

BFOR 650 – LAB REPORT 8 – Creating a Breach Detection Dashboard in Kibana

NAME: Phanindhar Reddy Karnati
ID: 001667635

8.1: NIDS Alerts

- I logged into Security Onion through the Ubuntu VM and opened the Kibana interface using the left-hand panel.

The screenshot shows the 'Overview' page of the Security Onion interface. On the left is a vertical sidebar with icons for Overview, Alerts, Dashboards, Hunt, Cases, PCAP, Grid, Downloads, Administration, Tools, Kibana, Grafana, CyberChef, Playbook, FleetDM, and Navigator. The main content area has a dark background with white text. It features a 'Getting Started' section with instructions on how to use the Alerts interface to hunt for threats. Below that is a 'What's New' section. A 'Customize This Space' section contains a code snippet for customizing the 'motd.md' file using SSH:

```
sudo cp /opt/so/saltstack/default/salt/soc/files/soc/motd.md /opt/so/saltstack/local/salt/soc/files/soc/
Then edit the new file as desired using your favorite text editor.

Finally, restart SOC to make the changes take effect:
sudo so-soc-restart
```

- I navigated to the Dashboards section and clicked on Create Dashboard, which opened a blank dashboard editor.

The screenshot shows the 'Editing New Dashboard' interface in Kibana. The top navigation bar includes tabs for pfSense - Login, Security Onion, and Dashboards - Elastic. The main area has a dark background with a central placeholder image of a dashboard. Below the image, text reads: 'This dashboard is empty. Let's fill it up!' with a sub-instruction: 'Create a visualization of your data, or add one from the library.' At the bottom are two buttons: 'Create visualization' and 'Add from library'. The bottom left corner of the screen shows the Unity desktop environment of the host machine.

3. I chose to create a new **Data Table** visualization, searched for "Alerts," and selected the **Security Onion – Alerts** as the data source.

The screenshot shows the Kibana Visualize Library interface. A modal window titled "New Data table / Choose a source" is open. In the search bar, the text "alerts" is entered. Below the search bar, there are two dropdown menus: "Type" and "Title". Under "Type", there are three items: "OSSEC - Alerts", "NIDS - Alerts", and "Security Onion - Alerts". The "Security Onion - Alerts" item is highlighted with a blue selection bar. On the right side of the modal, there is a list of recent visualizations, all of which have been created 3 hours ago. The visualizations listed are: "Line" (3 hours ago), "Markdown" (3 hours ago), "Data table" (3 hours ago), and another "Data table" (3 hours ago). At the bottom right of the modal, there is a "Create visualization" button.

4. In the data panel, I added a bucket by choosing Split Rows, selected Terms for aggregation, and used rule.name.keyword as the field. I set the size to 50 and clicked Update.

The screenshot shows the Kibana Data Table visualization editor. On the left, there is a table with the following data:

	Count
Windows Logon Success	57
PAM: Login session opened.	20
System Audit event.	14
PAM: Login session closed.	13
Service startup type was changed	12
Successful sudo to ROOT executed.	11
Software Protection service scheduled successfully	6
Listened ports status (netstat) changed (new port opened or closed).	5
Windows System error event	5
Ossec agent started.	3

On the right, the visualization configuration pane is visible. It includes sections for "Metrics" (Metric Count), "Buckets" (Split rows, Aggregation, Terms, Field: rule.name.keyword, Order by Metric: Count, Order: Descending, Size: 50), and "Advanced" settings (Group other values in separate bucket, Show missing values, Custom label).

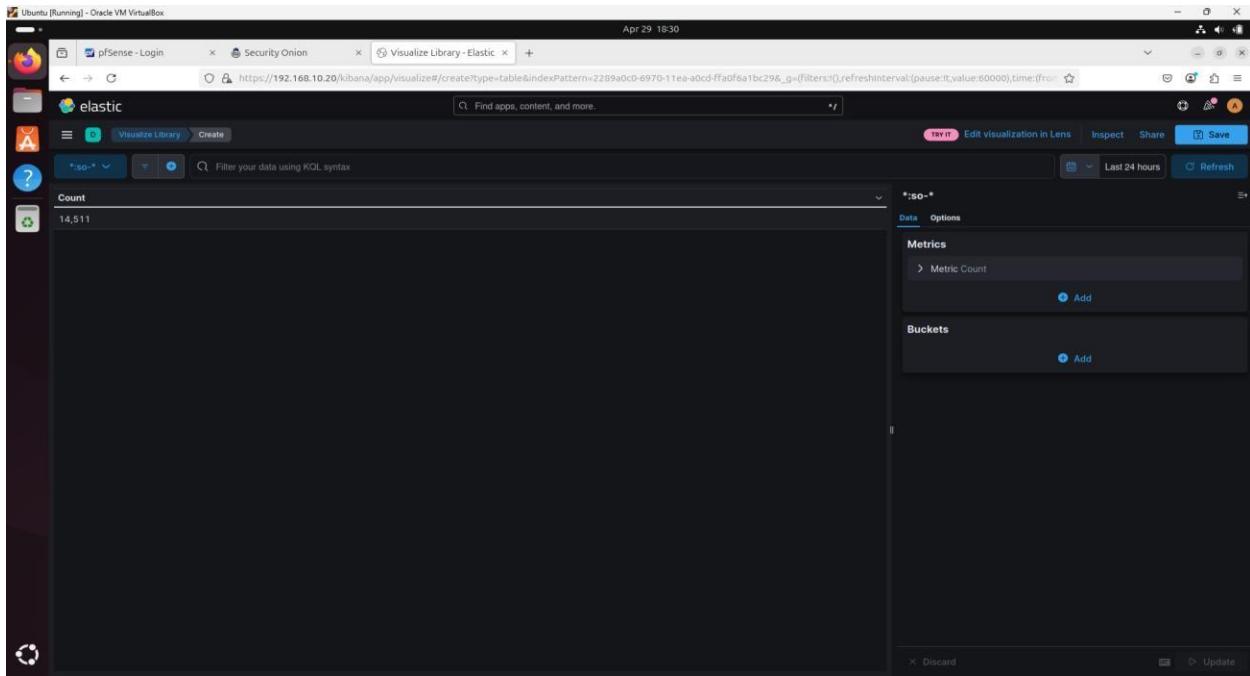
5. To filter by severity, I clicked Add Filter, set the field to event.severity, selected is between, and entered values 3 and 8.
6. After this, the data table provided a summary of frequent NIDS alerts. I saved it as Breach Detection – NIDS Alerts Summary and returned to the dashboard.
7. I saved the full dashboard as Breach Detection Portal using the Save button at the top.

The top screenshot shows the Kibana 'Visualize Library - Elastic' interface. A search bar at the top contains the URL `https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=5c3effd0-72ae-11ea-8dd2-9d8795a1200ba_g=[filters:[],refreshInterval:(pause:lt,value:60000),time:]`. Below the search bar, there's a 'rule' section with a dropdown for 'PAM' and a 'Preview' section showing 'event.severity: 3 to 8'. On the right side, there's a 'Metrics' panel titled 'Security Onion - Alerts' which lists 'Metric Count' and 'rule.name.keyword' as the field. The bottom part of the screen shows a table of frequent NIDS alerts.

The bottom screenshot shows the 'Save dashboard' dialog box. It has fields for 'Title' (set to 'Breach Detection Portal'), 'Description' (empty), and 'Tags' (empty). There's also a checkbox labeled 'Store time with dashboard' with the note: 'This changes the time filter to the currently selected time each time this dashboard is loaded.' At the bottom are 'Cancel' and 'Save' buttons.

8.2: Zeek Notices

- I clicked **Create Visualization** from inside the Breach Detection Portal and chose a **Data Table** type. I used the ***.so-*** data source.



2. In the search bar, I typed event.dataset:notice AND event.module:zeek to focus on Zeek notice logs and clicked **Update**.

The screenshot shows the Kibana Visualize Library interface. The search bar at the top contains the query "event.dataset:notice AND event.module:zeek". The main area displays a single metric named "Count" with a value of 39. On the right side, there are sections for "Metrics" and "Buckets". The "Metrics" section has a "Metric Count" option with an "Add" button. The "Buckets" section is currently empty, with an "Add" button.

3. I added a **Split Rows** bucket, chose **Terms** for aggregation, and set the field to notice.message.keyword, with size 50.

The screenshot shows the same Kibana Visualize Library interface as the previous one, but with a more complex configuration. A "Split rows" bucket has been added under the "Buckets" section. The "Aggregation" dropdown is set to "Terms", and the "Field" dropdown is set to "notice.message.keyword". The "Order by" dropdown is set to "Metric: Count" with "Descending" selected. The "Size" input field is set to "50". There are also checkboxes for "Group other values in separate bucket" and "Show missing values", both of which are checked. The "Custom label" and "Advanced" sections are visible below the main configuration area.

- I then added another Split Rows bucket for source.ip to pair source IPs with their notice messages.

The screenshot shows the Kibana Visualize Library interface. A search query 'event.dataset:notice' AND 'event.module:zeek' is entered. The results panel displays 'No results found'. The configuration panel on the right is set up for a 'Split rows' aggregation with 'source.ip' as the field, ordered by 'Metric: Count' in descending order, with a size of 50. Other settings include 'Group other values in separate bucket' and 'Show missing values'.

- Since there were no Zeek notice logs visible, I saved the widget anyway as **Breach Detection – Zeek Notices Summary**.

The screenshot shows the 'Save visualization' dialog box. The title is 'Breach Detection - Zeek Notices Summary'. The 'Add to dashboard' section has 'Existing' selected with 'Breach Detection Portal' chosen. The 'Save and go to Dashboard' button is highlighted.

8.3: Zeek Intel Logs

- I created another Data Table visualization and selected the *.so-* data source.

The screenshot shows the Kibana Visualize Library interface. The top navigation bar includes tabs for 'Visualize Library' and 'Elastic'. The main search bar contains the query 'Count'. The results table shows a single row with the value '14,511'. On the right side, there are sections for 'Metrics' and 'Buckets', each with an 'Add' button. The bottom right corner features 'Discard' and 'Update' buttons.

- I entered event.dataset:intel in the filter bar to isolate intel-related logs and clicked Update.

The screenshot shows the same Kibana Visualize Library interface as the previous one, but with a different filter applied. The search bar now contains the query 'event.dataset: intel'. The results table shows a single row with the value '0'. The right-hand sidebar remains the same, and the bottom right corner still has 'Discard' and 'Update' buttons.

3. I added a Split Rows bucket, set aggregation to Terms, and chose intel.sources.keyword as the field with size 50.
4. I added another Split Rows for source.ip to display the IPs alongside intel alerts.
5. There were no active Zeek Intel logs, so I saved the visualization as Breach Detection – Intel Logs Summary.

