Workbook

SANS | GIAC CERTIFICATIONS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org

All reference links are operational in the browser-based delivery of the electronic workbook.

Welcome to the FOR509 Electronic Workbook

E-Workbook Overview

This electronic workbook contains all lab materials for SANS FOR509, Enterprise Cloud Forensics and Incident Response. Each lab is designed to address a hands-on application of concepts covered in the corresponding courseware and help students achieve the learning objectives the course and lab authors have established.

Some of the key features of this electronic workbook include the following:

- Convenient copy-to-clipboard buttons at the right side of code blocks
- Inline drop-down solutions, command lines, and results for easy validation and reference
- Integrated keyword searching across the entire site at the top of each page
- Full-workbook navigation is displayed on the left and per-page navigation is on the right of each page
- Many images can be clicked to enlarge when necessary

Enjoy!

Authors:

- David Cowen | dlcowen@gmail.com | @hecfblog
- Pierre Lidome | plidome@gmail.com | @Texaquila
- Josh Lemon | jlemon@sans.org | @joshlemon
- Megan Roddie | megansroddie@gmail.com | @megan_roddie

Using the E-Workbook

You can access the SANS FOR509 workbook from your host system by connecting to the IP address of your VM. Run `ip a` in the SOF-ELK console to get the IP address of your VM (also shown on the console screen). Next, in a browser on your host machine, connect to the URL using that IP address (i.e. `http://<%VM-IP-ADDRESS%>`).

We hope you enjoy the SANS FOR509 class and workbook! To get the most out of your lab time in class, we recommend following the guidance in How to Approach the Labs.

Acknowledgements

The authors would like to acknowledge the support and contributions of the following people that assisted in making the FOR509 class the polished course it is today. Without their love and support, there would be a lot more typos, unusual screenshots, and odd labs. Thank you for all your support.

- Phil Hagen
- Chad Tilbury

- Benjamyn Whiteman
- Arjun Bhardwaj
- Beta 1 Class (June 2021)

Trademarks

Lab 1.1: Visualize Data in SOF-ELK®

Objectives

- Learn about the Discover, Visualize, and Dashboard analytics tabs
- Create filters and searches
- Create a dashboard

Background

SOF-ELK is a powerful platform. In this tutorial, we will focus on the elements you will need to complete the FOR509 labs. We encourage you to experiment on your own and load your own data in the future.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2020-04-01 00:00 UTC** to **now**.

Load Data

> **Note**
>
> Raw logs for all labs are on the VM. Feel free to use them in your own tools.

The Microsoft 365 Unified Access Log has been exported from the portal and saved in `/home/elk_user/lab-1.1_source_evidence.zip`

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract `ual.csv` from the ZIP file with the following commands:

Command lines

```
cd ~
unzip -q lab-1.1_source_evidence.zip -d /logstash/office365
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you have 5,462 documents loaded in the `office365` index:

Command line

```
sof-elk_clear.py -i list
```

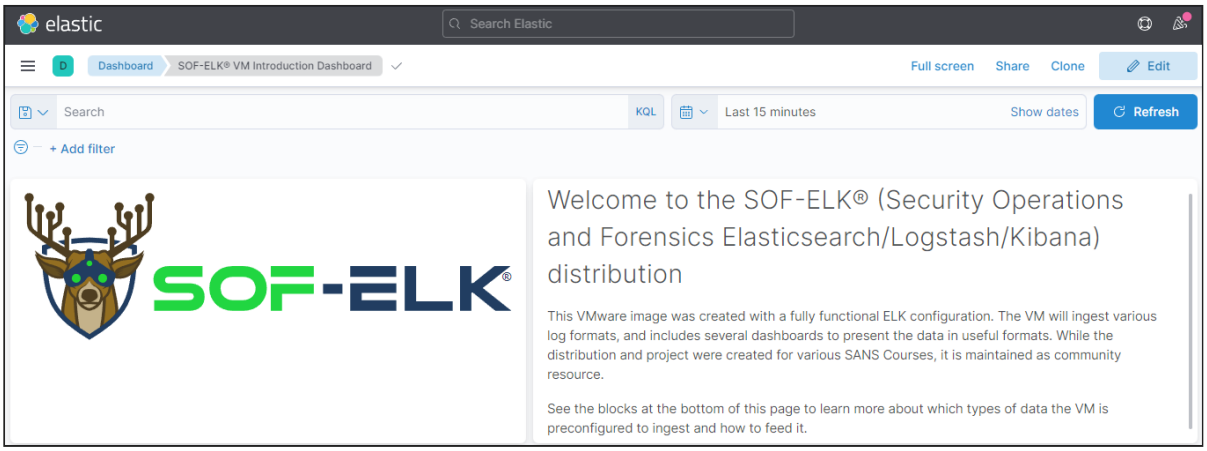
Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- office365 (5,462 documents)
```

5. Once completed, the data will be available in the `office365-*` index.

Lab Content

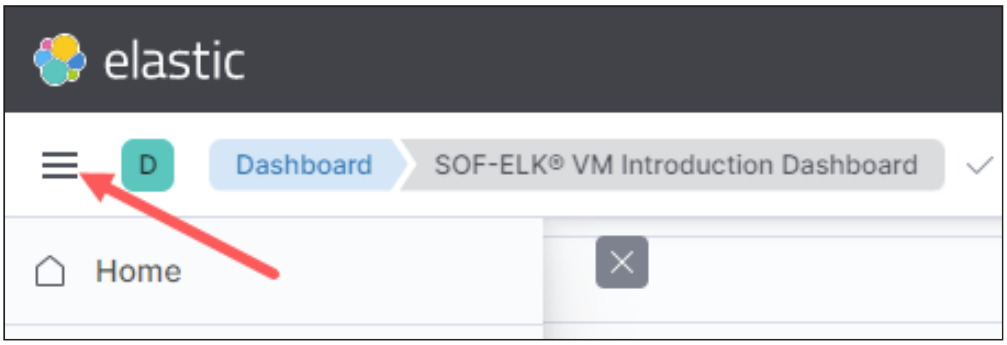
Kibana will initially show you the SOF-ELK VM Introduction Dashboard.



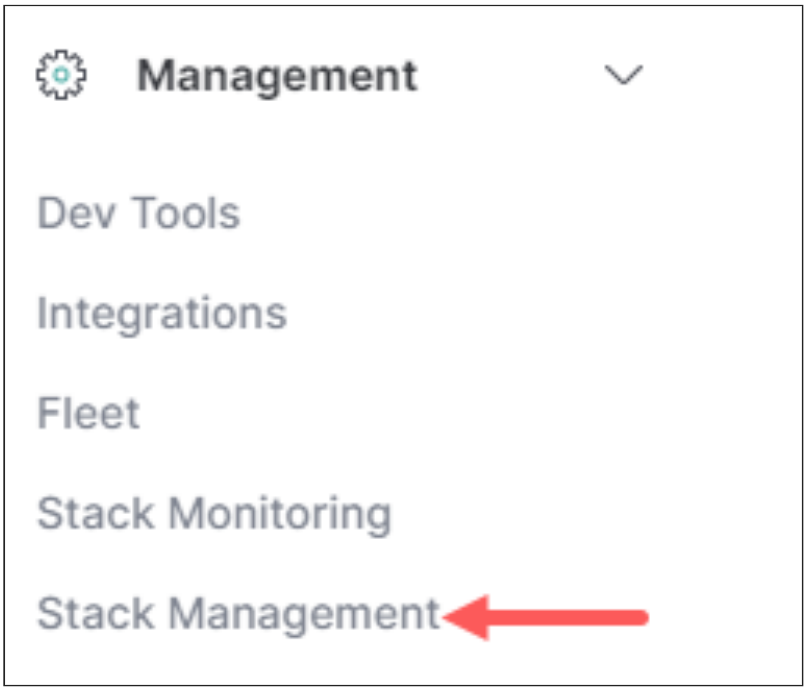
Change Default Number of Rows

Before we start using SOF-ELK, we need to make a configuration change to pre-load a larger number of events (than the default 500) and avoid issues in future labs.

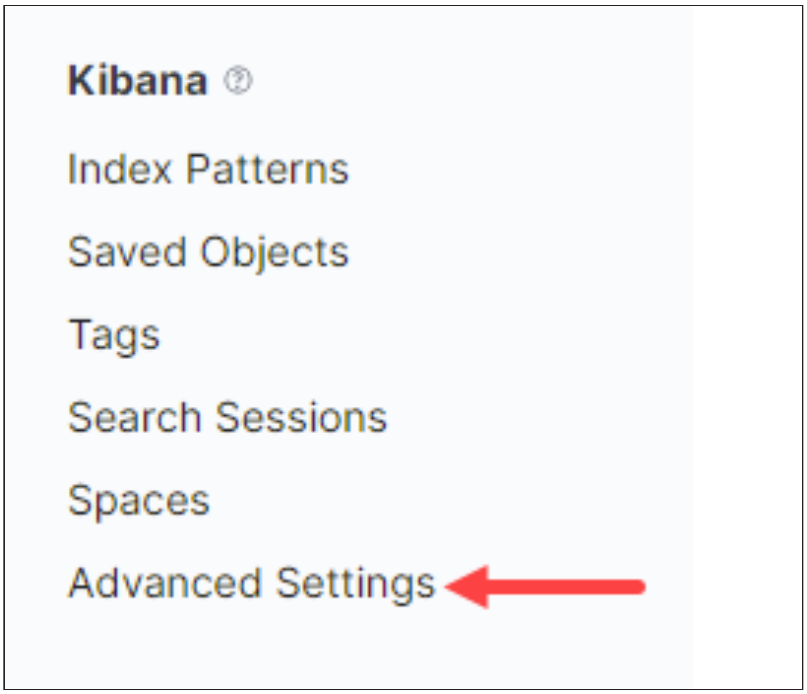
Open the menu by selecting the symbol with the three horizontal lines



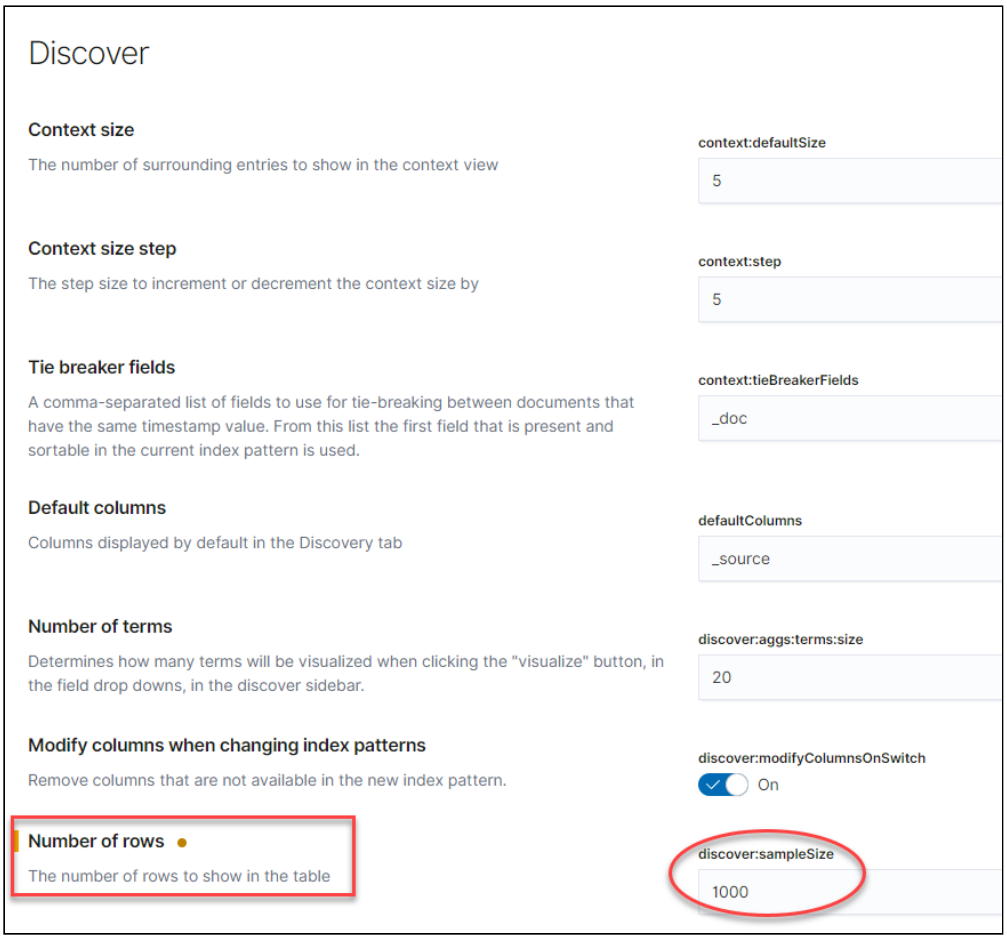
Under the "Management" menu, select "Stack Management"



Select the "Advanced Settings" option



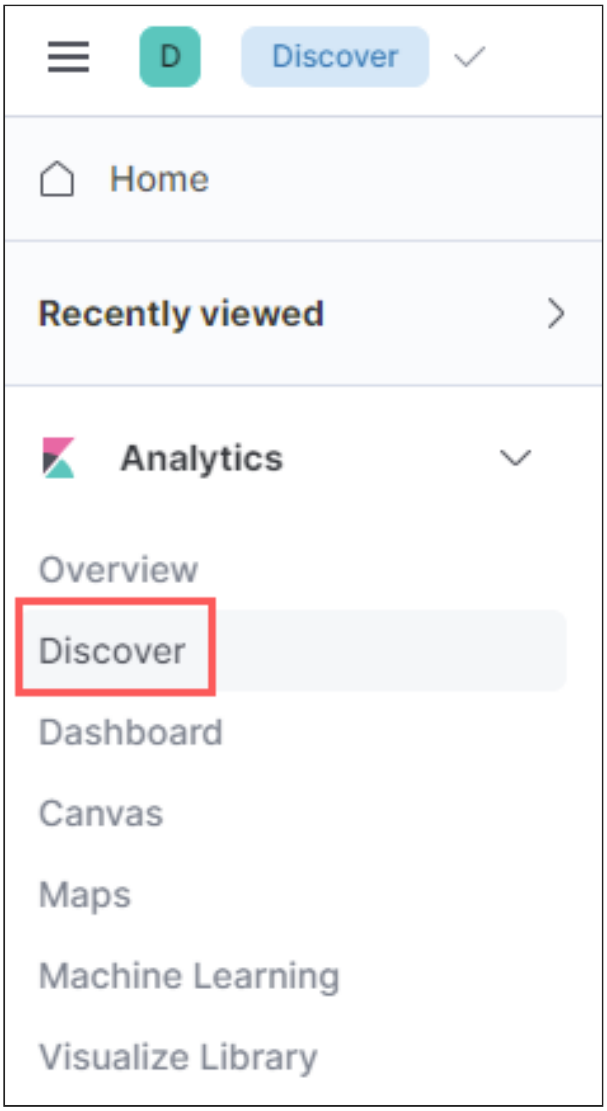
Scroll a good way down to the "Discover" section. Change "Number of rows" to 1000



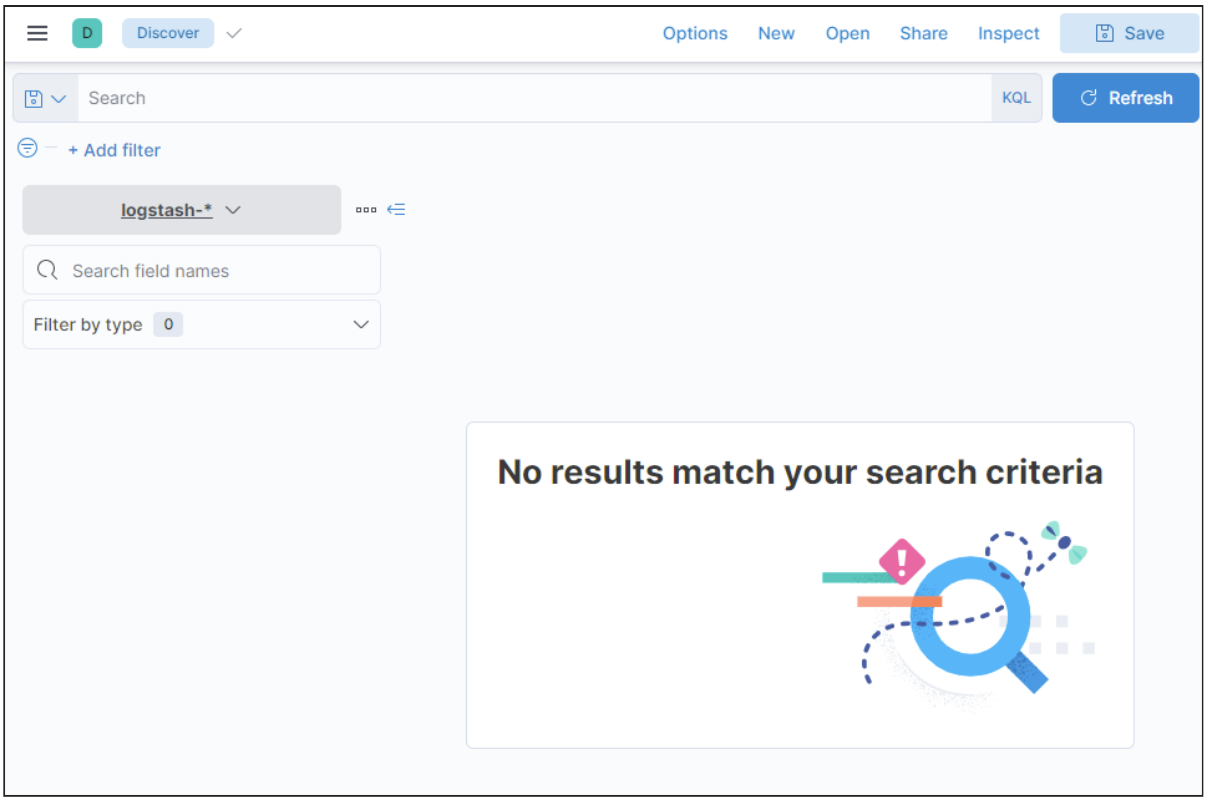
Click "Save Changes"

Explore SOF-ELK

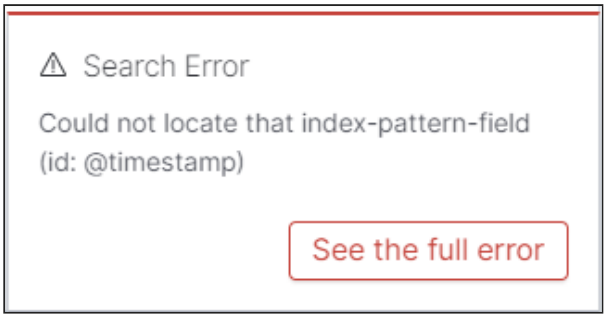
We will start with the `Discover` tab (top left).



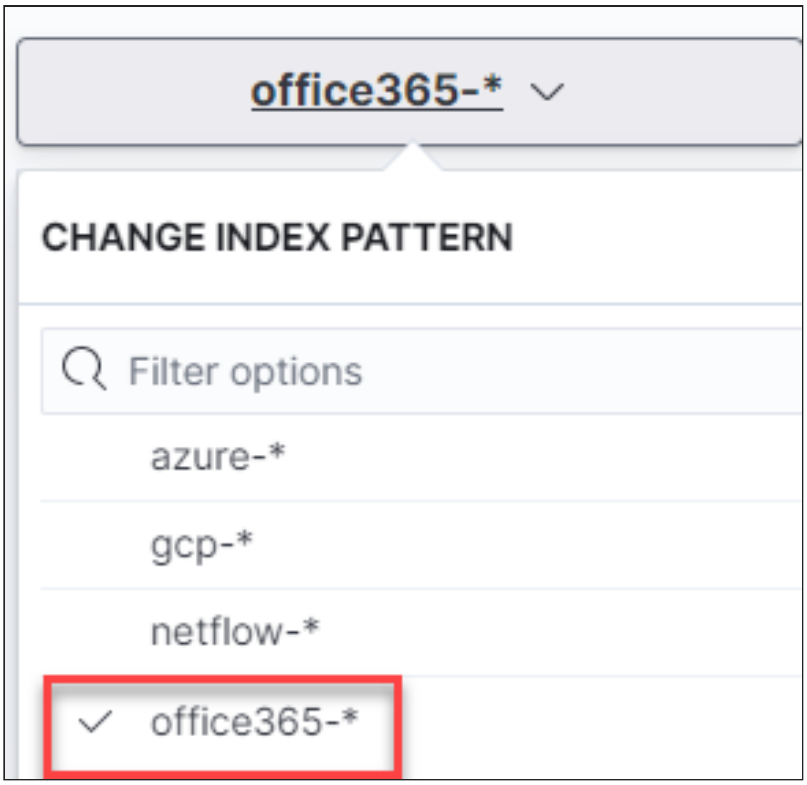
By default, Kibana starts with the `logstash-*` index, but since we don't have any data loaded in that index for FOR509, you are going to get an error message.



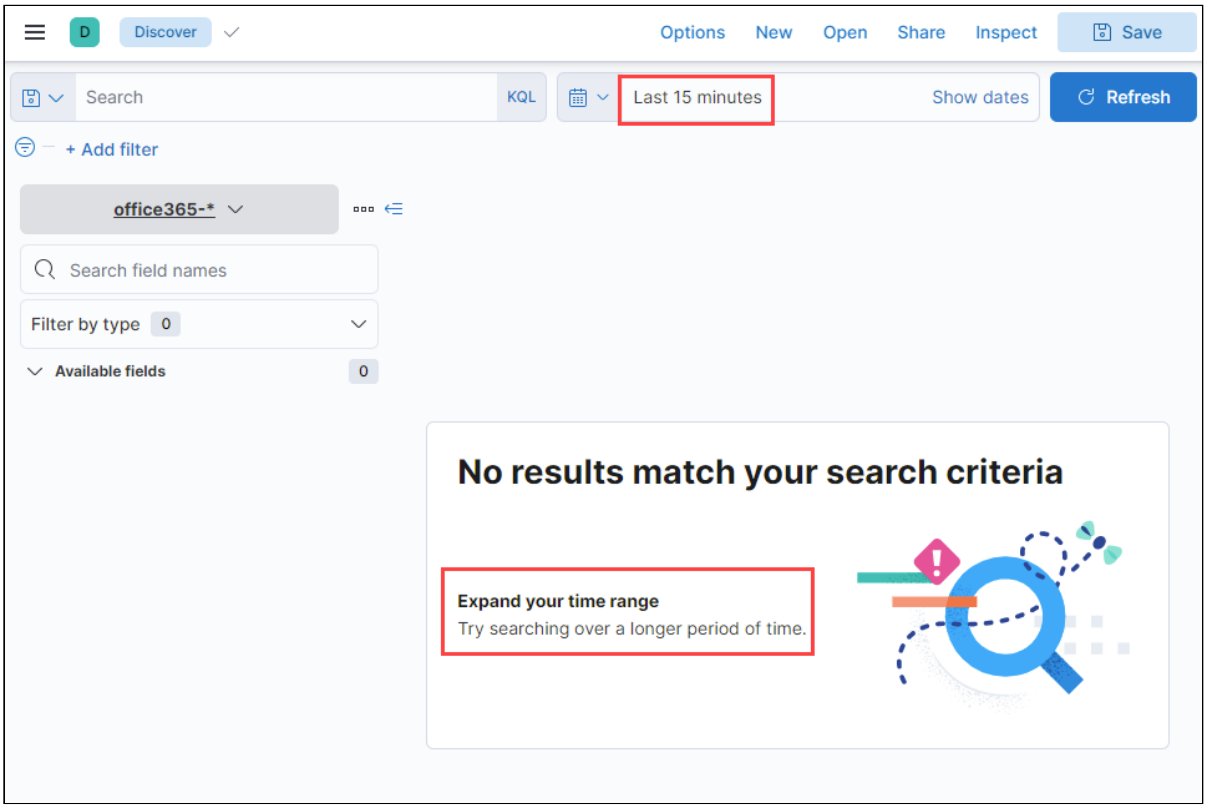
You will also get a popup on the lower right of your screen, which you can close.



Our data is stored in different indices. Select the `office365-*` index:

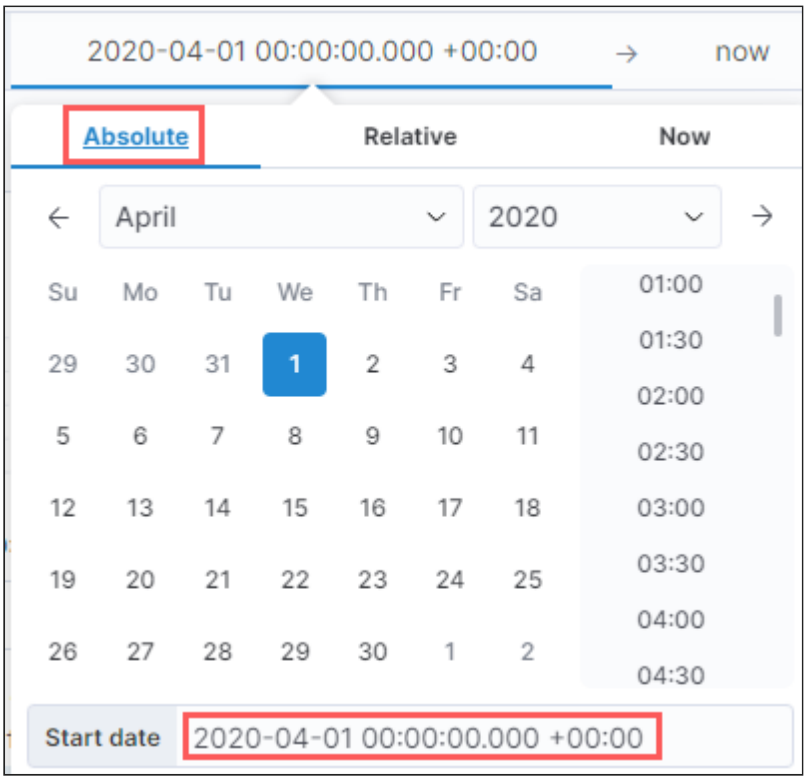


You will get another error since you don't have any data from the "Last 15 minutes".



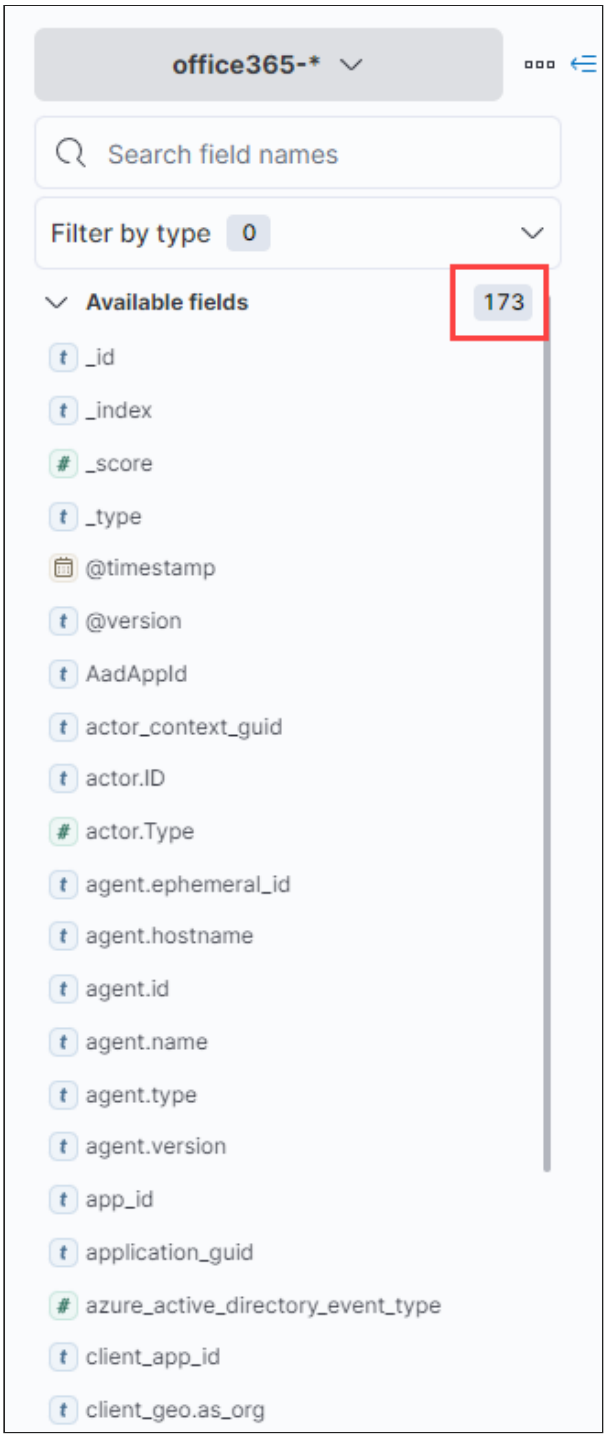
So, first thing is to set your time frame. For the purposes of this tutorial, set your time frame to `2020-04-01 00:00:00.000 +00:00` to `now`.

To set your time frame, you will click on the date and select `Absolute`. This will bring up a calendar. A great shortcut is to copy and paste the date from your workbook in the box at the bottom (highlighted in red).



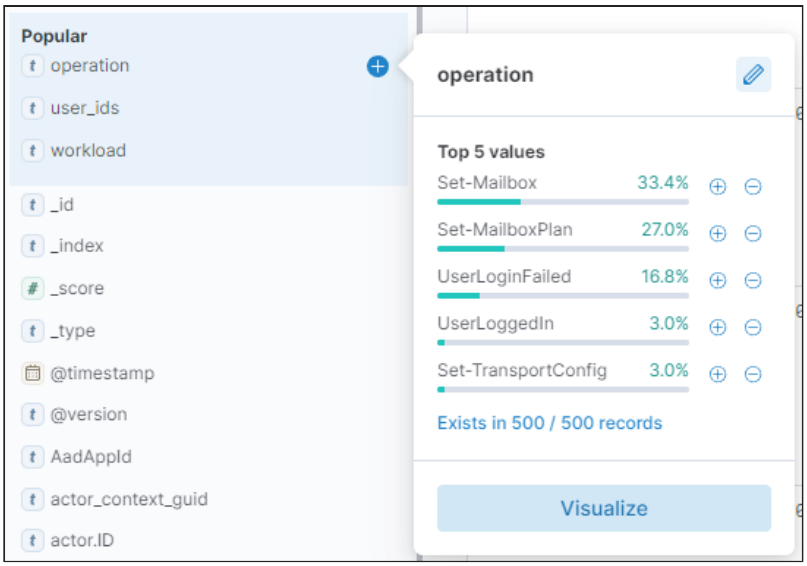
Don't forget to hit `Update`.

On the left, you will have a list of all the fields present in that index (173 in our example, you may see a different number as the logstash script gets updated from time to time). Fields are sorted alphabetically with recently used fields at the top. This is important to remember, as you may be looking for a specific field down the list only for it to be at the top because you recently used it.



The number of available fields will change based on the active filters and searches. As you load more data or change indices, this number will dynamically change to represent the number of fields available in your visible dataset.

If you click on a field (not on the plus sign), it will give you the top 5 values. Note that this quick analysis is limited to 1000 records (per the change we made earlier in the Kibana settings), even if you have thousands in your index. Also, the results will be different based on the time frame you have selected as well as any filters or searches that are currently active.

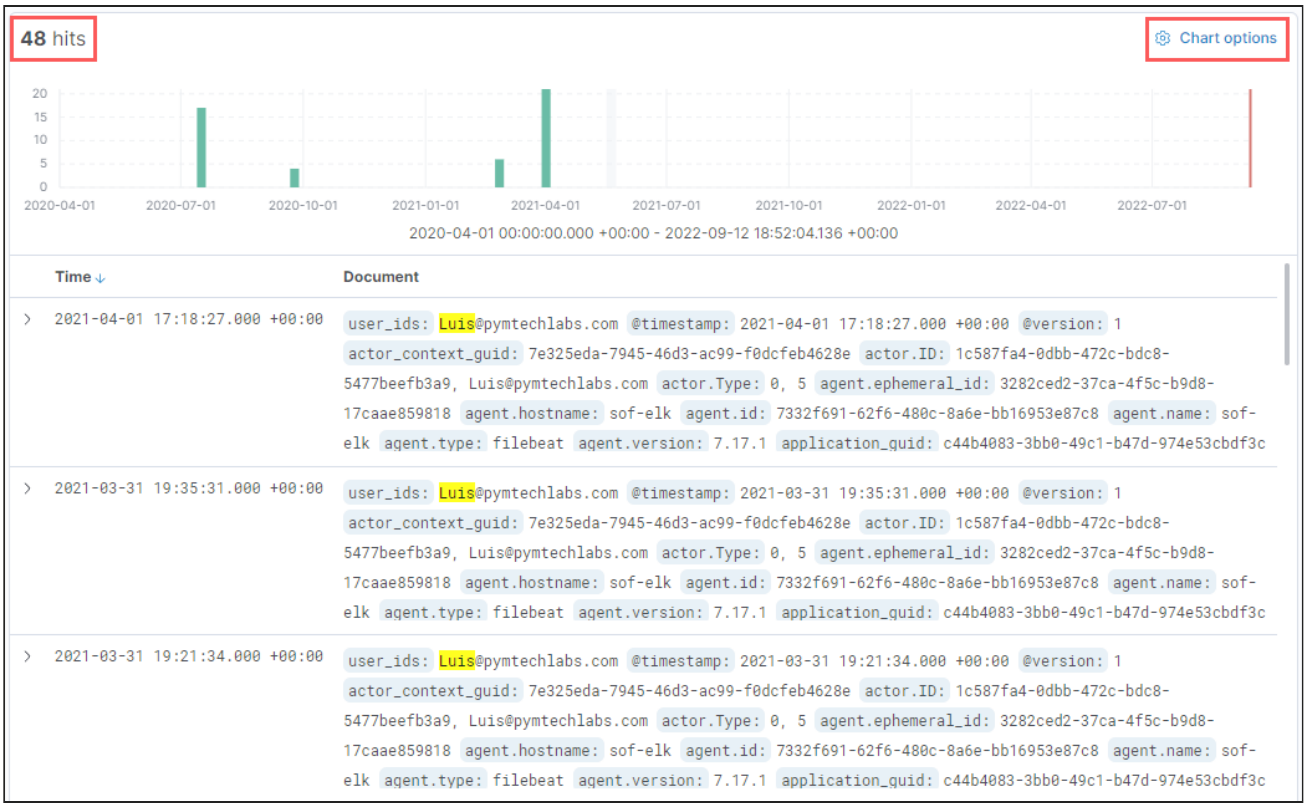


1. Filters and searches in the Discover analytics tab

   The first step is to reduce our data set based on preliminary information we may have received. For the purposes of this example, let's say that we received information that user account `Luis` may be subject to a brute force attack. We will therefore search for failed logins. This is a simple enough search and we could do everything on the search bar, but let's use a filter to demonstrate how useful filters are for more complex queries.

   Select `+ Add filter` for the Edit Filter menu to pop up. Under "Field", use the dropdown menu to select `user_ids`. For the operator, we will use `is`. We just need to enter the value of `luis`. This will work because `user_ids` is an analyzed field. If you accidentally select `user_ids.keyword`, the value would need to be `Luis@pymtechlabs.com`. Kibana will try to help you by showing you that choice as you type.

   The main screen now shows us a graph as well as the list of raw records. On the top left, we see the number of records that remain after our filter is applied. On the top right, you can hide the graph if you want to save some real estate on your screen.



You can expand any of the records by selecting `>` next to that record.

This long list of records isn't the ideal way to view the data. Instead, we can select a few fields and create a table. Select `user_ids`, `operation` and `workload` by selecting the `+` next to each of them.

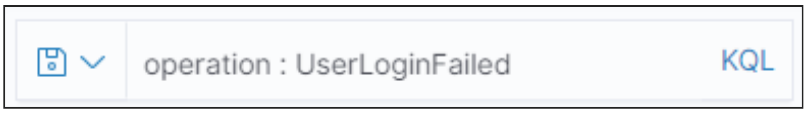
You now have a much simpler view of the records and can still select `>` if you want to view the details of any record.

48 hits

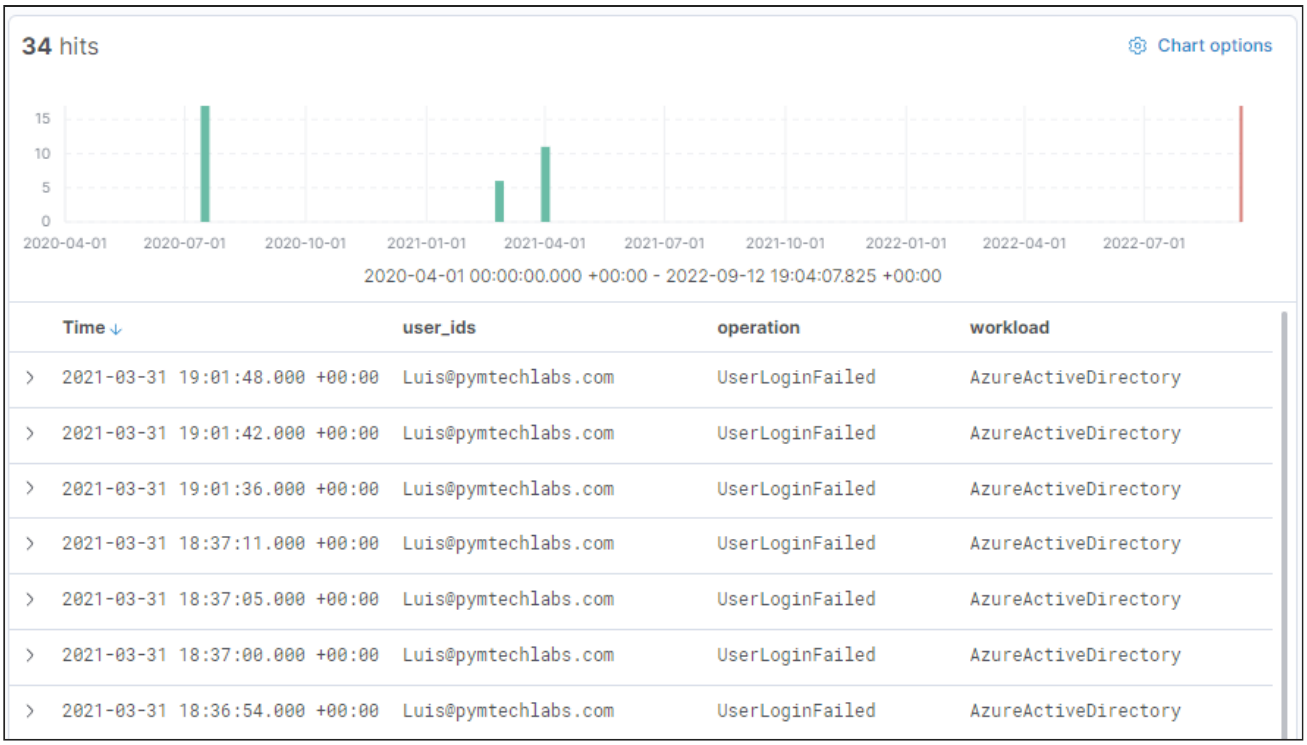
	Time	user_ids	operation	workload
>	2021-04-01 17:18:27.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:35:31.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:21:34.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:13:36.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:12:49.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:12:44.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory
>	2021-03-31 19:11:08.000 +00:00	Luis@pymtechlabs.com	UserLoggedIn	AzureActiveDirectory

We now want to limit our search to failed logins. In the search bar enter

```
operation : UserLoginFailed
```

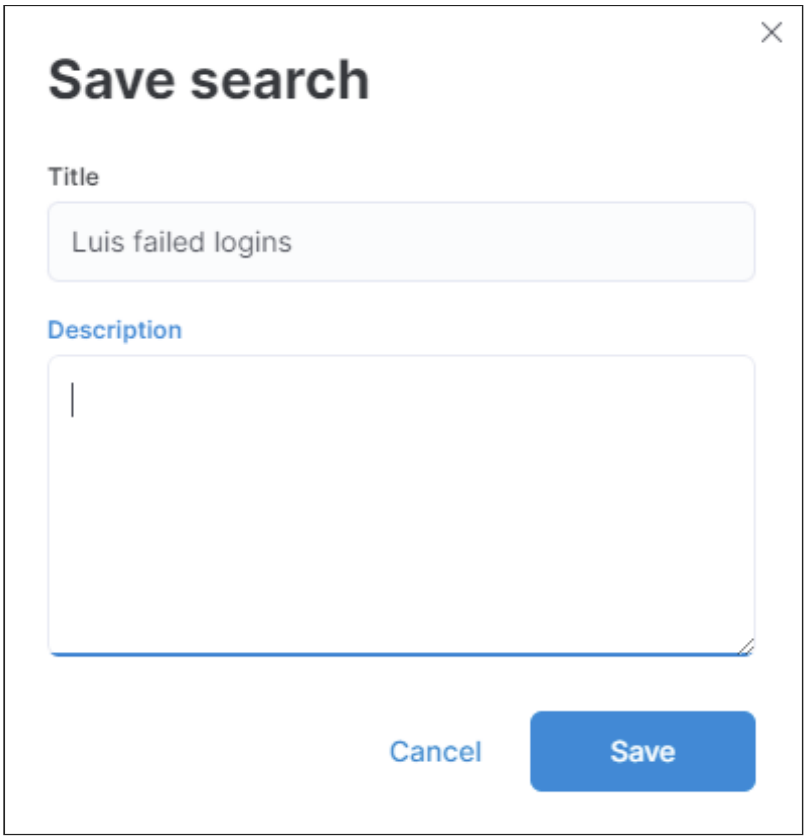


and don't forget to click `Update`.



We now see that Luis had 34 failed logins over the time frame and that these failed logins are distributed over 3 separate days. It's an interesting data point, but before we draw any conclusions, let's use the visualization feature and compare successful versus failed logins for all our users.

Before we move on, let's save this filter and search. Call it `Luis failed logins`.

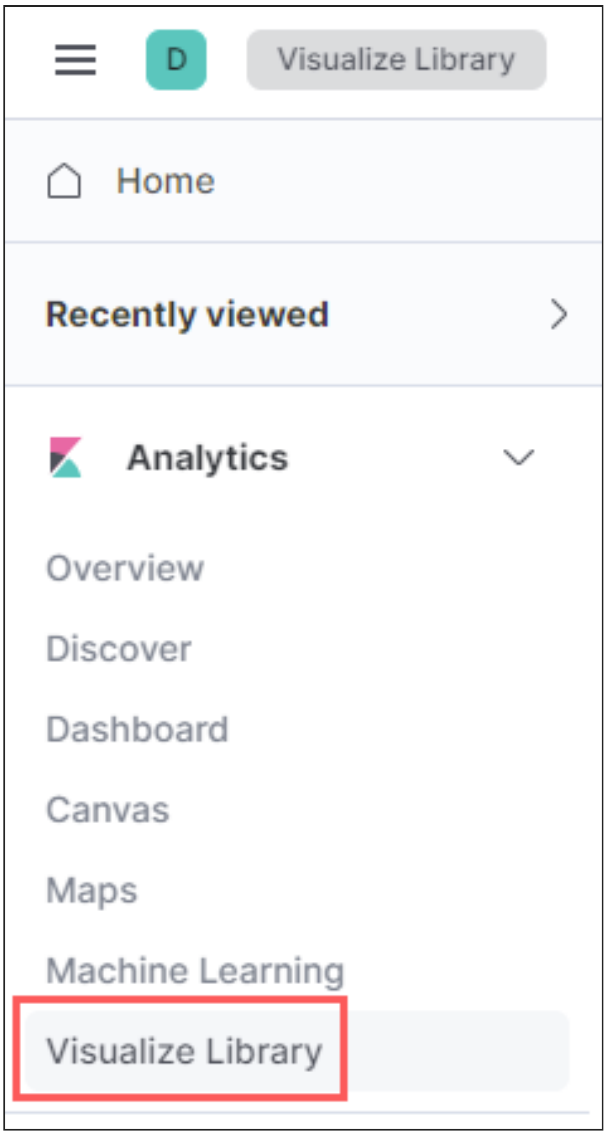


Note

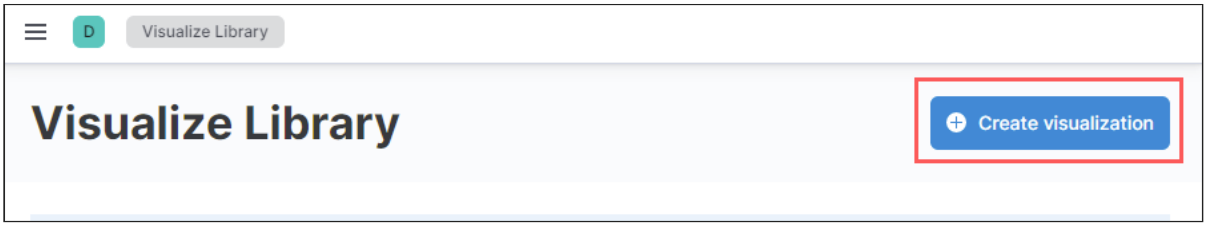
Before you move to the next section, clear your filters and don't forget to hit `Update`.

2. Creating graphs with the Visualization tools

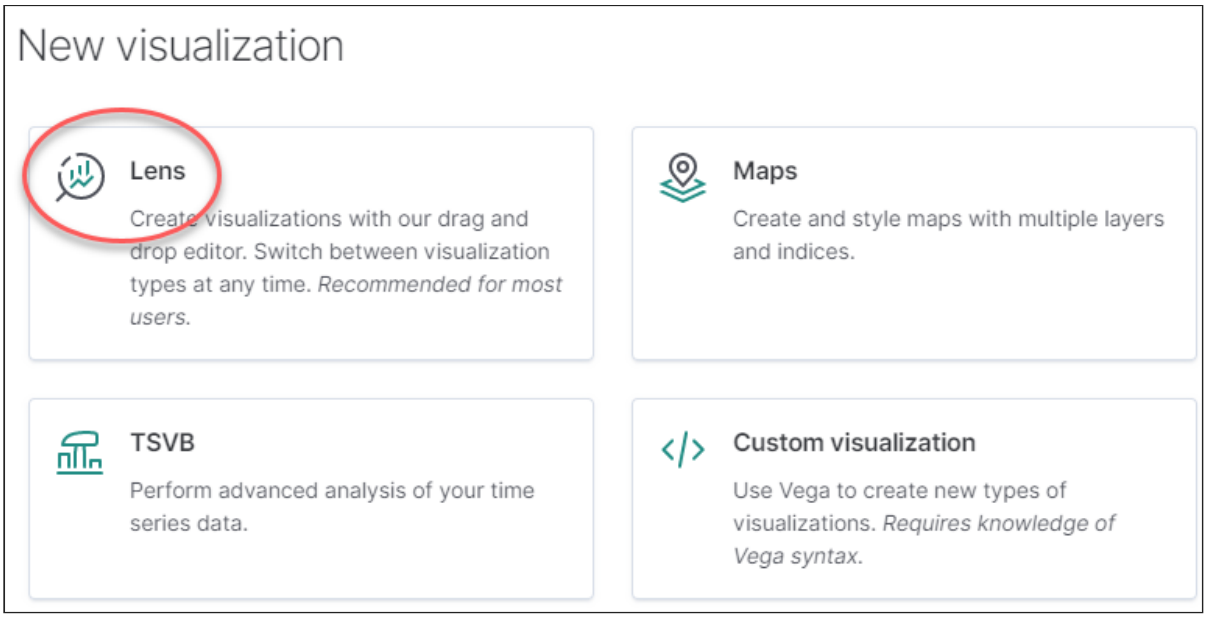
   Switch to the Visualize Library tab.



You will see a list of previously saved visualizations, but to create a new visualization, select `Create visualization`.

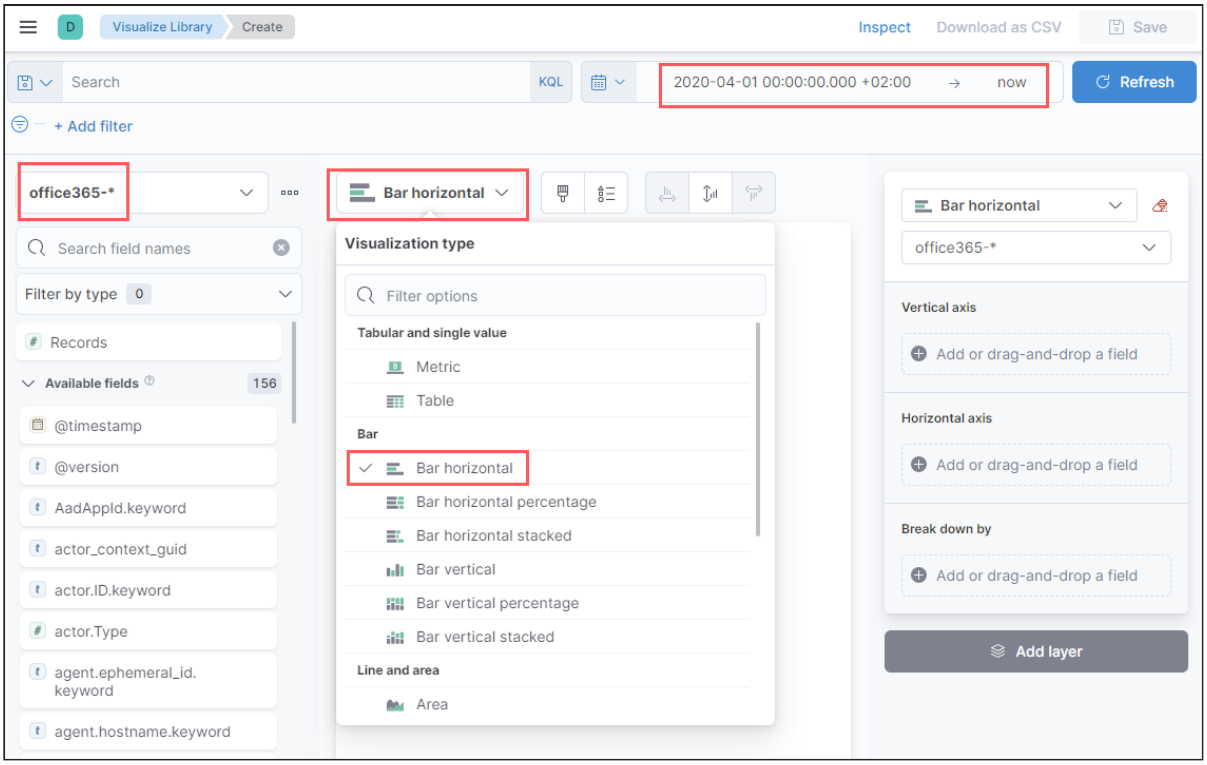


Select `Lens` which is the easiest way to create a visualization because it provides drag-and-drop functionality.

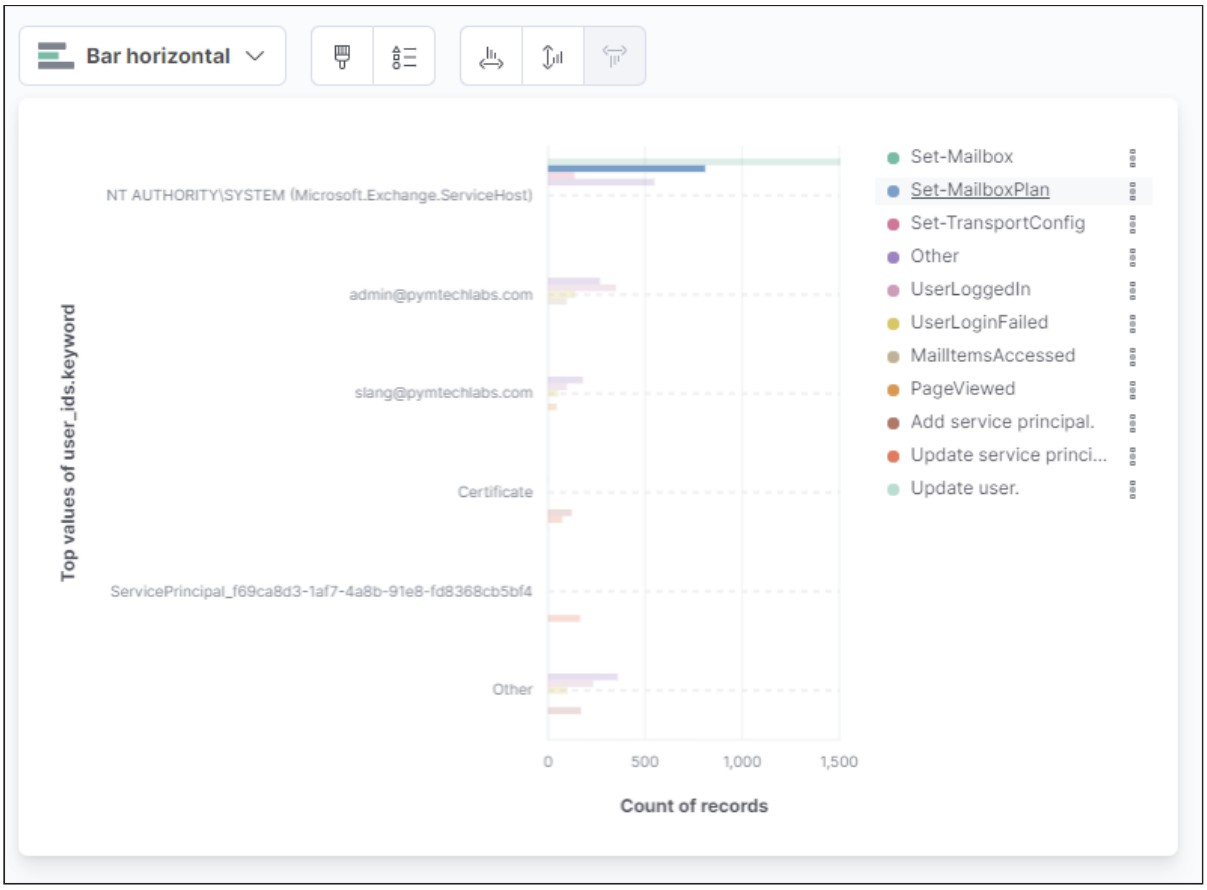


Make sure your index is still set to `office365-*` and the time frame `2020-04-01 00:00:00.000 +00:00` to `now`.

You can select any chart type you would like. In this case, we will select `Bar Horizontal`.



First drag and drop the field `user_ids.keyword`. **Second** drag and drop the field `operation`. The fields are dragged from the left pane.

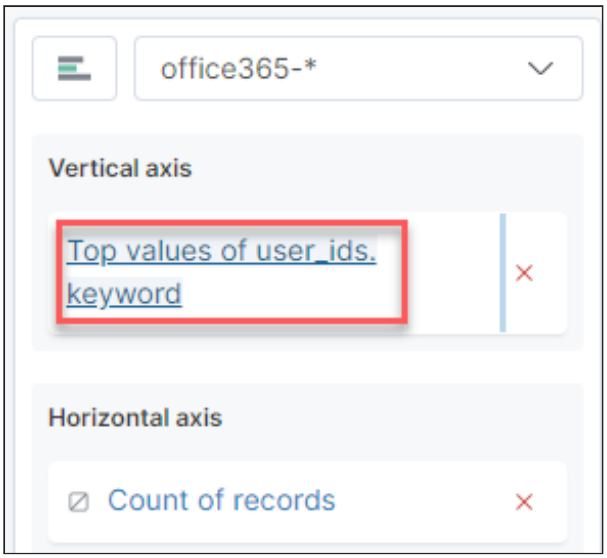


We now have two issues to deal with:

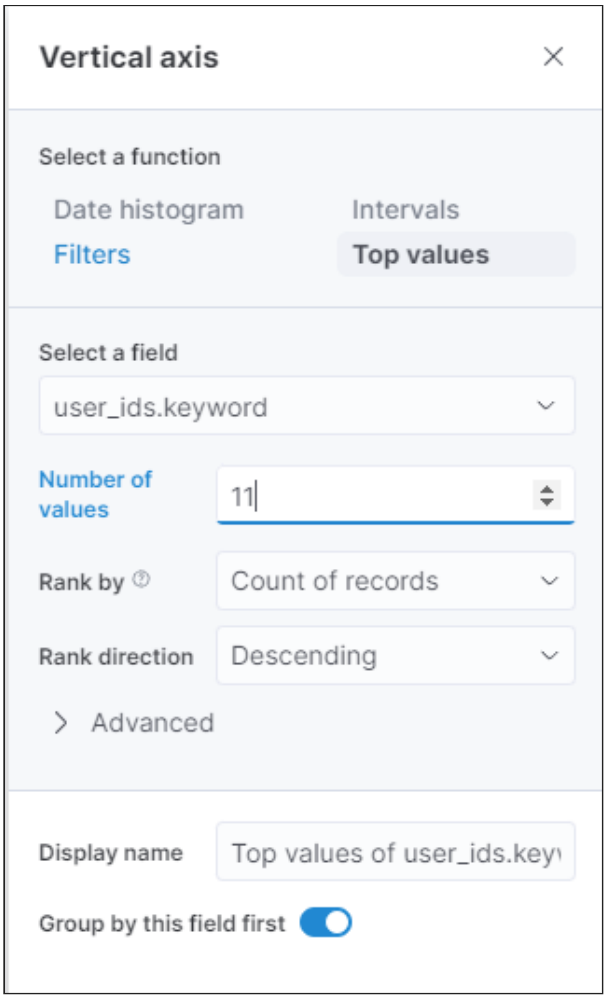
- We have too many `operation` that have nothing to do with logins
- We have system-type IDs

Create a filter to limit `operation` to `UserLoggedIn` and `UserLoginFailed`. The operator `is one of` allows you to create a list of items, which is very useful in this situation.

We now have a chart that only shows us successful and failed logins. Before we write a search to remove the system-type user IDs, we need to expand the `Other` category to make sure we see all records. On the top right, select `Top values of user_ids.keyword`.



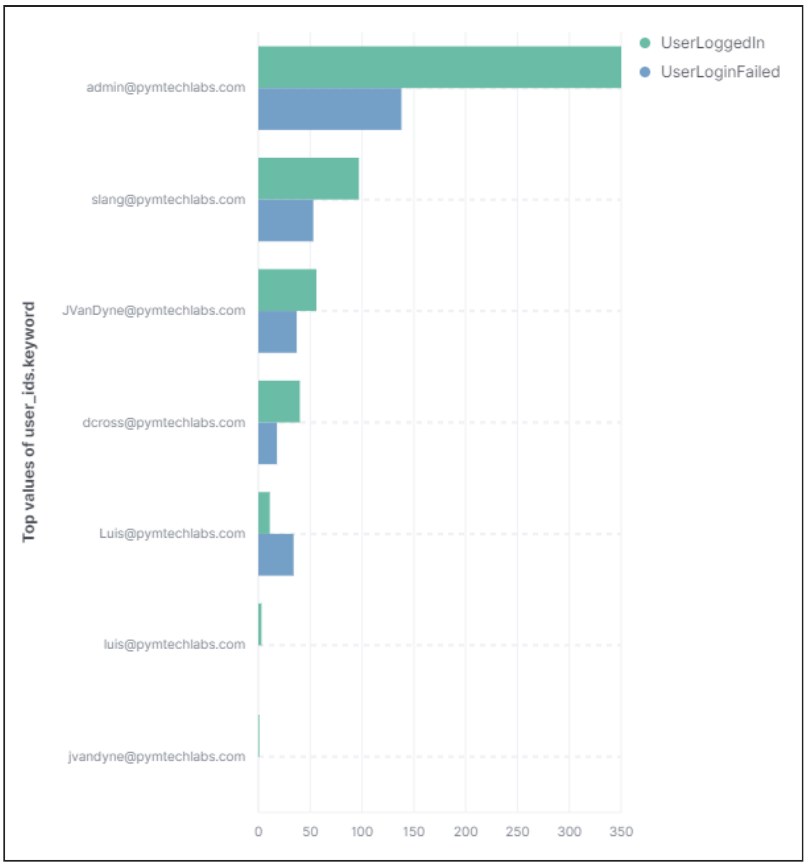
Increase the `Number of values` until the `Other` category disappears. In this case, 11 will do it.



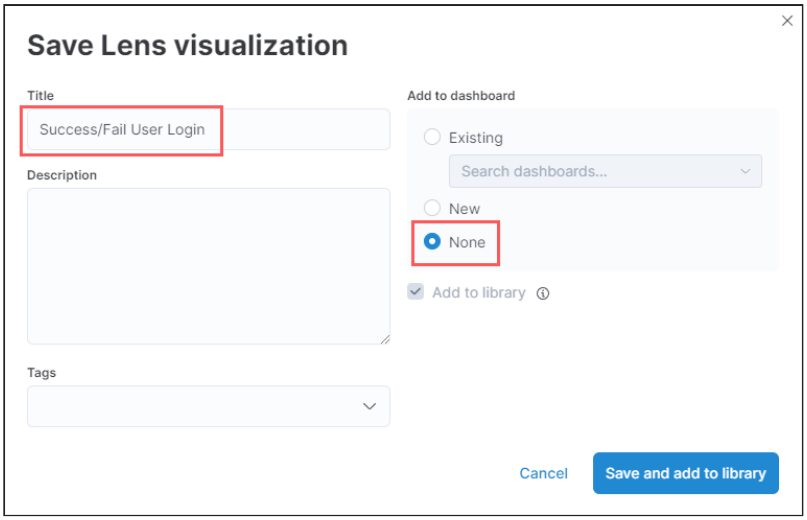
Use the following search:

```
not user_ids : (675* OR 1fb* OR Unknown OR "Not Available")
```

We now have a nice chart of every user's successful and failed logins. Looks like we should be more concerned about all the failed logins against `admin@pymtechlabs.com`. Some of you may think, why would you use such an obvious username? The answer is simple: free data generation! Yes, the account has 2-factor authentication enabled.



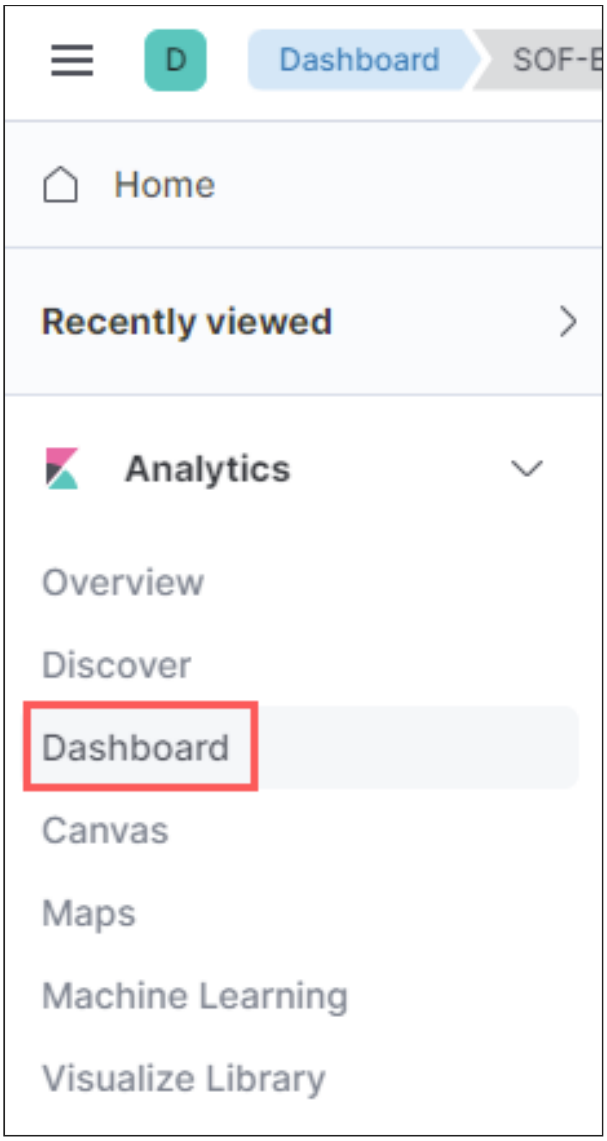
The last step is to save our chart. On the top right, select `Save` and call it `Success/Fail User Logins`.



Be sure to select "None" which will save the visualization and keep you in the visualization tab. If you select "New" you will be taken to the dashboard section, and the steps will be different from the ones below.

3. Create a dashboard

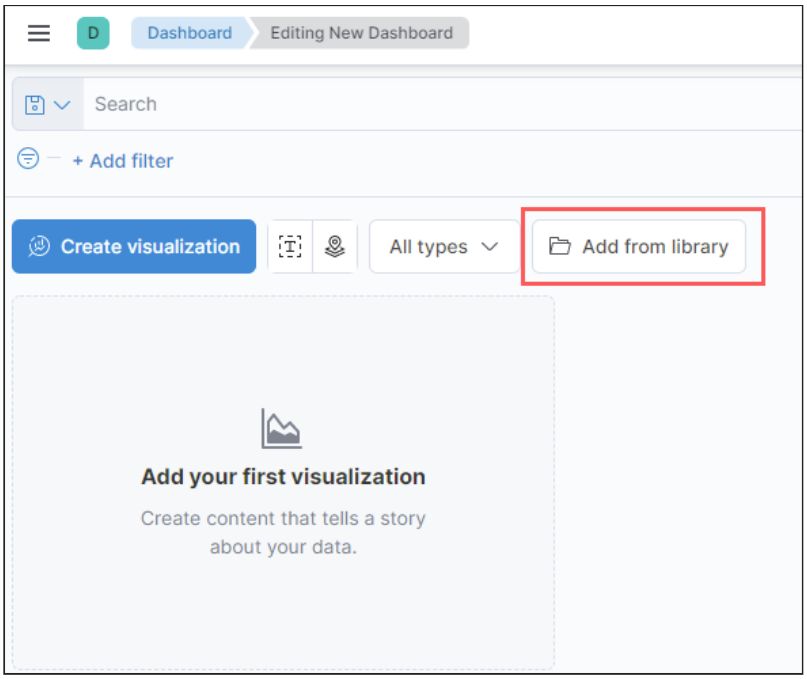
   There's nothing like a nice dashboard to impress management. So let's create a dashboard. If you selected "None" in the prior step, you will need to select the `Dashboard` tab, otherwise Kibana will already be in that section.



This will take you back to the introduction dashboard. Click on the word `Dashboard` to open the dashboard menu.

Select `Create dashboard`.

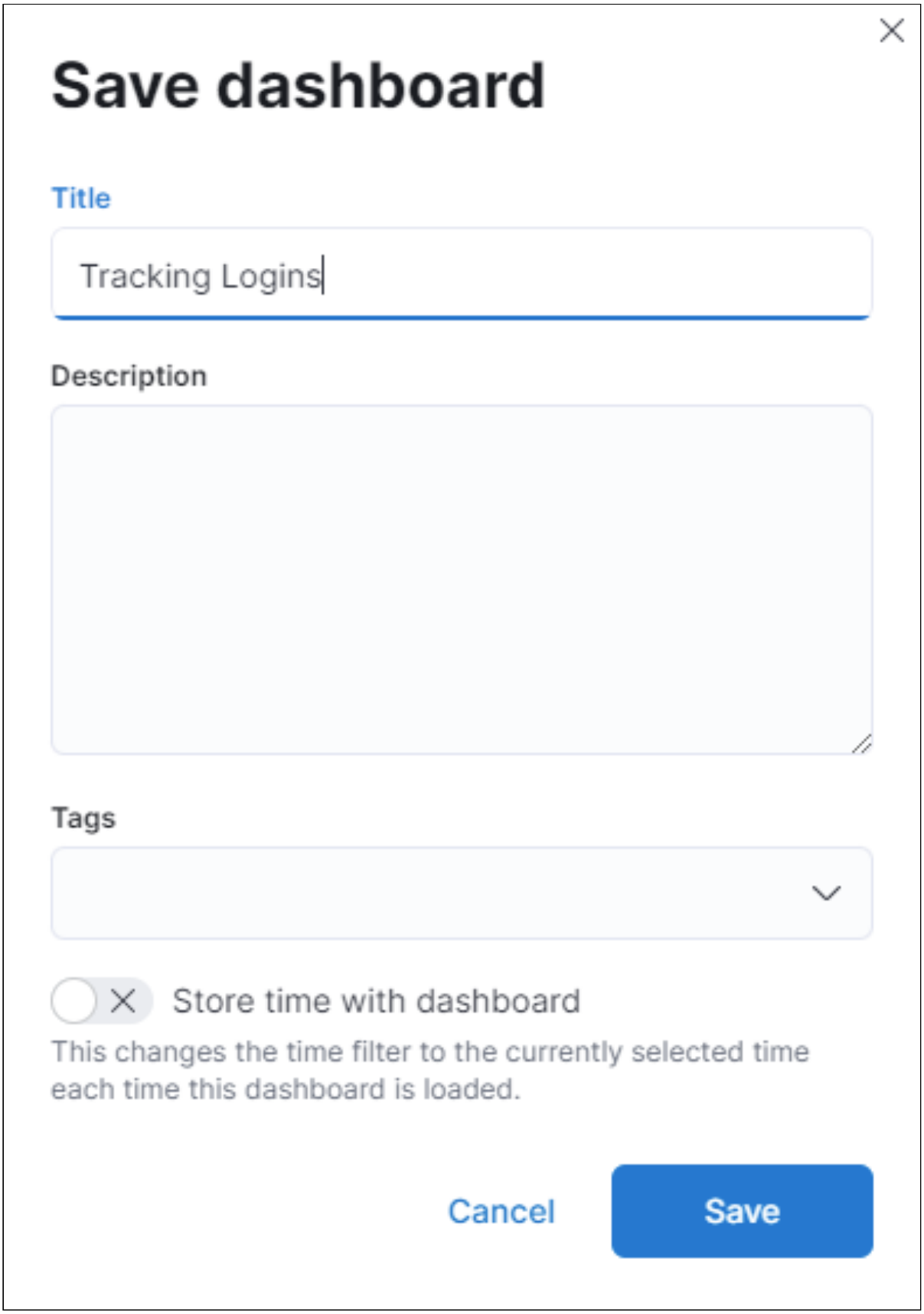
Since we previously saved our search and our chart, we just need to add them from the library.



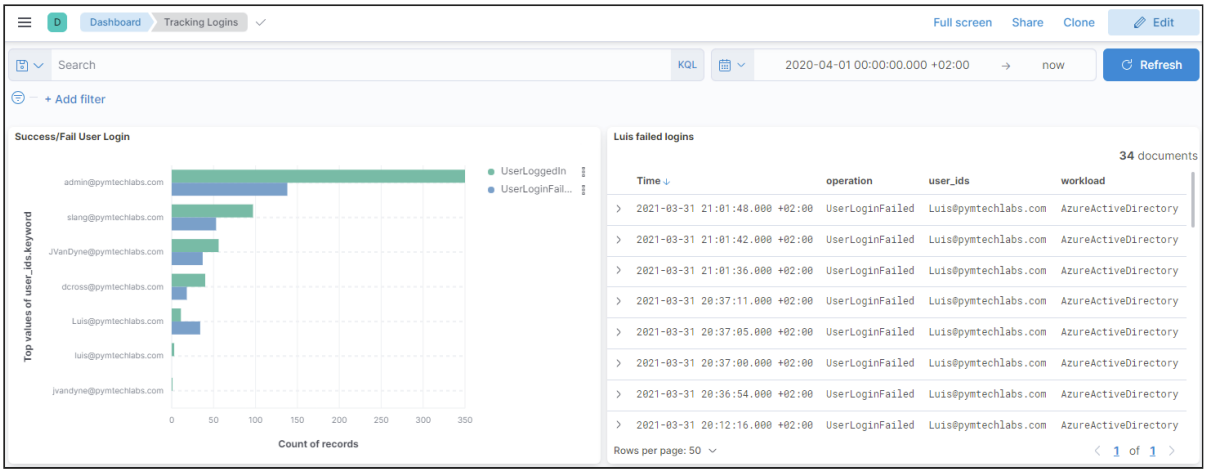
Repeat the procedure twice to add:

- `Success/Fail User Logins` char
- `Luis failed logins` search

You can now save your dashboard.



