 24 - Hints

- How can you <u>lift</u> the indicators owned or controlled by T15?
- How can you **lift** the "seed" node(s) for T15?
- How can you **lift** the indicators used by T15?
- How can you lift all indicators associated with T15?
- Is there a Research tool **display mode** that would help you determine whether T15 and APT1 are coherent / fully connected clusters?

Return to <u>Question 24</u>
Go to <u>Question 24 - Answer</u>

Question 25 - Hints

- How can you **lift** the domain whois records associated with T15?
- How can you identify the earliest registration date for the whois records? Is there a
 Storm command that can simplify this?
- Can you use Storm to <u>lift by time or interval</u> to find other whois records in Synapse that were registered on the same date?
 - Additional detail (if needed) is available on working with <u>time</u> and <u>intervals</u> in Synapse.
- How can you determine which threat groups (#rep or #cno) an FQDN is associated with?

Return to <u>Question 25</u> Go to <u>Question 25 - Answer</u>

Advanced Challenges

Question 26 - Hints

- How can you <u>lift</u> the node (file:mime:pe:vsvers:keyval) representing the PE VERSIONINFO data you want to ask about?
 - **Note:** the values are strings and are **case-sensitive.**
- How can you <u>pivot</u> (or navigate) from the VERSIONINFO to nodes representing files that have this metadata (file:mime:pe:vsvers:info)?
- How can you **pivot** (or navigate) from those nodes to the associated files?
- How can you view (or <u>filter</u> on) the files to identify any files that are **not** associated with any reporting?

Return to <u>Question 26</u>
Go to <u>Question 26 - Answer</u>

Question 27 - Hints

Tip: You will need to use Storm for this challenge; the number of nodes involved in the queries make it infeasible to answer using UI navigation.

- How can you lift the APT1 FQDNs reported by Mandiant and the Comment Crew FQDNs reported by Symantec?
- How can you remove any duplicates from your results?
- How can you **pivot** from the FQDNs to any DNS requests for the FQDNs?
- How can you **pivot** from the DNS requests to the files that make the requests?
- How can you remove any duplicates from your results?
- How can you **filter** your results to exclude files reported by Vertex (cno.*) or by other third-party organizations?
- How can you **lift** the APT1 files reported by Mandiant and the Comment Crew files reported by Symantec?
- How can you remove any duplicates from your results?
- How can you pivot from the files to any DNS requests made by the files?
- How can you **pivot** from the DNS requests to the FQDNs queried?
- How can you remove any duplicates from your results?

 How can you filter your results to exclude files reported by Vertex (cno.*) or by other third-party organizations?

Return to <u>Question 27</u>
Go to <u>Question 27 - Answer</u>

Question 28 - Hints

- How can you <u>lift by regular expression</u> to find the antivirus / malware signature names that contain the string 'barkiofork'?
- How can you <u>pivot</u> (or navigate) from the signature names to the scan results from various scanners / vendors?
- How can you **pivot** (or navigate) from the scan results to the software / engine names associated with the results?
- How can you remove any duplicates from your results?
- How can you return to your scan results and **pivot** (or navigate) from the results to the files detected by the scans?
- How can you remove any duplicates from your results?
- How can you view your results to identify any associated malware families?

Return to <u>Question 28</u>
Go to <u>Question 28 - Answer</u>

Question 29 - Hints

Tip: You will need to use Storm for this challenge; the UI does not give you the flexibility necessary to answer this question.

- How can you <u>lift</u> the APT1 files?
- How can you use a <u>subquery filter</u> to show only those files that have associated antivirus / antimalware data (it:av:scan:result) with a malicious verdict?
- How can you use a **subquery filter** with a mathematical operator to show only those files with ten or fewer associated malicious verdicts?

 How can you use two subquery filters with mathematical operators to show only those files with ten or fewer associated malicious verdicts but at least two or more malicious verdicts?

Return to <u>Question 29</u> Go to <u>Question 29 - Answer</u>

Question 30 - Hints

- How can you <u>lift</u> the FQDN tom-pc?
- How can you **pivot** (or navigate) from the FQDN to the associated DNS queries?
- How can you **pivot** (or navigate) from the DNS queries to the files that make those queries?
- How can you view tags that identify any malware families associated with the files?
- How can you **pivot** (or navigate) from the DNS queries to the hosts where those queries originated?

Return to <u>Question 30</u> Go to <u>Question 30 - Answer</u>

Question 31 - Hints

Tip: You will need to use Storm for this challenge; the number of nodes involved in the queries make it impractical to answer using UI navigation.

- How can you <u>lift</u> the APT1 files?
- How can you <u>pivot</u> from the files to the "file add" events that occur when the files are executed?
- How can you pivot from the "file add" events to the full file paths for the files written?
- How can you remove duplicate results?
- How can you pivot from the file paths to their associated file names?
- How can you use a <u>compound filter</u> to limit your results to only those file names with 'doc' or 'pdf' extensions?
- How can you remove duplicate results?

 How could you write a Storm query similar to the one above, but for "file write" operations (it:exec:file:write)?

Extra credit:

• Can you use the Storm <u>tee</u> command to execute two **pivot out** operations concurrently to go from the APT1 files (file:bytes) to the it:exec:file:write **and** it:exec:file:add nodes?

Return to <u>Question 31</u>
Go to <u>Question 31</u> - Answer

Question 32 - Hints

- Starting with the username **uglygorilla** (inet:user=uglygorilla), try using the **Explore button** in Optic to see what nodes that username is connected to (and what those connected nodes are connected to....?)
- As you start to identify adjacent (and "related") nodes, it may help to track "nodes of interest" by applying a tag to help "draw a box", so to speak, around your research.
 This could be a "temporary" / unofficial tag such as #mytag.story.uglygorilla (whatever you choose to make up!).
- Does viewing the nodes in other display modes help identify connections? (**Note:** if you tag all the "nodes of interest" you can simply query (lift) all those nodes by whatever tag you choose.)
- You may wish to refer to the main <u>APT1 report</u> (p.51-55) for Mandiant's discussion of "Ugly Gorilla".

Note: This research is open ended, so there is no "right" way to solve this challenge!

Return to <u>Question 32</u>
Go to <u>Question 32 - Answer</u>

Answer Key

Each question has its own Answer section; each Answer is divided into subsections:

- **Answer:** just the answer, with minimal additional explanation. Use this section to simply check your work and move on to the next challenge!
- Answer Explanations: a more thorough explanation or breakdown of how to get
 the answer, or why some approaches are better than others. Use this section if
 you're not quite sure how we got the answer we did, or if you want more
 information on how Synapse and Storm allow you to answer the challenge
 question.

Each Answer Explanations section includes:

- **UI Answer:** How to solve the challenge using Synapse's UI navigation.
- Storm Answer: How to solve the challenge using Storm.

Some challenges are impractical to solve using the UI alone; specifically, using the UI's **Explore** button and / or **query** context menu are less effective when working with large numbers of nodes. This is where Storm is handy because Storm allows you to more narrowly focus your query and your results.

Realistically, most analysts use a combination of Storm and UI features in their workflow, so separating answers into "just the UI" or "just Storm" is a bit artificial. But hopefully this gives you an idea of the power and flexibility of Synapse, and which modes are most useful for answering certain types of questions.

The Answer Explanations section may also include **Bonus Notes** - additional notes that dive deeper into some aspect of the challenge question. These tend to go "above and beyond" by diving into some of the technical details of Synapse and Storm. Use this section if you want to geek out with us on a topic, but the Bonus Notes aren't necessary to answer the challenge question.

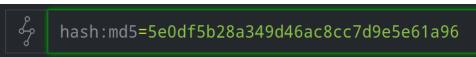
Basic Challenges

Question 1 - Answer

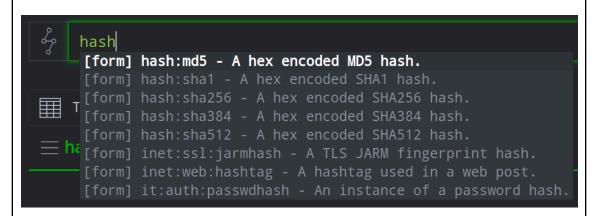
Using the Storm query bar in <u>Lookup mode</u>, you can ask about (<u>lift</u>) any **basic**indicator by pasting the raw indicator into the query bar:



• Using the Storm query bar in <u>Storm mode</u>, you can lift **any object** by specifying the name of the object (**form** - in this case, hash:md5) and the **value** of the specific object you're interested in (in this case, 5e0df5b28a349d46ac8cc7d9e5e61a96):



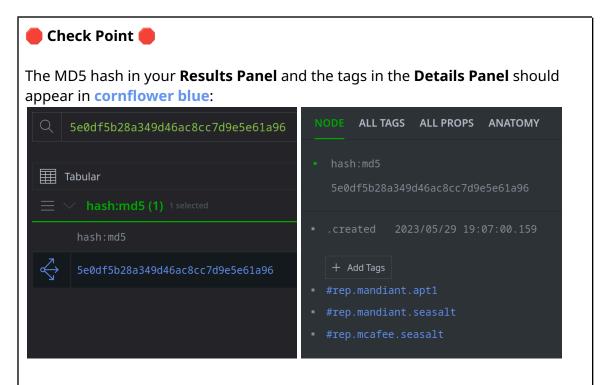
Tip: As you type in the query bar in **Storm mode**, Synapse will suggest matching values (forms, properties, tags, commands, etc.) to allow you to select/autocomplete your text.



Autocomplete is on by default, but can be turned off by toggling the **autocomplete activate on typing** switches on the **PREFERENCES** tab under **Your Settings**.

- The MD5 hash has the following tags:
 - rep.mandiant.apt1
 - o rep.mandiant.seasalt
 - rep.mcafee.seasalt

- The tags indicate that:
 - Mandiant associates the hash with SEASALT and APT1.
 - McAfee associates the hash with SEASALT.



If your results do not look like the images above, go to the <u>Before You Begin</u> section of this document to make sure you have selected the correct **Workspace** and **View** for the Scavenger Hunt.

- Yes, the hash has an associated file (file:bytes node) in Synapse.
- The hash is referenced in two articles (media:news nodes).

Return to <u>Question 1</u>
Go to <u>Question 1 - Answer Explanations</u>

Question 2 - Answer

• You can ask about (lift) a basic indicator by:

Pasting any basic indicator into the query bar in Lookup mode:



Lifting the node using Storm using the query bar in Storm mode:

• There are **twelve** DNS A records associated with the FQDN.

Return to <u>Question 2</u>
Go to <u>Question 2 - Answer Explanations</u>

Question 3 - Answer

- You can ask about (**lift**) a basic indicator by:
 - o Pasting the indicator into the query bar in **Lookup mode:**



• **Lifting** the node using Storm with the query bar in **Storm mode:**

```
inet:email=lfengg@163.com
```

 There are eight inet:whois:email records associated with eight FQDNs registered with this email address. • All eight FQDNs were reported by Mandiant as associated with APT1.

Return to <u>Question 3</u>
Go to <u>Question 3 - Answer Explanations</u>

Question 4 - Answer

• In the **Research Tool**, use the query bar in **Storm Mode** to **lift** all nodes with the rep.mandiant.apt1 tag:



- **5,749** nodes are tagged rep.mandiant.apt1.
- FQDNs (inet:fqdn) were the most reported indicator, with 2,073 reported / tagged.

Return to <u>Question 4</u>
Go to <u>Question 4 - Answer Explanations</u>

Go to **Question 4 - Bonus Notes**

Question 5 - Answer

• You can **lift only** the MD5 hashes by specifying the **form** (hash:md5) you want to ask about along with the tag:



• The **GREENCAT** malware family (rep.mandiant.greencat) had the largest number of associated hashes (103).

Return to <u>Question 5</u>
Go to <u>Question 5 - Answer Explanations</u>

Intermediate Challenges

Question 6 - Answer

- **Eleven** indicators were reported by both Mandiant and Symantec.
- The indicators included:
 - Domains / FQDNs (inet:fqdn)
 - File names (file:base)
 - IP addresses (inet:ipv4)

Return to Question 6

Go to Question 6 - Answer Explanations

Question 7 - Answer

- Symantec reported 1,872 Comment Crew indicators that were not reported by Mandiant.
- The indicators included:
 - File names (file:base)
 - Hashes and associated files (hash:md5 / hash:sha1 / hash:sha256 / hash:sha512 / file:bytes)
 - File metadata (file:mime:pe:vsvers:keyval)
 - File paths (file:path)
 - Email metadata (specifically subject lines inet:email:header)
 - Domains / FQDNs (inet:fqdn)
 - IP addresses (inet:ipv4)
 - Registry keys (it:dev:regkey)
 - Strings(it:dev:str)

Return to <u>Question 7</u>

Go to <u>Question 7 - Answer Explanations</u>

Question 8 - Answer

- **Eleven** indicators were reported by Symantec as Comment Crew and Crowdstrike as Putter Panda.
- Reported indicators included:

Hashes and associated files (hash:md5 / hash:sha1 / hash:sha256 / hash:sha512 / file:bytes)

File names (file:base)

Return to Question 8

Go to Question 8 - Answer Explanations

Go to Question 8 - Bonus Notes

Analytical Challenge 1 - Notes

It can be very challenging to determine whether two threat groups with two different names reported by two different organizations are "the same". One factor to consider is any overlap in publicly reported indicators - that is, two differently named groups that have many indicators in common might be "the same" group.

This method is not foolproof:

- It is "generally accepted" that APT1 and Comment Crew are two names for "the same" group, but there are very few overlaps in the indicators reported publicly by Mandiant and Symantec.
- It is "generally accepted" that Comment Crew (aka "APT1") and Putter Panda (aka "APT2") are **not** "the same" group, yet they share a small number of indicators in common, based on public reporting.

There are possible explanations for both discrepancies. For example:

- Organizations reporting on "the same" group may purposely avoid publishing the same indicators so that their reporting does not appear duplicative.
- Organizations may cluster or attribute activity differently, or may have different visibility into threat operations, so may come to different conclusions about the same indicator.

This is one reason we (Vertex) find it helpful to use tags to track different organizations' assertions separately, and to separate third-party assertions (which we have to take at face value) from assertions we make based on our own direct access to data.

What other factors would you consider when deciding if two groups are "the same"?

Return to Analytical Challenge 1

Question 9 - Answer

• **437** MD5 hashes have corresponding files in Synapse (that is, there are 437 files (file:bytes nodes) corresponding to MD5 hashes reported by Mandiant).

Return to <u>Question 9</u>
Go to <u>Question 9 - Answer Explanations</u>
Go to <u>Question 9 - Bonus Notes</u>

Question 10 - Answer

- Mandiant reported 2,073 FQDNs.
- The top six most common hostnames (subdomain names) are:
 - o www
 - news
 - o mail
 - o pop
 - o smtp
 - update
- 17 FQDNs use the hostname 'e'.

Return to <u>Question 10</u>
Go to <u>Question 10 - Answer Explanations</u>
Go to <u>Question 10 - Bonus Notes</u>

Question 11 - Answer

- FQDN jobsadvanced.com has resolved to **nine** unique IPv4 addresses.
- One IPv4 address is associated with parking infrastructure based on tags in Synapse.

- One IPv4 address is associated with threat actor infrastructure (threat cluster T15) based on tags in Synapse.
- The IPv4 was used by T15 **between February 13, 2013 and April 2, 2013** (specifically, 2013/02/13 16:00:29 and 2013/04/02 14:28:06).

Return to <u>Question 11</u>
Go to <u>Question 11 - Answer Explanations</u>

Go to Question 11 - Bonus Notes

Question 12 - Answer

- The FQDN jobsadvanced.com and all of its subdomains have resolved to 14 unique IPv4 addresses.
- AS 40034 hosted the majority of the IPv4s (five).
- The **British Virgin Islands** (VG) and the **United States** (US) both hosted the most IPv4s (tied at five each).

Return to **Question 12**

Go to Question 12 - Answer Explanations

Go to Question 12 - Bonus Notes

Question 13 - Answer

117 of the FQDNs reported by Mandiant are zones.

Return to <u>Question 13</u>
Go to <u>Question 13 - Answer Explanations</u>

Question 14 - Answer

• There are 149 zones associated with all of the FQDNs reported by Mandiant.

Return to **Question 14**

Go to Question 14 - Answer Explanations

Go to Question 14 - Bonus Notes

Analytical Challenge 2 Notes

While we can't know why Mandiant reported some indicators but not others, we can look at the existing data in Synapse and form some hypotheses as to why:

- It could be simple oversight; Mandiant reported several thousand indicators so it is certainly possible that a few were overlooked.
- Some APT1 malware used legitimate services (such Google Docs) for command and control (C2). This means that the malware performs a DNS query for a "legitimate" FQDN (such as docs.google.com). The ("legitimate") FQDN is in fact associated with malware (GDOCUPLOAD) and with APT1. But the zone google.com is not associated with either.
- One of APT1's well-known techniques was to compromise legitimate web sites and add web pages (or modify existing pages) to contain hidden HTML comments that provided C2 instructions to malware. Similar to the Google Docs example, the malware performs a DNS query for a valid FQDN (such as www.woodagency.com) associated with malware, but the zone itself (woodagency.com) is not used by the malware and is not registered or controlled by APT1.

We can easily explain these nuances in prose ("The malware communicates with the compromised web site www.woodagency.com") but "how" we distinguish a "malicious" FQDN, a "compromised" FQDN, and a "legitimate FQDN used maliciously" in Synapse may take a bit more thought. You may be able to tell the difference by looking at "nearby" data. But in many cases you want that context directly associated with the FQDN itself (e.g., as tags). If these distinctions are important based on how you use Synapse (as an intelligence repository and analysis system? To integrate with detection capabilities?), take that into consideration when creating tag trees and deciding "how" to tag data.

Return to Analytical Challenge 2

Question 15 - Answer

- There are 26 whois records (inet:whois:rec) for FQDN youipcam.com.
- The earliest registration date (:created) recorded in Synapse for this FQDN is **April** 27, 2011 (2011/04/27).
- The most recent capture date (:asof) for registration data for this FQDN was October 4, 2018 (2018/10/04).

The FQDN was sinkholed by Kleissner & Associates on July 16, 2016 (2016/07/16).

Return to <u>Question 15</u>
Go to <u>Question 15 - Answer Explanations</u>
Go to <u>Question 15 - Bonus Notes</u>

Question 16 - Answer

- There are eight email addresses associated with domain whois data for FQDN youipcam.com.
- Email address **jfeng3554@hotmail.com** was the registrant email associated with the FQDN during the time APT1 likely controlled the domain.
- That email address was also used to register FQDNs syscation.com and syscation.net.

Return to <u>Question 16</u>
Go to <u>Question 16 - Answer Explanations</u>

Question 17 - Answer

- The email address was used to register one FQDN (progammerli.com).
- There are 75 whois records (inet:whois:rec) for this FQDN.
- There are 42 whois records that contain the email address in the raw text (:text property).
- The registrant name "william michael" was used along with the email address to register the FQDN.
- The same registrant name was also used to register the FQDN comrepair.net.

Return to <u>Question 17</u>
Go to <u>Question 17 - Answer Explanations</u>

Analytical Challenge 3 Notes

Determining whether two sets of activity or two indicators are "related" is an analytical assessment. Every organization (and potentially every analyst!) will have a slightly different standard for making this decision.

One consideration may be the "uniqueness" of an indicator. A registrant name like "John Smith" is a very common English name, so may be more likely to occur independently in multiple unrelated records. The use of unusual names or unusual spelling may be "more unique" and possibly "more likely" to be related. (That said, "unique" indicators that are publicly known or visible can be easily imitated for use as a simple false flag.)

Depending on the data or indicators you are comparing, you might weigh additional characteristics. In the case of domain whois records, are there other features or similarities that might help link the two? Do the records use the same domain registrar or name servers? Were they registered on the same date or close together in time?

Another consideration is any additional data (unrelated to the domain whois records themselves) that you can link to the FQDNs that might tie them together. Examples include:

- shared network infrastructure
 - the FQDNs or their subdomains resolve to the same IP address within the same time frame; or
 - the FQDNs resolve to different IP addresses that share other characteristics,
 like an SSL/TLS certificate or a particular set of software / services.
- both domains (or associated subdomains) are used in the same activity
 - o for example, one FQDN used as part of a phishing email that delivers malware and the other FQDN used as C2 for that malware.

Return to Analytical Challenge 3

Question 18 - Answer

- The hash is associated with
 - file:bytes=sha256:9ec9221f685b446874bb6dfc5509b4304f8d8b78b10fa3b8ba06cf4f505c0f84.
- According to Mandiant, the file is part of the TARSIP_ECLIPSE malware family (rep.mandiant.tarsip_eclipse).
- The file makes DNS queries for the FQDNs un.linuxd.org and tom-pc.
- The FQDNs were **not** reported by Mandiant.
- Symantec reported un.linuxd.org as a Comment Crew indicator (rep.symantec.commentcrew).

Return to <u>Question 18</u>
Go to <u>Question 18 - Answer Explanations</u>
Go to <u>Question 18 - Bonus Notes</u>

Question 19 - Answer

- There are **no** files that query the FQDN earthsolution.org.
- There are **five** files that query various subdomains of earthsolution.org.
- **Three** of the files were reported by Mandiant.
- The files query the following subdomains of earthsolution.org:
 - ctcs.earthsolution.org
 - moto2.earthsolution.org
 - vop.earthsolution.org
- The files also query the following FQDNs (i.e., these FQDNs were queried when the files were executed in a malware sandbox / sandboxes):
 - o ctcs.bigdepression.net
 - hdredirect-lb7-5a03e1c2772e1c9c.elb.us-east-1.amazonaws.com
 - motoa.purpledaily.com
 - time.windows.com
 - o tom-pc

Return to Question 19

Go to Question 19 - Answer Explanations

Question 20 - Answer

- **37** APT1 files have PDB paths.
- There are **25 unique** PDB paths.
- **Eleven** PDB paths were **not** reported by Mandiant.
- A total of **131** files have one of the 25 unique PDB paths.
- **94** of those files were **not** reported by Mandiant.
- **56** files that were **not** reported by Mandiant **were** reported by Symantec (specifically associated with Comment Crew).
- 72 files that were not reported by Mandiant were reported by Symantec in some capacity (as associated with Comment Crew, Nitro, or Poison Ivy).

Return to Question 20

Go to Question 20 - Answer Explanations
Go to Question 20 - Bonus Notes

Question 21 - Answer

- **Ten** APT1 files were signed with (or contain) code-signing certificates.
- **Four** unique x509 certificates were used to sign the ten files.
- **Six** files in Synapse are signed with the "Microsoft" certificate.
 - Four of the files were reported by either Mandiant or Symantec.
 - **Two** files were **not** reported by either organization.
 - The files are **not** associated with any malware families.

Return to Question 21

Go to Question 21 - Answer Explanations

Go to Question 21 - Bonus Notes

Question 22 - Answer

• There are **two** articles that reference the FQDN.

Return to <u>Question 22</u>
Go to <u>Question 22</u> - Answer Explanations

Question 23 - Answer

- **Two** Vertex threat clusters overlap with the APT1 indicators.
- The clusters are **T15** and **T19**.