Breach Detection - Intel Logs Summary

Event Type	Count
PAM: Login session closed	38
Successful sudo to ROOT executed	29
Service startup type was changed	20
System Audit event	14
Software Protection service scheduled successfully	8
Listened ports status (netstat) changed (new port opened or cl...	7

8.4: Suspicious Process and File Creation

1. I created a new Data Table using the Security Onion Sysmon data source.

Security Onion - Sysmon

Count: 3,217

2. I added a Split Rows bucket using process.executable.keyword, set aggregation to Terms, and limited the size to 50.

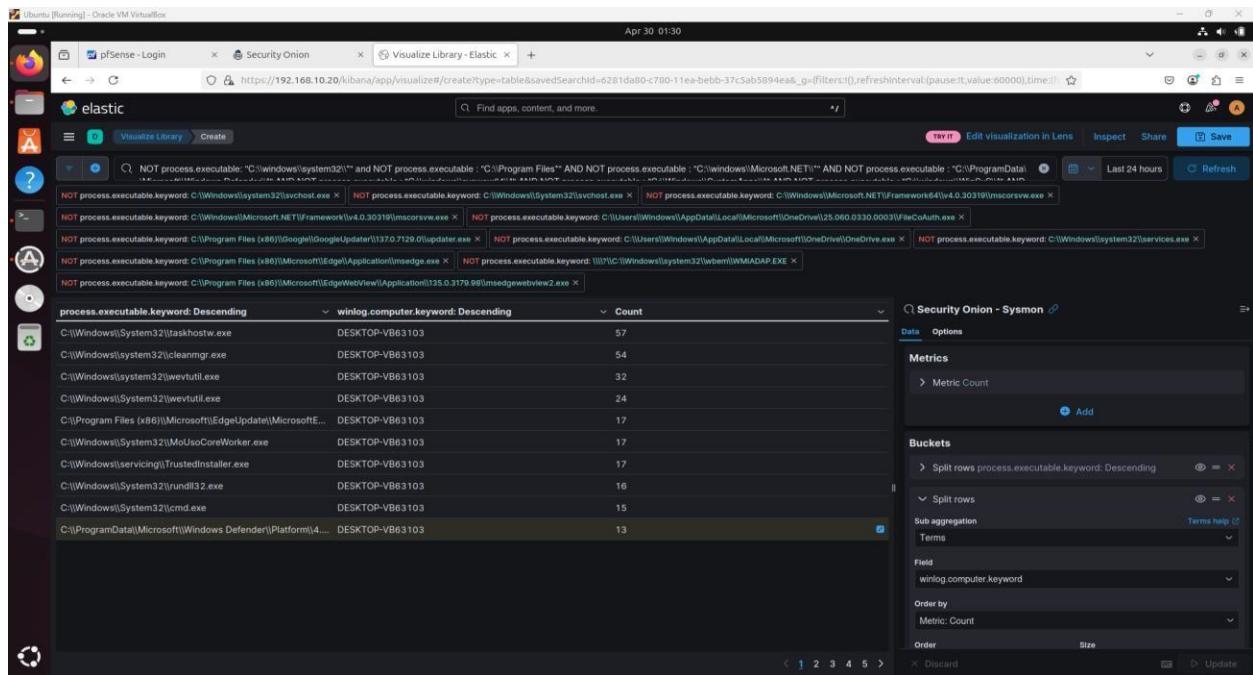
3. Then I added another split row using winlog.computer.keyword to display the computer associated with each process.

Field	Value	Count
process.executable.keyword	Descending	1,486
winlog.computer.keyword	Descending	1,486
C:\Windows\system32\svchost.exe	DESKTOP-VB63103	1,486
C:\Windows\System32\svchost.exe	DESKTOP-VB63103	214
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\in...	DESKTOP-VB63103	188
C:\Windows\Microsoft.NET\Framework\v4.0.30319\msco...	DESKTOP-VB63103	161
C:\Users\Windows\AppData\Local\Microsoft\OneDrive\2...	DESKTOP-VB63103	114
C:\Users\Windows\AppData\Local\Microsoft\OneDrive\O...	DESKTOP-VB63103	92
C:\Windows\system32\services.exe	DESKTOP-VB63103	88
C:\Program Files (x86)\Google\GoogleUpdate\137.0.712...	DESKTOP-VB63103	87
C:\Program Files (x86)\Microsoft\Edge\Application\msed...	DESKTOP-VB63103	69
\\?\C:\Windows\system32\wbem\WMIDAP.EXE	DESKTOP-VB63103	68

- In the filter bar, I entered a complex NOT filter to exclude known safe executable paths from directories like System32, Program Files, and Windows Defender.

```
NOT process.executable: "c:\\windows\\system32\\\\*" AND NOT
process.executable: "c:\\Program Files\\\\*" AND NOT
process.executable: "c:\\windows\\Microsoft.NET\\\\*" AND NOT
process.executable: "c:\\ProgramData\\\\Microsoft\\\\Windows Defender\\\\*" AND NOT
process.executable: "c:\\windows\\syswow64\\\\*" AND NOT
process.executable: "c:\\Windows\\\\SystemApps\\\\*" AND NOT
process.executable: "c:\\windows\\WinSxS\\\\*" AND NOT
process.executable: "c:\\windows\\servicing\\\\*" AND NOT
process.executable: "c:\\windows\\softwaredistribution\\\\*"
```

Click on **Update** and manually eliminate all other safe executables and do this process over time to keep the safe executables filtered out.



- I saved this visualization as Breach Detection – Suspicious Image Paths.

8.5: Suspicious PowerShell Commands

- I created a **Data Table** in the Breach Detection Portal using the *:so-* data source.

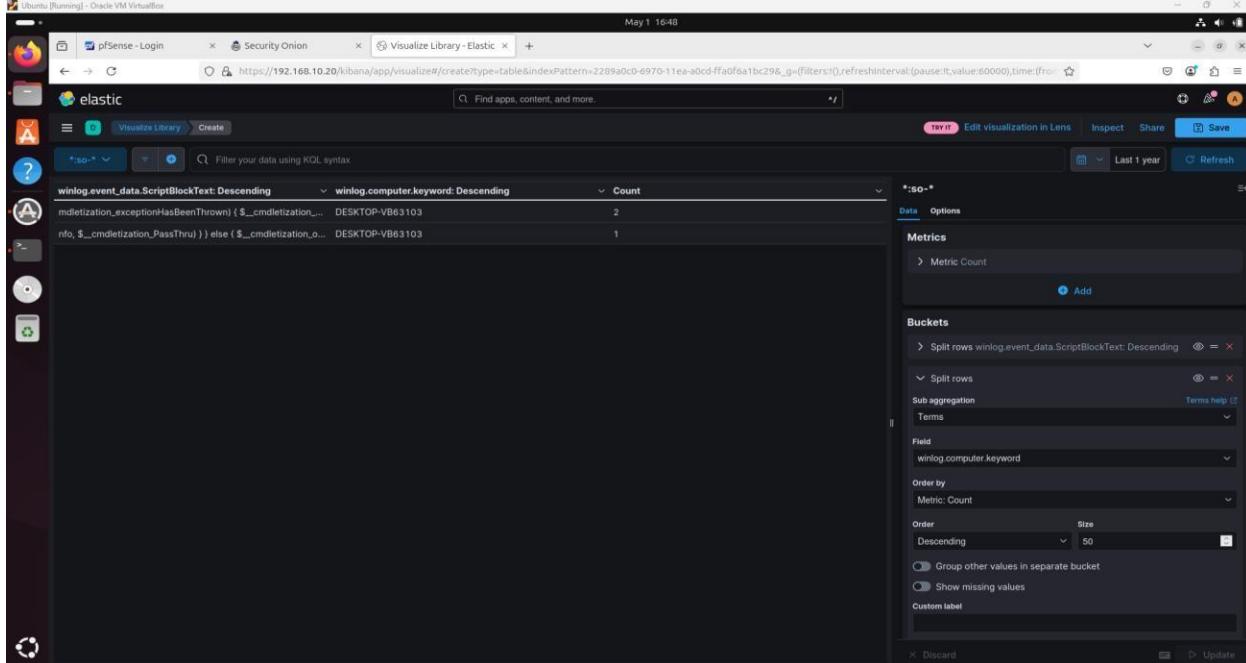
The screenshot shows the Kibana Visualize Library interface. A visualization titled "Visualize Library - Elastic" is displayed. The visualization type is set to "Table". The index pattern is "2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g-(filters:[],refreshInterval:[pause:lt,value:60000]),time:(from:now-1d, to:now)" and the time range is "Last 1 year". The visualization has a single metric "Count" which shows a value of 33,880. The right panel is titled "*:so-*" and contains sections for "Metrics" (Metric Count) and "Buckets". The "Metrics" section has an "Add" button. The "Buckets" section also has an "Add" button. There are "Data" and "Options" tabs at the top of the right panel. The bottom right corner of the visualization area has "Discard" and "Update" buttons.

- I added a Split Rows bucket with winlog.event_data.scriptBlockText.keyword and set the size to 50.

The screenshot shows the Kibana Visualize Library interface with the same visualization setup as the previous screenshot. However, the right panel now displays a configuration for a "Split rows" bucket. The "Field" is set to "winlog.event_data.ScriptBlockText" and the "Order by" field is "Metric: Count". The "Order" dropdown is set to "Descending" and the "Size" is set to "50". There are checkboxes for "Group other values in separate bucket" and "Show missing values". The "Custom label" and "Advanced" sections are also visible. The bottom right corner of the visualization area has "Discard" and "Update" buttons.

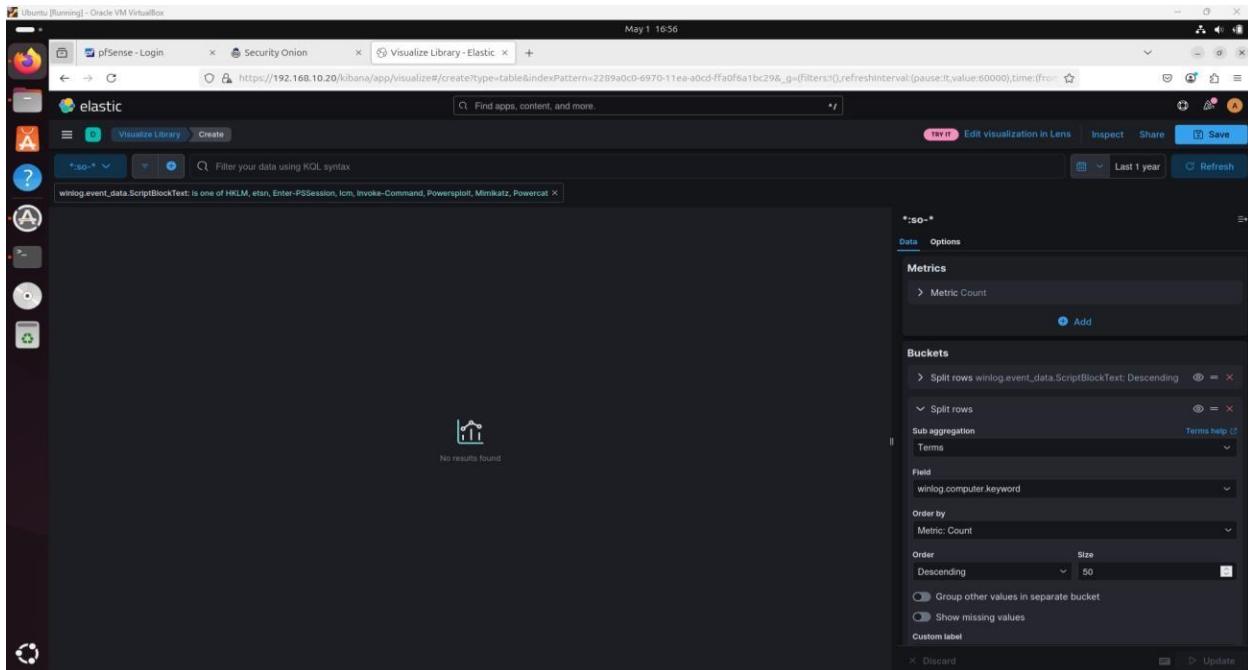
3. I added another Split Rows using winlog.computer.keyword to group the script logs by host.