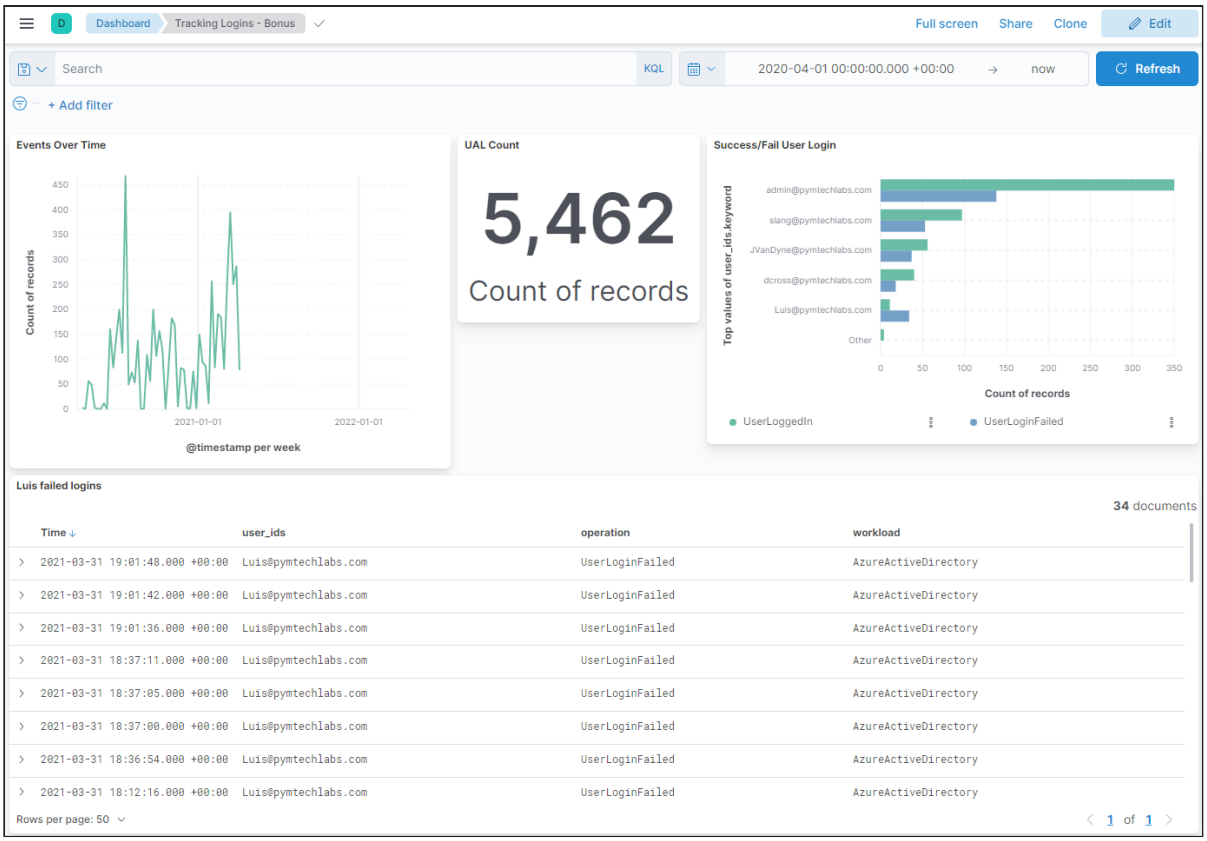
You now have a complete dashboard.



As you can see, dashboards are very easy to create. The possibilities are only limited by your creativity.

Bonus Section

If you are experienced and would like an additional challenge, try to recreate the following dashboard:



You will need to create two new visualizations:

- A line visualization with `@timestamp` as the plotted field on the horizontal axis.

- A metric visualization that counts the number of records.

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Kibana is a powerful platform, yet easy to use.
- Now that you have a basic understanding of SOF-ELK, you are ready to explore the FOR509 labs!

Lab 1.2: Exploring the UAL

Objectives

- Review logs from Microsoft 365 to understand user activity
- Search for possible unauthorized access

Background

At the end of June 2020, Janet Van Dyne was attending the International Biochemical Engineering conference in Iceland. While there, she finally heard back from the payroll department regarding the problems she was having with her paycheck. However, something tingled her bio-synthetic wings regarding the email. She sent Darren Cross an email asking for his advice, but with the time difference, it was going to be a while before she received an answer. Wanting the issue resolved as soon as possible, she clicked on the link in the email.

The link took her to the usual Microsoft 365 login page where she entered her credentials, but not to the payroll site as she expected. She informed Frank Pym of the suspicious email, and it's now your task to figure out what happened.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2020-06-15 00:00 UTC** to **2020-07-15 23:30 UTC**.

Load Data

The Microsoft 365 unified access log was ingested into SOF-ELK in Lab 1.1.

Lab Content

Time Frame

We weren't given the exact time frame for the conference; we were only told "at the end of June 2020." Let's set our time frame to: `2020-06-15 00:00:00.000 +00:00` to `2020-07-15 23:30:00.000 +00:00`

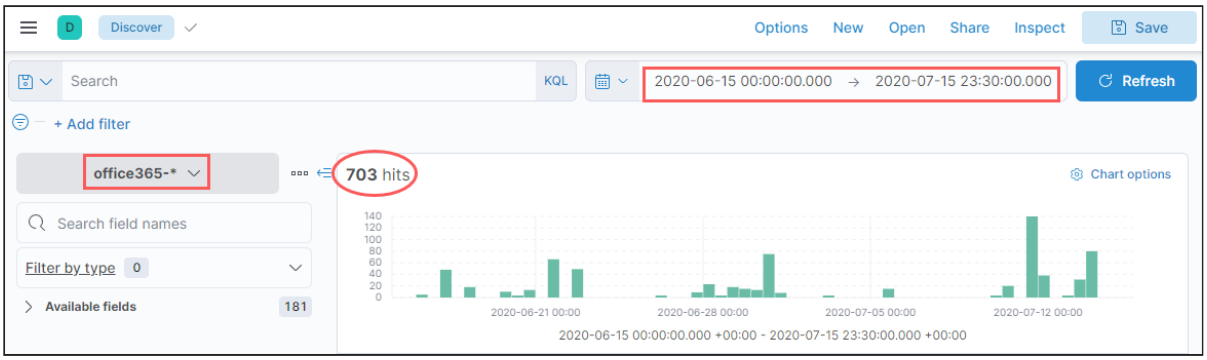
Search for Unauthorized Access

Start in the Discover tab

Set your index to `office365-*`

Be sure to clear any prior filters.

You should see `703 hits`. If you don't, make sure you have selected the correct index.



1. Filter for Janet Van Dyne

Create a filter to only show events related to Janet Van Dyne. How many records remain?

Hint

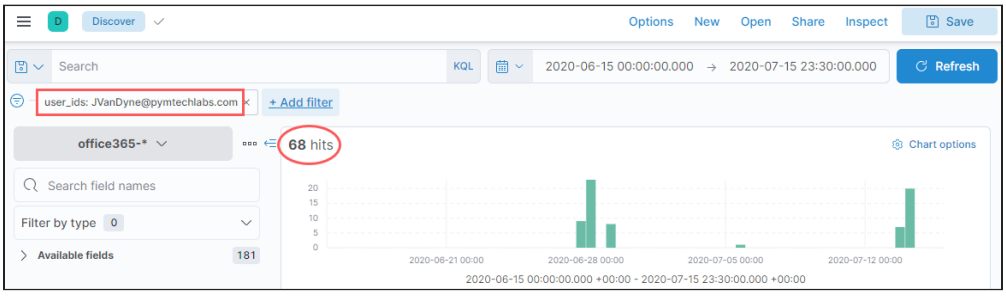
Filter on the field `user_ids` using Janet's email `JVanDyne@pymtechlabs.com`

Answer

```
user_ids: "JVanDyne@pymtechlabs.com"
```

You could write the filter on the search bar, but using the "Add filter" option keeps your working area cleaner. Either way works just fine.

There are 68 records.



Update your view to show where Janet connected from

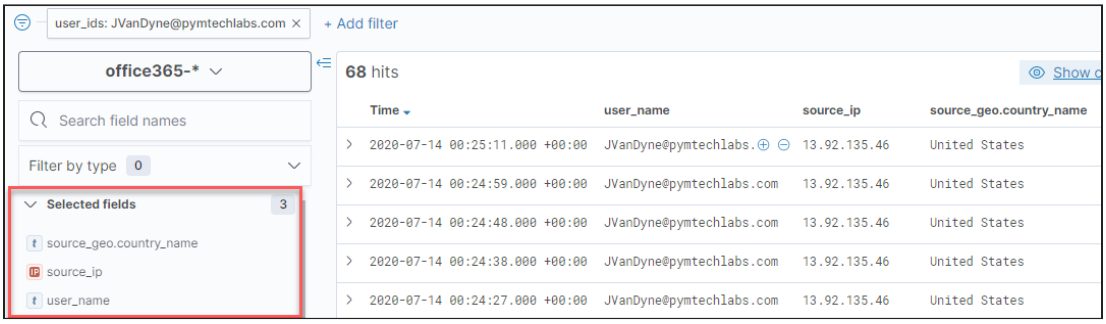
Hint

Select the fields `source_ip` and `source_geo.country_name`

Answer

Thanks to the GeoIP data, we can find out where these IPs are from.

We recommend you also add the field `user_name` to verify you have the correct records.



2. Look for suspicious locations

When was Janet attending the conference?

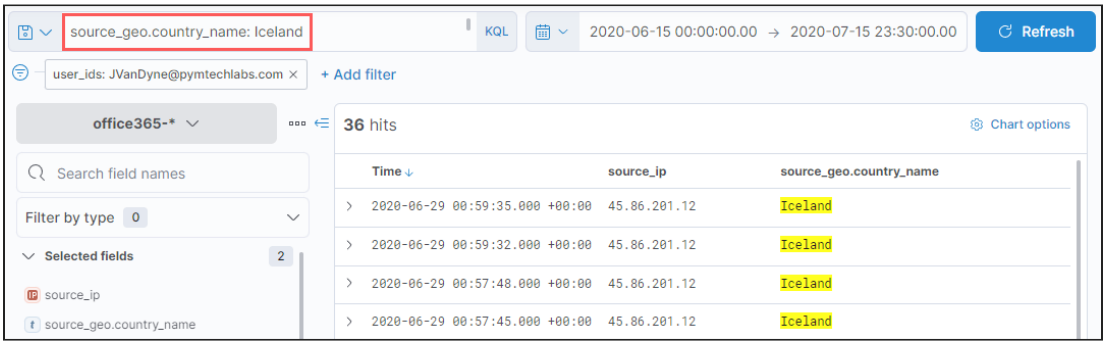
Hint

Look for IPs from Iceland.

Answer

```
source_geo.country_name: Iceland
```

Janet was connected from Iceland starting on 2020-06-27 until 2020-06-29.



Are there entries that look suspicious?

Hint

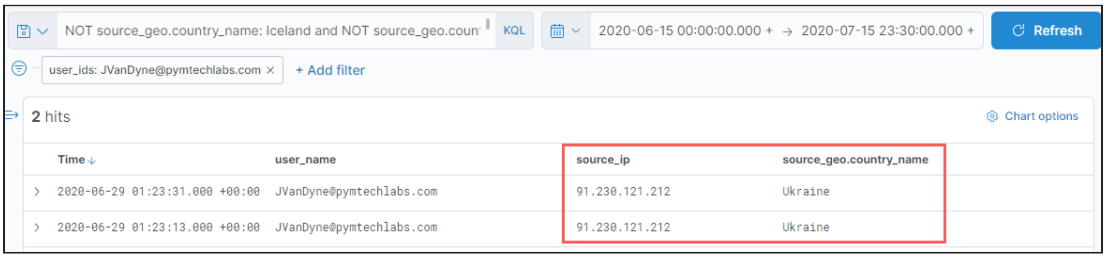
Look for IPs from countries other than Iceland since you know where Janet is currently located.

Answer

```
NOT source_geo.country_name: Iceland and NOT source_geo.country_name: "United States" and
source_geo.country_name: *
```

The last part of the filter `source_geo.country_name: *` is to eliminate entries that don't contain geolocation information. The uppercase for "NOT" isn't required. It's only there to emphasize that we are eliminating that specific condition.

Given that Janet Van Dyne is in Iceland attending a conference, the IP address from the Ukraine is potentially suspicious.



3. Analyze the logs

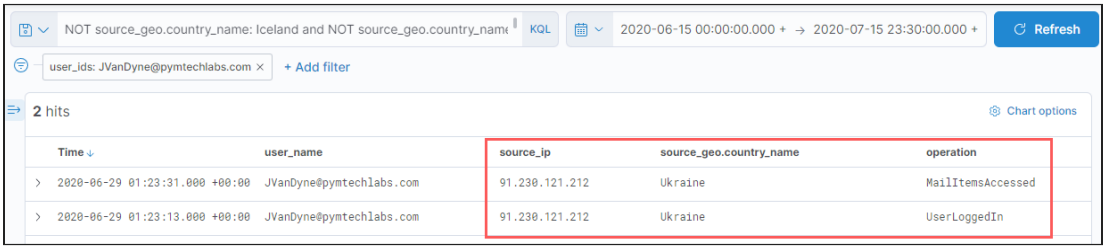
What activities did the suspicious IP address perform?

Hint

Add the `operation` field

Answer

The suspicious operations are `MailItemsAccessed` and `UserLoggedIn`



Let's examine the significance of these two operations in the next few questions.

For this next section, remove any active filter and selected fields.

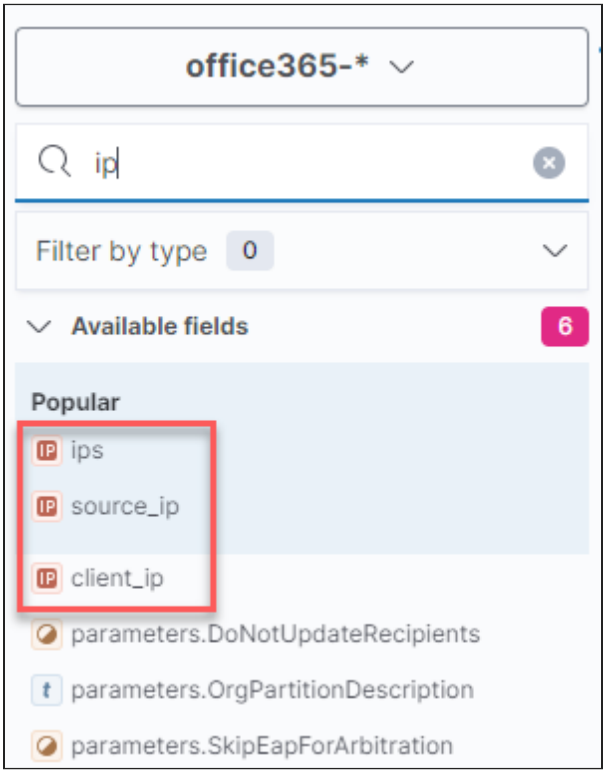
Which fields contain IP addresses?

Hint

In the "Search field names" box, type `ip`

Answer

- source_ip
- client_ip
- ips



The field `ips` is a list of all IPs found in the record. This is an enrichment created by Logstash. This is very useful since we don't always know which IP field a log will choose. Also, remember that the authors had to make some mapping choices in the logstash script. In reality, the UAL fields are called `ClientIp` and `ActorIPAddress`. This mapping is necessary due to the inconsistent naming across different logs.

How many events have the bad IP address?

Hint

Add the filter `ips: 91.230.121.212`

Answer

```
ips: 91.230.121.212
```

There are 3 events with the attacker's IP address

Note

You may want to extend the time from the known date of the bad activity until `now`. This provides the widest possible view in case the attacker performed multiple actions over multiple days. Remember that our current view is limited to the `office365-*` index. In a real situation, you would want to check this IP address against all your indices.

After the login event, what's the attacker's first action?

Hint

Be sure to consider the timestamp of each event.

Answer

The attacker's first action was to accessed Janet's mailbox

t	mailbox_owner_upn	JVanDyne@pymtechlabs.com
t	operation	MailItemsAccessed

What method did the attacker use to access Janet's mailbox?

Hint

Open the event's detail and look at the `client_info_string` field.

Answer

The attacker used the IMAP4 protocol

t	client_info_string	Client=POP3/IMAP4;Protocol=IMAP4

IMAP is a favorite method for bad actors to test if they have captured valid credentials. `client_info_string` is a good field to monitor for potential problems.

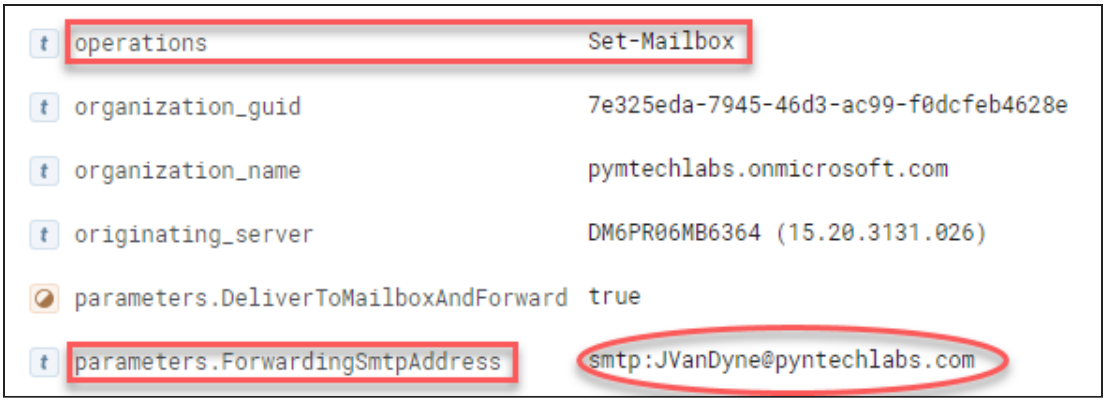
What modification did the attacker make to Janet's mailbox?

Hint

Check the last log entry and look for the fields that start with the word `parameters`.

Answer

The attacker set a forwarding address



Is there something unusual about that email address (no hint)?

The attacker registered a domain that has one letter misspelled compared to the real domain: pyntechlabs instead of pymtechlabs

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Make sure to audit all Set-Mailbox actions and look for the parameters `DeliverToMailboxAndForward` and `ForwardingSmtpAddress`.
- There is a similar parameter called `ForwardingAddress` that has the same effect; however, it requires administrative privileges to set.

Lab 1.3: Privilege Escalation with Graph API

Objectives

- Understand the power (and risk) of Graph API permissions
- Review potential malicious activities
- Review remediation activities within the context of incident response

Background

Hank Pym was alerted that Janet Van Dyne has been given the Global Administrator role. That's not right!

Your mission is to figure out how this happened and assess any potential damage. In addition, you need to audit Hank Pym's remediation steps.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.

Load Data

The Azure Active Directory and Azure Tenant logs have been saved in `/home/elk_user/lab-1.3_source_evidence.zip`

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract `lab-1.3_source_evidence.json` from the ZIP file with the following commands:

Command lines

```
cd ~
unzip -q lab-1.3_source_evidence.zip -d /logstash/azure
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you have 141 documents loaded in the azure index:

Command line

```
sof-elk_clear.py -i list
```

Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- azure (141 documents)
- office365 (5,462 documents)
```

5. Once completed, the data will be available in the `azure-*` index.

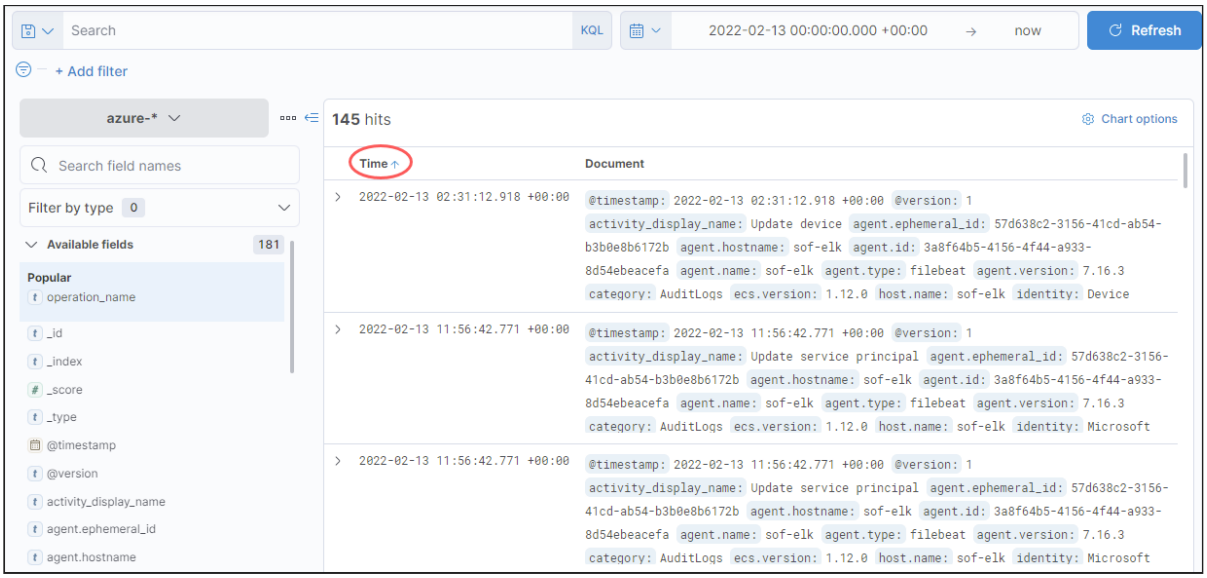
Lab Content

Time Frame

`2022-02-13 00:00:00.000 +00:00` to `now`

Set your index to `azure-*`

Suggest you sort from oldest to newest entry and make sure that you see 141 hits.



Option 1: Advanced version for experienced students

In this version of the lab, you are on your own to figure out the events that led to Janet Van Dyne being given the Global Administrator role.

Just as you would be expected in a "real world" incident, you should provide a report with the following:

1. A timeline of events. Each event should be supported with the appropriate log entry.
2. Once Janet Van Dyne was given the global administrator role, did she take any action?

If you choose the advanced version of this lab, you will need to finish within the same timeframe as the rest of the class.

Hint: Occasionally looking at the raw logs will provide additional information that we weren't able to parse in SOF-ELK. A program like `jq` is very useful for that purpose.

The solution to these questions is in the guided version.

Option 2: Guided version

In this version of the lab, we will guide you through the steps to investigate the events that led to Janet Van Dyne being given the Global Administrator role.

Start in the Discover Tab

Be sure to clear any prior filters.

1. Find the entry where Janet Van Dyne is given the Global Admin role

What's the name of the operation that would perform such an action?

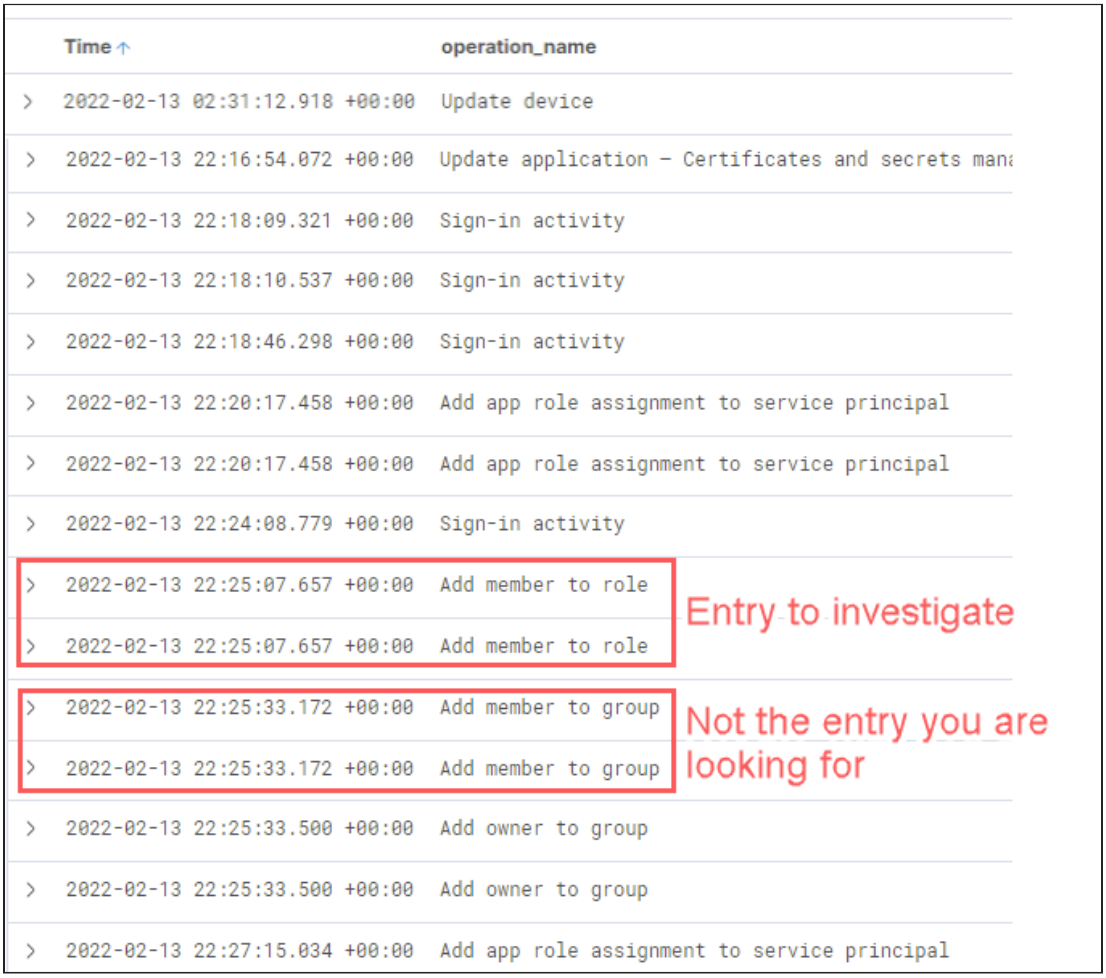
Hint

People frequently say "global admin group." That's incorrect. Global admin is a role, not a group. This is important in selecting the correct operation.

Answer

Add the `operation_name` field to your view and scroll down until you see the operation called `Add member to role`. Alternatively, you can search for that operation:

```
operation_name: "Add member to role"
```



2. Find who granted the role

How many unique `Add member to role` operations did you find?

Hint

Each entry starts with the field `time`.

Answer

There are two unique entries at the following timestamps:

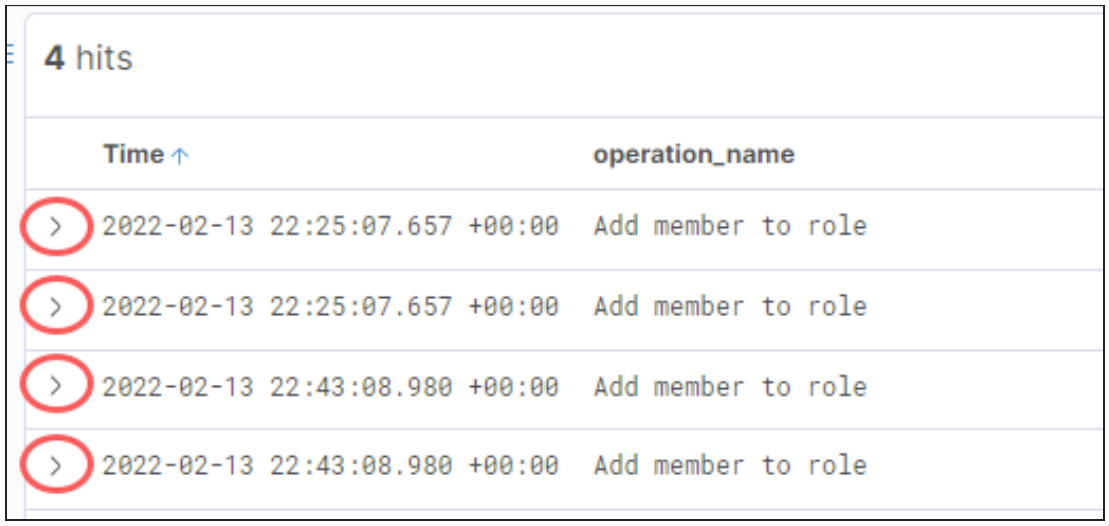
- 2022-02-13 22:25:07
- 2022-02-13 22:43:08

As discussed in the class, the Azure logstash parser splits entries that have information stored in an array (hence the four lines in Kibana). You will need to look at each entry to find the information you are looking for.

Which accounts were modified?

Hint

Expand each entry by clicking on `>`



Look for the section called `target_resource`

Answer

- 2022-02-13 22:25:07: `JVanDyne@pymtechlabs.com`
- 2022-02-13 22:43:08: `Hydra@pymtechlabs.com`

Which role was granted?

Hint

Under the same section, look at the entries that end in `*.newValue`

Answer

In both cases, you will notice the same value for `target_resource_modifications.Role.ObjectID.newValue`. The preceding field `target_resource_modifications.Role.DisplayName.newValue` provides the name in human readable form: "Company Administrator", which is better known as Global Admin. What really matters is the value of `Role.ObjectID` which will be different for every tenant. You can find this value for your own tenant with the following PowerShell command:

Command line

```
Get-AzureADDirectoryRole | ?{$_.DisplayName -eq "Global Administrator"}
```

For Pymtechlabs, the Global Admin role Object ID is `dca97a1c-fe1d-45ca-b599-ee095a2dd316`

Who granted the role?

> **Hint**
>
> Look for the field `identity`.

Answer

The change was initiated by `QuantumApp`.

t identity	QuantumApp
t input.type	log
IP ips	20.190.160.97
t log.file.path	/logstash/azure/lab-1.3_source_evidence.json
# log.offset	37,691
t logged_by_service	Core Directory
t operation_name	Add member to role
t operation_type	Assign
t property_category	RoleManagement

The Azure logstash parser simplifies this field too much. It's preferable to look at the raw logs using the following command (in your SOF-ELK terminal window) to query the raw logs and proceed to the next question:

Command line

```
cat /logstash/azure/lab-1.3_source_evidence/lab-1.3_source_evidence.json | jq
'select(.operationName == "Add member to role")'
```

Look for the block called `initiatedBy`. You will find `QuantumApp` and more importantly see that it's associated with a service principal of `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`. It's the same for both entries.

Expected result fragment

```
"initiatedBy": {
  "app": {
    "appId": null,
    "displayName": "QuantumApp",
    "servicePrincipalId": "c0f487ec-a338-4ebd-8ecd-4afef80daa1f",
    "servicePrincipalName": null
  }
},
```

Which elements would help you ascertain that the identity from the previous question is a Graph API application (no hint)?

Answer

QuantumApp is a Graph API application based on the facts that:

a. It acted via a service principal

b. The IP address in the record belongs to Microsoft

t source_geo.as_org	MICROSOFT-CORP-MSN-AS-BLOCK
# source_geo.asn	8,075
t source_geo.city_name	Amsterdam
t source_geo.continent_code	EU
t source_geo.country_code2	NL
t source_geo.country_code3	NL
t source_geo.country_name	Netherlands
# source_geo.latitude	52.375
source_geo.location	{ "coordinates": [4.8975, 52.3759], "type": "Point" }
# source_geo.longitude	4.898
t source_geo.postal_code	1012
t source_geo.region_code	NH
t source_geo.region_name	North Holland
t source_geo.timezone	Europe/Amsterdam
IP source_ip	20.190.160.97

What is the significance of these findings (no hint)?

Answer

We were expecting one account to be given the role of Global Admin, but instead we found two. We found that both accounts were granted this permission by the Graph API application called QuantumApp. Based on the scenario, we expected to find Janet Van Dyne's account, but not Hydra. Where did this account come from? This will also need to be investigated. During investigations, it's very common to find other "interesting" facts which can quickly lead to information overload. To keep our investigation on track, we need to start a timeline:

- 2022-02-13 22:25:07: `JVanDyne@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`
- 2022-02-13 22:43:08: `Hydra@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`

3. Investigate the application

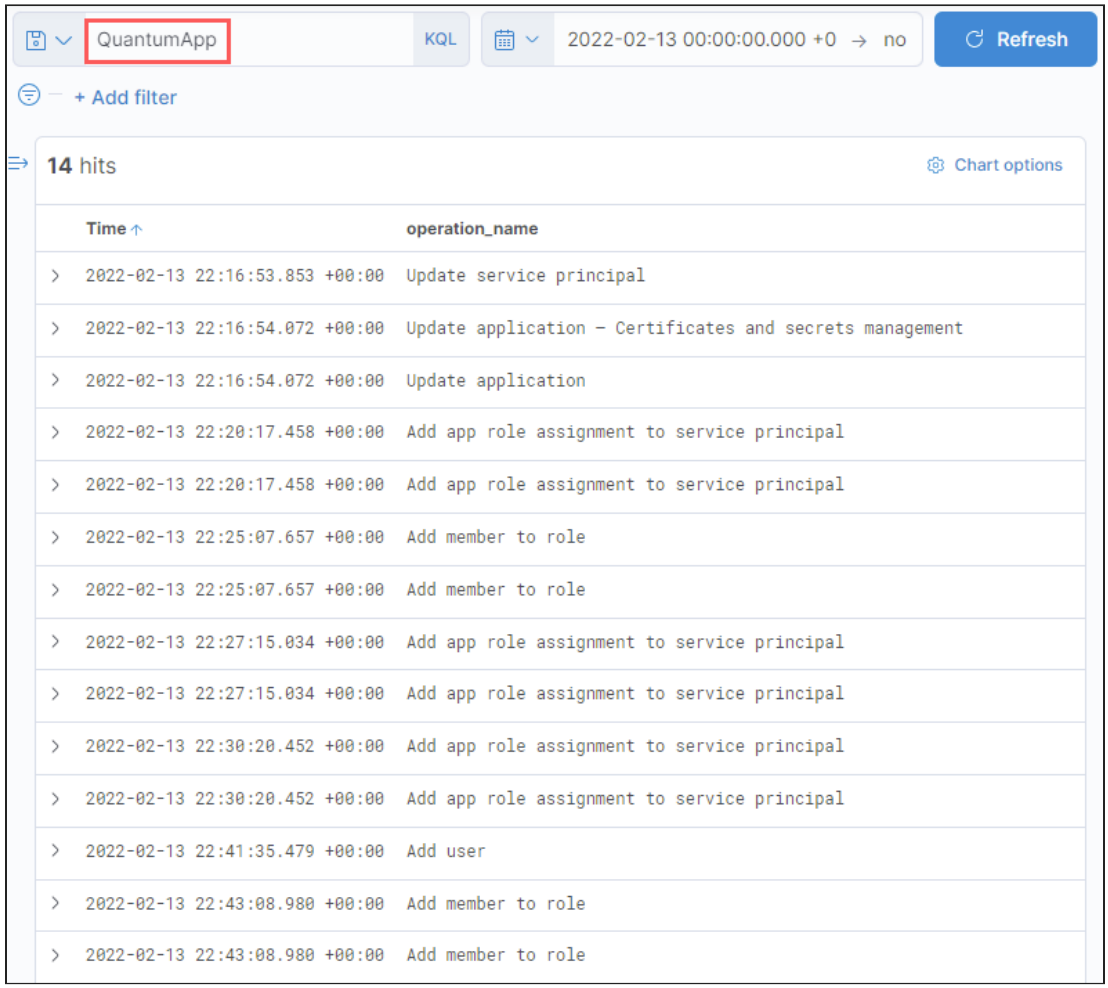
What activities are attributed to this app?

Hint

Search for `QuantumApp` in SOF-ELK.

Answer

We have two operations of particular interest to investigate: - `Add app role assignment to service principal` - `Add user`



Time	operation_name
2022-02-13 22:16:53.853 +00:00	Update service principal
2022-02-13 22:16:54.072 +00:00	Update application – Certificates and secrets management
2022-02-13 22:16:54.072 +00:00	Update application
2022-02-13 22:20:17.458 +00:00	Add app role assignment to service principal
2022-02-13 22:20:17.458 +00:00	Add app role assignment to service principal
2022-02-13 22:25:07.657 +00:00	Add member to role
2022-02-13 22:25:07.657 +00:00	Add member to role
2022-02-13 22:27:15.034 +00:00	Add app role assignment to service principal
2022-02-13 22:27:15.034 +00:00	Add app role assignment to service principal
2022-02-13 22:30:20.452 +00:00	Add app role assignment to service principal
2022-02-13 22:30:20.452 +00:00	Add app role assignment to service principal
2022-02-13 22:41:35.479 +00:00	Add user
2022-02-13 22:43:08.980 +00:00	Add member to role
2022-02-13 22:43:08.980 +00:00	Add member to role

What are the three role assignments that have been added?

Hint

Open each `Add app role assignment to service principal` record and look for the field `target_resource_modifications.AppRole.Value.newValue`.

Answer

- 2022-02-13T22:20:17.4585613Z: RoleManagement.ReadWrite.Directory
- 2022-02-13T22:27:15.0346247Z: User.ReadWrite.All
- 2022-02-13T22:30:20.4525771Z: Directory.ReadWrite.All

t target_resource_modifications.AppRole.DisplayName.newValue	"Read and write all directory RBAC settings"
t target_resource_modifications.AppRole.Id.newValue	"9e3f62cf-ca93-4989-b6ce-bf83c28f9fe8"
t target_resource_modifications.AppRole.Value.newValue	"RoleManagement.ReadWrite.Directory"

t target_resource_modifications.AppRole.DisplayName.newValue	"Read and write all users' full profiles"
t target_resource_modifications.AppRole.Id.newValue	"741f803b-c850-494e-b5df-cde7c675a1ca"
t target_resource_modifications.AppRole.Value.newValue	"User.ReadWrite.All"

t target_resource_modifications.AppRole.DisplayName.newValue	"Read and write directory data"
t target_resource_modifications.AppRole.Id.newValue	"19dbc75e-c2e2-444c-a770-ec69d8559fc7"
t target_resource_modifications.AppRole.Value.newValue	"Directory.ReadWrite.All"

Bonus

If you want to get the same information from the raw logs, you can use the following command:

Command line

```
cat /logstash/azure/lab-1.3_source_evidence/lab-1.3_source_evidence.json | jq 'select(.identity
== "QuantumApp")' | jq 'select (.operationName == "Add app role assignment to service
principal")'
```

For each record, look at `properties`, then `targetResources`, then `modifiedProperties`. Focus on the field `AppRole.Value`.

Expected result fragments

```
{
  "displayName": "AppRole.Value",
  "oldValue": null,
  "newValue": "\"RoleManagement.ReadWrite.Directory\""
},
```

```
{
  "displayName": "AppRole.Value",
  "oldValue": null,
  "newValue": "\"User.ReadWrite.All\""
},
```

```
{
  "displayName": "AppRole.Value",
  "oldValue": null,
  "newValue": "\"Directory.ReadWrite.All\""
},
```

Who is changing whom?

Hint

- SOF-ELK: Look for `identity` and `target_resource`
- Raw logs: Look for `initiatedBy` and `targetResources`

This question is easier to answer by looking at the raw logs.

Answer

QuantumApp is changing its own permissions!

Expected result fragment


```
"initiatedBy": {
  "app": {
    "appId": null,
    "displayName": "QuantumApp",
    "servicePrincipalId": "c0f487ec-a338-4ebd-8ecd-4afef80daa1f",
    "servicePrincipalName": null
  }
},
"targetResources": [
  {
    "id": "9f032356-d695-4089-b670-587e606a82cf",
    "displayName": "Microsoft Graph",
    "type": "ServicePrincipal",
    "modifiedProperties": [
      {
        "displayName": "AppRole.Id",
        "oldValue": null,
        "newValue": "\"9e3f62cf-ca93-4989-b6ce-bf83c28f9fe8\""
      },
      {
        "displayName": "AppRole.Value",
        "oldValue": null,
        "newValue": "\"RoleManagement.ReadWrite.Directory\""
      },
      {
        "displayName": "AppRole.DisplayName",
        "oldValue": null,
        "newValue": "\"Read and write all directory RBAC settings\""
      },
      {
        "displayName": "ServicePrincipal.ObjectID",
        "oldValue": null,
        "newValue": "\"c0f487ec-a338-4ebd-8ecd-4afef80daa1f\""
      },
      {
        "displayName": "ServicePrincipal.DisplayName",
        "oldValue": null,
        "newValue": "\"QuantumApp\""
      },
```

At this point, you are certainly wondering, how is this possible? This is not something you can answer from the logs unless you go back to the creation of the application (which may or may not be possible depending on your company's log retention policy).

This is possible because at some point, QuantumApp was granted `AppRoleAssignment.ReadWrite.All` which is a very powerful permission since it allows an app to grant permissions to another app (including itself).

We know the service principal for QuantumApp. What's its Application ID?

Hint

There are three key elements to defining a Graph API application:

- Display Name: QuantumApp
- Service Principal: c0f487ec-a338-4ebd-8ecd-4afef80daa1f
- Application ID:

Look for the field:

- SOF-ELK: `target_resources.displayName` or `target_resource_modifications.SPN.newValue`
- Raw logs: `ServicePrincipal.AppId`

Answer

- Application ID: efd53fc0-2438-43f5-851a-d88e46323305

This value will be very valuable to track the sign-in events.

- SOF-ELK

- Raw Logs

Expected result fragment


```
{
  "displayName": "ServicePrincipal.AppId",
  "oldValue": null,
  "newValue": "\"efd53fc0-2438-43f5-851a-d88e46323305\""
},
```

Which user is being added?

Hint

Open the `Add user` record and look for the field `target_resources.userPrincipalName`.

Answer

2022-02-13 22:41:35: `Hydra@pymtechlabs.com`

This is the account we saw earlier and now know that the QuantumApp application created this account. Adding accounts is a great way to establish persistence.

Bonus

If you want to get the same information from the raw logs, you can use the following command:

Command line

```
cat /logstash/azure/lab-1.3_source_evidence/lab-1.3_source_evidence.json | jq
'select(.operationName == "Add user")'
```

Update your timeline (no hint)

Answer

- 2022-02-13 22:20:17: QuantumApp assigned `RoleManagement.ReadWrite.Directory` permission to itself
- 2022-02-13 22:25:07: `JVanDyne@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`
- 2022-02-13 22:27:15: QuantumApp assigned `User.ReadWrite.All` permission
- 2022-02-13 22:30:20: QuantumApp assigned `Directory.ReadWrite.All` permission
- 2022-02-13 22:41:35: QuantumApp created user `Hydra@pymtechlabs.com`
- 2022-02-13 22:43:08: `Hydra@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`

4. Who is controlling the application?

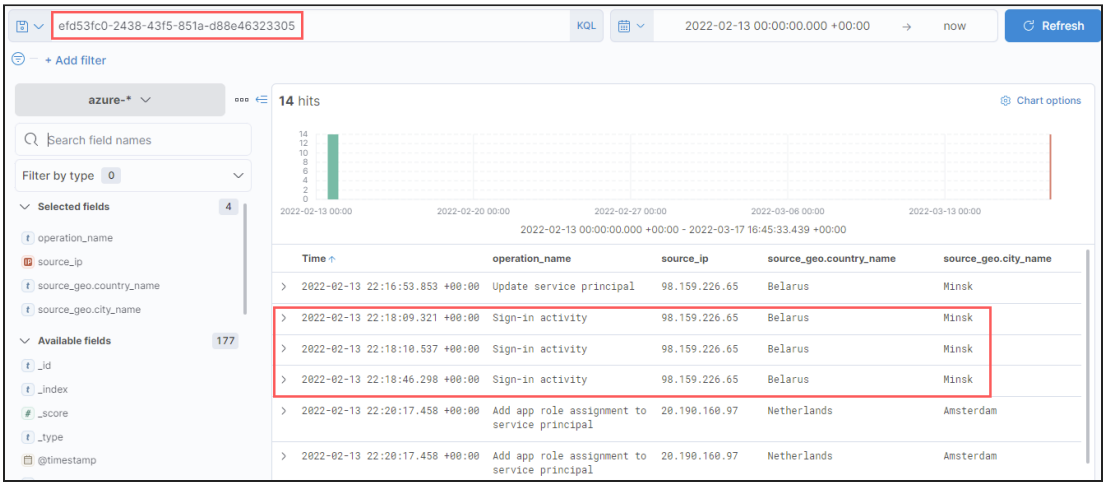
What's QuantumApp doing right before 2022-02-13 22:20:17?

Hint

In SOF-ELK, query the application ID we found earlier: `efd53fc0-2438-43f5-851a-d88e46323305`

Answer

Looks like there is some interesting sign-in activity. Add the fields `source_ip`, `source_geo.country_name`, and `source_geo.city_name` for even more interesting information.



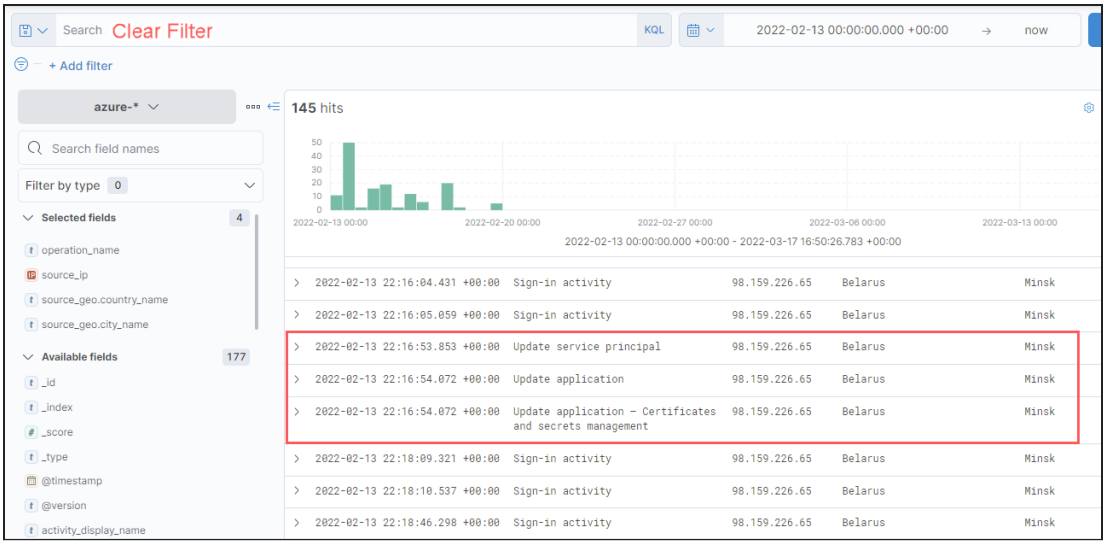
What else is this suspicious IP doing?

Hint

Remove the Application ID filter and focus on the events right before the sign-in activity.

Answer

A new secret (same as password) is being added to QuantumApp.



Who performed the interesting operation from the last question?

Hint

Expand any one of the three entries. However, the entry with the operation name `Update application – Certificates and secrets management` is the most interesting.

Answer

Janet Van Dyne added a new secret to QuantumApp (requires looking at the raw log to see that QuantumApp is the target). A secret is the same thing as a password.

initiating_user_ip	98.159.226.65
initiating_user_principal_name	JVanDyne@pymtechlabs.com
input.type	log
ips	98.159.226.65
log.file.path	/logstash/azure/lab1.3-aad_sorted1.json
log.offset	24,538
logged_by_service	Core Directory
operation_name	Update application – Certificates and secrets management
operation_type	Update

If you look at the entry at timestamp `2022-02-13 22:15:27.693`, you will see that JVanDyne connected using Microsoft Azure PowerShell and successfully completed both primary and secondary authentication. At `2022-02-13 22:16:04`, JVanDyne then connects to AzureAD (info requires digging in the raw logs).

Given that Minsk, Belarus is not Janet's normal location, it looks like her credentials have been compromised.

Update your timeline (no hint)

Answer

- 2022-02-13 22:15:27: JVanDyne connects to Azure
- 2022-02-13 22:16:04: JVanDyne connects to AzureAD
- 2022-02-13 22:16:54: New secret created for QuantumApp
- 2022-02-13 22:18:09: QuantumApp sign-in activity (also at 22:18:10 and 22:18:46)
- 2022-02-13 22:20:17: QuantumApp assigned `RoleManagement.ReadWrite.Directory` permission
- 2022-02-13 22:25:07: `JVanDyne@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`
- 2022-02-13 22:27:15: QuantumApp assigned `User.ReadWrite.All` permission
- 2022-02-13 22:30:20: QuantumApp assigned `Directory.ReadWrite.All` permission
- 2022-02-13 22:41:35: QuantumApp created user `Hydra@pymtechlabs.com`
- 2022-02-13 22:43:08: `Hydra@pymtechlabs.com` granted Global Admin role by QuantumApp, service principal `c0f487ec-a338-4ebd-8ecd-4afef80daa1f`

5. Any further actions?

Any action taken by the threat actor after the privilege escalation?

Hint

Look for activity after the last log entry from your timeline.

Answer

At `2022-02-13 22:47:28`, the threat actor logs on again as JVanDyne using PowerShell. This is necessary in order to establish a PowerShell session with a token that has the Global Admin permission. However, we no longer see any activity originating from Belarus.

Add the field `initiating_user_principal_name` to your display, and you will see that on 2022-02-16, JVanDyne is taking some new actions but from the United States. Is this legitimate? No way to tell from the logs. It would be up to you to follow your incident response process once you discovered that there is a high likelihood that the JVanDyne identity has been compromised.

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Certain Graph API permissions can lead to privilege escalation
- In this example we escalated from `AppRoleAssignment.ReadWrite.All` to `RoleManagement.ReadWrite.Directory` to Global Admin role
- In the future other escalation paths are bound to be discovered
- Understand that GUIDs in logs can represent Azure AD object, API permissions, and many other things (some of which aren't documented)
- Logs entries that refer to actions taken by service principals will show Microsoft IP addresses, requiring much deeper investigation
- Be sure to safeguard credentials to Graph API applications
- This lab was inspired by the work from the SpecterOps team: https://posts.specterops.io/azure-privilege-escalation-via-azure-api-permissions-abuse-74aee1006f48

Lab 2.1: Using SOF-ELK® with Azure Logs

Objectives

- Become familiar with the Pymtechlabs environment
- Discover users in Pymtechlabs
- Discover the machines in Pymtechlabs
- Discover the "normal" IPs Pymtechlabs employees may be using to access cloud resources

Background

Pymtechlabs has received intelligence that some of its cloud systems have been compromised. The regular IT team had an emergency and is unavailable. Before they left, they downloaded the logs from Azure and imported them to the SIEM (SOF-ELK). You will need to figure out which system(s) and account(s) have been compromised.

Since we haven't talked about the different log sources yet, this lab will guide you through the various steps to familiarize you with SOF-ELK and the Pymtechlabs environment.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-12 00:00 UTC** to **2021-04-02 00:00 UTC**.

Load Data

The Azure logs have been exported and saved in `/home/elk_user/lab-2.1_source_evidence.zip`

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract the files from the ZIP file above with the following commands:

Command lines

```
cd ~
unzip -q lab-2.1_source_evidence.zip
cp ./lab-2.1_source_evidence/*.json /logstash/azure
```

Warning

Do not run these commands more than once!

3. Wait 2-3 minutes for the data to be processed.
4. Verify that you have 5,615 documents loaded in the `azure` index:

Command line

```
sof-elk_clear.py -i list
```

Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- azure (5,615 documents)
- office365 (5,462 documents)
```

5. Once completed, the data will be available in the `azure-*` index.

Lab Content

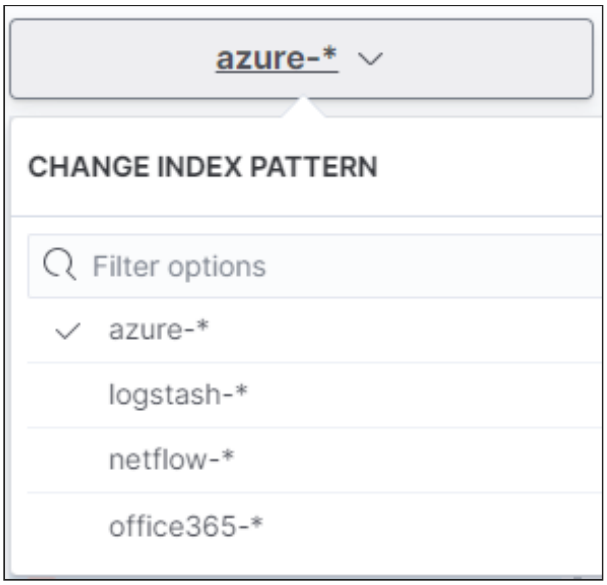
Time Frame

`2021-03-12 00:00:00.000 +00:00` to `2021-04-02 23:30:00.000 +00:00`

Review the Indices in SOF-ELK

Start in the Discover tab

When data is imported in SOF-ELK, it's assigned to a specific index. In this lab we will be using the `azure-*` index.



This index contains any one of the following logs:

- insights-activity-logs
- insights-logs-auditlogs
- insights-managedidentitysigninlogs
- insights-noninteractiveusersigninlogs
- insights-serviceprincipalsigninlogs
- insights-logs-signinlogs

The activity, audit, and signin logs have different schemas, so as you look at various events, you will notice the following:

- Not all fields contain data
- Some data is repeated in different fields
- Data in a field may be formatted one way in one record and a different way in another

It's important to be aware of these issues and not let them impede your investigation.

Discover Users in Pymtechlabs

Our investigation will be easier if we know the various users in the Pymtechlabs network.

1. Fields with username information

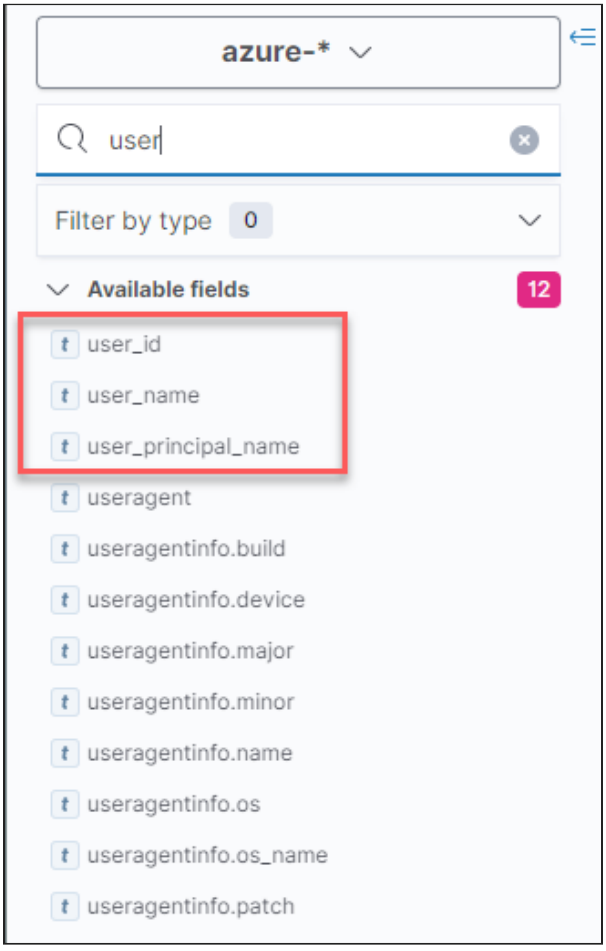
Looking at the field names on the left, what are some fields that may contain username information?

Hint

In the "Search field names" box, type `user`

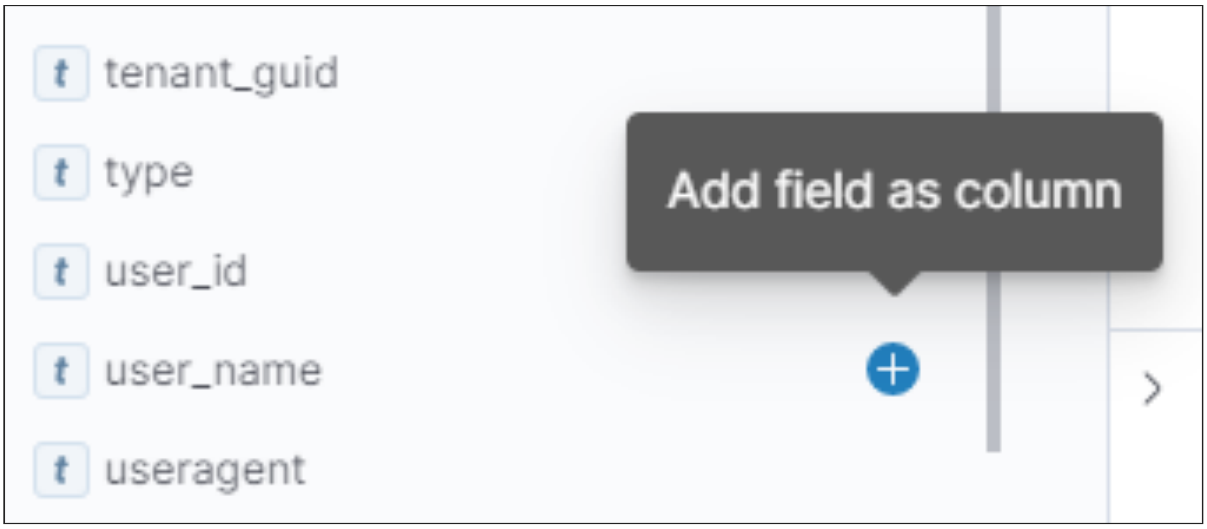
Answer

- user_id
- user_name
- user_principal_name



- user_id: The GUID of the user
- user_name: The name of the user
- user_principal_name = UPN - the email & login ID

For each field in the previous question, select the `+` sign next to it to add it as a column.



Your screen will change from a list with all the raw records to a list with the 3 fields mentioned above as columns:

Time	user_name	user_principal_name	user_id
2021-04-01 17:19:35.532 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9
2021-04-01 17:19:34.798 +00:00	-	-	-
2021-04-01 17:19:34.728 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9
2021-04-01 17:19:34.643 +00:00	-	-	-
2021-04-01 17:18:41.393 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9

Some records do not contain any information. It would be nice to eliminate them from the search. Use the following filter:

```
user_name: *
```

Don't forget to hit **`Update`** after entering your filter.

2021-04-01 16:04:54.156 +00:00	Janet Van Dyne	jvandyne@pymtechlabs.com	76362af6-e38a-41e7-adab-e21ad7e23a20
2021-04-01 16:04:54.123 +00:00	Janet Van Dyne	jvandyne@pymtechlabs.com	76362af6-e38a-41e7-adab-e21ad7e23a20
2021-04-01 02:00:21.839 +00:00	Hank Pym	admin@pymtechlabs.com	675be0f4-2486-4443-bef6-d37d9043ae99
2021-03-31 22:39:02.496 +00:00	Hank Pym	admin@pymtechlabs.com	675be0f4-2486-4443-bef6-d37d9043ae99
2021-03-31 21:38:06.509 +00:00	Hank Pym	admin@pymtechlabs.com	675be0f4-2486-4443-bef6-d37d9043ae99
2021-03-31 20:37:52.483 +00:00	Hank Pym	admin@pymtechlabs.com	675be0f4-2486-4443-bef6-d37d9043ae99
2021-03-31 20:17:44.901 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9
2021-03-31 19:46:25.741 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9
2021-03-31 19:46:25.489 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9
2021-03-31 19:46:24.326 +00:00	Luis	luis@pymtechlabs.com	1c587fa4-0dbb-472c-bdc8-5477beefb3a9

We now have a long list of records with repeated names. Let's jump to the Visualization tab to create a table of unique usernames.

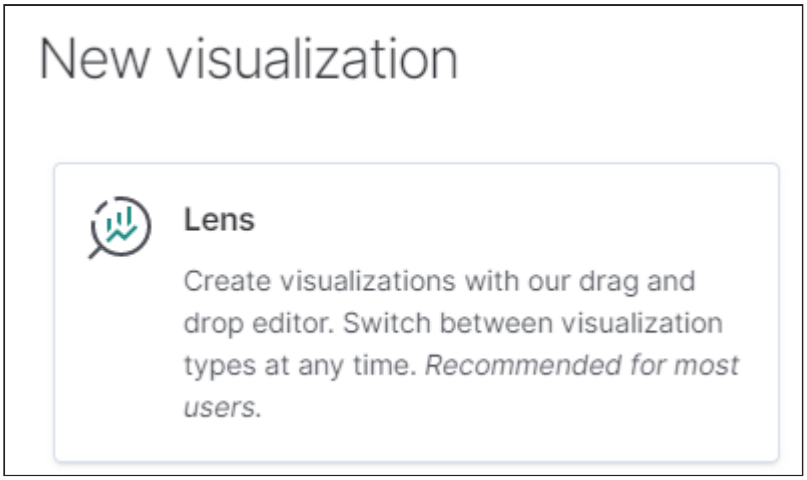
> **Note**
>
> Clear the `user_name: *` filter and hit `Update` before you proceed further

2. Create a table of unique usernames

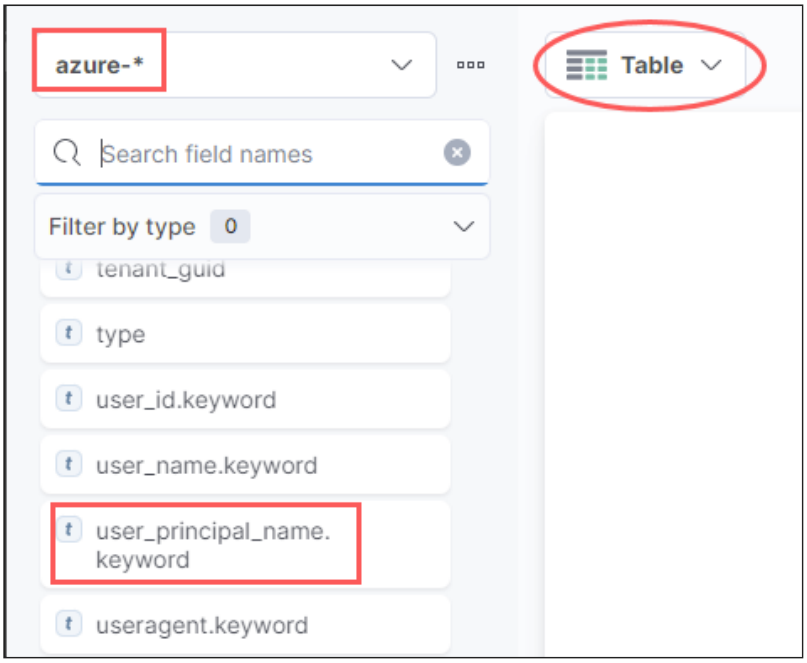
   This question will be answered in the Visualize Library tab

   Jump to the `Visualize Library` tab in Kibana and select `Create visualization`.

   Choose `Lens` because it's the easiest way to create a new visualization.



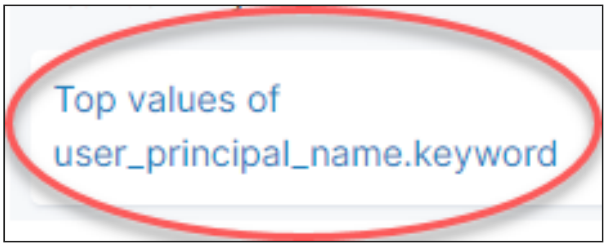
   Be sure to be on the correct index, `azure-*`, or the field filters menu will be empty.



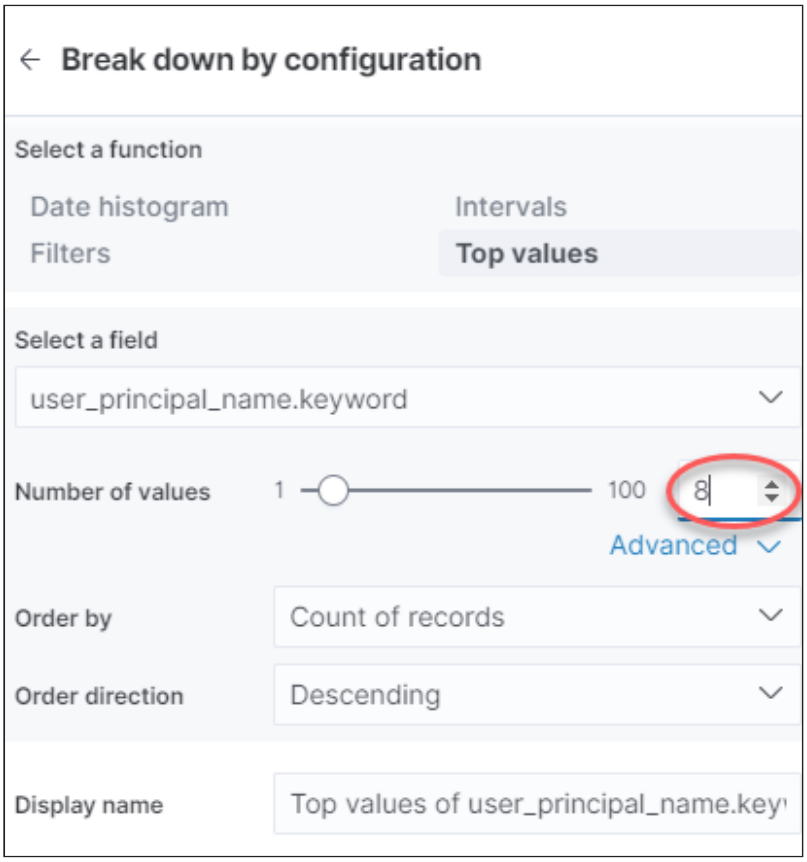
Change your graph type to `Table` and drag and drop `user_principal_name.keyword` to the middle of your screen.

We now get a table, but there is a field called `Other` that's hiding valuable information. We need to allow more rows in our table.

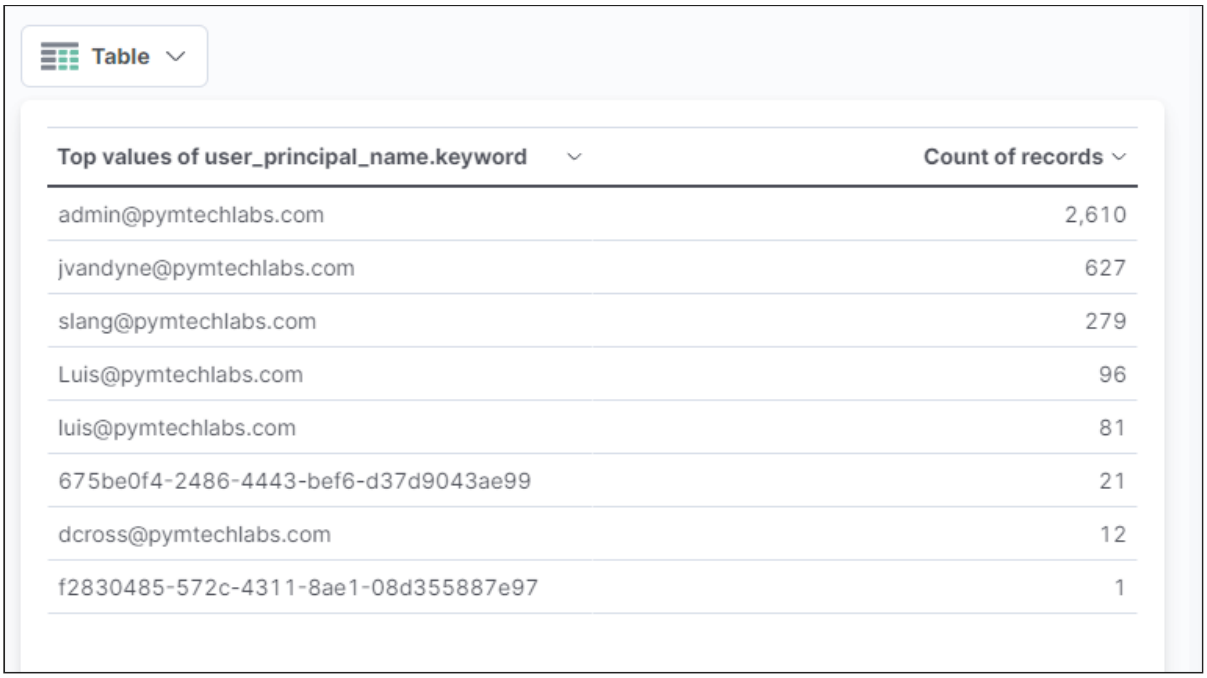
On the right of your screen, select `Top values of user_principal_name.keyword`.



You will now get a menu where you can add more rows.



You now have a table that shows you the unique UPNs for Pymtechlabs as well as the number of occurrences each UPN appears in the logs.



Note that luis is spelled once with a lowercase `l` and once with an uppercase `L`. This has no special meaning other than to point out the inconsistencies between different log schemas.

> **Note**
>
> If you have a different count, it's most likely because you didn't clear the `user.name: *` filter. You can do that now and hit `Refresh` to get the same results, as shown in the picture above.

3. Question about the admin UPN

Who does the `admin@pymtechlabs.com` UPN belong to?

Hint

Drag the `user_name.keyword` field on your existing table

Answer

`admin@pymtechlabs.com` belongs to Hank Pym

Top values of user_principal_name.keyword	Top values of user_name.keyword	Count of records
admin@pymtechlabs.com	Hank Pym	1,036
jvandyne@pymtechlabs.com	Janet Van Dyne	627
slang@pymtechlabs.com	Scott Lang	279
luis@pymtechlabs.com	Luis	82
675be0f4-2486-4443-bef6-d37d9043ae99	675be0f4-2486-4443-bef6-d37d9043ae99	21
dcross@pymtechlabs.com	Darren Cross	12
f2830485-572c-4311-8ae1-08d355887e97	f2830485-572c-4311-8ae1-08d355887e97	1

You will notice that some GUIDs sneaked into the `user_principal_name` field. The GUID ending in `ae99` belongs to Hank Pym (admin@pymtechlab.com) and the one ending in `7e97` belongs to Scott Lang (`slang@pymtechlabs.com`). Again, log entries choose to store information in various, inconsistent ways.

In your own environment (outside the scope of this lab), another way to convert the GUID to the user name is with the PowerShell command `Get-AzureADObjectByObjectId -objectids <GUID>` . For example, in pymtechlabs.com we would get the following results:

```
PS C:\> Get-AzureADObjectByObjectId -objectids 675be0f4-2486-4443-bef6-d37d9043ae99

ObjectId                             DisplayName UserPrincipalName     UserType
--------                             ----------- -----------------     --------
675be0f4-2486-4443-bef6-d37d9043ae99 Hank Pym    admin@pymtechlabs.com Member
```

```
PS C:\> Get-AzureADObjectByObjectId -objectids f2830485-572c-4311-8ae1-08d355887e97

ObjectId                             DisplayName UserPrincipalName     UserType
--------                             ----------- -----------------     --------
f2830485-572c-4311-8ae1-08d355887e97 Scott Lang  slang@pymtechlabs.com Member
```

Discover the Machines in Pymtechlabs

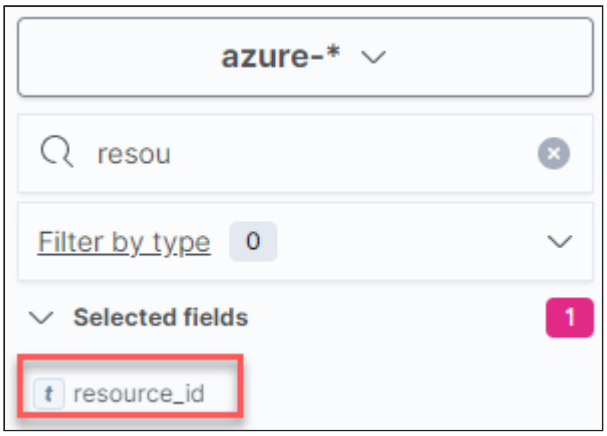
There is no simple query to discover the names of the machines in Pymtechlabs. However, remember that virtual machines belong to the `Microsoft.Compute` provider. Every log entry contains the name of the resource that's being operated on. This information is contained in the field `resource_id`.

Use the Discover tab for this question

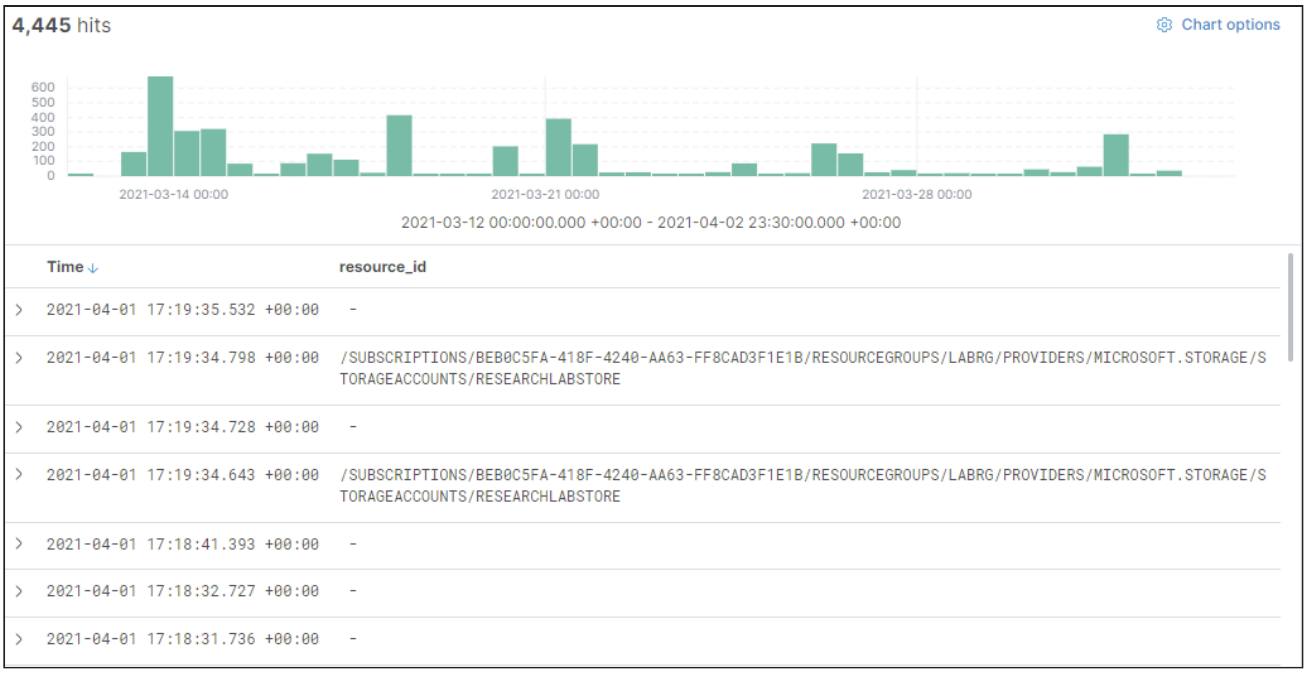
When you select the Discover tab (after completing your work in the Lens tab), you will get a message indicating that you have unsaved changes. Just select `Confirm`.

Once you are back in the Discover tab, if there is any remaining filter, be sure to remove it.

In the Discover tab, select the field `resource_id`.



You will get a table that shows every resource found in the logs as well as a lot of blank lines.



1. Filter for virtual machines

Write a filter to limit the search to virtual machines

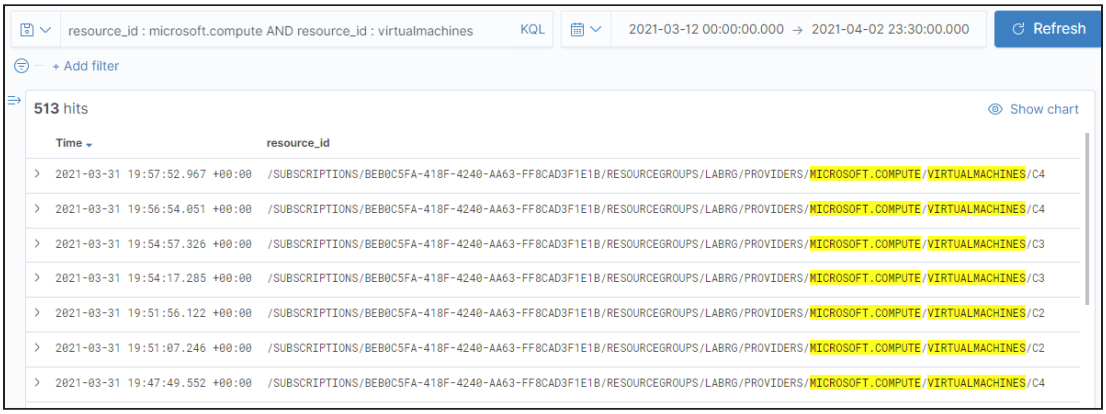
Hint

Virtual machines are created via the `microsoft.compute` resource provider

Answer

Create the following search:

```
resource_id : microsoft.compute AND resource_id : virtualmachines
```



2. Get a list of unique machines

Use the Visualize Library tab for this question

What are the names of the machines in Pymtechlabs?

Hint

Using the same process we used to find the unique UPNs, create a table of unique values using the `resource_id.keyword` field.

Answer

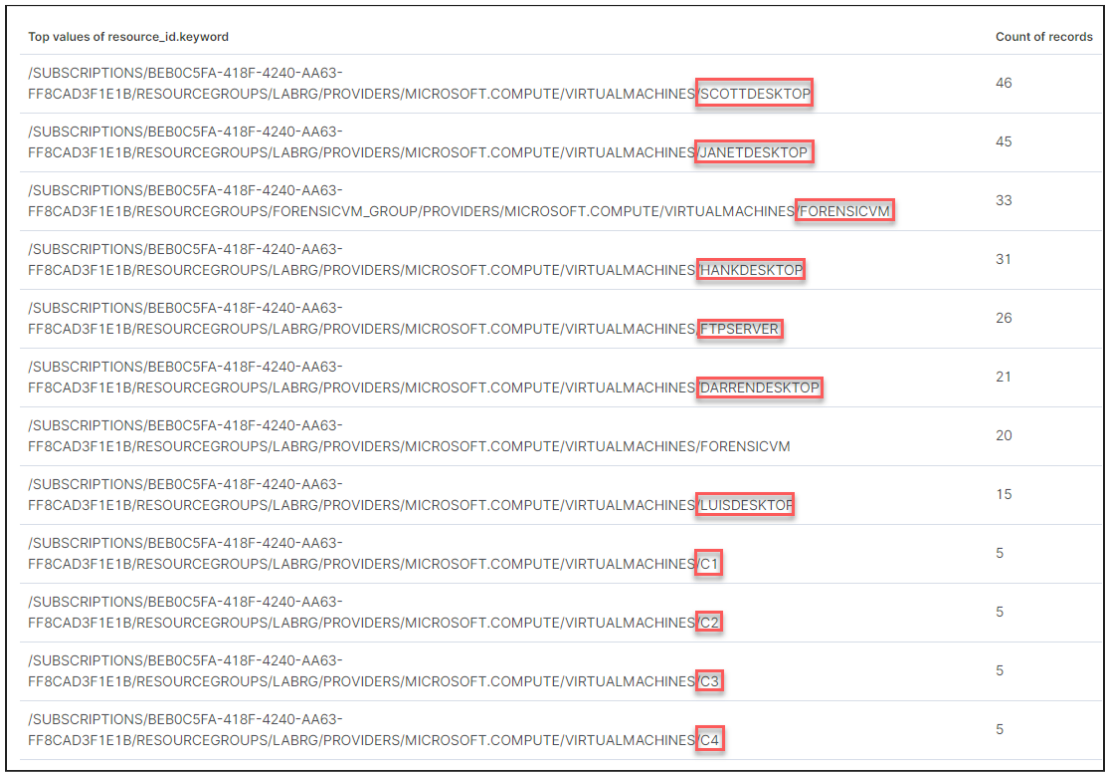
A simple search is not enough due to a lot of noise we need to filter. One potential search is:

```
resource_id : microsoft.compute AND resource_id : virtualmachines AND not extensions AND not
MICROSOFT.AUTHORIZATION AND not MICROSOFT.INSIGHTS
```

Also, don't forget to expand the number of rows in your table.

The names of the machines in Pymtechlabs are:

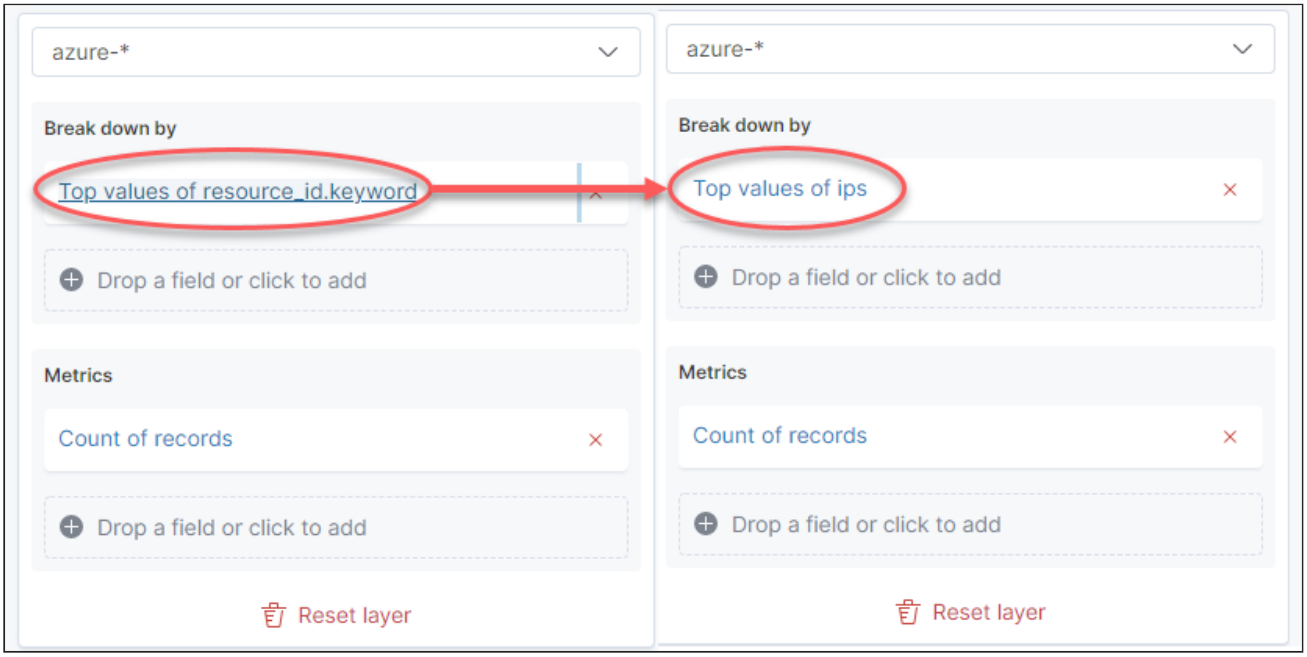
- hankdesktop
- scottdesktop
- janetdesktop
- darrendesktop
- luisdesktop
- ftpserver
- forensicvm
- c1
- c2
- c3
- c4



Top values of resource_id.keyword	Count of records
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/SCOTTDESKTOP	46
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/JANETDESKTOP	45
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/FORENSICVM_GROUP/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/FORENSICVM	33
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/HANKDESKTOP	31
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/FTPSERVER	26
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/DARRENDESKTOP	21
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/FORENSICVM	20
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/LUISDESKTOP	15
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1	5
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C2	5
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C3	5
/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C4	5

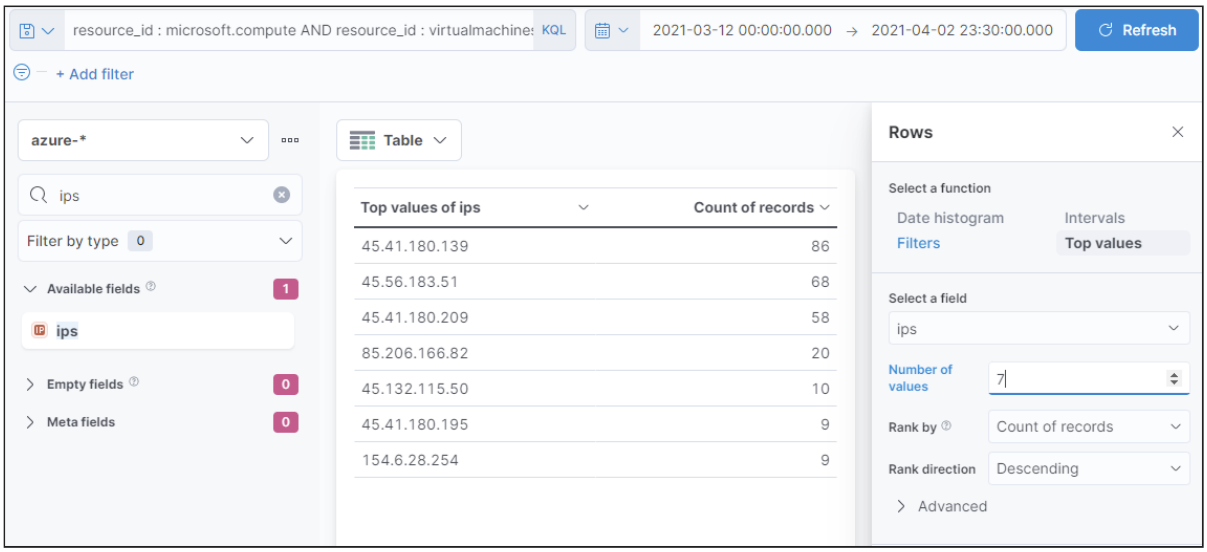
Discover the "Normal" IPs Pymtechlabs Employees May Be Using to Access Cloud Resources

We have a table with our unique VM names. If we just replace `resource_id.keyword` with `ips`, we will get the answer we are looking for. Be sure to leave the noise filter on and don't forget to add rows to your table.



What IP addresses did you find (no hint)?

- 45.41.180.139
- 45.56.183.51
- 45.41.180.209
- 85.206.166.82
- 45.132.115.50
- 45.41.180.195
- 154.6.28.254



Given that Pymtechlabs' corporate network uses the 45.x.x.x subnet, which IPs should be investigated further?

Hint

Add `AND NOT ips:45.0.0.0/8` at the end of the previous search.

Answer

- 85.206.166.82
- 154.6.28.254

Information about these IP addresses

- `85.206.166.82` : This address is not part of Pymtechlabs' corporate network. We will discover its meaning in a future lab.
- `154.6.28.254` : This address is not part of Pymtechlabs' corporate network and should be investigated further. However, it turns out to be a VPN address used during testing.

Import Data into SOF-ELK

We previously copied logs to the `/logstash` folder to ingest data in SOF-ELK. However, the logs had already been packaged in a single JSON file. In this bonus section, we are giving you the individual raw files, which are stored in `PT1H.json` files for every hour in which a log entry was generated. The lab will cover the steps to combine these individual files in a single file so it can be loaded in SOF-ELK.

Log on to the SOF-ELK VM with the following credentials:

- Username: ***elk_user***
- Password: ***forensics***

You may log on to the console; however, it's recommended to log in via SSH.

The audit log has been downloaded in the folder `/home/elk_user/lab-2.1_source_evidence/insights-logs-auditlogs`.

1. You can observe the directory structure by using the `tree` command.

Command lines

```
cd /home/elk_user/lab-2.1_source_evidence/insights-logs-auditlogs
tree
```

Expected results

```
[elk_user@sof-elk insights-logs-auditlogs]$ tree
.
`-- tenantId=7e325eda-7945-46d3-ac99-f0dcfeb4628e
    `-- y=2021
        `-- m=03
            |-- d=13
            |   |-- h=02
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   |-- h=18
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   |-- h=19
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   |-- h=21
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   `-- h=22
            |       `-- m=00
            |           `-- PT1H.json
            |-- d=14
            |   |-- h=01
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   `-- h=15
            |       `-- m=00
            |           `-- PT1H.json
            |-- d=18
            |   `-- h=02
            |       `-- m=00
            |           `-- PT1H.json
            |-- d=20
            |   `-- h=01
            |       `-- m=00
            |           `-- PT1H.json
            |-- d=21
            |   |-- h=01
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   `-- h=02
            |       `-- m=00
            |           `-- PT1H.json
            |-- d=26
            |   |-- h=01
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   |-- h=02
```

```
            |   |   `-- m=00
            |   |       `-- PT1H.json
            |   `-- h=20
            |       `-- m=00
            |           `-- PT1H.json
            `-- d=31
                |-- h=01
                |   `-- m=00
                |       `-- PT1H.json
                `-- h=17
                    `-- m=00
                        `-- PT1H.json

42 directories, 16 files
```

2. Combine these `PT1H.json` files into a single file: `insights-logs-auditlogs.json`.

Command lines

```
cd /home/elk_user/lab-2.1_source_evidence
find ./insights-logs-auditlogs/ -type f -name PT1H.json -exec cat {} + |tee insights-logs-
auditlogs.json
ls -l insights-logs-auditlogs.json
```

Notional results

```
[elk_user@sof-elk lab-2.1_source_evidence]$ find ./insights-logs-auditlogs/ -type f -name
PT1H.json -exec cat {} + |tee insights-logs-auditlogs.json
< JSON output omitted >
[elk_user@sof-elk lab-2.1_source_evidence]$ ls -l insights-logs-auditlogs.json
-rw-rw-r-- 1 elk_user elk_user 774372 Oct 26 23:33 insights-logs-auditlogs.json
```

3. Copy the file to the correct Logstash directory so it can be ingested into SOF-ELK:

Command lines

```
cp insights-logs-auditlogs.json /logstash/azure
```

4. We just added 266 documents to the azure index. You should now have a total of 5,881 documents in the azure index:

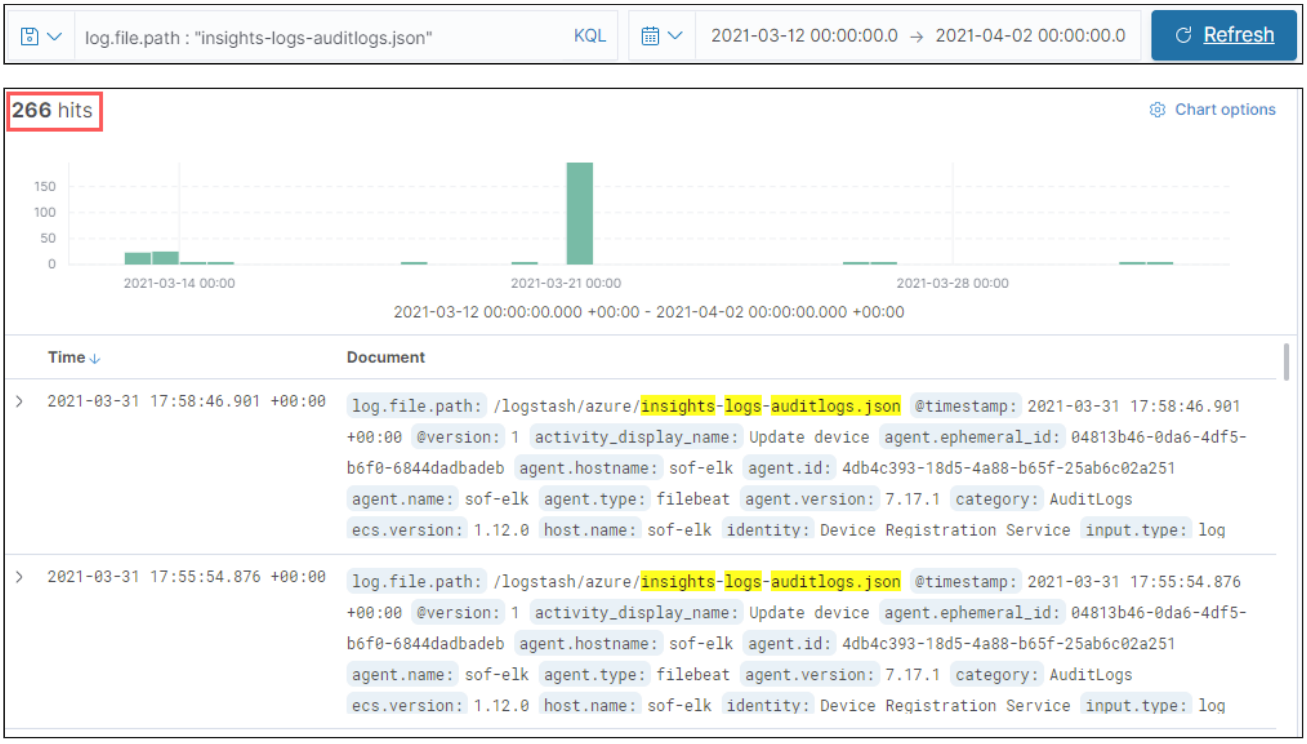
Command lines

```
sof-elk_clear.py -i list
```

Expected results

```
[elk_user@sof-elk lab-2.1_source_evidence]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- azure (5,881 documents)
- office365 (5,462 documents)
```

5. You can now check in Kibana that you have events from that log (give SOF-ELK a couple minutes to ingest all the data):
   `log.file.path: "insights-logs-auditlogs.json"`



Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Pymtechlabs has 5 users

- Pymtechlabs has 11 virtual machines
- Normal IPs for Pymtechlabs appear to be in the 45.x.x.x subnet

Lab 2.2: AAD Password Spray Attack

Objectives

- Become familiar with the Azure Active Directory (AAD) logs
- Search logs for anomalous activity
- Visualize a password spray attack

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-12 00:00 UTC** to **2021-04-02 23:30 UTC**.

Load Data

Make sure you have completed Lab 2.1, as this lab requires the data from that lab.

Lab Content

Time Frame

`2021-03-12 00:00:00.000 +00:00` to `2021-04-02 23:30:00.000 +00:00`

Search for Failed Logins

Start in the Discover tab

Set your index to `Azure-*`

To search for possible AAD password spray attacks, we will look for clusters of failed logins.

1. Select log source

Which log source would show user logins (no hint)?

```
insights-logs-signinlogs.json
```

Create a filter to limit your searches to this log source (no hint)

Instead of a filter, you could use a search query. However, it's nicer to have less clutter on the search bar.

2. Narrow down the search

   Per the sign-in logs schema, the field `ResultType` reflects the result of the sign-in operation (success or failure). The field `ResultSignature` contains the error code, if any, for the sign-in operation. In practice, the sign-in logs provide the error code in the `ResultType` field, and the `ResultSignature` seems to always contain the word "None." Azure is such a fast-changing environment that the documentation is not always up to date.

   In SOF-ELK, we mapped the error code to the field `result_type`.

   The most important values are:

   - 0: Success
   - 50074: MFA failed
   - 50126: Failed password
   - 50140: Interrupted due to "Stay signed in?" message

   You can find a complete list at https://for509.com/aad-errorcodes.

   Pymtechlabs uses multiple-factor authentication (MFA). Therefore, when dealing with a password spray attack, we would expect the authentication to fail at the first level.

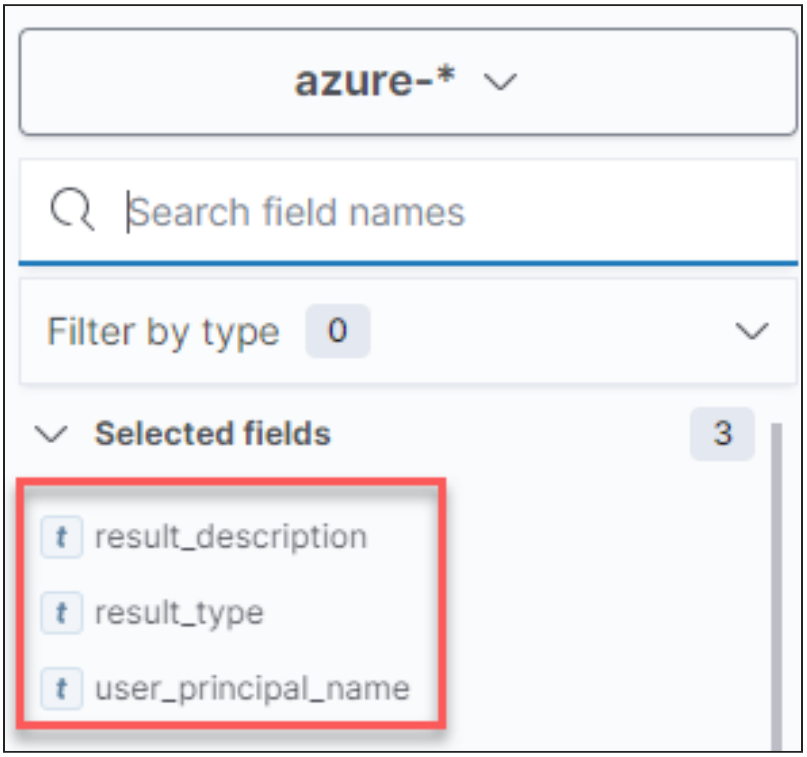
First, select the relevant field. Which ones would you select?

Hint

Select fields that show usernames. Select fields that will contain information about logins (they start with `result_`).

Answer

- `result_type`
- `result_description`
- `user_principal_name`
- Optional: authentication_details (will use up a lot of real estate on your screen)



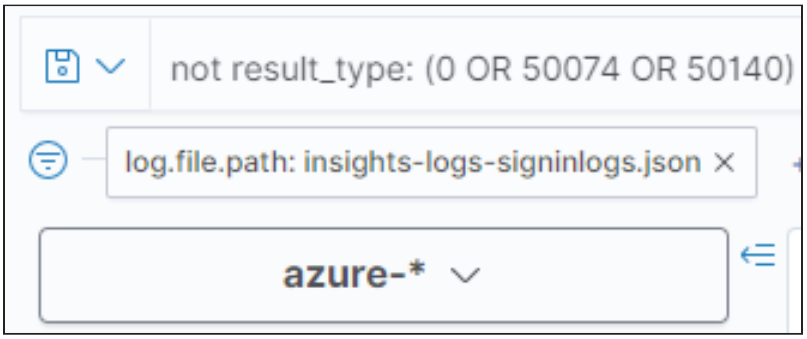
Now that we have all login events, what filter would you write to only see the failed password events?

Hint

Write a negative filter with the field `result_type`.

Answer

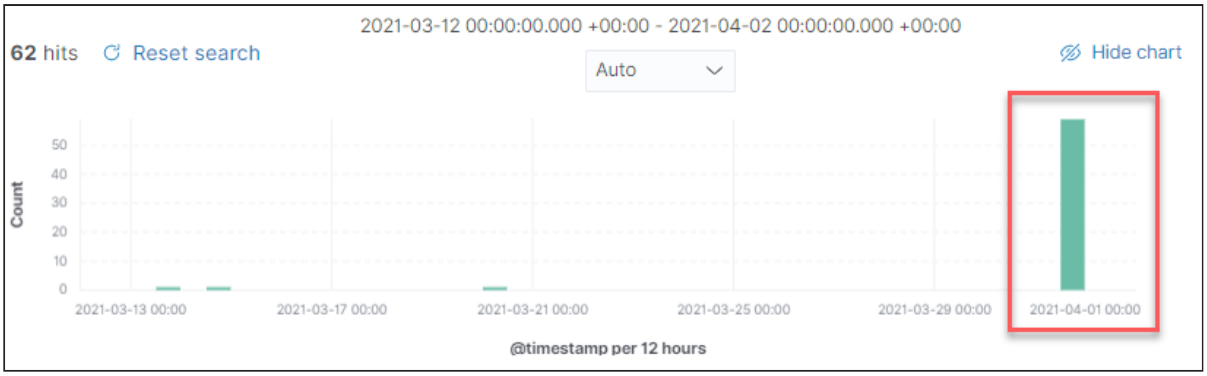
```
not result_type: (0 OR 50074 OR 50140)
```



Why not simply search for `result_type: 50126` (no hint)?

While this would work in this situation, the Microsoft documentation shows over 20 failure codes. As such, it's better to be more specific and only exclude the values we don't want.

We now have a list of all the failed password attempts. In itself, this doesn't prove a password spray attack. However, Kibana gives a graph of the events over time.



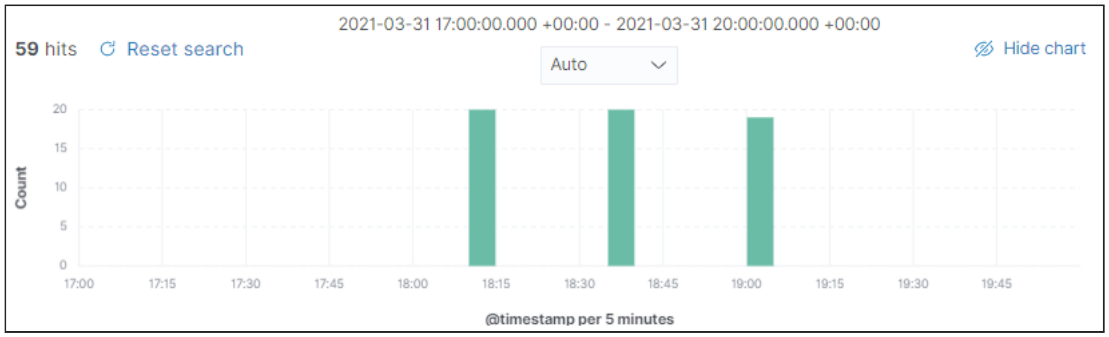
You will notice a few events here and there, followed by a huge spike on 2021-04-01 (time is approximate at this scale; we will need to narrow it down in the next step).

3. Investigate the failed login spike

   We need to narrow down our search by changing the time frame.

What happens to the histogram resolution when you change your time frame to `2021-03-31 17:00:00.000 +00:00` -> `2021-03-31 20:00:00.000 +00:00` (no hint)?

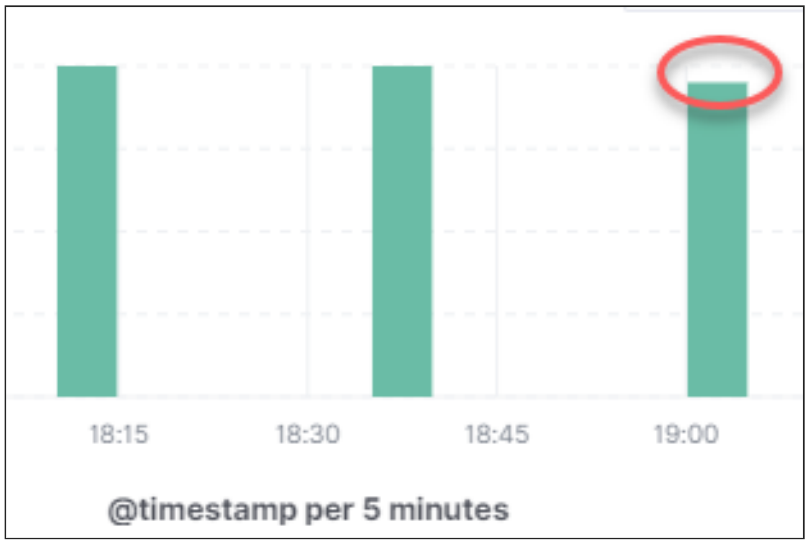
`2021-03-31 17:00:00.000 +00:00` to `2021-03-31 20:00:00.000 +00:00`



Notice that by narrowing down the time frame, the resolution changed to 5 minutes. You could narrow down the time frame some more and go down to a 1 minute resolution but that's not necessary in this case.

With this 5-minute resolution, you can clearly see 3 clusters of failed logins.

Do you notice anything strange on the 3rd cluster (no hint)?

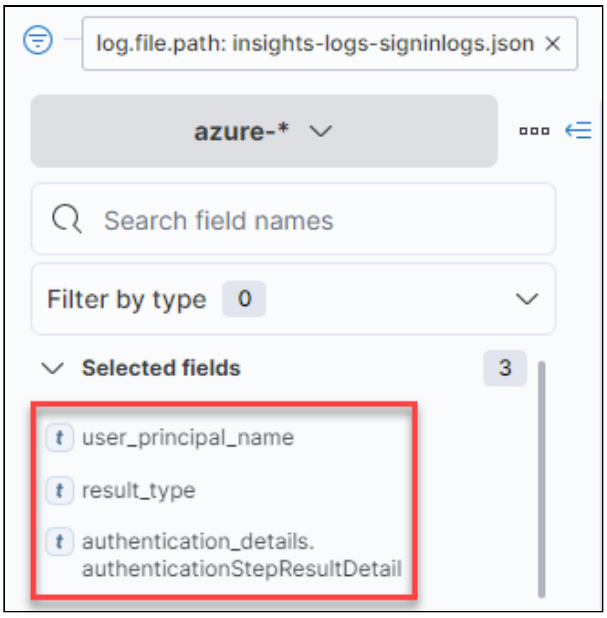


It looks like there is one less failed login on the 3rd cluster. It could be that the bad actor simply tried fewer passwords, or it could be something much more concerning. On to the next step.

4. Any successful logins during this time?

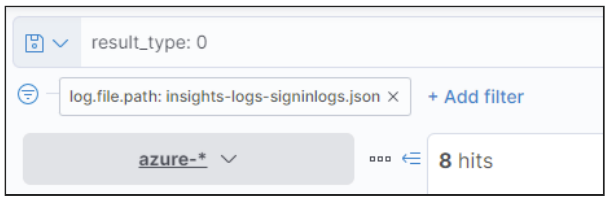
   The last thing we want to see is a successful login in the middle of a password spray attack. Let's investigate this possibility.

   Before you proceed, change your filter to show `user_principal_name`, `result_type`, and `authentication_details.authenticationStepResultDetail`.

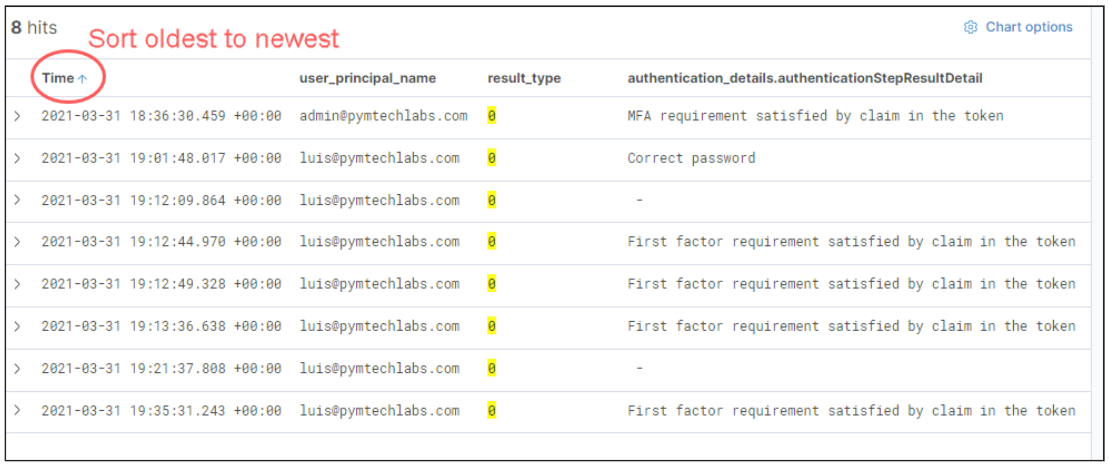

log.file.path: insights-logs-signinlogs.json
azure-*
Search field names
Filter by type 0
Selected fields 3
user_principal_name
result_type
authentication_details.
authenticationStepResultDetail

What filter would you write to test for this possibility (no hint)?

```
result_type: 0
```



Be sure to keep the same narrow time frame and sort the results from oldest to newest



The first event is Hank Pym logging in at the same time as the attack (a coincidence and nothing to do with our investigation). However, the event at `2021-03-31 19:01:48.017` shows Luis successfully logged in with the correct password. This is becoming really concerning. It's not conclusive yet (one more step)!

Bonus question: How is a simple password login possible when MFA is enabled (no hint)?

There are two possibilities:

a. Legacy authentication is enabled on this tenant (example: Activesync), which is something Microsoft is trying hard to obsolete.

b. Luis is part of an AAD Conditional Access rule. It's very typical in corporate environments where some accounts are exempted from MFA. Not a good practice, but sometimes necessary.

5. Examine the event's details

   We need to make sure that the event we found doesn't represent a legitimate login. Before we call the user to confirm, let's check where this login originated from.

Do you see an IP address that could be out of the ordinary (no hint)?

Open the event and check the IP address 85.206.166.82.

This IP address is from Lithuania and is very different from the 45.x.x.x IP addresses we saw in the first lab. It also doesn't belong to Microsoft.

source_ip	85.206.166.82

There is a very strong possibility that we have a serious situation on our hands!

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- `Luis@pymtechlabs.com` has very likely been compromised
- 85.206.166.82 is likely the attacker's IP address
- We must now investigate what actions were taken with this compromised account

Lab 2.3: Tracking Resource Creations

Objectives

- Review the process to narrow the log data
- Review the VM creation process
- Continue the investigation of a potentially compromised account

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-31 17:00 UTC** to **2021-04-02 00:00 UTC**.

Load Data

Make sure you have completed Lab 2.1, as this lab requires the data from that lab.

Lab Content

Time Frame

Since the account was compromised on 2021-03-31, we want to narrow our time frame to minimize the amount of noise.
`2021-03-31 17:00:00.000 +00:00` to `2021-04-02 00:00:00.000 +00:00`

Track suspicious Activity

Start in the Discover tab

Set your index to `azure-*`

We have strong suspicions that `luis@pymtechlabs.com` has been compromised. We should check what operations may have been performed by this account subsequent to the breach.

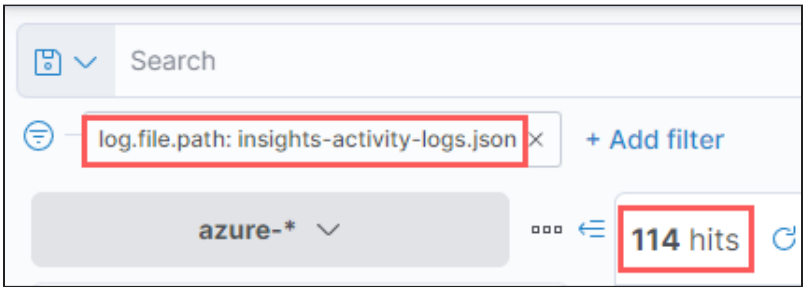
1. Narrow the data set to a specific log

Which field would you filter for?

Hint

Filter for the `insights-activity-logs.json` log

Answer



Your first instinct might be to write a filter such as `user_principal_name: luis@pymtechlabs.com`. However, remember that not every schema includes every type of information. While it would work in this case, it's best to filter on the log type first and then narrow down your search.

Note that we used a filter rather than a search in order to keep our search bar clear for further search terms.

2. Look for suspicious activity

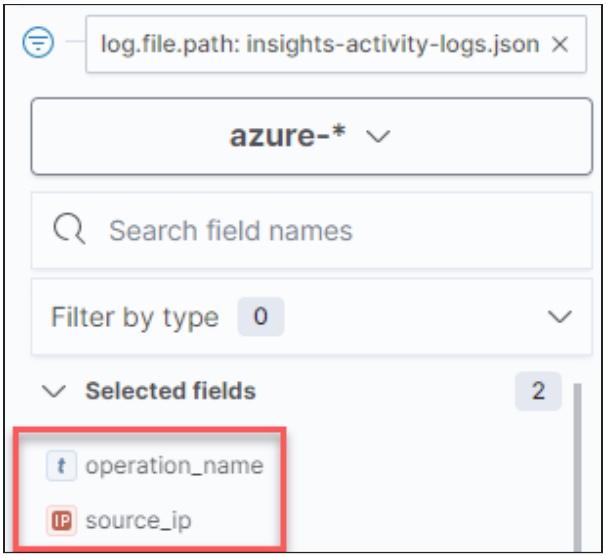
To narrow down our search what field would show you activity in the subscription?

Hint

Use fields that show the name of the operation and the IP address

Answer

`operation_name` and `source_ip`



What does WRITE indicate to you (no hint)?

`New resources are being created`

Time	operation_name	source_ip
2021-04-01 17:19:34.798 +00:00	MICROSOFT.STORAGE/STORAGEACCOUNTS/LISTKEYS/ACTION	85.206.166.82
2021-04-01 17:19:34.643 +00:00	MICROSOFT.STORAGE/STORAGEACCOUNTS/LISTKEYS/ACTION	85.206.166.82
2021-03-31 19:58:51.131 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:57:52.967 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:56:54.051 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:56:51.873 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82

How do we associate this activity with `luis@pymtechlabs.com`?

Hint

Filter based on the UPN and the suspicious IP address

Answer

Add the filter:

```
user_principal_name: luis@pymtechlabs.com AND source_ip: 85.206.166.82
```

Time ▲	operation_name	source_ip
2021-03-31 19:30:36.250 +00:00	MICROSOFT.ADDONS/SUPPORTPROVIDERS/LISTSUPPORTPLANINFO/ACTION	85.206.166.82
2021-03-31 19:30:36.659 +00:00	MICROSOFT.RECOVERYSERVICES/LOCATIONS/BACKUPVALIDATEFEATURES/ACTION	85.206.166.82
2021-03-31 19:30:37.599 +00:00	MICROSOFT.RECOVERYSERVICES/LOCATIONS/BACKUPVALIDATEFEATURES/ACTION	85.206.166.82
2021-03-31 19:30:39.505 +00:00	MICROSOFT.ADDONS/SUPPORTPROVIDERS/LISTSUPPORTPLANINFO/ACTION	85.206.166.82
2021-03-31 19:30:46.055 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/VALIDATE/ACTION	85.206.166.82
2021-03-31 19:30:49.535 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/VALIDATE/ACTION	85.206.166.82
2021-03-31 19:30:55.839 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/VALIDATE/ACTION	85.206.166.82
2021-03-31 19:30:58.279 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/VALIDATE/ACTION	85.206.166.82
2021-03-31 19:30:58.649 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE	85.206.166.82
2021-03-31 19:31:02.574 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE	85.206.166.82
2021-03-31 19:31:03.269 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:07.224 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:12.446 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82
2021-03-31 19:31:16.737 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:16.934 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82

We now see that the person using the `luis@pymtechlabs.com` account is creating a bunch of new resources.

3. Time to dig deeper

What are the names of the virtual machines?

Hint

Filter for the creation of virtual machines.

Answer

```
user_principal_name: luis@pymtechlabs.com AND operation_name: VIRTUALMACHINES AND source_ip:
85.206.166.82
```

The names of the virtual machines are `C1, C2, C3, C4`.

Hint: Add `resource_id` to your list of fields to see the name of the machines.

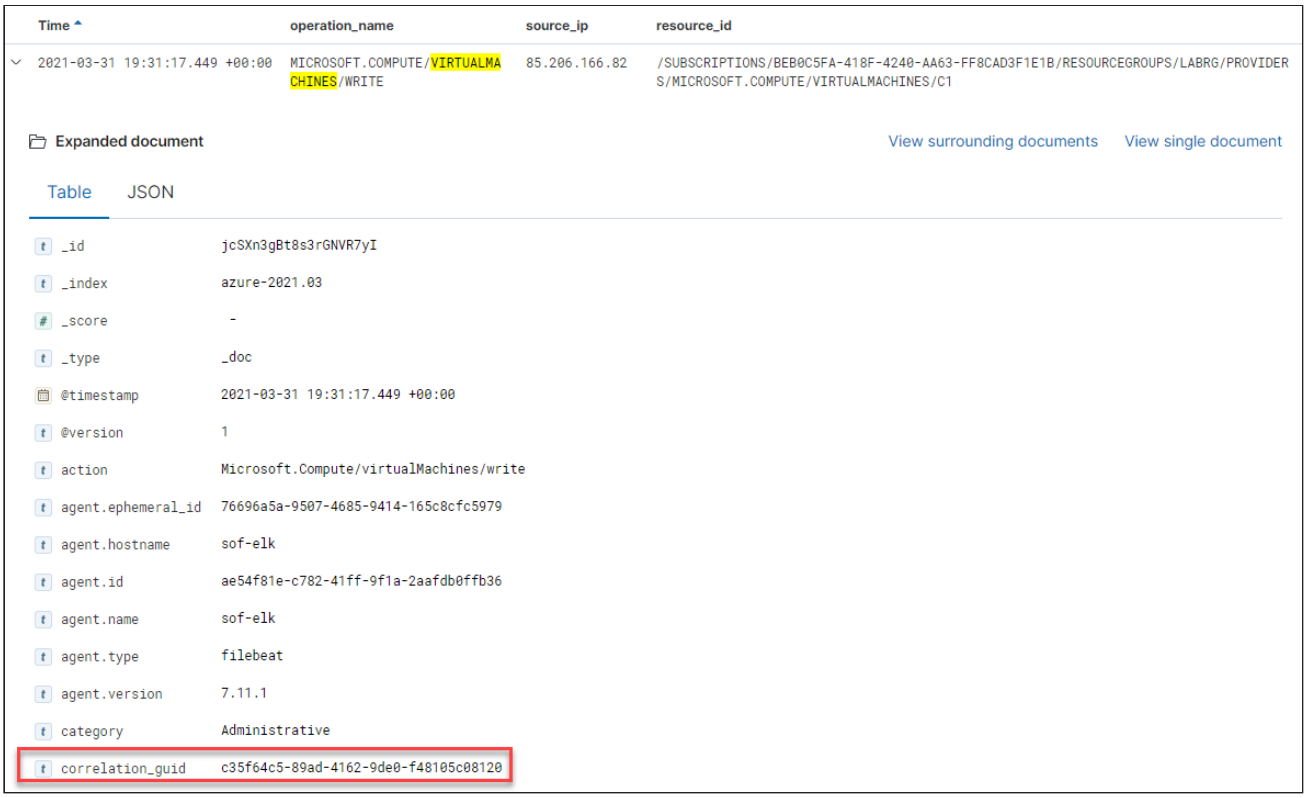
Time	operation_name	resource_id
2021-03-31 19:31:17.449 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1
2021-03-31 19:31:21.290 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1
2021-03-31 19:32:31.399 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1
2021-03-31 19:41:00.915 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C2
2021-03-31 19:41:04.621 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C2
2021-03-31 19:41:26.108 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1
2021-03-31 19:41:51.825 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C2
2021-03-31 19:42:36.280 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C1
2021-03-31 19:44:07.990 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C3
2021-03-31 19:44:11.992 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C3
2021-03-31 19:44:52.579 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C3
2021-03-31 19:46:47.390 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C4
2021-03-31 19:46:50.427 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOFT.COMPUTE/VIRTUALMACHINES/C4

Why would the bad actor create all these VMs? Is there something special about them?

4. VM creation

   When a VM is created, a large number of operations are performed by Azure. Let's look at that sequence of operations for a clue.

   This sequence of events is tracked by the field `correlation_guid`. You can pick any of the records to select one of the numbers. For example, we will select the one for the C1 VM:

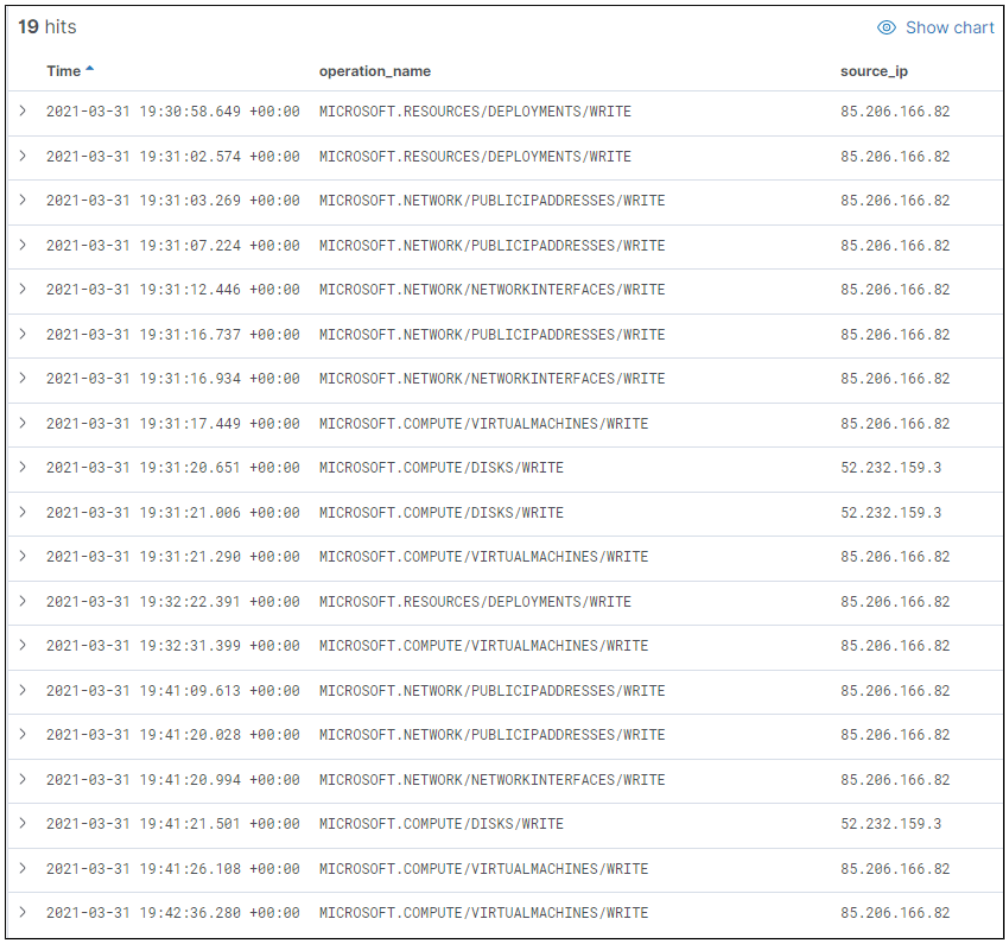


Change your search to that number:

```
correlation_guid : "c35f64c5-89ad-4162-9de0-f48105c08120"
```

What are the steps to building a VM (no hint)?

- Create a deployment with the selected operating system
- Create a network interface
- Assign an IP address (internal and public)
- Create the OS disk
- Create the VM



19 hits — Show chart

Time ▲	operation_name	source_ip
2021-03-31 19:30:58.649 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE	85.206.166.82
2021-03-31 19:31:02.574 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE	85.206.166.82
2021-03-31 19:31:03.269 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:07.224 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:12.446 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82
2021-03-31 19:31:16.737 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:31:16.934 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82
2021-03-31 19:31:17.449 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:31:20.651 +00:00	MICROSOFT.COMPUTE/DISKS/WRITE	52.232.159.3
2021-03-31 19:31:21.006 +00:00	MICROSOFT.COMPUTE/DISKS/WRITE	52.232.159.3
2021-03-31 19:31:21.290 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:32:22.391 +00:00	MICROSOFT.RESOURCES/DEPLOYMENTS/WRITE	85.206.166.82
2021-03-31 19:32:31.399 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:41:09.613 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:41:20.028 +00:00	MICROSOFT.NETWORK/PUBLICIPADDRESSES/WRITE	85.206.166.82
2021-03-31 19:41:20.994 +00:00	MICROSOFT.NETWORK/NETWORKINTERFACES/WRITE	85.206.166.82
2021-03-31 19:41:21.501 +00:00	MICROSOFT.COMPUTE/DISKS/WRITE	52.232.159.3
2021-03-31 19:41:26.108 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82
2021-03-31 19:42:36.280 +00:00	MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE	85.206.166.82

To gain some real estate, remove `operation_name` and `source_ip` from your selected fields. Add the field `response_body`

Time	resource_id	response_body
2021-03-31 19:30:58.649 +00:00	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOF	-
2021-03-31 19:31:02.574 +00:00	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOF	-
2021-03-31 19:31:03.269 +00:00	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOF	-
2021-03-31 19:31:07.224 +00:00	/SUBSCRIPTIONS/BEB0C5FA-418F-4240-AA63-FF8CAD3F1E1B/RESOURCEGROUPS/LABRG/PROVIDERS/MICROSOF	{"name":"C1-ip","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Network/publicIPAddresses/C1-ip","etag":"W/\"050f65e9-d011-423d-a521-f4b48f6ccd0d\"","location":"southcentralus","properties":{"provisioningState":"Updating","resourceGuid":"afde5f41-f54e-4c19-a67d-2cd8b2e8bf88","publicIPAddressVersion":"IPv4","publicIPAllocationMethod":"Dynamic","idleTimeoutInMinutes":4,"ipTags":[]},"type":"Microsoft.Network/publicIPAddresses","sku":{"name":"Basic"}}

From the `response_body` information, what types of VMs were created?

Hint

Look for the property called `vmSize`

Answer

The machines are `Standard_NV4as_v4` types


```
{"name":"C1","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Compute/virtualMachines/C1","type":"Microsoft.Compute/virtualMachines","location":"southcentralus","properties":{"vmId":"4cda4db8-8c3a-48f8-beaa-1bb423f149d1","hardwareProfile":{"vmSize":"Standard_NV4as_v4"},"storageProfile":{"imageReference":{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest","exactVersion":"18.04.202103250"},"osDisk":{"osType":"Linux","createOption":"FromImage","caching":"ReadWrite","managedDisk":{"storageAccountType":"Standard_LRS"},"diskSizeGB":30},"dataDisks":[]},"osProfile":{"computerName":"C1","adminUsername":"luis","linuxConfiguration":{"disablePasswordAuthentication":false,"provisionVMAgent":true,"patchSettin
```


Is there anything special about this type of VM (no hint)?

These are GPU-enabled VMs. This means that the bad actor's action on objective is likely to run crypto mining software.

5. Bonus

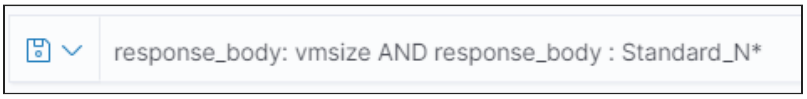
What query could you use to quickly search for GPU-enable VMs?

Hint

Take advantage of the fact that you know the property that stores the type of VM being created

Answer

```
response_body: vmsize AND response_body : Standard_N*
```



Time	response_body
2021-03-31 19:31:21.290 +00:00	{"name":"C1","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Compute/virtualMachines/C1","type":"Microsoft.Compute/virtualMachines","location":"southcentralus","properties":{"vmId":"4cda4db8-8c3a-48f8-beaa-1bb423f149d1","hardwareProfile":{"vmSize":"Standard_NV4as_v4"},"storageProfile":{"imageReference":{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest","exactVersion":"18.04.202103250"},"osDisk":{"osType":"Linux","createOption":"FromImage","caching":"ReadWrite","managedDisk":{"storageAccountType":"Standard_LRS"},"diskSizeGB":30},"dataDisks":[]},"osProfile":{"computerName":"C1","adminUsername":"luis","linuxConfiguration":{"disablePasswordAuthentication":false,"provisionVMAgent":true,"patchSettings":{"patchMode":"ImageDefault"}},"secrets":[],"allowExtensionOperations":true,"requireGuestProvisionSignal":true},"networkProfil
2021-03-31 19:41:04.621 +00:00	{"name":"C2","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Compute/virtualMachines/C2","type":"Microsoft.Compute/virtualMachines","location":"southcentralus","properties":{"vmId":"70aacd5c-27bf-48f6-b81b-1df26bdb6d98","hardwareProfile":{"vmSize":"Standard_NV4as_v4"},"storageProfile":{"imageReference":{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest","exactVersion":"18.04.202103250"},"osDisk":{"osType":"Linux","createOption":"FromImage","caching":"ReadWrite","managedDisk":{"storageAccountType":"Standard_LRS"},"diskSizeGB":30},"dataDisks":[]},"osProfile":{"computerName":"C2","adminUsername":"Luis","linuxConfiguration":{"disablePasswordAuthentication":false,"provisionVMAgent":true,"patchSettings":{"patchMode":"ImageDefault"}},"secrets":[],"allowExtensionOperations":true,"requireGuestProvisionSignal":true},"networkProfil
2021-03-31 19:44:11.992 +00:00	{"name":"C3","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Compute/virtualMachines/C3","type":"Microsoft.Compute/virtualMachines","location":"southcentralus","properties":{"vmId":"d3612af5-2457-420b-9930-d3445923fc90","hardwareProfile":{"vmSize":"Standard_NV4as_v4"},"storageProfile":{"imageReference":{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest","exactVersion":"18.04.202103250"},"osDisk":{"osType":"Linux","createOption":"FromImage","caching":"ReadWrite","managedDisk":{"storageAccountType":"Standard_LRS"},"diskSizeGB":30},"dataDisks":[]},"osProfile":{"computerName":"C3","adminUsername":"luis","linuxConfiguration":{"disablePasswordAuthentication":false,"provisionVMAgent":true,"patchSettings":{"patchMode":"ImageDefault"}},"secrets":[],"allowExtensionOperations":true,"requireGuestProvisionSignal":true},"networkProfil
2021-03-31 19:46:50.427 +00:00	{"name":"C4","id":"/subscriptions/beb0c5fa-418f-4240-aa63-ff8cad3f1e1b/resourceGroups/labRG/providers/Microsoft.Compute/virtualMachines/C4","type":"Microsoft.Compute/virtualMachines","location":"southcentralus","properties":{"vmId":"f1956003-b462-4422-b165-79ee8df48094","hardwareProfile":{"vmSize":"Standard_NV4as_v4"},"storageProfile":{"imageReference":{"publisher":"Canonical","offer":"UbuntuServer","sku":"18.04-LTS","version":"latest","exactVersion":"18.04.202103250"},"osDisk":{"osType":"Linux","createOption":"FromImage","caching":"ReadWrite","managedDisk":{"storageAccountType":"Standard_LRS"},"diskSizeGB":30},"dataDisks":[]},"osProfile":{"computerName":"C4","adminUsername":"luis","linuxConfiguration":{"disablePasswordAuthentication":false,"provisionVMAgent":true,"patchSettings":{"patchMode":"ImageDefault"}},"secrets":[],"allowExtensionOperations":true,"requireGuestProvisionSignal":true},"networkProfil

Potential Sentinel query

While Microsoft Sentinel is outside the scope of this class, here is a potential query to search for GPU-enabled VMs:

```
AzureActivity
| where OperationNameValue == "MICROSOFT.COMPUTE/VIRTUALMACHINES/WRITE"
| where Properties has "created" and Properties has "vmsize"
| extend responseBody = todynamic(tostring(Properties_d.responseBody))
| where responseBody.properties.provisioningState == "Creating"
| where responseBody.properties.hardwareProfile.vmSize contains_cs "_N"
| project TimeGenerated,vmSize=responseBody.properties.hardwareProfile.vmSize, Caller,
CallerIpAddress, ResourceGroup, CorrelationId, SubscriptionId
```

Shoutout to Romain Pracca for this query.

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

We showed you a step-by-step approach so that you can use the same logic for other investigations. However, if you know what you are looking for, you can quickly search for it. Just be sure to consider the context of the results, as it might be perfectly normal for your users to create GPU-enabled VMs (in this example).

- `Luis@pymtechlabs.com` is definitively compromised
- The bad actors have created VMs for the purpose of crypto mining
- Further investigation is required to make sure the breach is contained

Lab 2.4: Detecting Data Exfiltration

Objectives

- Obtain storage account access keys
- Discover a log for storage account data transfer
- Compare this to a traditional FTP transfer

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-31 17:00 UTC** to **2021-04-02 00:00 UTC**.

Load Data

The Azure Logs have been exported and saved in `/home/elk_user/lab-2.4_source_evidence.zip`

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract the files from the ZIP file above with the following commands:

   Command lines

   ```
   cd ~
   unzip lab-2.4_source_evidence.zip
   ```

   Wait for the archive to be extracted.

Command lines

```
cp ~/lab-2.4_source_evidence/insights-logs-networksecuritygroupflowevent.csv /logstash/nfarch
```

Command lines

```
cp ~/lab-2.4_source_evidence/insights-logs-storageread.json /logstash/azure
```

Warning

Do not run these commands more than once!

3. Wait 3-5 minutes for the data to be processed (netflow data takes a while to load)
4. Verify that you have 6,217 documents loaded in the `azure` index and 188,735 documents loaded in the `netflow` index:

Command lines

```
sof-elk_clear.py -i list
```

Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- azure (6,217 documents)
- netflow (188,735 documents)
- office365 (5,462 documents)
```

Lab Content

Time Frame

`2021-03-31 17:00:00.000 +00:00` to `2021-04-02 00:00:00.000 +00:00`

Search for Data Exfil

Start in the Discover tab

Set your index to `azure-*`

We know that our bad actor has created new VMs for the purpose of mining cryptocurrency. What else could they be up to? A typical action on objective is to exfiltrate data. Let's look for any activity that could indicate such actions.

1. A storage account When thinking about exfiltrating data, we need to consider where the data may be stored. There are two obvious locations:
2. A disk attached to a VM

The first option is the most interesting from a cloud perspective.

To access a storage account, it's necessary to have access keys. Let's start there.

Clear any filters that you may have from prior labs.

1. Storage account keys

Did someone retrieve the account keys?

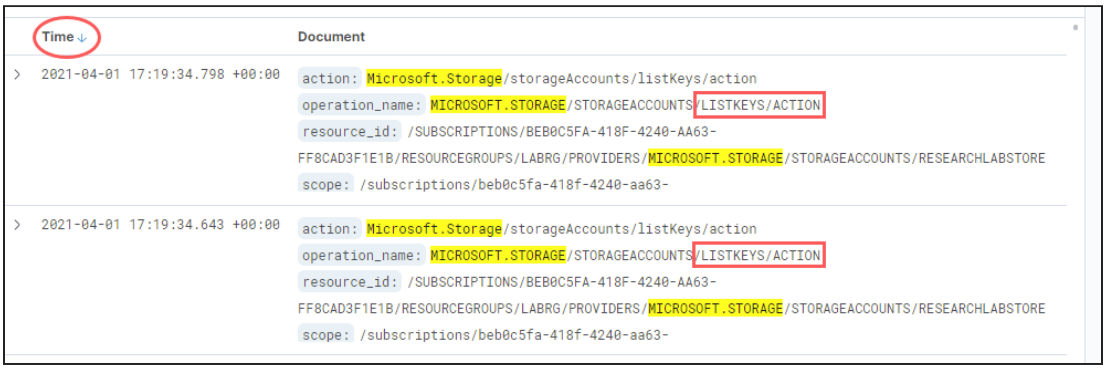
Hint

Start by filtering the log for storage related entries.

Answer

```
Microsoft.Storage
```

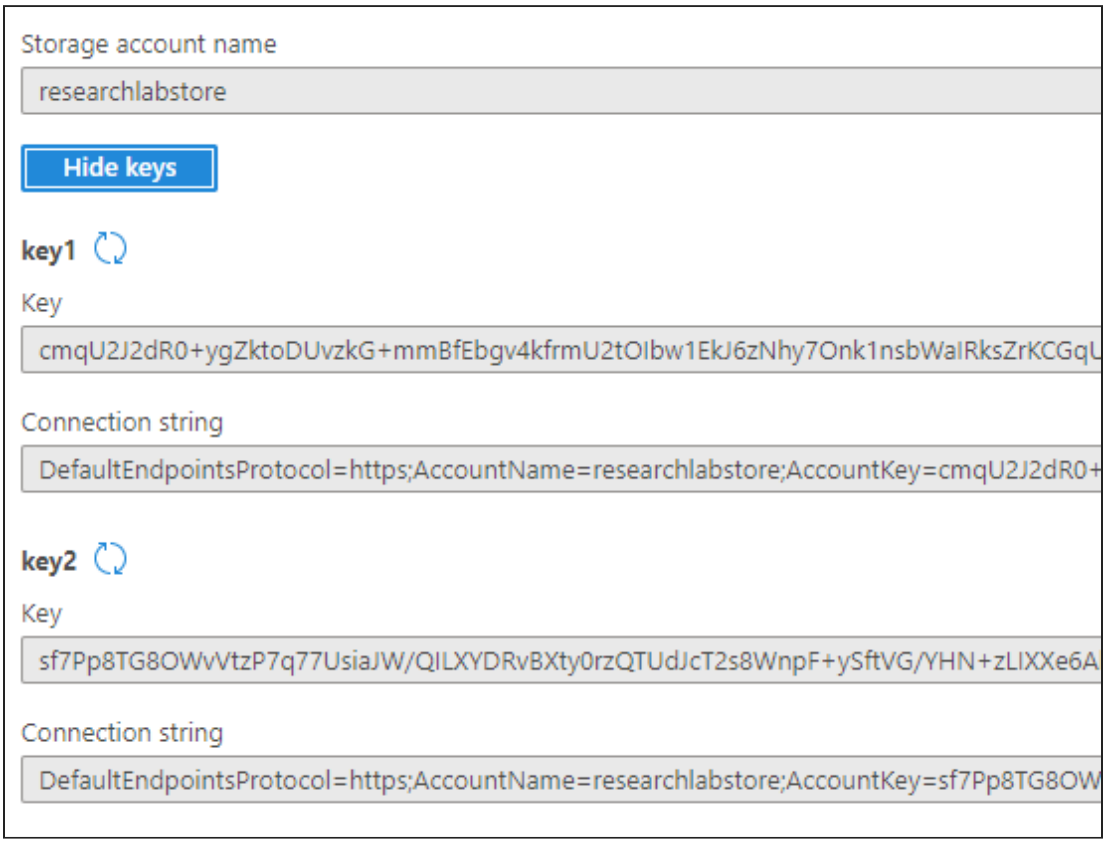
Enter `Microsoft.Storage` as a search term:



Look for events with an `operation_name` of `Microsoft.Storage/storageAccounts/listKeys/action`.

The operation `LISTKEYS/ACTION` indicates that someone looked at the secret keys for the storage account.

In case you are not familiar with the storage account secret keys, this is what they look like in the console:



Yes, the keys are intentionally cut off to prevent anyone hacking my storage account :)

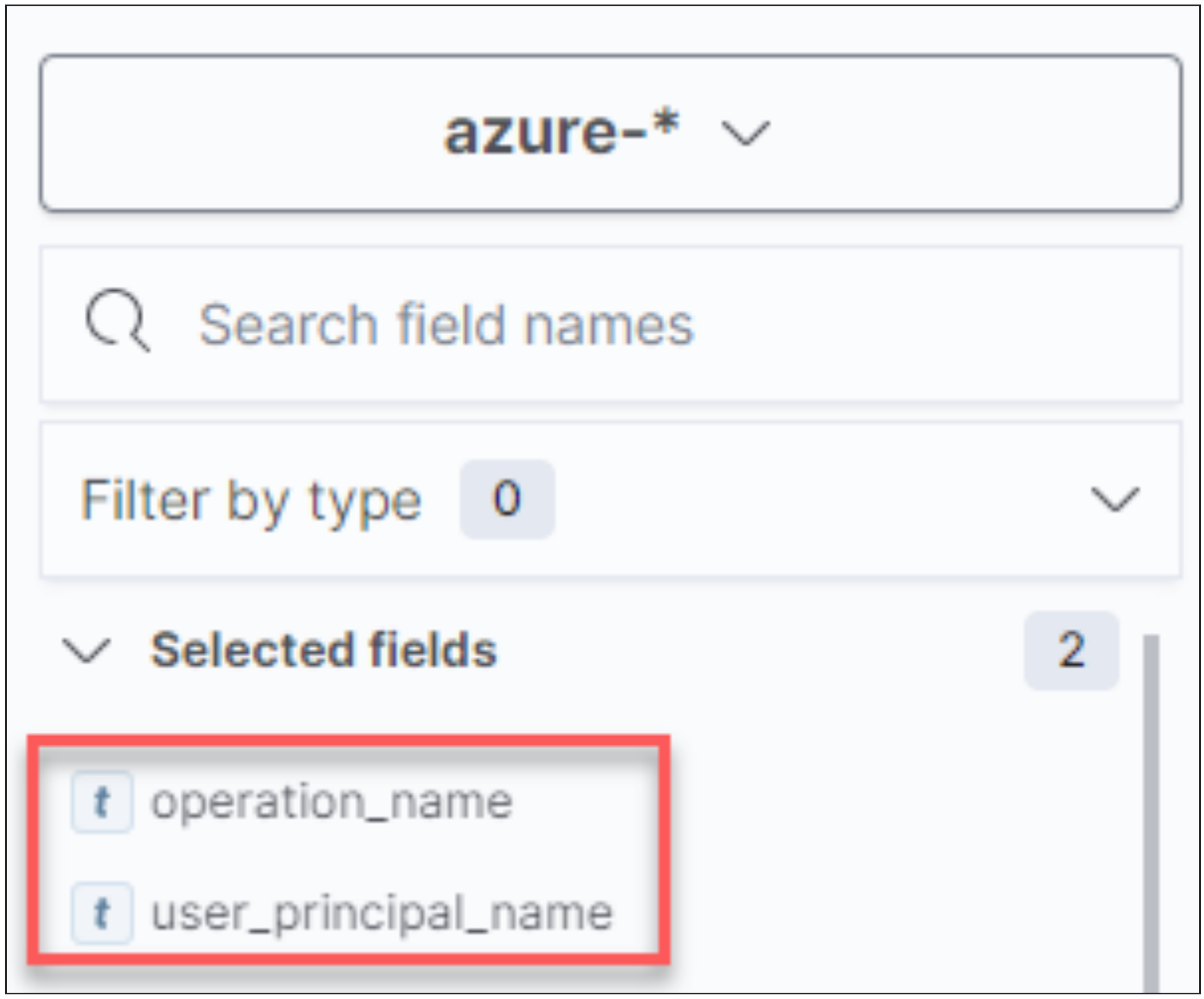
Which storage account was accessed and who accessed it?

> **Hint**
>
> Select two fields which are most likely to contain the information. Focus on entries from the `Microsoft.Storage` provider.

Answer

`researchlabstore` was accessed by `Luis@pymtechlabs.com`

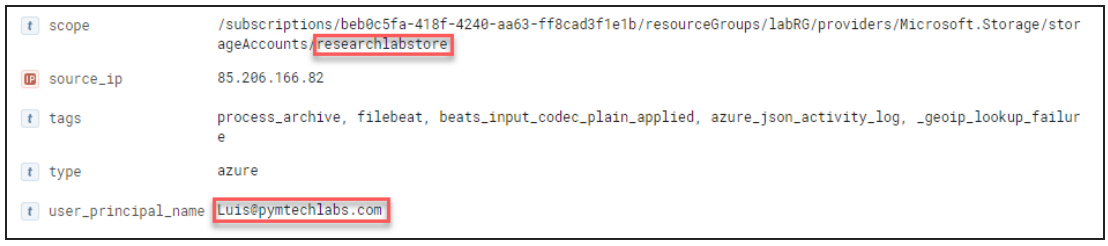
Select the fields `operation_name` and `user_principal_name`.



You will now see that `Luis@pymtechlabs.com` is the one who performed the LISTKEYS operation.

	Time	operation_name	user_principal_name	result_type
>	2021-04-01 19:19:34.798 +02:00	MICROSOFT.STORAGE/STORAGEACCOUNTS/LISTKEYS/ACTION	Luis@pymtechlabs.com	Success
>	2021-04-01 19:19:34.643 +02:00	MICROSOFT.STORAGE/STORAGEACCOUNTS/LISTKEYS/ACTION	Luis@pymtechlabs.com	Start

Notice that one record shows that the operation started and one shows that the operation succeeded. If you open anyone of the two records, you will see that the fields `scope` and `resource_id` show that the target is a storage account called `researchlabstore`.



Why was Luis able to access the keys?

Hint

What permission would Luis need to be able to access the keys?

Answer

Luis would need to have access to the storage account to see the keys. This access could be granted if Luis has the appropriate permission at the subscription or resource group level. In this case Luis has the contributor role at the subscription level which is a very typical role to assign to users.

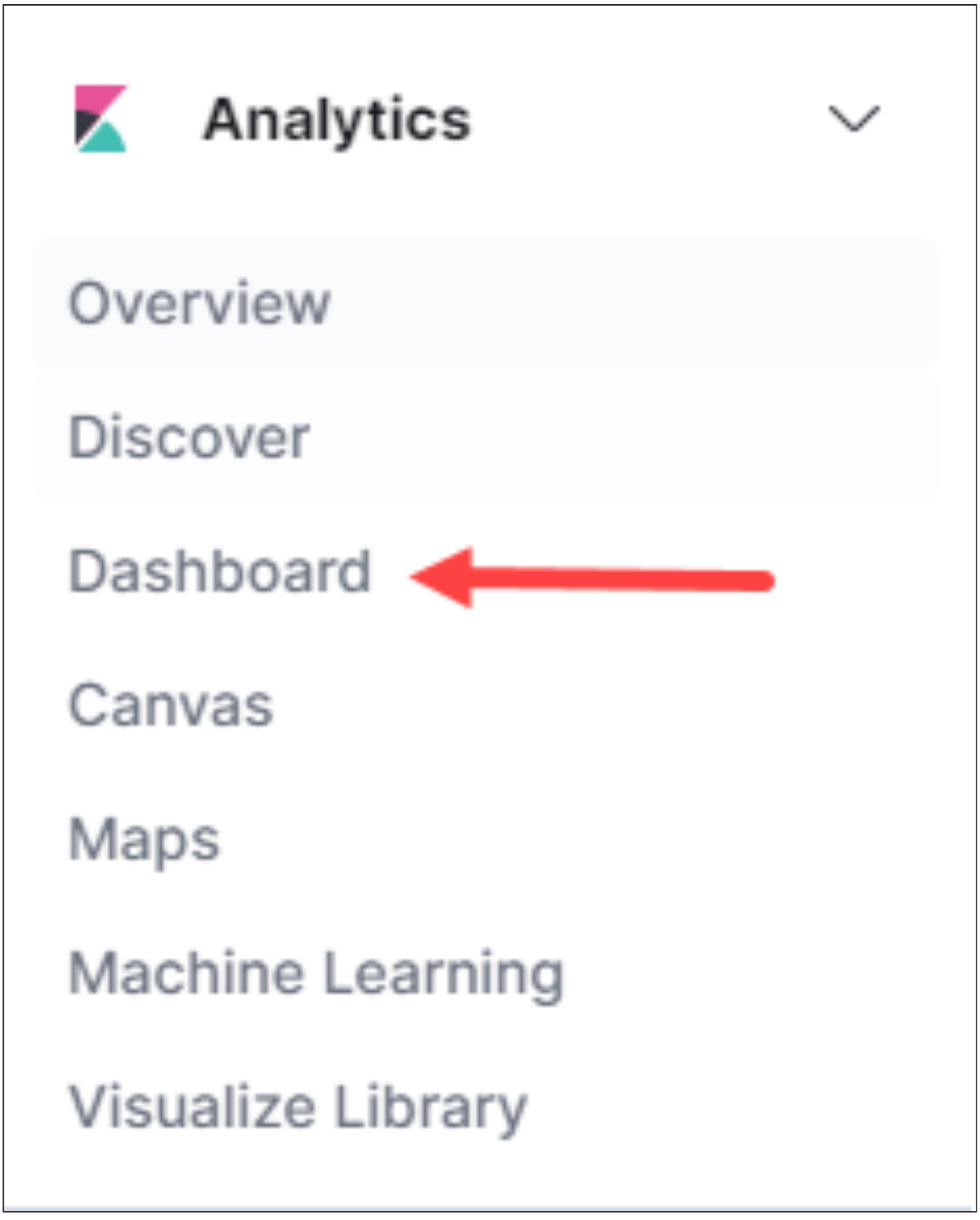
2. Transferring data

   We know that Luis accessed the storage account key. Shouldn't there be some log entries?