Return to Question 23

Go to Question 23 - Answer Explanations

Question 24 - Answer

- There are **1,517** nodes tagged as being "owned or controlled" by T15.
- The FQDN aunewsonline.com was the "seed" node for T15.
- There are **seven** nodes tagged as being "used" by T15.

- **Yes,** T15 is a coherent cluster when displayed in **Force Graph** display mode, the nodes associated with T15 are interconnected (i.e., there are no detached clusters or nodes floating off by themselves).
- **No**, APT1 is not a coherent cluster when displayed in **Force Graph** display mode, the nodes associated with APT1 are not all interconnected.

Return to Ouestion 24

Go to Question 24 - Answer Explanations

Go to Question 24 - Bonus Notes

Question 25 - Answer

- There are **494** domain whois records (inet:whois:rec) associated with T15 (#cno.threat.t15.own).
- T15 registered **24** FQDNs.
- The FQDNDomain **yahoodaily.com** was the earliest FQDN registered by T15, on **September 8, 2009** (2009/09/08 01:50:21).
- **Seven** FQDNs in Synapse were registered on that same date.
- The FQDN reutersnewsonline.com was registered on that date and is associated with APT1, but not T15.

Return to Question 25

Go to Question 25 - Answer Explanations

Advanced Challenges

Question 26 - Answer

- 19 files in Synapse have this VERSIONINFO.
- 10 of the files are unreported by Vertex or other third-party organizations.

Return to Question 26

Go to Question 26 - Answer Explanations

Question 27 - Answer

- **182** unique files query one or more FQDNs reported by Mandiant (APT1) or Symantec (Comment Crew).
- **Eleven** of those files are **not** associated with Vertex or public reporting.
- **215** unique FQDNs are queried by one or more files reported by Mandiant (APT1) or Symantec (Comment Crew).
- **94** FQDNs are **not** associated with Vertex or public reporting.
 - Note: This number includes FQDNs that may be incidental queries / sandbox artifacts as well as DNS queries made by malware.

Return to Question 27

Go to Question 27 - Answer Explanations

Question 28 - Answer

- 101 antivirus / antimalware signatures contain the string 'barkiofork'.
- 24 software packages / vendors have 'barkiofork' signatures.
- **40** files in Synapse are detected by these signatures.
- Based on annotations (tags), the files are associated with the following malware families:
 - AURIGA (Mandiant)
 - WARP (Mandiant)

Return to Question 28

Go to Question 28 - Answer Explanations

Question 29 - Answer

- 435 APT1 files have associated antivirus / antimalware detection data.
- **Four** files are detected by ten or fewer vendors.
- **Two** files are detected by ten or fewer vendors but by at least one vendor.

Return to Question 29

Go to Question 29 - Answer Explanations

Go to Question 29 - Bonus Notes

Question 30 - Answer

- 29 unique files make DNS queries for the FQDN tom-pc.
- The files are associated with the following malware families:
 - BISCUIT
 - o SEASALT
 - TABMSGSQL
 - TARSIP-ECLIPSE
 - TARSIP-MOON
 - WEBC2-YAHOO
- **29** unique hosts are associated with the DNS queries.
- All of the hosts have a :desc property value of Rising MOVES.

Return to Question 30

Go to Question 30 - Answer Explanations

Go to Question 30 - Bonus Notes

Analytical Challenge 6 Notes

Assuming that the malware families identified by Mandiant and McAfee are correct, six different malware families make queries for tom-pc. While the queries could indicate that the six families share some legacy source code that makes such an odd query, there could be other, more likely explanations.

Looking at the sandbox (it:host) data for the queries, all of the tom-pc queries were seen by instances of the Rising MOVES sandbox. It is possible that Rising's technology is able to observe queries that other vendors can't; but a more likely explanation is that the queries are a result of something specific to Rising's configuration.

29 queries is not a very large sample size. If you collect additional examples of Rising MOVES execution data, and notice that many (all?) of Rising's sandbox data includes this query, that may convince you that the DNS queries are an artifact that is specific to Rising and unrelated to the malware samples themselves.

Tags in Synapse are commonly used to annotate things we think are "interesting" - threat clusters or malware families, for example. But tags can be used to provide any type of valuable "situational awareness" that an analyst might want to see when they look at a node in Synapse. Sometimes we may want to know that something in Synapse is "not interesting" (along with some idea of why).

One option is to create a tag or tag tree to indicate this. For example, a tag such as cno.common.sandbox could be placed on the inet:dns:request nodes that query tom-pc (and the inet:fqdn=tom-pc node) to indicate that they are sandbox artifacts (and therefore probably safe to ignore). (The tag could even be applied automatically to new sandbox execution data added to Synapse by using a trigger!)

Having this "situational awareness" means that other analysts don't have to repeat the same research to come to the same conclusion, and can instead focus on more valuable analytical tasks!

Return to Analytical Challenge 6

Question 31 - Answer

For **file add** operations:

- There are **270** unique file paths created via **file add** operations during the execution of any / all of the APT1 files.
- Those paths include 138 unique file names.
- There are twelve unique DOC and PDF file names.

For **file write** operations:

- There are 392 unique file paths created via write operations during the execution of any / all of the APT1 files.
- Those paths include **166** unique file names.

• There are **27** unique DOC and PDF file names.

Extra Credit:

• Storm query using **tee** command:

```
file:bytes#rep.mandiant.apt1 | tee {-> it:exec:file:add }
    { -> it:exec:file:write } | -> file:path | uniq
    | -> file:base | uniq | +(:ext=doc or :ext=pdf)
```

There are **28** unique DOC and PDF file names created by file add and file write operations.

```
Return to <u>Question 31</u>
Go to <u>Question 31 - Answer Explanations</u>
Go to <u>Question 31 - Bonus Notes</u>
```

Question 32 - Answer

• The username 'uglygorilla' is associated with the email address uglygorilla@163.com, which was used to register the APT1 FQDN hugesoft.org.

Note: FQDN hugesoft.org is part of APT1 according to Mandiant, as well as part of Vertex's threat cluster T19 (#cno.threat.t19).

- Mandiant linked the email address uglygorilla@163.com to an online account at bbs.chinamil.com.cn. The account used the username "绿野" (translated as "Greenfield") and a "real name" of "JackWang".
- Mandiant identified additional online accounts using the username 'uglygorilla' and / or email address 'uglygorilla@163.com', including forums at rootkit.com and Chinese software development site pudn.com.
- The 'uglygorilla' account at site pudn.com was registered with the purported real name of "汪东" (WANG Dong).
- A year after the release of the APT1 report, WANG Dong (a.k.a. "Jack Wang", a.k.a. "uglygorilla") was one of five individuals <u>indicted by the U.S. Department of Justice</u> in May 2014 on charges related to computer hacking by the Chinese military.

Return to <u>Question 32</u> Go to <u>Question 32 - Answer Explanations</u>

Answer Explanations

This section provides detailed explanations for how each challenge can be solved. Both UI and Storm-based solutions are provided where they exist (keeping in mind that most analysts will leverage a combination of Storm and UI features and shortcuts as part of their workflow).

Basic Challenges

Question 1 - Answer Explanations

- Q1 UI Answer
- Q1 Storm Answer

Q1 UI Answer

To look up the hash:

• In the **Research Tool**, using the query bar in **Lookup mode**, paste the hash into the query bar and press **Enter** to run your query:

5e0df5b28a349d46ac8cc7d9e5e61a96

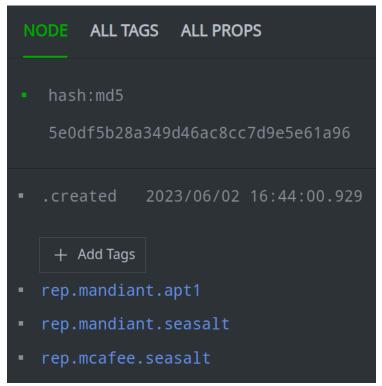
You can use **Lookup mode** to "ask about" **basic indicators** without using the Storm query language:

- Hashes (MD5, SHA1, SHA256)
- Domains (fully qualified domain names, or FQDNs)
- IPv4 addresses
- Email addresses
- o URLs
- CVE numbers (e.g., cve-2017-0145)
- Common cryptocurrency addresses (e.g., Bitcoin, Ethereum)

Lookup mode is a good way to get started with Synapse and is helpful for basic use cases. However, we encourage you to get used to asking questions using Storm! In fact, for later questions we'll assume you know how to do common <u>lift</u> operations with Storm.

To view the tags:

• In the **Results Panel**, select your MD5 hash to view additional information about the hash in the **Details Panel** (including any tags on the hash):



To view the tag definitions:

To obtain the meaning / definition of a tag:

• In the **Details Panel**, hover your mouse over the tag to view a tooltip with the tag definition:

```
APT1 (Mandiant)

Indicator or activity Mandiant calls (or associates with) APT1.

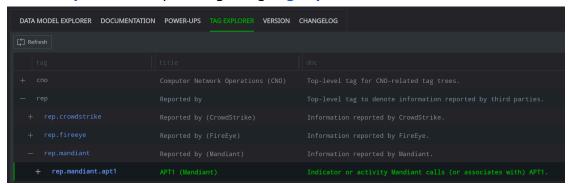
rep.mandiant.apt1

rep.mandiant.seasalt

rep.mcafee.seasalt
```

Or:

In the <u>Help Tool</u> look up the tag using <u>Tag Explorer</u>:



To find an associated file:

After you lift the MD5...

 In the Results Panel, click the Explore button next to the hash:md5 node to navigate to adjacent ("connected") nodes:

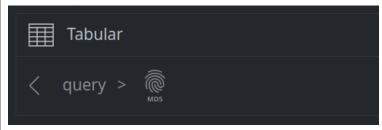


After "exploring", you will see a file: bytes node (if one exists), along with any other "connected" nodes:



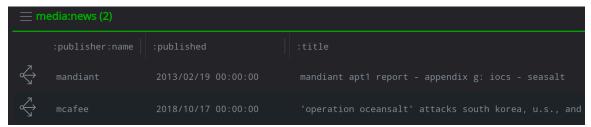
An MD5 hash (hash:md5) is related to ("connected" to) an associated file (file:bytes) via the file's :md5 property (file:bytes:md5).

Tip: When you use the **Explore button,** Synapse starts to leave a trail of "breadcrumbs" to remind you how you got to your current set of results (i.e., "what you clicked" to get here). When you "explore" from the MD5 hash, Synapse adds two breadcrumbs: one representing your original query, followed by a fingerprint / MD5 icon that indicates you explored from a hash:md5 node to get to the current results. You can click any breadcrumb to return to that point in your navigation:



To find any associated articles:

• In your current set of results, in the **Results Panel**, locate the two articles (media:news nodes) that contain (reference) the hash:



Synapse uses 'refs' (for "references") <u>light edges</u> to link an object (node) to a thing that references that node. In this case, the articles (media:news nodes) **reference** the MD5 (hash:md5 node)

Return to **Question 1**

Q1 Storm Answer

Tip: For the first few challenges, we encourage you to use the **Explore button** to get a feel for Synapse's UI navigation. However, we'll still include a way to answer the question using Storm.

Navigation (pivoting or traversing) in Storm may look a bit foreign at first, but will become clearer with practice!

To look up the hash:

• Asking about (lifting) common indicators in **Storm mode** is very similar to using Lookup mode. The difference is that in Lookup mode, Synapse "recognizes" simple indicators like MD5 hashes; in Storm mode you need to "tell" Synapse what kind of object (form) you're looking for:

```
hash:md5=5e0df5b28a349d46ac8cc7d9e5e61a96
```

In Storm you:

- Specify the **form** (type of node) you want to ask about (in this case, hash:md5).
- Specify "how" you want to find or ask about the object. In this case we use the equals (=) sign to tell Synapse we want the MD5 with this exact value.
- Specify the value for the form you're asking about (i.e., the hash 5e0df5b28a349d46ac8cc7d9e5e61a96).

So the Storm query above means you want to **lift** the **form** hash:md5 whose **value** is **equal** to 5e0df5b28a349d46ac8cc7d9e5e61a96.

To view the tags:

Since we're just getting started (and since we're looking at a single node), it's
easiest to view the tags on the node using the **Details Panel** (as described above in
the **Q1 UI Answer**).

But (since you asked) you can also use Storm to ask **directly** about the tags on a node using a specialized pivot operation called **pivot to tags**:

```
hash:md5=5e0df5b28a349d46ac8cc7d9e5e61a96 -> #
```

This is getting a bit ahead of ourselves, but to "pivot to tags":

- Specify your source or starting node(s) (in this case our hash:md5 node).
- Use the **pivot out** operator: ->
 - Since most pivots are "pivot out" operations, the "right arrow" is simply called a "pivot" for short.
- Specify the **target** of your pivot. Use the "tag" (hashtag) symbol (#) for the special use case where you want to pivot from a node to the **tags on the** node.

To find the associated file:

• Since we're just getting started (and since we're looking at a single node), it's easiest to use the **Explore button** to find the "connected" file (as described above in the **Q1 UI Answer**).

You can use Storm to ask **directly** about any associated file(s) using a **pivot out** operation:

```
hash:md5=5e0df5b28a349d46ac8cc7d9e5e61a96 -> file:bytes
```

To pivot in Storm:

- Specify your **source** or starting node(s) (in this case our hash:md5 node).
- Use the **pivot** operator: ->

Specify the **target** of your pivot (in this case, any files - file:bytes nodes - our hash might be "connected" to).

To find associated articles:

Since we're just getting started (and since we're looking at a single node), it's
easiest to use the **Explore button** to find any "connected" articles (as described
above in the **Q1 UI Answer**).

You can use Storm to ask **directly** about any associated articles by **traversing** the light edges that connect the nodes:

```
hash:md5=5e0df5b28a349d46ac8cc7d9e5e61a96 <(refs)- media:news
```

To traverse edges in Storm:

- Specify your source or starting node(s) (in this case our hash:md5 node).
- Use the edge traversal operator (a left or right arrow that contains the "name" of the edge - in this case < (refs) -).
- Specify the **target** of your traversal (in this case, any articles media:news nodes - our hash might be "referenced" by.

Tip: Light edges have a **direction** - articles "reference" indicators, indicators don't "reference" articles. You can navigate (cross or traverse) a light edge "in either direction", depending on your "starting point". Because we're starting our investigation from the MD5 hash, we cross the edge "backwards" to the articles that reference the hash.

Return to Question 1

Question 2 - Answer Explanations

- Q2 UI Answer
- O2 Storm Answer

Q2 UI Answer

To look up the FQDN:

• In the **Research Tool**, using the query bar in **Lookup mode**, paste the indicator into the query bar and press **Enter** to run the query:

```
globalowa.com
```

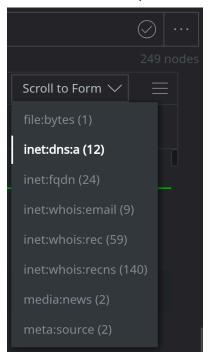
To find the DNS A records:

• In the **Results Panel**, **click** the **Explore button** next to the FQDN to navigate to any adjacent nodes:



Using **Explore** will display **all** "connected" nodes (over 200).

• Use the **Scroll to Form** button to view the summary results (including the **12** inet:dns:a nodes) or to navigate to the results you select.



Return to Question 2

Q2 Storm Answer

Since we're just getting started, it's easy to find the answer using the UI as described in the **Q2 UI Answer**. But if you want to charge ahead and see how to get the answer using Storm, read on!

To look up the FQDN:

• In the **Research Tool**, using the query bar in **Storm mode**, **lift** the FQDN:

```
inet:fqdn=globalowa.com
```

To find the DNS A records:

Use Storm to either:

• Build on your previous query and **pivot out** to the DNS A records:

```
inet:fqdn=globalowa.com -> inet:dns:a
```

 Or, <u>lift</u> the DNS A records that have globalowa.com as the value of their :fqdn property:

```
inet:dns:a:fqdn=globalowa.com
```

Both queries give you the same answer - they are just different ways to ask the question!

Return to Question 2

Question 3 - Answer Explanations

- Q3 UI Answer
- Q3 Storm Answer

Q3 UI Answer

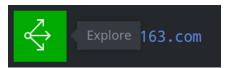
To look up the email address:

• In the **Research Tool**, with the query bar in **Lookup mode**, add the email address and press **Enter** to run the query:

lfengg@163.com

To find the FQDNs registered with this email:

• In the **Results Panel**, **click** the **Explore button** to navigate to any adjacent nodes:



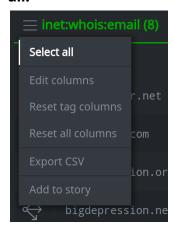
Locate the inet:whois:email nodes in the results (use the Scroll to Form button
if necessary):



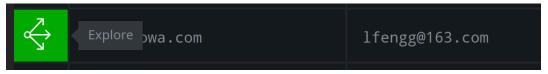
Looking at the inet:whois:email nodes, we can see the email address and associated FQDNs from the **properties** that are displayed for these nodes. To view the FQDNs (and any tags) directly, we need to navigate again.

To view the FQDNs and any associated tags:

• Use the **hamburger menu** next to the **inet:whois:email** header and choose **Select** all:



• Click the Explore button next to any selected node to navigate to adjacent nodes:

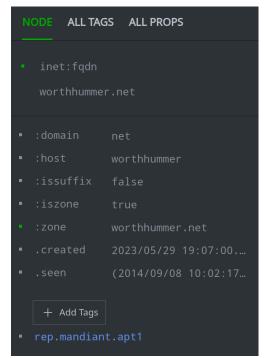


 Locate the inet: fqdn nodes in the results (use the Scroll to Form button if necessary):



To see which FQDNs have the tag #rep.mandiant.apt1 applied to them:

• In the **Results Panel**, select an individual node to view its tags in the **Details Panel**:



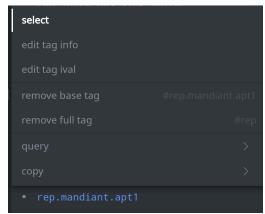
Viewing nodes one by one gets tedious! Another option:

• In the **Details Panel**, select the **ALL TAGS** tab:



The **ALL TAGS** tab shows us **all** of the tags that appear on **any** of our results, but does not necessarily mean that **all** nodes have **all** of these tags.

• Click the #rep.mandiant.apt1 tag and choose Select:



The **Select** option will highlight (select) all nodes in the **Results Panel** that have the #rep.mandiant.apt1 tag (i.e., all eight FQDNs and the email address):



This method is mostly useful when you have a small set of results.

Tip: If you want to view some (or all) tags directly in the Results Panel, you can configure Tabular mode to <u>display tags</u> as a <u>column</u>.

Return to Question 3

Q3 Storm Answer

Since we're just getting started, it's easy to find the answer using the UI as described in the **Q3 UI Answer**. But if you want to charge ahead and see how to get the answer using Storm, read on!

To look up the email address:

• In the **Research Tool**, using the query bar in **Storm mode**, **lift** the email address:

```
inet:email=lfengg@163.com
```

To find the FQDNs registered with this email:

• Pivot from the email address to the inet:whois:email nodes:

```
inet:email=lfengg@163.com -> inet:whois:email
```

...then **pivot** from the inet:whois:email nodes to the associated FQDNs:

```
inet:email=lfengg@163.com -> inet:whois:email -> inet:fqdn
```

To find the FQDNs reported as APT1:

Add a <u>filter</u> to only show nodes with the #rep.mandiant.apt1 tag:

```
inet:email=lfengg@163.com -> inet:whois:email -> inet:fqdn
+#rep.mandiant.apt1
```

Return to Question 3

Question 4 - Answer Explanations

• Question 4 - Bonus Notes

Q4 UI and Storm Answer

Note: Because this challenge does not require any navigation of the data (you don't need to use the Explore button or pivot from your original results), you get the "UI answer" and the "Storm answer" the same way (i.e., using a single query to lift the nodes and view the results).

To retrieve the indicators:

• In the **Research Tool**, using the query bar in **Storm mode**, enter the following and press **Enter** to run the query:

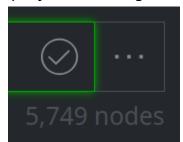
```
#rep.mandiant.apt1
```

Tip: Lookup mode allows us to ask about specific indicators, like hashes or email addresses. To ask about "nodes that have a tag" (like rep.mandiant.apt1), we need to use **Storm mode**.

You can "ask about" nodes that have a specific tag by entering the tag name (rep.mandiant.apt1) preceded by the "hashtag" symbol (#). In Storm, this **lifts** all nodes that have the tag.

To find the total number of indicators:

• Once all the nodes are loaded, the total node count is displayed underneath the query bar on the right side:



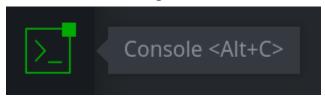
To find the total number of indicators using **count**:

When you run a query, Synapse automatically **displays** all of the results. If you only
care about the **total count** of the indicators and don't need to see them, the fastest
way to answer this question is by using the Storm <u>count</u> command:

```
#rep.mandiant.apt1 | count
```

Count will tally the number of results from your query but will not display the nodes. The pipe character (|) is used in Storm to separate Storm operations (such as lifts) from Storm commands (such as count).

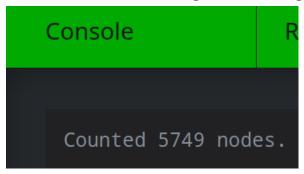
• **Count** returns a status message with the total number of results (nodes). Messages are displayed in the **Console Tool.** You will see a blinking square over the Console Tool icon when messages are available:



The square will be green for informational messages, yellow for warnings, and red for errors.

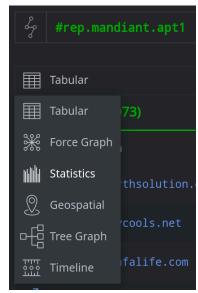
In the **Toolbar**, click the **Console Tool** icon to switch to that tool and view the message.

• You should see the following status message in the **Console Tool:**



To find the "most reported" indicator:

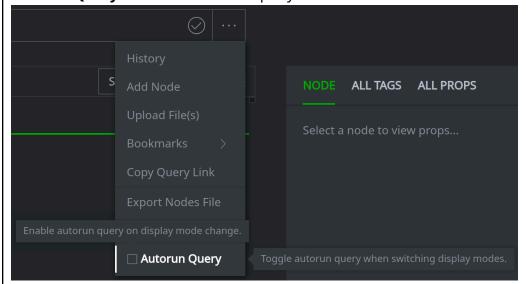
In the Research Tool, from the Display Mode Selector, select Statistics:



 Using the query bar in **Storm mode**, press **Enter** to re-run your query to **lift** all of the APT1 indicators:

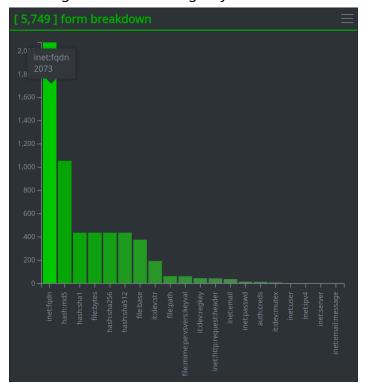
#rep.mandiant.apt1

Tip: By default, Synapse will **not** automatically re-run your query when you switch display modes. You can change this behavior by checking the box next to **Autorun Query** on the main Storm query bar menu:



You can also configure this option from the **RESEARCH** tab under **Your Settings**.