4. I filtered for suspicious script content using winlog.event_data.scriptBlockText and

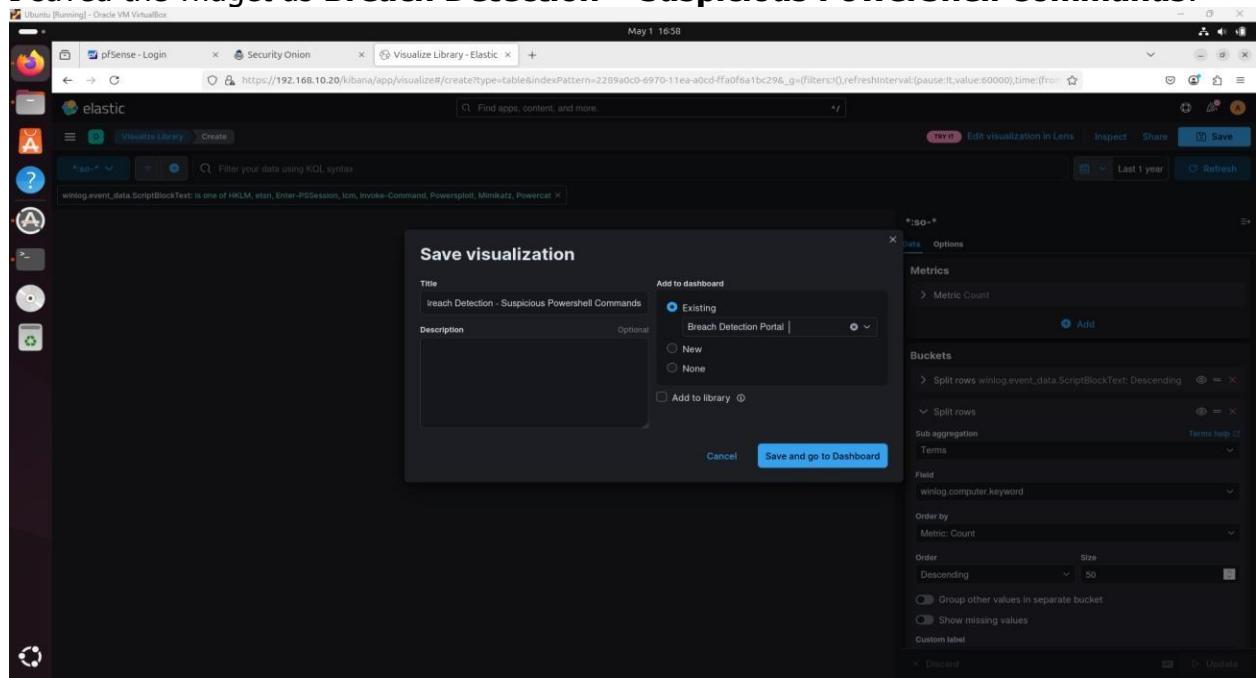


the operator is one of, then input known suspicious PowerShell command patterns.

- | | |
|-----------------|--|
| HKLM | - for detecting interaction with the registry |
| etsn | - to detect remote access attempts |
| Enter-PSSession | - to detect remote access attempts |
| Icm | - to detect remote command execution attempts |
| Invoke-Command | - to detect remote command execution attempts |
| Powersploit | - to detect the use of the powersploit framework |
| Mimikatz | - to detect the use of the hacking tool Mimikatz |
| Powercat | - to detect the use of the powercat framework |



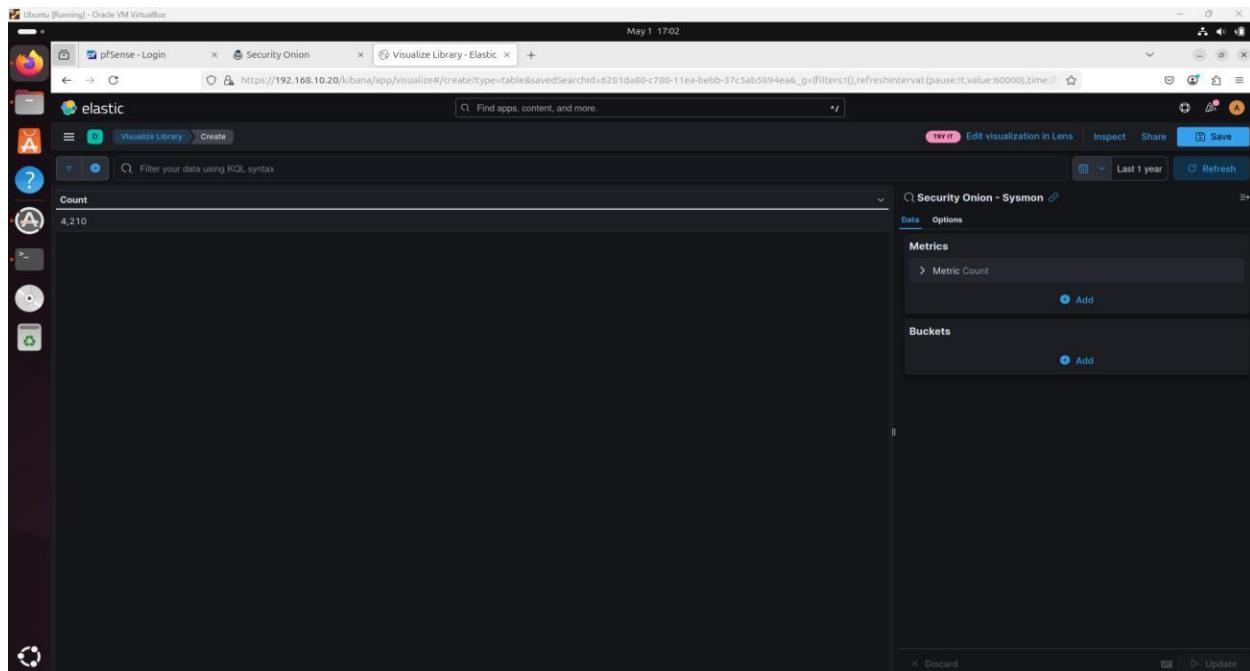
5. I saved the widget as **Breach Detection – Suspicious PowerShell Commands**.



6. The way PowerShell is started can give away any malicious intent or intended abuse as well. The following table summarizes some suspicious ways PowerShell can be invoked (started):

Parameter	Variations	Purpose of parameter
-noprofile	-nop	Skips loading of profile.ps1 and avoids logging
-encoded	-e -en -enc	Lets a user run encoded PowerShell scripts
-ExecutionPolicy bypass	-ep bypass, -exp bypass, -exec bypass	Bypasses any execution policy
-windowStyle hidden		Prevents the creation of a window; may generate false positives
-version 2	-v 2, -version 2.0	Forces use of PowerShell version 2

- Detecting suspicious invocation of the PowerShell engine can show malicious intent. We will add a data table that lists any process creation logs that contain these suspicious command-line parameters. The logs this data is in are provided by the Sysmon logging engine that we deployed earlier. Using Wazuh to add Sysmon logging.
- Click on **Create Visualization** to add a new widget in the **Breach Detection Portal** dashboard and select **Data table** and search for and add **Security Onion Sysmon** data source.



9. Add a **Split Rows** data bucket and set **aggregation** to **Terms**, with **process.command_line.keyword** as the **Field**, set **Size** as **50** and click **Update**.

process.command_line.keyword: Descending	Count
'C:\Users\Windows\AppData\Local\Microsoft\OneDrive\25.060.0330.0003\FileCoAuth...'	61
C:\Windows\system32\svchost.exe -k netvsc -p -s gpvc	38
taskhostw.exe	37
C:\Windows\System32\mousocoreworker.exe -Embedding	19
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.7129.0\)update.exe" --system ...'	18
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.7129.0\)update.exe" --system ...'	17
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.7129.0\)update.exe" --wake --sy...'	17
'C:\Program Files (x86)\(Microsoft\Edge Update\MicrosoftEdgeUpdate.exe" /ua /installso...'	13
'C:\Windows\System32\LocationNotificationWindows.exe"	13
C:\Windows\servicing\TrustedInstaller.exe	11

10. We will be adding the source computer name to show this with the PowerShell invocation event entries. To do this, add an additional **Split Rows** data bucket and set **Aggregation** to **Terms**, with **winlog.computer.keyword** as the **Field** selection. Set the **Size** option to **50** and click **Update**.

process.command_line.keyword: Descending	winlog.computer.keyword: Descending	Count
'C:\Users\Windows\AppData\Local\Microsoft\OneDrive\...' DESKTOP-VB63103	61	
C:\Windows\system32\svchost.exe -k netvsc -p -s gpvc DESKTOP-VB63103	38	
taskhostw.exe DESKTOP-VB63103	37	
C:\Windows\System32\mousocoreworker.exe -Embedding DESKTOP-VB63103	19	
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.71... DESKTOP-VB63103	18	
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.71... DESKTOP-VB63103	17	
'C:\Program Files (x86)\(Google\GoogleUpdater\137.0.71... DESKTOP-VB63103	17	
'C:\Program Files (x86)\(Microsoft\Edge Update\Microsoft... DESKTOP-VB63103	13	
'C:\Windows\System32\LocationNotificationWindows.exe" DESKTOP-VB63103	13	
C:\Windows\servicing\TrustedInstaller.exe DESKTOP-VB63103	11	

11. We will be adding the username that started the PowerShell process to show with the PowerShell invocation event entries. To do this, add an additional **Split Rows** data bucket and set **Aggregation** to **Terms**, with the **user.name.keyword** as the **Field** selection. Set the **Size** option to **50** and click **Update**.

The screenshot shows the Kibana interface with a table visualization titled "Visualize Library - Elastic". The table lists log entries with columns: process.command_line.keyword, winlog.computer.keyword, user.name.keyword, and Count. The data shows various processes like taskhostw.exe, taskhost.exe, and Google processes, all originating from DESKTOP-VB63103 and running as NT AUTHORITY\SYSTEM or NT AUTHORITY\LOCAL SERVICE. The configuration panel on the right is titled "Security Onion - Sysmon" and shows the setup for the "Split rows" data bucket. It has a "Sub aggregation" of "Terms" for the "Field" "user.name.keyword" ordered by "Count" in descending order with a "Size" of 50. There are also options to "Group other values in separate bucket" and "Show missing values". A "Custom label" section is present.

12. This generates a complete list of command-line logs. We are only going to filter for the interesting ones. Click on the **Add Filter** button and set the **Field** option to **process.command_line** and the **Operator** option to **is one of**. Now, one at a time, add in the following values and click **Save** when finished:

- noprofile
- nop
- encoded
- en
- enc
- enco
- encod
- encode
- ExecutionPolicy bypass**
- ep bypass**
- exp bypass**
- exec bypass**
- bypass**
- windowStyle hidden**

The next screenshot shows the configured widget:

This screenshot shows the Kibana interface on an Ubuntu VM. The top navigation bar includes tabs for 'pfSense - Login', 'Security Onion', and 'Visualize Library - Elastic'. The URL in the address bar is [https://192.168.10.20/kibana/app/visualizer/create?type=table&savedSearchId=6281da80-c700-11ea-bebb-37c3ab5894ea&_g={filters:\[\],refreshInterval:'pause_it,value:00000},time:{from:now-1y%2Fd,to:now}}](https://192.168.10.20/kibana/app/visualizer/create?type=table&savedSearchId=6281da80-c700-11ea-bebb-37c3ab5894ea&_g={filters:[],refreshInterval:'pause_it,value:00000},time:{from:now-1y%2Fd,to:now}}). The main area displays a 'Metrics' visualization titled 'Security Onion - Sysmon'. The visualization shows a single chart with the title 'Metric Count' and a single data point labeled 'No results found'. The configuration panel on the right shows the search query: 'process.command_line: Is one of -noprofile, -nop, -encoded, -en, -enc, -enco, -encode, -ExecutionPolicy bypass, -ep bypass, -exp bypass, -exec bypass, -bypass, -windowStyle hidden'. The 'Buckets' section is set to 'Split rows process.command_line.keyword: Descending'. The 'Sub aggregation' dropdown is set to 'Terms help'. The 'Field' dropdown is set to 'user.name.keyword'. The 'Order by' dropdown is set to 'Metric: Count'. The 'Order' dropdown is set to 'Descending' with a size of 50. There is also an option to 'Group other values in separate bucket'.