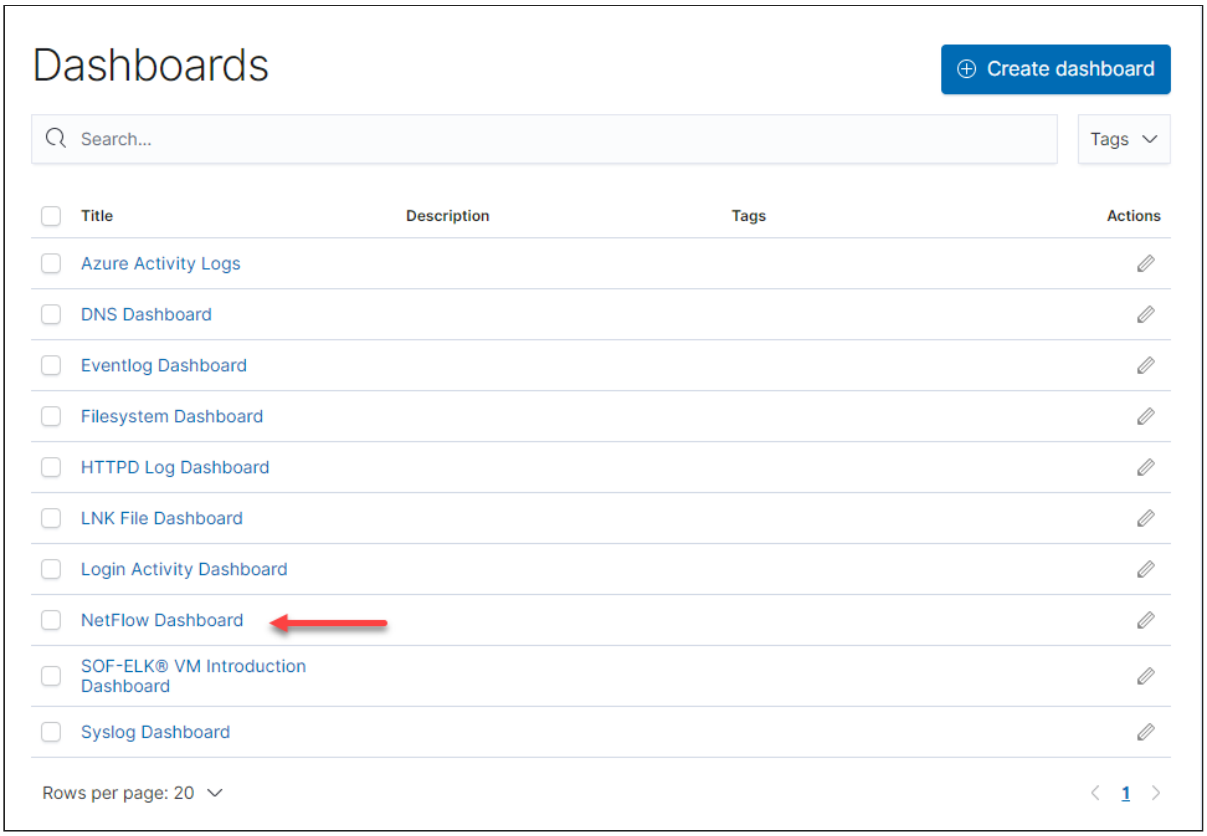
   Remember that we are only looking at the activity log and the sign-in logs in the `azure-*` index.

   It's time to look at a different index. If data is going to be transferred out, it would be logical to expect associated network traffic.

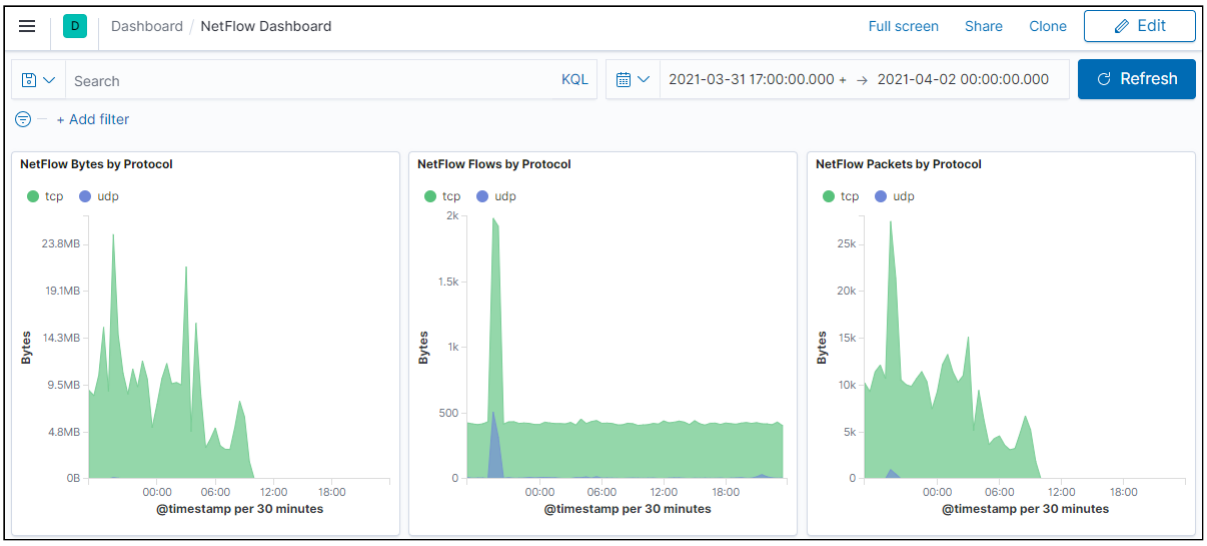
   Clear your search filter and switch to the Dashboard tab:



Select the NetFlow dashboard:



During our time frame, we didn't have much data transferred. That being said, if there was data exfiltration, we don't know the size of the files. So far, this is inconclusive.



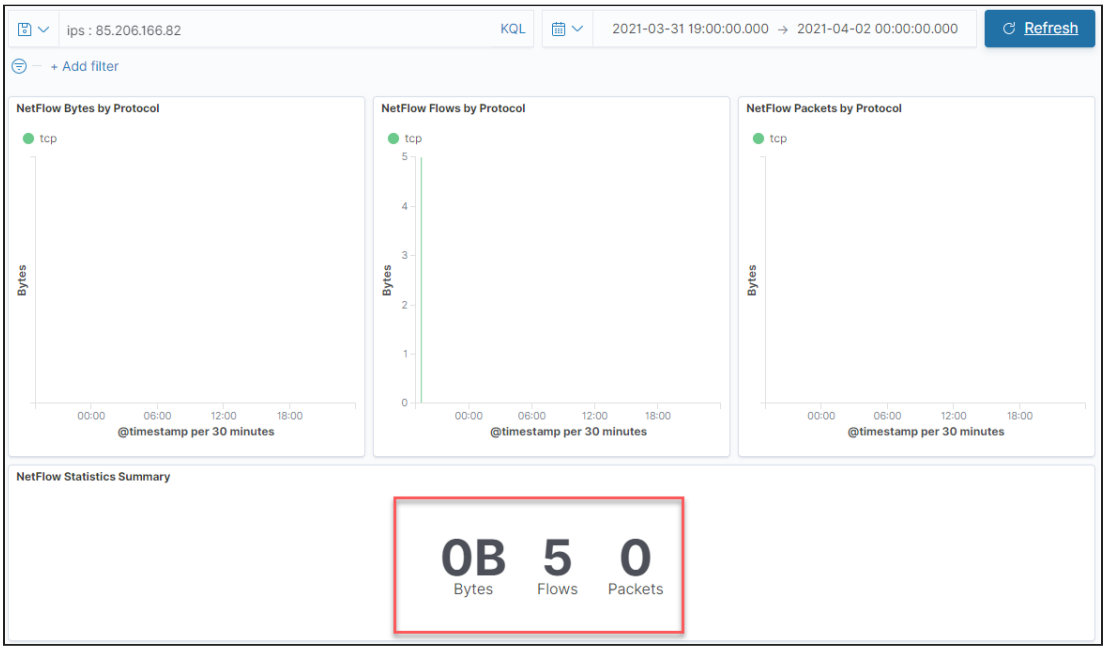
Write a filter to narrow down the dashboard information to our suspect's IP address

Hint

Our suspect's IP address is 85.206.166.82

Answer

```
ips: 85.206.166.82
```



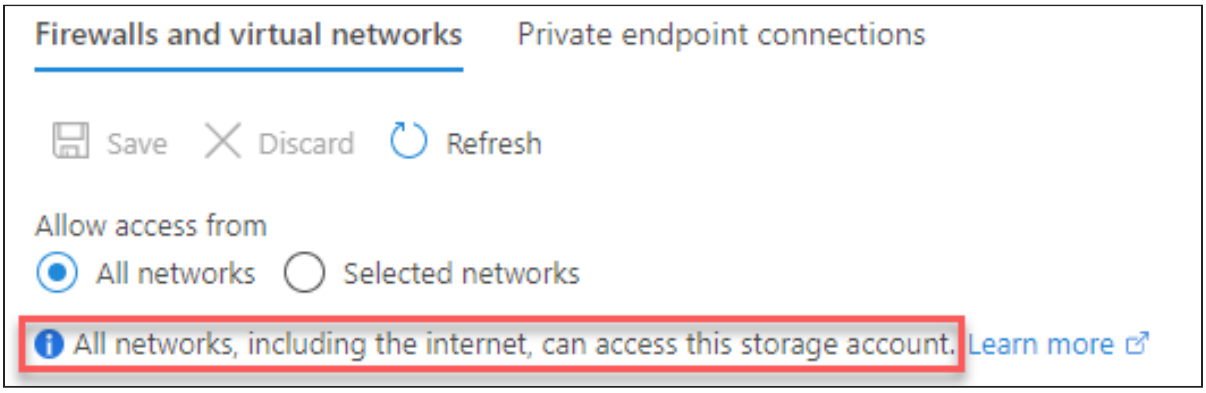
Well, that's a bit unexpected: no data exfiltration. The 5 flows are from Luis testing the connection to the crypto-mining machines by doing a quick SSH.

	Time ↓	destination_ip	destination_port
>	2021-03-31 21:52:10. ⊕ ⊖	10.1.0.12	22
>	2021-03-31 21:51:39.000Z	10.1.0.14	22
>	2021-03-31 21:50:58.000Z	10.1.0.13	22
>	2021-03-31 21:50:06.000Z	10.1.0.12	22
>	2021-03-31 21:48:26.000Z	10.1.0.11	22

Analysis

Let's step back for a second and think about what a storage account represents. It's storage in the Azure cloud that has its own set of permissions. It's not controlled by the network security group (NSG). It doesn't even have an IP address. As such, we won't find any data transfer logged in the netflow data.

If you look at the default storage account configuration in the Azure console (Network sub-menu), you will see that storage accounts are open to "All networks, including the internet" by default:



3. Where are the logs?

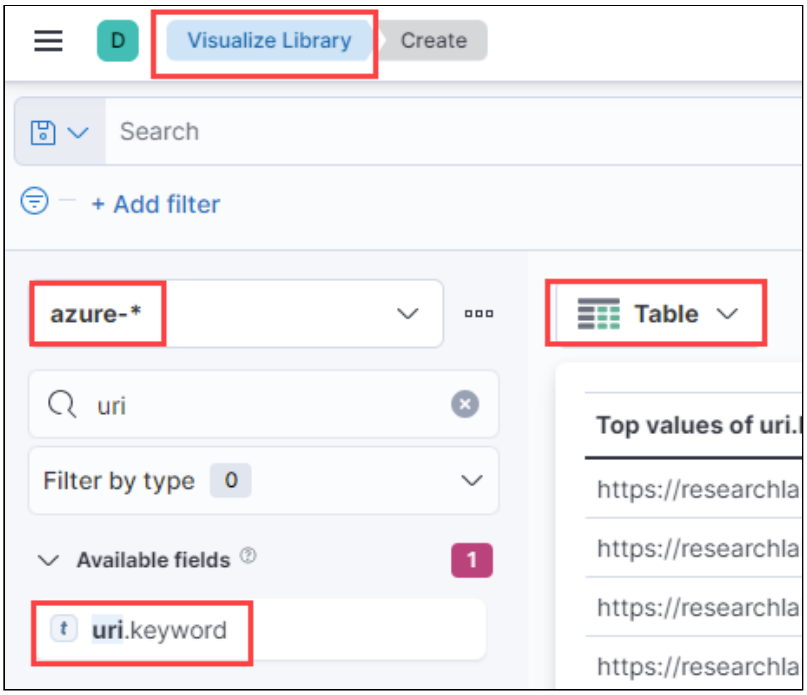
   So, is there really no log of data being transferred from a storage account?

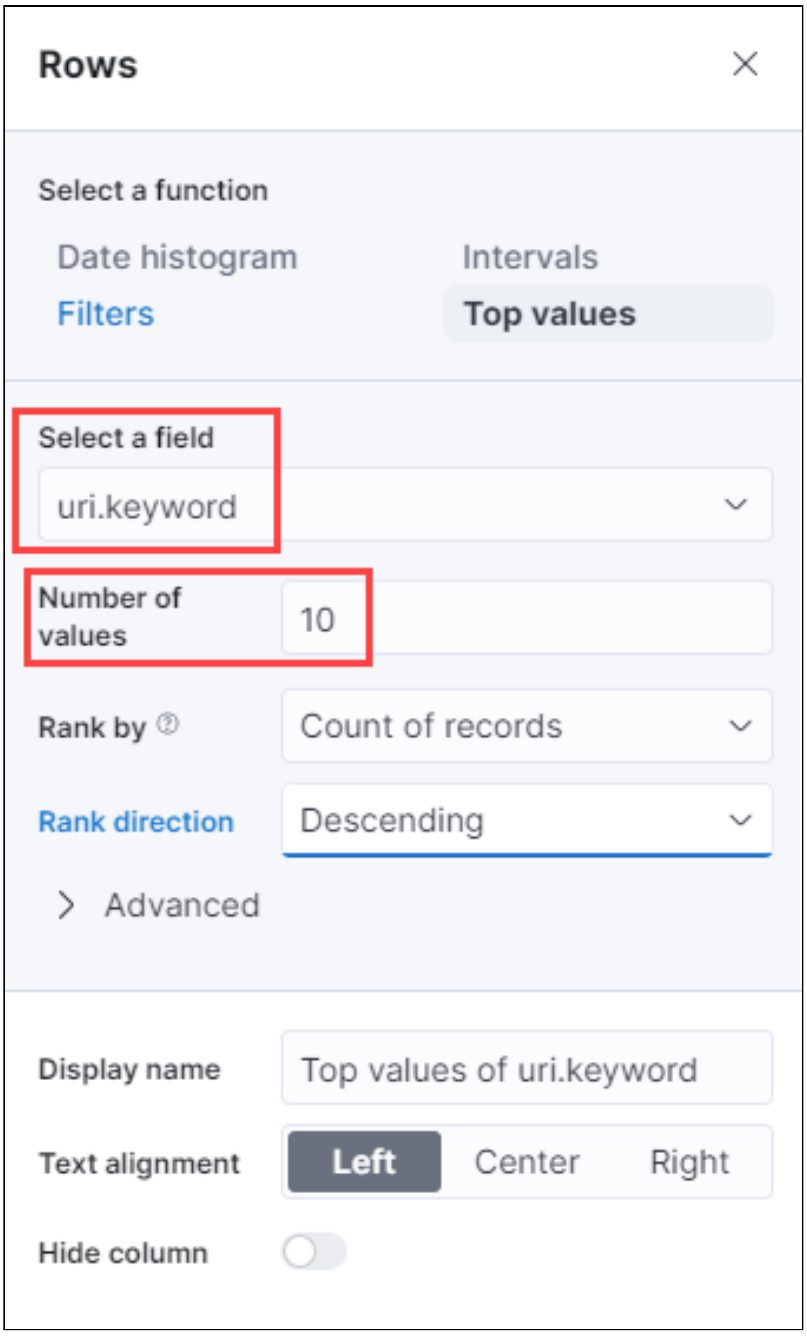
   Actually, there is, but they aren't enabled by default. There are three logs:

   - storageread
   - storagewrite
   - storagedelete

   We imported the storageread log to the `azure-*` index.

   Go back to the `Visualize Library` tab, select `Create visualization`, and then `Lens`, and then make sure your index is set to `azure-*`. Change your graph type to `Table`. Drag and drop the field called `uri.keyword`. Expand the number of rows.




Rows
Select a function
Date histogram
Intervals
Filters
Top values
Select a field
uri.keyword
Number of values
10
Rank by
Count of records
Rank direction
Descending
Advanced
Display name
Top values of uri.keyword
Text alignment
Left
Center
Right
Hide column

What are the names of the files that were exfiltrated?

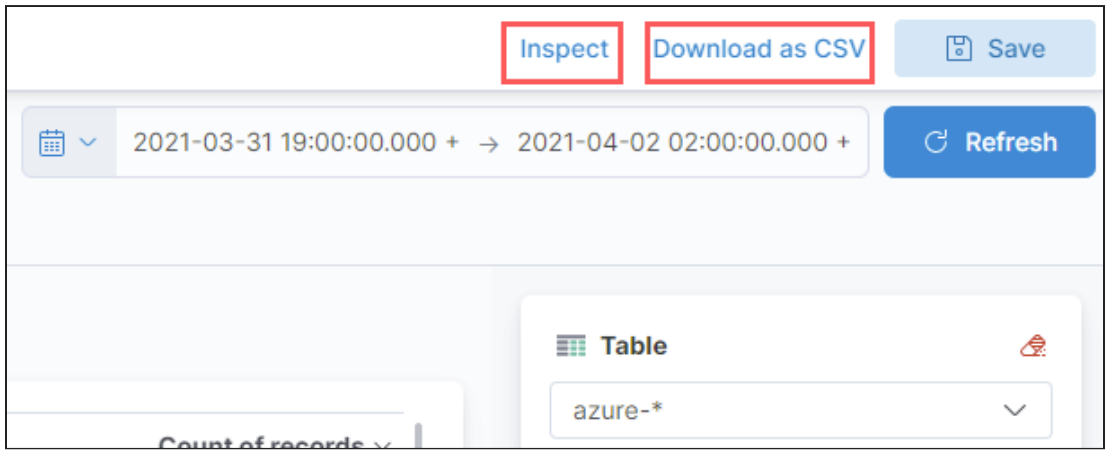
Hint

Looks for the file name at the end of the URL that ends in .7z.

Unfortunately, Kibana is no longer wrapping text fields in the table view. There are two options:

a. Use `inspect` to look at the full field name (works well for small datasets).

b. Use `Download as CSV` and analyze in Excel (or similar application).



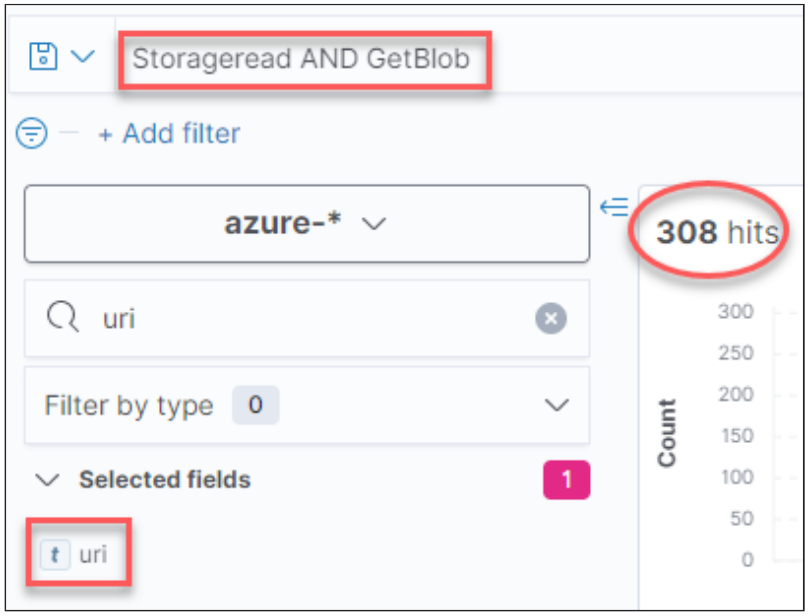
Answer

- Truth_serum.7z
- Pym_particle.7z
- Quantum_parts_for_sale.7z
- Quantum_tunnel.7z
- Ant_control_helmet.7z
- Quantum_energy.7z
- SHIELD.7z
- Sokovia_Accords.7z
- Quantum_realm.7z
- Avengers_Roster.7z

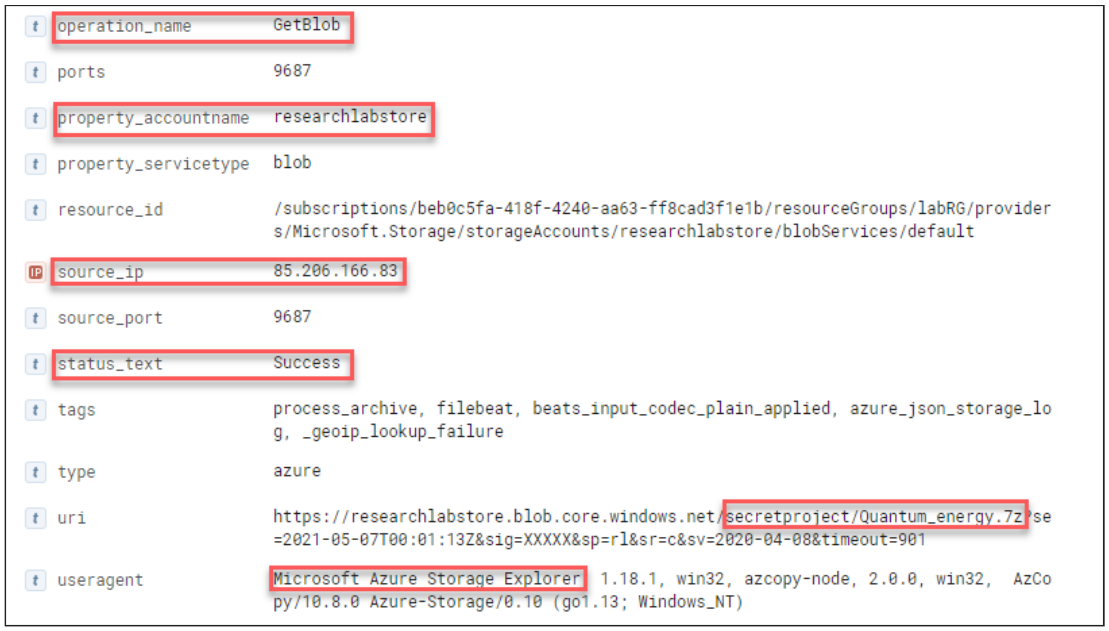
What does `secretproject` represent in the URI (no hint)?

`secretproject` is the name of the container within the `researchlabstore` storage account.

You can verify that by going back to the `Discover` tab and looking at the details of one of the records. Use `Storageread AND GetBlob` for your search parameter and select the field `uri`.



Time	uri
2021-04-01 00:04:36.082 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_energy.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:31.464 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:20.878 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:17.998 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:17.538 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Avengers_Roster.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:16.827 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:16.256 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901
2021-04-01 00:04:16.232 +00:00	https://researchlabstore.blob.core.windows.net/secretproject/Quantum_realm.7z?se=2021-05-07T00:01:13Z&sig=XXXXX&sp=rl&sr=c&sv=2020-04-08&timeout=901



The detailed record matches everything we would have expected:

- The `operation_name` is `GetBlob` which indicates we downloaded a blob
- The `property_accountname` is `researchlabstore` which is the storage account Luis retreived the keys from
- The `source_ip` is `85.206.166.82` which is our suspicious IP address
- The `status_text` is `success` which indicates a successful data exfiltration
- The `uri` shows the `secretproject` container as well as the filename
- Bonus: We have a `useragent` string that shows us that the bad actor used Microsoft Azure Storage Explorer (which is the tool we discussed in class)

Warning

Make sure to clear out any filters you may have applied for this lab.

Bonus

In the introduction, we mentioned that another, more traditional, way to exfiltrate data is using a disk on a VM. Let's examine the logs you can expect if the exfiltration is performed via FTP. There are many paths you can take to search for this information: Dashboard, Visualize Library, and Discover. It's up to you to explore each one.

The key elements of your search should be:

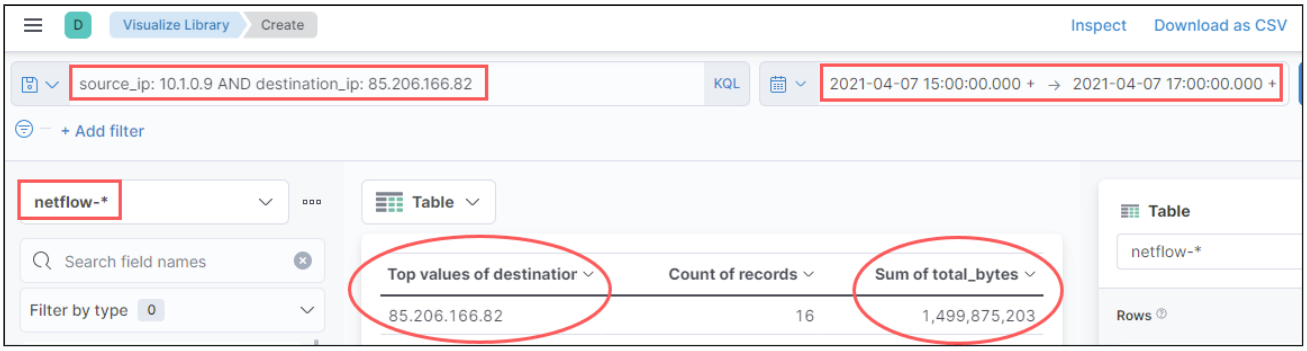
- Time frame: `2021-04-07 13:00:00.000 +00:00` to `2021-04-07 15:00:00.000 +00:00`
- FTP server: `10.1.0.9`
- FTP client: `85.206.166.82`
- Index: `netflow-*`

Be sure to clear any existing filters and notice the new time frame

Suggested search query:

```
source_ip: 10.1.0.9 AND destination_ip: 85.206.166.82
```

As an example, here is a table (from the `Visualize` tab) showing the data transfer:



Key Takeaways

- Logging of storage account transfers is turned off by default.
- Logs (Read, Write, or Delete) should be enabled depending on your threat profile.
- Netflow data won't help for cloud storage account incidents.

Lab 3.1: Reviewing CloudTrail Logs

Objectives

- Learn how to analyze IAM logs in CloudTrail
- Learn how to determine where an access originated from
- Learn how to determine how the access occurred

Background

Welcome to Pymtech's AWS Group! We need you to take a look at our CloudTrail IAM logs to figure out what's been going on. We've noticed some strange activity in our AWS tenant, and we are hoping you can figure it out. We've exported the logs out of CloudTrail and into SOF-ELK® under the AWS index.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-01 23:18 UTC** to **2021-05-01 23:18 UTC**.

Load Data

In lab 0 you loaded a GeoIP database, which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data, and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the AWS parsers, and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

The AWS CloudTrail logs for this lab have already been exported from our S3 Bucket. We've tried to make the loading process as close to what you will do in the real world as possible. As such, with how AWS stores data, there is a two step load process.

The CloudTrail Logs for lab-3.1 have been saved into `/home/elk_user/lab-3.1_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract the all the files from the preceding ZIP file with the following commands:

Command lines

```
cd ~
unzip -q lab-3.1_source_evidence.zip
```

3. Run the AWS CloudTrail ingestion script

Command lines

```
cd ~
aws-cloudtrail2sof-elk.py -r ./lab-3.1_source_evidence -w /logstash/aws/lab-3.1.json
```

Warning

Do not run this command more than once!

4. Wait 3-4 minutes for the data to be processed. Once it is complete, the data will be available in the `aws-*` index.
5. Verify that you have 213,775 documents loaded in the `aws` index:

Command lines

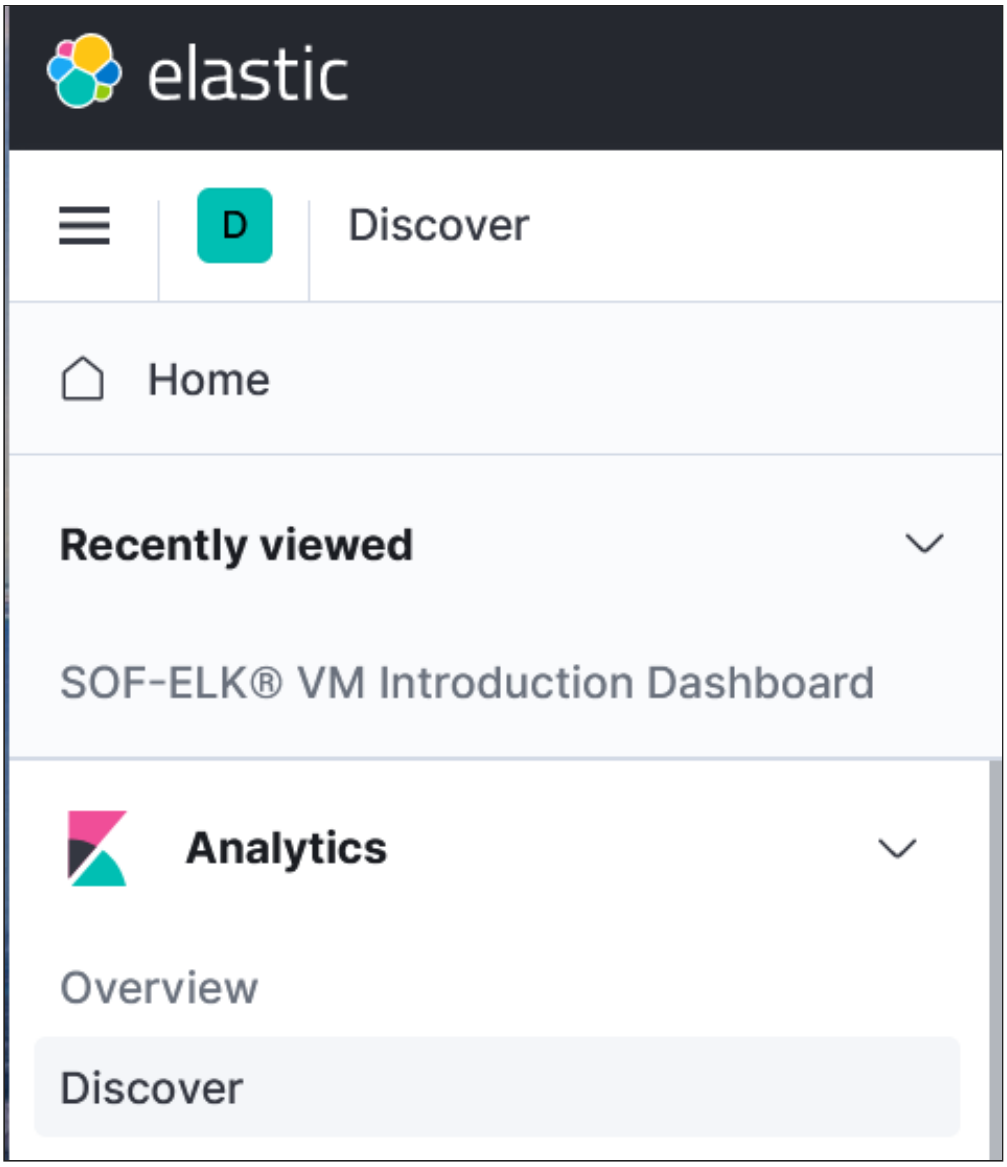
```
sof-elk_clear.py -i list
```

Expected results

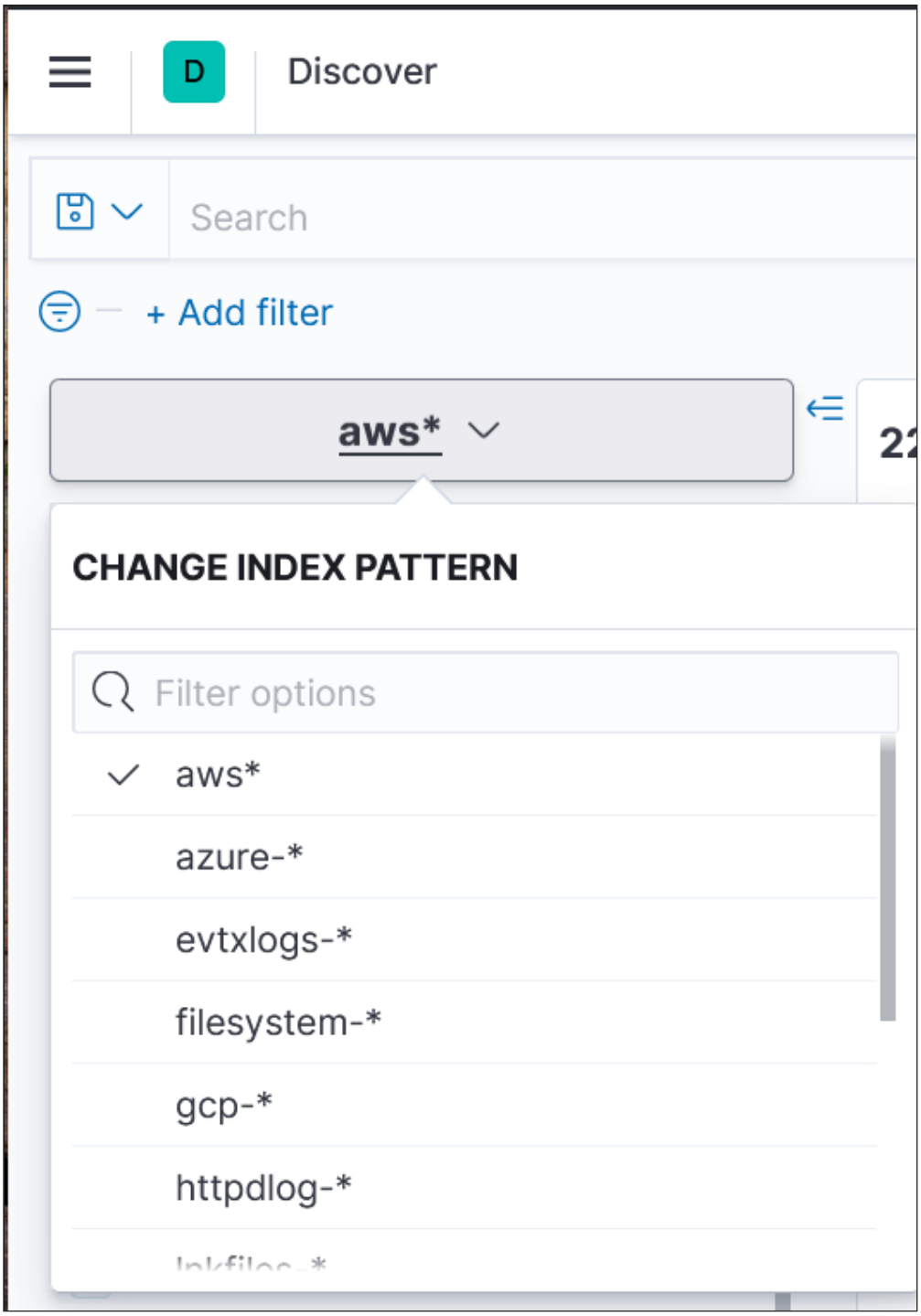
```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (213,775 documents)
- azure (6,217 documents)
- netflow (188,735 documents)
- office365 (5,462 documents)
```

Lab Content

Once you've connected to Kibana you're going to want to click on the menu on the upper left and select 'Discover'

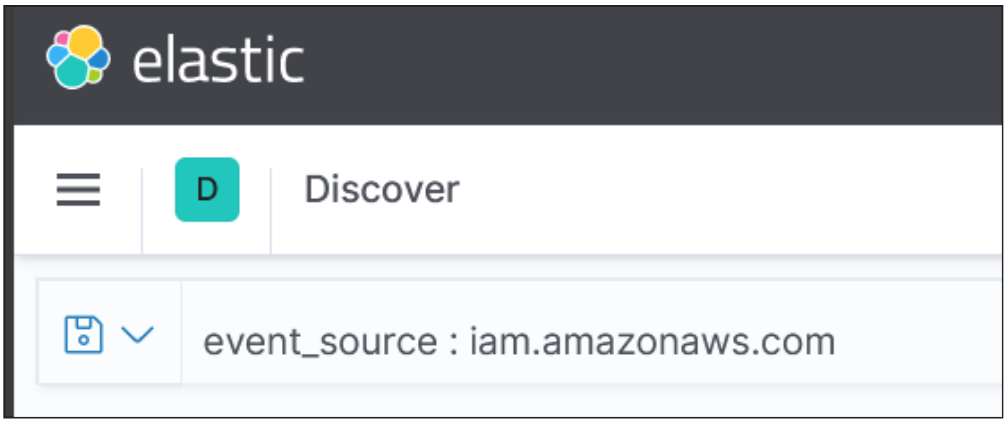


Now that you are in the Discover screen, you will want to choose the AWS index, click on the down arrow and select aws as shown in the picture below.



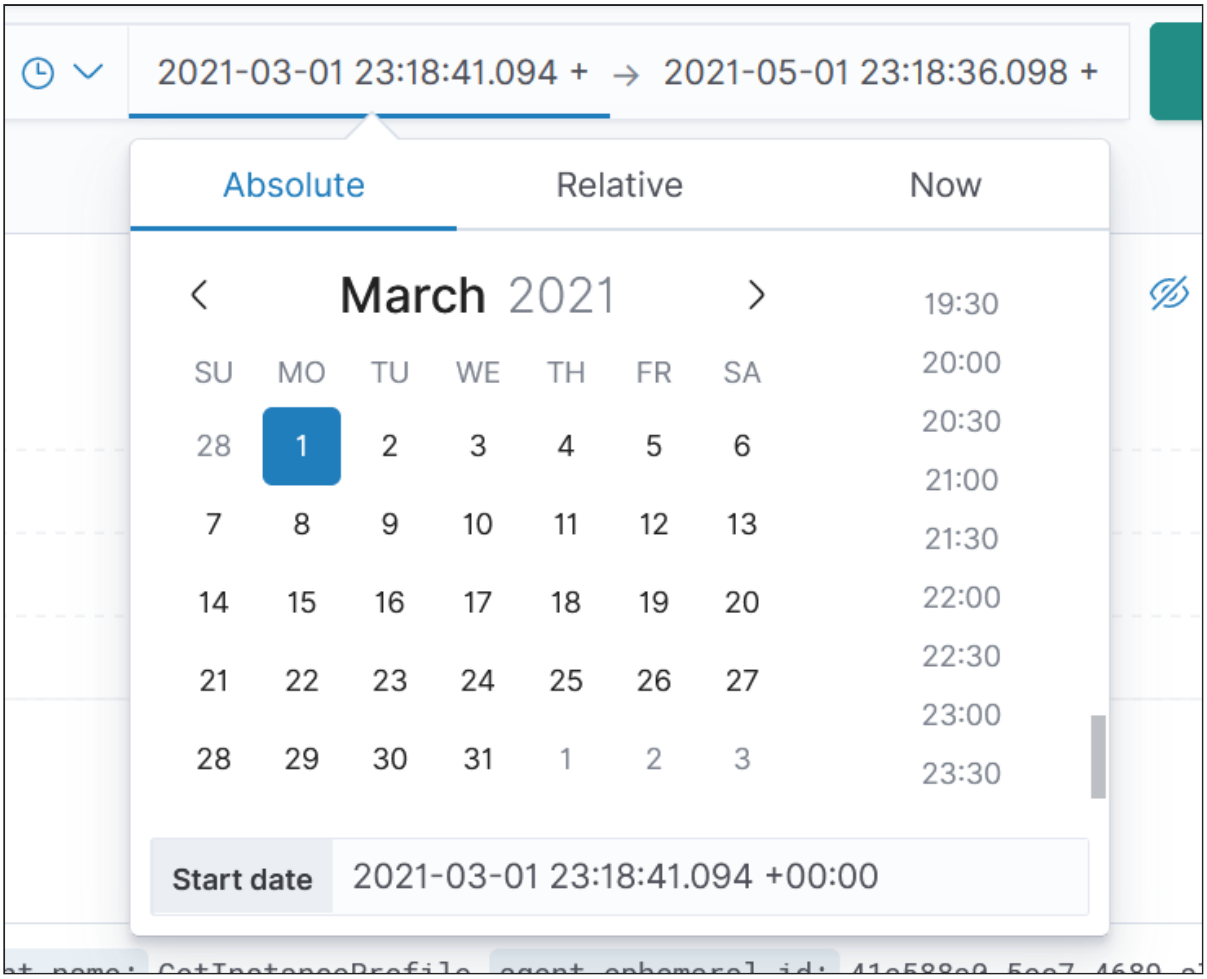
In the filter bar on the top, you will want to filter down our CloudTrail logs to just those that were generated by the IAM service. This service tracks the authentications to the AWS tenant, whether through the console, API or from assumed roles.

```
event_source: iam.amazonaws.com
```

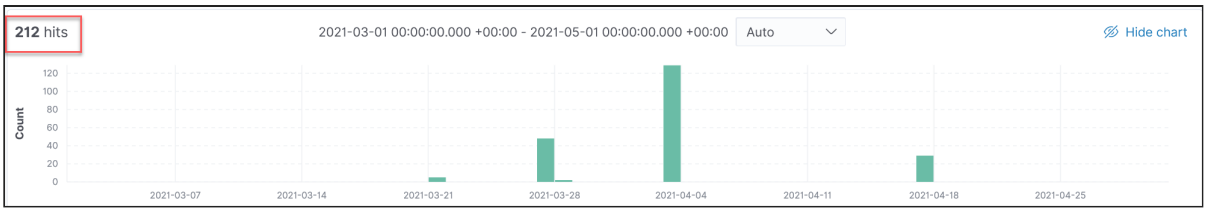


Now that we've selected the right index, we need to set the time frame we want to examine. Pymtechlabs IT staff let us know that strange things started happening after March 1, 2021 and before May 1, 2021. Let's set that as our time frame by using the date filter box on the upper right just like you did in the Lab 1.1. It should look like the following screenshot:

`2021-03-01 23:18:41.094 +00:00` to `2021-05-01 23:18:36.098 +00:00`



You should now have 212 events and a time graph that looks like the one below.



Let's start focusing on the events on 2021-03-21 which seems to be the beginning. If you're not seeing records from 2021-03-21 then click on the down arrow next to the "Time" column to sort it.

Let's take a look at the first record.

What is the Source IP of this entry?

Hint

Look for field named source_host

Answer

47.185.244.137

How did the access occur?

Hint

Look for field named useragent

Answer

The AWS Console

What type of event was this?

Hint

Look for field named event_type

Answer

An API call however since this is from the console, this was an action performed from the web console

What account was used?

Hint

Look for the ARN field

Answer

arn:aws:iam::305681518678:root This is the AWS super user

In what region was the action recorded?

Hint

Look for the aws_region field

Answer

us-east-1, Many resources we will investigate are region specific. Learning where to look for them is critical in doing your investigaion.

When you click on items within the web console, they are recorded as API calls because the website is making API calls on your behalf.

If an authentication occurs to AWS a record gets recorded, however how that authentication took place (API key, console login, etc...) changes where you would find the authentication event that started the session.

Can you find the record where the console session authenticated to AWS?

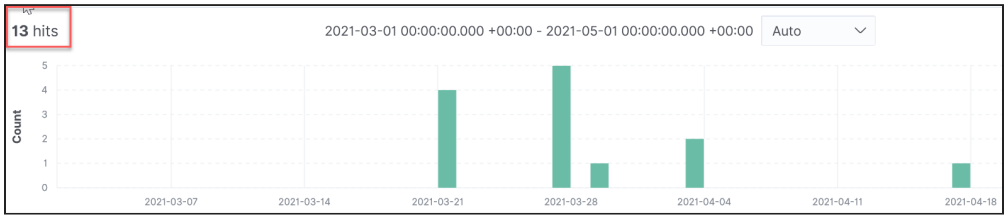
Hint

Logins into the web console are not recorded in the IAM logs, it's a different service/API End point. Change the event_source filter to signin.amazonaws.com

```
event_source: signin.amazonaws.com
```

Answer

- ***The IAM event source records API calls to the IAM service***
- ***The Signin event source records logins***



Take a look at the four events generated on 2021/03/21 and answer the following questions:

How did the user authenticate to AWS (Console, API or IAM user)

Hint

Look at the field event_name

Answer

ConsoleLogin means logged in at the web console

What operating system was the user using?

Hint

Look at the field useragent

Answer

Windows 10. Note that this is what was provided by the client and can be changed by an attacker.

What version of the browser was the user using?

Hint

Look at the field useragent

Answer

Firefox 86. Note that this is what was provided by the client and can be changed by an attacker.

How else could you tell that this was a Console Login?

Hint

Look at the field event_type

Answer

AWSConsoleSignin

On what other days did logins occur?

Hint

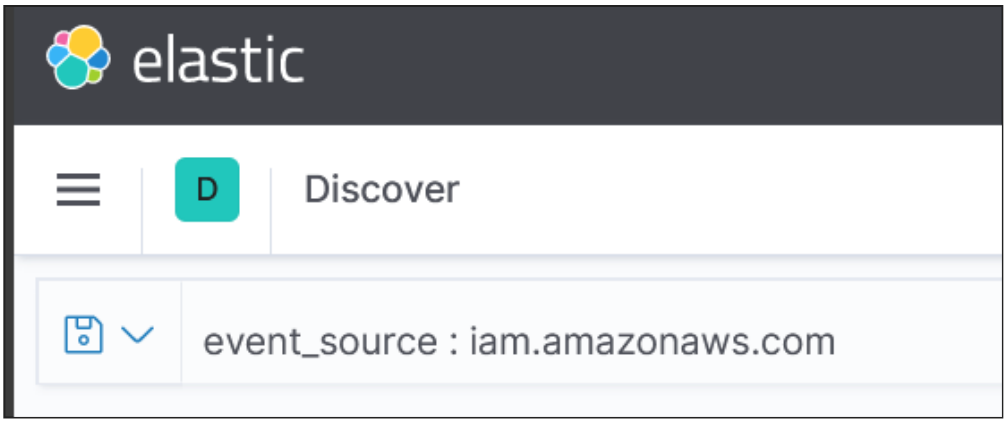
Look at the histogram

Answer

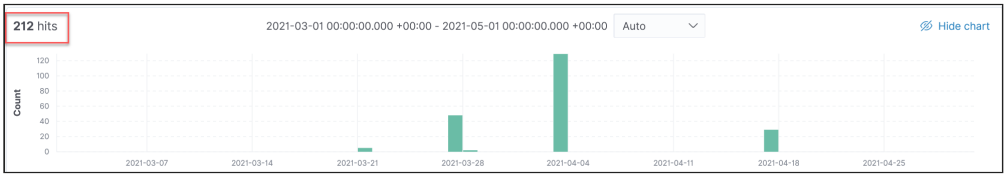
2021/03/27, 2021/03/29, 2021/04/03, 2021/04/17

Change your search back to iam.amazonaws.com

```
event_source : iam.amazonaws.com
```



Make sure your date filter is still set to 2021/03/01-2021/05/01.



Is there any day that looks more interesting than another?

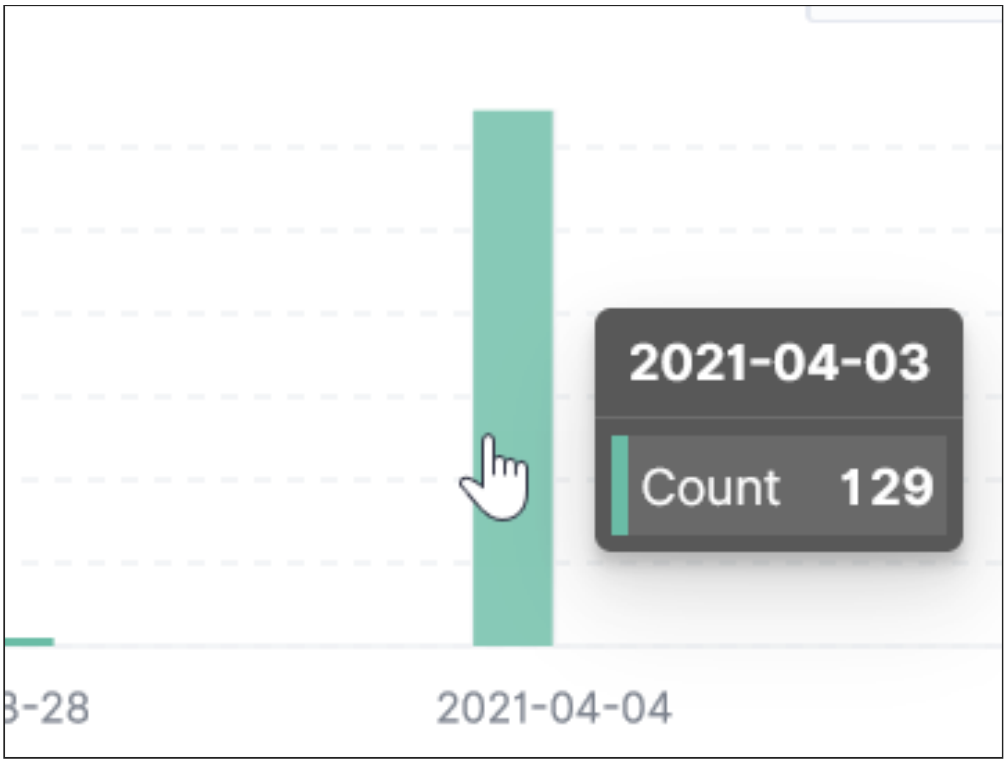
Hint

Look at the counts per day

Answer

2021/04/03 has a large spike of events

Select the day with the large amount of activity by clicking on it's green histogram bar. By clicking on this day we are changing our time filter to only events on 2021-04-03.



Find the record with the timestamp 2021-04-03 18:53:50 and answer the following questions.

What kind of access was this (Console/API/Assumed Role)?

Hint

Look at the field event_type

Answer

AwsApiCall

Did this access originate from the console?

Hint

Look at the field useragent

Answer

No it was the AWS Boto3 API library for windows

What API key was used to make this access?

Hint

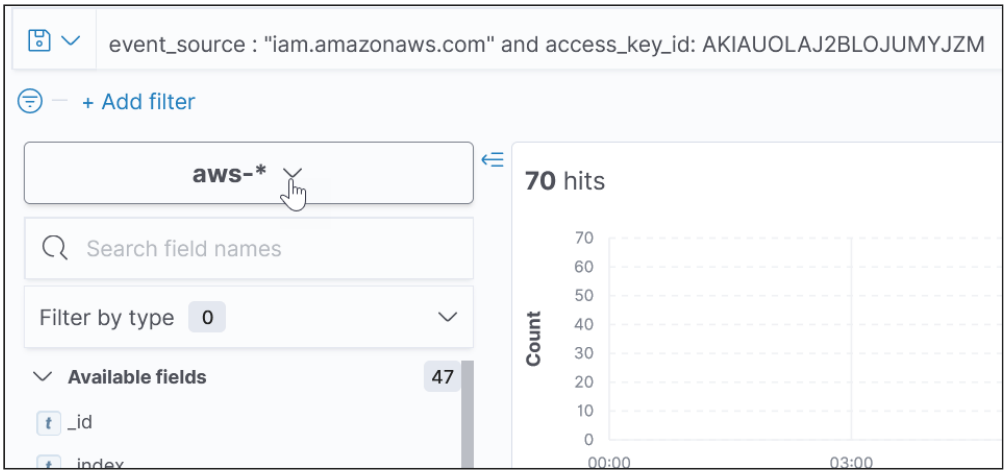
Look at the field access_key_id

Answer

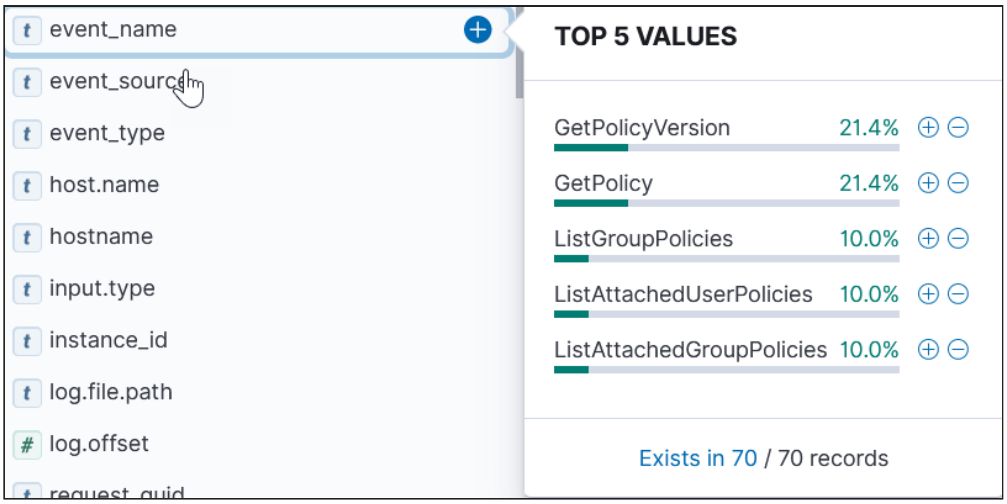
AKIAUOLAJ2BLOJUMYJZM

Add a filter to your view to focus just on the API key you found in the last question, and you should get 70 hits on 2021-04-03.

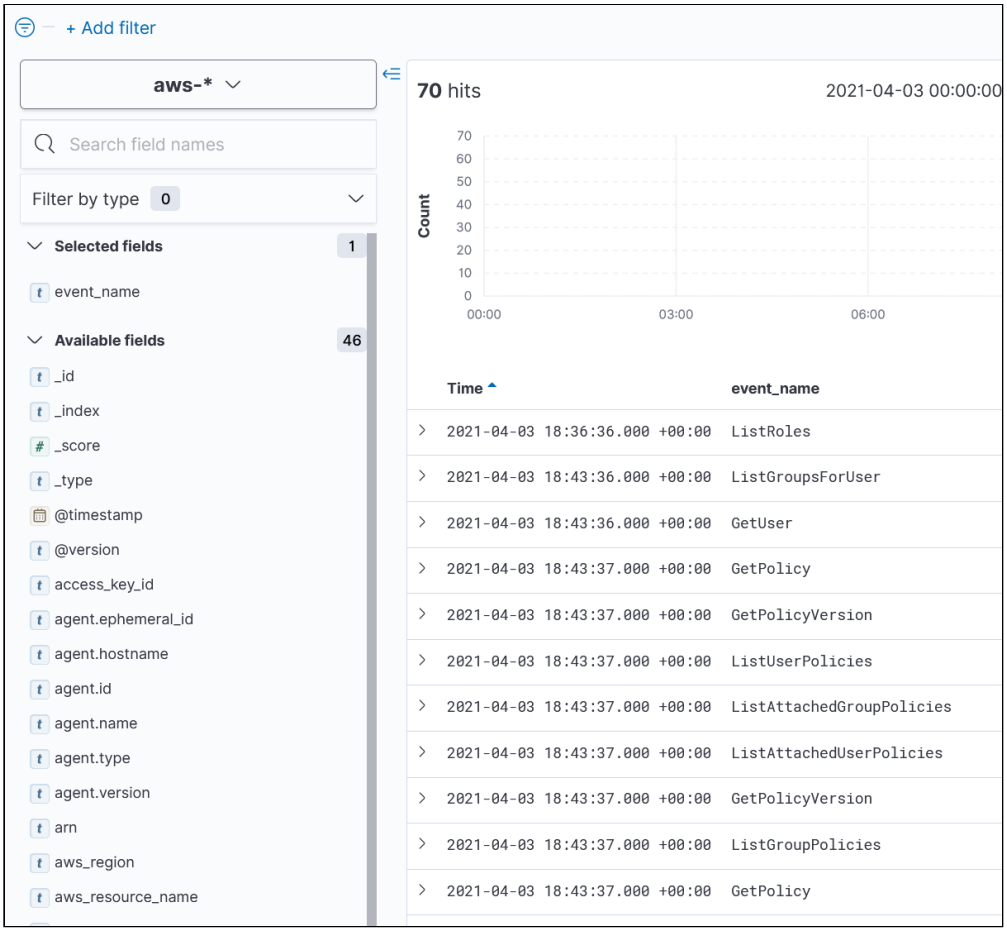
```
event_source : "iam.amazonaws.com" and access_key_id: AKIAUOLAJ2BLOJUMYJZM
```



Click on the field event_name on the left side of the window, it will show you the top five requested events.



Click on the plus sign to reduce the amount of data shown in the middle column.



Read through the event names you see on 2021-04-03 and answer the following questions:

Whose API key was used to do this?

Hint

Look at the ARN field

Answer

arn:aws:iam::305681518678:user/hpym

How long did the access key run commands against the API on this day?

Hint

Look at first and last timestamp

Answer

18:36 - 18:53 17 minutes

What was the first operation taken with the API?

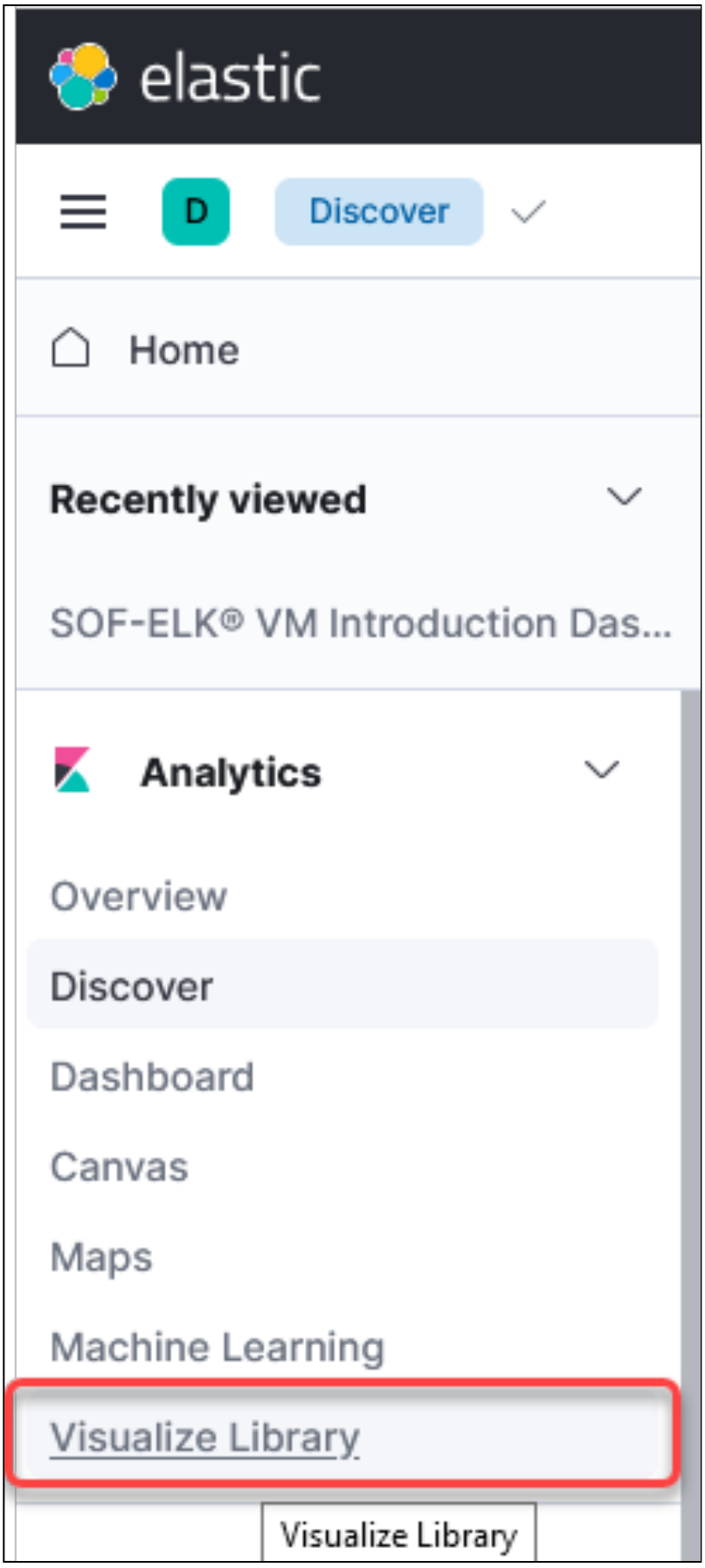
Hint

Make sure you are still sorted by time

Answer

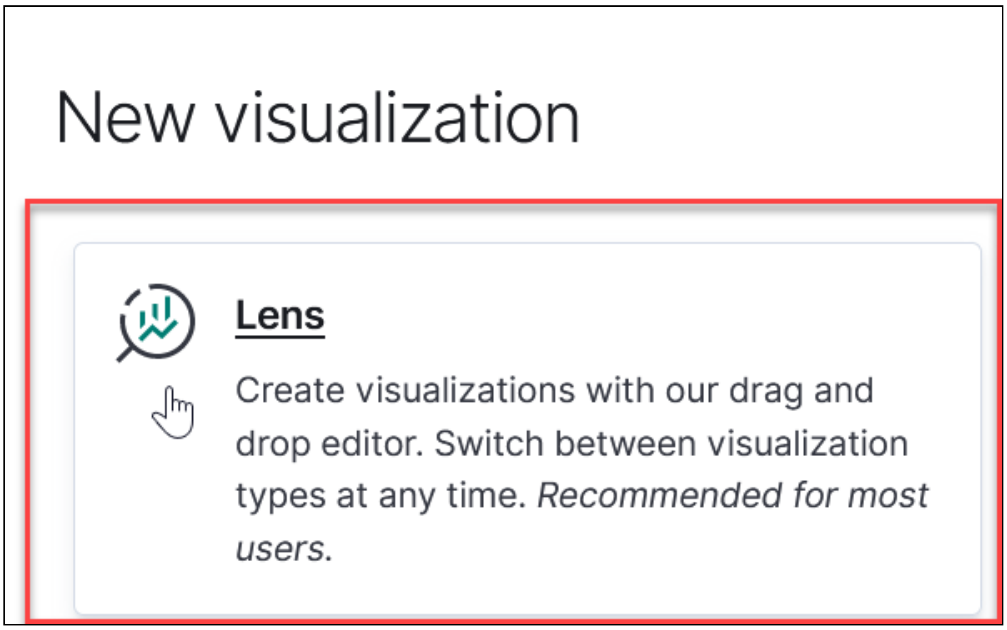
ListRoles

Let's change our view of this data now to understand what is happening with Hank's API key. Switch to the Visualize Library view by clicking on the Hamburger menu again and choosing Visualize Library.

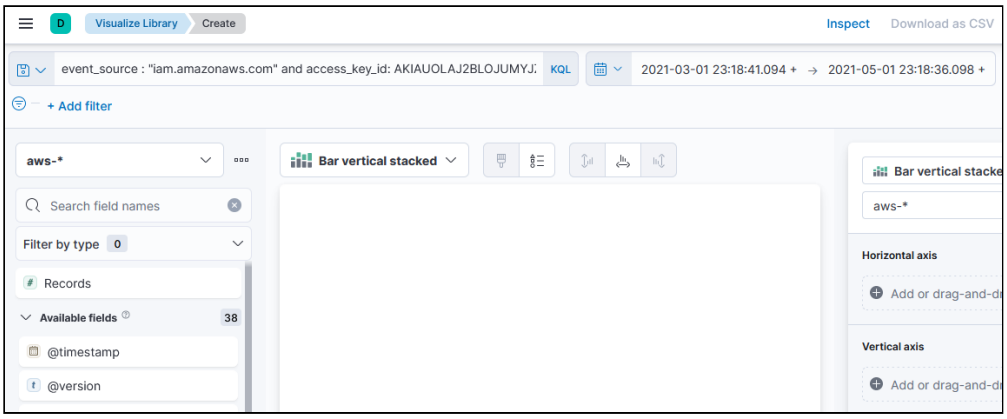

elastic
D
Discover
Home
Recently viewed
SOF-ELK® VM Introduction Das...
Analytics
Overview
Discover
Dashboard
Canvas
Maps
Machine Learning
Visualize Library
Visualize Library

Choose Create Visualizaiton

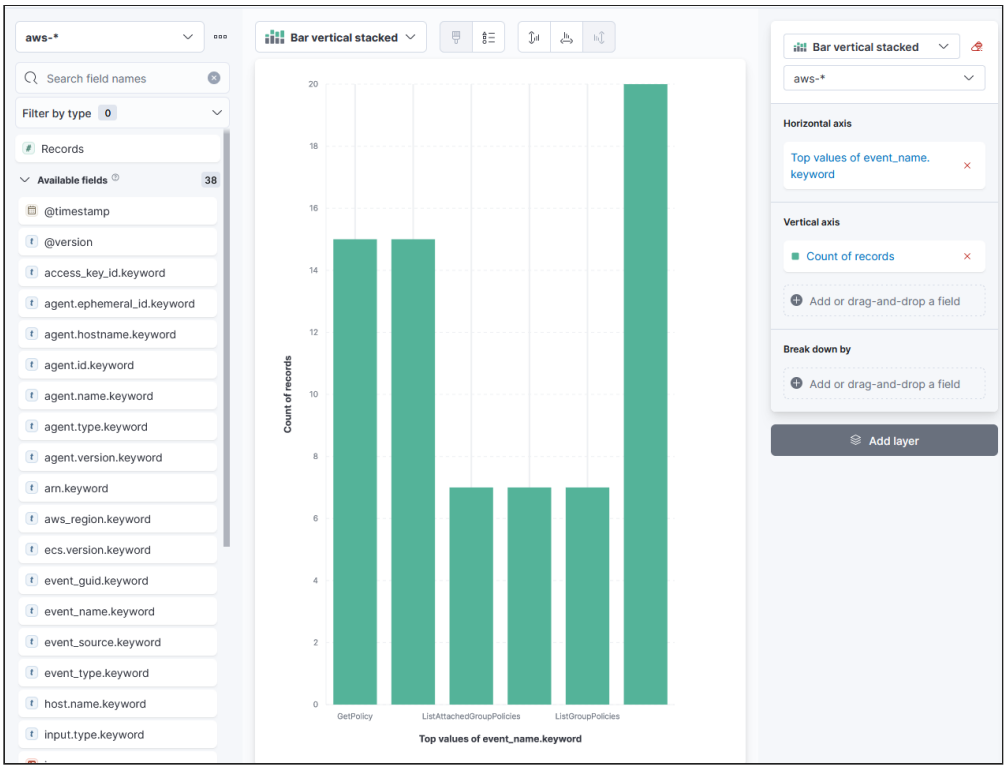
Choose the Lens Visualization



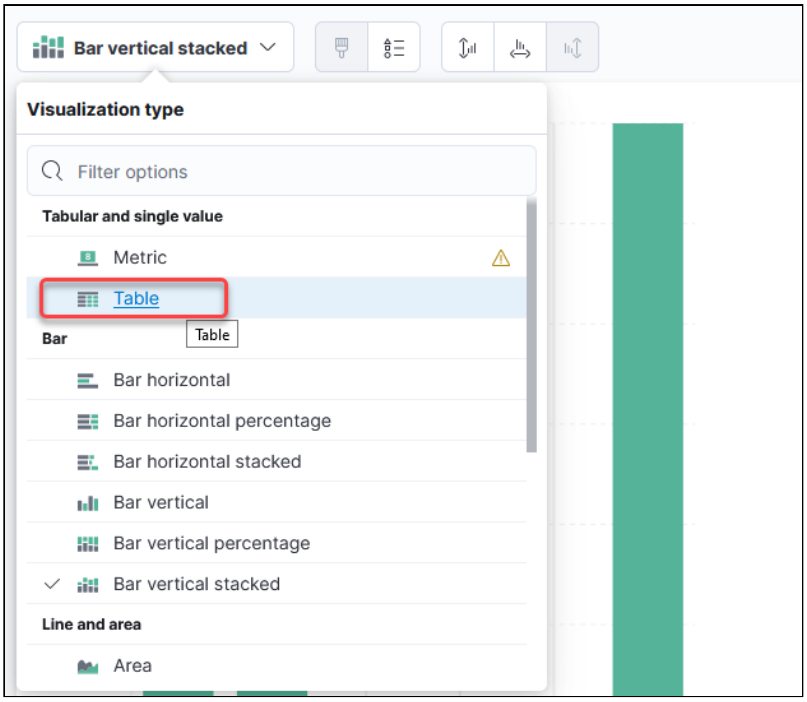
Make sure to select the right index, and your search query and date range should have been brought over. You should have 38 fields available and it should look this:



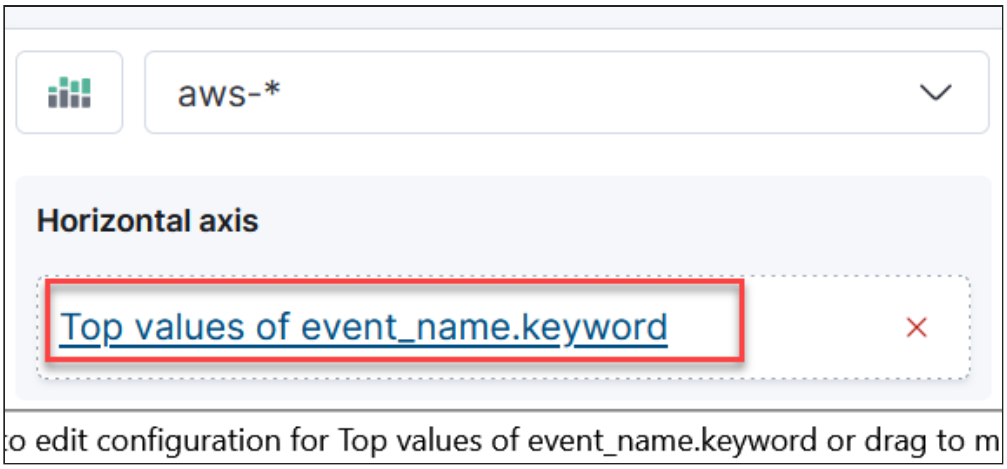
Click on the event_name.keyword field and drag it into the middle to get the view shown below:



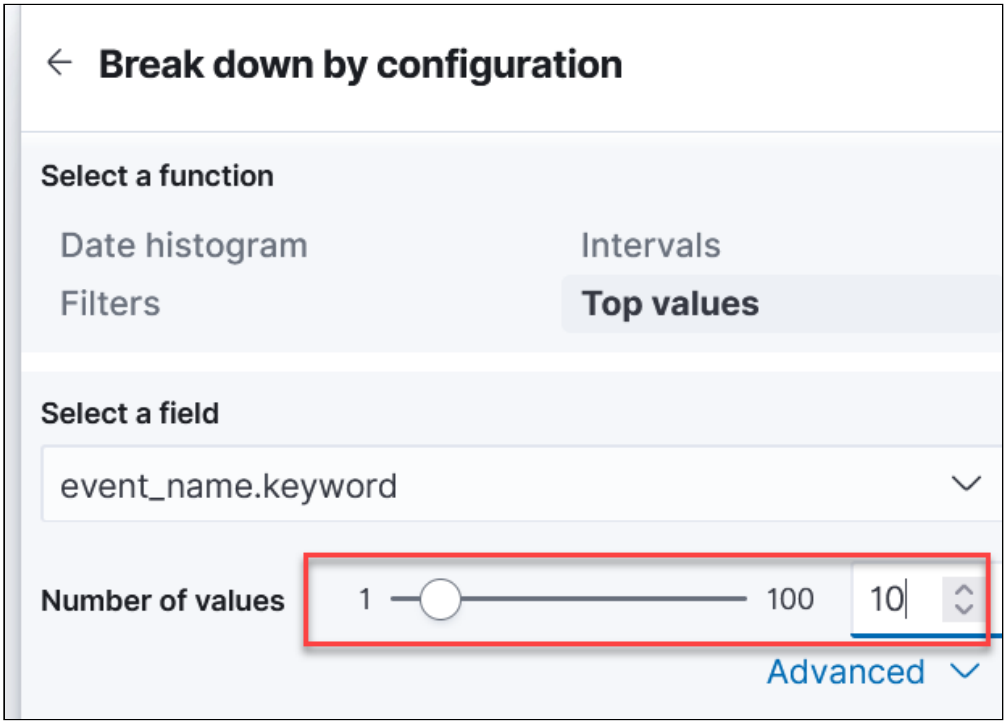
Click on the dropdown that says Bar vertical stacked and switch to Table:



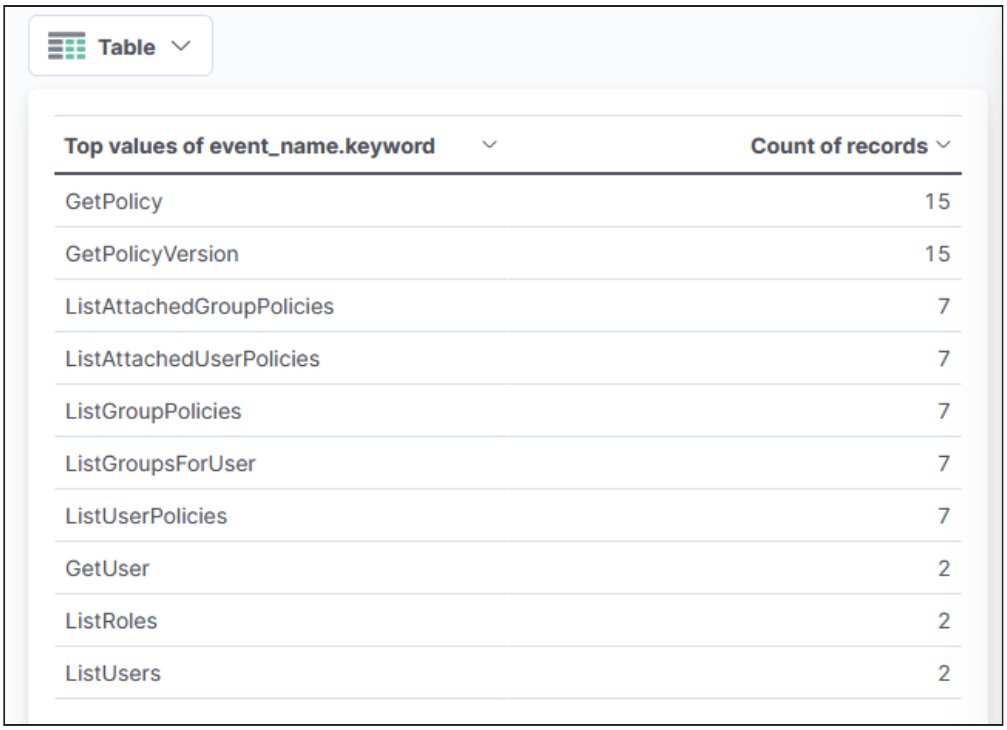
Now click on the Top values box on the right side to include all of the events, not just the top ones.