• Use the **form breakdown** chart to identify the histogram bar representing the most reported form (i.e., the bar on the left representing inet:fqdn nodes). Hovering over the bar will give you the total number of nodes of that form:



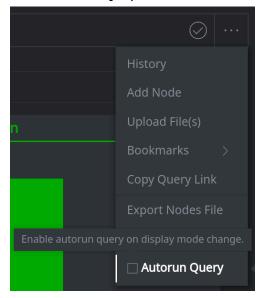
Tip: In Tabular display mode, you could optionally use the **Scroll to Form** button to view the summary results and find the form with the greatest number. This works for smaller result sets, but it may be harder to "spot" the highest value for larger result sets.

Return to Question 4

Question 4 - Bonus Notes

- Research Tool vs Console Tool. While you can load 5,000+ nodes in the Research
 Tool, your browser performance will start to lag for large numbers of nodes. The
 Console Tool can be a useful or more "lightweight" way to answer some questions
 if you don't need to visualize the output.
- Research Tool Display Modes. The Research Tool includes a Display Mode Selector so you can choose how to visualize your data. The results of any query you run in the query bar will be displayed based on the current display mode. When switching

between display modes, by default you will **need to re-run your query** to view the results using the new display mode. If you want the current query to run **automatically** when switching display modes, click the **meatball menu** (the three horizontal dots) to the right of the Storm query bar and check the box next to the **Autorun Query** option:



You can also configure this option under **Your Settings** (on the **RESEARCH** tab).

'rep' tags and third-party reporting. Synapse uses tags (labels) to annotate nodes (objects). Tags provide context to nodes and can be used to group nodes together. Synapse does not come with any "built in" tags; you must choose a set of tags that are useful for your analysis. (Power-Ups that can apply tags may use certain defaults - such as rep. * for tags reported by third-parties - which you can optionally override.)

The Vertex Project analysts use different sets of tags (<u>tag trees</u>) to distinguish context based on our own assessments (using the cno.* tag tree) from assessments reported by other organizations (using the rep.* tag tree).

Return to **Question 4**

Question 5 - Answer Explanations

Q5 UI and Storm Answer

Note: Because this challenge does not require any navigation of the data (you don't need to use the Explore button or pivot from your original results), you get the "UI answer" and the "Storm answer" the same way (i.e., using a single query to lift the nodes and view the results).

To find the APT1 MD5 hashes reported by Mandiant:

In the Research Tool, with the query bar in Storm mode, run the following query:

```
hash:md5#rep.mandiant.apt1
```

Tip: Because **Lookup mode** only allows us to ask about specific indicators, we need to use **Storm mode** to ask about kinds of indicators and / or indicators with specific tags.

You can "ask about" specific objects that have a particular tag by entering the object (form) name (hash:md5) followed by the "hashtag" symbol (#) and the tag name (rep.mandiant.apt1). In Storm, this **lifts** all nodes of the specified kind that have the tag.

To find the most reported malware family:

In the Research Tool, use the Display Mode Selector to choose Statistics mode:

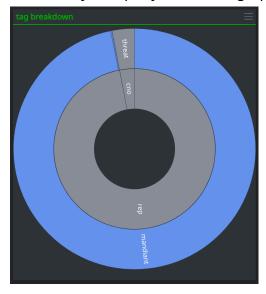


• Using the query bar in **Storm mode**, run the following query:

hash:md5#rep.mandiant.apt1

Statistics mode generates **sunburst charts** for any dot-separated object. This includes **tags** applied to the nodes in your results (because tags use a dotted namespace).

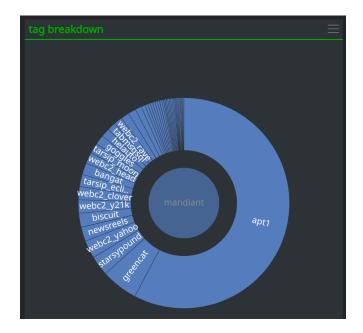
• The **tag breakdown** sunburst graph displays **all** the tags applied to **all** the nodes returned by the query. The initial graph returned looks like this:



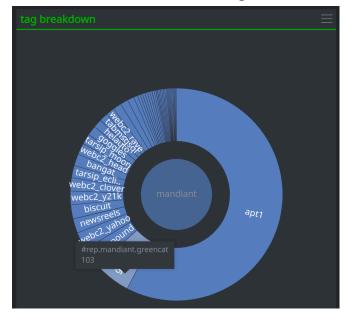
The chart above shows the "top level" tags (cno and rep) in the center ring of the sunburst, with each tag's second-level tags (cno.threat and rep.mandiant) in the outermost ring.

• You can "drill down" into the chart by clicking any segment of the sunburst to view tags lower in the tag tree (click the center of the chart to "drill up" again or use the hamburger menu to select **reset**).

Clicking the **mandiant** section allows you to drill down into the rep.mandiant tag tree. The most common tag (after rep.mandiant.apt1) is rep.mandiant.greencat:



Hovering over the greencat portion of the sunburst will display a tooltip showing the number of nodes with that tag (103):



Return to **Question 5**

Intermediate Challenges

Question 6 - Answer Explanations

Q6 UI and Storm Answer

Note: Because this challenge does not require any navigation of the data (you don't need to use the Explore button or pivot from your original results), you get the "UI answer" and the "Storm answer" the same way (i.e., using a single query to lift the nodes and view the results).

To find the indicators:

Full query:

```
#rep.mandiant.apt1 +#rep.symantec.commentcrew
```

Step by step:

Note: For this and subsequent explanations, when we say "step by step", we're simply breaking down the Storm query into its individual components for clarity. Running the full query above all at once gets you the answer you need. You don't need to go "step by step" (i.e., run the first query, then add the next step and run the second query, etc.), although you can if you want to see how each step in the query affects your results.

These two approaches show that you can use Storm to **either** ask exactly what you want to know "all at once" **or** build a query step by step (operation by operation) as you figure out where you want to go!

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode,** first **lift** (lift by tag) all of the APT1 indicators:

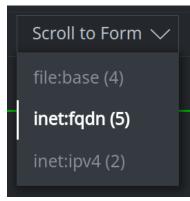
```
#rep.mandiant.apt1
```

• ...then **filter by tag** to show only those indicators that were **also** reported by Symantec:

```
#rep.mandiant.apt1 +#rep.symantec.commentcrew
```

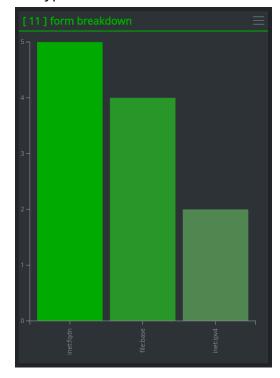
To view the types of indicators:

• In **Tabular mode**, use **Scroll to Form** to browse the types of indicators returned:



Alternatively:

• In **Statistics mode**, re-run your query and use the **form breakdown** chart to view the types of indicators returned:



Return to **Question 6**

Question 7 - Answer Explanations

- Q7 UI Answer
- Q7 Storm Answer

Q7 UI and Storm Answer

Note: Because this challenge does not require any navigation of the data (you don't need to use the Explore button or pivot from your original results), you get the "UI answer" and the "Storm answer" the same way (i.e., using a single query to lift the nodes and view the results).

To find the indicators:

Full query:

```
#rep.symantec.commentcrew -#rep.mandiant.apt1
```

Step by step:

• In the **Research Tool, Tabular display mode,** with the query bar in **Storm mode,** first **lift** the Symantec Comment Crew indicators...

```
#rep.symantec.commentcrew
```

...then **filter by tag** to **exclude** the indicators that were also reported by Mandiant:

```
#rep.symantec.commentcrew -#rep.mandiant.apt1
```

To view the types of indicators:

• You can view the total indicators reported by Symantec and the types of indicators using the same methods you used for <u>Question 6</u>.

Return to **Question 7**

Question 8 - Answer Explanations

- Q8 UI Answer
- Q8 Storm Answer
- Question 8 Bonus Notes

Q8 UI and Storm Answer

Note: Because this challenge does not require any navigation of the data (you don't need to use the Explore button or pivot from your original results), you get the "UI answer" and the "Storm answer" the same way (i.e., using a single query to lift the nodes and view the results).

To find the indicators:

Full query:

```
#rep.symantec.commentcrew +#rep.crowdstrike.putterpanda
```

Step by step:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode**, first **lift** the Symantec Comment Crew indicators...

```
#rep.symantec.commentcrew
```

...and then **filter by tag** to **include** those indicators that were **also** reported by Crowdstrike:

```
#rep.symantec.commentcrew +#rep.crowdstrike.putterpanda
```

To view the types of indicators:

 You can view the total indicators reported by both organizations and the types of indicators reported using the same methods you used for <u>Question 6</u> and <u>Question</u> <u>7</u>.

Return to <u>Question 8</u>

Question 8 - Bonus Notes

Reported indicators vs. tagged indicators / "pushing" tags. Technically, only
three indicators overlap, based on indicators actually reported by both
organizations: two MD5 hashes and one file name. The difference lies in the MD5
hashes (hash:md5, which were reported by both organizations) vs. the files
(file:bytes nodes) associated with the hashes.

When organizations report indicators, they report **hashes** (whether MD5 or SHA1 or SHA256), not "files". When Vertex analysts are able to identify a file (file:bytes) that represents the reported hash, we copy (or "push") the rep tag(s) associated with the hash (e.g., rep.symantec.commenterew) to the file:bytes node, and from the file to the file's other hashes (such as hash:sha1). This provides additional **context** to other data (nodes) within Synapse.

For example, let's say Symantec reports an MD5 as malicious, and a Vertex analyst tags the hash:md5 node. Another Vertex analyst looks up a suspicious SHA1 hash (hash:sha1) that corresponds to the same file. If we did not "push" Symantec's tags to the additional hashes, the second analyst would not have that valuable context about the hash. The analyst would need to navigate from the (untagged) hash:sha1 to the (untagged) file:bytes and finally to the tagged hash:md5 to determine the SHA1 was malicious.

In contrast, by "pushing" the tags to the associated file and hashes, the second analyst has immediate "situational awareness" that the SHA1 is malicious.

Return to Question 8

Question 9 - Answer Explanations

- Q9 UI Answer
- Q9 Storm Answer
- Question 9 Bonus Notes

Q9 UI Answer

To find the hashes:

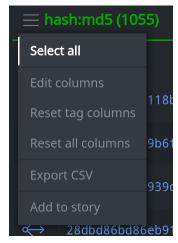
 In the Research Tool, Tabular display mode, use the query bar in Storm mode to lift the MD5 hashes reported by Mandiant:

hash:md5#rep.mandiant.apt1

There are **1,055** hashes returned by this query.

To find the associated files:

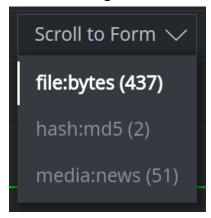
• Click the hamburger menu next to the hash:md5 header and choose Select all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



Locate or navigate to the file:bytes nodes (use Scroll to Form if necessary):



Return to Question 9

Q9 Storm Answer

To find the hashes:

• In the **Research Tool**, use the query bar in **Storm mode** to **lift** the MD5 hashes reported by Mandiant:

```
hash:md5#rep.mandiant.apt1
```

There are **1,055** hashes returned by this query.

To find the associated files:

Pivot from the hashes to any associated file:bytes nodes:

```
hash:md5#rep.mandiant.apt1 -> file:bytes
```

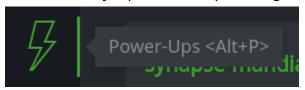
Return to **Ouestion 9**

Question 9 - Bonus Notes

Enriching indicators. Your Synapse demo instance includes "enriched" indicator
data - that is, Vertex analysts already took the MD5 hashes reported by Mandiant
and used various Synapse Power-Ups to "enrich" the hashes (i.e., submit them to
various third-party data sources to obtain additional information such as file
metadata, file multiscanner data, or file execution data).

The same is true for other indicators - for example, in many cases FQDNs and IPv4 addresses have been enriched using Power-Ups that connect to third-party passive DNS and whois services. Enrichment is a standard part of the analysis process, but for purposes of this demo, The Vertex Project performed some of that enrichment for you.

You can view Synapse Power-Ups through the **Power-Ups Tool:**



You do not need to use any Power-Ups to solve the Scavenger Hunt, but they are available in your demo instance for you to test. **INSTALLED** Power-Ups have already been loaded into your instance; you can load other Power-Ups from the **AVAILABLE** tab.

Note: Some Power-Ups require API keys to communicate with third-party data sources; these keys are not provided by Synapse or The Vertex Project. You may need to obtain keys (free or paid) and configure them in Synapse to use some Power-Ups.

Note that further enriching the demo data may affect the answers to some of the Scavenger Hunt questions (e.g., your node counts may vary if you add data to Synapse).

• Implicit vs. explicit pivot syntax. When you pivot in Storm, you navigate between objects (nodes) that share a property value. In this case, you are pivoting from hash:md5 nodes to any files that have the same MD5 (file:bytes:md5 property).

When you pivot in Storm, you need to tell Synapse:

- where you are starting from (in this case, the hash:md5 nodes we lifted with our query); and
- where you want to pivot to (in this case, any file:bytes nodes with an :md5 property that matches any of our starting nodes).

In the query above, we simply told Synapse we wanted to pivot from hash:md5

nodes to file:bytes nodes - we did not specifically tell Synapse that we wanted the file:bytes:md5 property. The query uses what we call <u>implicit pivot syntax</u> - the :md5 property is "implied" because Synapse "knows" that file:bytes:md5 is the logical "target" of the pivot if you're coming from an MD5 hash (hash:md5).

You could optionally use **explicit pivot syntax** to tell Synapse **exactly** what target property you want, but in this particular case it is not required:

```
hash:md5#rep.mandiant.apt1 -> file:bytes:md5
```

Implicit pivot syntax (implicit syntax) is nice because it is intuitive and in many cases, it "just works". However, there are times when you may want (or need) to use **explicit pivot syntax** (explicit syntax) to tell Synapse **exactly** what you want.

You can learn more about implicit vs. explicit pivot syntax in the **Storm Reference**.

Return to Question 9

Question 10 - Answer Explanations

- Q10 UI Answer
- Q10 Storm Answer
- Question 10 Bonus Notes

Q10 UI Answer

To retrieve the APT1 FODNs:

• In the **Research Tool**, use the query bar in **Storm mode** to **lift** the FQDNs reported by Mandiant:

```
inet:fqdn#rep.mandiant.apt1
```

• Note the total number of inet:fqdn nodes returned:

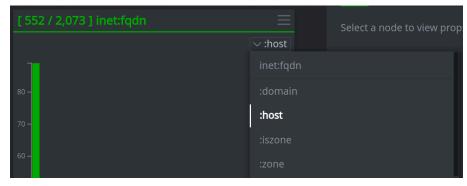


To find the most commonly used hostnames:

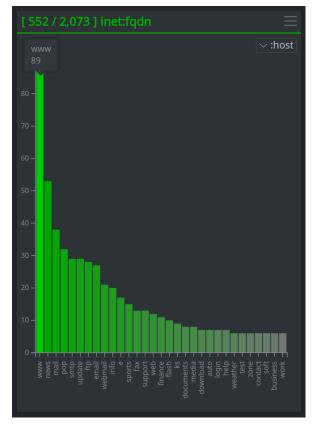
• In the **Research Tool**, **Statistics display mode**, use the query bar in **Storm mode** to re-run your query:

```
inet:fqdn#rep.mandiant.apt1
```

• Use the **inet:fqdn** histogram to view the FQDNs used by APT1. You can specify which FQDN **property** you want the histogram to display. Use the drop-down menu to group your results by the **:host** property:



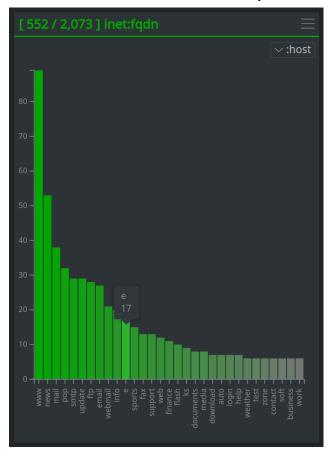
• Hover over the largest bars on the left to see the associated hostnames and counts:



To find the APT1 FODNs with hostname 'e':

There are a few options to find this answer using the UI. Since we are already in Statistics mode, we can find the answer here.

• In the **inet:fqdn** histogram, locate the bar representing the hostname 'e'. Hover over the bar to see the number of FQDNs with this hostname:



Return to Question 10

Q10 Storm Answer

To retrieve the APT1 FQDNs:

• In the **Research Tool**, with the query bar in **Storm mode**, run the following query and view the total number of results returned:

inet:fqdn#rep.mandiant.apt1

 Alternatively, use the <u>count</u> command to display the total number of FQDNs in the Console Tool:

```
inet:fqdn#rep.mandiant.apt1 | count
```

To find the most commonly used hostnames:

Use Statistics display mode and use the inet:fqdn bar chart (distributed by host)
to identify the most commonly used subdomains / hostnames (as described in the
Q10 UI Answer above).

To find the APT1 FQDNs with hostname 'e':

Full query:

```
inet:fqdn#rep.mandiant.apt1 +:host=e
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the APT1 FQDNs...

```
inet:fqdn#rep.mandiant.apt1
```

...then filter to only show those FQDNs with the hostname 'e':

```
inet:fqdn#rep.mandiant.apt1 +:host=e
```

Return to <u>Question 10</u>

Question 10 - Bonus Notes

• **Synapse performance / optimizing Storm queries.** As you learn Storm, you will realize that there is usually more than one way to use Storm to ask (and answer) the same question.

For example, to find the APT1 FQDNs with hostname 'e'), we started by:

- lifting the tagged FQDNs, and
- filtering the FQDNs to only those with hostname 'e':

```
inet:fqdn#rep.mandiant.apt1 +:host=e
```

Another way to get the same answer is to start by:

- o lifting all FQDNs with hostname 'e', and
- filtering the FQDNs to only those tagged APT1:

```
inet:fqdn:host=e +#rep.mandiant.apt1
```

Both queries will return the same answer; neither query is "better" than the other. However, there are cases where some queries are faster or more efficient than others.

We can run each of the queries above using the **Console** tool to see how quickly we get our answer. For example:

```
> inet:fqdn#rep.mandiant.apt1 +:host=e
...
<results>
...
complete. 17 nodes in 194 ms (88/sec).
```

Compared to:

```
> inet:fqdn:host=e +#rep.mandiant.apt1
...
<results>
...
complete. 17 nodes in 5 ms (3400/sec).
```

While both queries give you the same answer, there is a difference in "how much work" Synapse has to do to get the answer for you:

- For the first query, Synapse has to lift all the APT1 FQDNs (over 2,000 of them), and then drop (**filter**) all the ones that **don't** have the hostname 'e'.
- For the second query, Synapse has to lift all of the FQDNs that have the hostname 'e' (fewer than 20 of them), then drop (filter) those that don't have the APT1 tag.

For the second query, Synapse does less work, so you get your answer a bit faster.

For this query (and for many queries), the difference in response time is negligible and not very noticeable to the user (the difference above is just milliseconds). But for larger or more complex queries, "how" you ask a question may make a much bigger difference.

One good rule of thumb for helping to optimize your Storm queries is to always **start by lifting the smallest number of nodes possible.** Your lift operation is usually Synapse's first action - it has to retrieve that data from the data store. Having Synapse retrieve fewer nodes first (before performing any subsequent filters, pivots, etc.) is generally more efficient.

Return to Question 10

Question 11 - Answer Explanations

- Q11 UI Answer
- Q11 Storm Answer
- Question 11 Bonus Notes

Q11 UI Answer

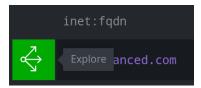
To find the DNS A records and associated IPv4 addresses:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode,** run the following query:

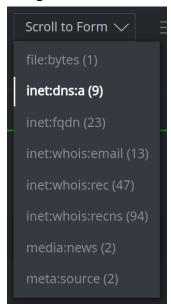
inet:fqdn=jobsadvanced.com

You can optionally just enter the FQDN into the query bar in **Lookup mode** as your starting point. But you're already getting used to using Storm for your initial queries, so why stop now?

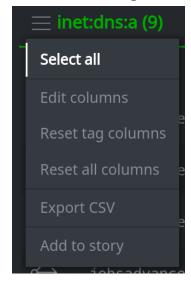
• In the **Results Panel**, **click** the **Explore button** next to the FQDN to navigate to adjacent nodes:



• Navigate to the inet:dns:a nodes (use **Scroll to Form** if necessary):



• Click the hamburger menu next to the inet:dns:a header and choose Select all:



• **Click** the **Explore button** next to any selected node to navigate to adjacent nodes:



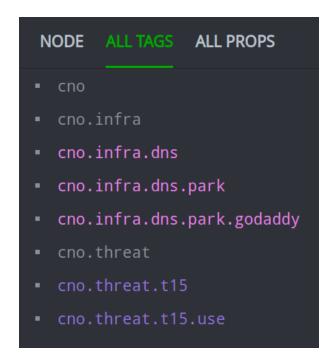
• Locate the inet:ipv4 nodes (use **Scroll to Form** if necessary) to see the total number of results:



To find the IPv4s that are parking infrastructure:

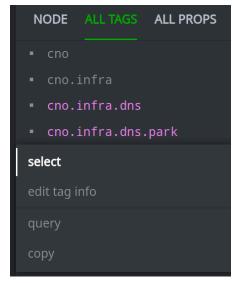
Continuing from your previous navigation:

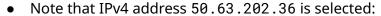
• In the **Details Panel**, select the **ALL TAGS** tab to view **all** tags on **any** of the currently displayed results:



The tag cno.infra.dns.park indicates parking infrastructure (generally), while cno.infra.dns.park.godaddy indicates parking infrastructure associated with domain registrar GoDaddy.

• In the **Details Panel**, **click** the tag **cno.infra.dns.park** and choose **select** to select any nodes with this tag in the Results Panel:







Tip: the color cues provided by the <u>Tag colors</u> configured in your Workspace are also a good way to identify the associated IPv4 address.

To find the IPv4s that are threat actor infrastructure:

• In the **Details Panel**, use the **ALL TAGS** tab to view **all** tags on **any** of the currently displayed results.

The tag cno.threat is the top-level tag that Vertex uses for threat clusters that we identify and track internally. Cno.threat.t15 indicates nodes associated with threat cluster T15 generally. The tag cno.threat.t15.use refers to objects the cluster makes use of (like an IPv4 address), but does not fully control (T15 does not "own" this IPv4 or associated netblock).

(In contrast, the tag cno.threat.t15.own refers to objects the cluster owns or controls, like malware they create or FQDNs they register.)

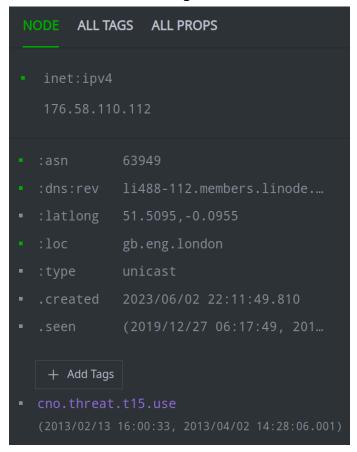
• In the **Details Panel**, **click** the tag **cno.threat** and choose **select** to select any nodes with a Vertex threat cluster tag in the Results Panel. Note that IPv4 address 176.58.110.112 is selected:



To find when the IPv4 was used by threat cluster T15:

• Ensure that IPv4 176.58.110.112 is selected in your **Results Panel**.

• In the **Details Panel**, select the **NODE** tab. View the timestamps associated with the cno.threat.t15.use tag:



You can also add the tag's timestamps as columns to your Results Panel.

Return to <u>Question 11</u>

Q11 Storm Answer

To find the IPv4 addresses:

Full query:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
```

Or:

```
inet:dns:a:fqdn=jobsadvanced.com -> inet:ipv4
```

Step by step:

Option 1:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the FQDN...

```
inet:fqdn=jobsadvanced.com
```

...then **pivot** to the DNS A records...

```
inet:fqdn=jobsadvanced.com -> inet:dns:a
```

...then **pivot** to the IPv4 addresses:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
```

Option 2:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the DNS A records for FQDN jobsadvanced.com directly...