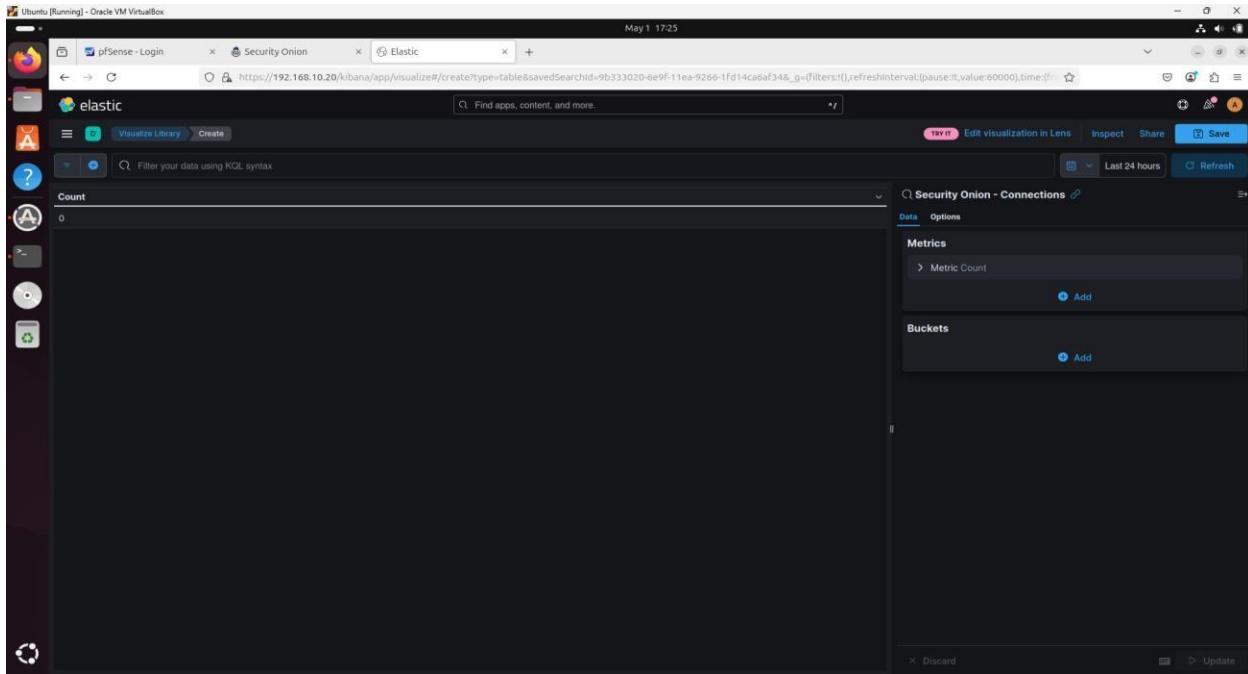
13. Click on **Save**, saving the widget as **Breach Detection – Suspicious PowerShell Invocation**.

This screenshot shows the Kibana interface on an Ubuntu VM. The top navigation bar includes tabs for 'pfSense - Login', 'Security Onion', and 'Elastic'. The URL in the address bar is [https://192.168.10.20/kibana/app/dashboard#/view/14691e70-2549-11fd-a4f0-23a23f0270a3?_g={filters:\[\],refreshInterval:'pause_it,value:60000},time:{from:now-1y%2Fd,to:now}}](https://192.168.10.20/kibana/app/dashboard#/view/14691e70-2549-11fd-a4f0-23a23f0270a3?_g={filters:[],refreshInterval:'pause_it,value:60000},time:{from:now-1y%2Fd,to:now}}). The main area displays a dashboard titled 'Breach Detection - Suspicious PowerShell Commands' and 'Breach Detection - Suspicious PowerShell Invocation'. The 'Panel settings' sidebar on the right shows the title 'Breach Detection - Suspicious PowerShell Invocation' and the description 'Suspicious PowerShell Commands'. The 'Apply' button is visible at the bottom right of the sidebar.

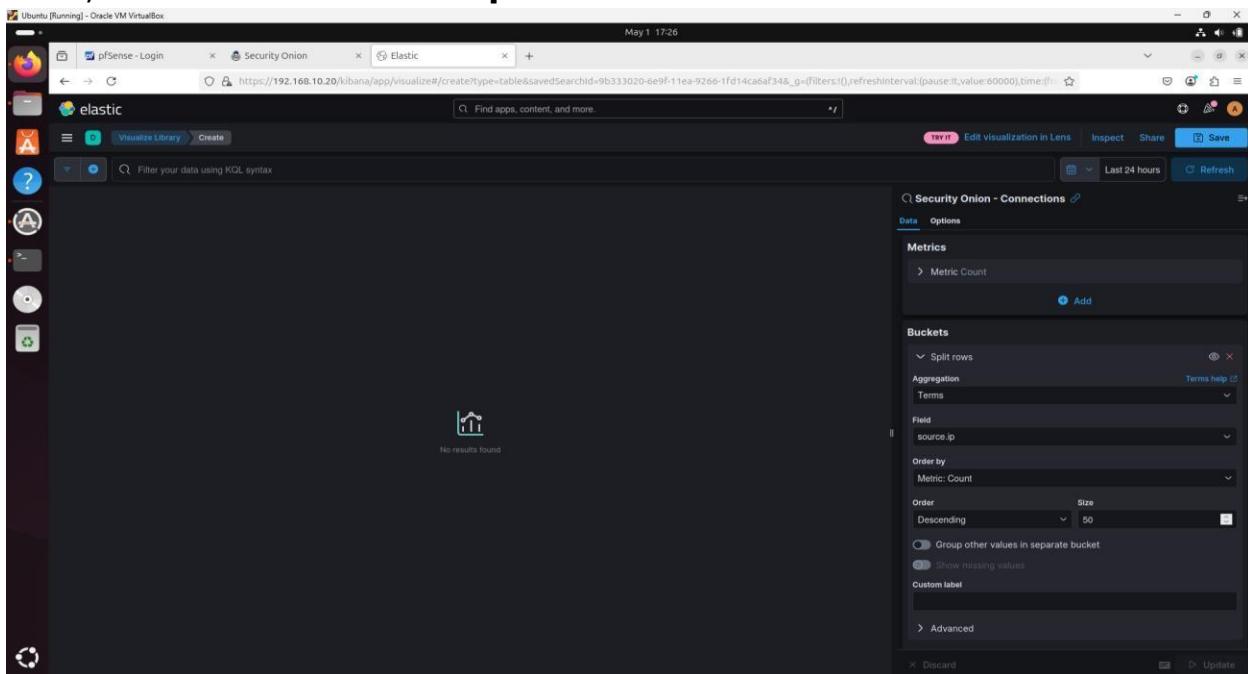
We now have a visualization around suspicious PowerShell commands and how PowerShell is invoked. Next, we are going to look at suspicious network connections.

8.7: Suspicious Egress Connections

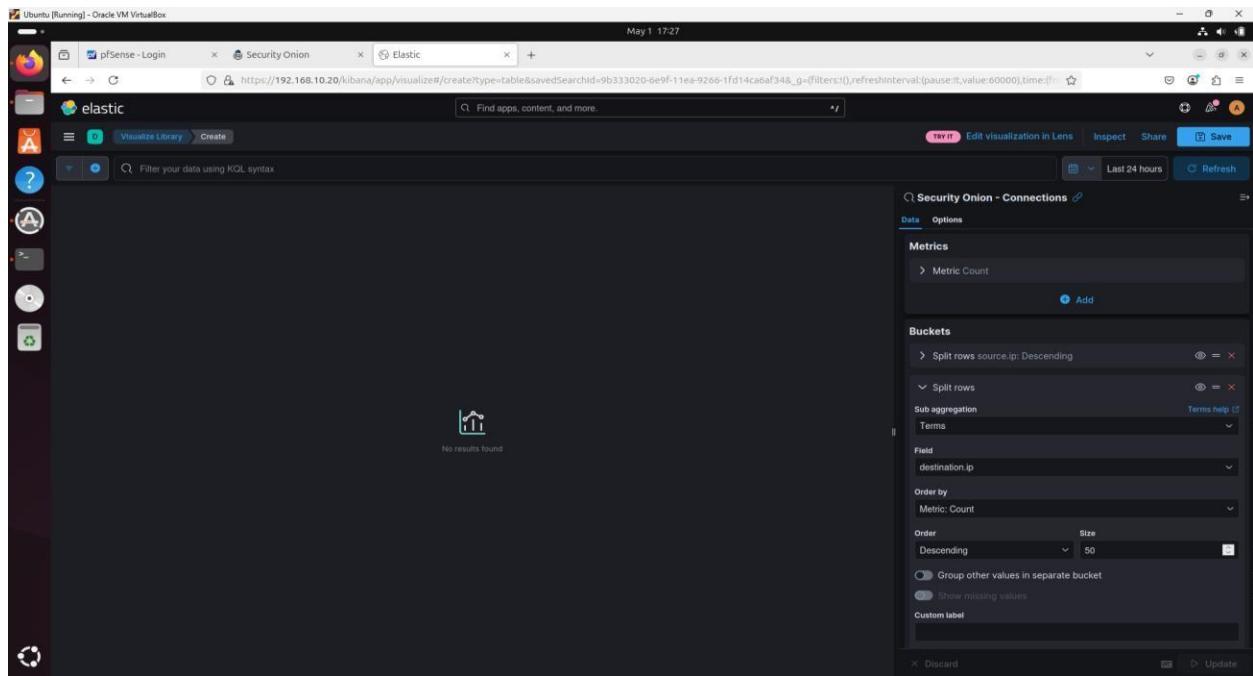
- I created a Data Table using the Security Onion - Connections data source.



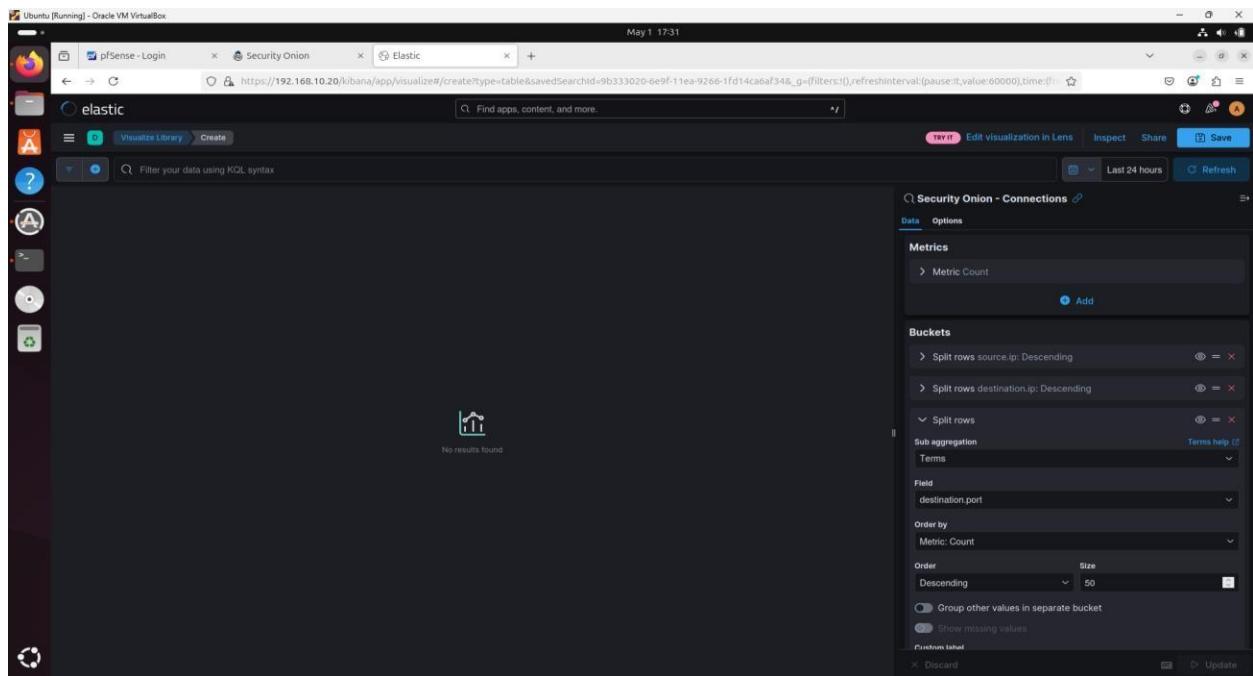
- I added **Split Rows** buckets for source.ip, destination.ip, destination.port, network.transport.keyword, network.bytes, and connection.state_description.keyword. a **Split Rows** data bucket and set **aggregation** to **Terms**, with **source.ip** as the **Field**, set **Size** as **50** and click **Update**.



3. I filtered the logs using `connection.local.originator: true` to show outgoing traffic from the internal network.



4. I also added `connection.local.responder: false` to exclude local destinations.



5. After sorting by count in descending order, I saved the visualization as **Breach Detection – Suspicious Egress Connections**.

The screenshot shows the Kibana interface on a dark-themed Ubuntu desktop. The browser tab is 'pfSense - Login'. The main view displays a search bar and a results section with a 'No results found' message. On the right, the visualization configuration panel is open for 'Security Onion - Connections'. It includes sections for 'Metrics' (Metric Count), 'Buckets' (Split rows source.ip: Descending, Split rows destination.ip: Descending, Split rows destination.port: Descending, Split rows network.transport.keyword: Descending), 'Sub aggregation' (Terms help), 'Field' (network.transport.keyword), 'Order by' (Metric: Count), 'Order' (Descending), 'Size' (10), and options for 'Group other values in separate bucket' and 'Show missing values'. Buttons for 'Discard', 'Update', and 'Save' are at the bottom.

6. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **network.bytes** as the **Field** selection. Set the **Size** option to **50**.

This screenshot shows the same Kibana interface after the changes were made. The 'Buckets' section now includes an additional entry: 'Split rows network.transport.keyword: Descending'. The 'Field' dropdown has been changed from 'network.transport.keyword' to 'network.bytes'. The 'Size' dropdown has been changed from '10' to '50'. The rest of the configuration remains the same as in the previous screenshot.

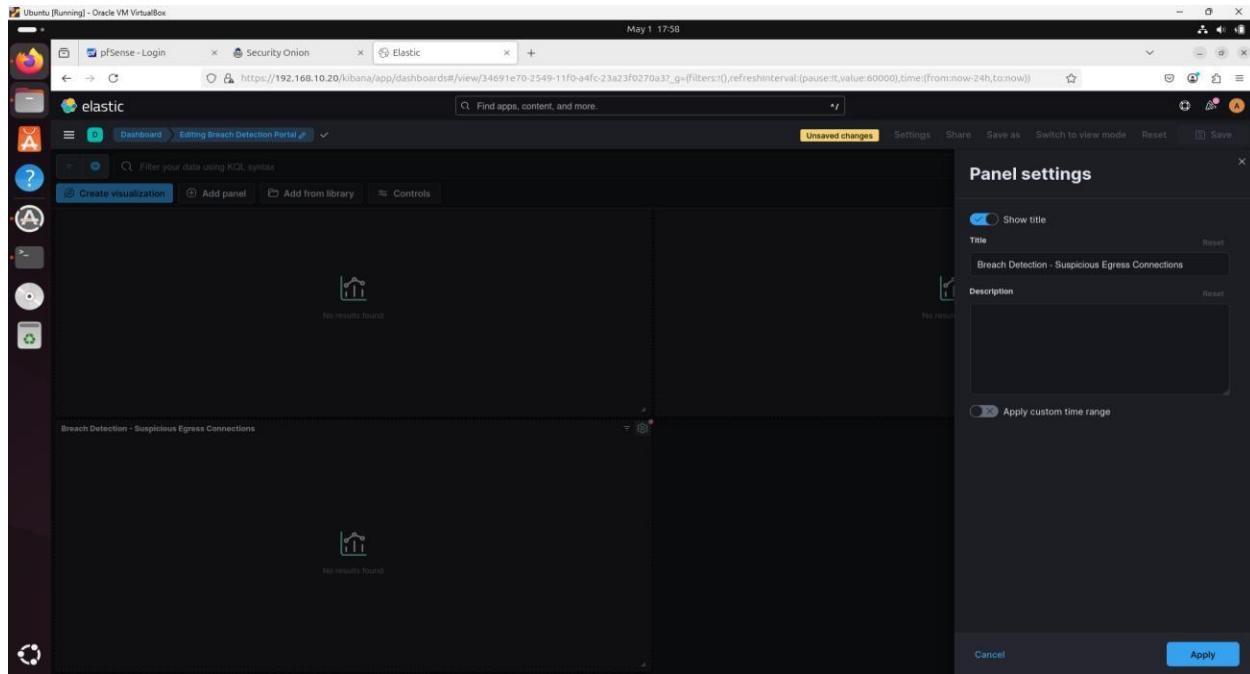
7. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **connection.state_description.keyword** as the **Field** selection. Set the **Size** option to **50**.

The screenshot shows the Kibana visualization editor interface. The URL in the browser is [https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=\(filters:\[\],refreshInterval:\(pause:0,value:60000\),time:\(from:now-24h/ago,to:now\)\)](https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=(filters:[],refreshInterval:(pause:0,value:60000),time:(from:now-24h/ago,to:now))). The left sidebar shows icons for pfSense, Security Onion, and Elastic. The main area has tabs for 'TRY IT', 'Edit visualization in Lens', 'Inspect', 'Share', and 'Save'. A search bar at the top says 'Find apps, content, and more.' Below it, there's a 'Visualize Library' section with 'Create' and a search bar 'Filter your data using KQL syntax'. The main configuration panel is titled 'Security Onion - Connections' and shows a 'Data' tab selected. It contains a 'Split rows' aggregation with 'connection.state_description.keyword' as the field and 'Size' set to 50. Other options like 'Sub aggregation', 'Field', 'Order by', and 'Advanced' are also visible.

8. This generates a summary view of all connection logs. We want to view established connections, but egress only. In order to do this, add the following filters: Filter the data table on **connection.local.originator: true**—this filters the view to show connections originating from the local network (Industrial Zone) only. Filter the data table on **connection.local.responder: false**— this filters the view to show connections to external destinations (enterprise and internet) only.

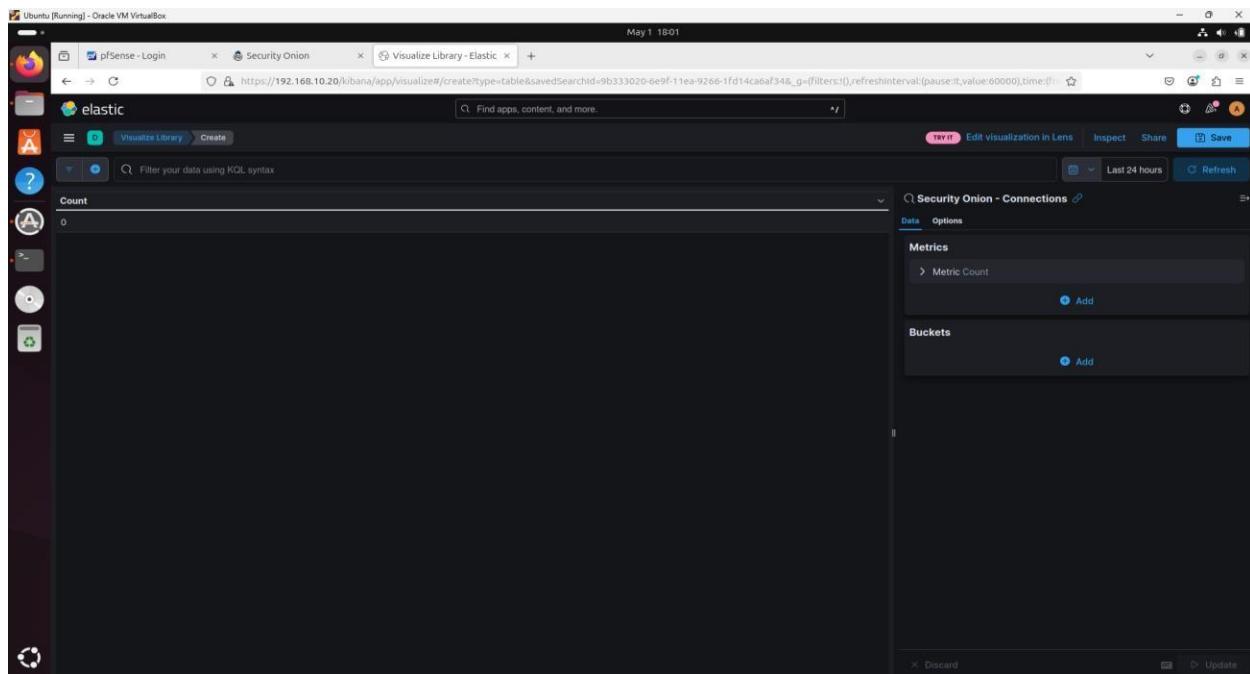
The screenshot shows the Kibana visualization editor interface, similar to the previous one but with additional filters applied. The URL is the same: [https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=\(filters:\[\],refreshInterval:\(pause:0,value:60000\),time:\(from:now-24h/ago,to:now\)\)](https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=(filters:[],refreshInterval:(pause:0,value:60000),time:(from:now-24h/ago,to:now))). The left sidebar and main interface are identical to the first screenshot, but the 'Data' tab shows the 'connection.state_description.keyword' field with a 'Size' of 50, and two filters are listed in the 'Filters' section: 'connection.local.originator: true' and 'connection.local.responder: false'.

9. Make sure to sort by **Count** (descending) and click on **Save**, saving the widget as **Breach Detection – Suspicious Egress Connections**.

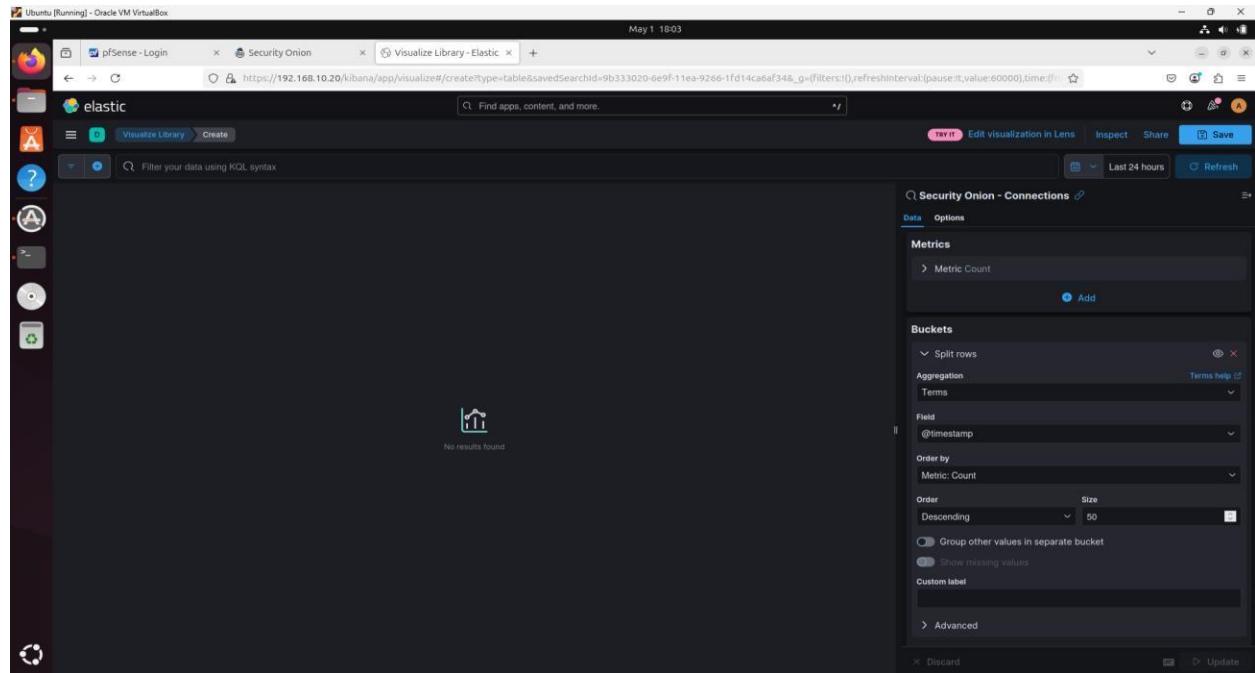


8.8: Suspicious Ingress Connections

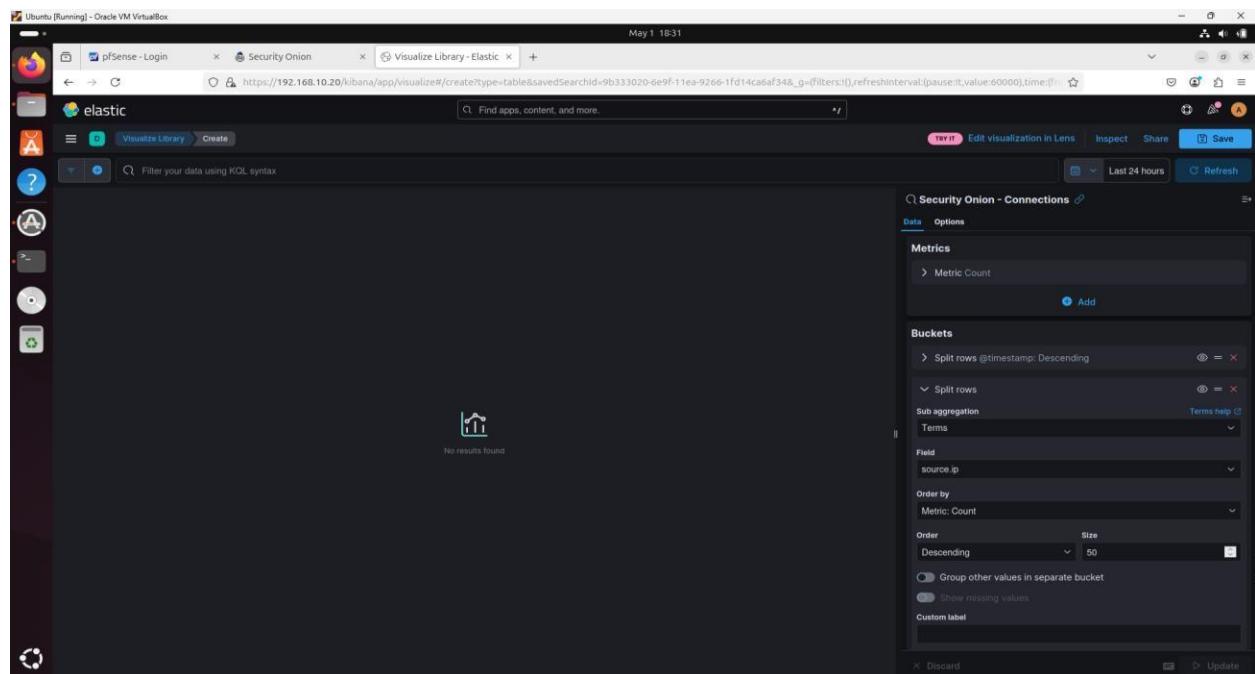
1. I created another **Data Table** using the **Connections** data source.