Change the number of values to 10 to display all of them.



Now you should see a list of API events with how many times they occurred on that day with that specific API key.



Answer the following questions:

What events are recorded only twice?

Hint

Make sure you have all 10 values displayed

Answer

ListRoles, ListUsers, GetUser

What two events were recorded the most?

Hint

The list should be in order so look at the top two values

Answer

GetPolicy and GetPolicyVersion

How many times was ListUserPolicies recorded

Hint

Find the relevant event in the table

Answer

7

Switch back to the Discover view and click confirm on the dialog box that says "Leave Lens with unsaved work."

Now that you've gotten to see the pattern of activity, let's look at the records in order.

What does this activity show?

Hint

Take a look at the repeating pattern

Answer

The API key is being used to enumerate roles and users, and for each user,the policies attached.

If an attacker were to get ahold of an API key for your organization, this would be the type of activity you could expect to see. Most accounts and roles have the ability to query IAM read only. This would let an attacker know which users have elevated roles or policies that they could target to escalate privileges.

Looks like we need to talk to Hank Pym and find out if this was his own testing or an attacker.

Warning

Make sure to clear your filters before moving on to the next lab

Key Takeaways

- You can now determine the difference between a Console login and an API authentication.
- Profiling account accesses and their origin is important when determining if credentials have been compromised.
- Account roles and policy enumeration can be a clue that an attacker is looking to escalate privileges.
- Always look for new keys being created or accounts/API keys logging in from new IPs.

Lab 3.2: Finding Rogue VMs

Objectives

- Learn how to analyze EC2 CloudTrail logs.
- Learn how to find evidence of new virtual machines being created.
- Learn how to analyze the configuration of the system created.

Background

In the prior lab, we noted that an API key appears to have been probing for roles within our tenant. The IAM and Signin event sources had a small number of log entries. Switching to the EC2 log, we find a larger set to analyze.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-01 00:00 UTC** to **2021-05-01 23:30 UTC**.

Load Data

Make sure you have completed Lab 3.1, as this lab requires the data from that lab.

Lab Content

Time Frame

`2021-03-01 00:00:00.000 +00:00` to `2021-05-01 23:30:00.000 +00:00`

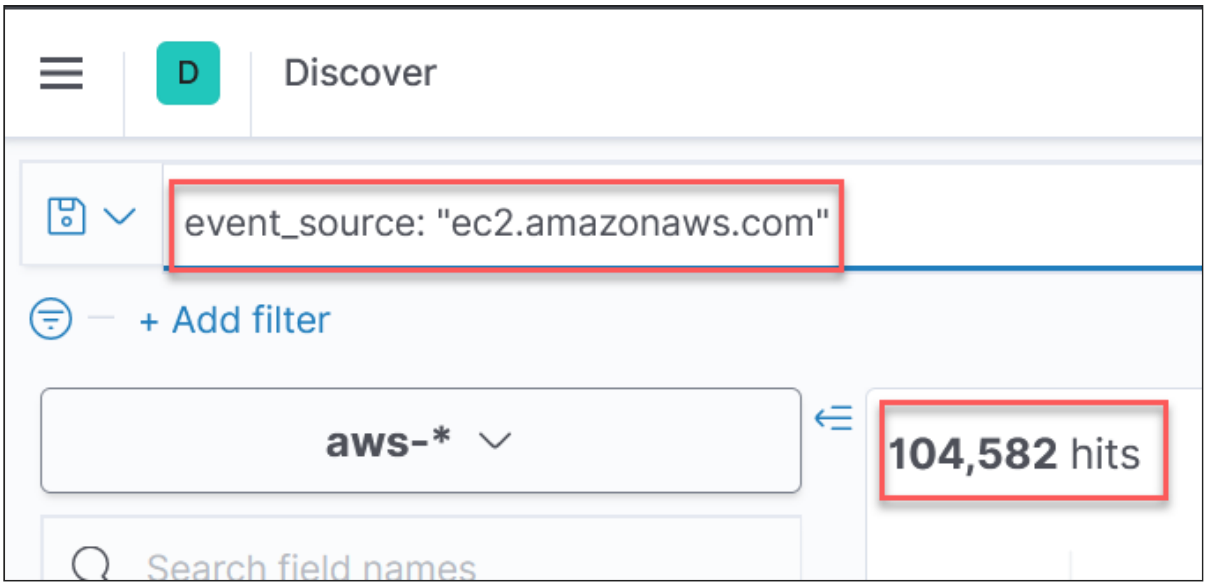
Analyze EC2 CloudTrail Logs

Start in the Discover tab

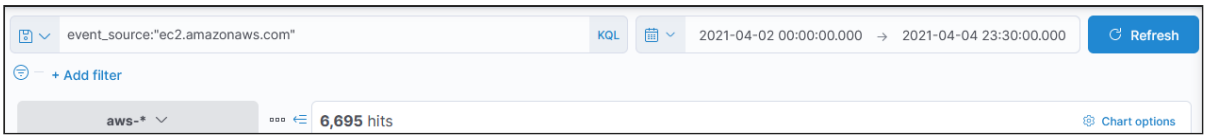
Set your index to `aws-*`

1. Change your filter to focus on EC2 events by filtering your event_source for ec2.amazonaws.com as show below:

```
event_source:"ec2.amazonaws.com"
```



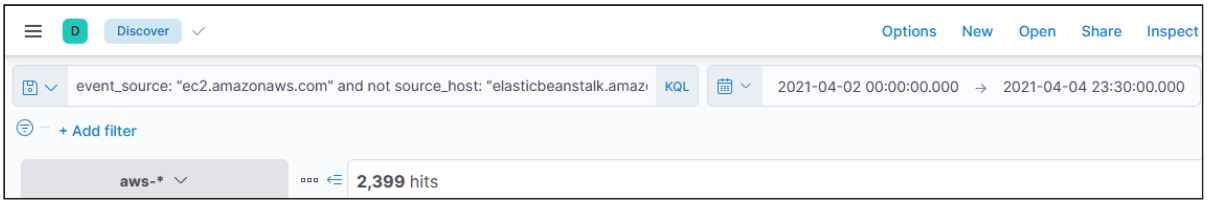
2. You now have 104,582 log entries to look at. That is likely too much for an analyst to start with, so let's focus on the facts we know. The API activity occurred on 2021-04-03. What EC2 events happened on that day. Change your date filter to: `2021-04-02 00:00:00.000 +00:00` to `2021-04-04 23:30:00.000 +00:00` Doing this you should now have 6,695 hits to work with.



3. It looks like our company is using an auto scaling solution called AWS Elastic Beanstalk.

4. Let's filter out the Elastic Beanstalk entries to see if any new virtual machines were created on our day of interest. We will do this by filtering out the entries where elasticbeanstalk.amazonaws.com is the source of the request.

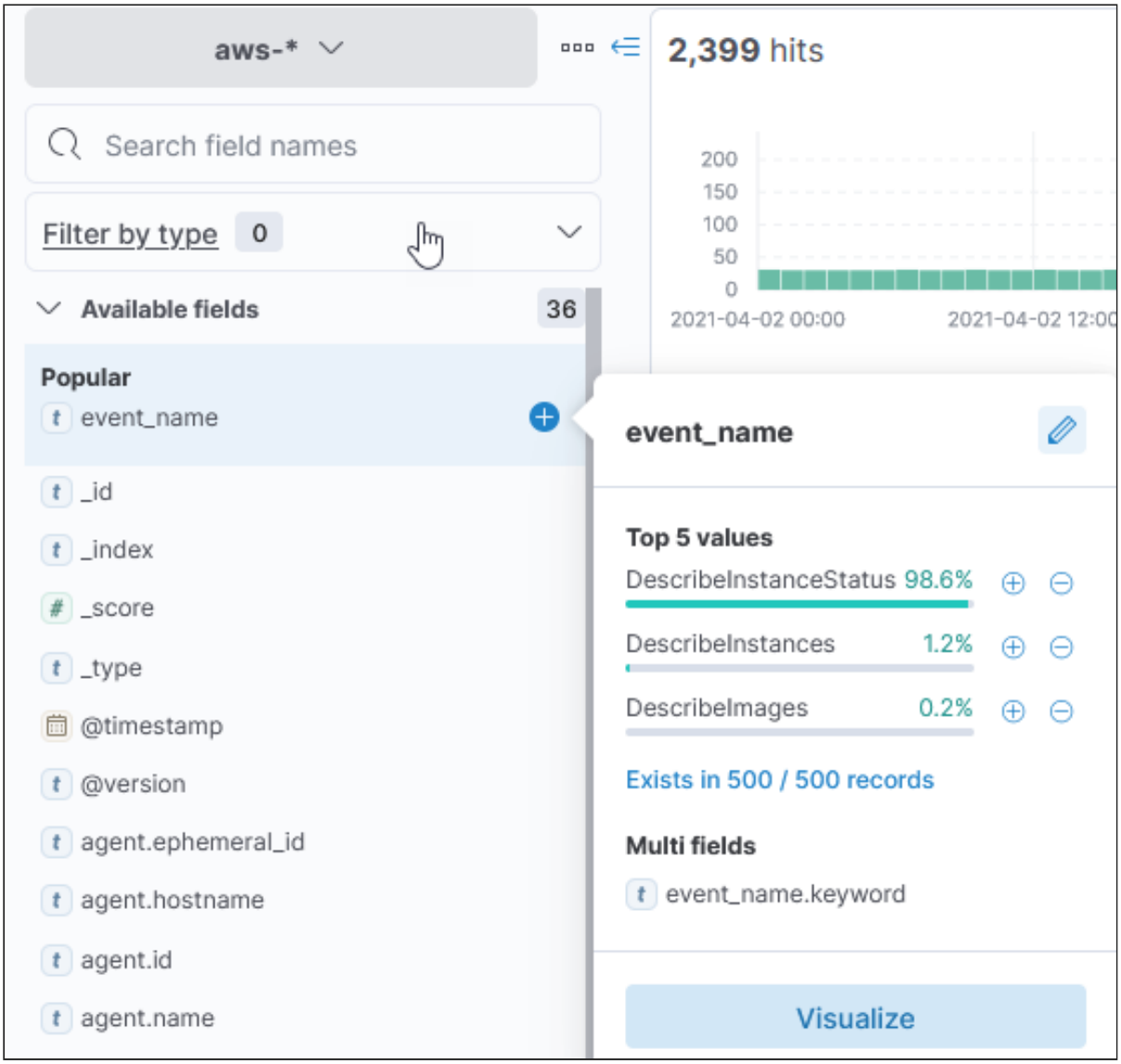
```
event_source: "ec2.amazonaws.com" and not source_host: "elasticbeanstalk.amazonaws.com"
```



We now have 2,399 events, a much more manageable number. Let's now look for what events occurred.

Make sure your time is sorted from newest to oldest!

Click on event_name on the left side and click the plus sign.



What was the first event to be recorded at 2021-04-02 00:01:27?

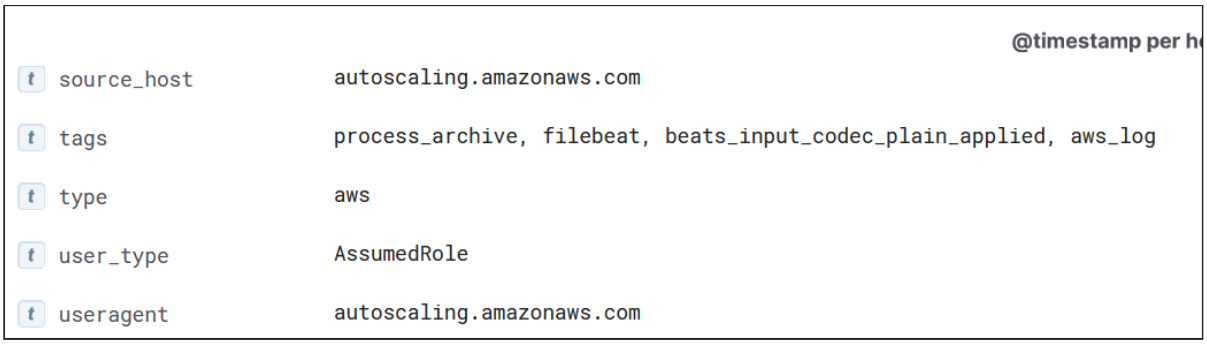
Hint

Look at the event_name

Answer

DescribeInstanceStatus. This is a call you will see many times that shows that some resource was checking to make sure the EC2 instance was running.

Expand out the entry, and you will the see the source_host is autoscaling.amazonaws.com.



Let's click on the plus signs to add source_host and arn to our filtered column view. It should look like this:

Looking at the ARN, we see that this not a root user or IAM user like hpym.

What is an assumed-role?

Hint

Think about how IAM roles are attached to EC2 instances

Answer

This is the IAM role attached to the service or EC2 instance that made the call.

What role is being assumed?

Hint

Look at the ARN column

Answer

In this case, it's the Autoscaling service using authenticating with its IAM role to the EC2 service. Specifically, it's AWSServiceRoleForAutoScaling a built-in AWS role.*

5. Many times attackers will breach a service, instance, or other AWS resource that has an IAM role attached. It's very common for the attacker to then, at a minimum, leverage or at a maximum, steal the role's credentials and use that to expand their access into AWS.

   We are not concerned about what the AWS Autoscaling service is doing, so let's exclude that as a source_host as well:

   ```
   event_source: "ec2.amazonaws.com" and not source_host: "elasticbeanstalk.amazonaws.com" and not
   source_host: "autoscaling.amazonaws.com"
   ```

   You should now have 229 hits to go through

Look at the event that occurred on 2021-04-03 22:07:07:

What does RunInstances mean?

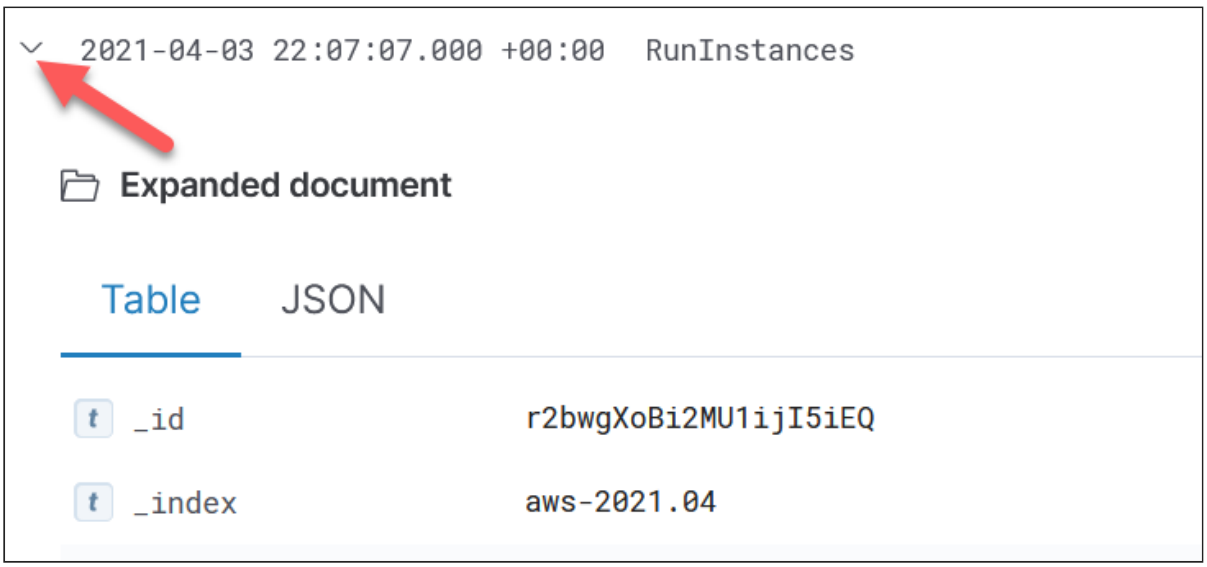
Hint

Look at your slides

Answer

RunInstances is what is logged when an EC2 VM is created

Click the arrow to expand the record:



What is the imageID of the EC2 VM?

Hint

Look at the field image_id

Answer

ami-037ce2fddcbfdbf52

What kind of VM was it?

Hint

Look at the instance_type field

Answer

g4ad.4xlarge

What was the name of the ssh key pair generated?

Hint

Look at the field key_name

Answer

NewEC2

What account was used to create the EC2 VM?

Hint

Look at the field arn

Answer

root

6. Looks like someone started a new EC2 instance. Let's determine what else was created. Change your filter to look for all event_name entries that start with create* with the following search:

```
event_source: "ec2.amazonaws.com" and not source_host: "elasticbeanstalk.amazonaws.com" and not
source_host: "autoscaling.amazonaws.com" and event_name: create*
```

This gets the results down to 21 hits.

What resources were created?

Hint

Look for the unique event names

Answer

CreateTags, CreateNetworkInterface, CreateKeyPair and Create SecurityGroup

In most organizations, the root user isn't used for any minor changes like creating an EC2 instance, instead IAM accounts are used for that. We need to investigate further to find out if our root account was compromised.

Warning

Make sure to clear your filters before moving on to the next lab

Key Takeaways

- You can now filter down to EC2 logs in CloudTrail
- You now know how to find and interpret actions being taken against EC2 hosts

Lab 3.3: VPC Flow Logs

Objectives

- Understand how to read AWS Flow Logs.
- Learn how to profile network traffic.
- Find possible exfiltration of data.

Background

Pymtech Labs got an alert from the SOC. It looks like between 2021/05/02 and 2021/05/03 a large data transfer was spotted going out of our new AWS network. We need you to look at the netflow from our AWS environment to see if you can find out what happened.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-05-02 00:00 UTC** to **2021-05-04 00:00 UTC**.

Load Data

In Lab 0 you loaded a GeoIP database which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the AWS parsers and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

The AWS VPC Flow Logs for this lab have already been exported from our S3 Bucket. We've tried to make the loading process as close to what you will do in the real world as possible.

The VPC Flow Logs for lab-3.3 have been saved into `/home/elk_user/lab-3.3_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract the all the files from the preceding ZIP file for ingestion into Kibana above with the following command:

Command lines

```
cd ~
unzip -q lab-3.3_source_evidence.zip  -d /logstash/nfarch
```

Warning

Do not run this command more than once!

3. Wait 2-3 minutes for the data to be processed. Once it is complete, the data will be available in the `netflow-*` index.
4. Verify that you have 258,499 documents loaded in the `netflow` index:

Command lines

```
sof-elk_clear.py -i list
```

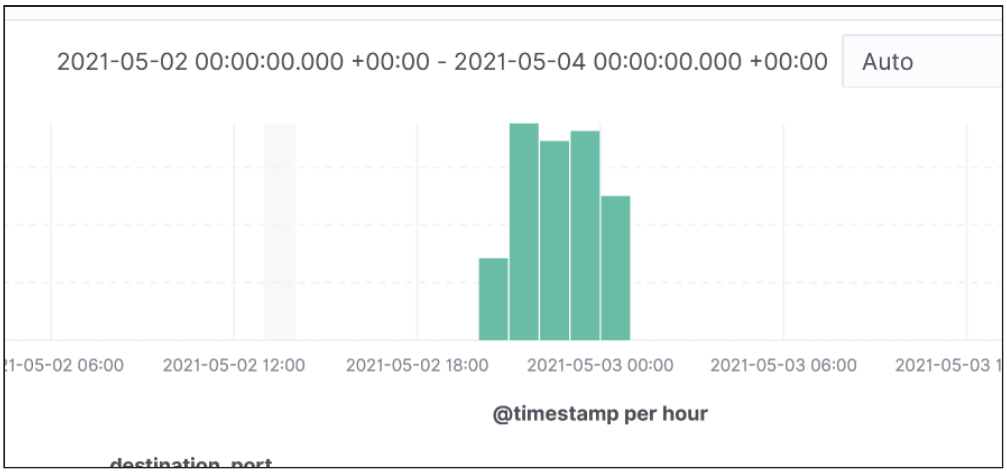
Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (213,775 documents)
- azure (6,217 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

Lab Content

Time Frame

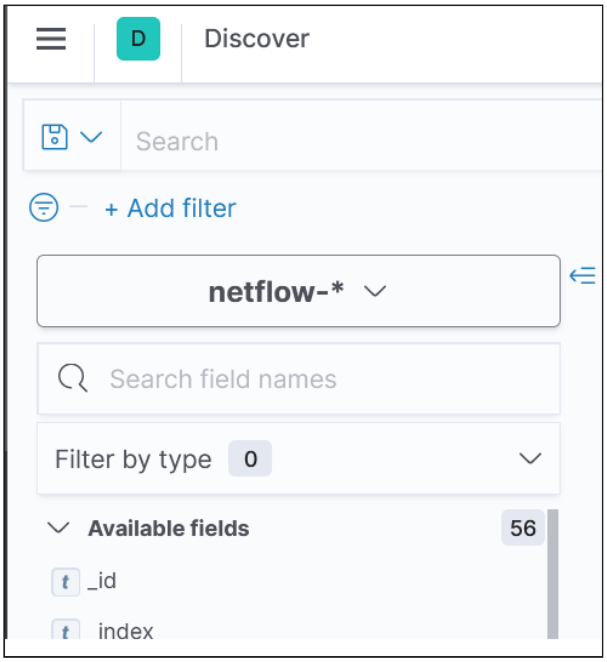
`2021-05-02 00:00:00.000 +00:00` to `2021-05-04 00:00:00.000 +00:00`



AWS Flow Logs

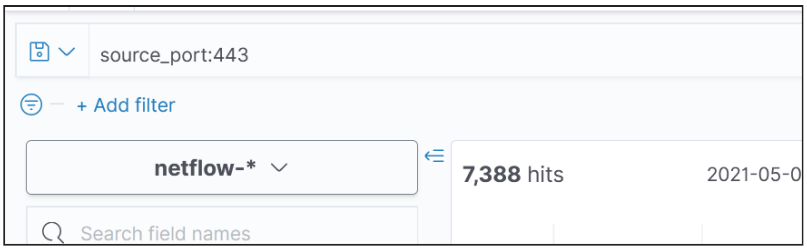
Start in the Discover tab

Set your index to `netflow-*`



1. Add a filter for source_port:443.

```
source_port:443
```



Make sure you have sorted by time descending.

2. Look for the record that occurred on "2021-05-03 00:40:57.000Z" and answer the following questions:

What is the source of the traffic?

Hint

Look at the source_ip field

Answer

54.224.26.45

What was the destination of the traffic?

Hint

Look at the destination_ip field

Answer

192.168.52.168

How long did the session last?

Hint 1

Subtract flow_end from flow_start

Hint 2

Look at flow_duration

Answer

6 seconds

How much data was transmitted and received?

Hint

Look at total_bytes

Answer

34,999 bytes

What organization owns the ip that hosted the data transferred?

Make sure you've done lab 0 and imported the MaxMind database

Hint

Look at the source_geo.as_org

Answer

Amazon

What country is the ip hosted in?

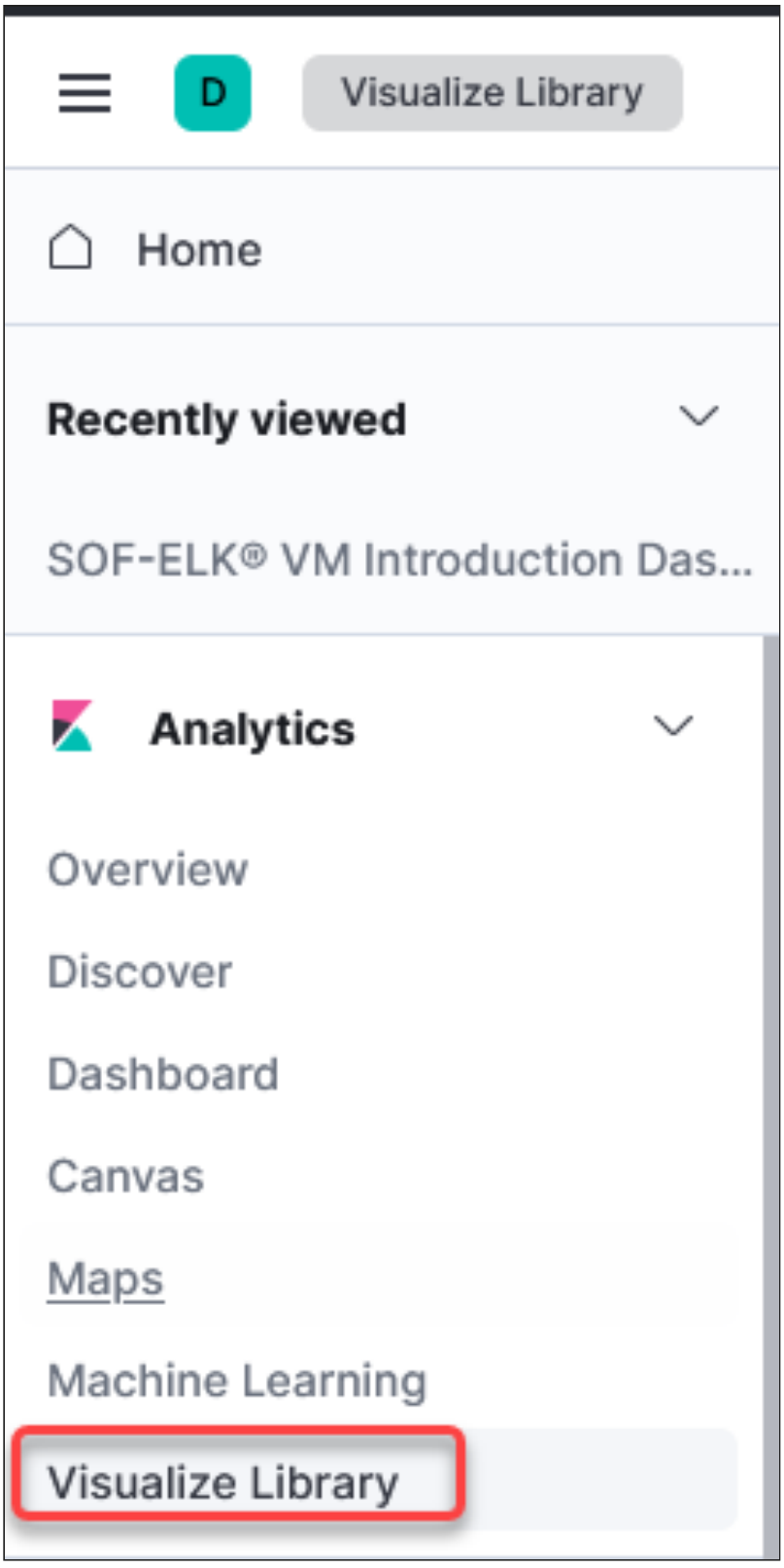
Hint

Look at source_geo.country_name

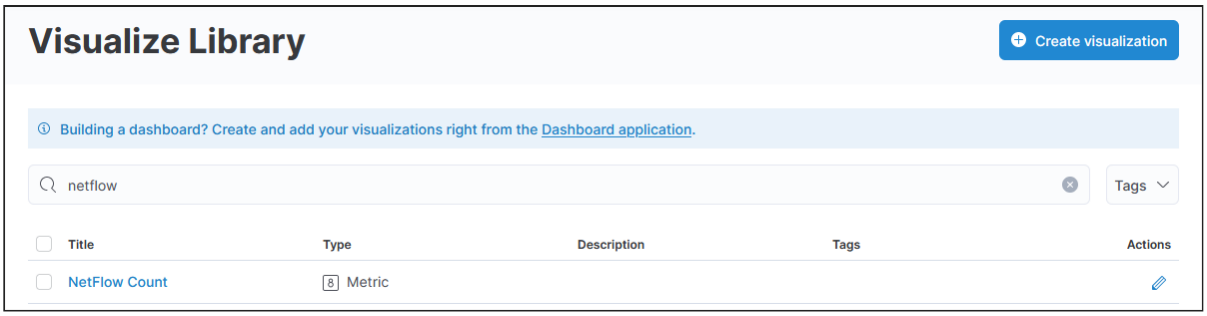
Answer

United States

3. Now that you know how to read a log, let's change our view to look for the alleged data theft. Click on the menu button and then Visualize Library.

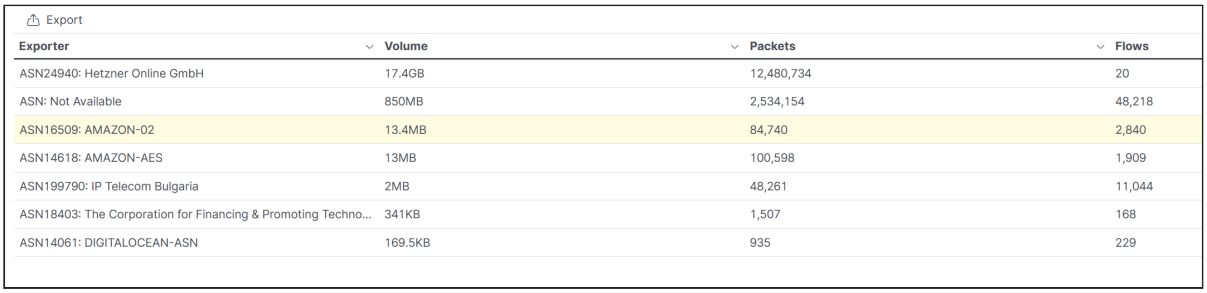


1. Filter for netflow visualizations.



1. Click on "Netflow Statistics by Destination AS"

1. You should now see a view similar to the following:



Exporter	Volume	Packets	Flows
ASN24940: Hetzner Online GmbH	17.4GB	12,480,734	20
ASN: Not Available	850MB	2,534,154	48,218
ASN16509: AMAZON-02	13.4MB	84,740	2,840
ASN14618: AMAZON-AES	13MB	100,598	1,909
ASN199790: IP Telecom Bulgaria	2MB	48,261	11,044
ASN18403: The Corporation for Financing & Promoting Techno...	341KB	1,507	168
ASN14061: DIGITALOCEAN-ASN	169.5KB	935	229

Which organization has the largest amount of data transfer?

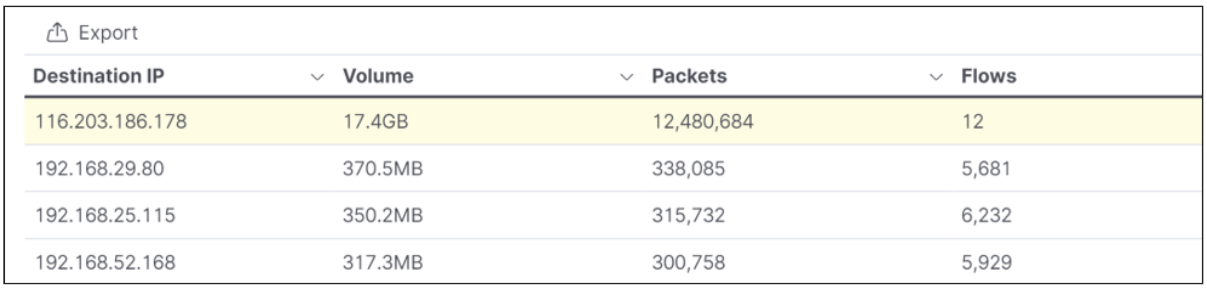
Hint

The top row has the name

Answer

Hetzner Online GmbH with 17.4 GB

1. Go back to the Visualize page and filter for netflow again. This time choose "Netflow Statistics by Destination IP." You should see something similar to the following:



Destination IP	Volume	Packets	Flows
116.203.186.178	17.4GB	12,480,684	12
192.168.29.80	370.5MB	338,085	5,681
192.168.25.115	350.2MB	315,732	6,232
192.168.52.168	317.3MB	300,758	5,929

Which IP corresponded to the traffic we saw ranked by org in the previous step (no hint)?

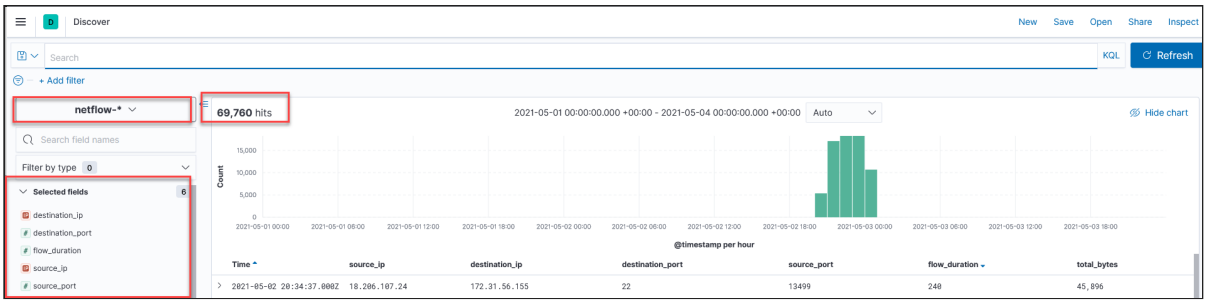
116.203.186.178

1. Going to that IP shows a service called bashupload.com.

What does bashupload allow for (no hint)?

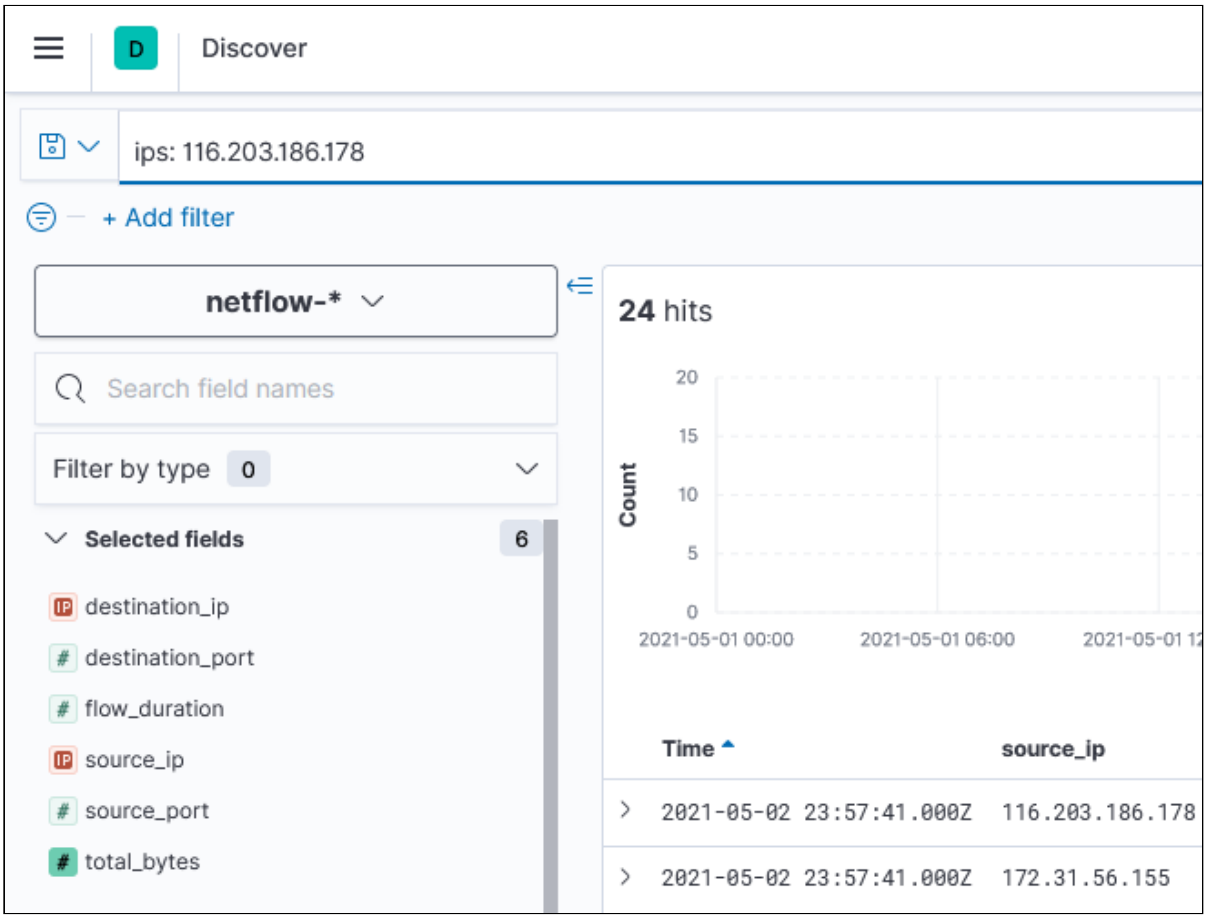
Uploading files from the command line

1. Let's look deeper at what was happening before the upload began. Switch back to the Discover interface, select the netflow index, and add the following fields to make a filtered-down view:
    - source_ip
    - destination_ip
    - destination_port
    - source_port
    - flow_duration
    - total_bytes



Now with the fields filtered down we will perform a search for all activity involving 116.203.186.178 which is the IP we saw the data exfiltrated to.

```
ips: 116.203.186.178
```



With 24 hits, now we can see that the upload activity in much more clarity. Answer the following questions.

What ip is the upload occurring from?

Hint

Look at the source_ip where the destination is 116.203.186.178 and the destination port is 80

Answer

172.31.56.155

What is the earliest record of communication with 116.203.186.178

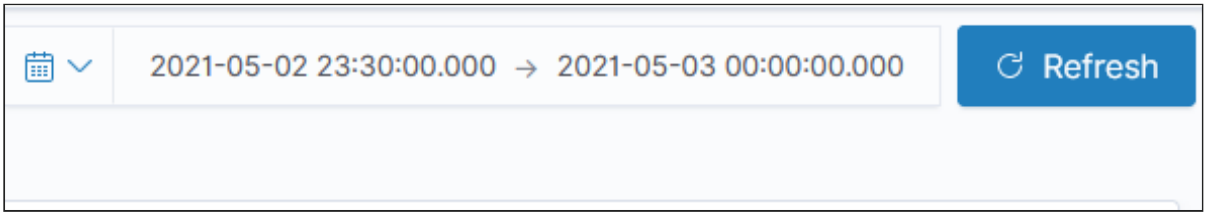
Hint

Make sure your timestamps are sorted

Answer

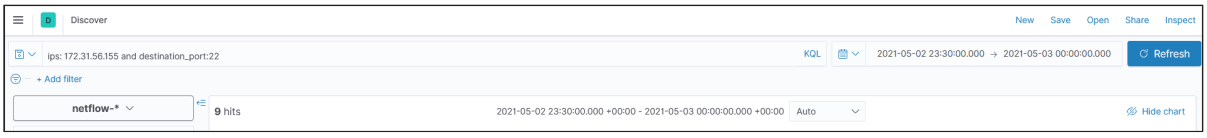
2021-05-02 23:57:41

Let's change our date filter to focus down on a 30-minute time frame during the upload. Change your date filter to 2021-05-02 23:30:00 to 2021-05-03 00:00:00.



Next, we can see what was communicating with our compromised host prior to the exfil occurring. Change your search to the vm's ip and the destination port to 22, which is the SSH port, to see who was communicating with the vm prior to the upload.

```
ips: 172.31.56.155 and destination_port:22
```



You should now have nine records returned.

What is IP is most likely involved

Hint

Look at flow duration

Answer

18.206.107.24. The other IPs have too short of a flow duration and too few bytes transferred

Who does this IP belong too?

Hint

Add one of the geo columns or google the ip

Answer

This is an AWS IP

Taking a look at this IP it belongs to AWS. This appears to have been an access from another AWS resource, we will have to look closer into this!

Warning

Make sure to clear your filters before moving on to the next lab

Key Takeaways

- You've learned how to read an AWS VPC Flow Log.
- You've learned how to narrow the timeframe to find likely sources. A review of the host artifacts would help to validate our theory.
- We've discovered a large amount of data exfiltrated. We should look into this and find out how it happened.

Lab 3.4: S3 Analysis

Objectives

- Learn how to read S3 server access logs.
- Determine how an S3 bucket ACL gets changed.
- Determine who accessed the S3 bucket.

Background

Scott had a long frustrating day and set an S3 bucket to public access to get a data transfer going to an agency. He forgot to make the bucket private afterward and wants to know if you can determine if anyone downloaded the data from the bucket.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-04-30 00:00 UTC** to **2021-05-02 00:00 UTC**.

Load Data

In Lab 0 you loaded a GeoIP database which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the AWS parsers and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

The AWS S3 Server Access Logs for this lab have already been exported from our S3 Bucket. We've tried to make the loading process as close to what you will do in the real world as possible. As such, with how AWS stores data there, we have augmented the Logstash HTTPd parser to support AWS S3 Server Access Logs since they are a modified form of standard web server logs.

The AWS S3 Server Access Logs for lab-3.4 have been saved into `/home/elk_user/lab-3.4_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract all the files from the preceding ZIP file with the following command:

Command lines

```
cd ~
unzip -q lab-3.4_source_evidence.zip
```

Run the AWS CloudTrail ingestion script:

Command lines

```
aws-cloudtrail2sof-elk.py -r ./lab-3.4_source_evidence -w /logstash/aws/lab3_4.json
```

Then extract the AWS S3 Server Access Logs:

Command lines

```
unzip -q lab-3.4_s3_evidence.zip -d /logstash/httpd
```

Warning

Do not run these commands more than once!

3. Wait 2 minutes for the data to be processed. Once it is complete the data will be available in the `aws-*` index and the `httpdlog-*` index.
4. Verify that you have 240,899 documents loaded in the `aws` index and 1,381 documents in the `httpdlog` index.

Command lines

```
sof-elk_clear.py -i list
```

Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- httpdlog (1,381 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

Lab Content

Time Frame

Based on Scott's recollection of when he made changes, set your time frame as follows: `2021-04-30 00:00:00.000 +00:00` to `2021-05-02 00:00:00.000 +00:00`

S3 Server Access Logs

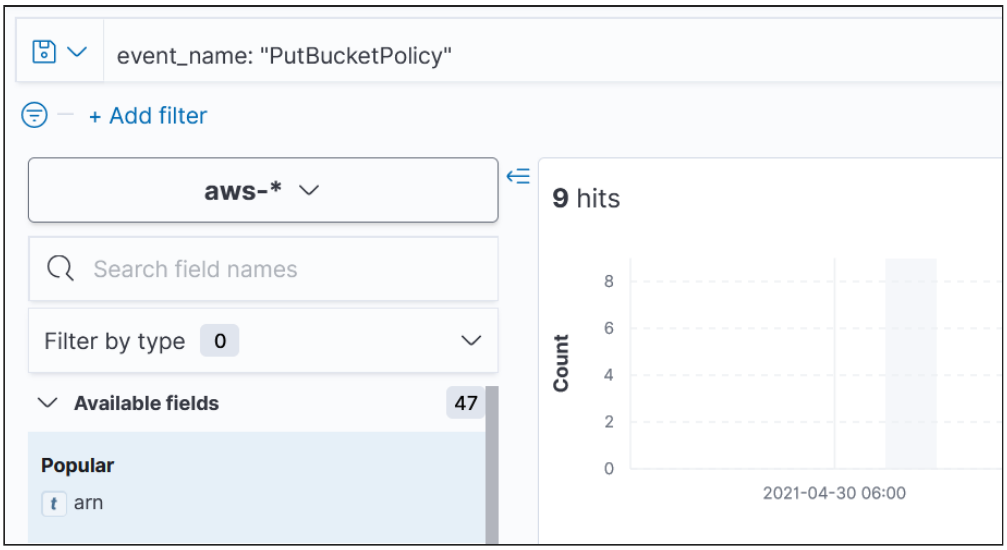
Start in the Discover tab

Set your index to `aws-*`

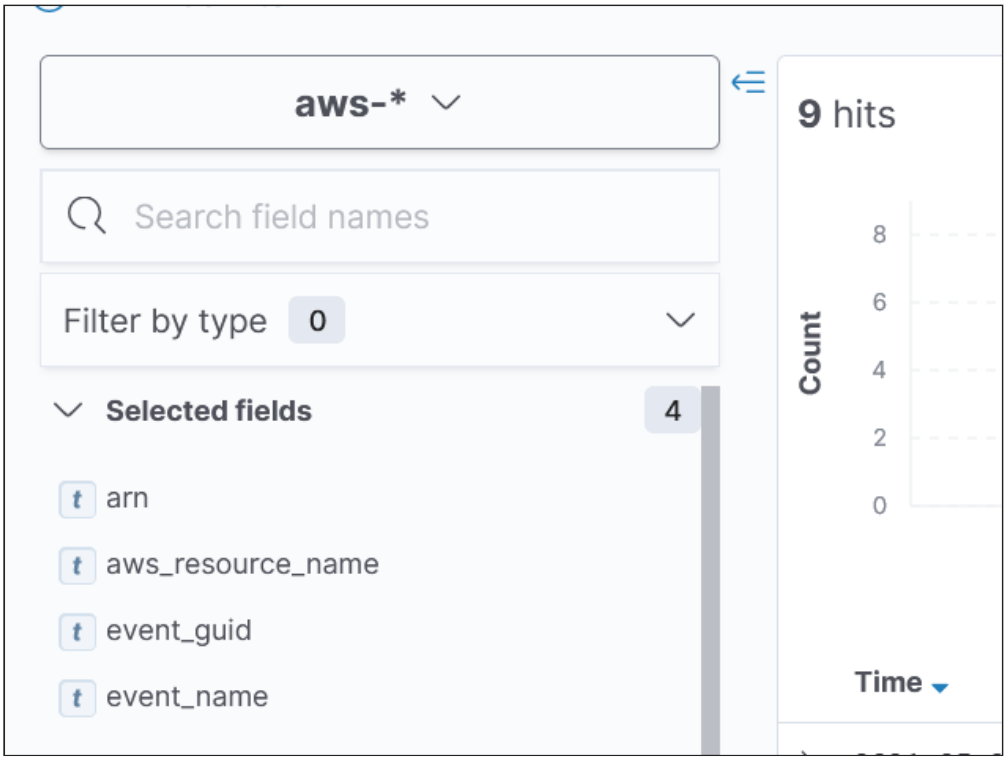
Set a filter for the event_name PutBucketPolicy.

```
event_name: "PutBucketPolicy"
```

You should have nine records returned.



- Choose the following fields in the left column by clicking the blue plus sign:
  - arn
  - event_name
  - aws_resource_name
  - event_guid



What day did Scott change the permissions on the bucket (no hint)?

2021-05-01

What bucket was affected?

Hint

Look at the aws_resource_name field

Answer

arn:aws:s3:::pymresearch

In order to determine what was changed we have to go deeper than we can with just SOF-ELK. SOF-ELK is an amazing log analysis tool, but sometimes the log structure varies too much between log types to be able to pull out every available element.

We can use the information we gather from SOF-ELK to allow us to dig deeper with other tools like "jq". To pull out the detail for a specific record we need a unique identifier for that record. In the results on your SOF-ELK query, the event_guid is that unique identifier. Let's start with the record at time 2021-05-01 02:42:57, which has an event_guid of "0cd5b2f2-529e-458d-ab04-ee805217c479".

Go to the SOF-ELK terminal (we recommend using an SSH client over the VM console).

In our SOF-ELK terminal execute the following command

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "0cd5b2f2-529e-458d-ab04-ee805217c479")
| .requestParameters'
```

Expected results

```
[elk_user@sof-elk ~]$ cat /logstash/aws/lab3_4.json | jq 'select(.eventID ==
"0cd5b2f2-529e-458d-ab04-ee805217c479") | .requestParameters'
{
"bucketPolicy": {
  "Version": "2021-4-30",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": [
        "arn:aws:s3:::pymresearch/*"
      ]
    }
  ]
},
"bucketName": "pymresearch",
"Host": "s3.amazonaws.com",
"policy": ""
}
```

- Looking at this first policy change, we can see the following:
  - Sid is PublicRead meaning this is a policy change to the reading of objects in this bucket without authentication.
  - Effect is allow meaning that whatever action is specified will be allowed.
  - Action shows what S3 operations are being affected (in this case, GetObject and GetObjectVersion).
  - Resource is showing us what bucket is being affected.

Go through the other eight policy changes and answer the following questions.

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "24196e22-654c-4ad3-8885-27ef09afa4ad")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "8af0fc90-f6bc-4696-80f2-76c3a3e4dc36")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "5f7df8b4-25db-40b0-915a-2b6fe0deb8b5")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "3acfeaac-b3f8-4071-894f-73d3d48c528d")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "1a72bea8-8fae-4bf9-8413-b3af06dac5eb")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "6d7bd065-939d-4ee3-aaf0-210e6f717432")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "eddda341-4a00-4955-b924-cee059bb35db")
| .requestParameters'
```

Command lines

```
cat /logstash/aws/lab3_4.json | jq 'select(.eventID == "c6664ccf-4384-4172-b86e-30827f895ba0")
| .requestParameters'
```

Which event ID added the list bucket permission first?

Hint

Look in the time order shown above

Answer

5f7df8b4-25db-40b0-915a-2b6fe0deb8b5

Which event ID started a series of blank policies?

Hint

Blank policies have no data in the bucketPolicy field

Answer

3acfeaac-b3f8-4071-894f-73d3d48c528d. Blank policy fields normally means there was an error in the json syntax entered

Which event ID added the GetBucketLocation permission

Hint

Look at what's changing in the actions. GetBucketLocation returns the region where a bucket can be found in.

Answer

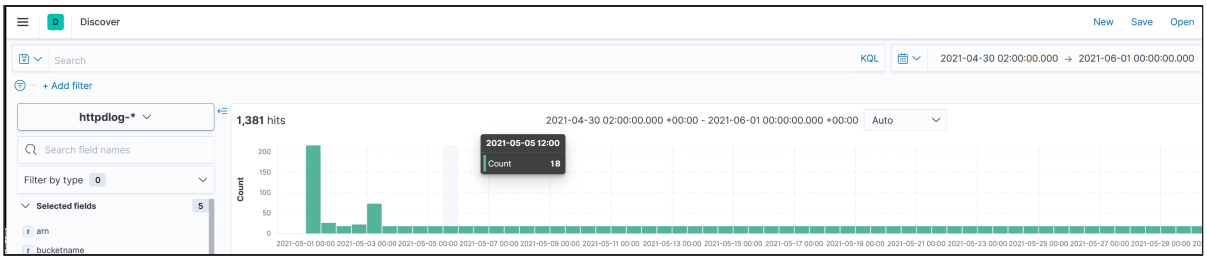
eddda341-4a00-4955-b924-cee059bb35db

What was the final event ID that made changes to the bucket policy (no hint)?

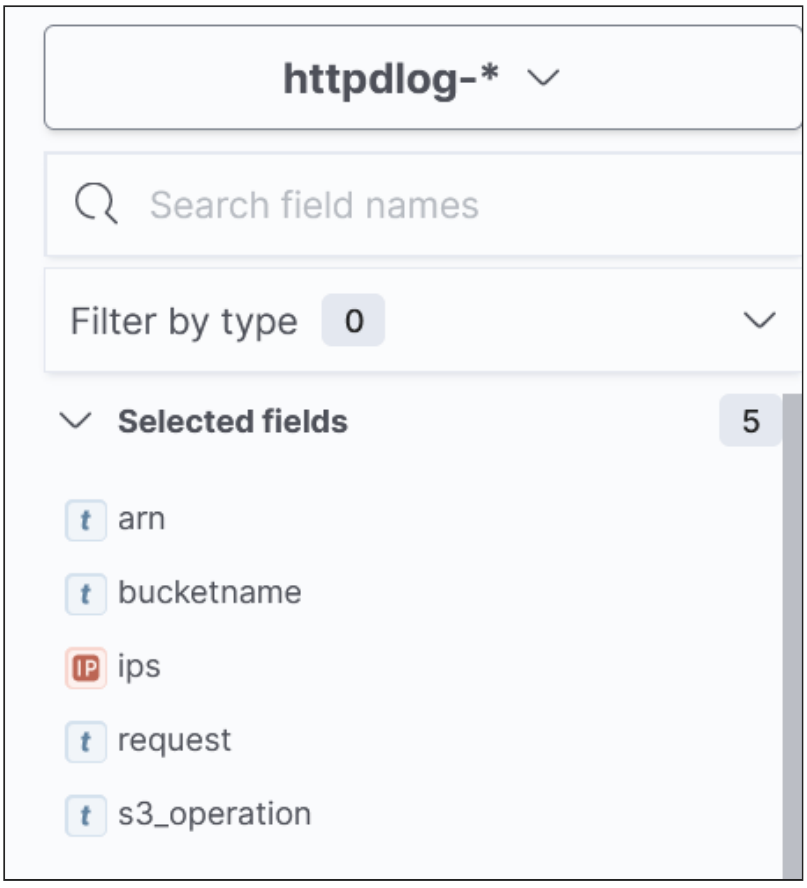
c6664ccf-4384-4172-b86e-30827f895ba0

Now that we know when Scott made the bucket public, let's take a look at the server access logs to see who took advantage of his mistake.

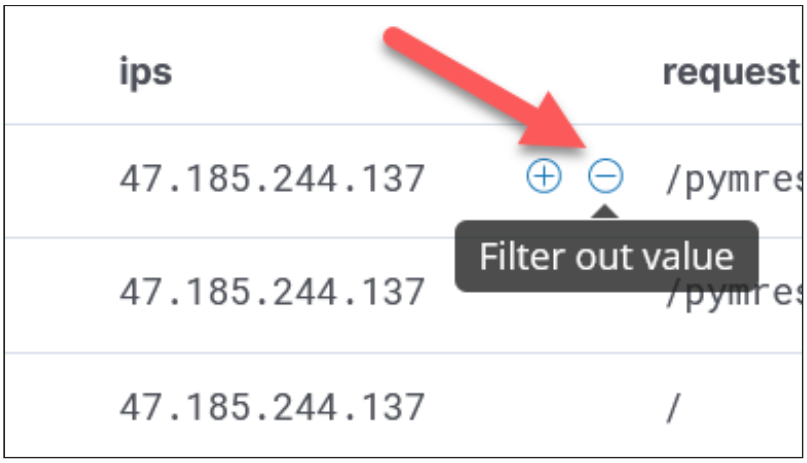
Go back to Kibana and in the Discover tab, switch to the httpdlog index. Set your date range to: `2021-04-30 00:00:00.000 +00:00` to `2021-06-01 00:00:00.000 +00:00` You should have 1,381 records return.



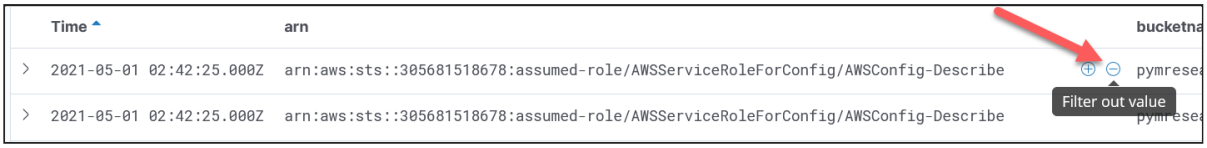
- Let's add the following fields by clicking on the blue plus signs on the left side:
  - arn
  - bucketname
  - ips
  - request
  - s3_operation



We are looking for access that occurred outside of Pymtech so let's exclude the company's IP of 47.185.244.137. Do that by clicking on the minus sign next to the IP value.



You should now have 1,169 hits. We now see that the AWSConfig service has been making a lot of accesses. Let's exclude the ARN by again clicking on the minus sign in the field name.



This brings us down to 20 hits.

Looking at the 20 hits, answer the following questions.

What does an ARN of - mean?

Hint

"-" means null or no entry

Answer

No authentication. This is a public access

What files were accessed with ARN's of -?

> **Hint**
>
> Look at the request column

> **Answer**
>
> ***Pympacket.doc, Pympacket.pdf, Pympacket.vsd, PymProtocol.vsd, PymTech.c, and PymWrapper.py***

What does the operation REST.GET.OBJECT mean (no hint)?

A file is being downloaded

What country did the access occur from?

> **Hint**
>
> Expand the record and look for geo fields

> **Answer**
>
> ***Russia***

Well that's not good. Looks like someone did find the public bucket and the PymPacket project was taken! We will have to see if this threat actor was anywhere else in the AWS tenant. Scott is letting us know that PymPacket is running on a Kubernetes cluster in our tenant.

> **Warning**
>
> Make sure to clear your filters before moving on to the next lab

Key Takeaways

- You now can determine when a bucket policy was modified.
- You can now determine what policy changes were being made.
- You can now determine what files are being downloaded from an S3 bucket and by whom.

Lab 3.5: Tracking Lateral Movement

Objectives

- Learn what AWS services attackers focus on to expand their access.
- Learn what AWS services can be leveraged for lateral movement.
- Determine the impact of a single compromised account.

Background

From the AWS VPC Flow Logs lab, we know that a host was compromised and data was exfiltrated out. In many instances, this is accomplished when API keys are stolen and an attacker looks to move laterally within the cloud tenant. We need you to determine where this attack came from and what was accessed.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-04-30 23:30 UTC** to **2021-05-03 23:30 UTC**.

Load Data

We already extracted these logs in Lab 3.4, so you should be good to go if you completed Lab 3.4.

Lab Content

Time Frame

Based on Scott's recollection of when he made changes, set your time frame as follows: `2021-04-30 23:30:00.000 +00:00` to `2021-05-03 23:30:00.000 +00:00`

Tracking Lateral Movement

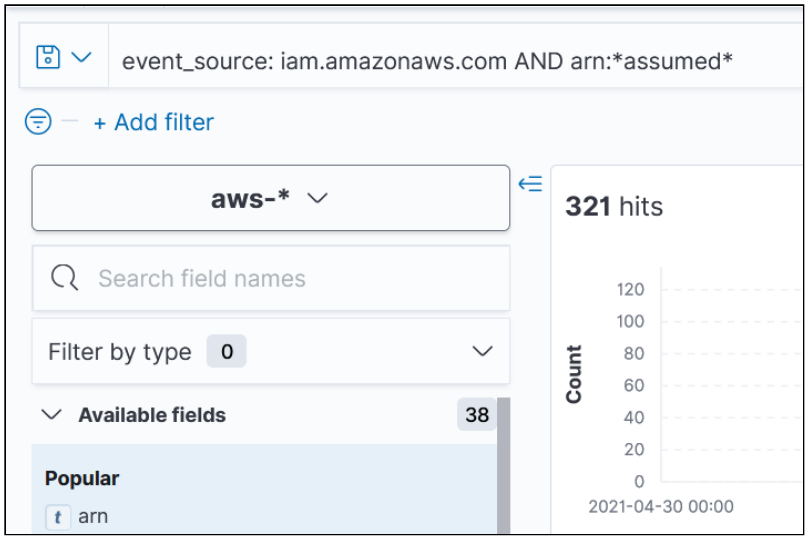
Start in the Discover tab

Set your index to `aws-*`

One of the most common ways attackers get ahold of keys is to compromise a host and steal its IAM role. To find out if this happened here, let's check our IAM logs for accesses from assumed-roles.

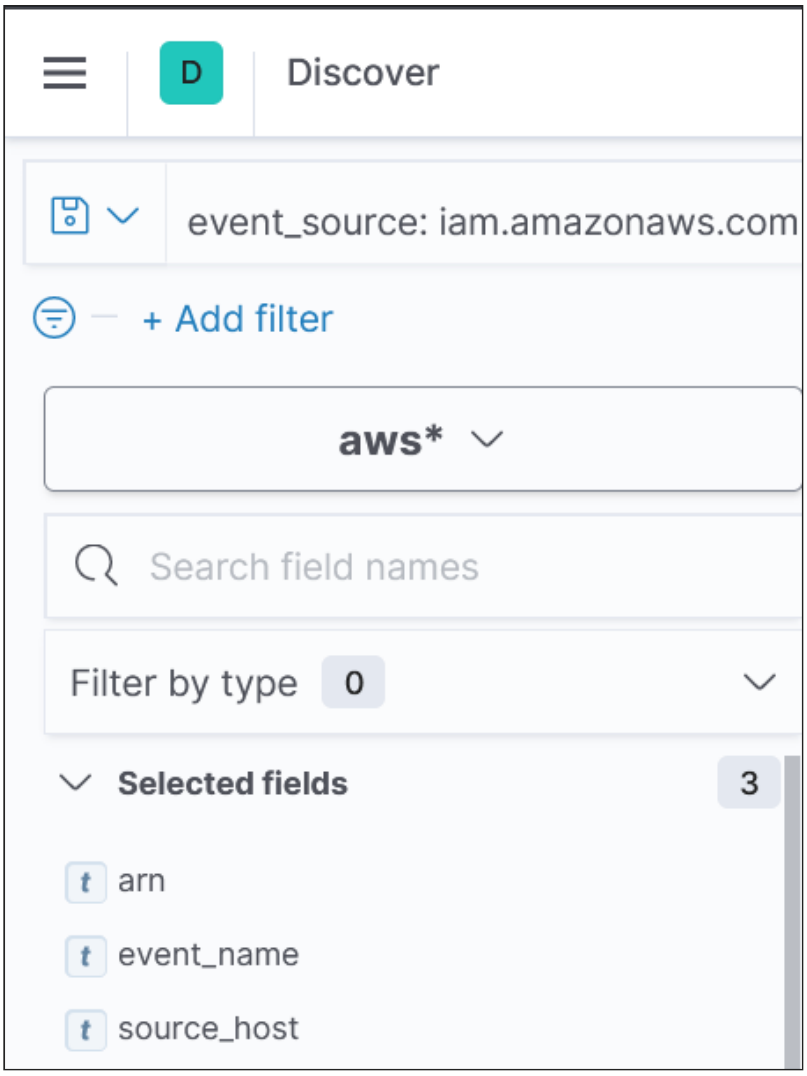
1. Filter for the event_source of iam.amazonaws.com and an assumed role in the ARN.

```
event_source: iam.amazonaws.com AND arn:*assumed*
```



You should have 321 hits.

Select three fields to focus on what was happening here: arn, event_name and source_host. Remember that you can select fields by moving your mouse over them on the available fields list on the left and clicking the plus sign.



2. Scroll down and look for a source_host that isn't from Amazon.

What host did you find other than Amazon?

Hint

Try a filter like event_source: iam.amazonaws.com AND arn: *assumed* AND NOT source_host: *amazon*

```
event_source: iam.amazonaws.com AND arn: *assumed* AND NOT source_host: *amazon*
```

Answer

5.62.19.62

What country does that IP belong to?

Hint

Try a website like https://www.ip2location.com/demo

Answer

Russia

What role did the attacker compromise?

Hint

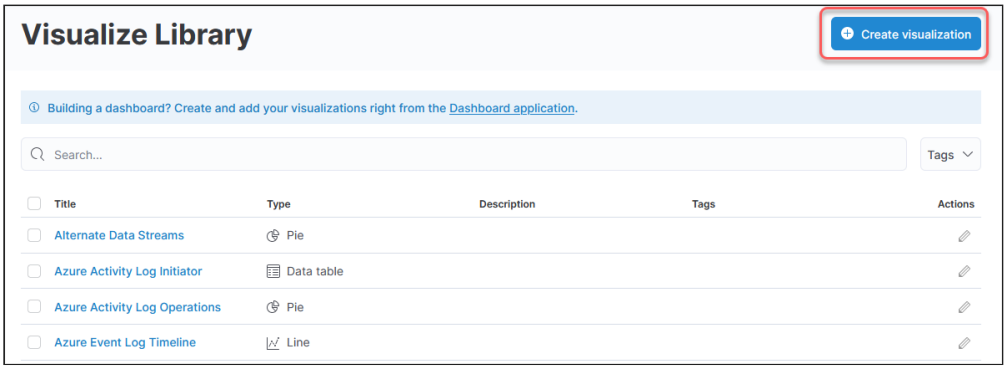
Look at the ARN field

Answer

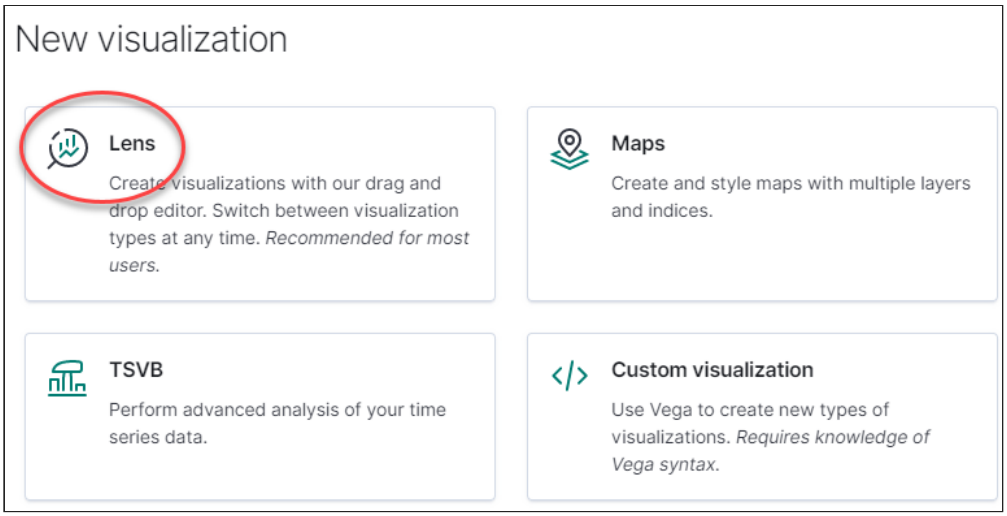
eksctl-dev-cluster-nodegroup-stan-NodeInstanceRole-1VPQKPTS212T6, which is the IAM role created for Pymtechlabs' Kubernetes cluster

3. Now that we have an IP let's find out what other Amazon APIs it was accessing. Click on the Visualize Library screen again, but this time we are going to create a visualization.

   Click on Create Visualization.

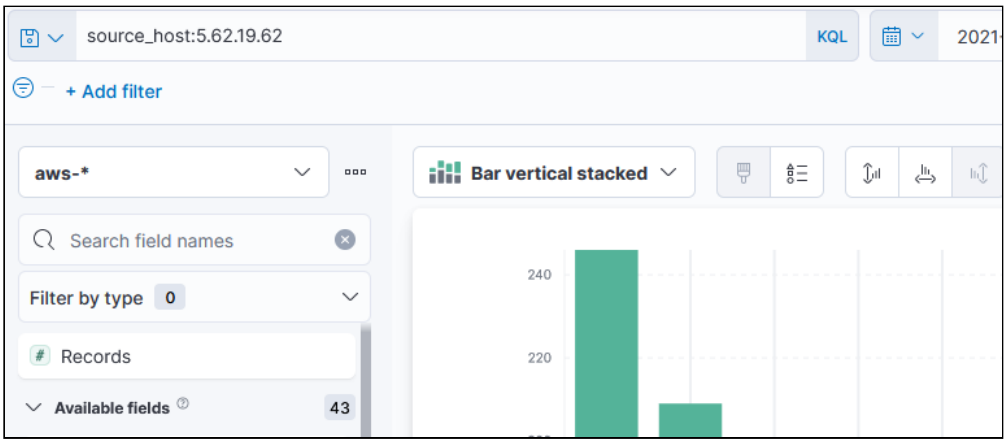


   Choose the Lens visualization.

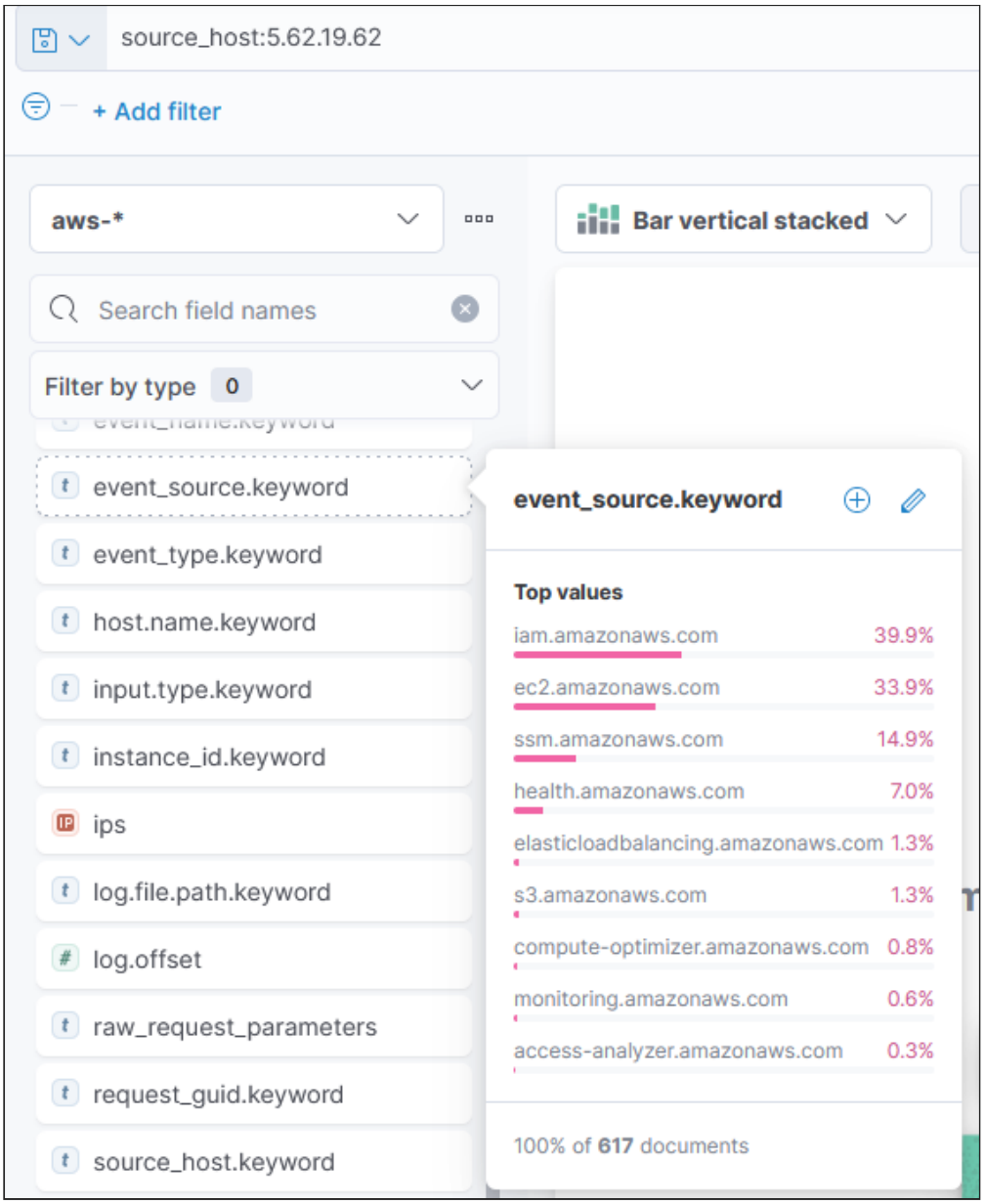


Add a filter for our IP at the top and make sure the AWS index is selected.

```
source_host:5.62.19.62
```



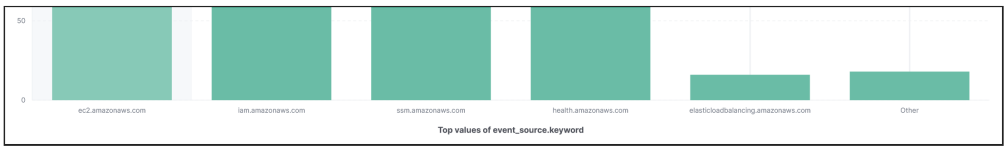
Select event_source.keyword from the available fields and drag and drop it into the middle frame.



You should now see the top event sources (API calls) that our attacker's IP is associated with.

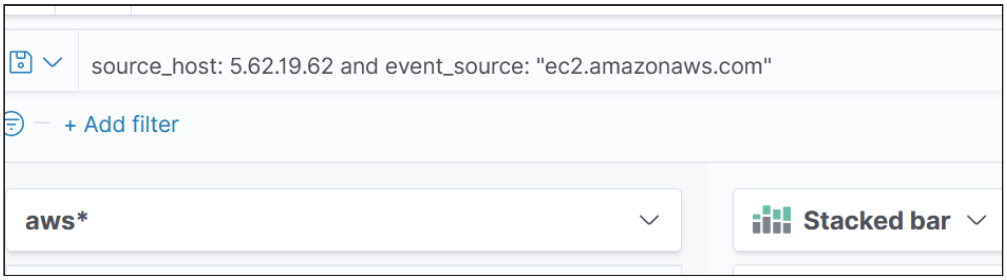
What are the top 3 services (no hint)?