```
inet:dns:a:fqdn=jobsadvanced.com
```

•then **pivot** to the IPv4 addresses:

```
inet:dns:a:fqdn=jobsadvanced.com -> inet:ipv4
```

To find the IPv4s that are parking infrastructure:

Full query:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
+#cno.infra.dns.park
```

Step by step:

 Once you pivot to the inet:ipv4 nodes, you need to identify the tag(s) used to represent parking infrastructure. Use the ALL TAGS tab in the Details Panel, as described in the Q11 UI Answer. Once you have identified the parking infrastructure tag, you can modify your earlier
 Storm query and add a <u>filter by tag</u> to include IPv4s that are tagged as parking infrastructure:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
    +#cno.infra.dns.park
```

To find the IPv4s that are threat actor infrastructure:

Full query:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4 +#cno.threat
```

Step by step:

• Similar to the previous section, you need to identify the tag(s) that Vertex uses to represent threat activity. You can then modify your query to <u>filter by tag</u> to only **include** IPv4 addresses with that tag:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
+#cno.threat
```

To find the time when the IPv4 was in use by the threat group:

Vertex analysts added a time interval to the IPv4's associated
 #cno.threat.t15.use tag to show when we believe the IP was in use by T15.

View the timestamp as described in the <u>Q11 UI Answer</u> (i.e., by viewing the IPv4's information in the **Details Panel**, **NODE tab** or by <u>adding the tag's timestamp</u> <u>columns</u> to your **Results Panel** display).

Return to **Question 11**

Question 11 - Bonus Notes

• **Lift vs. lift and pivot:** in the Storm answer above we showed you two options to find the IPv4 addresses our FQDN resolved to:

 Lift the FQDN jobsadvanced.com and then pivot to the DNS A records and IPv4 addresses:

```
inet:fqdn=jobsadvanced.com -> inet:dns:a -> inet:ipv4
```

This query selects (**lifts**) the FQDN by its **primary property.**

Or, simply lift the DNS A records directly and pivot to the IPv4 addresses:

```
inet:dns:a:fqdn=jobsadvanced.com -> inet:ipv4
```

This query selects (**lifts**) the DNS A records by specifying a value for a **secondary property** (in this case the FQDN).

From a performance standpoint, the difference is insignificant, so use whichever one works for you! Your preference often depends on how your "analyst brain" thinks:

- If your thought process starts with "Okay, for FQDN jobsadvanced.com I need to find the DNS A records..." you might find it easier to write inet:fqdn=jobsadvanced.com -> inet:dns:a.
- If your thought process starts with "I need to see the DNS A records for jobsadvanced.com..." you might find it easier to write inet:dns:a:fqdn=jobsadvanced.com.

Note: So that we don't have to show **both** methods for **every** answer, we'll simply choose one. Just keep in mind that for many Storm queries that start with "lift by primary property and pivot" operations, there is an equivalent "lift by secondary property" operation.

• **Tag trees / tag hierarchies:** One benefit of Synapse's "dotted notation" for tags is that it allows you to lift or filter on tags at any level of specificity.

For example, when you apply the tag #cno.infra.dns.park.godaddy to a node, Synapse technically applies **all** of the following tags:

- o #cno
- o #cno.infra
- o #cno.infra.dns
- o #cno.infra.dns.park
- #cno.infra.dns.park.godaddy

In the challenge question and answer above, you can refer to any IPv4 that is

parking infrastructure owned by **any** organization using a single filter:

-#cno.infra.dns.park

This is much easier than potentially having to specify multiple filters using full tag names:

-#cno.infra.dns.park.godaddy -#cno.infra.dns.park.confluence

Return to Question 11

Question 12 - Answer Explanations

- Q12 UI Answer
- Q12 Storm Answer
- Question 12 Bonus Notes

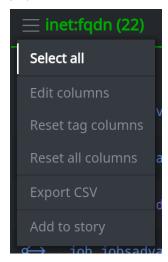
Q12 UI Answer

To find the IPv4 addresses:

 In the Research Tool, Tabular display mode, with the query bar in Storm mode, run the following query to lift all of the FQDNs that are part of the jobsadvanced.com zone:

```
inet:fqdn:zone=jobsadvanced.com
```

Click the hamburger menu next to the inet:fqdn table header and choose Select
 all:



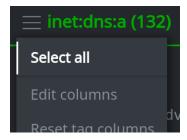
• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



• Use **Scroll to Form** to navigate to the inet:dns:a nodes:



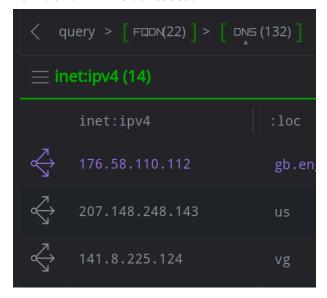
• Click the **hamburger menu** next to the **inet:dns:a** table header and chose **Select** all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



• Locate the inet:ipv4 nodes (use **Scroll to Form** if necessary) to view the total number of IPv4 addresses:



To find the most used Autonomous System (AS):

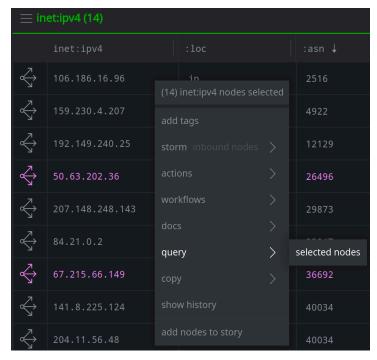
It is easiest to use **Statistics mode** to find the answer. First we need to update our starting query to **just** ask about the IPv4 addresses we have identified.

Continuing from your previous navigation:

• In the **Research Tool, Tabular** display mode, click the **hamburger menu** next to the **inet:ipv4** table header and choose **Select all:**



 Right-click any selected node and choose query > selected nodes from the context menu:



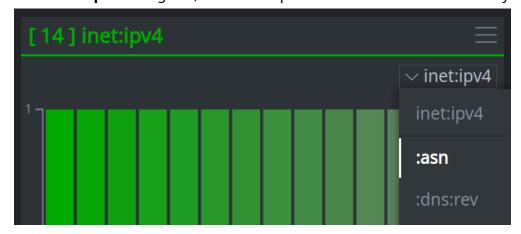
Note that Optic automatically populates the Storm query bar and runs a **new** query that **only** asks about (lifts) these 14 IPv4 addresses:

```
inet:ipv4=106.186.16.96 inet:ipv4=159.230.4.207 inet:ipv4=192.149.240.25 inet:ipv4=50.63.20
```

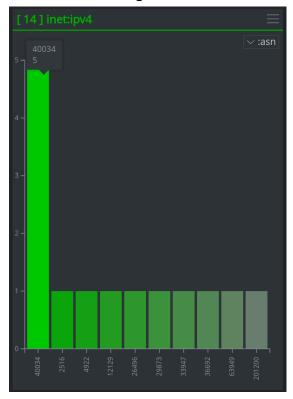
• Use the **Display Mode** selector to select **Statistics** mode:



- Place your cursor in the **query bar** and press **Enter** to re-run the query for the 14 IPv4 addresses.
- In the **inet:ipv4** histogram, use the drop down menu to sort the chart by **:asn:**

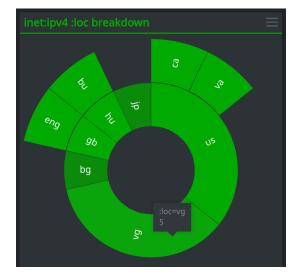


• **Hover over** the largest bar to view the AS number and count:



To find the most used country (:loc):

• In **Statistics view**, use the **inet:ipv4:loc breakdown** sunburst chart to view the locations (broken out on their dotted boundaries). **Hover over** the sunburst sections to view the counts for each section:

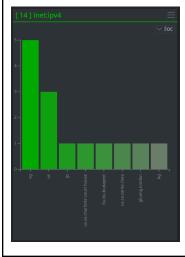


Both **us** and **vg** hosted five IPv4 addresses.

Tip: The geolocation information stored in the **:loc** property is typically provided by third-party data sources as a dotted string representing

<country>.<region>.<city>. Some location strings may be general (fr) and others
may be more specific (ca.bc.vancouver). While you can also use the inet:ipv4
histogram to sort by :loc property, the property is tallied exactly as it appears so us.ca.santa clara is counted separately from us.

This can give the false impression that most IPv4 addresses were located in the British Virgin Islands, even though the total number of addresses in the United States (in various locations) is the same:



Return to Ouestion 12

Q12 Storm Answer

To find the IPv4 addresses:

Full query:

```
inet:fqdn:zone=jobsadvanced.com -> inet:dns:a -> inet:ipv4 | uniq
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** all of the FQDNs in the jobsadvanced.com **zone**...

```
inet:fqdn:zone=jobsadvanced.com
```

• ...then **pivot** to the inet:dns:a records (132 results)...

```
inet:fqdn:zone=jobsadvanced.com -> inet:dns:a
```

• ...then **pivot** to the inet:ipv4 nodes (also 132 total results):

```
inet:fqdn:zone=jobsadvanced.com -> inet:dns:a -> inet:ipv4
```

Your results contain many duplicate IPv4 addresses. This is because in some cases, more than one domain resolves to the same IPv4 address. When we pivot from the DNS A records to the IPv4 addresses, Synapse returns one inet:ipv4 for **each** pivot from **each** inet:dns:a node, which may return multiple "copies" of some IPv4s.

 ...use the <u>uniq</u> command to remove duplicate results. This reduces your results from 132 total IPv4s to only 14 unique IPv4s:

To find the most used Autonomous System (AS) and country:

- Use the Research Tool's **Statistics display mode** to identify the most used Autonomous System (AS) and the country with the most IPv4 addresses.
- In the Research Tool, switch to the **Statistics** display mode and re-run the above query:

• Refer to the **Q12 UI Answer** for a description of how to use the **inet:ipv4** and **inet:ipv4 :loc breakdown** charts to find the answers.

Return to Question 12

Question 12 - Bonus Notes

• **FQDNs and zones:** An FQDN "zone" in Synapse is effectively a **zone of control** - whoever registers the FQDN jobsadvanced.com controls the ability to create subdomains for that FQDN and to manage DNS records (such as A records) for that FQDN and its subdomains.

Note that an FQDN that **is** a zone is also part of its own zone (so inet:fqdn=jobsadvanced.com has the secondary property

inet:fqdn:zone=jobsadvanced.com).

Lifting all FQDNs that have inet:fqdn:zone=jobsadvanced.com will return all of jobsadvanced.com's subdomains (regardless of depth - it will return foo.jobadvanced.com as well as bar.baz.jobsadvanced.com) as well as the FQDN jobsadvanced.com itself.

For detailed information about FQDNs, zones, and suffixes in Synapse, see our in-depth documentation on <u>inet:fqdns</u>.

Return to Question 12

Question 13 - Answer Explanations

- Q13 UI Answer
- Q13 Storm Answer

Q13 UI Answer

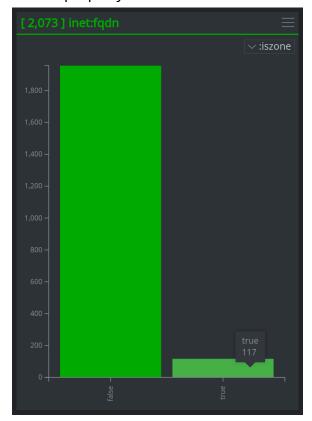
• In the **Research Tool**, use the **Display Mode Selector** to select **Statistics** mode:



Using the query bar in **Storm mode**, run the following query to **lift** all of the APT1 FQDNs:

inet:fqdn#rep.mandiant.apt1

• In the **inet:fqdn** bar chart, use the drop down menu to sort the results by the **:iszone** property. Hover over the **True** column to view the total count:



Return to Question 13

Q13 Storm Answer

Full query:

```
inet:fqdn#rep.mandiant.apt1 +:iszone=true
```

Step by step:

• In the **Research Tool, Tabular** display mode, using the query bar in **Storm mode**, first **lift** the APT1 FQDNs...

```
inet:fqdn#rep.mandiant.apt1
```

• ...then **filter** to display only those FQDNs that are zones:

```
inet:fqdn#rep.mandiant.apt1 +:iszone=true
```

Tip: The :iszone property is a Boolean value (an FQDN is either a zone, or it's not a zone). The values true / false and 1 / 0 can be used interchangeably for Boolean properties, so +:iszone=1 will also work in the above query.

Return to <u>Question 13</u>

Question 14 - Answer Explanations

- Q14 UI Answer
- O14 Storm Answer
- Question 14 Bonus Notes

Q14 UI Answer

N/A

Return to Question 14

Q14 Storm Answer

To retrieve all the zones associated with APT1 FQDNs:

Full query:

```
inet:fqdn#rep.mandiant.apt1 :zone -> inet:fqdn | uniq
```

Step by step:

 In the Research Tool, with the query bar in Storm mode, first lift the APT1 FQDNs...

```
inet:fqdn#rep.mandiant.apt1
```

 ...then pivot from the :zone property of those FQDNs to the associated inet:fqdn nodes for the zones...

```
inet:fqdn#rep.mandiant.apt1 :zone -> inet:fqdn
```

Note that we used **explicit pivot syntax** to specifically tell Synapse we want to pivot **from** the FQDN :zone property to FQDNs (inet:fqdn nodes) matching those property values.

...then use the uniq command to deduplicate the results:

```
inet:fqdn#rep.mandiant.apt1 :zone -> inet:fqdn | uniq
```

Return to Question 14

Question 14 - Bonus Notes

Explicit vs. implicit pivot syntax. In the <u>Question 9 - Bonus Notes</u> we introduced implicit vs. explicit pivot syntax. Our Storm pivots so far have used <u>implicit pivot</u> syntax - we've been able to pivot simply using the names of the forms (objects) we want to pivot between. Implicit syntax works in many cases because Synapse "knows" which properties we want to use to pivot, based on our source and target nodes.

In the query above, we used **explicit pivot syntax** to specifically tell Storm to pivot from **only** the :zone property to the associated inet:fqdn nodes. If we **don't** do this, we'll get extra results that we don't want. Try it yourself - run the following queries to see the difference:

```
inet:fqdn#rep.mandiant.apt1-> inet:fqdn | uniq
```

vs.

```
inet:fqdn#rep.mandiant.apt1 :zone -> inet:fqdn | uniq
```

If you are pivoting from an FQDN to another FQDN - like in this question - Synapse "knows" that if you're pivoting **to** an FQDN, you must want to pivot **from** any FQDN properties on your starting nodes. However, an inet:fqdn node has more than one secondary property that is an inet:fqdn:

inet:fqdn:domain
inet:fqdn:zone

By default (i.e., using implicit pivot syntax) Synapse will pivot from **both** of those properties to the associated FQDNs.

For purposes of answering this question, we don't care about the :domain property, so we use explicit syntax to remove any ambiguity and tell Synapse to

only pivot from the :zone property to the associated inet:fqdn.

Return to Question 14

Question 15 - Answer Explanations

- Q15 UI Answer
- Q15 Storm Answer
- Question 15 Bonus Notes

Q15 UI Answer

To find the whois records:

• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** the FQDN:

inet:fqdn=youipcam.com

You could also start with the query bar in **Lookup mode** and simply enter the FQDN **youipcam.com**.

• In the **Results Panel**, **click** the **Explore button** to navigate to adjacent nodes:



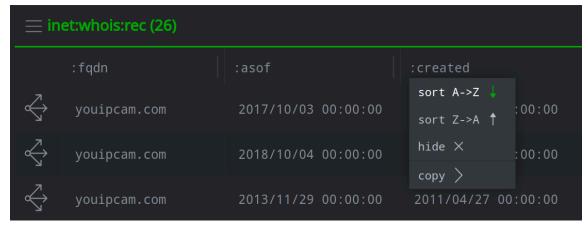
• Navigate to the inet:whois:rec nodes (use **Scroll to Form** if necessary).



To find the earliest registration date:

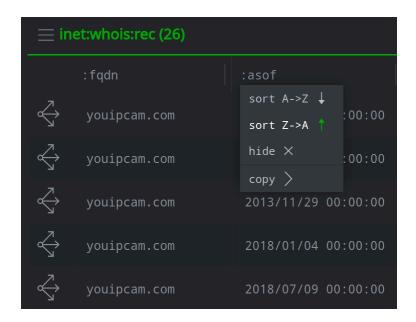
Your Workspace should display columns for various dates for the inet:whois:rec nodes (e.g,. :asof, :created, etc.). If they are not visible, you can <u>select which properties to display</u>.

Click the :created column header to sort the values in ascending order (sort A->Z) and find the earliest registration date:



To find the most recent capture date:

• **Click** the :asof column header to **sort** the values in descending order (**sort Z->A**) and find the most recent capture date:



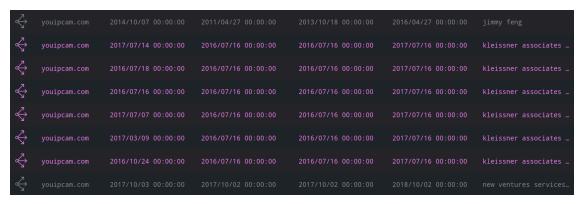
To find the registration date for Kleissner & Associates:

Your Workspace should display a column for the :registrant property for the inet:whois:rec nodes. (If this property is not visible, you can <u>select which properties to display</u>.)

• **Click** the :registrant column header to **sort** the values in ascending order:

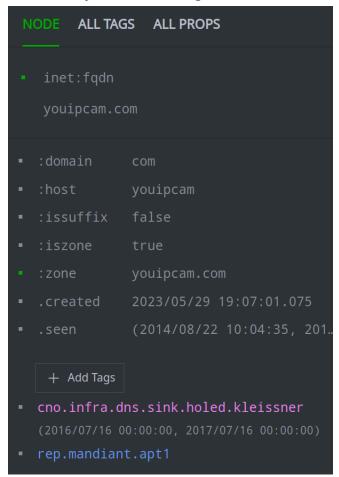


• Locate the records for the registrant **kleissner associates s.r.o.** to find the registration (:created) date:



The records associated with Kleissner appear in color, based on the tag(s) on the inet:whois:rec nodes and your Workspace's tag color rules. The tag cno.infra.dns.sink.reg.kleissner indicates that the whois records represent domain registration data associated with DNS sinkholing activity carried out by Kleissner & Associates (VirusTracker).

The FQDN youipcam.com has also been annotated to indicate that the domain was sinkholed by Kleissner, along with the relevant dates:



Return to **Question 15**

Q15 Storm Answer

To find the whois records:

• In the **Research Tool**, with the query bar in **Storm mode**, run the following query to lift the whois records for the FQDN:

```
inet:whois:rec:fqdn=youipcam.com
```

You can also **lift** the FQDN and **pivot** to the whois records: **inet:fqdn=youipcam.com -> inet:whois:rec.**

To find the earliest registration date:

 Use the Storm min command to find a whois record with the earliest registration (:created) date for the FQDN:

```
inet:whois:rec:fqdn=youipcam.com | min :created
```

To find the most recent capture date:

 Use the Storm <u>max</u> command to find a whois record with the most recent capture (:asof) date for the FQDN:

```
inet:whois:rec:fqdn=youipcam.com | max :asof
```

<u>To find the registration date for Kleissner & Associates:</u>

- In the **Details Panel**, use the **ALL TAGS** tab to identify the tag that indicates a registration record is associated with the sinkholer organization Kleissner & Associates (#cno.infra.sink.reg.kleissner).
- **Filter** the whois records to show **only** those associated with Kleissner:

```
inet:whois:rec:fqdn=youipcam.com
+#cno.infra.dns.sink.reg.kleissner
```

• The :created property on any of the records shows the registration date.

Return to Question 15

Question 15 - Bonus Notes

Whois records (inet:whois:rec) vs whois contacts (inet:whois:contact).
 Various Synapse Power-Ups can create domain whois data (such as synapse-nettools or synapse-whoxy). Power-Ups will always create a whois record (inet:whois:rec node); they may also create one or more inet:whois:contact nodes linked to the whois record.

Whether the contacts are created depends on the data returned from the external data source. If the data returned is **structured**, Synapse can parse the response to create the inet:whois:contact nodes. If the data returned is simply a text blob of the registration information, Synapse will only create the inet:whois:rec node (and store the text blob in the inet:whois:rec:text property).

Return to Question 15

Question 16 - Answer Explanations

- Q16 UI Answer
- Q16 Storm Answer

Q16 UI Answer

To find the whois email addresses:

• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** the FQDN youipcam.com:

inet:fqdn=youipcam.com

• In the **Results Panel**, **click** the **Explore button** to navigate to adjacent nodes:



 Use the Scroll to Form button to view or navigate to the inet:whois:email nodes:



To identify the registrant email used by APT1:

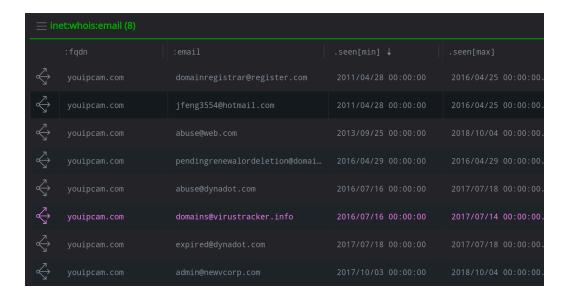
• In the **Results Panel**, click the .seen[min] column header to sort the inet:whois:email nodes in date order:



 We know from <u>Question 15</u> that Kleissner & Associates sinkholed the FQDN in July 2016, so APT1 must have controlled the FQDN before then.

Similarly, Mandiant reported the FQDN when they published the APT1 report in February 2013, so APT1 must have controlled the FQDN on or before that time.