2. I added **Split Rows** for @timestamp, source.ip, destination.ip, destination.port, network.transport.keyword, event.duration, network.bytes, and connection.state_description.keyword.



3. I filtered the logs using NOT source.ip: 172.25.100.0/24 to remove local sources.



- I applied a second filter using connection.local.responder: true to show incoming traffic to internal systems.

The screenshot shows a Kibana interface with a dark theme. At the top, there are tabs for 'pfSense - Login', 'Security Onion', and 'Visualize Library - Elastic'. The URL in the address bar is [https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-926b-1fd14ca6af34&_g=\(filters:\[\],refreshInterval:\(pause:0,value:60000\),time:\[from:now-24h,to:now\]\)](https://192.168.10.20/kibana/app/visualize#/create?type=table&savedSearchId=9b333020-6e9f-11ea-926b-1fd14ca6af34&_g=(filters:[],refreshInterval:(pause:0,value:60000),time:[from:now-24h,to:now])). The main area is titled 'Visualize Library - Elastic' with a 'Create' button. A search bar at the bottom says 'Filter your data using KQL syntax'. On the right, there's a configuration panel for a visualization titled 'Security Onion - Connections'. It includes sections for 'Metrics' (Metric Count), 'Buckets' (Split rows @timestamp: Descending, Split rows source.ip: Descending, Split rows destination.ip: Descending), 'Sub aggregation' (Terms), 'Field' (destination.ip), 'Order by' (Metric: Count), 'Order' (Descending), and 'Size' (50). There are also checkboxes for 'Group other values in separate bucket' and 'Show missing values'. Buttons for 'Discard' and 'Update' are at the bottom.

- After sorting by duration in descending order, I saved the widget as **Breach Detection – Suspicious Ingress Connections**.

This screenshot is identical to the one above, showing the 'Visualize Library - Elastic' interface. The configuration panel for 'Security Onion - Connections' is visible, with the 'Order by' field set to 'Metric: Count' and 'Order' set to 'Descending'. The 'Size' is set to 50. The 'Save' button is highlighted in blue at the top right of the panel. The overall layout and elements are the same as the first screenshot.

6. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **network.transport.keyword** as the **Field** selection. Set the **Size** option to **50**.

The screenshot shows the Kibana Visualize Library interface. A search bar at the top right contains the URL: https://192.168.10.20/kibana/app/visualizer#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=(filters:[],refreshInterval:(pause:0,value:60000),time:(from:now-24h/ago,to:now))&_t=Security%20Onion%20-%20Connections. The main area displays a table with the heading "No results found". On the right, the "Data" tab of the configuration panel is selected, showing the following settings:

- Data:** Split rows @timestamp: Descending
- Options:** Split rows source.ip: Descending, Split rows destination.ip: Descending, Split rows destination.port: Descending
- Sub aggregation:** Terms help (Terms)
- Field:** network.transport.keyword
- Order by:** Metric: Count
- Order:** Descending, Size: 50
- Advanced:** Group other values in separate bucket, Show missing values

7. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **event.duration** as the **Field** selection. Set the **Size** option to **50**.

The screenshot shows the Kibana Visualize Library interface. A search bar at the top right contains the URL: https://192.168.10.20/kibana/app/visualizer#/create?type=table&savedSearchId=9b333020-6e9f-11ea-9266-1fd14ca6af34&_g=(filters:[],refreshInterval:(pause:0,value:60000),time:(from:now-24h/ago,to:now))&_t=Security%20Onion%20-%20Connections. The main area displays a table with the heading "No results found". On the right, the "Data" tab of the configuration panel is selected, showing the following settings:

- Data:** Split rows destination.ip: Descending
- Options:** Split rows destination.port: Descending, Split rows network.transport.keyword: Descending
- Sub aggregation:** Terms help (Terms)
- Field:** event.duration
- Order by:** Metric: Count
- Order:** Descending, Size: 50
- Advanced:** Group other values in separate bucket, Show missing values

8. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **network.bytes** as the **Field** selection. Set the **Size** option to **50**.

The screenshot shows the Kibana Visualize Library interface. On the left, there's a sidebar with icons for pfSense, Security Onion, and elastic. The main area has tabs for 'Visualize Library' and 'Create'. A search bar at the top says 'Find apps, content, and more.' Below it, a date range selector shows 'May 1 18:38' and 'Last 24 hours'. The right side displays a visualization titled 'Security Onion - Connections'. The configuration pane shows:

- Data Options: Split rows @timestamp: Descending
- Sub aggregation: Terms
- Field: network.bytes
- Order by: Metric: Count
- Order: Descending
- Size: 50
- Checkboxes for 'Group other values in separate bucket' and 'Show missing values'

No results found is displayed below the visualization.

9. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **connection.state_description.keyword** as the **Field** selection. Set the **Size** option to **10**.

The screenshot shows the Kibana Visualize Library interface. The layout is identical to the previous screenshot, with the same sidebar and search bar. The visualization titled 'Security Onion - Connections' is shown, but the configuration pane has been updated:

- Data Options: Split rows @timestamp: Descending
- Sub aggregation: Terms
- Field: connection.state_description.keyword
- Order by: Metric: Count
- Order: Descending
- Size: 10
- Checkboxes for 'Group other values in separate bucket' and 'Show missing values'
- A 'Custom label' input field is present.
- An 'Advanced' button is visible at the bottom.

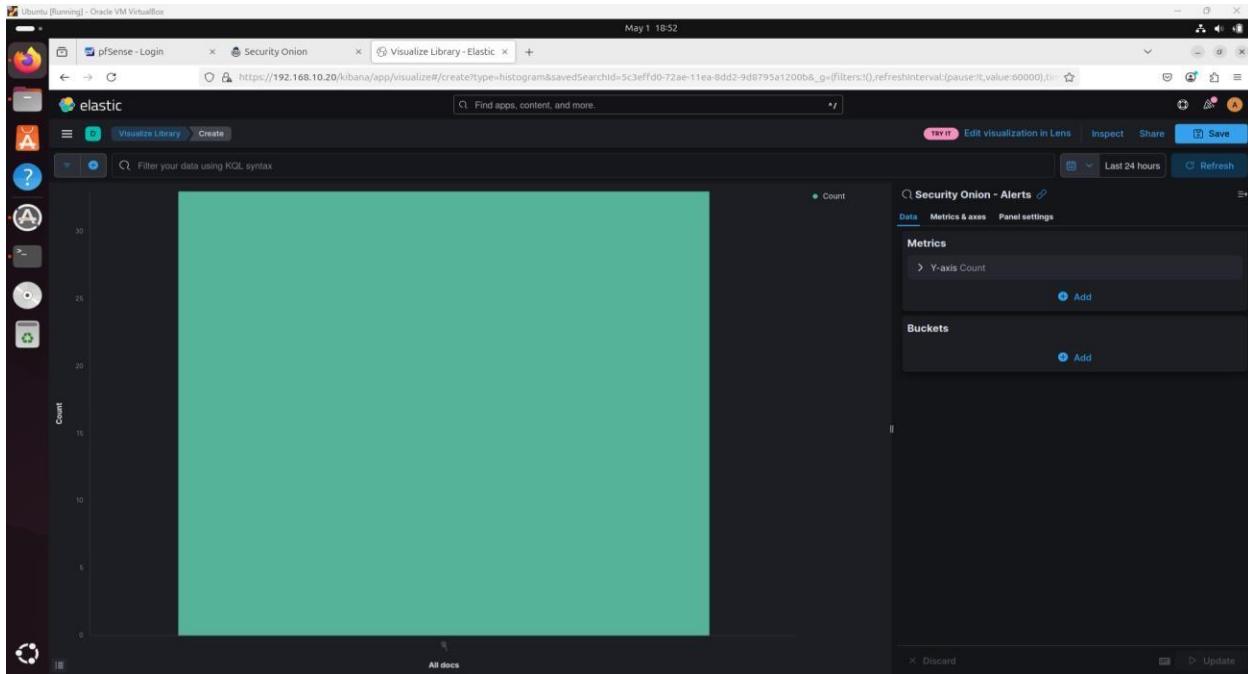
No results found is displayed below the visualization.

10. This generates a summary view of all connection logs. We want to view established connections, but egress only. To do this, add the following filters: Enter the following search string for the view: **NOT source.ip: 172.25.100.0/24**. This will filter out any IP addresses that are local to the industrial network (adapt this to your subnet). Filter the data table on **connection.local.responder: true**—this filters the view to show connections to local destinations (Industrial Zone) only.

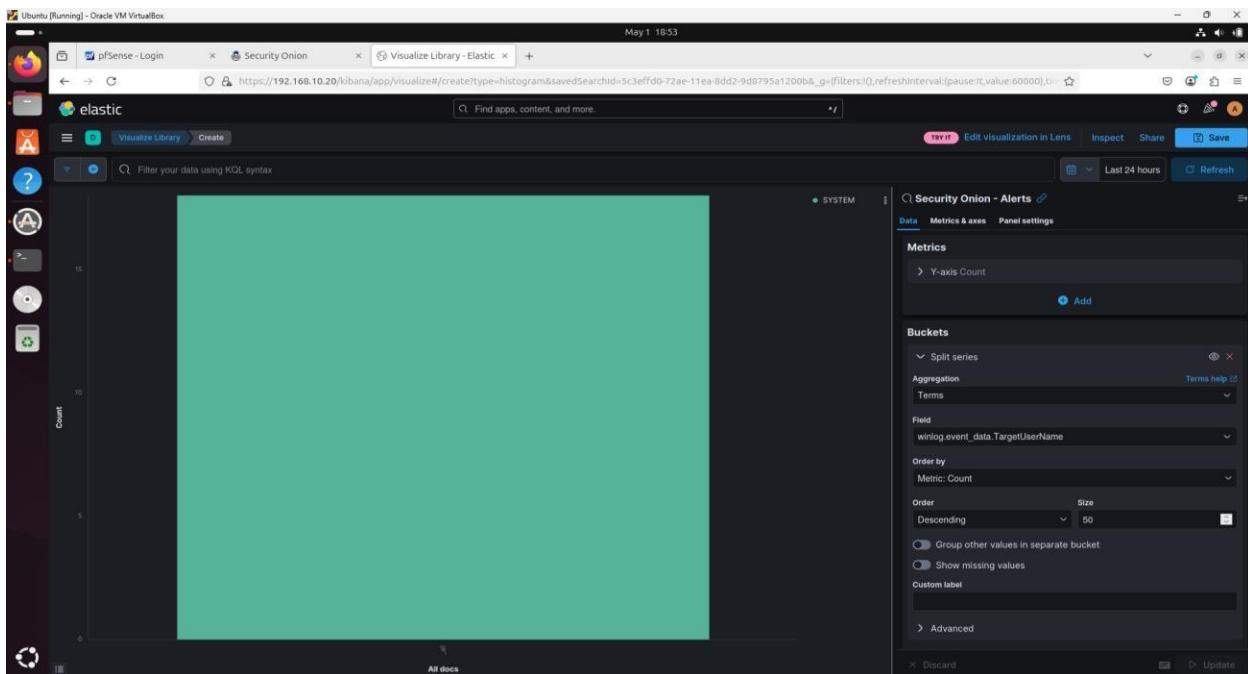
11. Make sure to sort by **Duration** (descending) and click on **Save**, saving the widget as **Breach Detection – Suspicious Ingress Connections**.