EC2, IAM, and SSM



4. Let's check to see what the most called EC2 commands are by changing our filter to source_host: 5.62.19.62 and event_source: "ec2.amazonaws.com" and then clearing out the prior field selection.

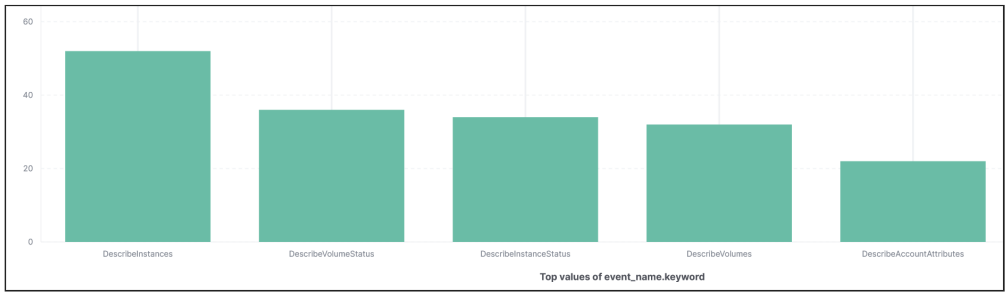
```
source_host:5.62.19.62 AND event_source:"ec2.amazonaws.com"
```



Drag and drop the event_name.keyword field into the middle screen to find the top API events logged for EC2.

What are the top three event names (not including Other) (no hint)?

DescribeInstances, DescribeVolumeStatus, and DescribeVolumes



What is the attacker doing?

Hint

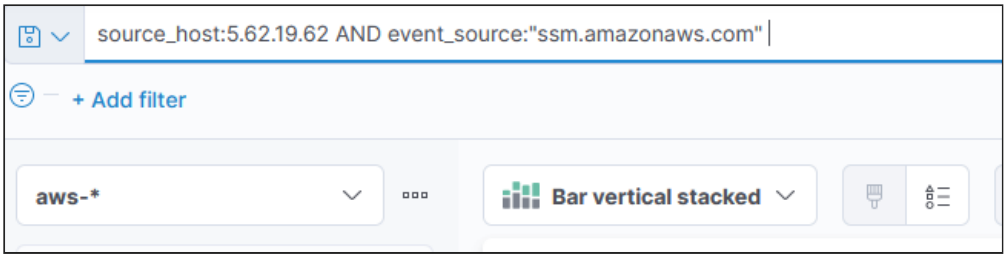
Describe events will list out resources by type

Answer

The attacker is listing out all the EC2 VMs we have in our tenant

5. Now we need to see what the attacker was doing with SSM. SSM allows administrators to perform actions across their VM fleet. Change the event_source filter to ssm.amazonaws.com.

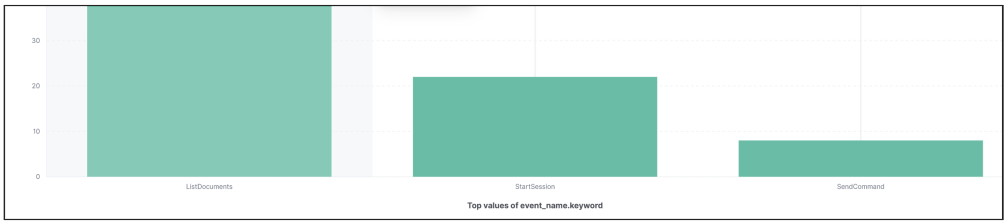
```
source_host:5.62.19.62 AND event_source:"ssm.amazonaws.com"
```



Your graph should change to list three SSM commands executed.

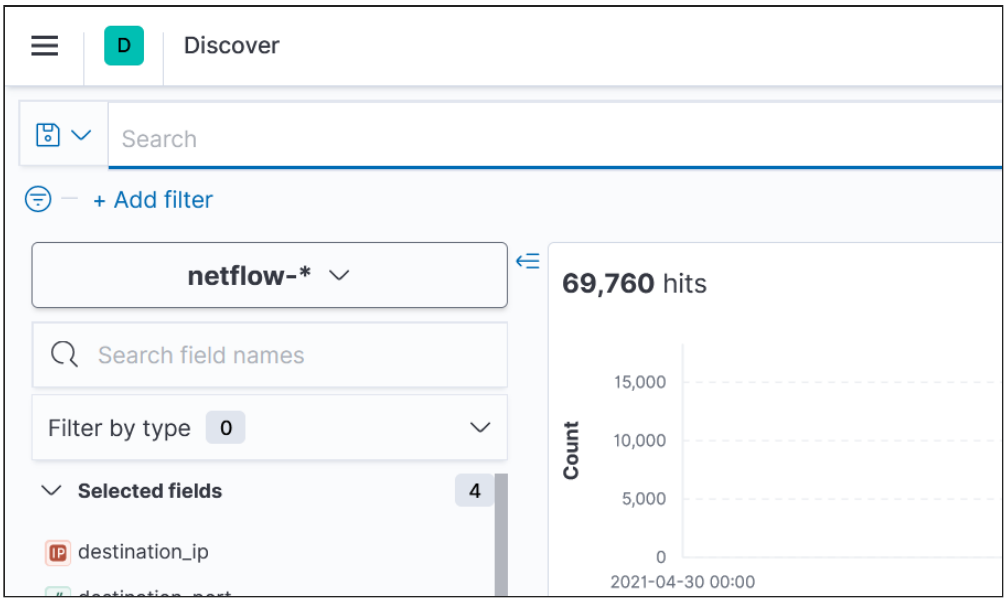
What were the three commands sent via SSM (no hint)?

ListDocuments, StartSession, and SendCommand



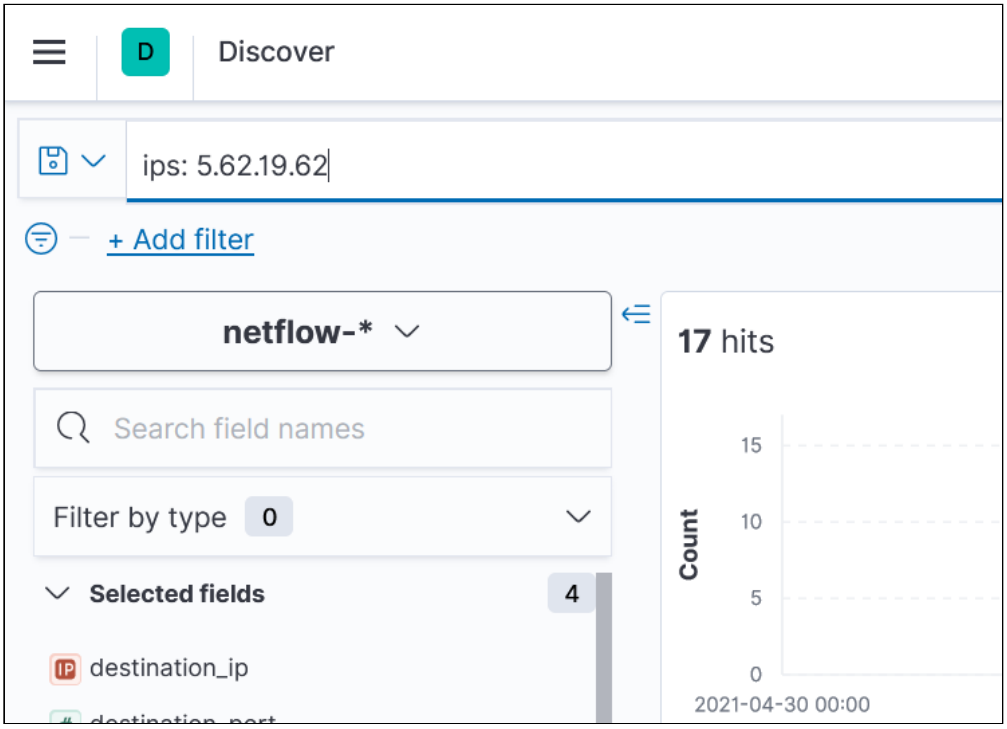
- ListDocuments is part of SendCommand. It allows an authorized user (or attacker) to send commands to EC2 VMs and have them executed as root.
- StartSession will open a command shell on the remote EC2 VM, without the need for a password or SSH key. Looks like our attacker got access to an IAM role and then moved over to executing commands on our EC2 system!

6. If our Kubernetes cluster was compromised can we find any evidence when it occurred? Without access to the underlying container (which may be gone now) we can always check VPC Netflow! Let's switch to the Netflow index.



Now do a query for our threat actor's IP:

```
ips: 5.62.19.62
```



- Add the following fields to the main view:
  - destination_ip
  - destination_port
  - source_ip

- source_port

Time	source_ip	destination_ip	destination_port	source_port
2021-05-02 22:05:09.000Z	192.168.62.223	5.62.19.62	2374	80
2021-05-02 22:05:09.000Z	5.62.19.62	192.168.62.223	80	2374

What do we see in this traffic (no hint)?

The threat actor's IP is communicating with our AWS tenant on port 80

What normally runs on port 80 (no hint)?

Web servers

What should we do next (no hint)?

If the container has not been deleted, we could examine that. Otherwise what we need to do is find out if any vulnerable versions of applications, web servers or libraries are in use in our Kubernetes cluster.

You will find that many developers use pre-made template clusters from either the Amazon Marketplace or Github projects. These could be deployed via terraform or AWS CloudFormation. In either case they sometimes ship with old versions of applications that contain known vulnerabilities, and many developers don't update the contents of the container before deploying these for public access.

Warning

Make sure to clear your filters before moving on to the next lab

Key Takeaways

- When credentials are exposed proper role based access controls are essential.
- Once an attacker has credentials they can scan access and move between resources quickly.
- SSM will allow an attacker with the proper level of account access to execute commands across your fleet of EC2 VMs.

Lab 4.1: Google Workspace Admin BEC

Objectives

- Review logs from Google Workspace to understand user activity
- Search for unauthorized access and post-compromise actions

Background

Kurt contacted Long Con Security's support desk when his password wouldn't work. He attempted to recover his password, but when he selected "Forgot Password?" an unfamiliar phone number was shown as his recovery contact.

Long Con Security believes Kurt's account has been compromised, and Kurt confirms that the timing of this activity does correspond with an email he recently received from Darren Cross asking for his credentials via an attached form. We need to figure out what happened, starting with the source of the phishing email.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2022-01-24 00:00 UTC** to **2022-02-01 00:00 UTC**.

Load Data

In the setup instructions you loaded a GeoIP database which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the Google Workspace parsers and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

The Google Workspace logs for this lab have already been exported from our Google Workspace admin portal.

The Google Workspace logs have been saved into `/home/elk_user/lab-4.1_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials
    - Username: ***elk_user***
    - Password: ***forensics***

    You may log on to the console; however, it's recommended to log in via SSH.

2. Extract all the `.json` files from the preceding ZIP file into the Logstash folder for parsing by FileBeats with the following command:

Command lines

```
cd ~
unzip lab-4.1_source_evidence.zip -d /logstash/gws/
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you have 1,157 documents loaded in the `gws` index:

Command lines

```
sof-elk_clear.py -i list
```

Expected output

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- gws (1,157 documents)
- httpdlog (1,381 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

5. Once it is complete, the data will be available in the `gws-*` index.

Time Frame

`2022-01-24 00:00:00.000 +00:00` to `2022-02-01 00:00:00.000 +00:00`

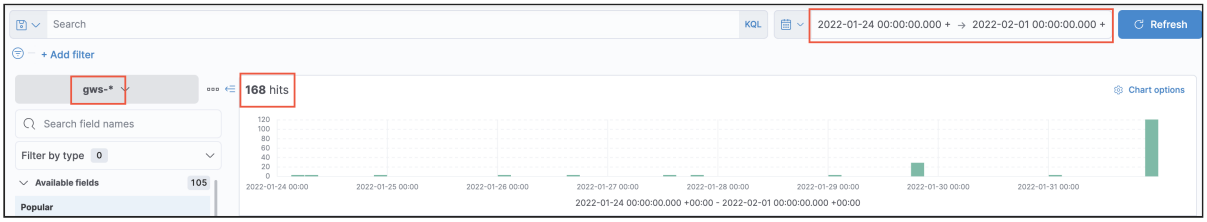
Search for Evidence of a Phishing Attempt

Start in the Discover Tab

Set your index to `gws-*` and apply the time frame provided above.

Be sure to clear any prior filters.

You should see `168` hits. If you don't, make sure you have selected the correct index and time range.



1. First, let's investigate the suspected phishing email.

Who is the sender of the phishing email?

Hint

Search for

```
recipient: "kurt@longconsecurity.com" AND attachment_count > 0
```

(Kurt mentioned opening an attachment)

Add `sender` and `subject` as selected fields.

Answer

After applying these filters a parseable number of results exist. Look at the subject and sender to identify the suspicious email—in this case, a supposed security alert from a Proton Mail account impersonating Darren Cross.

The sender of the phishing email is `mr_dcross@protonmail.com`.

2. Let's see who else might be targeted by this threat actor.

Who else received an email from the fake Darren Cross?

Hint

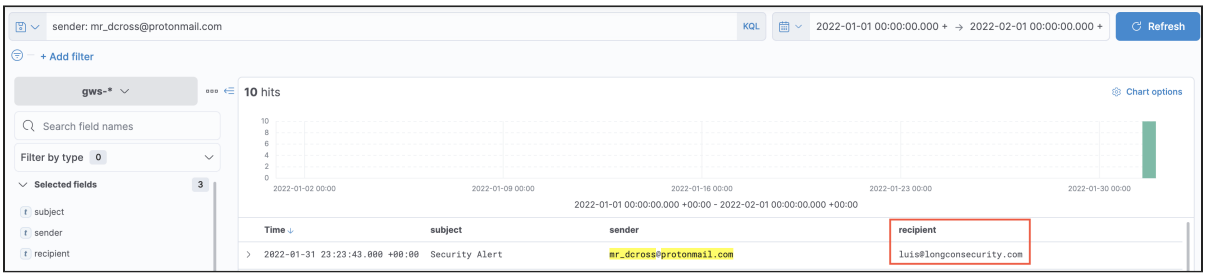
Search for

```
sender: mr_dcross@protonmail.com
```

Add `recipient` as a selected field.

Answer

The only other recipient of an email from this sender is `luis@longconsecurity.com`.



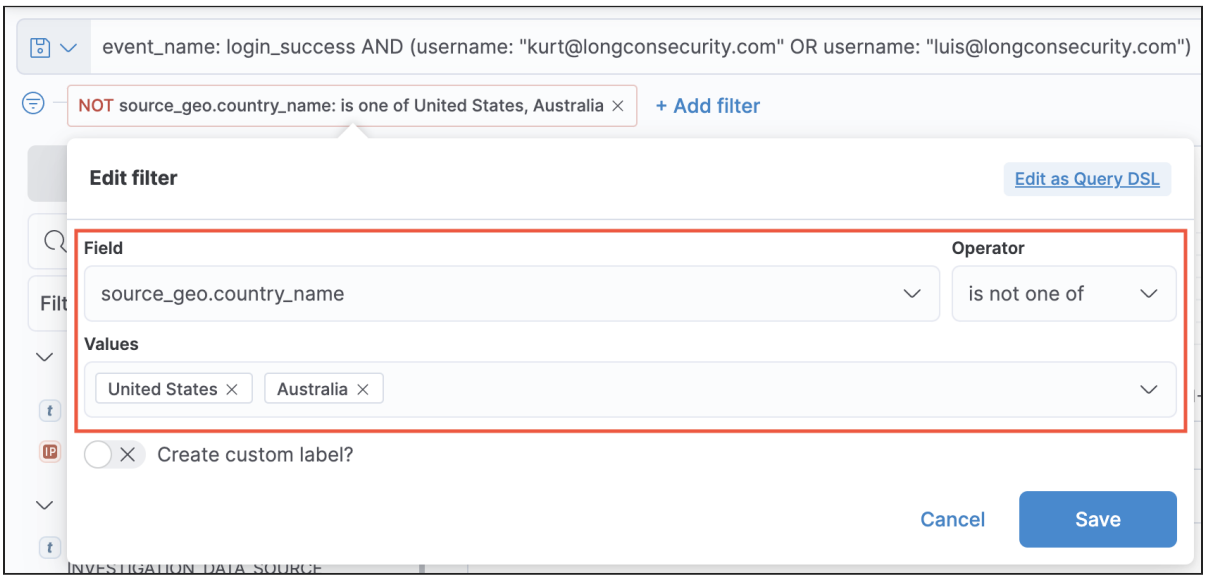
Identify Suspicious Login Activity

1. Next we want to identify any suspicious login activity using these accounts. Search for `event_name: login_success AND (username: "kurt@longconsecurity.com" OR username: "luis@longconsecurity.com")` to limit results to login activity involving our potential victim users and ensure the correct time range is set.
2. We know Kurt works in the United States and Luis works in Australia, and they do not travel for their jobs, so let's start by looking for suspicious logins based on location. Add `source_geo.country_name` to the selected filters with condition `is not one of` and add `United States` and `Australia` to the value list. This can also be done by appending `AND NOT (source_geo.country_name: "United States" OR source_geo.country_name: "Australia")` to your search.

From which country did the suspicious login occur?

Hint

Filter out US- and Australia-based activity by adding the filter shown below based on the criteria discussed above:



Answer

The source country of the login is `Hong Kong`.

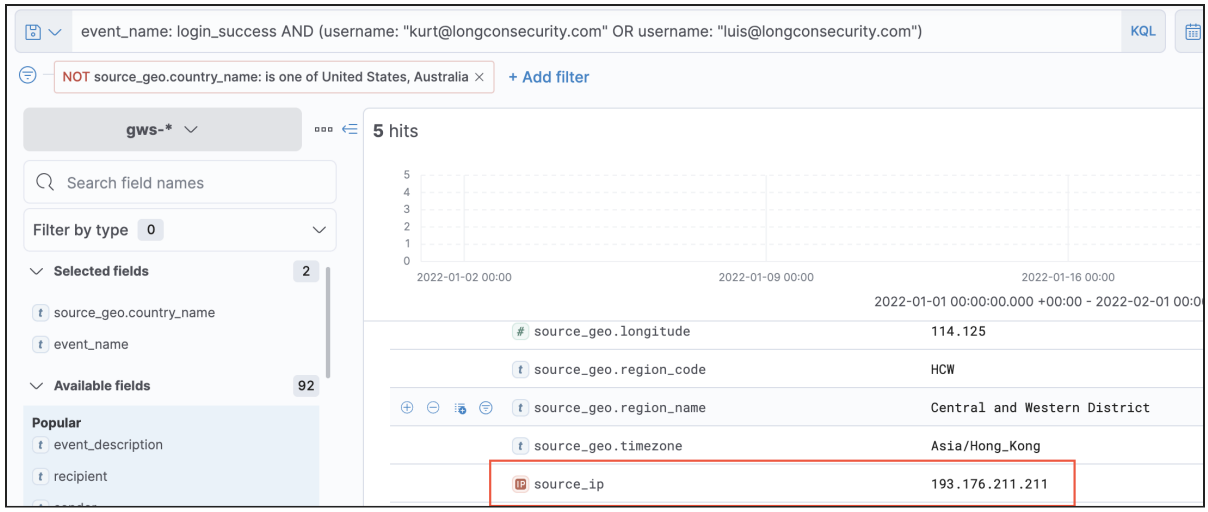
What is the source IP address of the suspicious login?

Hint

Look at the `source_ip` field associated with the login.

Answer

The source IP address is `193.176.211.211`.



Identify Post-Compromise Activity

1. To identify what the threat actor did after logging in, apply the following filter: `source_ip: 193.176.211.211`.

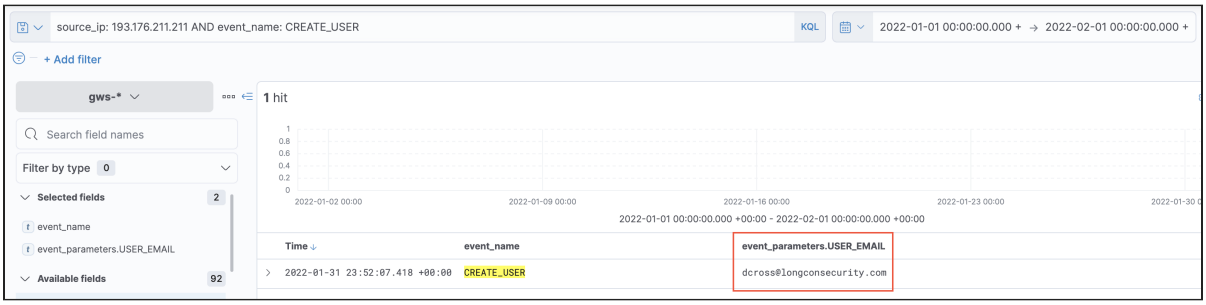
To establish persistence, the threat actor created a new account. What is new account's email?

Hint

```
source_ip: 193.176.211.211 AND event_name: CREATE_USER
```

Answer

The `event_parameters.USER_EMAIL` field indicates the new email address is `dcross@longconsecurity.com`.



2. The threat actor likely changed permissions or settings associated with this account to give more access. Find events involving this account using the following filter: `event_parameters.USER_EMAIL: "dcross@longconsecurity.com"`

What group email was the dcross account added to?

Hint

```
event_parameters.USER_EMAIL: "dcross@longconsecurity.com" AND event_name: ADD_GROUP_MEMBER
```

Answer

The `event_parameters.GROUP_EMAIL` field indicates the user was added to the `gcp-organization-admins@longconsecurity.com` group account.

What role was assigned to the dcross account?

Hint

```
event_parameters.USER_EMAIL: "dcross@longconsecurity.com" AND event_name: ASSIGN_ROLE
```

Answer

The `event_parameters.ROLE_NAME` field indicates the user was assigned the `_SEED_ADMIN_ROLE` role. This is notable because "_SEED_ADMIN_ROLE" is the role assigned to super admins.

3. Threat actors can add a forwarding address to email accounts so that copies of every email received by the victim are sent to an attacker-controlled inbox. This means that even if access to the organization is lost, the threat actor still has visibility over some of the email data.

What email address was added as a email forwarding destination?

Hint

```
event_name: email_forwarding_out_of_domain
```

Answer

The `event_parameters.email_forwarding_destination_address` field indicates that `mr_dcross@protonmail.com` was added as a forwarding address.

4. Lastly, we want to identify whether any sensitive files in Google Drive were exposed.

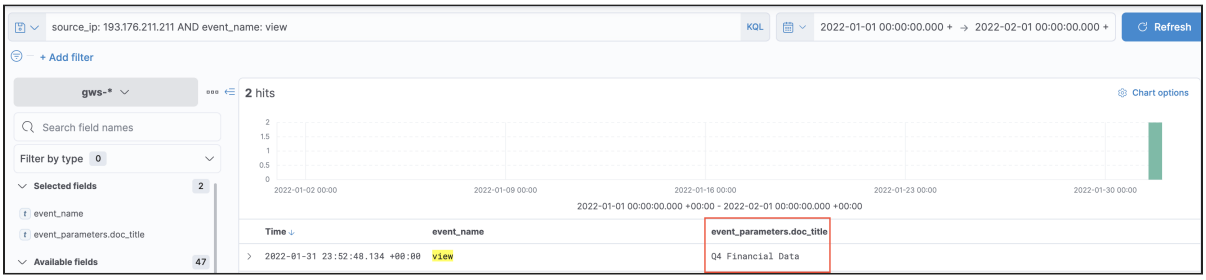
What file was accessed by the threat actor?

Hint

```
source_ip: 193.176.211.211 AND event_name: view
```

Answer

The `event_parameters.doc_title` field indicates a file named `Q4 Financial Data` was accessed.



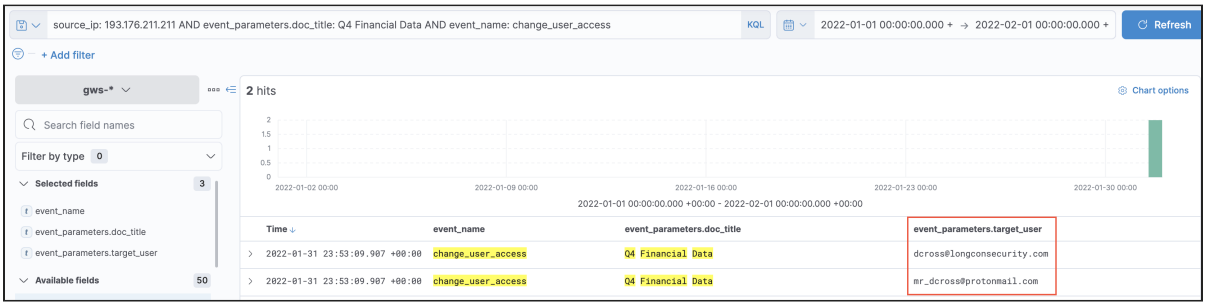
What two email addresses was this file shared with?

Hint

```
source_ip: 193.176.211.211 AND event_parameters.doc_title: Q4 Financial Data AND event_name:
change_user_access
```

Answer

The `event_parameters.target_user` field in the two relevant logs indicate the file was shared with `dcross@longconsecurity.com` and `mr_dcross@protonmail.com`.



Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- The `kurt@longconsecurity.com` account was compromised due to a successful phishing attempt.
- Abnormal login behavior could be detected based on the geolocation of the source IP.
- The attacker attempted to gain persistence by creating a new account (and providing it admin privileges) and setting up email forwarding.
- Google Drive audit logs provide insight into what files were viewed, edited, and shared.

Lab 4.2: OAuth Abuse with Third-Party Apps

Objectives

- Review Token logs from Google Workspace to observe activity occurring via a third-party app
- Identify permissions being granted and used via OAuth
- Identify activity occurring via a third-party app across email and Drive audit logs

Background

A recent audit of OAuth tokens issued to third-party applications revealed a potentially suspicious application that obtained permissions to an admin account. We need to review the logs to find the details of the permissions granted to the application and how they were leveraged.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2022-03-06 00:00 UTC** to **2022-03-08 00:00 UTC**.

Load Data

In the setup instructions you loaded a GeoIP database which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the Google Workspace parsers and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

In Lab 4.1, you loaded the Google Workspace Audit logs into SOF-ELK.

If you haven't completed either of these steps, please go back and do so now.

Time Frame

`2022-03-06 00:00:00.000 +00:00` to `2022-03-08 00:00:00.000 +00:00`

OAuth Token Abuse

Start in the Discover Tab

Set your index to `gws-*` and apply the time frame provided above.

Be sure to clear any prior filters.

You should see `56` hits. If you don't, make sure you have selected the correct index and time range.

1. First, let's identify information regarding the malicious application and the permissions it was granted. For context, Google Chrome appears in the token logs legitimately during OAuth activity. Additionally, the application named "Log Collection" is an application managed and used by Long Con Security, so it is trusted.

What is the name of the malicious application? Which account was targeted, and what permissions were granted?

Hint

```
event_name: authorize
```

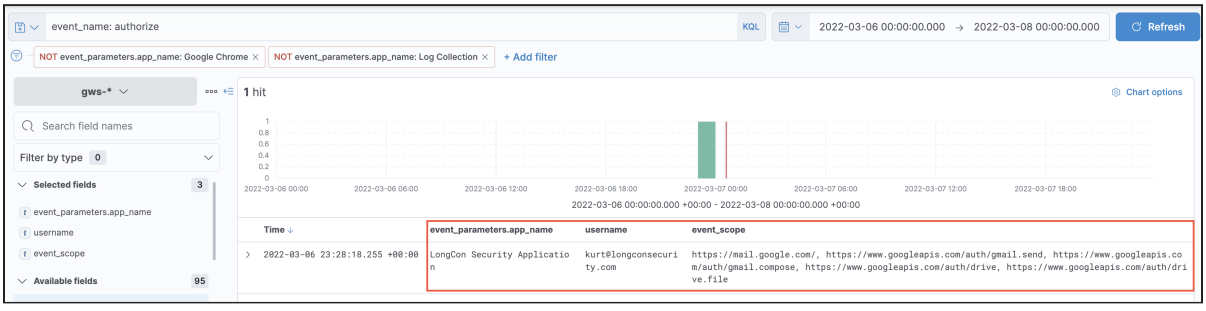
Add `event_parameters.app_name`, `username` and `event_scope` as selected fields. Filter out `Log Collection` and `Google Chrome` since we know those are legitimate.

Answer

After applying these filters we can see the authorization event in the token logs.

An application named `LongCon Security Application` received the following permissions for the account with username `kurt@longconsecurity.com`:

- `https://mail.google.com/`
- `https://www.googleapis.com/auth/gmail.send`
- `https://www.googleapis.com/auth/gmail.compose`
- `https://www.googleapis.com/auth/drive`
- `https://www.googleapis.com/auth/drive.file`



2. Next, we should review what the application used the granted permissions for.

What is the first action associated with the application?

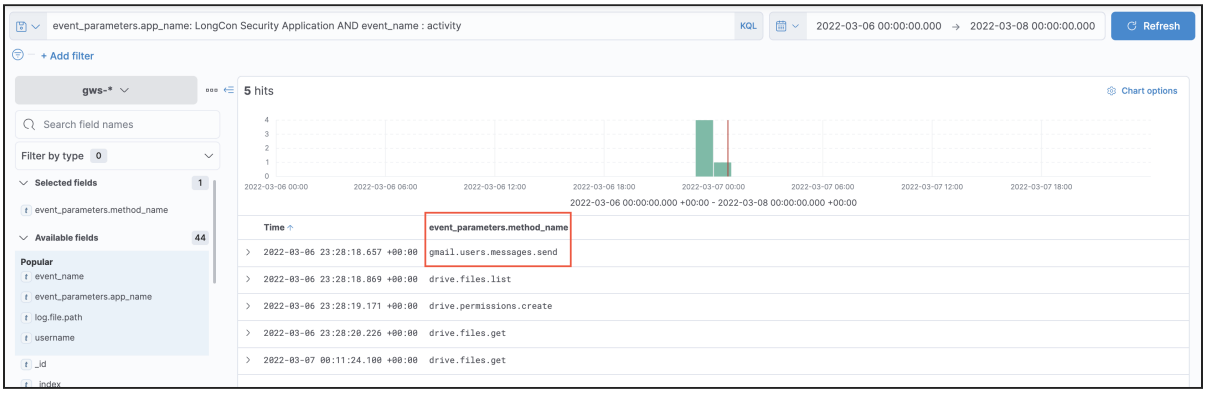
Hint

```
event_parameters.app_name: LongCon Security Application AND event_name : activity
```

Add `event_parameters.method_name` as a selected field.

Answer

The first event occurring after the application was granted applications has a method name of `gmail.users.messages.send`. As the name implies and as can be found in Gmail API documentation, this method is used to send a message to a specified recipient. The documentation on this API endpoint can be found here: https://developers.google.com/gmail/api/reference/rest/v1/users.messages/send



3. Now that we know the first action performed by the application was sending an email, we should attempt to find the email that was sent by the application.

What is the subject line of the email sent using the above method name?

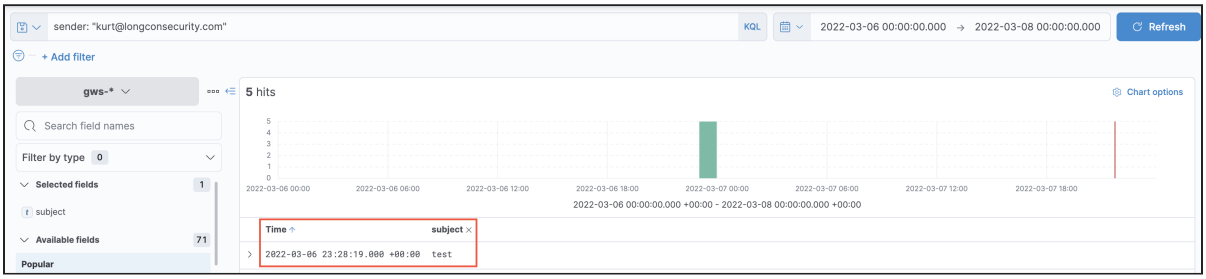
Hint

```
sender: "kurt@longconsecurity.com"
```

Add `subject` as a selected field. Look for email(s) occurring right after the token audit log event discovered above (which had a timestamp `2022-03-06 23:28:18.657 +00:00`).

Answer

The timestamp for the token log was `2022-03-06 23:28:18.657 +00:00`. In the email logs there are entries for an email with the subject line `test` being sent within milliseconds of that. Looks like the threat actor was testing whether the permissions were working as intended.



4. Let's continue our investigation by looking at the other actions carried out by the application.

What is the second Google Workspace application that was interacted with by the third-party application?

Hint

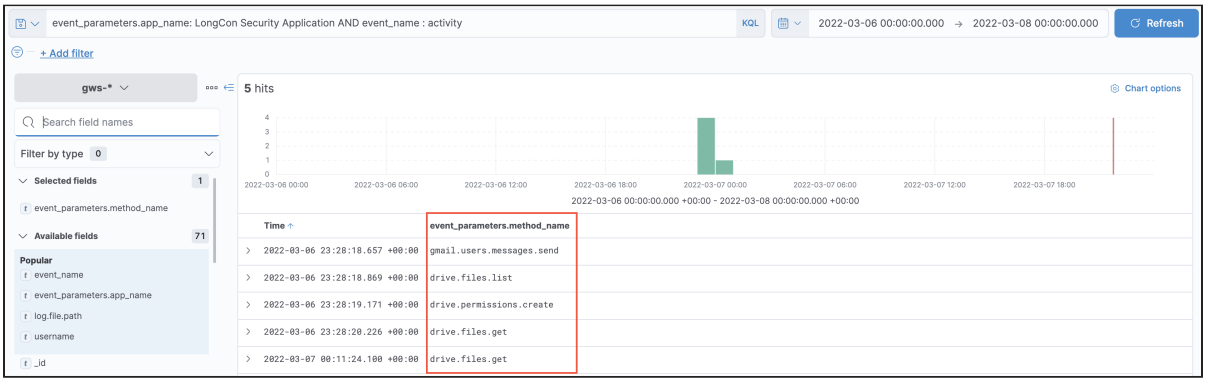
As with the previous search for application activity, search for

```
event_parameters.app_name: LongCon Security Application AND event_name : activity
```

Add `event_parameters.method_name` as a selected field.

Answer

The rest of the methods called by the `LongCon Security Application` are associated with Google Drive. They include `drive.files.list`, `drive.permissions.create`, and `drive.files.get`. Make note of the timestamps so we can correlate activity in other logs. The activities occurred between `2022-03-06 23:28:18.869 +00:00` and `2022-03-07 00:11:24.100 +00:00`.



Let's uncover further details of the Google Drive activity that was performed by the third-party application. Unfortunately, there is nothing in Google Drive logs that will show what files were revealed to the threat actor. However, per the documentation for the files.list method (https://developers.google.com/drive/api/v3/search-files), if no parameters were provided, all files and folders within Kurt's drive would be returned. It is impossible to know if the threat actor modified the parameters though based on what Google shows in the audit logs.

1. The threat actor created a shared link for a file called `RFP for New Client` in order to gain external access to access the file themselves.

At what date & time was the `RFP for New Client` file downloaded by an external user?

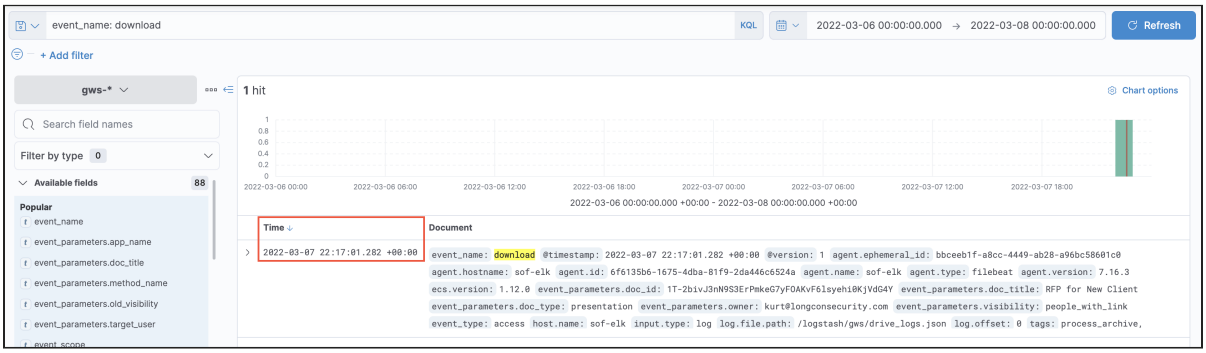
Hint

```
event_name: download
```

Answer

There is one `download` event returned for the specified timeframe. The event occurred at `2022-03-07 22:17:01.282 +00:00`.

The username field associated with this event is empty, indicating an anonymous user accessed the file, not a corporate account.



Unless the OAuth token is revoked, the malicious application could be used to perform any activities within the capabilities of the specified scopes, which includes interacting with Gmail and Google Drive. Upon discovering such activity, tokens should be revoked ASAP. Details of how to revoke tokens can be found in the course content.

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Token audit logs provide information about permissions given to third-arty applications via OAuth.
- Users can be socially-engineered into providing permissions to third-party applications.
- Token abuse activity should be remediated by revoking access to malicious tokens ASAP.

Lab 4.3: Google Workspace Data Exposure

Objectives

- Review Drive logs from Google Workspace to understand Google Drive activity
- Search for permissions changes and exposure of corporate data

Background

Long Con Security leverages Google Drive for editing and sharing files at the organization. Recently, concerns have been raised that employees are being careless with file permissions and, thus, potentially exposing sensitive company data to unauthorized users. We have been asked to audit recent Google Drive activity to identify how users are sharing and managing files to determine whether additional policies need to be put in place.

Preparation

To prepare:

1. Make sure your SOF-ELK VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2022-02-13 00:00 UTC** to **2022-02-25 00:00 UTC**.

Load Data

In the setup instructions you loaded a GeoIP database which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the Google Workspace parsers and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

In Lab 4.1, you loaded the Google Workspace Audit logs into SOF-ELK.

If you haven't completed either of these steps, please go back and do so now.

Time Frame

`2022-02-13 00:00:00.000 +00:00` to `2022-02-25 00:00:00.000 +00:00`

Document Sharing

Start in the Discover Tab

Set your index to `gws-*` and apply the time frame provided above.

Be sure to clear any prior filters.

You should see `191` hits. If you don't, make sure you have selected the correct index and time range.

1. First, let's look at a sample document to see what recent sharing activity has occurred. "Q4 Financial Data" is a document with sensitive financial information, so let's see how that has been shared.

Who was the Q4 Financial Data document shared with?

Hint

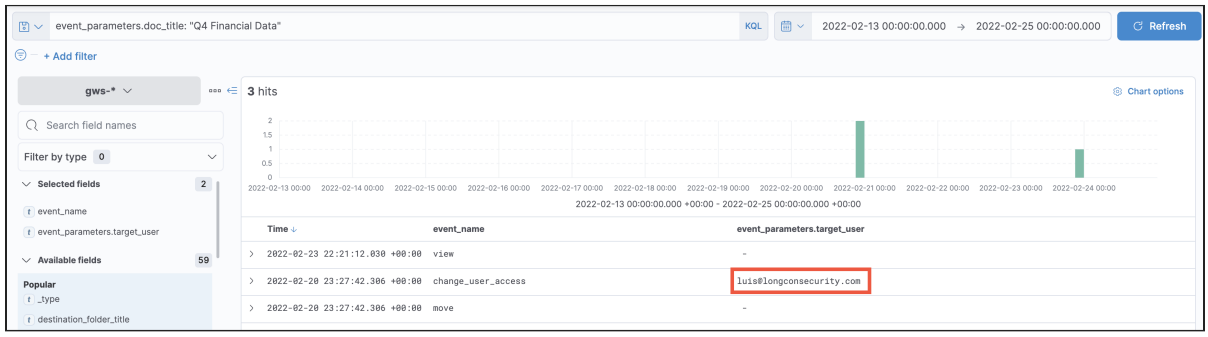
```
event_parameters.doc_title: "Q4 Financial Data"
```

Add `event_name` and `event_parameters.target_user` as selected fields.

Answer

After applying these filters we can see an event named `change_user_access`.

The target user associated with this event is `luis@longconsecurity.com`.



2. A bigger risk is files being shared outside the organization. Let's investigate files being shared to external users.

Which document was shared to an external user account and with whom?

Hint

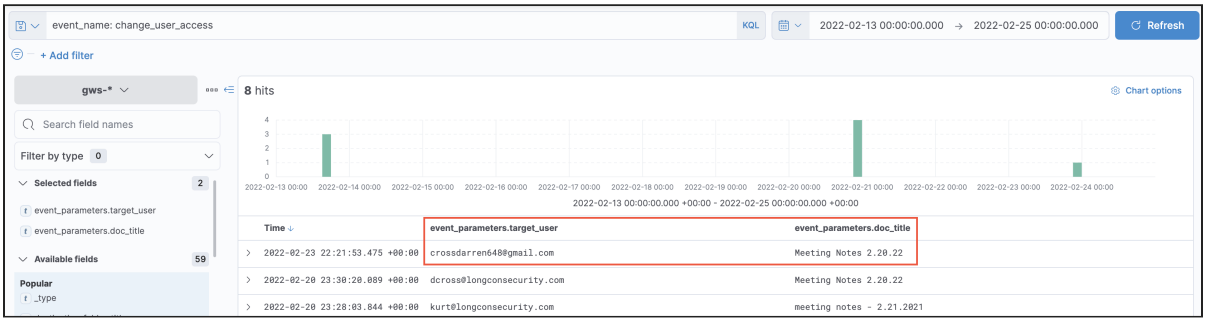
```
event_name: change_user_access
```

Add `event_parameters.doc_title` and `event_parameters.target_user` as selected fields.

Answer

After applying these filters there is one event with a `event_parameters.target_user` not belonging to longconsecurity.com.

The target user associated with this event is `crossdarren648@gmail.com` and the file name is `Meeting Notes 2.20.22`. Looks like Darren Cross was sharing files to his personal account.



After sharing the file with himself, Darren edited the file from the personal account. What is the timestamp associated with the first edit event?

Hint

```
username: crossdarren648@gmail.com
```

Add `event_name` and `event_parameters.doc_title` as selected fields.

Answer

After applying these filters the first event with an `event_name` of `edit` has a timestamp of `2022-02-23 22:22:32.818 +00:00`.

3. During an audit of files being shared externally, it was uncovered that a presentation named `RFP for New Client` was shared with edit access to anyone with a created link. We need to identify when changes were made to the file and whom.

When was the RFP for New Client presentation first edited by someone other than Kurt (the owner of the file)?

Hint

```
event_parameters.doc_title: "RFP for New Client" AND NOT username:kurt@longconsecurity.com
```

Answer

The timestamp on the earliest event is `2022-02-23 22:51:30.862 +00:00`.

Who edited the aforementioned file? (No hint)

Answer

The `username` field is empty. This means that whoever used the shared link was not logged into a Google account when editing the file.

4. Long Con Security employees store client-related files in a shared drive called "Clients." Let's gather some information about this shared drive and its exposure.

Who created the shared drive?

Hint

```
event_parameters.doc_title: Clients and event_name: create
```

Add `username` as a selected field.

Answer

The user associated with this event is `kurt@longconsecurity.com`.

Who did this user include as members of the shared drive?

Hint

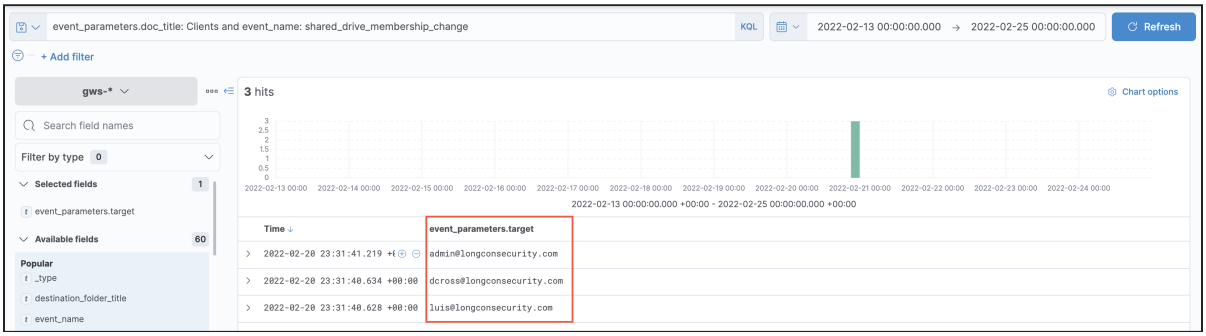
```
event_parameters.doc_title: Clients and event_name: shared_drive_membership_change
```

Add `event_parameters.target` as a selected field.

Answer

Kurt added three users to the shared drive:

- `admin@longconsecurity.com`
- `dcross@longconsecurity.com`
- `luis@longconsecurity.com`



5. Google also provides the capability to publish files to the web via a shareable link or as an embedded object. Let's determine whether any files have been potentially exposed via this method.

What file was published to the web?

Hint

```
event_parameters.new_publish_visibility: public_on_the_web
```

Add `event_parameters.doc_title` as a selected field.

Answer

A file named `Sales Presentation` was published publicly.

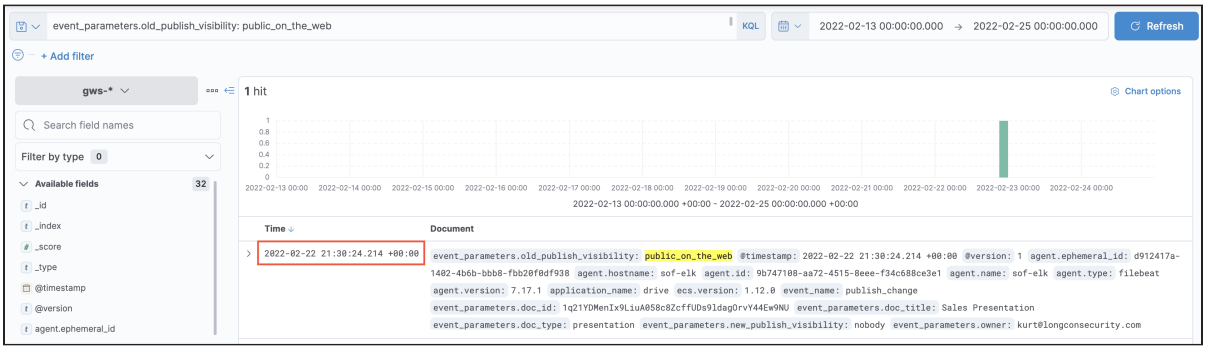
The user who shared the aforementioned file realized his mistake and disabled sharing. What is the timestamp associated with this event?

Hint

```
event_parameters.old_publish_visibility: public_on_the_web
```

Answer

The timestamp on the associated event is `2022-02-22 21:30:24.214 +00:00`.



Google Takeout

1. When Google Takeout is used to export data, a folder and the associated files are created in Drive, leading to events in the audit log. Let's find out if any recent Takeout requests were made.

Who created a Takeout export?

Hint

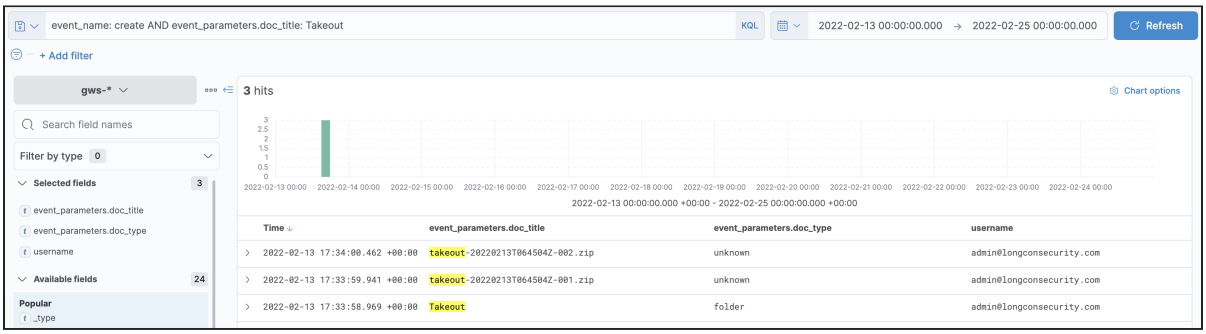
```
event_name: create AND event_parameters.doc_title: Takeout
```

Add `event_parameters.doc_title`, `event_parameters.doc_type`, `username` as selected fields.

Answer

After applying these filters we can see an event related to the creation of a folder called `Takeout`.

The username associated with this event is `admin@longconsecurity.com`.



What time was the first Takeout ZIP archive downloaded?

Hint

```
event_name: download AND event_parameters.doc_title: takeout
```

Add `event_parameters.doc_title` as a selected field.

Answer

After applying these filters we can see a two events involving the download of a Takeout-related file.

The event related to the download of file `takeout-20220213T064504Z-001.zip` has a timestamp of `2022-02-24 01:40:25.296 +00:00`.

Warning

Make sure to clear out any filters you may have applied for this lab.

Key Takeaways

- Google Drive audit logs provide insight into what files were viewed, edited, and shared.

- Users can unintentionally or purposely share files with sensitive content.
- Google Takeout will provide an account with an entire export of their data, allowing them to take potentially sensitive content out of the account.

Lab 4.4: Collecting Workspace Logs in Google Cloud via CLI

Objectives

- Get experience using command-line tools to directly access evidence in the cloud.
- Understand the speed at which logs can be collected using command-line tools.
- Develop the knowledge needed to collect logs directly from Google Cloud.

Background

So far in the course, we've looked at evidence that has already been pulled down from the cloud source of where it was generated. In this lab, we will attempt to collect evidence from Google Cloud directly. As part of setting up the LongConSecurity Google Workspace, the Administrator also enabled logs to be pushed into Google Cloud for collection. We're going to attempt to access those logs.

Your VM already has the `gcloud` tool loaded in it, so you don't need to install any additional tooling. This tool will be discussed further in the Google Cloud section of the class. However, understand it's a universal tool and operates the same on other operating systems.

Please note that this lab will not always have the same outcome as the logs will age out and change over time. The lab is intended as a walkthrough of how to directly access evidence; this lab is not structured with questions and answers.

Preparation

> **Warning**
>
> This lab requires internet access to Google Cloud. If your VM does not have access to the internet then this lab won't work correctly.

To prepare:

1. Make sure your SOF-ELK® VM is running
2. SSH into your VM from a console running on your local system. You will need to SSH to the IP address displayed on the VM after it fully boots up. You will find the IP address in the section that reads:

```
You can SSH to this system at the IP address x.x.x.x
```

1. Perform an update of the SOF-ELK configuration by running the following command after you've logged into SOF-ELK: `sudo sof-elk_update.sh`

Load Data

To allow for collection of data from Google Cloud we're going to need to load in a Service Account's JSON credential file. This will allow you to authenticate against the Google Cloud Project we're using to store data from LongConSecurity's Google Workspace.

Warning

The credentials below are rotated when a new major release of the class comes out. If you are looking at this lab years after you took the class, first of all thanks, but also understand the credentials may no longer work.

1. Log on to the SOF-ELK VM with the following credentials
   - Username: ***elk_user***
   - Password: ***forensics***

   You will need to log in via SSH from your host system. It is not recommended to use the VM console as you will need to copy and paste a lot of commands. Copy and paste is not enabled via the VM console.
2. Create a folder for the lab to hold the JSON Service Account Key file and the log files you're going to collect with the following commands:

   Command lines

   ```
   cd ~
   mkdir lab-4.4
   cd lab-4.4
   ```

3. Copy all of the below code and paste it into the SSH session.

Command lines

```
echo '{
  "type": "service_account",
  "project_id": "class-evidence-collection",
  "private_key_id": "13690599effd0157e4dcc459f97eefdc9e4468ca",
  "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQCvGBOMQhaHu0QZ\n+aEEsvntqcAC3XKj/
oPp6y+ZnaAjFt0o1ygFodLSStj+iV2KHasEhInpSNMy7t7U\nvovWXeFdpzo9TjMvj3cXBk2ZsmWXWssNnGQml4c3Cr/
LNqlkaa5ga3epjm8BP91P\nxddpGAp+uTwgDRb5Q0usCqMj5tbOM1PqO8n6Gw1CXiUkDT/
P1JrCYe1r4bvDD1FO\nO71qZ7dc7QuLul8XYmueGr1eCJZ9hY5oLJEnvdgbd/
Y9pAAV+0Xi1XK6+8j4ekqQ\nAZn3O2FYS9KypKhjUWZs8VW3L+BAzdbnWmB3Ph9d8/8K+EhlCFMNWC13rR9iMm1j\nOyAoVF8DAgMBAAE
kKDdRZkvzLwUZEjUYKBxtg7vdk0qnVN\nM6uAO6eitLRI9iTGD7AZvpblg35abCf0vh19hB+VJuacIfJ3avXWm8gmm/
aGW1L3\n72QxUebNY4DnlrUa2AuZHPRkiagqCJtoKjUZT1HfR6wSEe490kTg4h5KDrtj6MUp\n74lcDU77J2TRBouyg2Bf8NJd/
gWQhujIqvWITEWdiYAaQ7NoSB6kcA3EJSVklJJX\n8ygza+0e6MfYlX1Za1aVco71hGsS69bjbdaAeWS3sQKBgQDVv0VDJaNR6+bHRKVX
yyEYC75mV+DeNaQexyz4qGyFscln2+EuV3Gqkqw4zOXTJPxB5G\ndlOeQbk3983YUAK3yubKJXDr6xgakX/EcSC7qEIG/
5RSanQPefbBqJ4X1k0WAo99\nkrYW86CBo2k/
6WTJpmXEptyl0QKBgQDRtMFxOhb+85iUU9D6alt3s+P6kknxLC5H\niOzNAc0ccyLkdrMoI/
RQMBw3uA6AQeGOutFxSW3iBEvNdKIboSkP6eiXgpvMH68A\nApZtG9ofwJmN3kAbLhFmYylB8tGgIBf2gGlglRMyX3WWkRAnE3j1wSfuS
Cxa1\n1+gXAdCokwKBgQCMa2FnIVzPvrgXhMNSnzoDV21AzncAgdqyaMk3p75ZN2p6qwBY\n1Hte3FTuzPbCXcudmUg1Lc5c6+DXgjGZf
9udd7jRjQ745GrmjAGuC\nr8pNc4J/BDfcmyyjZTZe2Ozu\n-----END PRIVATE KEY-----\n",
  "client_email": "h03-log-fetch@class-evidence-collection.iam.gserviceaccount.com",
  "client_id": "105632961180424281117",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/h03-log-fetch%40class-
evidence-collection.iam.gserviceaccount.com"
}' > class-evidence-collection-13690599effd.json
```

You may need to hit `Enter` on the last line to complete the full command.

4. Verify that the JSON credential file matches correctly.

Command lines

```
md5sum class-evidence-collection-13690599effd.json
```

This should produce output that looks identical to the below. If it doesn't, you may not have performed Step 3 correctly.

Expected output

```
[elk_user@sof-elk lab-4.4]$ md5sum class-evidence-collection-13690599effd.json
f90bdc77b8082b7d0b5f211f58f92dd8  class-evidence-collection-13690599effd.json
```

5. Once this is complete, you can move on to access Log data in Google Cloud below.

Authenticating to Google Cloud

To start with we need to authenticate to Google Cloud with the Service Account credentials we just loaded.

Command lines

```
gcloud auth activate-service-account h03-log-fetch@class-evidence-collection.iam.gserviceaccount.com --
key-file=class-evidence-collection-13690599effd.json
```

You should see a response that displays:

Expected results

```
[elk_user@sof-elk lab-4.4]$ gcloud auth activate-service-account h03-log-fetch@class-evidence-
collection.iam.gserviceaccount.com --key-file=class-evidence-collection-13690599effd.json
Activated service account credentials for: [h03-log-fetch@class-evidence-
collection.iam.gserviceaccount.com]
```

We can then check that our Service Account is loaded correctly locally on our VM.

Command lines

```
gcloud auth list
```

You should see a response that displays:

Expected results

```
[elk_user@sof-elk lab-4.4]$ gcloud auth list

Credentialed Accounts
ACTIVE  ACCOUNT
*       h03-log-fetch@class-evidence-collection.iam.gserviceaccount.com

To set the active account, run:
$ gcloud config set account `ACCOUNT`
```

Our next step is to set the default Project we want `gcloud` to access. The Service Account above only exists in the `class-evidence-collection` Project, so we need to set that as our default.

Command lines

```
gcloud config set project class-evidence-collection
```

As this is likely the first time you've run this command in the virtual machine, you will see the following response from Google Cloud:

Expected results

```
[elk_user@sof-elk lab-4.4]$ gcloud config set project class-evidence-collection

WARNING: You do not appear to have access to project [class-evidence-collection] or it
does not exist.
Are you sure you wish to set property [core/project] to class-evidence-collection?

Do you want to continue (Y/n)?
```

You can safely select **Y** and press Enter.

Review Logs Available in Google Cloud

We should be now ready to check for logging buckets and attempt to collect logs.

1. Let's first check to see what logging buckets we have available.

Command lines

```
gcloud logging buckets list
```

Notional results

You should see output that will be *similar* to the below. Remember it may not be exactly the same as you're looking at a live cloud instance.

```
[elk_user@sof-elk lab-4.4]$ gcloud logging buckets list
LOCATION  BUCKET_ID        RETENTION_DAYS  RESTRICTED_FIELDS  LIFECYCLE_STATE  LOCKED
CREATE_TIME                    UPDATE_TIME
global    GWS-Log_Storage  30                                 ACTIVE
2022-03-16T12:17:44.311497764Z  2022-03-16T12:17:44.311497764Z
global    _Default         30                                 ACTIVE
global    _Required        400                                ACTIVE           True
```

2. The `GWS-Log_Storage` bucket is where all our Google Workspace logs are being stored. Let's see if we have access to it.

Command lines

```
gcloud logging read --bucket=GWS-Log_Storage --location=global --view=_AllLogs
```

Depending on the activity in the Google Workspace account, **the above command may or may not have returned data (this is one of the challenges with working on a live cloud)**. The above command only returns data from the last day, by default. If data was returned, you will also have noticed that it was not in JSON format, which is what we want.

3. Let's expand out our search to return all data in the last 30 days, given that's what our Logging Bucket is set up for, and also return data in JSON format with it redirected to a file on our system.

Warning

You cannot directly copy and paste the below command, you will need to adjust the timestamp to be today's date one month into the past.

Command lines

```
gcloud logging read 'timestamp<="2022-03-17T00:00:00Z" AND timestamp>="2022-02-17T00:00:00Z"' --
bucket=GWS-Log_Storage --location=global --view=_AllLogs --format="json" > gws_logs_in_gcp.json
```

The above command should have produced the `gws_logs_in_gcp.json` in the `/home/elk_user/lab-4.4` you are in.

4. You can check the data that you've collected by using the `less` command shown below.

Command lines

```
less gws_logs_in_gcp.json
```

This will open the file so you can use the arrows on your keyboard to scroll up and down. When you want to exit the file press `q` and you'll be returned to the command line prompt.

Information

The data you're able to collect will vary depending on how much data is within the Log Bucket. This is one of the challenges of doing live labs in a cloud forensics class.

Collection and Import to SOF-ELK

You would now have the Google Workspace logs extracted from Google Cloud in a JSON format that could be imported into SOF-ELK. We'll extend further on the process above in the Google Cloud section of the class, where we discuss Google Cloud logging in more detail.

Clean Up

Before you finish the lab, ensure you run the final command below to log out from Google Cloud. This will ensure we don't run into issues with multiple students leaving sessions running for the Service Account in the Google Cloud Project.

Command lines

```
gcloud auth revoke --all
```

After you run the above command you will need to reauthenticate, following the steps above, if you wish to do this lab again.

Key Takeaways

- You now have the ability to understand how to access Google Cloud from the command line.
- You understand how to view available logging buckets in Google Cloud and determine how many days of data they are holding.
- You have the ability to directly collect logs from a Google Cloud logging bucket.
- You understand some of the limitations with using the `gcloud` tool and the additional switches needed to collect logs for analysis.

Lab 5.1: Google Cloud IAM and Access Tracking

Objectives

- Understand what is logged within Google Cloud related to user access and what is able to be tracked within the logs.
- Understand how to manipulate Google Cloud IAM and access logs within SOF-ELK.

Background

Google Cloud logs from the Billing, IAM, and Resource Manager services have been extracted from the Google Cloud instance that you will use for the rest of the labs in this section. This is an intro to help you understand what user accounts are doing within Google Cloud and how to read some of the logs that you can extract. The logs extracted for this exercise are taken from a specific point in time. In general, they aren't very large; however, this may change if you have a very large number of Google Cloud user accounts.

Preparation

To prepare:

1. Make sure your SOF-ELK® VM is running.
2. Obtain access Kibana via the IP address shown on the SOF-ELK® VM.
3. Obtain access to the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-03-15 00:00 UTC** to **2021-05-02 00:00 UTC**.

Load Data

In the setup instructions you loaded a GeoIP database, which means that any new data you import will have IP addresses enriched with geolocation information. For this lab, this extra information will be very useful. As such, we didn't preload the data, and you will have the opportunity to see how simple it is to load data into SOF-ELK. Additionally, we are continually making changes to the Google Cloud parsers, and because of this we'll get you to update the config files in SOF-ELK before we load this lab's data set.

The Google Cloud JSON logs for this lab have already been exported from Google Logging. To help in keeping the amount of data to a reasonable level, we have trimmed the data set to only include Google Cloud `google.cloud.audit.AuditLog` Logging. However, there were no other changes made to the JSON that was extracted from Google Cloud, so it is still representative of what you'll experience outside of this course.

The Google Cloud IAM Audit Logs have been saved into `/home/elk_user/lab-5.1_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials
    - Username: ***elk_user***
    - Password: ***forensics***

You may log on to the console; however, it's recommended to log in via SSH.

2. Extract all the `.json` files from the preceding ZIP file into the Logstash folder for parsing by FileBeats with the following command:

Command lines

```
cd ~
unzip lab-5.1_source_evidence.zip -d /logstash/gcp/
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you have 450 documents loaded in the `gcp` index:

Command lines

```
sof-elk_clear.py -i list
```

Expected results

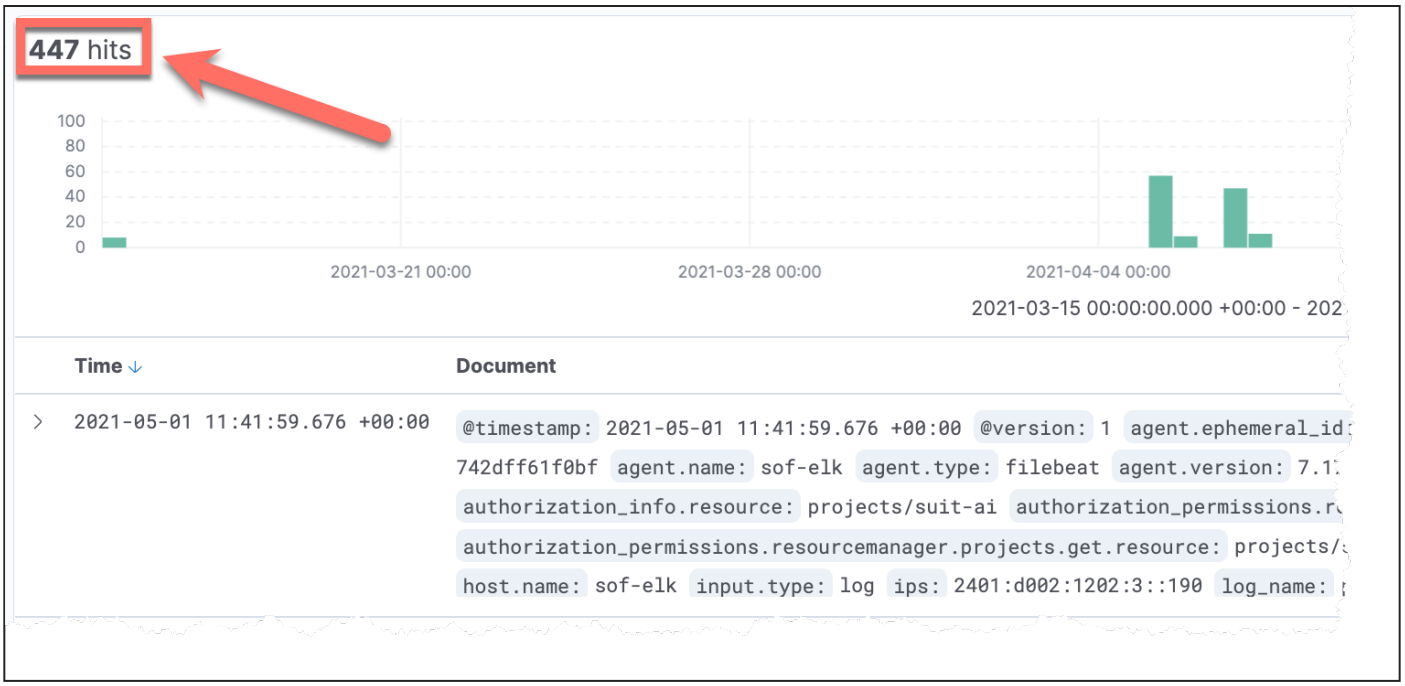
```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- gcp (450 documents)
- gws (1,157 documents)
- httpdlog (1,381 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

5. Once it is complete, the data will be available in the `gcp-*` index.

Start in the Discover tab.

Before we go too far, let's ensure the date range is correct for the data we'll be using in this lab. Set your search timeframe to `2021-03-15 00:00:00.000 +00:00` to `2021-05-02 00:00:00.000 +00:00` hit `Update`.

To ensure all your data has loaded correctly, you should now see **447 hits** just above the top left of the histogram.



Warning

If you do not see **447 hits** just above the top left of the histogram, you either haven't loaded your data correctly, or you have searches/filters applied that need to be cleared. If you are unable to correct this seek assistance from your instructor before going any further in this lab.

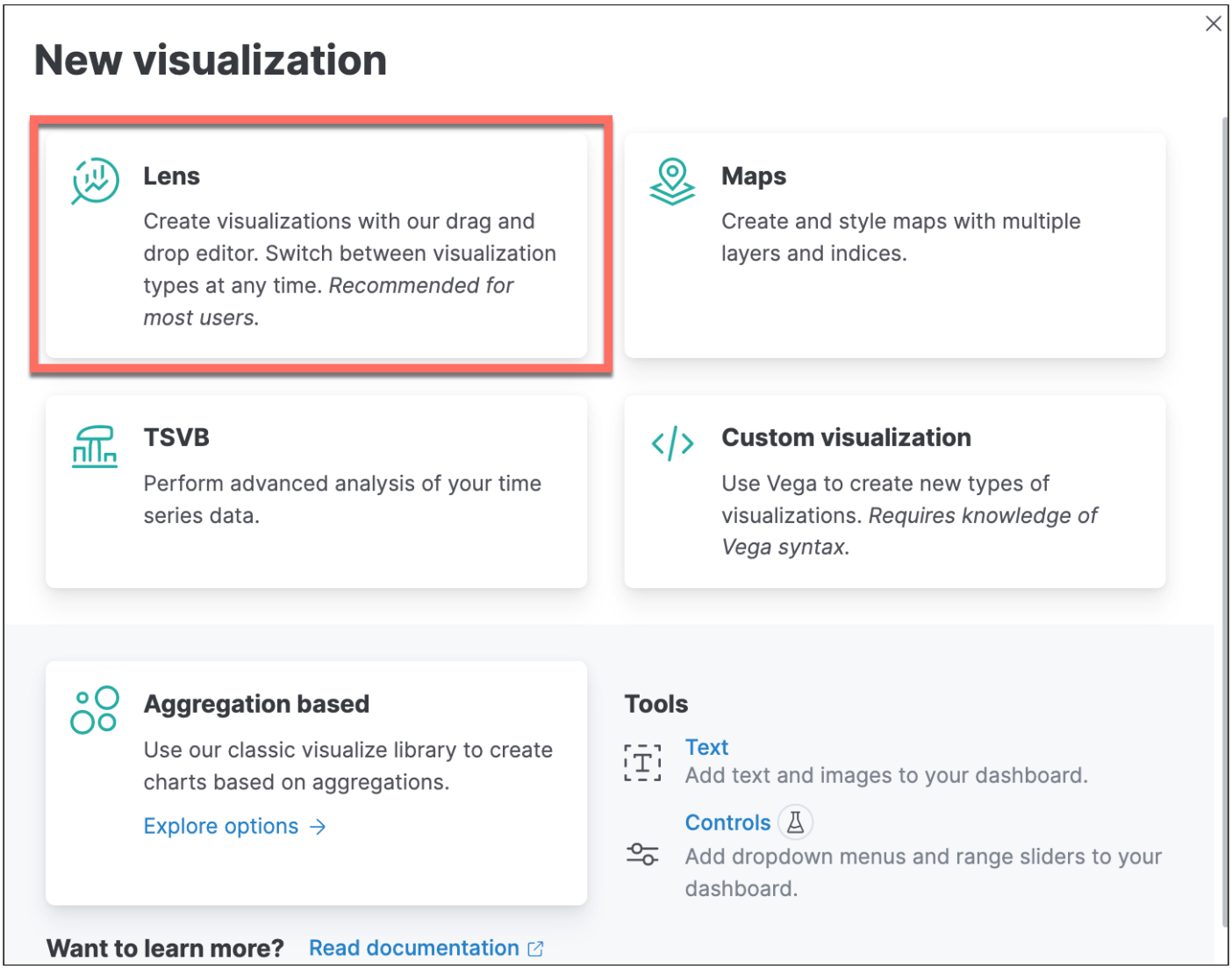
Reviewing Users Within Our Logs

This subsection will be done in the Visualize tab

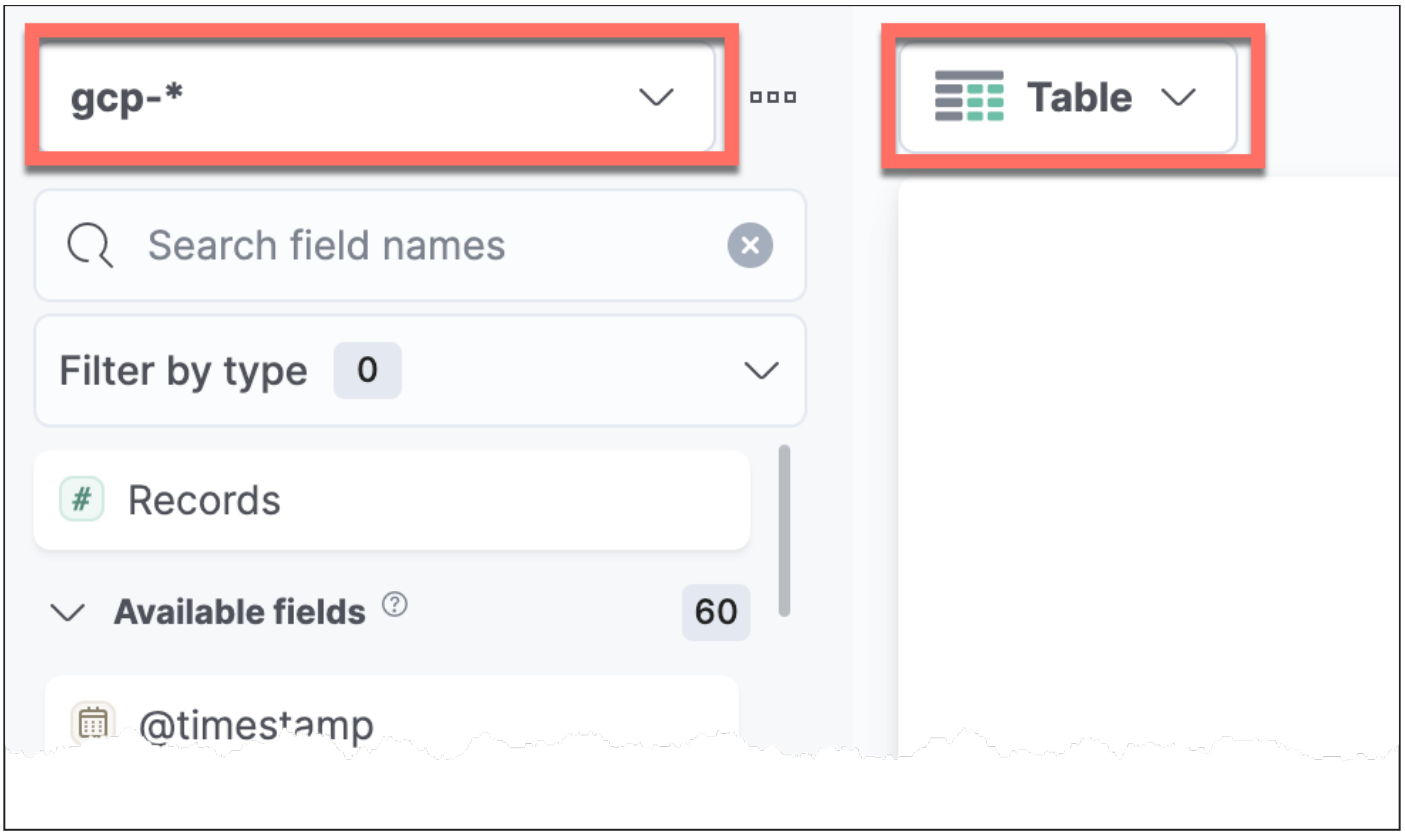
To start with, let's take a look at the users we have within the logs you have available to search. To do this, we want to generate a summary of all the usernames in the `username` field parsed out by Logstash.

Move to the `Visualize` tab in Kibana and select `Create visualization`.

We will use `Lens` because it's the easiest way to create a new visualization.

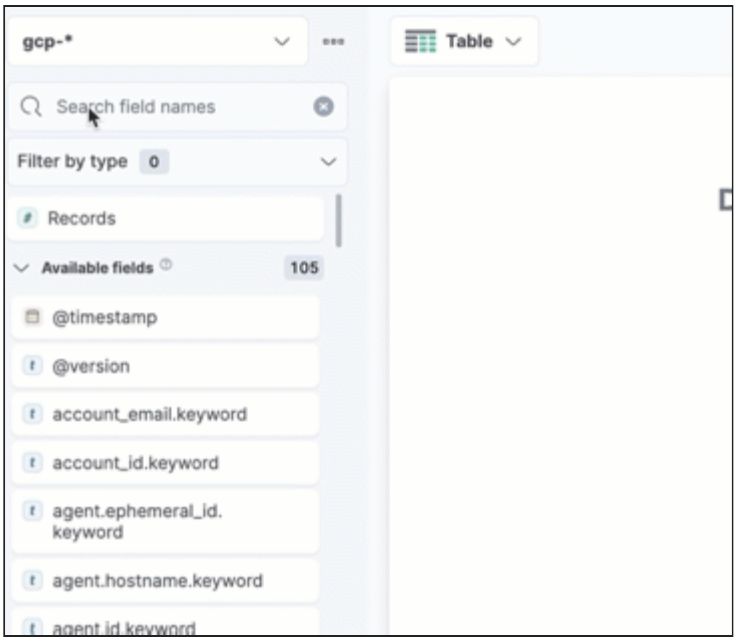


We will need to again move the `index` selector over to `gcp-*` to ensure we are looking at the Google Cloud data. We will also change the graph type to `Data table`.



The number of available filters may differ after you have performed a `sof-elk_update.sh` *as we continue to parse more fields over time.*

Drag and drop the `username.keyword` field into the table area.



1. Based on the table that is created inside of Lens Visualize, we can see the users' accounts and the number of records available for each user account.

How many records are there for each user account within the logs you have just summarized with Lens Visualize?

Hint

Review the "Count of records" in your Visualize table.

Answer

- `admin@longconsecurity.com` has **414** records
- `evil-user@longconsecurity.com` has **10** records
- `service-agent-manager@system.gserviceaccount.com` has **5** records

Table

Top values of username.keyword	Count of records
admin@longconsecurity.com	414
evil-user@longconsecurity.com	10
service-agent-manager@system.gserviceaccou...	5

2. One last thing that may be of use later on is to drag and drop the `source_ip` field into the table as well.

Based on the updated summary table, is there any overlap in the source IP addresses of where the user accounts were being accessed?

Hint

Review the `source_ip` values visually to see if there is more than one `username` account used from the same `source_ip` addresses. There are some IP addresses that are IPv6, this is not a mistake.

Answer

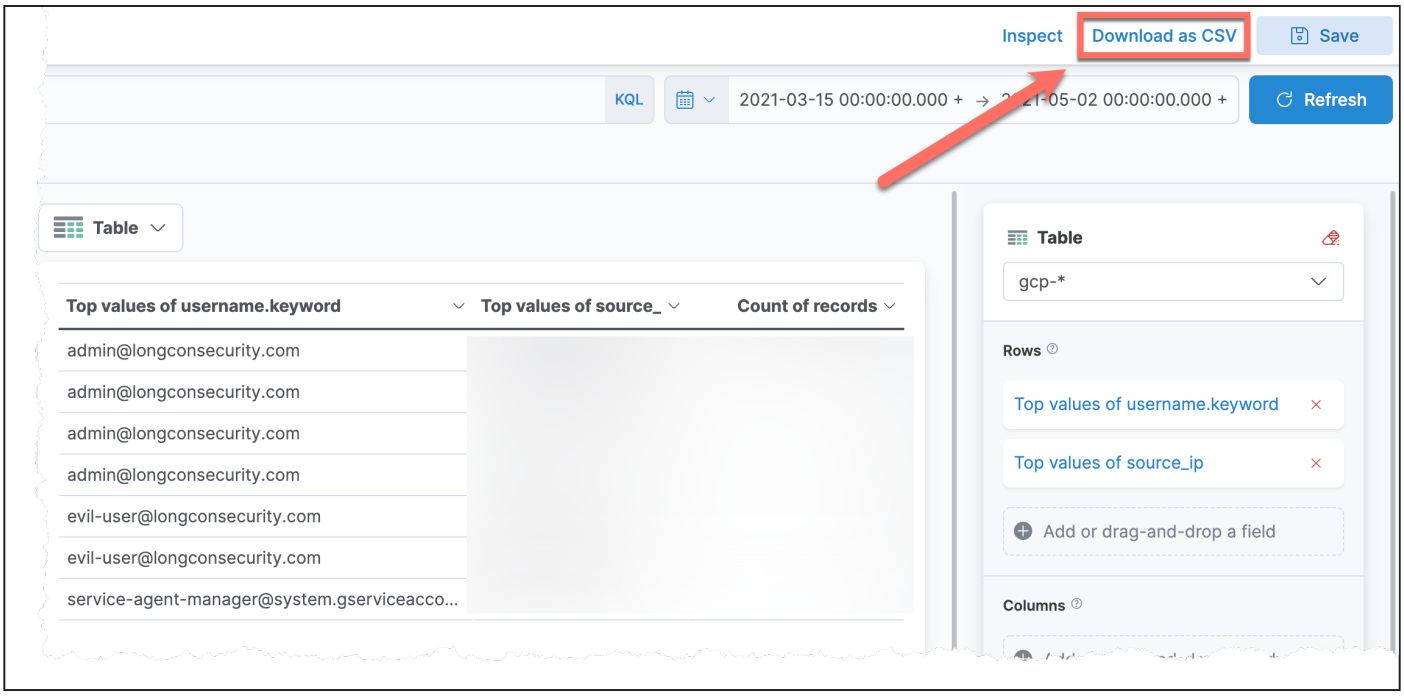
No, there doesn't appear to be any overlap in the IP addresses for the different users. Individual users have accessed their account from different locations, but there is no overlap in the source IP addresses for different accounts.

Top values of username.keyword	Top values of source_	Count of records
admin@longconsecurity.com	2401:d002:1202:3::1...	339
admin@longconsecurity.com	123.254.126.102	27
admin@longconsecurity.com	2401:d002:1202:3::16c	25
admin@longconsecurity.com	Other	23
evil-user@longconsecurity.com	2002:a05:651c:222::	7
evil-user@longconsecurity.com	2a0b:f4c1:2::242	3
service-agent-manager@system.gserviceacco...	2002:a49:1f8a::	5

You will notice that there is a mix of both IPv4 and IPv6 addresses as the source IP addresses. This isn't a mistake, it's the legitimate activity of our users in the Google Cloud using native IPv6.

You may want to export the data that you just created to answer the last question. This may become useful later; however, it's also a very common task that you'd want to do when investigating user access. This would allow you to have a record of where your users were accessing their Google Cloud accounts from.

To export the data from the Visualize Table, click on `Download as CSV` in the top right of the screen.



Now that we have an understanding of the user accounts within the logs, let's move back to reviewing user activity.

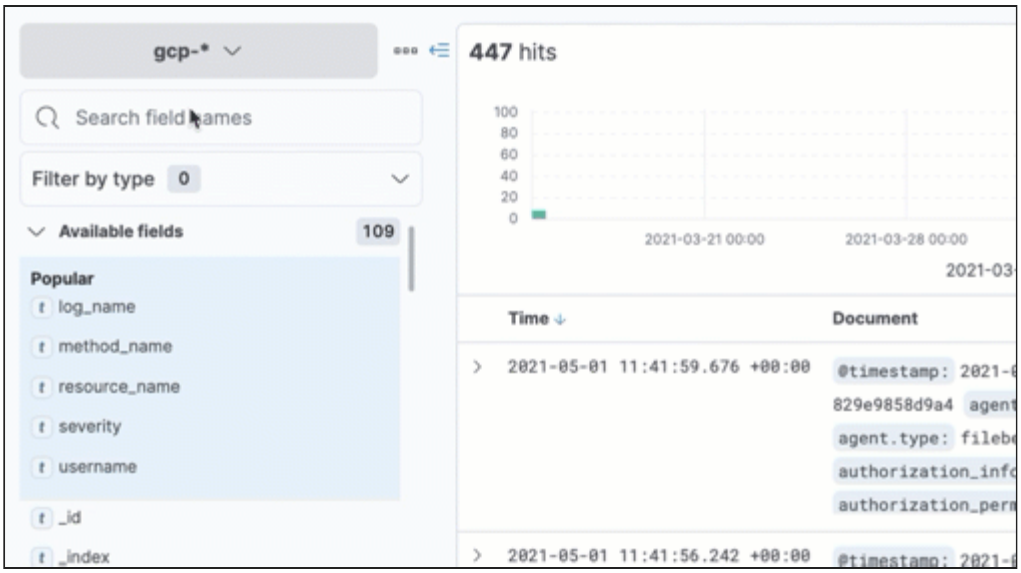
Reviewing User Activity

Move back into the Discover tab.

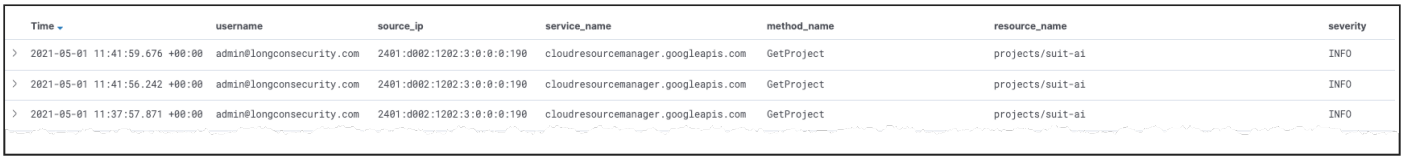
The initial view of the data within the Discovery view of SOF-ELK is very unstructured. In the first few steps of this exercise, we will add in specific columns to make the data more useful for reviewing Google Cloud VM Agent Logs. We will want to add in the following column names, in the order shown:

- `username`
- `source_ip`
- `service_name`
- `method_name`
- `resource_name`
- `severity`

To add in these names, you can search for them in the field selector under the index dropdown menu. When you start typing the field name, it will appear for you to select with `+`, which appears when you hover your mouse over the field name.



Provided you added the fields in the preceding order, your events in Discover should now appear similar to the following image.



Time	username	source_ip	service_name	method_name	resource_name	severity
2021-05-01 11:41:59.676 +00:00	admin@longconsecurity.com	2401:d002:1202:3:0:0:0:190	cloudresourcemanager.googleapis.com	GetProject	projects/suit-ai	INFO
2021-05-01 11:41:56.242 +00:00	admin@longconsecurity.com	2401:d002:1202:3:0:0:0:190	cloudresourcemanager.googleapis.com	GetProject	projects/suit-ai	INFO
2021-05-01 11:37:57.871 +00:00	admin@longconsecurity.com	2401:d002:1202:3:0:0:0:190	cloudresourcemanager.googleapis.com	GetProject	projects/suit-ai	INFO

There are three (3) different `service_name` items in the logs. Let's start by looking at only the `iam.googleapis.com` log.

1. Run the following search for all of the IAM logs and review the output.

   ```
   service_name: iam.googleapis.com
   ```

   As you scroll through the data, you will notice that there are a lot of entries related to a `resource_name` that include the term `subnetworks`. These are the VPC resources within Google Cloud that we will look at later in this section of the class.

2. Exclude all the VPC resources from your search by updating your search bar to the following:

   ```
   service_name: iam.googleapis.com AND NOT resource_name : "subnetworks"
   ```

   This should now leave you with a more manageable data set to look at.

Looking at the IAM events, there are interactions with two (2) VM instances (compute resources). Which user account interacted with them and what were the names of the VMs?

Hint

To answer this question, you will need to narrow your search even further by appending `AND resource_name: *compute*` to your current search. This will allow you to only see IAM events that involve the **compute** resource within Google Cloud.

Answer

You will see that the `admin@longconsecurity.com` user interacted with two compute resources:

- compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1
- compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-2

Time	username	method_name	resource_name
2021-04-11 01:56:32.033 +00:00	admin@longconsecurity.com	google.iam.admin.v1.QueryGrantableRoles	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-2
2021-04-11 01:56:32.032 +00:00	admin@longconsecurity.com	google.iam.admin.v1.QueryGrantableRoles	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1
2021-04-11 01:56:32.030 +00:00	admin@longconsecurity.com	google.iam.admin.v1.GetPolicyDetails	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-2
2021-04-11 01:56:32.029 +00:00	admin@longconsecurity.com	google.iam.admin.v1.GetPolicyDetails	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1
2021-04-10 13:51:38.172 +00:00	admin@longconsecurity.com	google.iam.admin.v1.QueryGrantableRoles	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1
2021-04-10 13:36:54.524 +00:00	admin@longconsecurity.com	google.iam.admin.v1.QueryGrantableRoles	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1
2021-04-10 13:36:54.524 +00:00	admin@longconsecurity.com	google.iam.admin.v1.GetPolicyDetails	//compute.googleapis.com/projects/suit-ai/zones/us-west1-b/instances/instance-1

3. We can also use the same searching technique to look for other Google Cloud resources as well.

Looking at the IAM events, there are interactions with storage resources. Which user account interacted with them, what storage type was it, and what were the names of the storage locations?

Hint

You can alter your search from the previous answer to include `storage`, which would look like this:

```
service_name: iam.googleapis.com AND NOT resource_name : "subnetworks" AND resource_name: *storage*
```

Answer

You will again see that the `admin@longconsecurity.com` user interacted with the following storage `buckets`:

- confidential_plans
- flight-maps

Time	username	resource_name
2021-04-26 12:54:00.067 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 12:54:00.066 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 12:00:33.638 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 12:00:21.563 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 12:00:21.560 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 11:22:10.459 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-26 11:22:10.288 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/confidential_plans
2021-04-13 04:14:51.715 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/flight-maps
2021-04-13 01:32:59.865 +00:00	admin@longconsecurity.com	//storage.googleapis.com/projects/_/buckets/flight-maps

4. Let's move back to our original search so we can see all the IAM events and review Service Account tracking.

```
service_name: iam.googleapis.com AND NOT resource_name : "subnetworks"
```

5. When you scroll through the IAM logs, you will notice `method_name` items that include the word "service". Let's take a closer look at those events.

Are there any `CreateServiceAccount` events, and if so, what are their `Account_Email` and `account_ID` values?

Hint

To find them, you will need to search for:

```
service_name: iam.googleapis.com AND NOT resource_name : "subnetworks" AND method_name : *Service*
```

Answer

Yes, there are two `CreateServiceAccount` events in our logs.

Time	username	method_name	resource_name
2021-04-27 14:50:18.660 +00:00	admin@longconsecurity.com	google.iam.admin.v1.ListServiceAccounts	projects/suit-ai
2021-04-06 14:07:58.116 +00:00	admin@longconsecurity.com	google.iam.admin.v1.DeleteServiceAccountKey	projects/-/serviceAccounts/115762896693196005388
2021-04-06 14:05:43.148 +00:00	admin@longconsecurity.com	google.iam.admin.v1.CreateServiceAccountKey	projects/-/serviceAccounts/115762896693196005388
2021-04-06 13:53:22.491 +00:00	admin@longconsecurity.com	google.iam.admin.v1.CreateServiceAccountKey	projects/-/serviceAccounts/115762896693196005388
2021-04-06 13:52:13.113 +00:00	admin@longconsecurity.com	google.iam.admin.v1.CreateServiceAccount	projects/suit-ai
2021-03-15 11:11:40.574 +00:00	-	google.iam.admin.v1.CreateServiceAccount	projects/suit-ai

To determine the `account_email` and `account_ID` values, you need to expand out each event individually. You will find both of these values beside each other.

- `48649762025-compute@developer.gserviceaccount.com` and `100820242691295494244`
- `sof-elk-srv-account@suit-ai.iam.gserviceaccount.com` and `115762896693196005388`

An example of the `sof-elk-srv-account@suit-ai.iam.gserviceaccount.com` `account_email` and `115762896693196005388` `account_ID` values when expended:

2021-04-06 13:52:13.113 +00:00 admin@longconsecurity.com 2401:d002:1202:3::190 google.iam.admin.v1.CreateServiceAccount

Expanded document

Table JSON

Actions	Field	Value
	t _id	WEH8u38BeUqOKw2oXQdE
	t _index	gcp-2021.04
	# _score	-
	t _type	_doc
	@timestamp	2021-04-06 13:52:13.113 +00:00
	t @version	1
	t account_email	sof-elk-srv-account@suit-ai.iam.gserviceaccount.com
	t account_id	115762896693196005388
	t agent.ephemeral_id	27735a1d-7e48-4e79-99fe-829e9858d9a4
	t agent.hostname	sof-elk

There are also `CreateServiceAccountKey` events in the previously run search. Which account were the keys created for?

Hint

To find them quickly, you can append `AND method_name : *CreateServiceAccountKey*` to your previous search, so it would look like this:

```
service_name: iam.googleapis.com AND NOT resource_name : "subnetworks" AND method_name : *Service*
AND method_name : *CreateServiceAccountKey*
```

Answer

The two `CreateServiceAccountKey` events both have the `resource name` value of `projects/-/serviceAccounts/11576289669319600538 8`. Therefore, the keys were created against the Service Account with an ID of `11576289669319600538 8`.

You can match this Service Account ID back to the IDs you identified in the previous question. Once you do this, it will match to `sof-elk-srv-account@suit-ai.iam.gserviceaccount.com`. Additionally, we can also see the user that created the Service Account Keys, `admin@longconsecurity.com`.

Within Google Cloud logs, it's common to find the resource names referenced by their unique ID. This is because you can have resource names with the same text in different projects. For example, you could have two VMs with the name "Super Smart VM" sitting in two different projects, but because they would appear in the same logs for the organization, it's always good to be familiar with object IDs.

Filtering Clean Up

Ensure you clear out any filters you may still have applied for this lab. This will ensure your next lab isn't impacted by any filters you used in this lab.

Key Takeaways

- You now have the ability to quickly understand what users are in Google Cloud IAM and access logs.
- You understand that Google Cloud logs don't always include the human-readable name given to an object, and you have the ability to match a Resource ID back to a human-readable name.
- You now have the ability to quickly narrow down IAM logs to search for relevant information related to accesses or actions performed by a user within Google Cloud.

Lab 5.2: Google VM Ops Agent: Agent Log Analysis

Objectives

- Understand the capabilities of Google Ops Agent.
- Develop an understanding of how to use Google Ops Agent logs within SOF-ELK.
- Understand how to identify different log sources from a host and how they are useful to an investigation.

Background

LongConSecurity has three VMs within a VPC that have Google Cloud's host-based Agent Logging installed on them. These were installed with the default commands within Google Cloud and were pushed to the VMs by Google Cloud Monitoring with Google Shell commands.

Preparation

To prepare:

1. Make sure your SOF-ELK® VM is running.
2. Obtain access Kibana via the IP address shown on the SOF-ELK® VM.
3. Obtain access to the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-04-10 00:00 UTC** to **2021-05-01 00:00 UTC**.

Load Data

The Google Cloud JSON logs for this lab have already been exported from Google Logging. To help in keeping the amount of data to a reasonable level, we have trimmed the data set to only include Google Cloud VM Agent Logging. However, there were no other changes made to the JSON extracted from Google Cloud, so it is still representative of what you'll experience outside of this course.

The Google Cloud VM Agent logs have been saved into `/home/elk_user/lab-5.2_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract all the `.json` files from the preceding ZIP file into the correct Logstash folder for parsing by FileBeats with the following command:

Command lines

```
cd ~
unzip lab-5.2_source_evidence.zip -d /logstash/gcp/
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you now have 10,306 documents loaded in the `gcp` index:

Command lines

```
sof-elk_clear.py -i list
```

Note

Your document count will be different if you haven't completed the previous lab.

Expected results

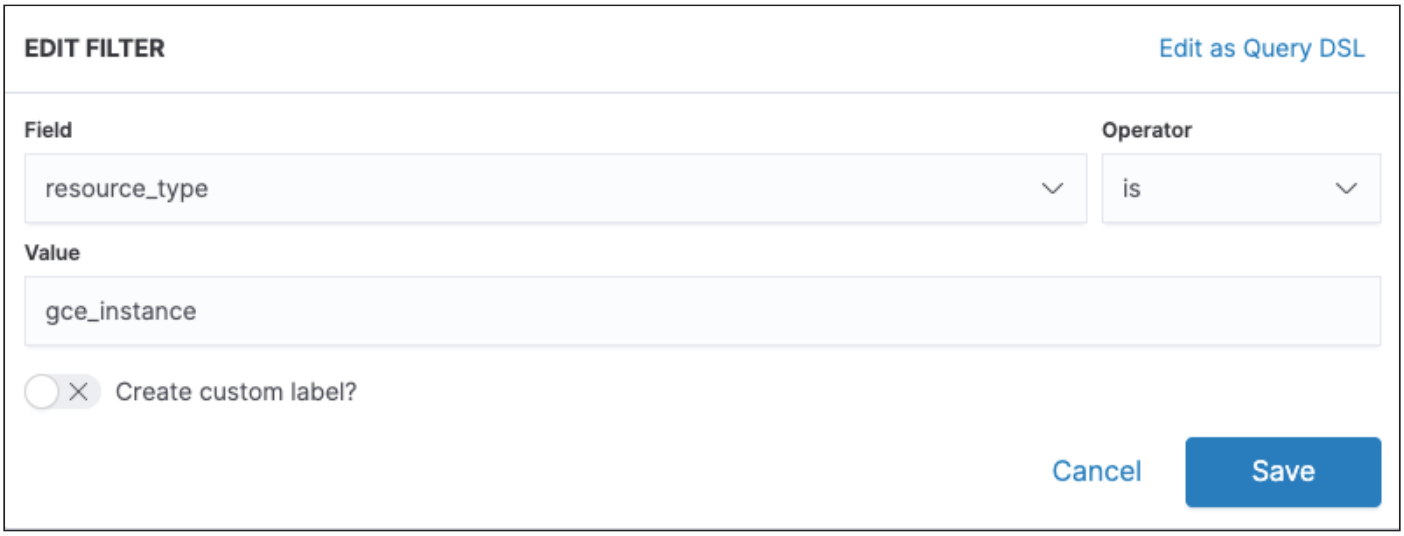
```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- gcp (10,306 documents)
- gws (1,157 documents)
- httpdlog (1,381 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

5. Once it is complete, the data will be available in the `gcp-*` index.

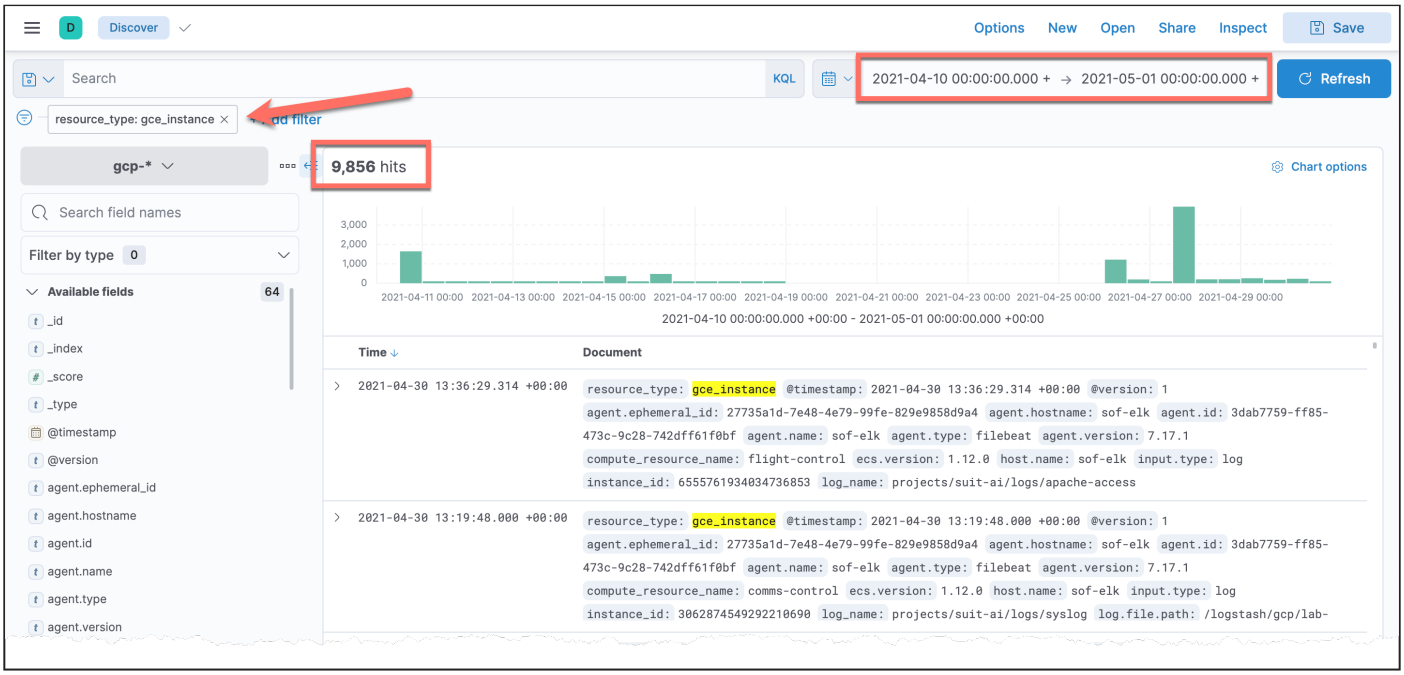
Lab Content

Start in the **Discover tab**.

Before we go too far, let's ensure the date range is correct for the data we'll be using in this lab. Set your search time frame to `2021-04-10 00:00:00.000 +00:00` to `2021-05-01 00:00:00.000 +00:00` and click `Update`. As you may be jumping between different exercises, let's also narrow down the logs we're looking at with "Add filter". Select the `Field` to be `resource_type`, set the `Operator` to be `is`, and then set the `Value` to be `gce_instance`.



Based on the time frame and filter you just applied, you should see **9,856 hits** or entries in your histogram.



The initial view of the data within the Discovery view of SOF-ELK is very unstructured. In the first few steps of this exercise, we will add in specific columns to make the data more useful for reviewing Google Cloud VM Agent logs.

Note

If any of the below field names do not appear when you search for them, try instead running an open search in the main search bar at the top of the Discover page. For example, if `text_payload` doesn't appear, search for this:

```
text_payload : *
```

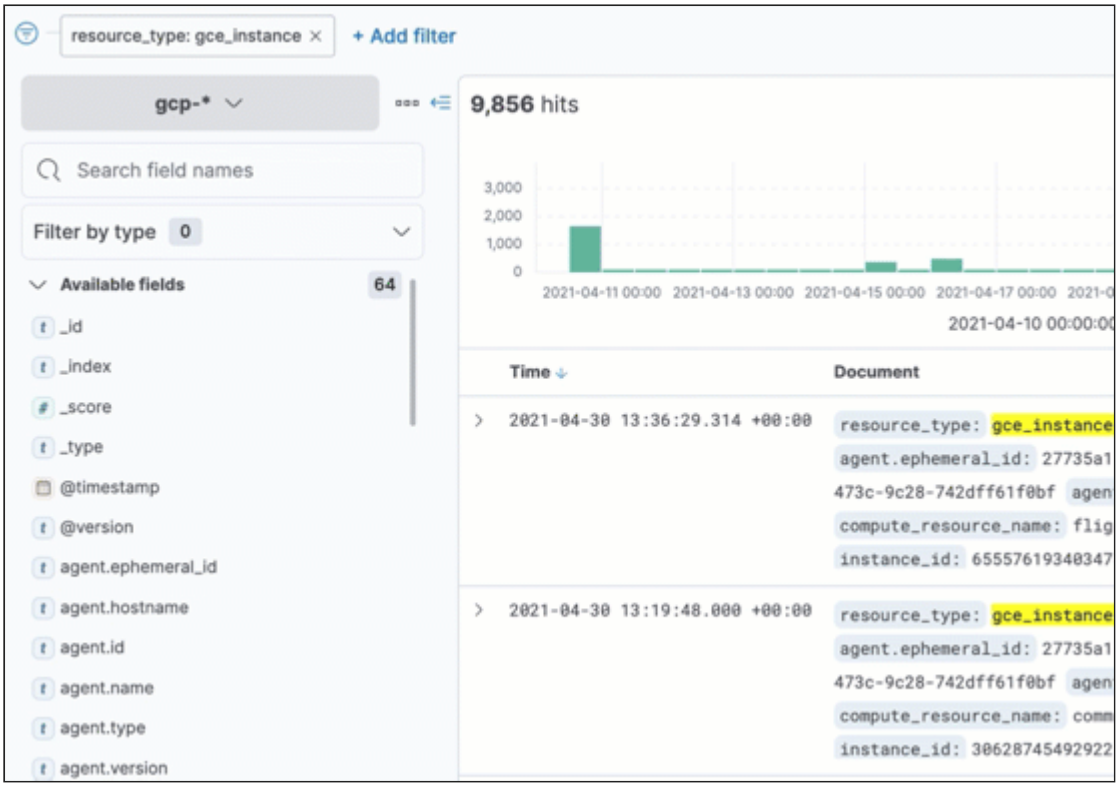
After you do this, the field you just searched for should appear above, as shown in the animation.

This issue occurs because Kibana only shows fields for the first 500 events, by default. Once you run a search the missing field will be in the newly loaded 500 events on screen and the field appears.

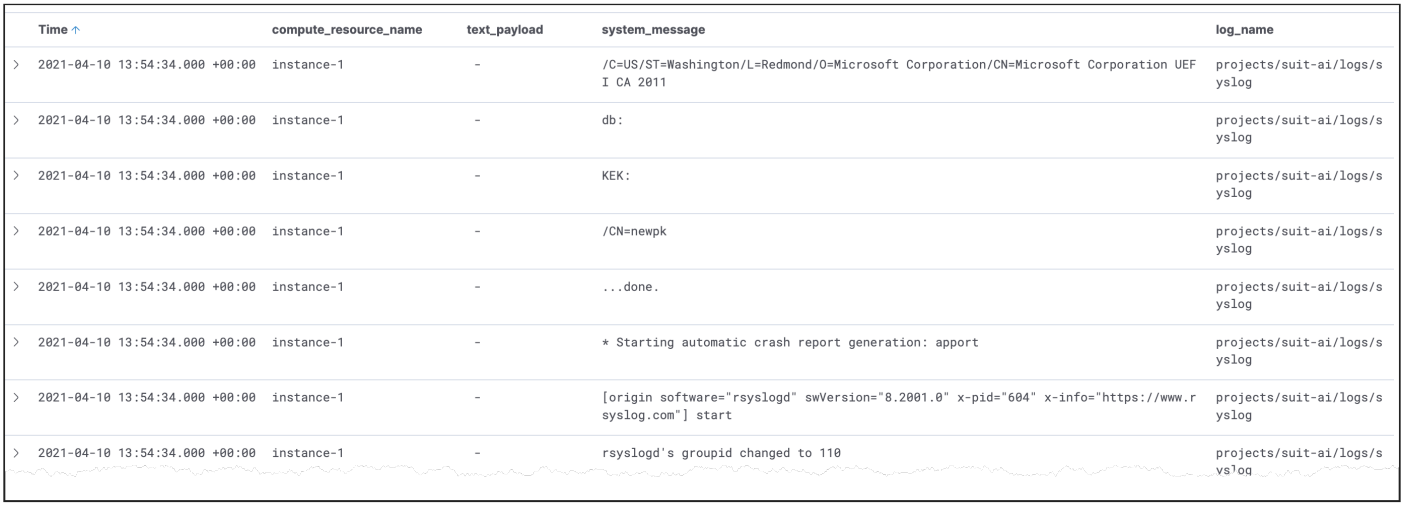
We will want to add in the following column names, in the order shown:

- `compute_resource_name`
- `text_payload`
- `system_message`
- `log_name`

To add in these names, you can search for them in the field selector under the index dropdown menu. When you start typing the field name, it will appear for you to select with `+`, which appears when you hover your mouse over the field name.



Provided that you added the fields in the specified order, your events in the Discovery viewer should now appear similar to the following image.



	Time ↑	compute_resource_name	text_payload	system_message	log_name
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	/C=US/ST=Washington/L=Redmond/O=Microsoft Corporation/CN=Microsoft Corporation UEFI CA 2011	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	db:	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	KEK:	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	/CN=newpk	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	...done.	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	* Starting automatic crash report generation: apport	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	[origin software="rsyslogd" swVersion="8.2001.0" x-pid="604" x-info="https://www.rsyslog.com"] start	projects/suit-ai/logs/syslog
›	2021-04-10 13:54:34.000 +00:00	instance-1	-	rsyslogd's groupid changed to 110	projects/suit-ai/logs/syslog

We're now ready to start investigating the events from our Google Cloud VM Agent logs.

Reviewing Available Logs

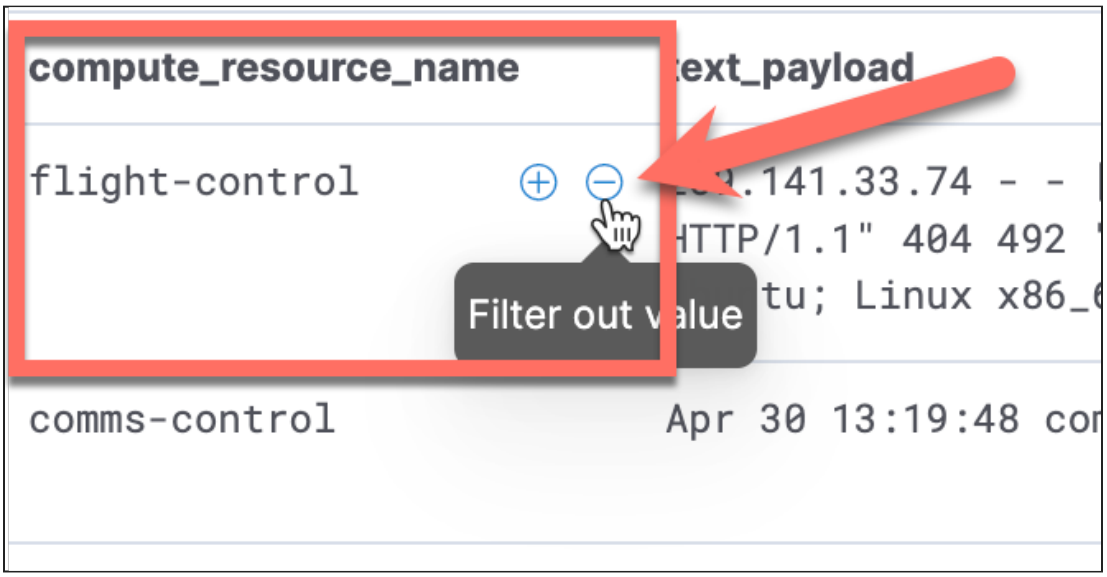
1. The logs collected by the Google Cloud VM Agent can vary, depending on what services are running on each VM. Within the log set for this lab, you have three different VMs to explore. Let's start off by simply understanding the logs we have to work with.

What are the 'instance IDs' and 'instance names' of the three systems that we have logs for within the data?

Hint

One of the fields we had you add to the Discover event view earlier will provide you with this information. The `compute_resource_name` is the label given to the VMs by the person who created them.

You can start by recording the names of the systems and then slowly excluding them from the search with a filter.



Answer

Using the method provided in the Hint above, should result in you identifying three system;

a. `flight-control`, which has an ID of `6555761934034736853`,

b. `comms-control`, which has an ID of `3062874549292210690`, and

c. `instance-1`, which has an ID of `651533120232493846`.

2. You will also notice from the previous question that there were many events that did not have an "instance name" but did have an "instance ID." Within Google Cloud, the correct way to refer to a resource is via its ID. To keep things simple in this lab, we will mainly use the "instance name"; however, you should continue to search with the "instance ID" and "instance name" together when reviewing GSE logs.

When did logs start and stop for each of the three systems?

Hint

The most accurate way to search for each of these is using the `instance_id` for each of the systems you identified in the previous question. An example of this for `instance-1` would be `instance_id : "651533120232493846"`.

Answer

Logs for `flight-control` started on `2021-04-26 11:47:06.092 UTC` and finished on `2021-04-30 13:36:29.314 UTC`.

```
instance_id : "6555761934034736853" OR compute_resource_name : "flight-control"
```

Logs for `comms-control` started on `2021-04-27 14:51:25.058 UTC` and finished on `2021-04-30 13:19:48.000 UTC`.

```
instance_id : "3062874549292210690" OR compute_resource_name : "comms-control"
```

Logs for `instance-1` started on `2021-04-10 13:54:01.228 UTC` and finished on `2021-04-18 12:47:05.461 UTC`.

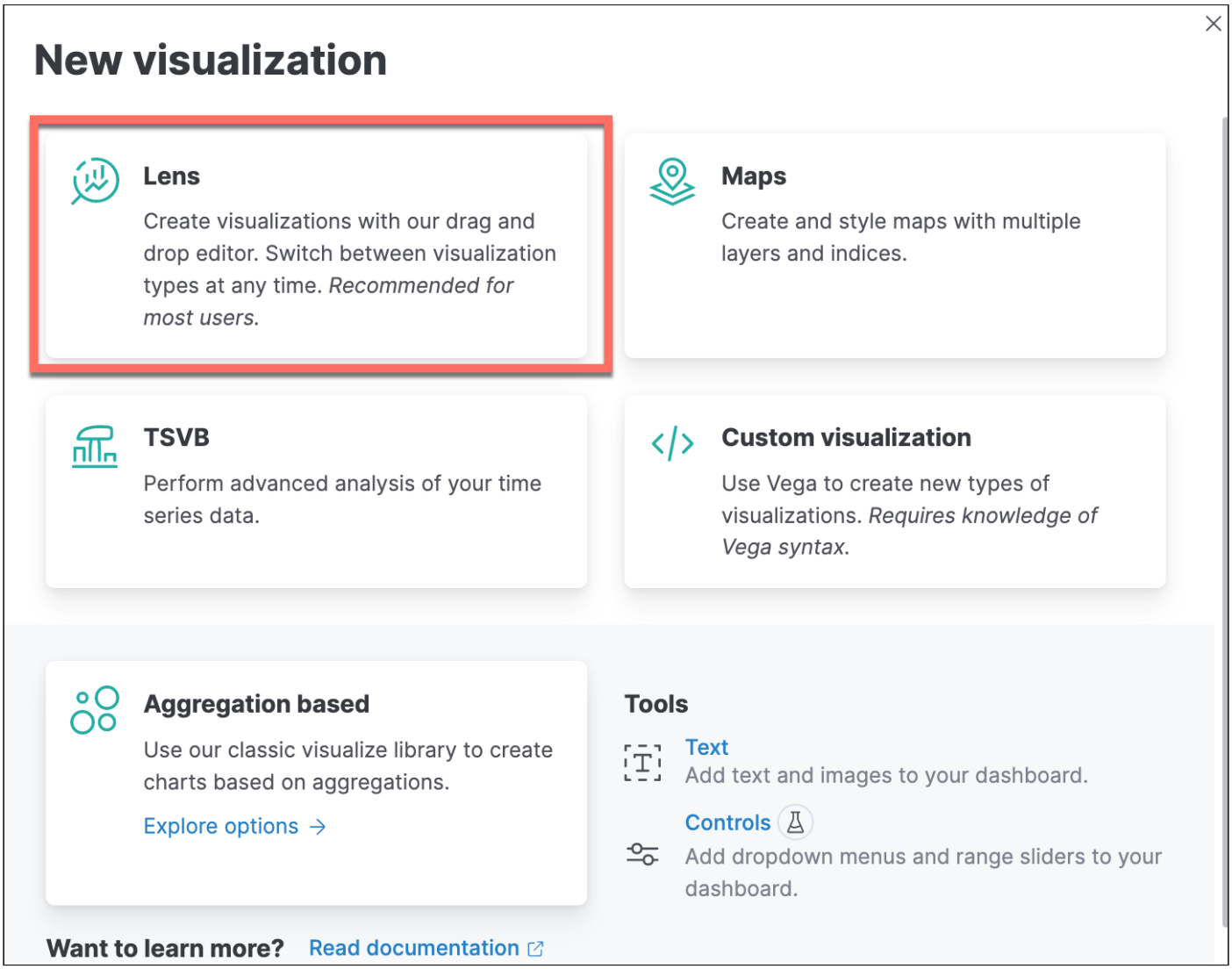
```
instance_id : "651533120232493846" OR compute_resource_name : "instance-1"
```

3. Once you have completed the preceding question, ensure you clear your search bar and hit 'Update'. Let's now take a look at the types of logs that are getting captured from each system.

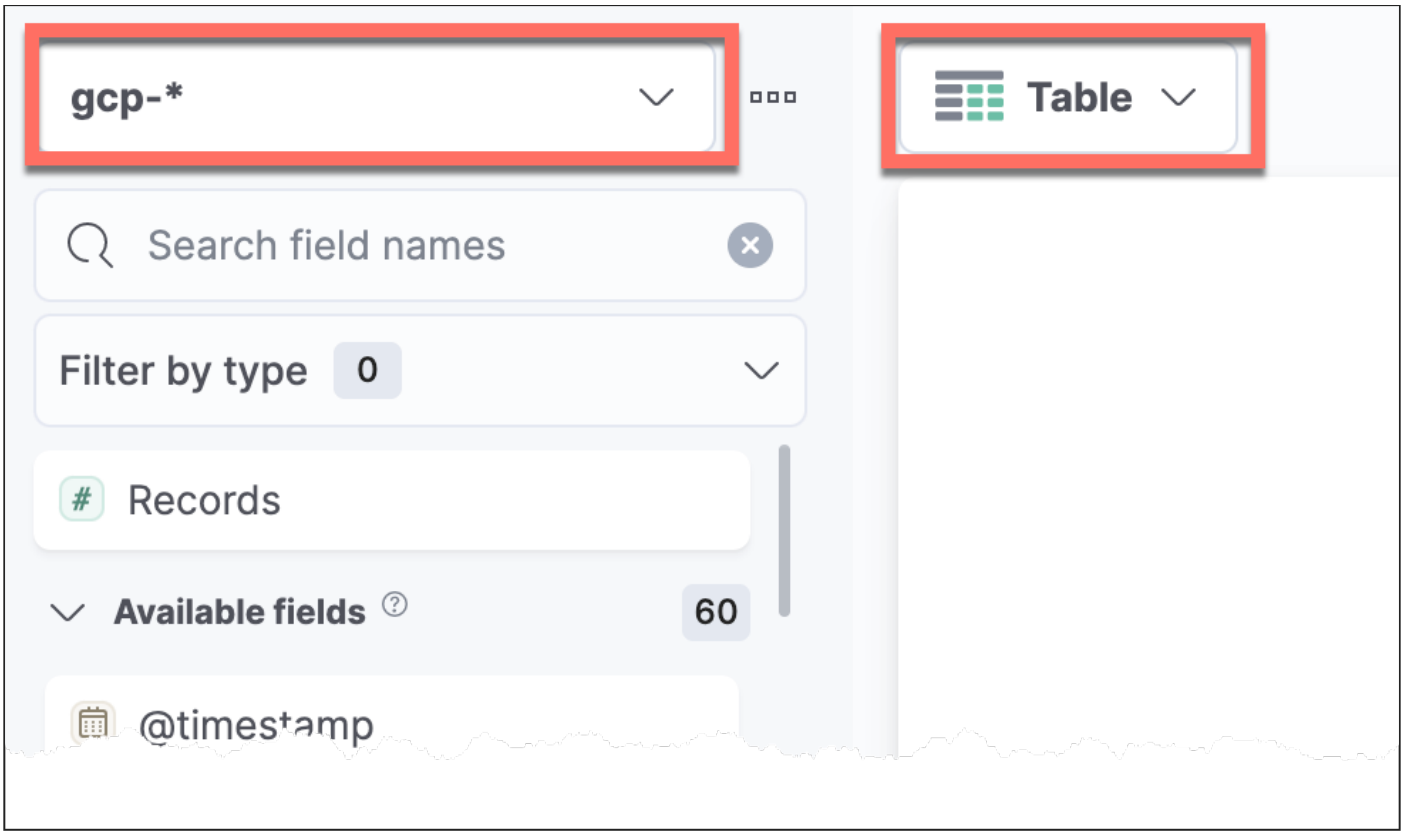
This question will be answered in the **Visualize tab**.

Move to the `Visualize` tab in Kibana and select `Create visualization`.

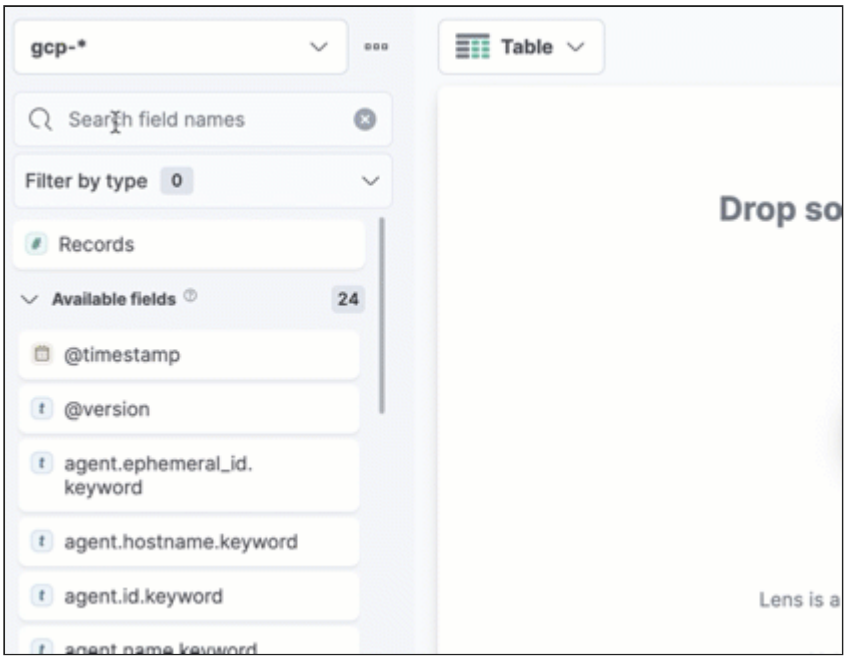
We will use `Lens` because it's the easiest way to create a new visualization.



We again need to again move the `index` selector over to `gcp-*` to ensure we are looking at the Google Cloud data. We will also change the graph type to `Data table`



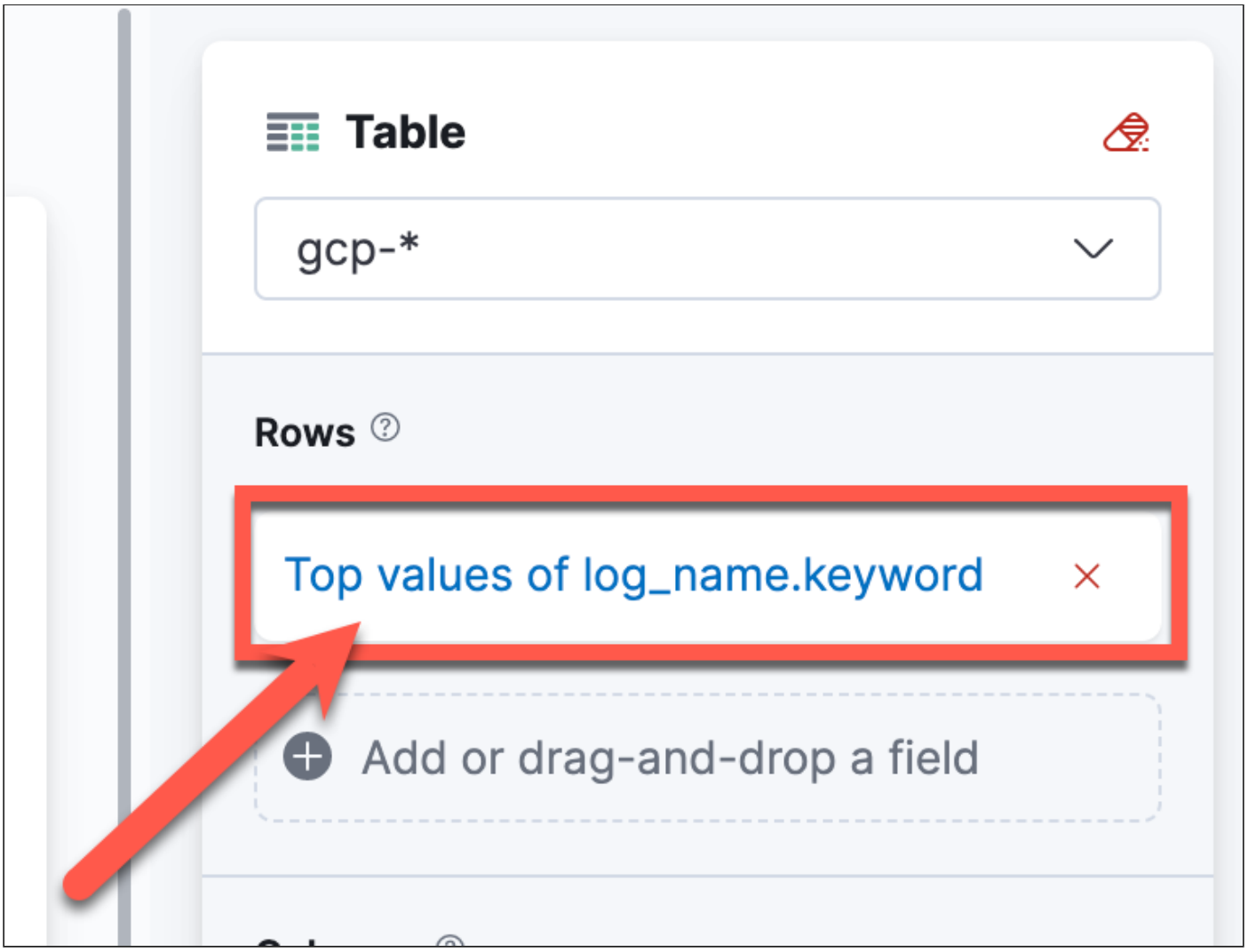
Type the `log_name.keyword` field name into the field search and then drag and drop the field into the table area.



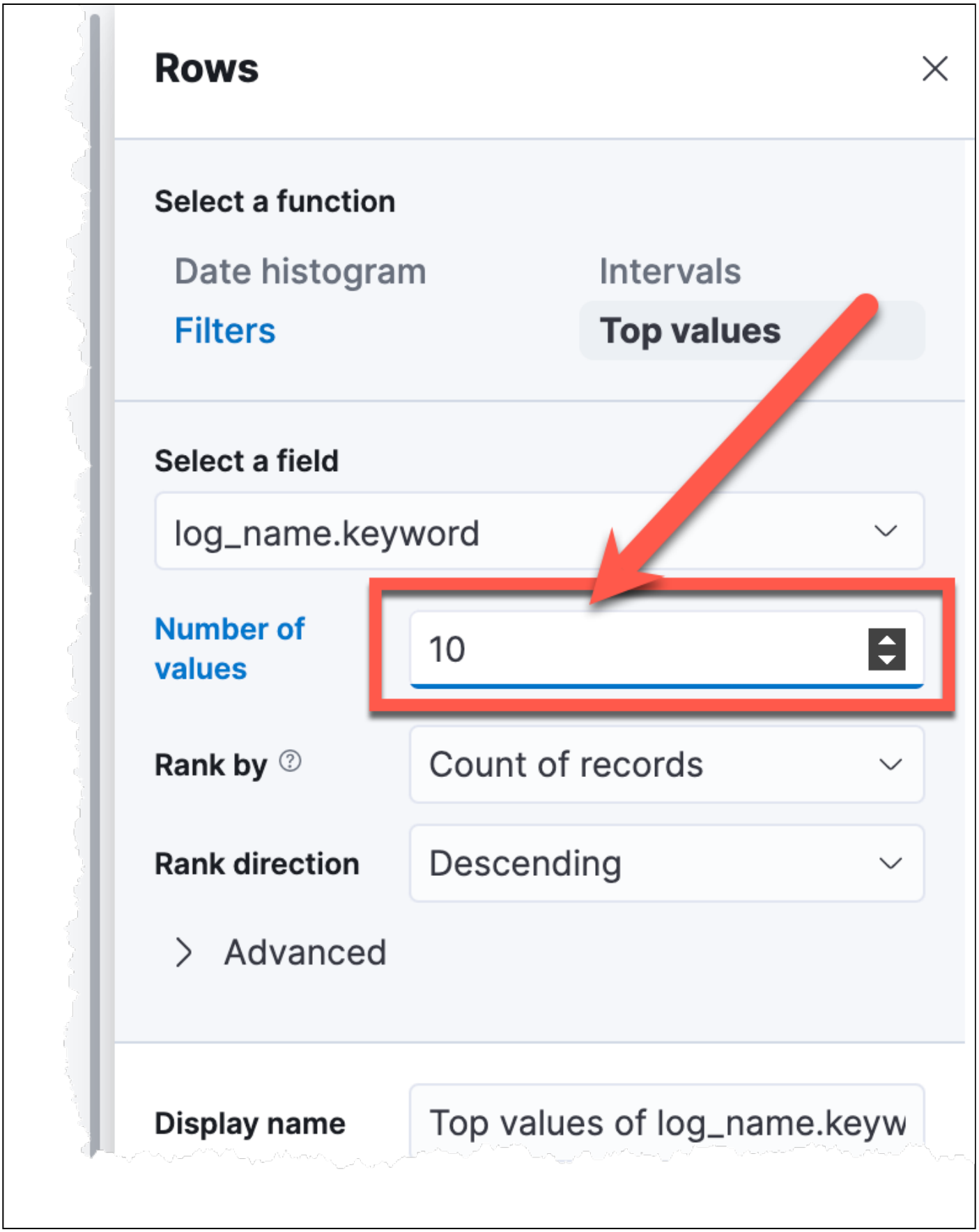
Based on the table that is created inside of the Lens section within Visualize, we can see the top five (5) log types available.

Top values of log_name.keyword	Count of records
projects/suit-ai/logs/syslog	9,200
projects/suit-ai/logs/apache-access	527
projects/suit-ai/logs/cloudaudit.googleapis.com...	463
projects/suit-ai/logs/compute.googleapis.com%...	58
projects/suit-ai/logs/GCEGuestAgent	27
Other	51

We still have 51 log entries that are in the category of 'Other'. This is because, by default, we can only see the top five (5) rows in the 'Data table'. There is a 'Break down by' section with our 'Top values of log_name.keyword' to the right of the table. It's this section that's limiting our table to the top five (5) entries.



Once you click on the 'Top values of log_name.keyword' field, it will expand the configuration menu and allow you to increase the 'Number of values'. For this lab, you can increase the number to 10 to expand out all the logging sources captured by the Google Ops Agent.



Of interest to us as investigators are the `projects/suit-ai/logs/syslog` logs, which are simply the `syslog` output from each VM, and the `projects/suit-ai/logs/apache-access` logs which are the `apache-access` logs from `Apache2` on the VMs.

1. To further understand which VMs are running Apache2, we could also drag over the `compute_resource_name.keyword` field.

What systems were producing `Apache2` logs?

Hint

`Apache2` produces both an "access" and an "error" log. Identify the systems that have both of these logs.

Answer

Top values of log_name.keyword	Top values of compute_resource_nar	Count of records
projects/suit-ai/logs/syslog	instance-1	3,296
projects/suit-ai/logs/syslog	comms-control	2,998
projects/suit-ai/logs/syslog	flight-control	2,906
projects/suit-ai/logs/apache-access	flight-control	289
projects/suit-ai/logs/apache-access	comms-control	238
projects/suit-ai/logs/apache-error	flight-control	16
projects/suit-ai/logs/apache-error	comms-control	10

We can now see from the table that both the `flight-control` and `comms-control` systems were producing `apache-access` logs.

Reviewing Apache Access Logs

1. What was occurring on the systems with Apache Access logs running?

This question will be answered in the **Discover tab**.

Move to the `Discover` tab in Kibana, where we'll run the rest of our searches.

To begin with, clear any existing filters you have applied by clicking the `x` to remove them. Now let's narrow down our events to only the Apache logs with a filter. You can do this by selecting `log_name` in the `Field`, then `is` in the `Operator`, and `projects/suit-ai/logs/apache-access` in the `Value` field.

1. Within these logs, a number of highly malicious activities are occurring.

How many `POST` requests occurred in the logs available?

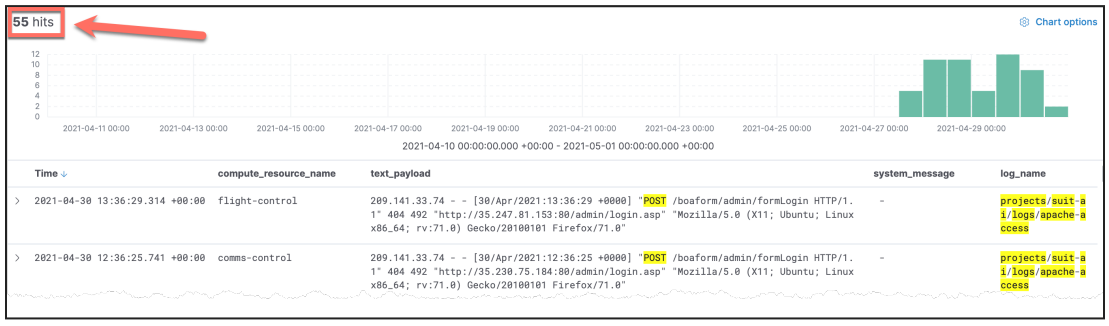
Hint

Because the logs are streamed to the Google Cloud VM Agent, the data is captured in a single field within the Google Cloud logs, so we cannot break them out. Due to this, we have to string-search the data we have available by simply using the term `POST`.

```
text_payload : *POST*
```

Answer

55 Events



How many of the `POST` requests were successfully processed by the Apache server?

Hint

Success in an HTTP request would result in a `200` response code. This, indicates that an Apache server successfully processed the request it received.

```
text_payload : *POST* AND "200"
```

Answer

8 Events

If you search for `POST AND "200"`, you will find there are eight (8) events that used a `POST` request and received a `200` response from the Apache web server.

text_payload
172.245.158.3 - - [29/Apr/2021:21:59:28 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
172.245.158.3 - - [29/Apr/2021:20:32:53 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
45.73.155.207 - - [29/Apr/2021:10:28:30 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
34.229.241.170 - - [29/Apr/2021:04:14:20 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
52.167.54.108 - - [28/Apr/2021:07:17:01 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
45.73.155.207 - - [28/Apr/2021:01:46:00 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.129 Safari/537.36"
64.227.106.142 - - [27/Apr/2021:21:25:00 +0000] "POST / HTTP/1.1" 200 3477 "-" "Anarchy99"

All the IP addresses that managed to issue a `POST` request and get a successful response (`200` response code), also attempted to access another resource on the system. What was it?

Hint

Select one of the search results from the previous question and try to limit your search to only a source IP address to see what else some of the IP addresses were attempting to access.

Answer

`GET /.env`

All the IP addresses also attempted, one or more times, to access the `/.env` resource on both Apache servers. However, when they attempted this, they were unsuccessful. We can determine this from the `404` response code, which means the web server gave an error and didn't return the requested resource.

text_payload

```
172.245.158.3 - - [29/Apr/2021:21:59:28 +0000] "POST / HTTP/1.1" 200 3477 "-" "Mozi
lla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.404
4.129 Safari/537.36"
```

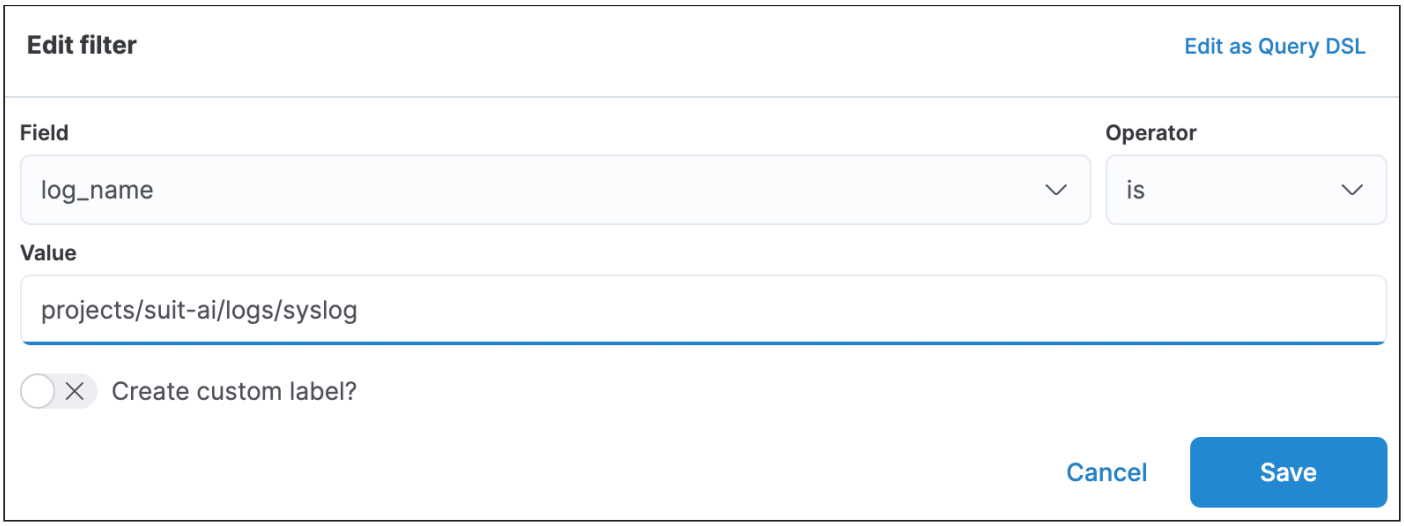
```
172.245.158.3 - - [29/Apr/2021:21:59:27 +0000] "GET /.env HTTP/1.1" 404 492 "-" "Mo
zilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.40
44.129 Safari/537.36"
```

`.env` files often store environment variables used by web applications. It's common for threat actors to scan a web server looking for unsecured `.env` files to read the sensitive information they may contain. For example, this could be passwords, API keys, database credentials, etc. Understanding `.env` files in detail is beyond the scope of this class; however, understanding that threat actors will scan your web application once it's connected to the internet is part of connecting any device to the internet.

Reviewing Syslog Logs

Before we finish with VM Agent logs, let's take a look at the syslog logs. These can be some of the most useful messages when investigating a compromise, but can also be the cause of frustration when we're investigating.

Change the `filter` you have applied so that you are filtering all logs based on the `projects/suit-ai/logs/syslog` `Value` and then click `Save`.



1. Now that we have this filter applied, let's see what we can find within the Syslog logs.

How many `CRON` logs can you find within the available data set?

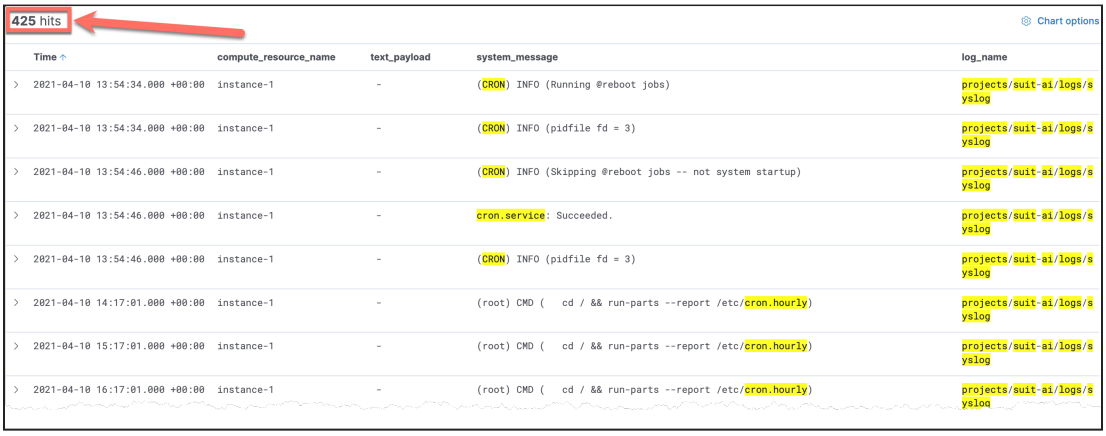
Hint

This can be run simply by using a search string of `"CRON"` or by searching the two message fields:

```
text_payload : *CRON* OR system_message : *CRON*
```

Answer

425 Event Logs



You will notice that some of the earliest events that appear don't have their Google Cloud VM Agent logs in the `text_payload` field. Instead, they use the `system_message` field, which is produced by SOF-ELK's parsers. They are actually in the the Google Cloud JSON field `jsonPayload.message`. We're not really sure why the location of the Google Cloud VM Agent output changed during our log capture, however, this is a good example to understand that Google Cloud can change the logging format.

Can you find any unusual SSH login activity, and, if so, why is it unusual?

Hint

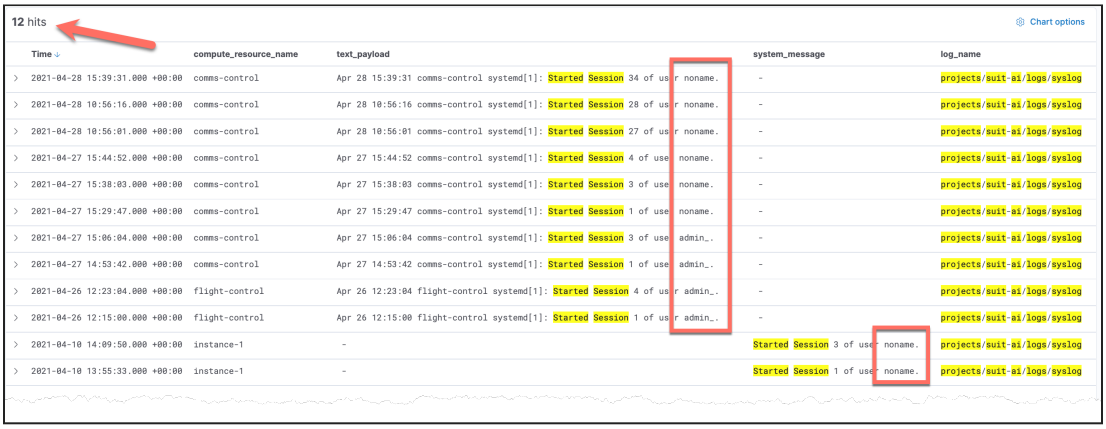
Simply searching for `SSH` or `SSHD` doesn't return many useful results. In fact, it looks like there is no SSH logging that is being pushed to Syslog. This is because, by default, SSH will log to `/var/log/sshd/sshd.log`, which is not captured by Google Cloud's VM Agent. There are ways to force SSH logging into Syslog, or you could alter Google Cloud's VM Agent config to include the `/var/log/sshd/sshd.log` file.

As a second step, we can try to look for authentication events within Syslog. The easiest way to look for these is with the string search `"Started Session"` in the Search Bar.