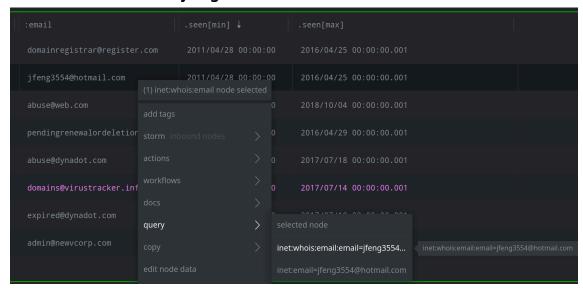
Only **four** of the email addresses were seen prior to July 2016, and only **one** (jfeng3554@hotmail.com) does not appear to be associated with a domain registrar, making it the most likely candidate:



To identify other FQDNs registered by this email address:

Continuing your previous navigation:

 In the Results Panel, right-click the email address jfeng3554@hotmail.com on the associated inet:whois:email node and select query > inet:whois:email:email=jfeng3554...



The context menu option allows you to easily ask Synapse about (**lift**) all inet:whois:email nodes whose :email property value is jfeng3554@hotmail.com.

• Synapse will run the new query for you and display the results:



Return to Question 16

Q16 Storm Answer

To find the whois email addresses:

Full query:

```
inet:fqdn=youipcam.com -> inet:whois:email
```

You can also lift the whois email records for the FQDN directly: inet:whois:email:fqdn=youipcam.com

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **Lift** the FQDN youipcam.com...

```
inet:fqdn=youipcam.com
```

...then pivot to the associated inet:whois:email nodes:

```
inet:fqdn=youipcam.com -> inet:whois:email
```

If you want to view both the original FQDN (to view the tags and any tag timestamps / dates) and the inet:whois:email records together, you can use the <u>pivot and join</u> operation:

```
inet:fqdn=youipcam.com -+> inet:whois:email
```

To identify the registrant email used by APT1:

• In the **Results Panel**, click the **.seen[min]** column header to **sort** the inet:whois:email nodes in date order.

 We know from <u>Question 15</u> that Kleissner & Associates sinkholed the FQDN in July 2016, so APT1 must have controlled the FQDN prior to that date.

Similarly, Mandiant reported the FQDN as an APT1 indicator when they published the APT1 report in February 2013, so APT1 must have controlled the FQDN on or before that time.

Only **four** of the eight email addresses were seen prior to July 2016, and only **one** (jfeng3554@hotmail.com) does not appear to be associated with a domain registrar, making it the most likely candidate.

To identify other FQDNs registered by this email address:

Full query:

```
inet:whois:email:email=jfeng3554@hotmail.com -> inet:fqdn
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the inet:whois:email nodes whose:email property value is the address we're interested in:

```
inet:whois:email=jfeng3554@hotmail.com
```

• ...then **pivot** to the associated FQDNs:

```
inet:whois:email:email=jfeng3554@hotmail.com -> inet:fqdn
```

Return to Question 16

Question 17 - Answer Explanations

- Q17 UI Answer
- Q17 Storm Answer
- Question 17 Bonus Notes

Q17 UI Answer

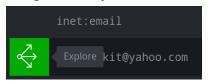
While parts of this challenge can be answered using the UI, some parts are most easily answered in Storm.

To find the FQDNs associated with the email address:

• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** the email address:

```
inet:email=issn.bgkit@yahoo.com
```

• In the **Results Panel**, click the **Explore button** next to the email address to navigate to adjacent nodes:

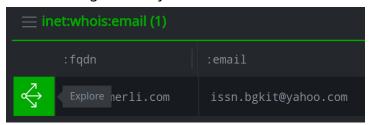


• Navigate to the inet:whois:email nodes (use **Scroll to Form** if necessary):

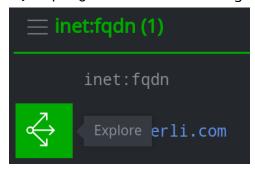


To view the whois records associated with the FQDN:

• In the **Results Panel**, click the **Explore button** next to the inet:whois:email node to navigate to adjacent nodes:



• From your new results, in the **Results Panel**, click the **Explore button** next to the FQDN progammerli.com to navigate to "connected" nodes:



View or navigate to the inet:whois:rec nodes using the Scroll to Form button:



To find only those whois records for the FQDN that contain the email address:

Because the email address appears in the raw text of the whois record the inet:whois:rec:text property) and not as its own property, there is no structured way for us to use the UI to view this email address (i.e., as its own column) or navigate from the whois records to "see" the associated details. This is a question that is answered most easily using Storm.

To try and find this answer using the UI alone, you could:

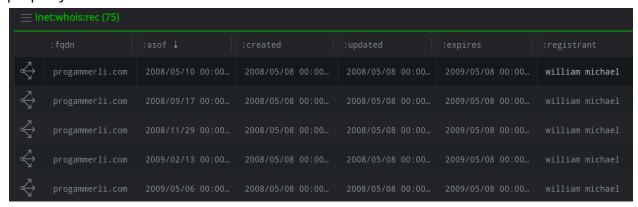
- **Sort** the inet:whois:rec nodes in date order (by the :asof column).
- **Select** a record and view its :text property to see if it contains the email address issn.bgkit@yahoo.com.

• When you locate a record with that email address, check other records that have the same **registration date** (:created property).

This "works" but is rather tedious - and where learning a bit of Storm comes in handy!

To identify the registrant name:

Once you've identified a whois record (or records) with the email address issn.bgkit@yahoo.com, you can use the **Results Panel** to view the :registrant property of the associated whois record:

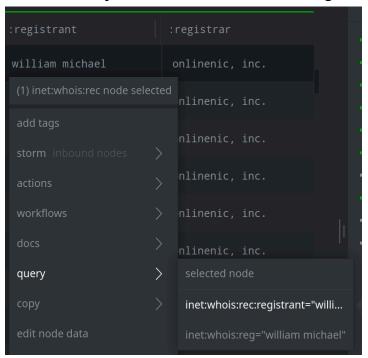


Fortunately, the person who registered this FQDN was very consistent with their registration information!

To find other whois records with the same registrant:

• In the **Results Panel, right-click** the **william michael** value in the :registrant column and select **query > inet:whois:rec:registrant="willi..."** from the context

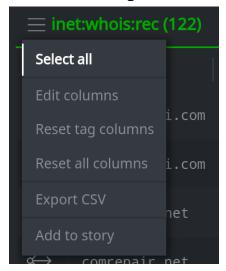
menu to lift **any / all** whois records with this registrant name:



Synapse will run the new query and display your results:



• Click the **hamburger menu** next to the **inet:whois:rec** header and select **Select all:**



 Click the **Explore button** next to any selected node to navigate to "connected" nodes:



• View or navigate to the associated FQDNs (inet:fqdn nodes):



Return to <u>Question 17</u>

Q17 Storm Answer

To find the FQDNs associated with the email address:

Full query:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the inet:whois:email nodes with the associated email address...

```
inet:whois:email=issn.bgkit@yahoo.com
```

• ...then **pivot** to the FQDNs:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
```

To view the whois records associated with the FODN:

• Building on your previous query, add a **pivot** from the FQDN to the whois records:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
-> inet:whois:rec
```

To find only those whois records for the FODN that contain the email address:

Building on your previous query, use Storm's <u>regular expression</u> comparison operator (~=) to add a **filter** so your results only include those records whose :text property contains the email address:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
-> inet:whois:rec +:text~=issn.bgkit@yahoo.com
```

To identify the registrant name:

Building on your previous query, **pivot** to the registrant (inet:whois:reg) node(s) to view the registrant(s):

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
   -> inet:whois:rec +:text~=issn.bgkit@yahoo.com
   -> inet:whois:reg
```

• ...then use the **uniq** command to de-duplicate your results:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
   -> inet:whois:rec +:text~=issn.bgkit@yahoo.com
   -> inet:whois:reg | uniq
```

To find other whois records with the same registrant and their FQDNs:

 Building on your previous query, **pivot** to other whois records with the same registrant...

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
   -> inet:whois:rec +:text~=issn.bgkit@yahoo.com
   -> inet:whois:reg | uniq | -> inet:whois:rec
```

• ...then **pivot** to the associated FQDNs...

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
   -> inet:whois:rec +:text~=issn.bgkit@yahoo.com
   -> inet:whois:reg | uniq | -> inet:whois:rec -> inet:fqdn
```

• ...then use the **uniq** command to de-duplicate the results:

```
inet:whois:email:email=issn.bgkit@yahoo.com -> inet:fqdn
  -> inet:whois:rec +:text~=issn.bgkit@yahoo.com
  -> inet:whois:reg | uniq | -> inet:whois:rec -> inet:fqdn
  | uniq
```

Return to <u>Question 17</u>

Question 17 - Bonus Notes

• Continuing a query vs. starting a new one. You've seen that Storm gives you many different options to eventually get "the same" answer. In this example, we illustrated how you can continually build on an existing Storm query by adding more operations (pivots, filters, commands). This allows you to follow your analysis where it takes you. You can easily build "complex" Storm queries by progressing step-by-step through your own thought process!

Of course, at any point you can create a new query with a different "starting point" based on the question you want to answer.

Return to Question 17

Question 18 - Answer Explanations

- Q18 UI Answer
- O18 Storm Answer
- Question 18 Bonus Notes

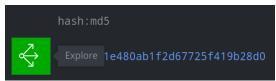
Q18 UI Answer

To identify the file:

• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** the MD5 hash:

```
hash:md5=3107de21e480ab1f2d67725f419b28d0
```

• In the **Results Panel**, click the **Explore button** next to the hash:md5 to navigate to adjacent nodes:



• In the **Results Panel**, view or navigate to the associated file:bytes node:

```
file:bytes

sha256:9ec9221f685b446874bb6dfc5509b4304f8d8b78b10fa3b8ba06cf4f505c0f84
```

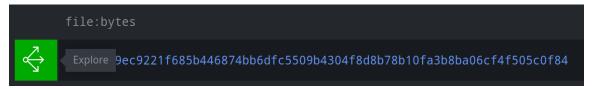
To identify the malware family:

- In the **Results Panel**, **select** the file:bytes node.
- In the **Details Panel, NODE** tab, view the tags associated with the file to identify the malware family:

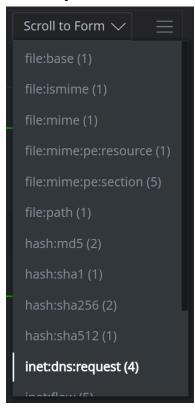
```
+ Add Tags
• rep.mandiant.apt1
• rep.mandiant.tarsip_eclipse
• rep.vt.checks_network_adapters
```

To identify the FQDNs:

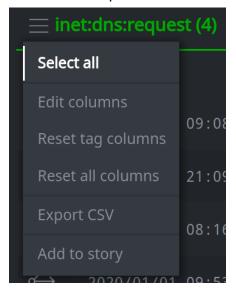
• In the **Results Panel**, click the **Explore button** next to the file:bytes node to navigate to adjacent nodes:



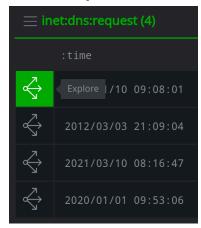
• View or navigate to the inet:dns:request nodes (use **Scroll to Form** if necessary):



• Use the **hamburger menu** next to the **inet:dns:request** header to **Select all** of the inet:dns:request nodes:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



• View or navigate to the inet:fqdn nodes (use **Scroll to Form** if necessary):



To see who reported the FQDNs:

• In the **Results Panel**, **select** a node and use the **Details Panel**, **NODE** tab to view any tags on the node (the colors provide a hint as to which node(s) have tags):



Return to Ouestion 18

Q18 Storm Answer

To identify the file:

• In the **Research Tool**, with the query bar in **Storm mode**, **lift** the file with the associated MD5 hash:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
```

You could also **lift** the MD5 and **pivot** to the file:

hash:md5=3107de21e480ab1f2d67725f419b28d0 -> file:bytes

To identify the malware family:

- In the Results Panel, select the file:bytes node and use the Details Panel,
 NODE tab to view tags on the node.
- Alternatively, you can build on your previous query to pivot to tags to view the tags:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0 -> #
```

Tip: See the **Question 18 Bonus Notes** for details on better distinguishing threat cluster tags from malware family tags.

To identify the FQDNs queried by the file:

Full query:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0 -> inet:dns:request
   -> inet:fqdn | uniq
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the file with the MD5 hash...

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
```

• ...then **pivot** to the DNS requests...

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
-> inet:dns:request
```

• ...then **pivot** to the FQDNs...

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
-> inet:dns:request -> inet:fqdn
```

• ...and use the **uniq** command to de-duplicate your results:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
  -> inet:dns:request -> inet:fqdn | uniq
```

To determine who reported an FQDN:

- In the **Results Panel**, select an inet:fqdn node and use the **Details Panel** to view tags on the node.
- Alternatively, you can build on your previous query to pivot to tags to view the tags:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0
  -> inet:dns:request -> inet:fqdn | uniq | -> #
```

 See the <u>Question 18 - Bonus Notes</u> for a discussion of using the threat intel portions of Synapse's data model to help identify reporting organizations based on tags.

Return to Question 18

Question 18 - Bonus Notes

• Malware sandbox data. When a file is executed in a malware sandbox, the sandbox records activity that occurs during execution. This should include activity performed by the sample itself, but may also include activity performed by other files extracted and executed by the original sample. Captured activity may even include actions that were not performed by the sample at all, but by other processes that are part of the sandbox environment. For example, a realistic sandbox running Microsoft Windows may coincidentally decide to run its weekly Windows Update cycle while you're testing your latest piece of malware.

This means that when a malware sandbox service provides an execution report that includes DNS queries, that typically means "a DNS query occurred while your sample was executing" but does not necessarily mean "your sample queried this FQDN". As an analyst, you may need to determine whether a particular query was made by your sample, by a different file that your sample dropped, or by some unrelated sandbox process.

• Tags and threats vs. tags and malware families. If the tags rep.mandiant.apt1 and rep.mandiant.tarsip_eclipse look the same, how can you tell whether the tag represents a threat cluster or a malware family?

One option is to include the distinction as part of the tag's definition (the :title and :doc properties on the syn:tag node). This is a reasonable approach (though it can get confusing when organizations use the same name for both a malware family and a threat cluster).

Another option is to take advantage of the threat intelligence / risk:* portion of the Synapse data model. You can create objects that represent threat clusters (risk:threat nodes) or malware families (risk:tool:software nodes), and associate the object with a tag (via the node's :tag property) that is used to

annotate and group nodes that are associated with the threat or malware family. If you're not sure whether a tag on a node represents one or the other (or both), you can pivot from the tags to the associated node(s) to find out.

In the example solution above, we viewed the tags associated with our file:bytes node to identify the malware family for the file. We assumed you "just knew" that APT1 was the threat cluster and TARSIP-ECLIPSE was the malware family:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0 -> #
```

A better solution would be to build on that query and **pivot** from the tags to the :tag property of any associated risk:tool:software nodes to find which tags (if any) represent malware:

```
file:bytes:md5=3107de21e480ab1f2d67725f419b28d0 -> #
-> risk:tool:software:tag
```

This confirms that TARSIP-ECLIPSE is a malware family reported by Mandiant:



Whether you use the threat intel portion of the Synapse data model is up to you. Some organizations simply use tags without creating risk:threat or risk:tool:software nodes; or they start by simply tagging nodes and later adopt the threat intel model. (Note that some Power-Ups will create threat intel-related objects automatically.) But the model gives you the ability to record additional detail about threats, malware families, and activity such as attacks or compromises. The model is <u>described in more detail</u> in the User Guide for the <u>vertex-threat-intel</u> Power-Up.

Tip: The threat intel parts of the Synapse data model are built into Synapse itself; you do not need to install the vertex-threat-intel Power-Up to use them. The Power-Up installs a visual Workflow that makes it easier to work with some of the existing model elements.

Return to <u>Question 18</u>

Question 19 - Answer Explanations

- Q19 UI Answer
- Q19 Storm Answer

Q19 UI Answer

To identify files that query earthsolution.org:

• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** the FQDN:

```
inet:fqdn=earthsolution.org
```

• In the **Results Panel**, click the **Explore button** next to the inet:fqdn to navigate to adjacent nodes:



• Use the **Scroll to Form** button to locate / navigate to any inet:dns:request nodes (there are none in the results):



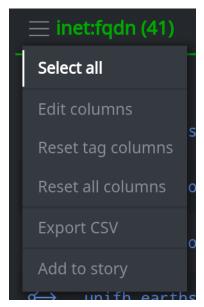
There are **no** inet:dns:request nodes associated with the FQDN earthsolution.org, meaning there are **no** files that query that specific FQDN (at least that Synapse knows about).

<u>To identify files that query any **subdomains** of earthsolution.org</u>:

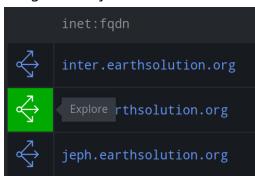
• In the **Research Tool, Tabular** display mode, use the query bar in **Storm mode** to **lift** all the FQDNs that are part of the earthsolution.org **zone**:

```
inet:fqdn:zone=earthsolution.org
```

• Click the **hamburger menu** to the right of the **inet:fqdn** header and choose **Select** all:



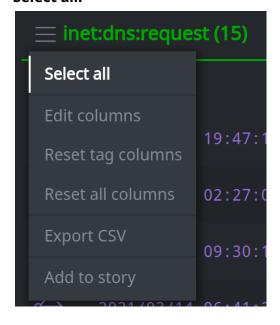
• In the **Results Panel**, click the **Explore button** next to any selected node to navigate to adjacent nodes:



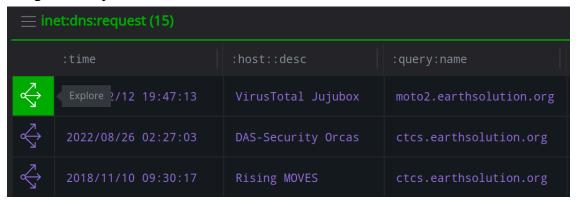
• Navigate to the inet:dns:request nodes (use **Scroll to Form** if necessary):



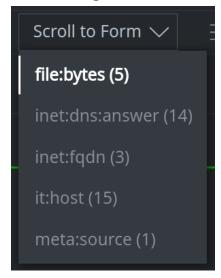
• Click the **hamburger menu** to the right of the **inet:dns:request** header and choose **Select all:**



• In the **Results Panel**, click the **Explore button** next to any selected node to navigate to adjacent nodes:



View or navigate to the file:bytes nodes (use Scroll to Form if necessary):

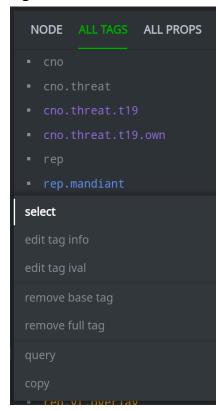


To identify the files reported by Mandiant:

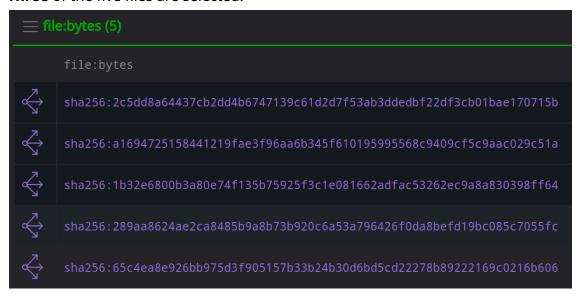
• In the **Results Panel**, **select** the individual nodes and view information about the node (including any tags) in the Details Panel.

Or:

• In the **Details Panel**, select the **ALL TAGS** tab. Locate the **click** the **rep.mandiant** tag and choose **select** from the menu to highlight nodes that have this tag:

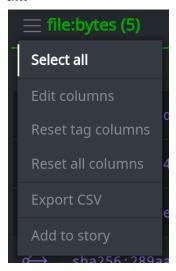


• Three of the five files are selected:

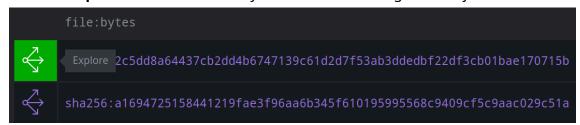


To identify other FQDNs queried by the files:

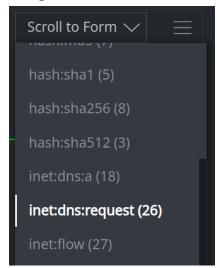
• Click the **hamburger menu** to the right of the **file:bytes** header and choose **Select** all:



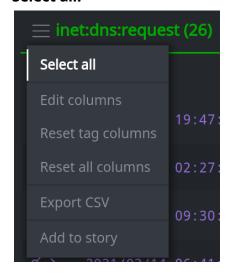
• Click the **Explore button** next to any selected file to navigate to adjacent nodes:



• Navigate to the inet:dns:request nodes (use **Scroll to Form** if necessary):



• Click the **hamburger menu** to the right of the **inet:dns:request** header and choose **Select all:**



• In the **Results Panel**, click the **Explore button** next to any selected node to navigate to adjacent nodes:



• Navigate to the inet:fqdn nodes (use **Scroll to Form** if necessary):



Return to Question 19

Q19 Storm Answer

To identify files that guery earthsolution.org:

Full query:

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the FQDN:

```
inet:fqdn=earthsolution.org
```

• ...then **pivot** to the DNS requests:

```
inet:fqdn=earthsolution.org -> inet:dns:request
```

• ...then **pivot** to the associated files:

```
inet:fqdn=earthsolution.org -> inet:dns:request -> file:bytes
```

•and use the **uniq** command to remove duplicate results:

Note: If you're building this query step by step, you will get **no** results (nodes) after your initial pivot to the inet:dns:request nodes (i.e., there are no DNS requests for this FQDN in Synapse). The final / full query will **also** return no nodes, but shows the full syntax for how to get an answer if you were composing and executing the full Storm query all at once.