8.9: Failed User Login Attempts

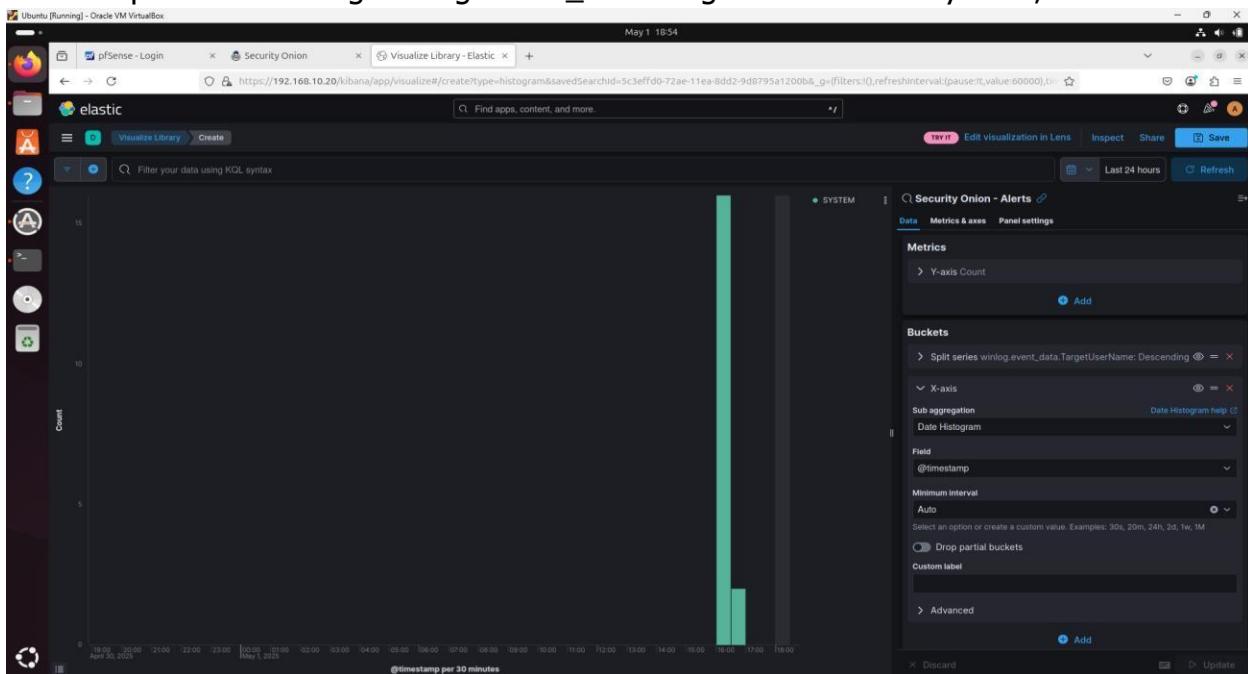
- I added a **Vertical Bar** visualization using **Security Onion – Alerts**.



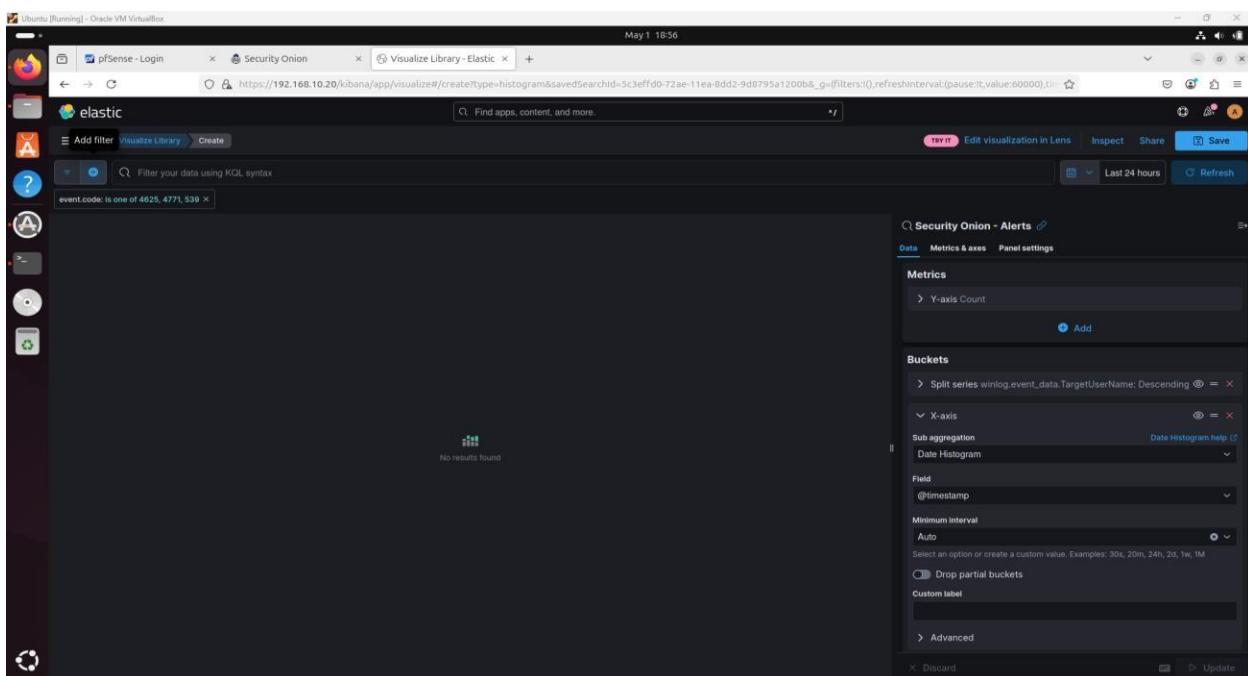
- I set a **Split Series** using `winlog.event_data.targetUserName.keyword`, **size 50**.



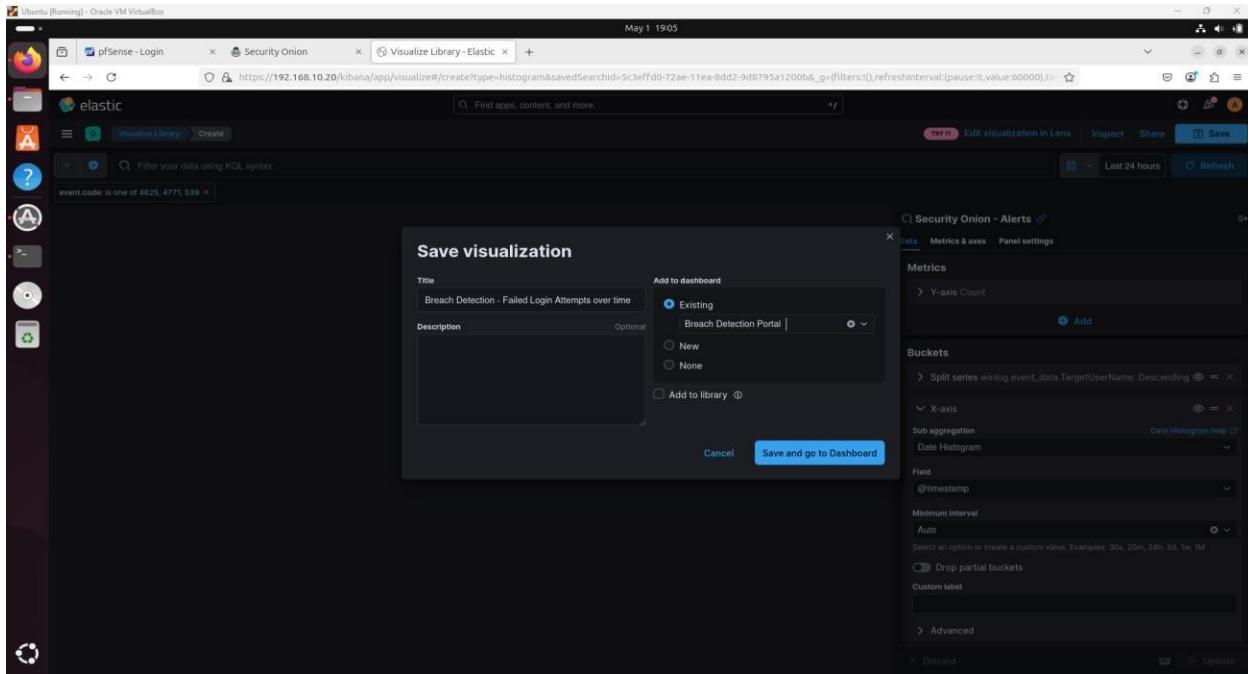
3. I set a Split Series using winlog.event_data.targetUserName.keyword, size 50.



4. I filtered by event codes 4625, 4771, and 539 which indicate failed login attempts.

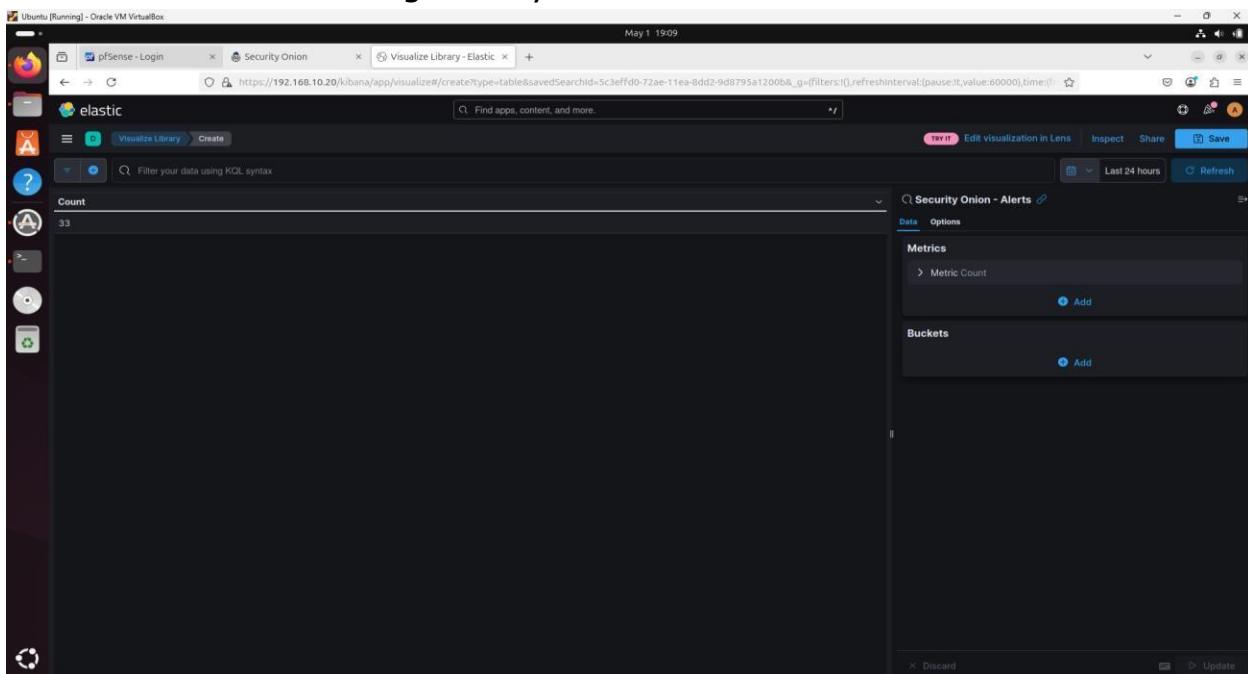


5. I saved this chart as Breach Detection – Failed Login Attempts Over Time.



8.10: New user creation and changes to user accounts

1. I created a Data Table using Security Onion – Alerts as the data source.



2. I added **Split Rows** buckets for rule.name.keyword, winlog.event_data.subjectUserName.keyword, winlog.event_data.targetUserName.keyword, and agent.name.keyword.