```
text_payload : "Started Session" OR system_message : "Started Session"
```

Answer

When you search for `"Started Session"`, you should see **12** login events with both the usernames `admin_` and `noname`.



How were the two user accounts created, and when were they created?

Hint

If you search for both the usernames in the search bar, `"noname" OR "admin_"`, you will be provided with an overview of activity with those user accounts.

You will notice that both the user accounts were created with the `GCEGuestAgent`. This is the built-in Google Cloud agent that looks after integration between a VM and the Google Cloud architecture (e.g., to enable web console actions). This agent would have created these user accounts as part of an instruction from the platform. This could be either due to Policy or due to Project-wide SSH keys that were implemented by a Policy.

Answer

Looking at the logs, we can see the following accounts were created;

a. `flight-control` on `2021-04-26 11:47:50.000 UTC`

b. `comms-control` on `2021-04-27 14:52:06.000 UTC`

c. `instance-1` on `2021-04-10 13:54:45.000 UTC` for `noname`

d. `instance-1` on `2021-04-10 14:37:18.000 UTC` for `admin_`

Time	compute_resource_name	text_payload	system_message
2021-04-27 14:52:13.000 +00:00	comms-control	Apr 27 14:52:13 comms-control GCEGuestAgent[723]: 2021-04-27T14:52:13.9644Z GCEGuestAgent Info: Creating user admin_.	-
2021-04-27 14:52:06.000 +00:00	comms-control	Apr 27 14:52:06 comms-control GCEGuestAgent[723]: 2021-04-27T14:52:06.5867Z GCEGuestAgent Info: Creating user noname.	-
2021-04-26 11:47:51.000 +00:00	flight-control	Apr 26 11:47:51 flight-control GCEGuestAgent[715]: 2021-04-26T11:47:51.5508Z GCEGuestAgent Info: Creating user admin_.	-
2021-04-26 11:47:50.000 +00:00	flight-control	Apr 26 11:47:50 flight-control GCEGuestAgent[715]: 2021-04-26T11:47:50.9947Z GCEGuestAgent Info: Creating user noname.	-
2021-04-10 14:37:18.000 +00:00	instance-1	-	2021-04-10T14:37:18.7242Z GCEGuestAgent Info: Creating user admin_.
2021-04-10 13:54:45.000 +00:00	instance-1	-	2021-04-10T13:54:45.3675Z GCEGuestAgent Info: Creating user noname.

A more targeted search would be:

```
("noname" OR "admin_") AND "creating"
```

Filtering Clean Up

Ensure you clear out any filters you may still have applied for this lab. This will ensure your next lab isn't impacted by any filters you used in this lab.

Key Takeaways

- You should understand how Google Cloud VM Agent logs are recorded and how they can be manipulated to investigate services running on VMs.
- You should have a better understanding of how to identify individual VMs and know that they do not always appear in the logs with human-defined names.

- You should have an understanding of how to quickly identify servers that are producing logs on a VM and how to then dive into those logs for further analysis.

Lab 5.3: Storage Abuse and Exfil

Objectives

- Understand Google Cloud Storage Logging format.
- Understand what to look for when analyzing Google Cloud Storage logs for misuse.
- Be able to link misuse events back to user accounts in order to track a timeline of behavior.

Background

LongConSecurity has received reports that highly confidential data has been taken from the organization. LongConSecurity knows that the data only existed in a locked-down Google Cloud Bucket, which it believes to be limited to only users with Organization Admins access. All the users with access to this Bucket are trusted, and LongConSecurity does not suspect any of them as the source of the data being taken.

Preparation

To prepare:

1. Make sure your SOF-ELK® VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-04-12 00:00:00** to **2021-04-28 00:00:00 UTC**.

Load Data

The Google Cloud JSON logs for this lab have already been exported from Google Logging. To help in keeping the amount of data to a reasonable level, we have trimmed the data set to only include Google Cloud Bucket logs. However, there were no other changes made to the JSON that was extracted from Google Cloud, so it is still representative of what you'll experience outside of this course.

The Google Cloud Bucket logs have been saved into `/home/elk_user/lab-5.3_source_evidence.zip`.

1. Log on to the SOF-ELK VM with the following credentials
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract all the `.json` files from the preceding ZIP file into the correct Logstash folder for parsing by FileBeats with the following command:

Command lines

```
cd ~
unzip lab-5.3_source_evidence.zip -d /logstash/gcp/
```

Warning

Do not run this command more than once!

3. Wait 2 minutes for the data to be processed.
4. Verify that you now have 10,480 documents loaded in the `gcp` index:

Command lines

```
sof-elk_clear.py -i list
```

Note

Your document count will be different if you haven't completed the previous labs.

Expected results

```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- gcp (10,480 documents)
- gws (1,157 documents)
- httpdlog (1,381 documents)
- netflow (258,499 documents)
- office365 (5,462 documents)
```

5. Once it is complete, the data will be available in the `gcp-*` index.

Lab Content

Review the Indices in SOF-ELK

Start in the Discover tab.

When data is imported in SOF-ELK, it's assigned to a specific index. In this lab, and in all other labs in this section, we will be using the `gcp-*` index.

This index contains all the Google Cloud logs you will investigate in this section of the course, except for Flow data.

Before we go too far, let's ensure the date range is correct for the data we'll be using in this lab. Set your search time frame to `2021-04-12 00:00:00.000 +00:00` to `2021-04-28 00:00:00.000 +00:00` and hit `Update`. Based on this time frame, you should see **7,349 hits** entries in your histogram.

> **Notice**
>
> The "hits" count assumes you have loaded Lab 4.1 and Lab 4.2 already.

The initial view of the data within the Discovery view of SOF-ELK is very unstructured. In the first few steps in this exercise, we will add in specific columns to make the data more useful for reviewing Google Cloud Bucket logs.

Let's start by filtering down our log data to only the relevant information for buckets. We can do this with the `Add filter` feature. Let's add a filter that checks whether the `bucket_name` field `exists`. This will result in only showing us the Google Cloud Bucket logs.

Note

If any of the below field names do not appear when you search for them, try instead running an open search in the main search bar at the top of the Discover page. For example, if `bucket_name` doesn't appear, search for this:

```
bucket_name : *
```

After you do this, the field you just searched for should appear above, as shown in the animation.

This issue occurs because Kibana only shows fields for the first 500 events, by default. Once you run a search the missing field will be in the newly loaded 500 events on screen and the field appears.

Next, we will add in the following column names, in the order shown:

- `bucket_name`
- `resource_name`
- `username`
- `method_name`
- `severity`

To add in these names, you can search for them in the field selector under the index dropdown menu. When you start typing the field name, it will appear for you to select with `+`, which appears when you hover your mouse over the field name.

Provided you added the fields in the specified order, your events in Discovery should now appear similar to the following image.

We're now ready to start looking through our Google Cloud Bucket logs.

Discover Unusual Activity in Google Cloud Bucket Logs

1. Part of reviewing Google Cloud Bucket logs involves looking for potential failed access to objects or failed permissions for tasks. We're going to see what we can discover when we look for this type of data in the logs.

There are two groupings of events in our histogram. Using the search field, when did the `ERROR` events start and finish?

Hint

You will need to use the search field to look for `severity: "ERROR"`.

Answer

You will see **18 hits** for data that match this search, starting from `2021-04-26 13:37:52.405 UTC` and finishing on `2021-04-26 13:41:13.508`.

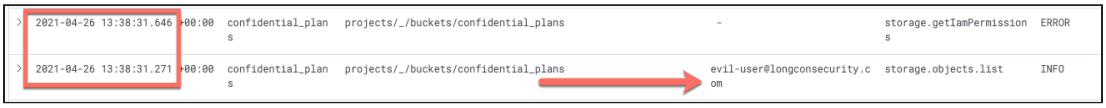
Are there any events preceding these `ERROR` events that involve authenticated user accounts, or account information that we could further investigate?

Hint

Identify any user accounts in the `username` field without altering the search from the previous question. Then try looking for the `username`'s immediately before all of the `ERROR` events.

Answer

You will have noticed that immediately before all of these `ERROR` events, one user account continually appears: `evil-user@longconsecurity.com`.



What Buckets has the user from the previous question been looking at?

Hint

You will need to use the search field with the user account we identified in the previous question, `username: "evil-user@longconsecurity.com"`.

Answer

Because the amount of data we are analyzing has been trimmed down for the exercise, it is easy to quickly scroll through the events to identify the `bucket_name` that appear.

bucket_name	resource_name	username
confidential_plans	projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip	evil-user@longconsecurity.com
confidential_plans	projects/_/buckets/confidential_plans	evil-user@longconsecurity.com
flight-maps	projects/_/buckets/flight-maps	evil-user@longconsecurity.com
flight-maps	projects/_/buckets/flight-maps	evil-user@longconsecurity.com
flight-maps	projects/_/buckets/flight-maps	evil-user@longconsecurity.com
confidential_plans	projects/_/buckets/confidential_plans	evil-user@longconsecurity.com

The two `bucket_names` that were accessed are; `confidential_plans` and `flight-maps`.

What, if any, files were accessed by the user from the previous questions?

Hint

You will need to use the search field and build on the previous searches we've used so far. By adding in `method_name: "storage.objects.get"` to our search, we will see the files that were downloaded by the user.

The overall search should be:

```
username: "evil-user@longconsecurity.com" AND method_name: "storage.objects.get"
```

Answer

Once you have run the search form the Hint section above, you will see there is only **1** file that appears to have been accessed by the user account `evil-user@longconsecurity.com`, and that file is listed in the `resource_name` field as `projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip`.

bucket_name	resource_name	username	method_name
confidential_plans	projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip	evil-user@longconsecurity.com	storage.objects.get

We can also see that this occurred in the `confidential_plans` Bucket.

How Did a Google Cloud Bucket Get Exposed?

1. In the previous section, we discovered that a user account that likely shouldn't have had permission to a Bucket was able to access and download a file. We now need to see if the logs will show us how this may have occurred.

When was the `Suit_CPU_Design.zip` file first added to the Bucket, and who added it?

Hint

You will need to use the search field, looking for the file we previously identified. Your search should be:

```
resource_name : "projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip"
```

Answer

Once this search is completed, you will see that the earliest event we have for this file is `2021-04-18 14:53:07.306 UTC`.

You will also see that the user that appears in this first event is `admin@longconsecurity.com`.

You'll notice that the first event doesn't contain detail along the lines of "file created". Instead, you see a `storage.objects.update`. This is how Google Cloud represents a new or updated item within a Bucket. It is also possible the item existed earlier than the time window we are looking at; however, it would still only show `storage.objects.update`.

Were there any permission changes within the `confidential_plans` Bucket that may explain why it was accessible to other users?

Hint

The best way to go looking for this type of data is using the `method_name` to see if there is anything related to a permission change.

Initially, you may try looking for `method_name: *IamPermissions` in the search bar. However, you will notice that this returns 38 entries.

You could try to narrow this search down by excluding all the `storage.getIamPermissions`, as we aren't interested in when people queried the permissions at this stage, just when they were changed. To do this, use this search:

```
method_name: *IamPermissions AND NOT method_name : storage.getIamPermissions
```

Answer

You will see one entry. If you expand out the entry, you will see there is a `policy_deltas` event that lists out the changes made during a `setIamPermissions` event.

t log.flags	multiline
# log.offset	115,979
t method_name	storage.setIamPermissions
t policy_deltas.action	ADD
t policy_deltas.member	allUsers
t policy_deltas.role	roles/storage.legacyBucketReader
t project_id	suit-ai
t resource_location	us-west1
t resource_name	projects/_/buckets/confidential_plans
t resource_type	gcs_bucket

This shows that the `admin@longconsecurity.com` user account set the `confidential_plans` Bucket to allow `allUsers` to access files within the Bucket.

Pivoting on Our Findings

1. Based on what we've found so far, can we now pivot on this information to see if anything else has gone wrong with our Google Cloud Buckets?

What other Buckets did `admin@longconsecurity.com` create or update?

Hint

The best way to go looking for this is narrowing a search to only the user you're looking for and to look for both the create and update methods for Buckets.

You can use this search:

```
username:"admin@longconsecurity.com" AND method_name : (storage.buckets.update OR
storage.buckets.create)
```

Unlike files in a Bucket, Buckets will have the `create` method for them.

Answer

You will see two matching Buckets, the `confidential_plans` Bucket created on `2021-04-18 12:45:28.808` and the `flight-maps` Bucket with the earliest update time of `2021-04-13 01:41:49.225`.

The `flight-maps` Bucket was created outside of the log data we have available within SOF-ELK.

Were there any other permissions applied to any Storage Bucket objects that would have allowed the `allUsers` permission?

Hint

Using the inforamtion we observed in the previous question, we can search `policy_deltas.member` in SOF-ELK looking for `allUsers` in the search bar.

```
policy_deltas.member : allUsers
```

Answer

Using the search from the Hint above, we find there are in fact **3** resources within the Google Cloud Bucket that have had their permissions all set to `allUsers`.

	Time	bucket_name	resource_name
>	2021-04-26 12:01:39.070 +00:00	confidential_plans	projects/_/buckets/confidential_plans
>	2021-04-18 14:53:07.306 +00:00	confidential_plans	projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip
>	2021-04-13 04:01:19.989 +00:00	flight-maps	projects/_/buckets/flight-maps/objects/test-flight/test_flight_route.jpeg

Using the same search you just ran for the last question, what `method_name` is used for files within Buckets to set the `allUsers` permission?

Answer

You should see that files within Buckets use the `storage.objects.update` `method_name` when a permission is changed. This should seem familiar now given the same `method_name` is used when adding a file to a Bucket.

resource_name	username	method_name	severity
projects/_/buckets/confidential_plans	admin@longconsecurity.com	storage.setIamPermissions	NOTICE
projects/_/buckets/confidential_plans/objects/Suit_CPU_Design.zip	admin@longconsecurity.com	storage.objects.update	NOTICE
projects/_/buckets/flight-maps/objects/test-flight/test_flight_route.jpeg	admin@longconsecurity.com	storage.objects.update	NOTICE

Filtering Cleanup

Ensure you clear out any filters you may still have applied for this lab. This will ensure your next lab isn't impacted by any filters you used in this lab.

Key Takeaways

- You should now understand that seeing `ERROR` messages within Google Cloud Bucket logs usually indicates unusual requests being made by users who likely don't have permissions to access certain resources.
- Resources that have had the `setIamPermission` applied indicate a change in the permissions on those resources.
- You should now understand how to track down permission changes to files and Buckets when they have been mistakenly exposed.
- Additions and IAM changes to files within Buckets don't have a dedicated "Create" or "IAM Change" type of method in the same way that Buckets do.

Lab 5.4: Google Cloud: Network Forensics

Objectives

- Understand the data that is captured by Google Cloud Flow logs and how to quickly search within them to aid an investigation.
- Understand how to import Google Cloud Flow logs into SOF-ELK and the pre-made dashboard available for searching Flow logs.
- Look for malicious network activity within Flow logs and understand the investigation value they can provide.

Background

Inside the Google Cloud instance for LongConSecurity.com was an unusually large spike in traffic on the evening of April 27, 2021 (UTC). The Google Cloud Network Admins have extracted the Google Cloud Flow logs for the VMs that were inside the `default` VPC and have provided you with those logs from just before and after the large increase in network traffic.

Preparation

To prepare:

1. Make sure your SOF-ELK® VM is running.
2. Access Kibana via the IP address shown on the SOF-ELK® VM.
3. Access the SOF-ELK® VM console via the user account `elk_user` and the password `forensics`.
4. Ensure you have loaded the relevant data as described in the next section.
5. Relevant timeframe: **2021-04-27 13:00:00 UTC** to **2021-04-27 21:00:00 UTC**.

Load Data

The Google Cloud JSON logs for this lab have already been exported from Google Logging. To help in keeping the amount of data to a reasonable level, we have trimmed the data set to only include Google Cloud Flow Logs. However, there were no other changes made to the JSON that was extracted from Google Cloud, so it is still representative of what you'll experience outside of this course.

The Google Cloud Flow Logs have been saved into `/home/elk_user/lab-5.4_source_evidence.zip` within your class virtual machine.

1. Log on to the SOF-ELK VM with the following credentials:
   - Username: ***elk_user***
   - Password: ***forensics***

   You may log on to the console; however, it's recommended to log in via SSH.
2. Extract all the `.json` files from the preceding ZIP file into the correct Logstash folder for parsing by FileBeats with the following command:

Command lines

```
cd ~
unzip lab-5.4_source_evidence.zip -d /logstash/nfarch/
```

Warning

Do not run this command more than once!

3. Wait 2-4 minutes for the data to be processed. Remember, there is enrichment occurring with the IP addresses and the GeoIP lookups as the data is loaded, so this may take a few minutes longer than the data from previous labs to load.
4. Verify that you now have 351,151 documents loaded in the `netflow` index:

Command lines

```
sof-elk_clear.py -i list
```

Expected results

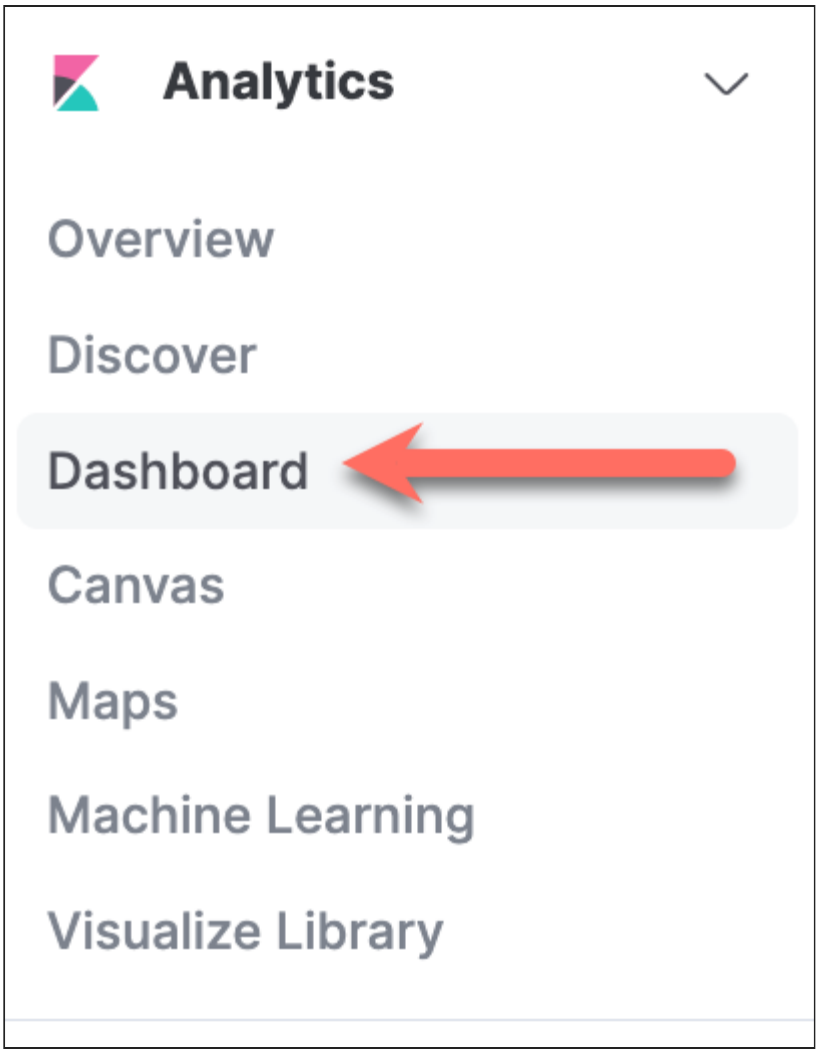
```
[elk_user@sof-elk ~]$ sof-elk_clear.py -i list
The following indices are currently active in Elasticsearch:
- aws (240,899 documents)
- azure (6,217 documents)
- gcp (10,480 documents)
- gws (1,157 documents)
- httpdlog (1,381 documents)
- netflow (351,151 documents)
- office365 (5,462 documents)
```

5. Once it is complete, the data will be available in the `netflow-*` index.

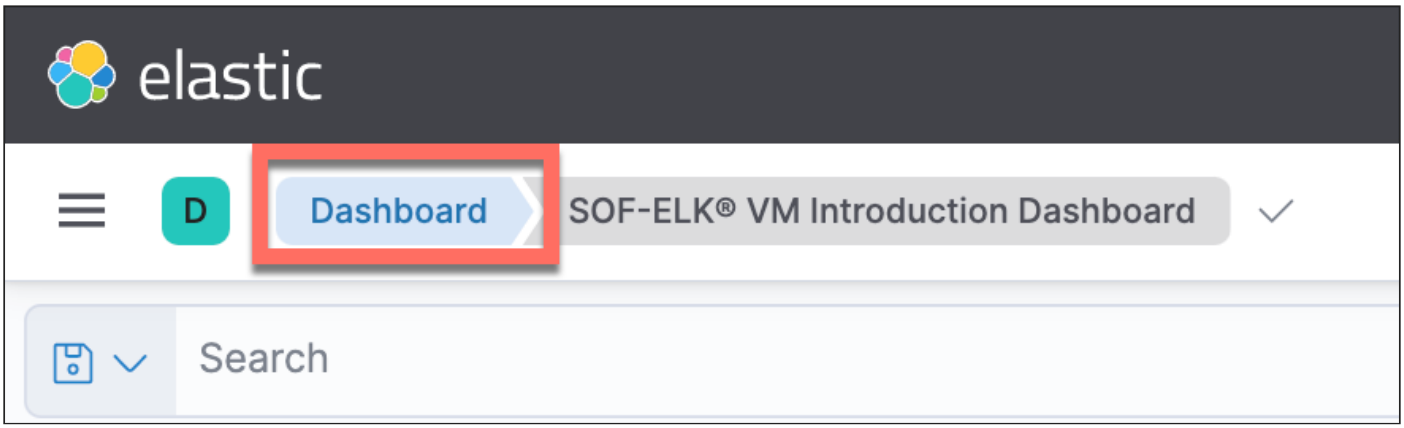
Lab Content

Start in the **Dashboard** tab.

For this lab, we're going to work mainly in the NetFlow dashboard, so you need to ensure you've cleared out any searches you had previously in SOF-ELK and navigate to the **Dashboard** tab:



Click on the word `Dashboard` to get the dashboard menu:



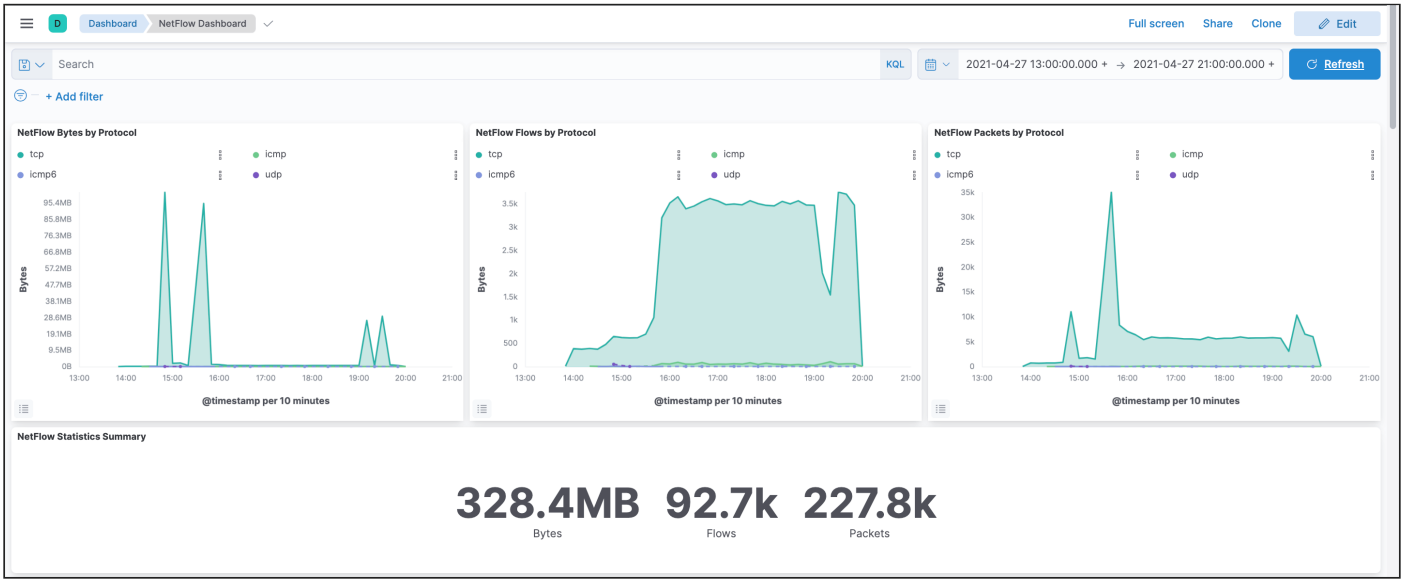
Select the **NetFlow** dashboard:

Dashboards

Search...

Title	Description
Azure Activity Logs	
DNS Dashboard	
Eventlog Dashboard	
Filesystem Dashboard	
HTTPD Log Dashboard	
LNK File Dashboard	
Login Activity Dashboard	
NetFlow Dashboard	
SOF-ELK® VM Introduction Dashboard	
Syslog Dashboard	

Let's ensure we have all the logs loaded correctly. You should see the following results in the top of your NetFlow dashboard once you filter the time to `2021-04-27 13:00:00.000 +00:00` -> `2021-04-27 21:00:00.000 +00:00`



Warning

If you do not see the same data displayed in your SOF-ELK NetFlow Dashboard, you should stop at this point. Either you are not looking at the correct time frame or your data hasn't finished loading or hasn't loaded correctly.

Discover the Google Cloud Flow Logs We Have Available

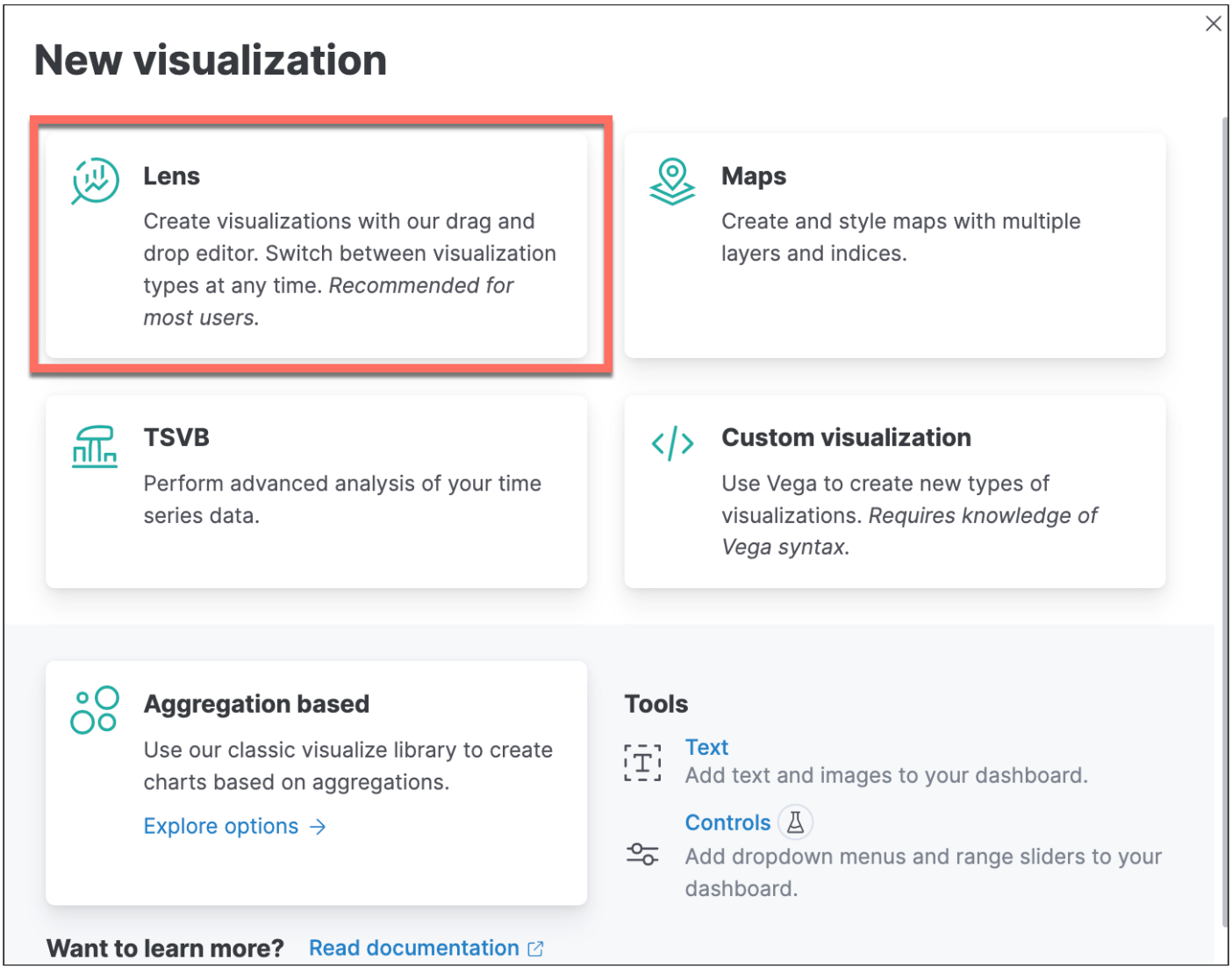
This section is in the **Visualize** tab

Let's start by looking at the Google Cloud Flow Logs that have been loaded, so we know where the logs have come from before we start to dive into them looking for malicious activity.

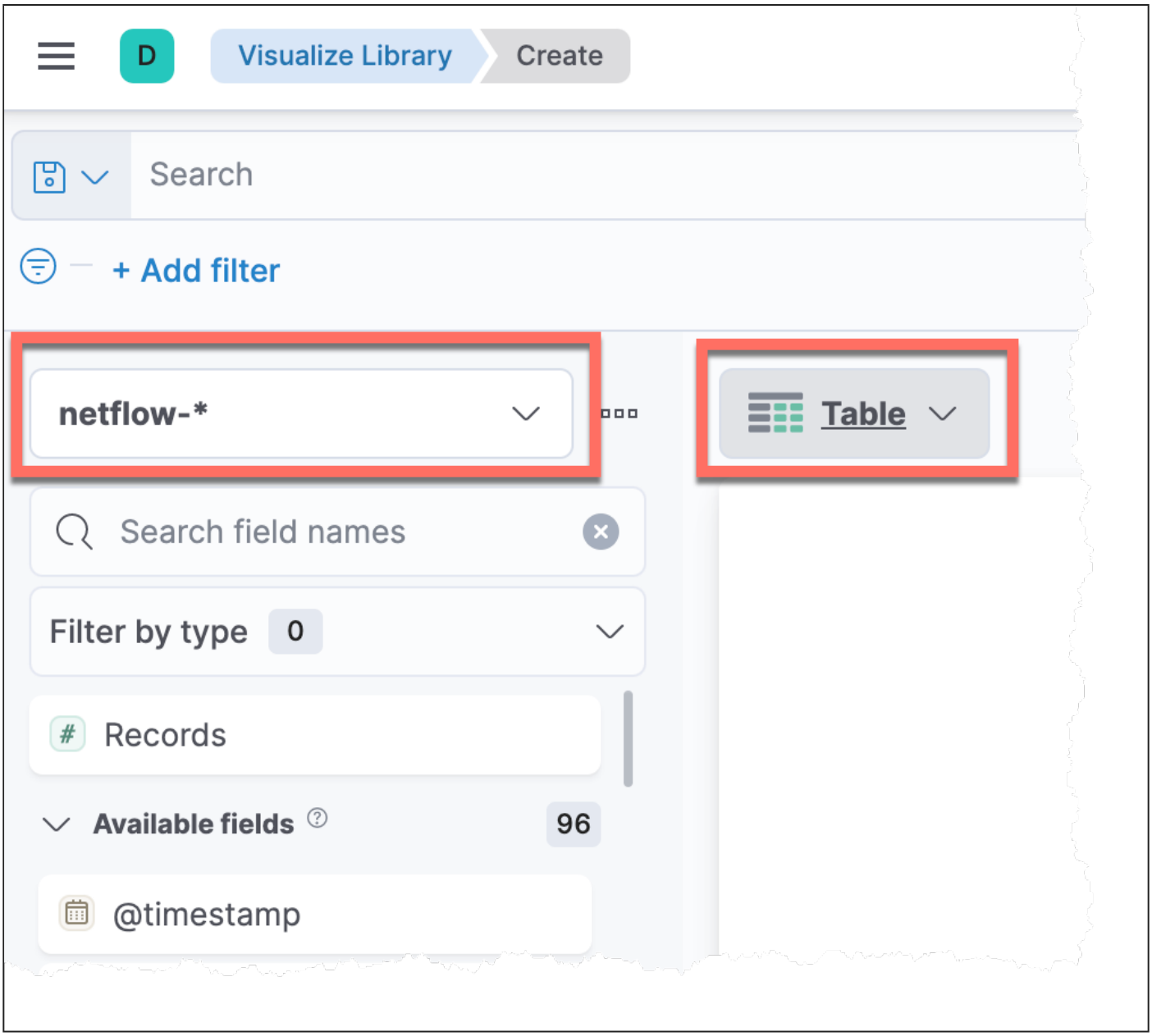
Because the NetFlow Dashboard in SOF-ELK is designed to look at all types of Flow data, it doesn't have Dashboard Widgets that show the VMs or the VPC that the data originated from. We'll need to do this ourselves using the Visualize dashboard.

Move to the `Visualize` tab in Kibana and select `Create visualization`.

We will use `Lens` because it's the easiest way to create a new visualization.

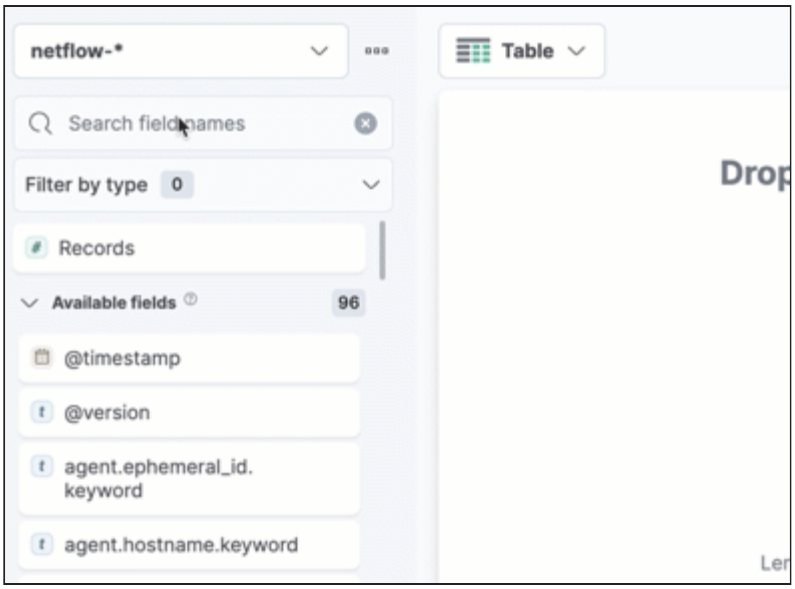


We will need to move the index selector over to `netflow-*` to ensure we are looking at the Google Cloud Flow data. We will also change the graph type to `Data table`.



Within each Google Cloud Flow record is a field called `destination_vm_name.keyword` or `source_vm_name.keyword` as well as a record called `destination_vpc_name.keyword` or `source_vpc_name.keyword`. The difference in whether there is a `source_` or `destination_` is based on the direction of the traffic. Flow records are not bidirectional, they only record traffic in one direction, so a TCP connection would have two separate Flow records.

To first understand what data we're looking at, let's start by dragging into our table visualization the `destination_vm_name.keyword` field and the `destination_vpc_name.keyword` field.



1. Based on the table we've just created:

How many inbound Flow records are available for each VM and what is the VPC that these VMs are contained within?

Hint

Review the "Count of records" fields in the visualization table you just created.

Answer

This should now allow us to see is that there are three VMs in the Google Cloud Flow Logs:

- comms-control = **7,105 records**
- flight-control = **4,155 records**
- pcap-capture = **2,991 records**

All of these VMs are within the `default` VPC.

Table

Top values of destination_vm_name.keyword	Top values of destination_vpc_name	Count of records
comms-control	default	7,105
flight-control	default	4,155
pcap-capture	default	2,991

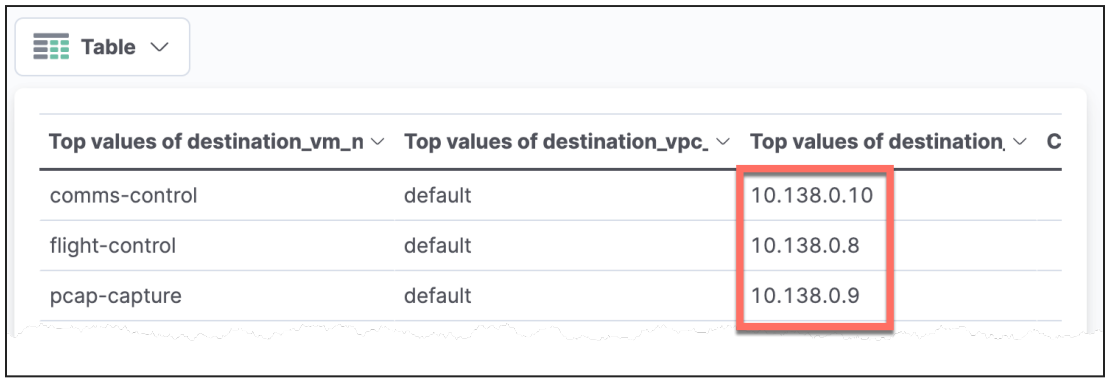
2. To the table we just created, add the `destination_ip` field.

What are the corresponding internal IP addresses for the VMs we identified earlier?

Answer

After adding in the `destination_ip` field, you should now see the following IP addresses match up with the VMs:

- comms-control = **10.138.0.10**
- flight-control = **10.138.0.8**
- pcap-capture = **10.138.0.9**



This information may seem trivial now; however, it will be useful in the next section when we look at all three of the VMs' traffic together.

Discover Unusual Traffic in the Google Cloud Flow Logs We Have Available

This section is in the **NetFlow Dashboard** tab.

Notice

All the screenshots in this section are of widgets within the NetFlow Dashboard. They don't show the whole dashboard, only the widget where the answer would be found. This has been done to make it easier to read.

Now that we know what VMs are within the Flow logs, along with the VPC they were contained in, we can now start to dive into the network traffic data. Move back over to the NetFlow dashboard, where we'll look at the Flow records in more detail.

1. Looking at the top three histogram graphs, we can see a visible rise in network traffic.

What VM (instance name and IP) is most likely responsible for the spike in Flow traffic?

Hint

When you filter on the three IP addresses identified earlier, you want to try and include both inbound and outbound traffic for the VMs. The best way to do this is with an `OR` statement in the search field of the dashboard.

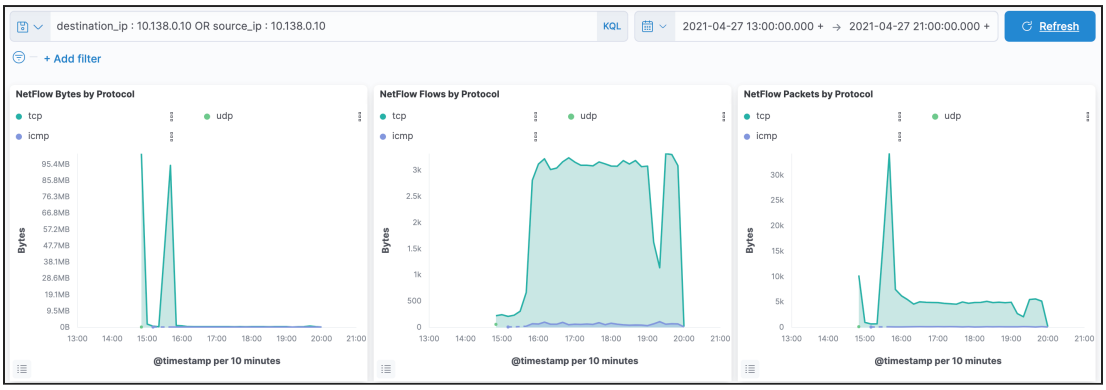
When you search on IP `10.138.0.10` using

```
destination_ip : 10.138.0.10 OR source_ip : 10.138.0.10
```

Answer

You can see there is a notable increase in traffic based on the top three histogram graphs. This system, which matches back to `comms-control`, appears to be the VM with a very unusual uptick in traffic.

Additionally, this `comms-control` VM appears to have almost 10 times more Flow records for the same period of time compared to the other two VMs.



2. Let's start to look at the type of traffic occurring with the VM we just identified. Leave the search filter mentioned in the preceding answer in the search bar so we can start to look at what type of traffic was being used by this VM.

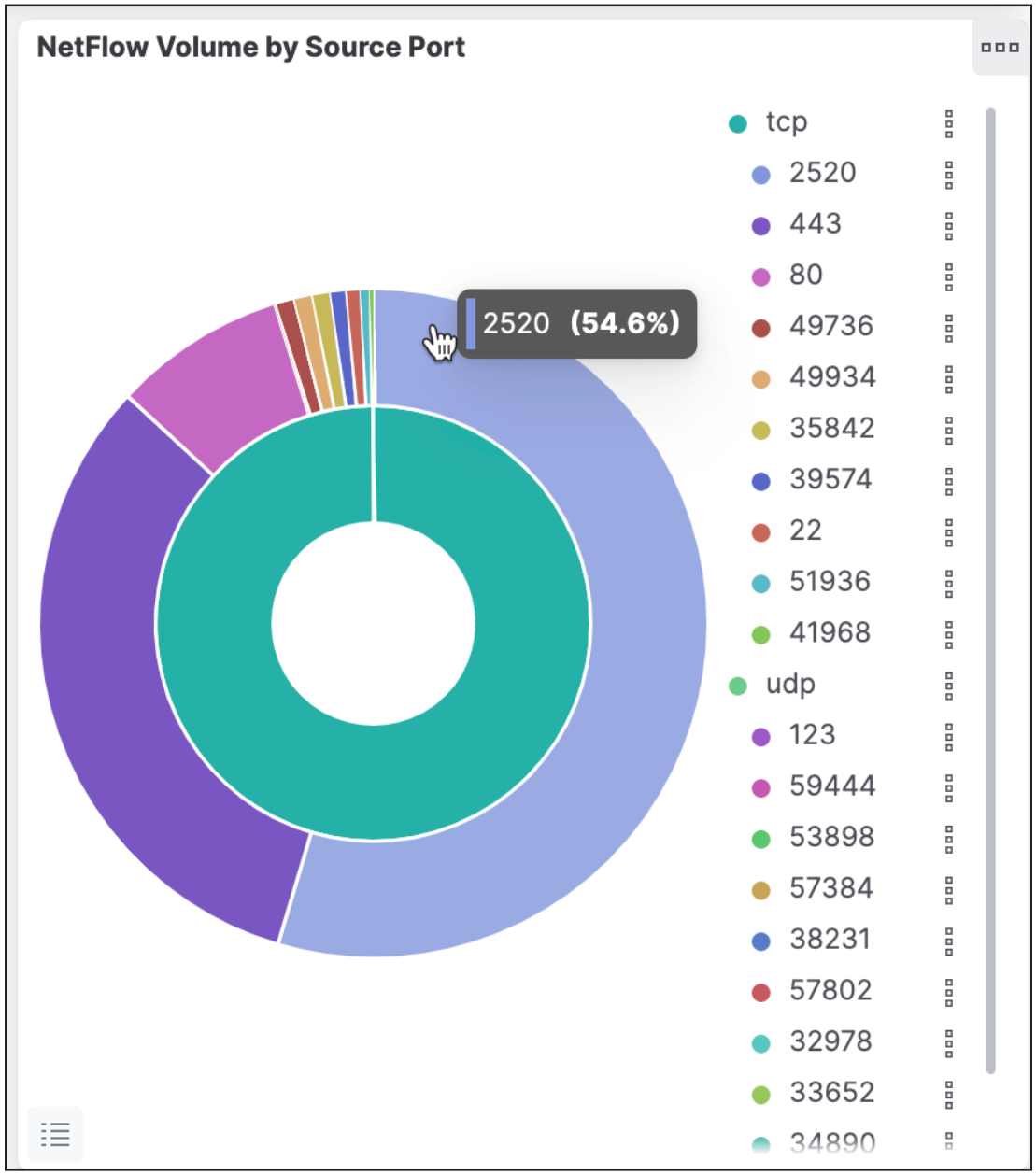
Which traffic type (TCP and Port) appears to be the result of the increased traffic?

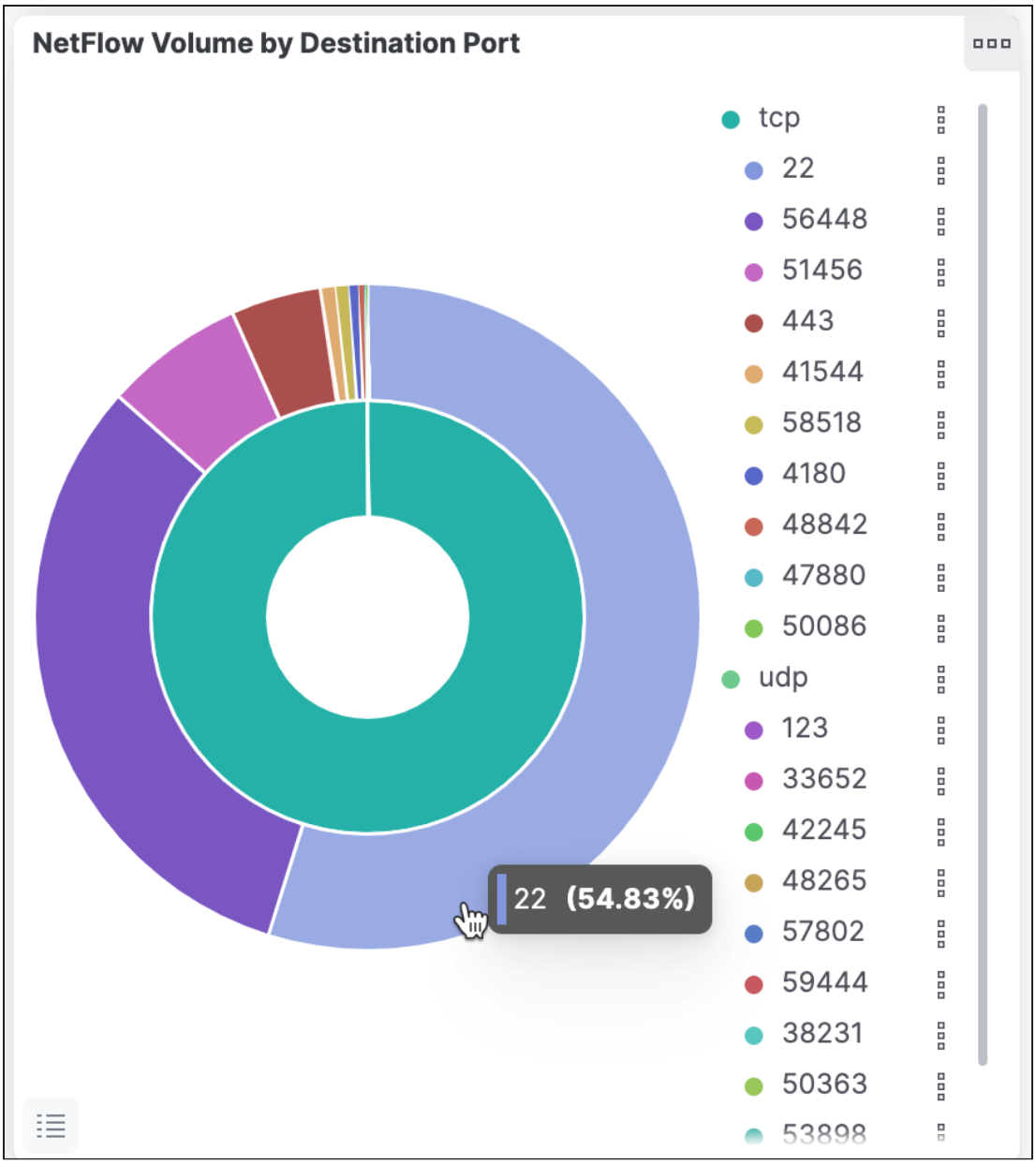
Hint

Scroll down the NetFlow dashboard and use both the `NetFlow Volume by Source Port` and `NetFlow Volume by Destination Port` donut graphs looking for the larges slice on each of them.

Answer

When viewing both `NetFlow Volume by Source Port` and `NetFlow Volume by Destination Port`, we can see there is only one slice of the ring graph that consumes a little over 54% of the traffic.





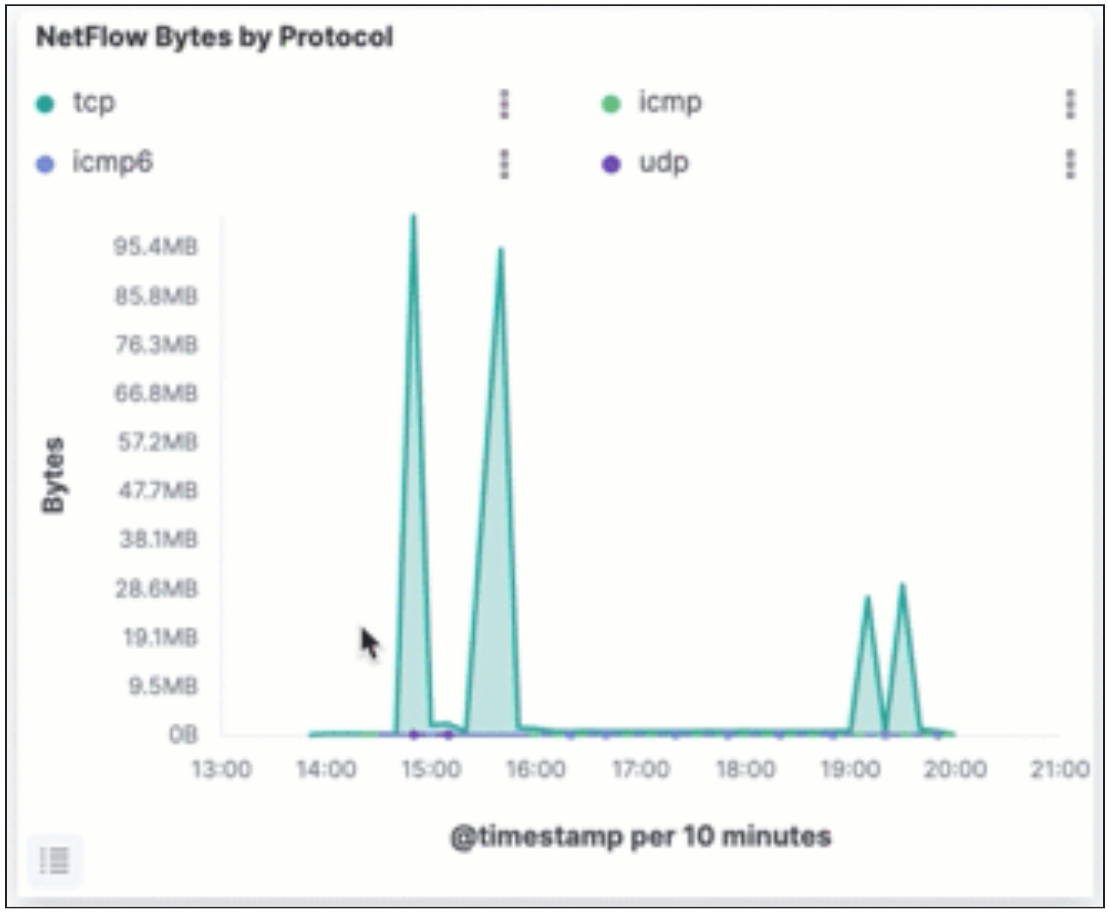
Based on the pie graphs, this appears to be:

- Source Port: `TCP 2520`
- Destination Port: `TCP 22`

When does the traffic start to significantly increase for the VM and traffic type we've identified?

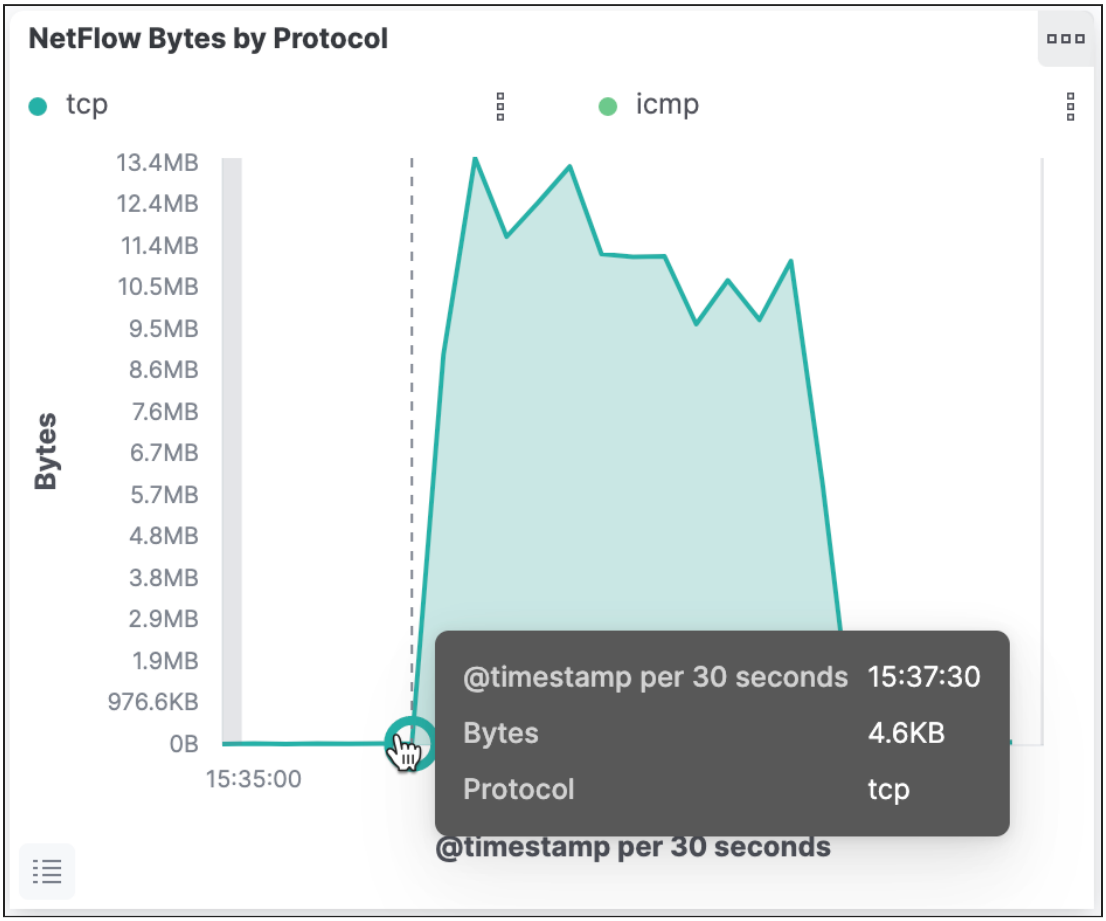
Hint

Using the histogram graphs in the top of the dashboard, zoom into the traffic spikes until you can easily see a start and stop time for the traffic. Then, hover over the start of the traffic increase to identify an approximate time.

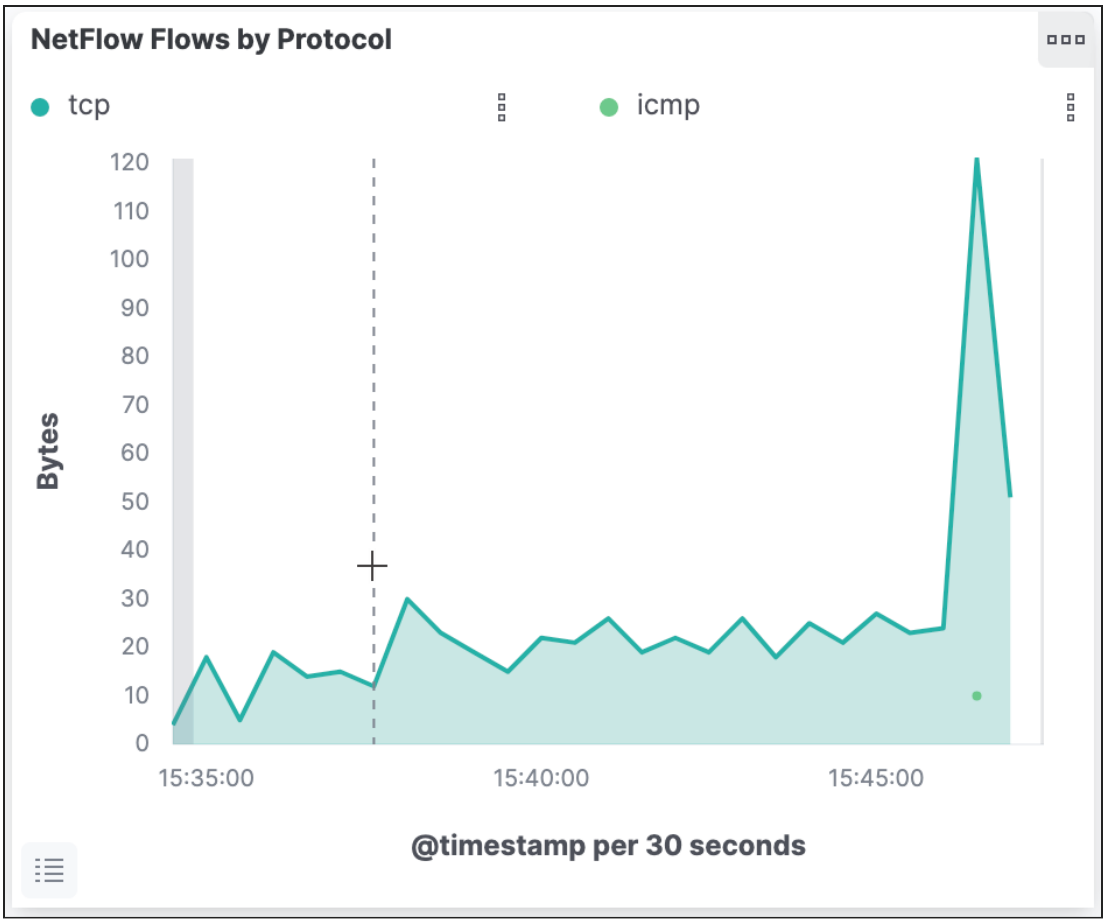


Answer

At approximately **2021-04-27 15:37:30 UTC** there was a traffic spike, in terms of the amount of bytes observed.



At approximately **2021-04-27 15:38:00 UTC** there was a noticeable change in the quantity of Flow records produced for outbound TCP 22 traffic.



3. If you altered your search time frame for the preceding question, reset it back to `2021-04-27 13:00:00.000 +00:00` -> `2021-04-27 21:00:00.000 +00:00`.

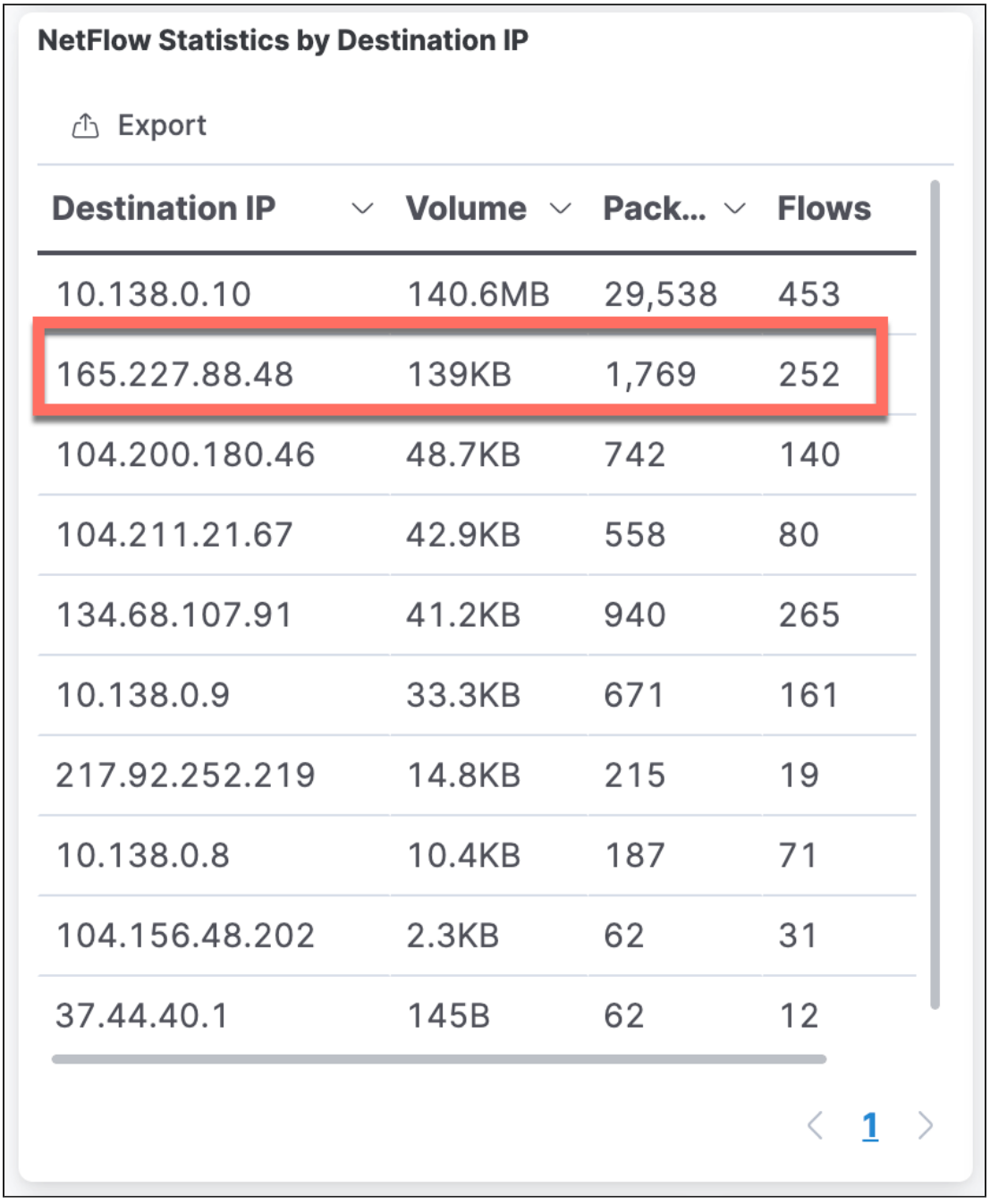
There is one external destination IP address that appears to have a lot more TCP 22 traffic than any others within our Flow logs. What is it?

Hint

To narrow down the Flow logs to only show outbound TCP 22 traffic, you would need to apply the `destination_port : 22 AND aprotocol.keyword : tcp` filter in the search bar.

Answer

The IP address `165.227.88.48` appears to communicate a lot more than any of the other IP addresses in our Flow logs.



Because we're only looking for *external* IP addresses, we would exclude the host's own IP address of `10.138.0.10`.

4. Given that we are seeing a lot of outbound traffic on TCP 22, let's narrow our search to only show outbound traffic from our suspicious VM.

Are there any other unusual outbound connections that we haven't already identified, and why have these not really appeared in our dashboards yet?

Hint 1

If we just search for `10.138.0.10` as a `source_ip`, we can filter to just look at outbound Flow logs.

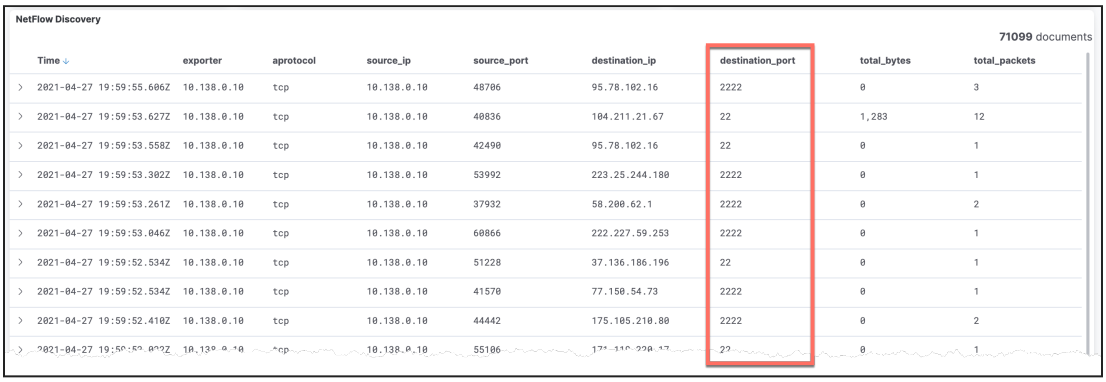
```
source_ip : 10.138.0.10
```

Hint 2

Reviewing the dashboard's widget for `NetFlow Volume by Destination Port` does show `TCP 443`. However, when you filter on this traffic type, you will notice that the overwhelming majority of this traffic is destined for Google. In fact, it's actually outbound HTTPS traffic from the VM back to Google Cloud. This is normal, and in fact is actually the means for how the logs from Lab 4.2 are produced. So you can exclude this traffic.

Answer

If you scroll further down to the individual Flow record entries at the bottom of the NetFlow dashboard, you will notice there are repeated `TCP 22` and `TCP 2222` connections.



NetFlow Discovery

71099 documents

Time	exporter	aprotocol	source_ip	source_port	destination_ip	destination_port	total_bytes	total_packets
2021-04-27 19:59:55.606Z	10.138.0.10	tcp	10.138.0.10	48706	95.78.102.16	2222	0	3
2021-04-27 19:59:53.627Z	10.138.0.10	tcp	10.138.0.10	40836	104.211.21.67	22	1,283	12
2021-04-27 19:59:53.558Z	10.138.0.10	tcp	10.138.0.10	42490	95.78.102.16	22	0	1
2021-04-27 19:59:53.302Z	10.138.0.10	tcp	10.138.0.10	53992	223.25.244.180	2222	0	1
2021-04-27 19:59:53.261Z	10.138.0.10	tcp	10.138.0.10	37932	58.200.62.1	2222	0	2
2021-04-27 19:59:53.046Z	10.138.0.10	tcp	10.138.0.10	60866	222.227.59.253	2222	0	1
2021-04-27 19:59:52.534Z	10.138.0.10	tcp	10.138.0.10	51228	37.136.186.196	22	0	1
2021-04-27 19:59:52.534Z	10.138.0.10	tcp	10.138.0.10	41570	77.150.54.73	2222	0	1
2021-04-27 19:59:52.410Z	10.138.0.10	tcp	10.138.0.10	44442	175.105.210.80	2222	0	2
2021-04-27 19:59:52.002Z	10.138.0.10	tcp	10.138.0.10	55106	171.110.220.17	22	0	1

The `TCP 2222` traffic doesn't appear in the ring graphs we've looked at so far because there is a very small amount of bytes actually transmitted in the `TCP 2222` requests; therefore, there is no traffic volume in them. However, they would have produced `SYN TCP` packets on the network, which is why they are captured in the Flow Logs.

Bonus Questions: What Have You Just Found?

1. Based on what you've observed so far, can you identify what may have occurred with the `comms-control` VM?

Based on the traffic you've seen, what are these connections indicative of?

Hint

Based on what we have observed in the traffic so far:

- Lots of concurrent connections to `TCP 22`
- Lots of concurrent connections to `TCP 2222`
- A lot of connections that didn't fully establish, based on them containing 0 bytes
- Connections occurring rapidly
- One connection to an external IP address that contained more traffic than any other connection.

Answer

It's likely you are looking at malware traffic that is scanning the internet and has established a C2 connection back to IP `165.227.88.48`.

Can you determine what malware this may be simply from the network traffic?

This is actually a pretty hard question to answer with *only* Flow logs. The best we can do is speculate and collect intel for further analysis of the VM.

Answer

If you Google for `malware`, `linux`, `22`, and `2222`, you may come across this story (https://for509.com/uwld7). It seems to match up with the evidence we're seeing in the Flow Logs.

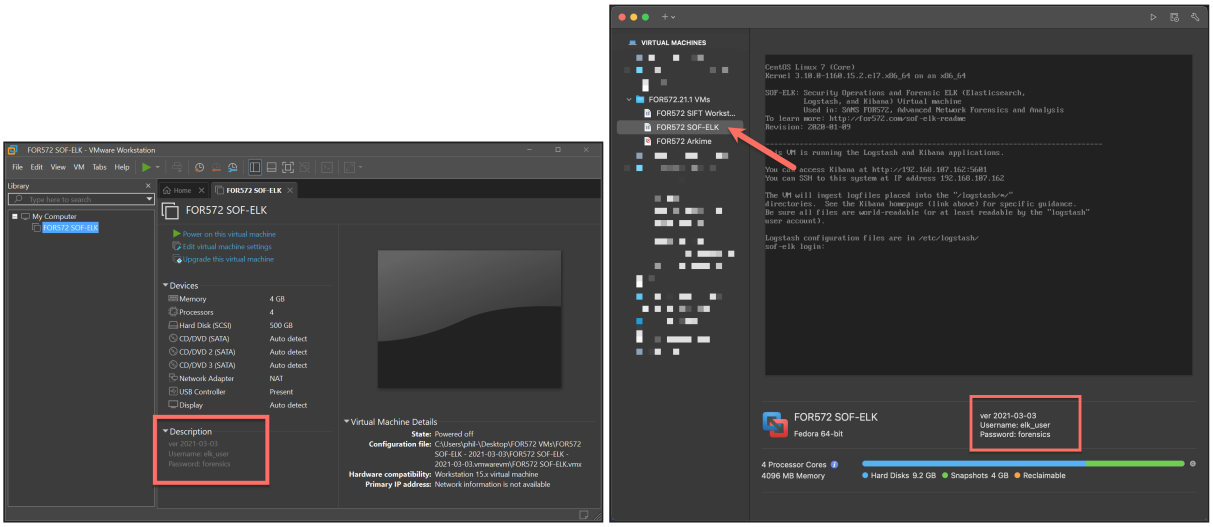
Key Takeaways

- You should now have a few strong skills to help you identify Google Cloud Flow logs when loaded into SOF-ELK.
- You should now know how to navigate SOF-ELK's NetFlow dashboard to look for unusual traffic patterns originating from Google Cloud VMs.
- You should have an appreciation for potential traffic that doesn't appear in the pre-made dashboard widgets and know how to check for this data.
- You should be able to identify/speculate on the malware that was likely infected on the VM we identified as producing unusual traffic.

Virtual Machine Credentials

The login credentials for all virtual machines used in this class are listed below for quick reference.

All login credentials are also displayed in the respective virtual machine's information panel. Below are screenshots showing the login credentials under VMware Workstation and VMware Fusion, respectively.



1. SOF-ELK®

    - Username: `elk_user`
    - Password: `forensics`

    This user has `sudo` access for all commands on the virtual machine.

Syntax Used in This Course

The FOR509 course documentation uses consistent syntax styles with which you should become familiar. This section helps you to make sense of what the material conveys, so you can focus more on course material than styling.

Syntax Descriptions and Examples

Note

The commands listed in this section of the lab are just for reference, so you can become familiar with text styles used in the course materials. *No need to actually run them in your SIFT Workstation VMware Image*!

1. Text blocks that appear in the format shown below contain commands that you would run in the SOF-ELK VM. These code blocks include an icon to the far right that allows you to copy the contents of the block, suitable for pasting into the shell in your class VMs.

   List the contents of the `/tmp/` directory

   ```
   cd /tmp/
   ls -l
   ```

   The results are shown in the box below.

   Results

   ```
   total 2836
   -rw------- 1 sansforensics sansforensics       0 Apr  3 17:39 config-err-S09tBf
   -rw------- 1 sansforensics sansforensics       0 Jul 21 18:39 config-err-zVMGjJ
   -rw-r--r-- 1 root          root                0 May 10 07:45 fileK8YYJh
   -rw-r--r-- 1 root          root                0 Jun 11 07:45 fileVAP3BY
   -rw-r--r-- 1 root          root                0 Jul 11 07:45 fileVeFMlj
   drwxrwxr-x 3 sansforensics sansforensics    4096 Jul  6 18:03 npm-57783-5d61223f
   drwxrwxr-x 3 sansforensics sansforensics    4096 Jul  6 18:04 npm-57819-3bc1b3dc
   ...
   ```

2. Direct questions are reflected in the material as shown below.

Who does the `admin@pymtechlabs.com` UPN belong to?

Hank Pym

Select a field to add to your existing table

The field is `user_name.keyword`

Expected results

Top values of user_principal_name.keyword	Top values of user_name.keyword	Count of records
admin@pymtechlabs.com	Hank Pym	1,036
jvandyne@pymtechlabs.com	Janet Van Dyne	627
slang@pymtechlabs.com	Scott Lang	279
luis@pymtechlabs.com	Luis	82
675be0f4-2486-4443-bef6-d37d9043ae99	675be0f4-2486-4443-bef6-d37d9043ae99	21
dcross@pymtechlabs.com	Darren Cross	12
f2830485-572c-4311-8ae1-08d355887e97	f2830485-572c-4311-8ae1-08d355887e97	1

3. When referring to literal strings inline with narrative text, the strings will be in depicted in Courier New font. For example, a search string of `not result_type: (0 OR 50074 OR 50140)` might be noted in the material inline, or via a call-out box as shown below:

```
not result_type: (0 OR 50074 OR 50140)
```

4. Some commands follow a "template" format, in which you will replace a part of the template with relevant information. These template command lines will include placeholders surrounded by the `<%` and `%>` enclosures with uppercase letters between them. This is an indication that you must alter the template command accordingly. For example, in the following command, you'd replace the `<%IP_ADDRESS%>` with the IP address of your SOF-ELK VM as an example.

```
ssh <%IP_ADDRESS%>
```

5. It is generally unadvisable to use the `root` administrative account for normal activities. We will follow best practices and use the `sudo` utility to perform administrative actions within the SOF-ELK VM environment wherever needed. The `elk_user` user has full `sudo` access to provide a reasonable balance between best practices and a practical classroom-based lab environment.
6. In the electronic workbook, some images are clickable, resulting in an enlarged version. This can be helpful when examining a detailed diagram or screenshot. An example of this is below.

Time	_source
2021-04-07 23:19:33.000 +00:00	parameters.DomainController: parameters.IgnoreDehydratedFlag: true parameters.AdminAuditLogEnabled: true parameters.Identity: NAMPR06A005.PROD.OUTLOOK.COM/Microsoft Exchange Hosted Organizations/pymtechlabs.onmicrosoft.com @timestamp: 2021-04-07 23:19:33.000 +00:00 result_status: true user_ids: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost)
2021-04-07 23:19:33.000 +00:00	parameters.DomainController: parameters.Identity: NAMPR06A005.PROD.OUTLOOK.COM/Microsoft Exchange Hosted Organizations/pymtechlabs.onmicrosoft.com @timestamp: 2021-04-07 23:19:33.000 +00:00 result_status: true user_ids: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost) organization_guid: 7e325eda-7945-46d3-ac99-f0dcfeb4628e @version: 1 type: office365 app_id: user_name: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost) tags: process_archive,
2021-04-07 23:19:32.000 +00:00	parameters.PrivacyStatementURL: http://go.microsoft.com/fwlink/?LinkID=259417 parameters.Identity: pymtechlabs.onmicrosoft.com parameters.PrivacyLinkDisplayEnabled: true @timestamp: 2021-04-07 23:19:32.000 +00:00 result_status: true user_ids: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost) organization_guid: 7e325eda-7945-46d3-ac99-f0dcfeb4628e @version: 1 type: office365 app_id: user_name: NT AUTHORITY\SYSTEM
2021-04-07 23:19:31.000 +00:00	parameters.DomainController: parameters.HygieneSuite: Premium parameters.Identity: pymtechlabs.onmicrosoft.com @timestamp: 2021-04-07 23:19:31.000 +00:00 result_status: true user_ids: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost) organization_guid: 7e325eda-7945-46d3-ac99-f0dcfeb4628e @version: 1 type: office365 app_id: user_name: NT AUTHORITY\SYSTEM (Microsoft.Exchange.ServiceHost) tags: process_archive,

Helpful Hints for Labs

To get the most out of each lab, we will step you through the different portions of the workbook. The workbook is specifically designed to enable students from a variety of backgrounds and with different skill levels to get the most out of each lab.

Important Note

The labs in this workbook require the specific version of the SOF-ELK VM made for FOR509. The public SOF-ELK distributions will not work for these labs!

Warning

- SOF-ELK is pre-configured to use the UTC time zone. *Do NOT change this setting*. In general, UTC is strongly preferred for forensic analysis. Using this single time zone ensures a consistent point of reference for evidence that may have originated in many different physical locales. All labs in this class have been engineered with UTC as a point of reference.
- Feel free to *increase* the CPUs and/or memory allocated to SOF-ELK. Do NOT decrease the allocated resources, or the labs may be extremley slow.
- *Do NOT upgrade packages in the image*
- *Do NOT upgrade VMware Tools*. This is known to cause problems, notably a complete lack of VMware's "Shared Folders" functionality.

Live Lab Updates

This electronic workbook can be refreshed with updates when available. To do so, simply run the following command from a shell prompt:

Command lines

```
workbook-update
```

Expected results (when updates are available)

```
[elk_user@sof-elk ~]$ workbook-update
Beginning update process...
- Updating workbook files

Complete!
```

Expected results (when no updates are available)

```
[elk_user@sof-elk ~]$ workbook-update
Beginning update process...
- No workbook updates available

Complete!
[elk_user@sof-elk ~]$
```

When updates are available, they will be available in the browser on your local system. Note that a page refresh may be required if the updates are to content that is currently displayed.

Exercise Objectives

This section is designed to help students understand the larger picture of what the objectives of the lab are meant to show or teach. In some cases, we might be demonstrating an analytical technique or where to find specific information in a log. We strongly recommend that students quickly look over these objectives when beginning the lab.

Exercise Preparation

This section outlines the specific system, the condition of that system, or the capabilities that must be enabled before moving into the actual lab. Skipping over this step could mean that your system might not be ready for analysis. As many of our labs require the use of Kibana, make sure that you have the correct timeframe and index selected.

Questions without Explanations and Questions with Step-by-Step Instructions

For most labs, we try to get you to focus on the core concepts and analytical techniques instead of just blindly duplicating our steps. The most important part of this course, especially if you are new, is to focus on the analytic techniques.

Note

In the printed workbook, the step-by-step instructions and explanations are provided right after the questions. In the electronic workbook, the step-by-step instructions and explanations are provided immediately following each question using a drop-down box such as this (click the box to see the solution):

> **Solution**
>
> Here's where an answer would go. There will be a drop-down box such as this following each individual question.

Takeaways

For every lab, the takeaway section highlights important information we found in the logs.

SANS Courseware License Agreement

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With this CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, USER AGREES TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, USER AGREES THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If User does not agree, User may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP® and PMBOK® are registered trademarks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

All reference links are operational in the browser-based delivery of the electronic workbook.