To identify files that query any **subdomains** of earthsolution.org:

Full query:

```
inet:fqdn:zone=earthsolution.org -> inet:fqdn:zone
-> inet:dns:request -> file:bytes | uniq
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the FQDNs that are part of the zone:

```
inet:fqdn:zone=earthsolution.org
```

...then pivot to the DNS requests for any of those FQDNs:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
```

• ...then **pivot** to the associated files:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
   -> file:bytes
```

• ...and use the **uniq** command to remove duplicate results:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
   -> file:bytes | uniq
```

To identify the files reported by Mandiant:

• Add a **filter** to your existing query:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
   -> file:bytes | uniq | +#rep.mandiant
```

To identify other FQDNs queried by the files:

Full query:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
  -> file:bytes | uniq | -> inet:dns:request -> inet:fqdn
  | uniq
```

Step by step:

• To find **all** the FQDNs queried by these files, we can take the original query we used to find the files:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
   -> file:bytes | uniq
```

• ...then **pivot** from the files to their inet:dns:requests:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
   -> file:bytes | uniq | -> inet:dns:request
```

• ...then **pivot** to the FQDNs queried:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
-> file:bytes | uniq | -> inet:dns:request -> inet:fqdn
```

• ...and use the **uniq** command to remove duplicate results:

```
inet:fqdn:zone=earthsolution.org -> inet:dns:request
  -> file:bytes | uniq | -> inet:dns:request -> inet:fqdn
  | uniq
```

Return to Ouestion 19

Question 20 - Answer Explanations

- Q20 UI Answer
- O20 Storm Answer

Q20 - UI Answer

To find the files with PDB paths:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode,** run the following query to **lift** the APT1 files:

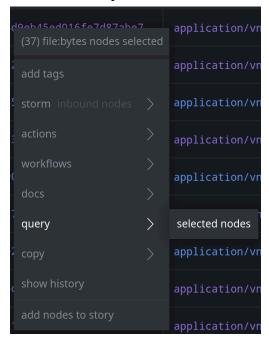
```
file:bytes#rep.mandiant.apt1
```

- Ensure the :mime:pe:pdbpath property is displayed as a column.
- **Click** the :mime:pe:pdbpath column header and **sort** the column in descending order (**Z->A**) to group the files with PDB paths together:



• Use your mouse to select **only** those files with PDB paths (i.e., using **shift-click** or **ctrl-click**).

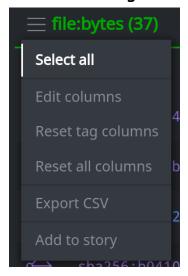
 Right-click any of the selected files and select query > selected nodes from the context menu (you should see 37 files selected):



 Synapse will load and run a new query to only lift the 37 selected files that have PDB paths.

To find the unique PDB paths:

• Click the hamburger menu next to the file:bytes header and choose Select all:



• **Click** the **Explore button** next to any selected node to navigate to adjacent nodes:



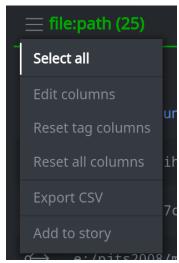
• Navigate to the file:path nodes (use **Scroll to Form** if necessary):



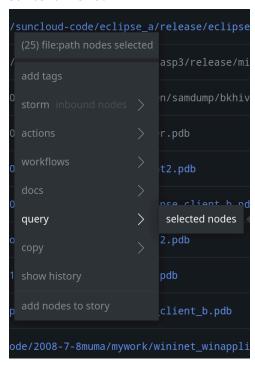
Note: While the UI provided your answer, when you "Explored" from the 37 files you navigated to more than 4,500 individual nodes to identify only 25 nodes that you care about. This is an example of where using Storm to ask the **specific** question that you want to answer is more efficient than Exploring **all** "connected" data in the UI.

To find paths **not** reported by Mandiant:

Click the hamburger menu next to the file:path header and choose Select all:



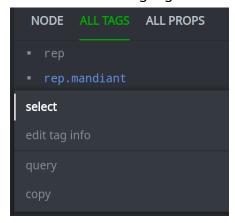
• **Right-click** any selected node and choose **query > selected nodes** from the context menu:



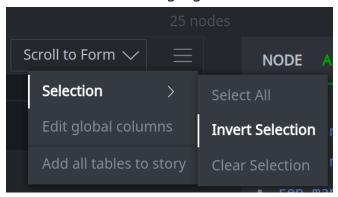
Synapse will load and run a new query to only lift the 25 selected file paths:

file:path=e:/xiaome/suncloud-code/eclipse_a/release/eclipse_client_b.pdb file:path=e:/code/moon1.5/re

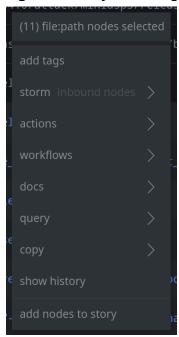
In the **Details Panel**, select the **ALL NODES** tab. **Click** the **rep.mandiant** tag and choose **select** to highlight the nodes **reported** by Mandiant:



Click the hamburger menu next to the Scroll to Form button and select Selection
 Invert Selection to highlight the nodes that do not have a rep.mandiant tag:

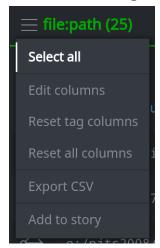


• **Right-click** any of the highlighted nodes to display the number of nodes selected:



To find other files with any of these PDB paths:

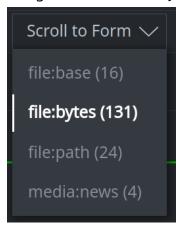
• Click the hamburger menu next to the file:path header and choose Select all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:

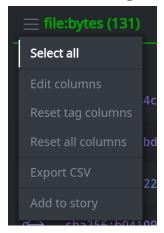


• Navigate to the file:bytes nodes (use **Scroll to Form** if necessary):

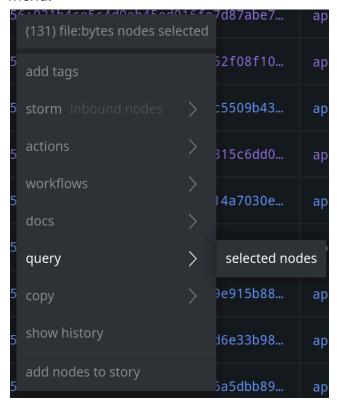


<u>To find which files were **not** reported by Mandiant, but reported by Symantec as "Comment Crew"</u>:

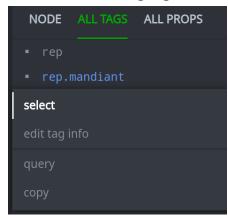
• Click the hamburger menu to the left of the file:bytes header and select Select all:



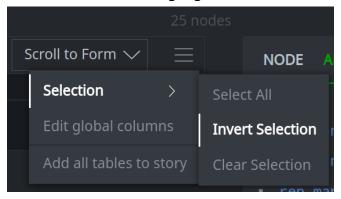
 Right-click any selected node and select query > selected nodes from the context menu:



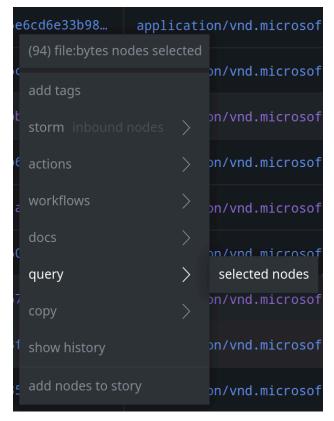
- Synapse will load and execute a new query to lift **only** the 131 files.
- In the **Details Panel**, select the **ALL TAGS** tab. **Click** the **rep.mandiant** tag and choose **select** to highlight the files **reported** by Mandiant:



Click the hamburger menu next to the Scroll to Form button and select Selection
 Invert Selection to highlight the nodes that do not have a rep.mandiant tag:

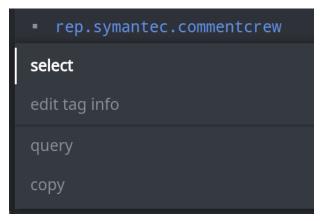


• **Right-click** any selected node and select **query > selected nodes** from the context menu:

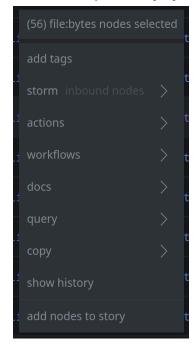


• Synapse will load and execute a new query to lift **only** the 94 files.

• In the **Details Panel, ALL TAGS** tab, **click** the **rep.symantec.commentcrew** tag and choose **select:**

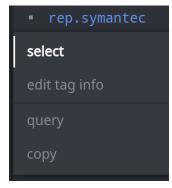


• **Right-click** any of the highlighted nodes to view the number of nodes selected (i.e., that were reported by Symantec as Comment Crew):

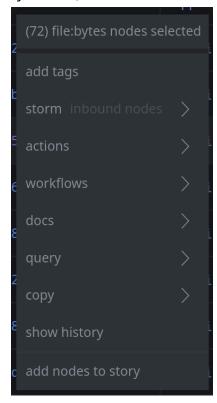


To find the files reported by Symantec in any capacity:

• In the **Details Panel, ALL TAGS** tab, **click** the **rep.symantec** tag and choose **select:**



• **Right-click** any selected file to view the number of nodes selected (i.e., reported by Symantec):



Return to <u>Question 20</u>

Q20 - Storm Answer

To find the files with PDB paths:

Full query:

```
file:bytes#rep.mandiant.apt1 +:mime:pe:pdbpath
```

Step by step:

• In the **Research Tool**, using the guery bar in **Storm mode**, first **lift** the APT1 files:

```
file:bytes#rep.mandiant.apt1
```

...then **filter** to those files with PDB paths:

```
file:bytes#rep.mandiant.apt1 +:mime:pe:pdbpath
```

To find the unique PDB paths:

Full query:

Step by step:

• Building on your previous query, first **pivot** to the file:path nodes...

```
file:bytes#rep.mandiant.apt1 +:mime:pe:pdbpath -> file:path
```

...then use the **uniq** command to remove duplicate results:

To find PDB paths **not** reported by Mandiant:

• Add a **filter** to see only those paths **not** reported by Mandiant:

To find other files with these PDB paths:

• **Remove** the filter you just applied. Add a **pivot** from the file paths to **any** files that have those PDB paths :

To find which files were **not** reported by Mandiant, but reported by Symantec as "Comment Crew":

• Add a **filter** to **remove** files reported by Mandiant...

• ...then add a **filter** to **show** files reported by Symantec as Comment Crew:

```
file:bytes#rep.mandiant.apt1 :mime:pe:pdbpath
  -> file:bytes:mime:pe:pdbpath | uniq | -#rep.mandiant.apt1
  +#rep.symantec.commentcrew
```

To find files were **not** reported Symantec in any capacity:

• ...modify your last filter to show all files reported by Symantec generally

```
file:bytes#rep.mandiant.apt1 :mime:pe:pdbpath
  -> file:bytes:mime:pe:pdbpath | uniq | -#rep.mandiant.apt1
  +#rep.symantec
```

Return to Question 20

Question 21 - Answer Explanations

- O21 UI Answer
- Q21 Storm Answer
- Question 21 Bonus Notes

Q21 UI Answer

Note: Typically we start our investigation by asking about (lifting) the object(s) we're interested in, and then navigating (i.e., using the Explore button) to view adjacent nodes.

In this case, if we start by lifting the 437 APT1 files (file:bytes#rep.mandiant.apt1), select all of the files, and then use the Explore button, we will find over **40,000** results - too many for our browser to display effectively.

This is a good example of where using Storm is far more efficient! If you still want to use the UI, you will need to get a bit creative with your "starting point" (initial query) and navigation.

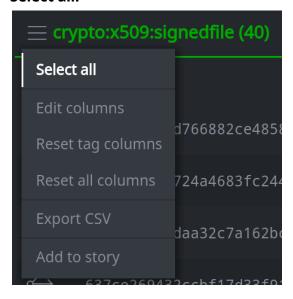
To find the signed APT1 files:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode,** run the following query to **lift** all crypto:x509:signedfile nodes:

crypto:x509:signedfile

We can optimize the performance of our queries by "lifting the smallest number of things first". In this case, there are fewer signed files (crypto:x509:signedfile nodes) than there are APT1 files (file:bytes#rep.mandiant.apt1). In addition, because we are specifically interested in signed files, lifting the crypto:x509:signedfile nodes allows us to start "closer" to our answer.

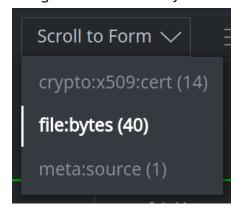
• Click the **hamburger menu** next to the **crypto:x509:signedfile** header and choose **Select all:**



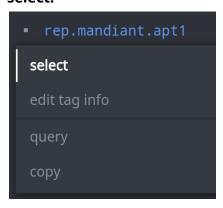
• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



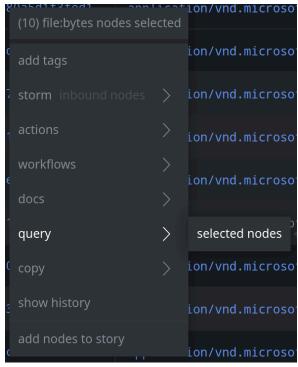
• Navigate to the file:bytes nodes (use **Scroll to Form** if necessary):



• In the **Details Panel, ALL TAGS** tab, click the rep.mandiant.apt1 tag and choose select:



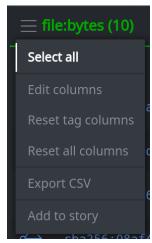
• **Right-click** any selected file and select **query > selected nodes:**



• Synapse will load and run a new query to lift **only** the signed files reported by Mandiant.

To find the unique certificates used to sign the files:

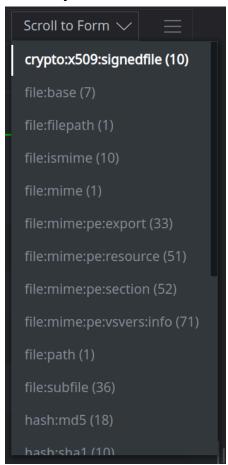
• Click the **hamburger menu** next to the **file:bytes** header and choose **Select all:**



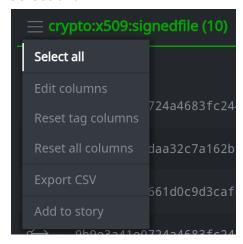
• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



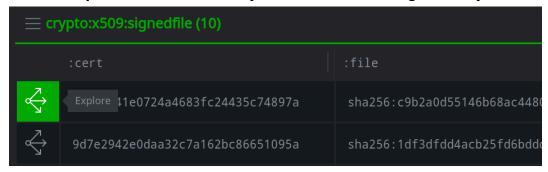
 Navigate to the crypto:x509:signedfile nodes (use Scroll to Form if necessary):



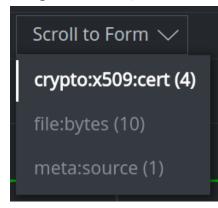
Click the hamburger menu next to the crypto:x509:signedfile header and choose
 Select all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:

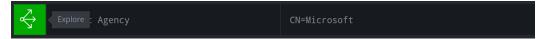


• Navigate to the crypto:x509:cert nodes (use **Scroll to Form** if necessary):



<u>To find all files signed with the "Microsoft" certificate:</u>

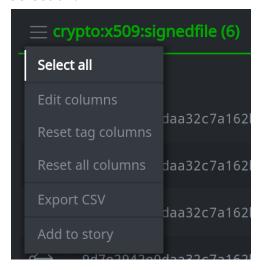
 Click the Explore button next to the crypto:x509:cert whose :subject is CN=Microsoft:



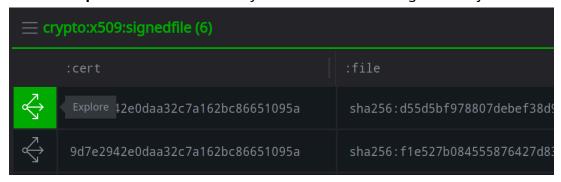
 Navigate to the crypto:x509:signedfile nodes (use Scroll to Form if necessary):



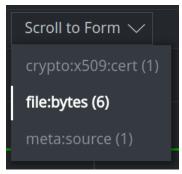
• Click the **hamburger menu** next to the **crypto:x509:signedfile** header and choose **Select all:**



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



Navigate to the file:bytes nodes (use Scroll to Form if necessary):



To the files that were **not** publicly reported by Mandiant or Symantec:

 Select individual nodes and use the **Details Panel** to view tags on the nodes (or display tags in a column).

Tip: The tag colors may provide a hint as to which files were not reported.

To see if any files are associated with specific malware families:

• Use the **Details Panel**, **ALL TAGS** tab to look for tags associated with malware families:



Return to Question 21

Q21 Storm Answer

To find the signed files:

Full query:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
  -> file:bytes
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the APT1 files:

```
file:bytes#rep.mandiant.apt1
```

• ...then **pivot** to the crypto:x509:signedfile nodes:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
```

• ...and **pivot** to only those files associated with a "signed file" node:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
   -> file:bytes
```

Bonus:

• You can also find the signed APT1 files using a subquery filter:

```
file:bytes#rep.mandiant.apt1 +{ -> crypto:x509:signedfile }
```

See the **Question 21 - Bonus Notes** for more details on subquery filters.

To find the certificates used to sign the files:

Full query:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
  -> crypto:x509:cert | uniq
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the APT1 files:

```
file:bytes#rep.mandiant.apt1
```

• ...then **pivot** to the crypto:x509:signedfile nodes:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
```

• ...then **pivot** to the certificates:

```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
   -> crypto:x509:cert
```

• ...then deduplicate the results with the **uniq** command:

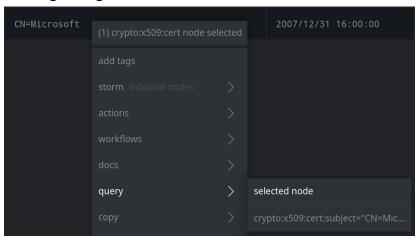
```
file:bytes#rep.mandiant.apt1 -> crypto:x509:signedfile
   -> crypto:x509:cert | uniq
```

To find all files signed with the "Microsoft" certificate:

While we could continue our previous query (i.e., by adding a **filter** to show only the X509 certificate with the specified subject), in this case we'll start a new Storm query to lift that certificate as our starting point.

Since we already have the certificate displayed in our Results Panel, we can:

• **Right-click** the certificate we're interested in and select **query > selected node** to lift just the crypto:x509:cert node. (This method is particularly useful when working with quid-based nodes!):



Full query:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the certificate node:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a

You can lift this certificate using the right-click method above.
```

• ...then **pivot** to any associated "signed file" nodes:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile
```

• ...and **pivot** to the signed files:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes
```

To identify files **not** reported by either Mandiant or Symantec:

• Add **filters** to your query to **exclude** any files reported by Mandiant or Symantec:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes -#rep.mandiant
-#rep.symantec
```

To identify files that are associated with malware families:

• **Remove** the filters from your previous query to display all of the signed files:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes
```

• In the **Details Panel, ALL TAGS** tab, view the associated tags to identify any malware families.

Tip: As described in the **Question 18 - Bonus Notes**, we can get a better answer as to whether the files are associated with any malware families by pivoting from the tags on the signed files to any malware families (risk:tool:software nodes) associated with those tags.

• Add a **pivot** to your previous query to pivot to the tags on the files:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes -> #
```

• ...then use the **uniq** command to remove duplicate results:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
-> crypto:x509:signedfile -> file:bytes -> # | uniq
```

• ...then **pivot** to any risk:tool:software nodes associated with those tags:

```
crypto:x509:cert=9d7e2942e0daa32c7a162bc86651095a
  -> crypto:x509:signedfile -> file:bytes -> # | uniq
  | -> risk:tool:software:tag
```

The query gives us **no** results, indicating the files are not associated with any malware families.

Return to Ouestion 21

Question 21 - Bonus Notes

• <u>Subquery filters.</u> To find **only** the APT1 files that are signed, we started with the APT1 files, then pivoted to the "signed file" nodes, and then pivoted "back" to **only** those files that are signed. We had to perform two pivots to end up with a subset of the files we started with.

This is because there is nothing about a file:bytes node itself that tells us whether the file is "signed" - there is nothing we can use to filter those files **directly** to **only** show the files that are signed.

Instead we need to check some "nearby" nodes - specifically, the adjacent crypto:x509:signedfile nodes. If a file:bytes node has an associated crypto:x509:signedfile node, we know it is a signed file. But in order to determine this, I need to navigate (pivot) over to those nodes and then pivot back again.

This is where **subquery filters** are useful. Standard filters in Storm allow you to filter based on some aspect of a **node itself**, such as a property or a tag. In contrast, subquery filters allow you to filter your **current** set of nodes based on some property (or tag) that exists on **"nearby" nodes** - even nodes that may be two or more pivots away!

In our subquery example above, the subquery filter acts as a sort of "what if" operation. The plus sign (+) tells us this is an "include" filter (just like a standard filter). The Storm inside the curly braces acts as our "what if". For **each** of the APT1 files, "**what if** I tried to pivot to an adjacent crypto:x509:signedfile node? Would there be one?" If the answer is yes, **include** that file in my results. If the answer is no, do not include (drop) the file.

Synapse "checks" this condition by performing the "what if" pivot for you in the background, and (because this is an "include" filter) only returning / displaying those APT1 files that do, in fact, have an associated crypto:x509:signedfile node.

Once you become familiar with them, subquery filters are an extremely powerful tool to help you ask (and answer) very precise questions!

• Extracting certificate data from files. The Synapse-Fileparser Power-Up can parse (extract) a broad range of data and / or metadata from various files and file types (MIME / Media types). One of the Synapse-Fileparser's many capabilities is that it can recognize, extract, and model x509 certificates that are present in a file blob. For files that contain multiple certificates (i.e., a certificate chain), Fileparser will make a "best effort" to identify the specific certificate used to sign the file, and model this as a crypto:x509:signedfile node. (Any additional certificates in the signing chain, if present (and parseable) are modeled as subfiles (file:subfile nodes) to show they were contained in the original file.)

Synapse-Fileparser does not **validate** signatures in any way, it simply looks for x509 content within a file blob and extracts that content into its own file(s) if found. This means that Fileparser also cannot distinguish between files that are actually signed and files that may have a certificate stored as a resource, appended to the file, etc.

Return to <u>Question 21</u>

Question 22 - Answer Explanations

- Q22 UI Answer
- Q22 Storm Answer

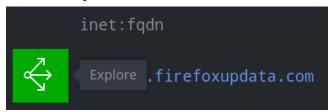
Q22 UI Answer

To find the articles:

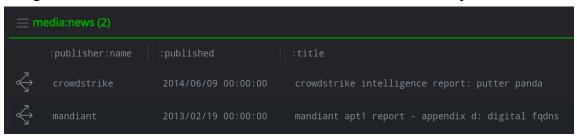
In the Research Tool, Tabular display mode, with the query bar in Storm mode,
 lift the FQDN:

inet:fqdn=update8.firefoxupdata.com

• Click the **Explore button** next to the node to navigate to adjacent nodes:



• Navigate to the media: news nodes (use **Scroll to Form** if necessary):



Return to <u>Question 22</u>

Q22 Storm Answer

To find the articles:

Full query

```
inet:fqdn=update8.firefoxupdata.com <(refs)- media:news</pre>
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the FQDN:

```
inet:fqdn=update8.firefoxupdata.com
```

• ...then **traverse** the light edges back to any media: news nodes:

```
inet:fqdn=update8.firefoxupdata.com <(refs)- media:news</pre>
```

Note: because the articles "reference" the FQDN you need to traverse the 'refs' light edge "backwards".

Return to Question 22

Question 23 - Answer Explanations

- Q23 UI Answer
- Q23 Storm Answer

Q23 UI Answer

To find the threat clusters / cluster names:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode**, first **lift** the APT1 indicators:

```
#rep.mandiant.apt1
```

• In the **Details Panel**, **ALL TAGS** tab, look for tags that start with #cno.threat:

```
cno.threat.t15
cno.threat.t15.own
cno.threat.t15.own.seed
cno.threat.t19
cno.threat.t19.own
cno.threat.t19.own.seed
```

Return to Question 23

Q23 Storm Answer

To find the threat clusters / cluster names:

Full query:

```
#rep.mandiant.apt1 -> # +syn:tag^=cno.threat | uniq
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the APT1 indicators:

```
#rep.mandiant.apt1
```

• ...then <u>pivot to tags</u> to pivot from the indicators to the associated tags:

```
#rep.mandiant.apt1 -> #
```

• ...then <u>filter by prefix</u> to limit the resulting syn:tag nodes to ones that start with cno.threat:

```
#rep.mandiant.apt1 -> # +syn:tag^=cno.threat
```

• ...then use the **uniq** command to de-duplicate results:

```
#rep.mandiant.apt1 -> # +syn:tag^=cno.threat | uniq
```

Return to <u>Question 23</u>

Question 24 - Answer Explanations

• Question 24 - Bonus Notes

Q24 UI and Storm Answer

Note: as the solutions to this challenge all involve basic **lift** operations (lift by tag), the answers are the same for both the UI and Storm.

To find the indicators Vertex asserts are **owned or controlled** by T15:

• In the **Research Tool, Tabular** display mode, using the query bar in **Storm mode, lift** the nodes owned or controlled by T15:

```
#cno.threat.t15.own
```

To find the "starting point" ("seed" node) for T15:

• **Lift** the seed node(s) for T15:

```
#cno.threat.t15.own.seed
```

To find the indicators **used** by T15:

• **Lift** the nodes used by T15:

```
#cno.threat.t15.use
```

To find **all** the indicators associated with T15:

• Lift all nodes associated with (controlled or used) by T15:

```
#cno.threat.t15
```

To determine whether T15 is a coherent cluster:

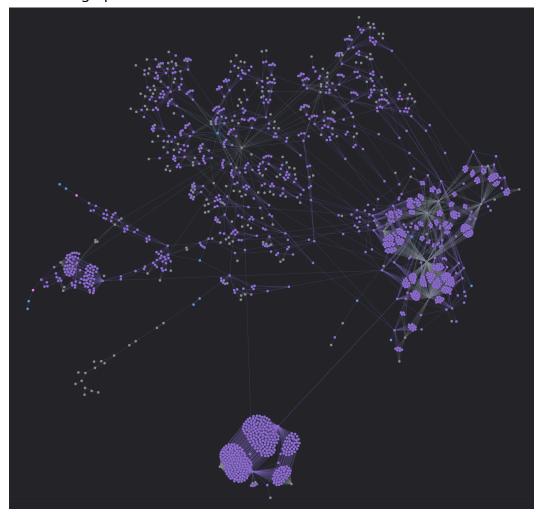
• In the **Research Tool**, use the **display mode selector** to switch to **Force Graph** mode:



• In **Force Graph** display mode, using the query bar in **Storm mode**, **lift** all the nodes associated with T15:

```
#cno.threat.t15
```

Your force graph should look similar to this:



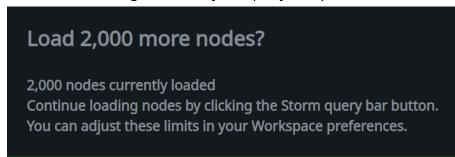
Force Graph mode generates a familiar "directed graph" projection from the Synapse data. Assuming that a threat cluster should represent a set of indicators and activity that are all **connected** in some way, a cluster should form a coherent (interconnected) force graph projection (i.e., all nodes are linked in some way; no isolated nodes or small disconnected clusters are present).

To determine whether APT1 is a coherent cluster:

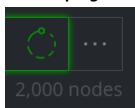
• In **Force Graph** mode, with the query bar in **Storm mode**, **lift** all the nodes associated with APT1:

#rep.mandiant.apt1

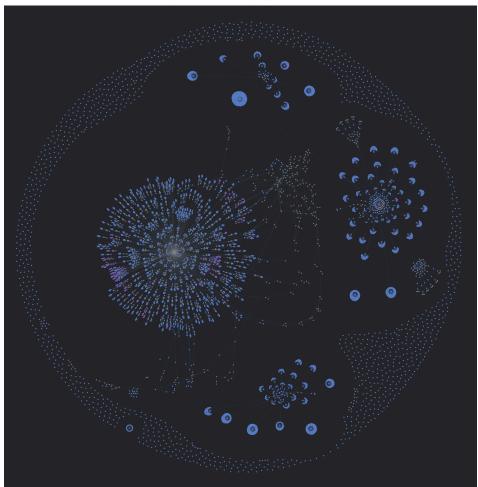
Note: There are nearly 7,000 nodes to display; Force Graph mode will prompt you to continue loading data until your query completes:



Click the **progress** icon in the query bar to continue loading data:







While much of Mandiant's APT1 data is connected in some way, there are a number of smaller, isolated clusters as well as numerous individual nodes (seen around the edges of the Force Graph). Based on Mandiant's publicly reported data, APT1 is not a fully interconnected cluster.

Return to Question 24

Question 24 - Bonus Notes

- Threat clusters and threat cluster tags. The solution for this challenge question was not difficult to find, but it is meant to expose you to some of the concepts that Vertex uses for tracking threat clusters and cluster-related data.
 - Our analysis vs. third party reporting. Vertex clusters are based on Vertex analysts directly reviewing our internal data in order to make clustering

decisions. While we monitor and ingest third-party reporting, we can't always **verify** other organizations' analysis. So we use tags to separate "other people's reporting" (rep.<reporter>.<name>, for example), from our own first-hand, verified analysis (cno.threat.<name>).

- Owned vs. used. Based on our experience tracking hundreds of thousands of indicators associated with thousands of threat clusters, we find it useful to distinguish between:
 - things a group "owns" (or controls), that can be considered unique to them (and that are pretty much guaranteed to be malicious / bad if you see them); and
 - things that a group "uses" but that may not be unique to that group, may also be used by other groups, and may not even be malicious.

For example, a malicious binary with an embedded C2 FQDN registered and controlled by a threat group is "owned" by that group. A copy of the publicly available Mimikatz tool that the group leveraged to capture user credentials is "used" by them. We use tags to distinguish the two assertions.

"Seed" nodes. Threat clusters grow and expand over time as we observe and learn more about related activity. Unfortunately, our analysis and assessments are not always perfect. Down the road, we may identify discrepancies in our threat clusters, such as an FQDN that we have attributed to two separate clusters.

One possibility is that the FQDN represents a point of overlap that indicates the two separate clusters are "the same" group, and we have the initial evidence that may allow us to merge the two clusters.

Another possibility is that the clusters really are separate and somewhere along the way one or more indicators were mis-attributed. We now have a potential mess that needs to be untangled.

Under these circumstances, it is helpful to have a **seed node** (or nodes) to show where the whole cluster began. This makes it easier for an analyst to retrace the clustering process and identify where something may have been added to the cluster in error. Being able to go back to the seed node is much

easier than picking an arbitrary starting point in an 80,000 node cluster and trying to figure out what happened!

- "Coherent" clusters. A cornerstone of Vertex's approach to analysis is that threat clusters and threat groups should be literal clusters of activity any nodes associated with a threat cluster or threat group should form a contiguous, interconnected graph. If there are disconnects in our cluster, this indicates:
 - The cluster is **not** actually a coherent or connected set of activity; we have likely mis-attributed something; or
 - We have failed to model (or tag) additional data that would make the cluster fully coherent.

"Does it have a coherent graph" is not the only test for whether we have attributed activity correctly, but it provides a good means to visually check our work.

• "Coherent" clusters and public reporting. Most third party reporting does not include enough information to create a coherent cluster. Reporting typically includes lists of indicators (hashes, domains, etc.) that do not form a connected graph when viewed.

This does not mean that public reporting is "wrong" - just that most organizations fail to include sufficient information for others to verify their analysis. You can choose to take the reporting at face value or attempt to validate it on your own (e.g., by attempting to "fill in" gaps in the graph by enriching indicators and establishing more connections).

Return to Question 24

Question 25 - Answer Explanations

- O25 UI Answer
- Q25 Storm Answer

Q25 UI Answer

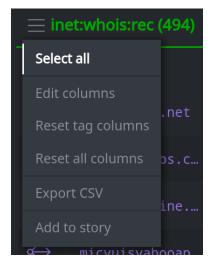
To find the whois records:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode**, **lift** the whois records associated with T15:

```
inet:whois:rec#cno.threat.t15.own
```

To find the total number of FQDNs:

• Click the **hamburger menu** next to the **inet:whois:rec** header and choose **Select** all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:

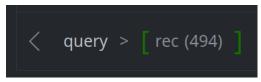


• Navigate to the inet:fqdn nodes (use **Scroll to Form** if necessary):

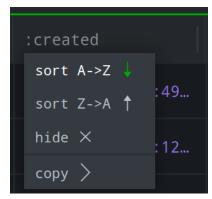


To find the earliest registered FQDN:

• In your **breadcrumbs**, click **query** to return to your original query:



• **Click** the **:created** column header and select **sort A->Z** to sort in ascending order:



Locate the earliest registered FQDN:



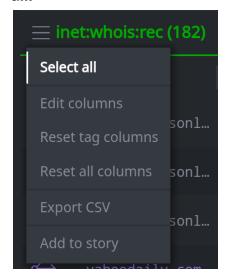
To find other FQDNs registered on the same date:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode, lift** all the domain whois records that were created on 2009/09/08 (i.e., between 2009/09/08 and 2009/09/09):

```
inet:whois:rec:created@=(2009/09/08,2009/09/09)
```

Note: Synapse also accepts "wildcard" syntax for dates, so you could also use the following query: inet:whois:rec:created=2009/09/08*

Click the hamburger menu next to the inet:whois:rec header and choose Select
 all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



• Navigate to the inet:fqdn nodes (use **Scroll to Form** if necessary):



To identify FQDNs associated with APT1 but not T15:

• Use the **Details Panel** to view tags on individual nodes. (The **Tag Colors** may help to distinguish between cno.threat.* and rep.* nodes).

Return to Question 25

Q25 Storm Answer

To find the whois records:

• In the **Research Tool**, using the query bar in **Storm mode**, **lift** the whois records associated with T15:

```
inet:whois:rec#cno.threat.t15.own
```

To find the earliest registered FQDN:

• Use the **min** command to find a whois record for the FQDN with the earliest registration:

```
inet:whois:rec#cno.threat.t15.own | min :created
```

• Then **pivot** to the FQDN:

```
inet:whois:rec#cno.threat.t15.own | min :created | -> inet:fqdn
```

To find other FQDNs registered on the same date:

Full query:

```
inet:whois:rec:created@=(2009/09/08, 2009/09/09) -> inet:fqdn | uniq
```

Step by step:

• Using the query bar in **Storm mode**, <u>Lift by time or interval</u> to find whois records created on September 8, 2009:

```
inet:whois:rec:created@=(2009/09/08, 2009/09/09)
```

Note: Synapse also accepts "wildcard" syntax for dates, so you could also use the following query: inet:whois:rec:created=2009/09/08*

...**pivot** to the FQDNs...

```
inet:whois:rec:created@=(2009/09/08, 2009/09/09) -> inet:fqdn
```

...and deduplicate the results with the **uniq** command:

To identify the FQDN:

Full query:

Step by step:

 Modify your previous query by adding a **filter** to your results to identify the FQDNs associated with APT1...

• ...but **not** associated with T15:

Return to Ouestion 25

Analytical Challenge 4 Notes

Each organization (and each analyst!) may have their own criteria for linking indicators or activity - deciding that indicators are related or associated with the same threat cluster.

With thousands of FQDNs registered each day, the simple fact that a set of domains were registered on the same date is probably not enough to assert the domains are "related" (even if "Mandiant says so"). Most analysts will look for additional correlation points. These may include things like:

- Other similarities, such as use of the same domain registrar or the same registrant information (if available).
- Whether the domains were registered very close in **time** in addition to being registered on the same date.
- Network infrastructure that might link the FQDNs (such as shared DNS A records or SSL/TLS certificates).
- Malware that uses both this FQDN and a known/attributed T15 FQDN for C2.

The specific criteria used to make a decision may vary by organization and the available data. Clustering assertions generally come down to identifying a sufficient number of data points that the "preponderance of evidence" makes it more likely that the indicators or activity are related vs. distinct

Return to Analytical Challenge 4

Advanced Challenges

Question 26 - Answer Explanations

- Q26 UI Answer
- Q26 Storm Answer

Q26 UI Answer

To find the files:

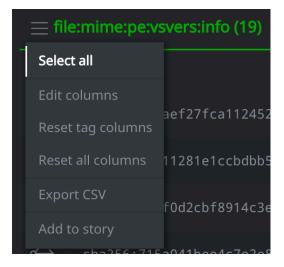
• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode**, **lift** the PE metadata node:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
```

• Click the **Explore button** next to the node to navigate to adjacent nodes:



• Click the **hamburger menu** next to the **file:mime:pe:vsvers:info** header and choose **Select all:**



Navigate to the file:bytes nodes in the results (use Scroll to Form if needed).

To find the unreported files:

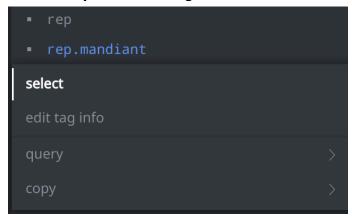
This question is more easily answered with Storm (using **filters**). However, you can:

• In the **Details Panel**, on the **ALL TAGS** tab, view all the tags present on all the nodes:



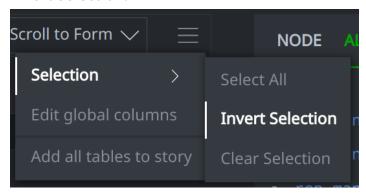
Note that the only security organization reporting on any of the files is Mandiant.

• Click the rep.mandiant tag and choose select:



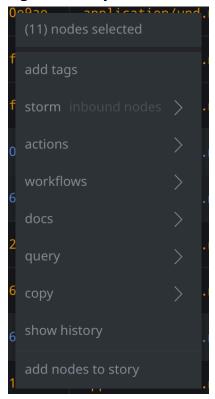
This selects / highlights all the files reported by Mandiant.

• Click the main hamburger menu (next to Scroll to Form) and choose Selection > Invert Selection:



This selects / highlights files **not** reported by Mandiant (but which may have been present in or reported by VirusTotal).

• **Right-click** any selected file to see the number of files selected:



Return to Question 26

Q26 Storm Answer

To find the files:

Full query:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
  -> file:mime:pe:vsvers:info -> file:bytes
```

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** the PE metadata node:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
```

• ...then **pivot** to the file:mime:pe:vsvers:info nodes:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
   -> file:mime:pe:vsvers:info
```

• ...then **pivot** to the associated file:bytes nodes:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
  -> file:mime:pe:vsvers:info -> file:bytes
```

To find the unreported files:

Full query:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
  -> file:mime:pe:vsvers:info -> file:bytes -#rep.mandiant
```

Step by step:

- Use the **Details Panel**, **ALL TAGS** tab to view all of the tags on all the nodes.
- Based on the tags present, add a **filter** to your previous results to exclude files associated with Mandiant's reporting:

```
file:mime:pe:vsvers:keyval=(OriginalFilename, JpgAsp.exe)
  -> file:mime:pe:vsvers:info -> file:bytes -#rep.mandiant
```

Return to Question 26

Question 27 - Answer Explanations

- Q27 UI Answer
- Q27 Storm Answer

Q27 UI Answer

N/A

Tip: Synapse is able to run queries and answer questions over very large data sets in a way that is still fast and efficient. You can "ask about" or navigate (pivot, etc.) tens of thousands or even hundreds of thousands of nodes without any problem.

That said, your browser will have difficulty attempting to **display** large numbers of results. Recall that when you use the **Explore button** to navigate, you "explore" to **all** adjacent ("connected") nodes - even ones you may not care about. If you start with a large number of nodes (such as 2,000+ FQDNs) and those nodes are "highly connected", you can quickly bog down your browser.

When working with large data sets, it is usually easier to ask (and answer) your exact question using Storm. Even if Synapse navigates "through" large numbers of nodes when running your query, a reasonable number of results can still be displayed as your "final" answer.

Return to <u>Question 27</u>

Q27 Storm Answer

To find the files:

Full query:

Step by step:

• In the **Research Tool**, with the query bar in **Storm mode**, first **lift** both the APT1 and Comment Crew FQDNs:

```
inet:fqdn#rep.mandiant.apt1 inet:fqdn#rep.symantec.commentcrew
```

• ...then use the **uniq** command to remove duplicates:

• ...then **pivot** to the DNS requests for the FQDNs:

•then **pivot** to the files that made the requests:

• ...and use the **uniq** command to remove duplicates:

To find the files that are unreported:

Full query:

```
inet:fqdn#rep.mandiant.apt1 inet:fqdn#rep.symantec.commentcrew
  | uniq | -> inet:dns:request -> file:bytes | uniq | -#cno
  -#rep.fireeye -#rep.mandiant -#rep.symantec
```

Step by step:

- In the **Details Panel**, use the **ALL TAGS** tab to view all of the tags on all of the nodes.
 - The tags include those associated with Vertex reporting (cno.threat) as well as reporting by several third-party security companies.
- Add a **Filter** to your previous query to remove files associated with Vertex reporting:

```
inet:fqdn#rep.mandiant.apt1 inet:fqdn#rep.symantec.commentcrew
  | uniq | -> inet:dns:request -> file:bytes | uniq | -#cno
```

• ...and add additional **filters** to remove files associated with other public reporting:

```
inet:fqdn#rep.mandiant.apt1 inet:fqdn#rep.symantec.commentcrew
  | uniq | -> inet:dns:request -> file:bytes | uniq | -#cno
  -#rep.fireeye -#rep.mandiant -#rep.symantec
```

Note: In our instance of Synapse, all "third party" reporting - whether from a security company (like Mandiant) or a data source (like VirusTotal or Shodan) - uses the rep tag prefix.

For purposes of our question ("how many files were not publicly reported"), we don't consider "the file was found in VirusTotal" to mean the same thing as "a security company reported on this file". In order to **keep** the VT files (rep.vt) but exclude files that appeared in blogs or other reports, we need to use filter syntax similar to the example above.

To find the FQDNs:

Full query

Step by step:

• Using the query bar in **Storm mode**, first **lift** both the APT1 and Comment Crew files:

```
file:bytes#rep.mandiant.apt1
  file:bytes#rep.symantec.commentcrew
```

• ...then use the **uniq** command to remove duplicates:

```
file:bytes#rep.mandiant.apt1
  file:bytes#rep.symantec.commentcrew | uniq
```

...then pivot to the DNS requests made by the files:

• ...then **pivot out** to the FQDNs gueried

• ...then use the **uniq** command to remove duplicates:

```
file:bytes#rep.mandiant.apt1
  file:bytes#rep.symantec.commentcrew | uniq
    | -> inet:dns:request -> inet:fqdn | uniq
```

To find the FQDNs that are unreported:

Full query:

```
file:bytes#rep.mandiant.apt1 file:bytes#rep.symantec.commentcrew
  | uniq | -> inet:dns:request -> inet:fqdn | uniq | -#cno
  -#rep.fireeye -#rep.mandiant -#rep.symantec
```

Step by step:

- In the **Details Panel**, use the **ALL TAGS** tab to view all of the tags on all of the nodes.
 - The tags include those associated with Vertex reporting (cno.*) as well as reporting by several third-party security companies.
- Add a **Filter** to your previous query to remove files associated with Vertex reporting:

...and add additional filters to remove files associated with other public reporting:

Return to <u>Question 27</u>

Question 28 - Answer Explanations

- Q28 UI Answer
- Q28 Storm Answer

Q28 UI Answer

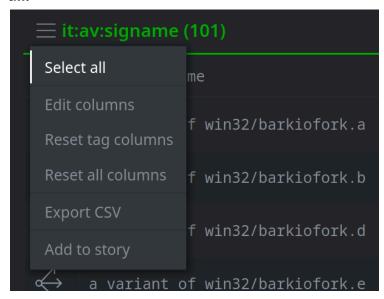
To find the signatures:

• In the **Research Tool, Tabular** display mode, with the query bar in **Storm mode, lift** the signature names that include the string 'barkiofork':

```
it:av:signame~=barkiofork
```

To find the software packages / vendors:

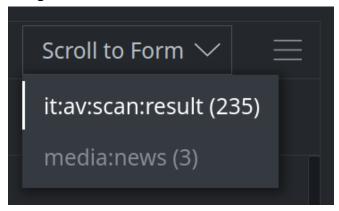
• Click the **hamburger menu** next to the **it:av:signame** header and choose **Select** all:



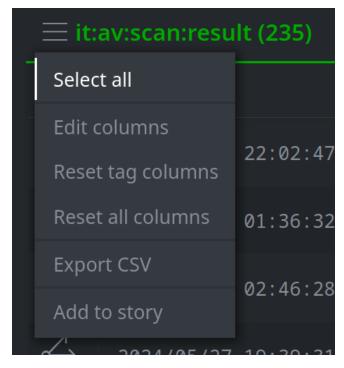
• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



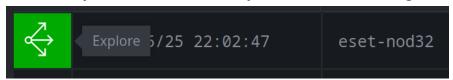
• Navigate to the it:av:scan:result nodes (use **Scroll to Form** if needed):



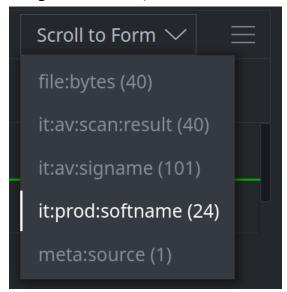
• Click the **hamburger menu** next to the **it:av:scan:result** header and choose **Select** all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:

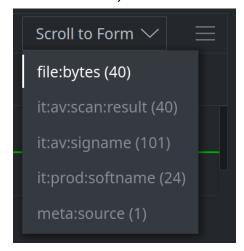


Navigate to the it:prod:softname nodes (use Scroll to Form if needed):



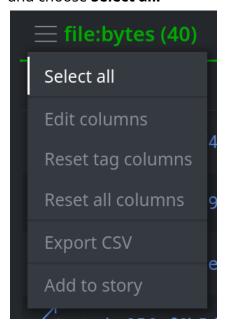
To find the detected files:

• In your current set of results, navigate to the file:bytes nodes (use **Scroll to Form** if needed):

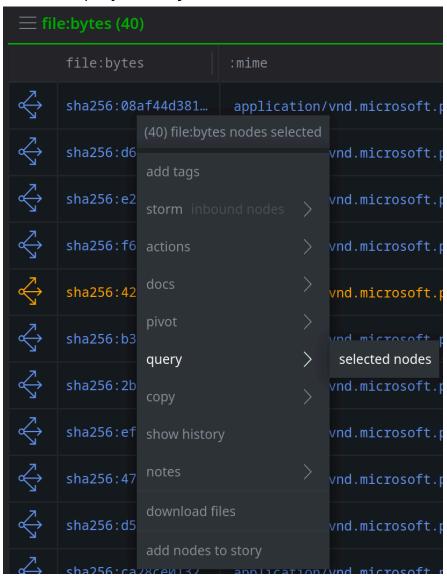


To find the associated malware families:

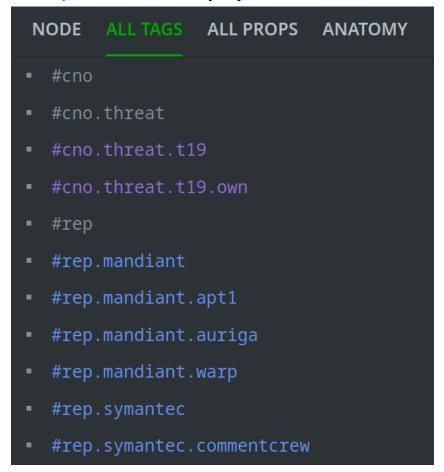
• From your current results, click the **hamburger menu** next to the **file:bytes** header and choose **Select all:**



 Right-click any selected file and select query > selected nodes to load and run a new Storm query that only asks about the detected files:



• In the **Details Panel**, use the **ALL TAGS** tab to view all the tags on all of the file:bytes nodes to identify any malware families:



Return to Question 28

Q28 Storm Answer

To find the signatures:

• Using the query bar in **Storm mode**, <u>lift by regular expression</u> to find the signature names that include 'barkiofork':

it:av:signame~=barkiofork

To find the software packages / vendors:

Full query:

Step by step:

• From your original query, **pivot** to the scan results:

```
it:av:signame~=barkiofork -> it:av:scan:result
```

• ...then **pivot** from the results to the scanner / engine names:

```
it:av:signame~=barkiofork -> it:av:scan:result
-> it:prod:softname
```

• ...and use the **uniq** command to remove duplicate results:

```
it:av:signame~=barkiofork -> it:av:scan:result
   -> it:prod:softname | uniq
```

To find the detected files:

Full query:

```
it:av:signame~=barkiofork -> it:av:scan:result -> file:bytes | uniq
```

Step by step:

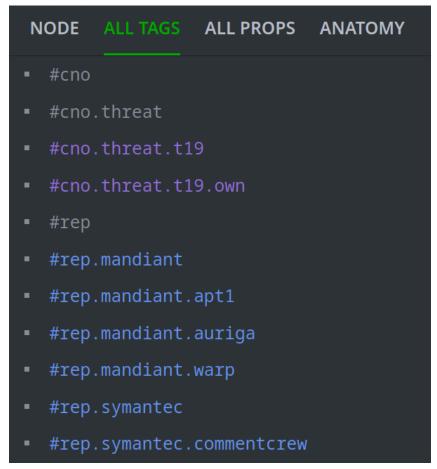
 Modify your query to **pivot** from the scan results to the associated files (instead of the scanner / engine names):

```
it:av:signame~=barkiofork -> it:av:scan:result -> file:bytes
```

• ...then use the **uniq** command to remove duplicate results:

To find the associated malware families:

• In the **Details Panel**, use the **ALL TAGS** tab to view all the tags on the nodes to identify any malware families:



Better Answer:

Because this instance of Synapse uses the rep.* tag tree for all third-party reporting (whether threat clusters, malware families, or other assertions) it may not be clear which tags represent malware families.

We can get a better answer by using the threat intel portions of Synapse's data model, which represents reported threat clusters and malware families as nodes (risk:threat and risk:tool:software, respectively). This allows you to pivot from the tags on any nodes to any malware families (risk:tool:software nodes) the tags represent.

Full query:

Step by step:

• Once you have identified the files detected by the 'barkiofork' AV signatures...

• ...Pivot to the tags on the files:

```
it:av:signame~=barkiofork -> it:av:scan:result -> file:bytes
| uniq | -> #
```

• ...use the **uniq** command to remove duplicate results:

```
it:av:signame~=barkiofork -> it:av:scan:result -> file:bytes
| uniq | -> # | uniq
```

• ...then **pivot** from the tags to the any risk:tool:software nodes associated with those tags (the risk:tool:software:tag property):

Return to Question 28

Analytical Challenge 5 Notes

Based on the data in Synapse, signatures referencing the malware family name "barkiofork" detect samples that Mandiant classifies as two different malware families.

This could indicate a number of things - for example:

- Mandiant may classify malware families differently from other companies / vendors (that is, other vendors may consider "barkiofork" to be a single malware family, where Mandiant considers those same samples to be two different families); or
- Some samples may be incorrectly identified as "barkiofork" by the various scanning engines/signatures.

Vendor detection logic (such as antivirus signatures) is often proprietary "black box" technology. We rarely have visibility into the exact logic a vendor uses to detect and classify malicious code. This is true regardless of the detection method - e.g., whether signature-based, execution-based, machine learning, etc. (Shared open-source rules, such as YARA rules, are an exception.)

The names of signatures that detect malware may provide a clue as to what a detected sample **might** be. Higher confidence that a signature **is correct** typically requires that you have confidence in the detection logic - either because you have visibility into the logic itself (such as a YARA rule), or you have tested enough samples against a signature to have confidence that the signature is reliable.

Return to Analytical Challenge 5

Question 29 - Answer Explanations

- Q29 UI Answer
- O29 Storm Answer
- Q29 Storm Answer Alternate Answer
- Question 29 Bonus Notes

Q29 UI Answer

N/A

Return to Question 29

Q29 Storm Answer

To find the files with any malicious antivirus / antimalware detection:

Full query:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:verdict=malicious }
```

Step by step:

• Using the query bar in **Storm mode**, first **lift** the APT1 files...

```
file:bytes#rep.mandiant.apt1
```

• ...then use a <u>subquery filter</u> to determine which files have associated detection data (i.e., which files "would" successfully pivot to **any** it:av:scan:result nodes if you actually executed the pivot):

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result }
```

• ...then **refine** your subquery filter to only include scan results where the :verdict is malicious:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:verdict=malicious }
```

To find the files detected as malicious by ten or fewer vendors:

 Add a mathematical comparison operator ("less than or equal to ten") to your subquery filter to determine which files have ten or fewer associated malicious verdicts:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:verdict=malicious }<=10</pre>
```

To find the files detected as malicious by ten or fewer vendors but at least two vendors:

• Add a second **subquery filter** and mathematical comparison ("greater than or equal to two"):

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:verdict=malicious }<=10 +{ -> it:av:scan:result
    +:verdict=malicious }>=2
```

Return to **Question 29**

Q29 Storm Answer - Alternate Answer

Synapse uses it:av:scan:result nodes to record an individual scan results from a single scanner / scan engine (using the :verdict property). Synapse can use the same form to record **collective** results from multiscanner services (such as VirusTotal) that provide summary results from multiple scanners. If you have access to multiscanner data, you could also answer this question based on collective results (using the it:av:scan:result:multi:count:* properties).

<u>To find the files with any malicious antivirus / antimalware detection, based on multiscanner data:</u>

Full query:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:multi:count:malicious }
```

Step by step:

Using the query bar in **Storm mode**, first **lift** the APT1 files...

```
file:bytes#rep.mandiant.apt1
```

• ...then use a <u>subquery filter</u> to determine which files have any associated detection data (i.e., which files "would" successfully pivot to **any** it:av:scan:result nodes if you actually executed the pivot):

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result }
```

• ...then **refine** your subquery filter to only include multiscanner results with malicious verdicts where the malicious count is greater than zero:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:multi:count:malicious>0 }
```

To find the files detected as malicious by ten or fewer vendors:

• Modify your query to specify that the number of malicious verdicts must be less than or equal to 10 (note that this will include nodes where the count is zero):

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:multi:count:malicious<=10 }</pre>
```

To find the files detected by ten or fewer vendors but at least two vendors:

 Add a second filter inside the subquery to ensure that files with ten or fewer malicious verdicts have at least two malicious verdicts:

```
file:bytes#rep.mandiant.apt1 +{ -> it:av:scan:result
    +:multi:count:malicious<=10 +:multi:count:malicious>=2 }
```

Return to Ouestion 29

Question 29 - Bonus Notes

We first described **subquery filters** in the <u>Question 21 - Bonus Notes</u>. Subquery filters are a powerful feature of Storm. They allow you to filter your current set of nodes based on the characteristics of **nearby** nodes. One way to think of subqueries is "what if" actions - what if I performed the Storm operations inside the subquery? Would my current set of nodes have any results?

A "basic" subquery can be used to determine **if** a Storm operation or set of operations would return **any** results for the source nodes - just as in the first part of this challenge's solution, we identified the APT1 files that have **any** malicious AV detection (it:av:scan:result nodes with :verdict=malicious).

You can also use mathematical operators (e.g., "greater than or equal to", "less than", or simply "equals") to also ask "what if..." about the **number** of results that would be returned if you ran the Storm inside the subquery.

In this challenge, we use subqueries with mathematical comparisons to find files with **low AV detection rates.** "Low detection" may help us find new malware not yet detected by many vendors, or malware with generally poor detection rates. Requiring "some minimal" level of detection (in this case, "at least two" vendors) may help to exclude any non-malicious (presuumably benign) files that have been submitted to AV scanning / multi-scanning services.

Return to <u>Question 29</u>

Question 30 - Answer Explanations

- Q30 UI Answer
- Q30 Storm Answer
- Question 30 Bonus Notes

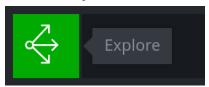
Q30 UI Answer

To find the files:

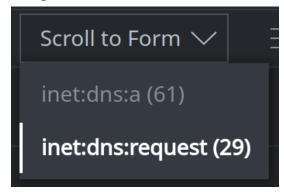
In the Research Tool, Tabular display mode, with the query bar in Storm mode,
 lift the FQDN 'tom-pc':

```
inet:fqdn=tom-pc
```

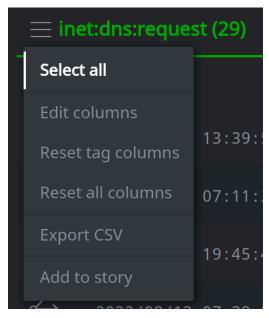
• Click the **Explore button** next to the FQDN to navigate to adjacent nodes:



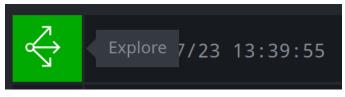
• Navigate to the inet:dns:request nodes (use **Scroll to Form** if necessary):



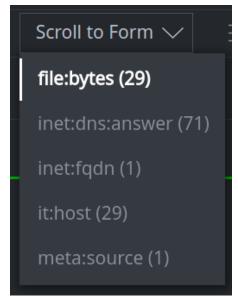
• Click the **hamburger menu** next to the **inet:dns:request** header and choose **Select** all:



• Click the **Explore button** next to any selected node to navigate to adjacent nodes:



• Navigate to the file:bytes nodes (use **Scroll to Form** if necessary):



To find the malware families:

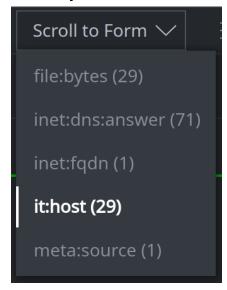
• In the **Details Panel**, use the **ALL TAGS** tab to view the tags on the nodes:

rep.mandiant
rep.mandiant.apt1
rep.mandiant.biscuit
rep.mandiant.seasalt
rep.mandiant.tabmsgsql
rep.mandiant.tarsip_eclipse
rep.mandiant.tarsip_moon
rep.mandiant.webc2_yahoo
rep.mcafee
rep.mcafee.seasalt
rep.symantec
rep.symantec.commentcrew

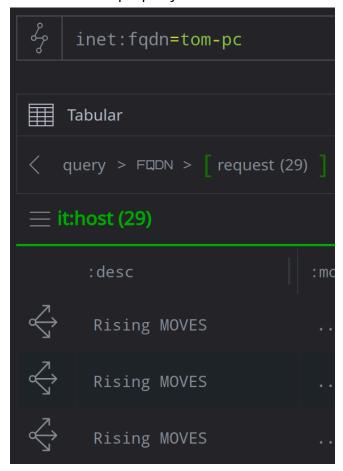
See the **Q30 Storm Answer** for a better way to identify malware families specifically.

To find the hosts:

• In your **current results**, navigate to the it:host nodes (use **Scroll to Form** if necessary):



• View the :desc property on the nodes:



Return to Ouestion 30

Q30 Storm Answer

To find the files:

Full query:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq
```

You could also **lift** the DNS requests directly: **inet:dns:request:query:name:fqdn=tom-pc**

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the FQDN 'tom-pc':

```
inet:fqdn=tom-pc
```

• ...pivot to the DNS requests:

```
inet:fqdn=tom-pc -> inet:dns:request
```

• ...**pivot** to the files that make the requests

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes
```

• ...and use the **uniq** command to de-duplicate results:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq
```

To find the malware families:

Full query:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq | -> #
    -> risk:tool:software:tag | uniq
```

Step by step:

• Building on your previous query, **pivot to tags** to show the syn:tag nodes for the tags on the files:

• ...then pivot to any malware family nodes (risk:tool:software) associated with those tags:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq
| -> # -> risk:tool:software:tag
```

• ...and use the **uniq** command to remove duplicate results:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq
| -> # -> risk:tool:software:tag | uniq
```

• You can optionally pivot again to just the **names** (it:prod:softname nodes) of the malware families:

```
inet:fqdn=tom-pc -> inet:dns:request -> file:bytes | uniq
   | -> # -> risk:tool:software:tag | uniq
   | :soft:name -> it:prod:softname
```

To find the hosts:

Full query:

```
inet:fqdn=tom-pc -> inet:dns:request -> it:host | uniq
```

Step by step:

• Using the guery bar in **Storm mode**, first **lift** the FQDN 'tom-pc':

```
inet:fqdn=tom-pc
```

...pivot to the DNS requests:

```
inet:fqdn=tom-pc -> inet:dns:request
```

...pivot to the hosts that make the requests

```
inet:fqdn=tom-pc -> inet:dns:request -> it:host
```

...and use the uniq command to de-duplicate results:

```
inet:fqdn=tom-pc -> inet:dns:request -> it:host | uniq
```

Return to **Question 30**

Question 30 - Bonus Notes

• it:host nodes. In this example, we noted that an it:host node is used to represent an instance of a malware sandbox. In this case, the it:host node and associated execution data are automatically created and modeled by Synapse Power-Ups that query various malware / sandbox services, such as VirusTotal or Hybrid Analysis. Using an it:host for a particular sandbox vendor and / or sandbox configuration gives you the flexibility to use Synapse and Storm to ask about all the things a malware binary does (regardless of the sandbox where it was executed) or to ask about the specific activity observed by a particular sandbox (if you want to know about differences between sandboxes, or exactly what was observed in a given sandbox).

An it:host node is generic enough to represent any kind of "host" (device) - malware sandbox or networked computer, real or virtual. The host does not even have to be a computer (server / desktop / laptop / tablet) - it could be a router, mobile device, IoT device, etc. We try to make the data model both as accurate (to "real world data") as possible and as flexible as possible so that the data model can be used for the broadest possible set of use cases - we never want to "model ourselves into a corner"!

Return to Question 30

Question 31 - Answer Explanations

- Q31 UI Answer
- O31 Storm Answer

Q31 UI Answer

N/A

Return to Question 31

Q31 Storm Answer

To find the file paths for 'file add' operations:

Full query:

```
file:bytes#rep.mandiant.apt1 -> it:exec:file:add -> file:path | uniq
```

Step by step:

• In the **Research Tool**, using the query bar in **Storm mode**, first **lift** the APT1 files:

```
file:bytes#rep.mandiant.apt1
```

• ...pivot out to the "file add" operations:

```
file:bytes#rep.mandiant.apt1 -> it:exec:file:add
```

...pivot out to the file paths:

```
file:bytes#rep.mandiant.apt1 -> it:exec:file:add -> file:path
```

• ...and use the **uniq** command to remove duplicates:

To find the file names for 'file add' operations:

Full query:

Step by step:

• Building on your previous query, **pivot** from the file paths to the file names...

• ...and use the **uniq** command to remove duplicates:

To find the Word and PDF file names for 'file add' operations:

• Building on your previous query, use a <u>compound filter</u> to limit your results to files that have a .doc **or** a .pdf extension

To find the file paths and file names for 'file write' operations:

Use the same queries as above, replacing it:exec:file:add with it:exec:file:write.

Extra Credit:

The challenge with the extra credit question is that it asks you to find an answer by pivoting through two different objects in Synapse: it:exec:file:add and it:exec:file:write nodes.

When you **pivot** in Synapse, you specify a particular target. For example:

```
file:bytes#rep.mandiant.apt1 -> it:exec:file:add
```

With standard pivot syntax, you can't pivot to two different targets; there is no "compound pivot syntax" for example, so we **can't** do something like:

```
file:bytes#rep.mandiant.apt1 -> (it:exec:file:add or
  it:exec:file:write)
```

Fortunately we can use the Synapse <u>tee</u> command to "pivot to both". **Tee** allows you to execute two sets of Storm operations in parallel, and join the results. This allows us to pivot to **both** it:exec:file:add and it:exec:file:write nodes (via two separate queries) and return the results as a single set of nodes.

Full query:

```
file:bytes#rep.mandiant.apt1 | tee {-> it:exec:file:add }
    { -> it:exec:file:write } | -> file:path | uniq | -> file:base
    | uniq | +(:ext=doc or :ext=pdf)
```

Step by step:

• First **lift** the APT1 files:

```
file:bytes#rep.mandiant.apt1
```

• ...then use **tee** to **pivot** to **both** the file add and file write operations:

```
file:bytes#rep.mandiant.apt1 | tee { -> it:exec:file:add }
    { -> it:exec:file:write }
```

Note that the results of this query include **both** it:exec:file:add and it:exec:file:write nodes.

• ...then use the same operations as above to get your answer (pivot to the associated file paths; remove duplicates; pivot to the associated file names; remove duplicates; and filter to only those file names with doc or pdf extensions):

```
file:bytes#rep.mandiant.apt1 | tee {-> it:exec:file:add }
    { -> it:exec:file:write } | -> file:path | uniq
    | -> file:base | uniq | +(:ext=doc or :ext=pdf)
```

Return to Question 31

Question 31 - Bonus Notes

• **File names.** You may have noticed that some of the file names we identify seem like reasonable decoy file names ('us hesitant in confirming north korean launch.pdf'). Others are more cryptic ('5176c625fda6f180da748f001c789deb.pdf'). Sandboxes simply report what they see, which may depend on the sandbox environment, the executable code used to create and name the decoy file, etc. For example, some decoy names may be hard-coded by the software developer; other names may be generated via a variable (e.g., the equivalent of "create a decoy file with the same name as the executable binary that writes the file"), which could result in odd file names if a malware sample was renamed before being executed.

Return to Question 31

Question 32 - Answer Explanations

Ouestion 32 - Bonus Notes

Q32 Answer

Note: There is no "UI specific" or "Storm specific" answer to this question. We hope you were able to explore the data using all available tools to research this challenge!

Using the **Explore button** to navigate from the original inet:user=uglygorilla node will show connections to:

- Email address uglygorilla@163.com
- Web accounts at rootkit.com and pudn.com.

- Email account uglygorilla@163.com leads to the domain whois record (via inet:whois:email) for FQDN hugesoft.org (and associated Threat Cluster T19 nodes and indicators).
- The pudn.com account includes:realname="汪东", linked to the person (ps:person) with names "汪东" and WANG Dong.

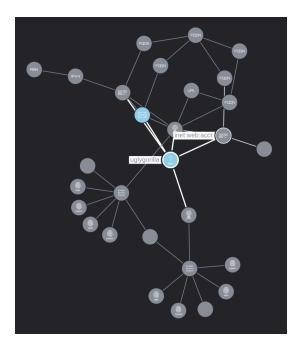
Storm queries using wildcard pivots can also help you explore and attempt to identify "connected" nodes or related data:

```
inet:user=uglygorilla <- *
inet:email=uglygorilla@163.com <- *</pre>
```

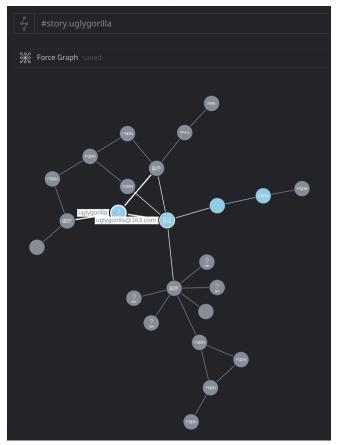
You can also use the **Explore** option from **Force Graph** display mode to explore / expand out from the original inet:user=uglygorilla node:



Selecting **explore node** in **Force Graph** mode will perform the same **Explore** operation as Tabular mode (technically, a combined <u>wildcard pivot out</u> / <u>wildcard pivot in</u> / <u>wildcard traverse all light edges</u>). However, Force Graph mode performs an additional wildcard pivot out from all nodes returned by your original query. (This results in additional potential connections displayed beyond those you specifically "asked about".)



Finally if you **tag nodes** associated with your exploration, you can view the tagged nodes in the **Force Graph** display mode to examine possible connections:



Note: Remember, if you use Synapse's **fork** and **merge** options and perform your research (and potential tagging of nodes, etc.) in a forked view, you have the option to either preserve your work (**merge** it into your production Synapse) or discard it (**delete** the fork when you're finished). This means you can research, test, and explore without worrying about mixing your research data with production data and analysis!

These are simply a few options for conducting more open ended research and analysis!

Return to Question 32

Question 32 - Bonus Notes

• Modeling more than indicators. True attribution - the ability to trace computer and network activity reliably to a real-world person or organization - requires the ability to show connections between traditional digital IOCs and indicators, and actions carried out by real people. The crossover could be as slight as a personal phone number used for operational activity, a network connection accidentally made from an IP address on a network registered to a real-world organization, or a digital purchase made with a personal credit card or cryptocurrency account.

The good news is Synapse can capture **all** of that with its data model. You're not limited to only representing malware behavior, DNS resolutions, or network or file system activity.

- Attribution what? If you're new to Synapse, the "uglygorilla" data in Synapse may have been a bit challenging to puzzle out without referencing the prose in the APT1 report describing Wang Dong and his activity. That's okay prose reporting plays an important role in our ability to convey analytical findings to different audiences. The important point is that we were able to use Synapse to capture more than just Mandiant's technical indicators. Vertex analysts were able to take Mandiant's descriptions of Wang Dong's accounts and activity and represent those in Synapse as well, where they become queryable parts of the Synapse data store. More importantly, they're literally linked in the data model so if someone questions your prose analysis and conclusions, the supporting connections are present in Synapse!
- What about those binaries? In the APT1 report, Mandiant noted that the name "UglyGorilla" was also referenced in strings present in some malware binaries:

"v1.0 No Doubt to Hack You, Writed by UglyGorilla, 06/29/2007"

'UglyGorilla' exists within some files (file:bytes nodes) as an arbitrary string, and arbitrary strings aren't extracted and modeled automatically within Synapse (specifically to avoid cluttering up the data store with a lot of potentially irrelevant strings data). However, if you wanted to specifically link those binaries to a string node you could do so manually using a light edge.

For example, assuming the file (file:bytes) below contained the string 'UglyGorilla', we could create a light edge between the file and the string using Storm's edit mode to add the light edge (query wraps):

```
file:bytes=sha256:cebb47280cd00814e1c085c5bc3fbac0e9f9116899909
1f199a1b1d209edd814 [ +(refs)> { it:dev:str='UglyGorilla' } ]
```

This would allow us to ask about files that contain the string 'UglyGorilla' using the following Storm query:

```
it:dev:str='UglyGorilla' <(refs)- file:bytes</pre>
```

(Recall that light edges have a "direction", but we can query / traverse them in either direction.)

Even better, using the **Synapse-YARA Power-Up**, you could write a YARA rule that checks for that string and links to any files that contain it via it:app:yara:match nodes!

Return to Question 32