The screenshot shows a Kibana interface with a table visualization titled "Security Onion - Alerts". The table lists event types and their counts:

rule.name.keyword	Count
Windows Logon Success	18
Service startup type was changed	4
Software Protection service scheduled successfully	2
System time changed	2
License Activation (slu.exe) failed	1
The database engine attached a database	1
The database engine has completed recovery steps	1
The database engine is initiating recovery steps	1
The database engine is replaying log file C:\Winnnt\system32\winsl50.log	1
The database engine is starting a new instance	1

The right panel shows the configuration for a "Split rows" bucket on the "rule.name.keyword" field, ordered by count in descending order. The "Metrics" section shows a metric count of 18.

3. I filtered by event codes: 4720, 4738, 4732, 4728, and 4722—each corresponding to account creation, privilege changes, or enabling user accounts.

The screenshot shows a Kibana interface with a table visualization titled "Security Onion - Alerts". The results are filtered to show "No results found". The right panel shows the configuration for a "Split rows" bucket on the "winlog.event_data.SubjectUserName" field, ordered by count in descending order. The "Metrics" section shows a metric count of 0.

4. I saved this as **Breach Detection – User Account Alerts Summary**.

The screenshot shows the Kibana interface on an Ubuntu desktop. The title bar says "Ubuntu [Running] - Oracle VM VirtualBox". The URL in the address bar is [https://192.168.10.20/kibana/app/visualizer#/create?_type=table&savedSearchId=5c3effd0-72ae-11ea-8dd2-9d8795a1200ba_g=\(filters:\[\],refreshInterval:\(pause:0,value:60000\),time:\[\]\)&version=1](https://192.168.10.20/kibana/app/visualizer#/create?_type=table&savedSearchId=5c3effd0-72ae-11ea-8dd2-9d8795a1200ba_g=(filters:[],refreshInterval:(pause:0,value:60000),time:[])&version=1). The visualization is titled "Security Onion - Alerts". The configuration pane shows the following settings:

- Metrics**: Metric Count
- Buckets**:
 - Split rows rule.name.keyword: Descending
 - Split rows winlog.event_data.SubjectUserName: Descending
 - Split rows
- Sub aggregation**: Terms
- Field**: winlog.event_data.TargetUserName
- Order by**: Metric: Count
- Order**: Descending, Size: 50
- Advanced**: Group other values in separate bucket, Show missing values

5. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **agent.name.keyword** as the **Field** selection. Set the **Size** option to **50**.

The screenshot shows the Kibana interface on an Ubuntu desktop. The visualization is now titled "Security Onion - Alerts". The configuration pane has been updated:

- Metrics**: Split rows rule.name.keyword: Descending
- Buckets**:
 - Split rows winlog.event_data.SubjectUserName: Descending
 - Split rows winlog.event_data.TargetUserName: Descending
 - Split rows
- Sub aggregation**: Terms
- Field**: agent.name.keyword
- Order by**: Metric: Count
- Order**: Descending, Size: 50
- Advanced**: Group other values in separate bucket, Show missing values

6. click on the **Add Filter** button, then set the **Field** option to **event.code** and the **Operator** option to **is one of**. Set the values to the following: **4720**—This is the event ID for new user creation; **4738**—This is the event ID for account changes; **4732**—This is the event ID indicating a user was added to a local privileged group; **4728**—This is the event ID indicating a user was added to a global privileged group; **4722**—This is the event ID indicating a user account was enabled.

7. Click on **Save**, saving the widget as **Breach Detection – User Account Alerts Summary**.

8.11: Downloaded files

- I attempted to create a **Data Table** using *:so-* with the filter event.dataset:http.

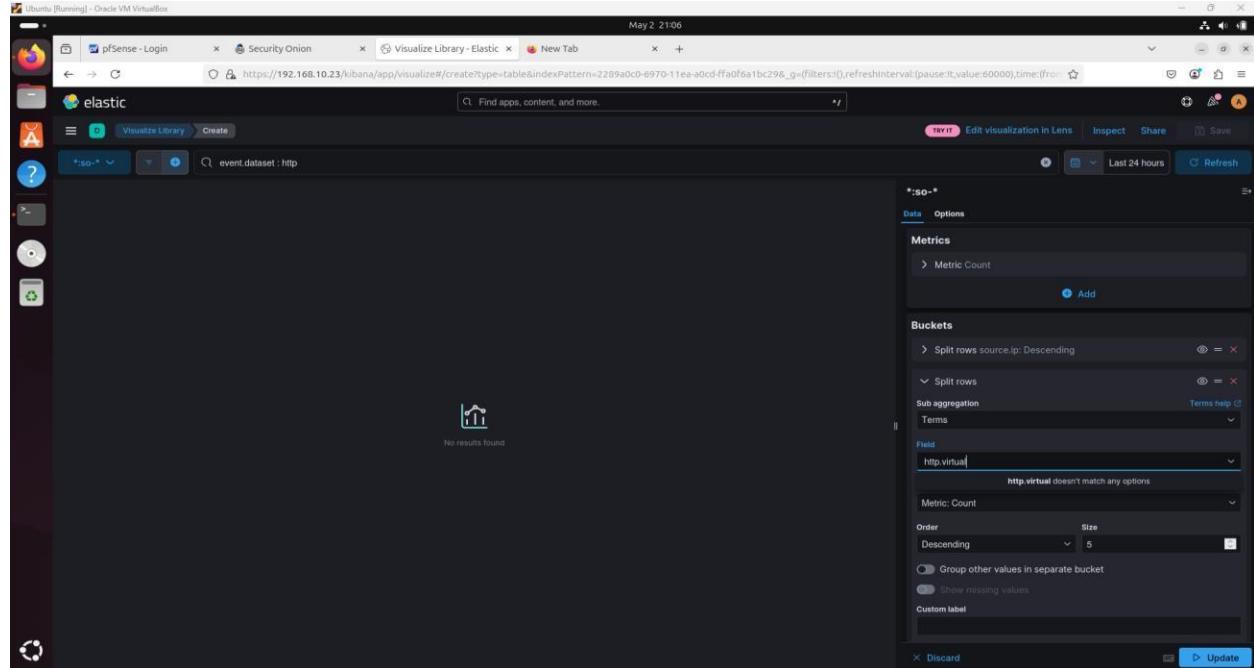
The screenshot shows the Kibana Visualize Library interface. A table visualization titled "event.dataset: http" is displayed. The table has one row with the value "38,782". The visualization settings on the right side show a filter of "*:so-*" and a metric count of "Metric Count". The URL in the browser is [https://192.168.10.20/kibana/app/visualize#/create?type=Table&indexPattern=2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g=\(filters:\[\],refreshInterval:\(pause:lt,value:60000\),time:\(from:now-1y,until:now\)\)](https://192.168.10.20/kibana/app/visualize#/create?type=Table&indexPattern=2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g=(filters:[],refreshInterval:(pause:lt,value:60000),time:(from:now-1y,until:now))).

- I added split rows for source.ip and http.virtual_host.keyword.

The screenshot shows the Kibana Visualize Library interface. A table visualization titled "event.dataset: http" is displayed. The table has one row with the value "0". The visualization settings on the right side show a filter of "*:so-*" and a metric count of "Metric Count". The URL in the browser is [https://192.168.10.20/kibana/app/visualize#/create?type=Table&indexPattern=2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g=\(filters:\[\],refreshInterval:\(pause:lt,value:60000\),time:\(from:now-1y,until:now\)\)](https://192.168.10.20/kibana/app/visualize#/create?type=Table&indexPattern=2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g=(filters:[],refreshInterval:(pause:lt,value:60000),time:(from:now-1y,until:now))).

3. Due to a misconfigured Zeek service, HTTP logs were not being collected, so I was unable to complete this visualization.
4. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **http.virtual_host.keyword** as the **Field** selection.

Note: Even after multiple tries, I couldn't get the http logs to be collected since zeek wasn't configured properly. I've tried configuring it saltstack and node.cfg but nothing worked. So skipping this virtualization.



8.12: SilentDefense alerts

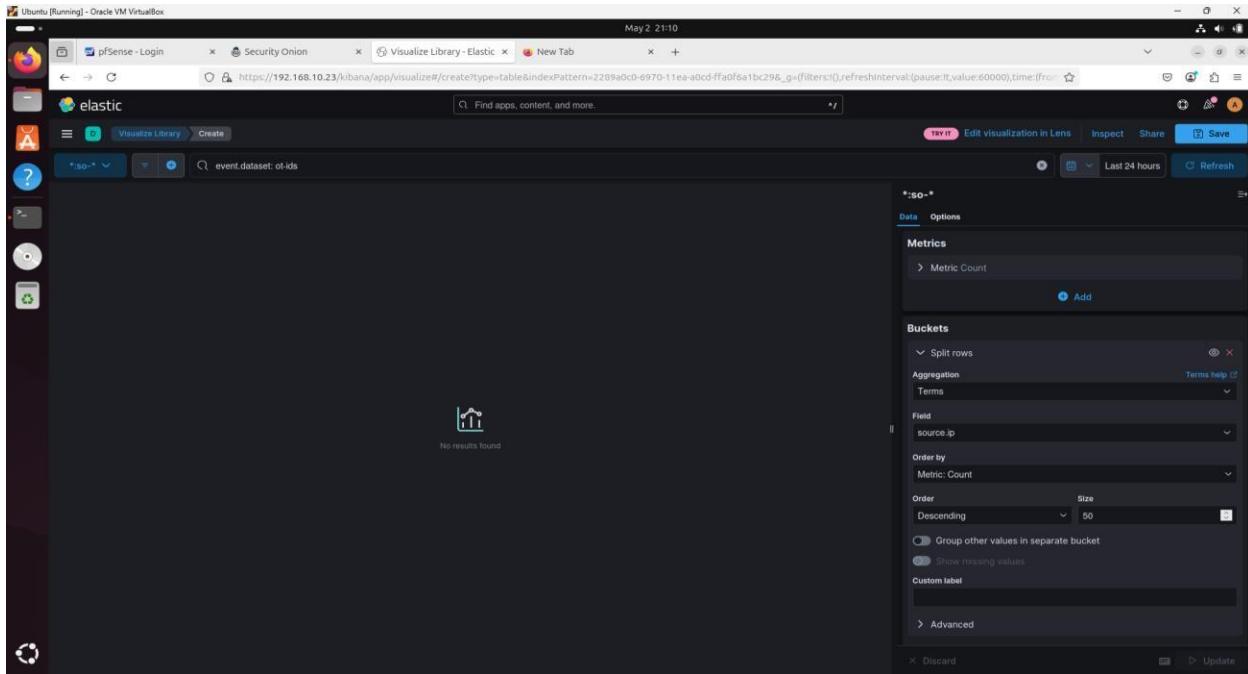
- I created a **Data Table** using `*:so-*` as the data source and filtered with `event.dataset:ot-ids`.

The screenshot shows the Kibana Visualize Library interface. A Data Table visualization is displayed, using the `*:so-*` dataset. The search bar at the top contains the query `event.dataset:ot-ids`. The visualization shows a single metric named "Count" with a value of 6,991. The right panel contains sections for "Metrics" (Metric Count) and "Buckets". The "Metrics" section has an "Add" button. The "Buckets" section also has an "Add" button. The bottom right corner of the visualization panel has "Discard" and "Update" buttons.

- Enter an `event.dataset:ot-ids` search term to filter out SilentDefense logs only, then hit **Update**.

The screenshot shows the Kibana Visualize Library interface again. The search bar now contains the query `event.dataset:ot-ids`. The visualization shows the "Count" metric with a value of 0. The right panel remains the same with "Metrics" and "Buckets" sections and their respective "Add" buttons. The bottom right corner still has "Discard" and "Update" buttons.

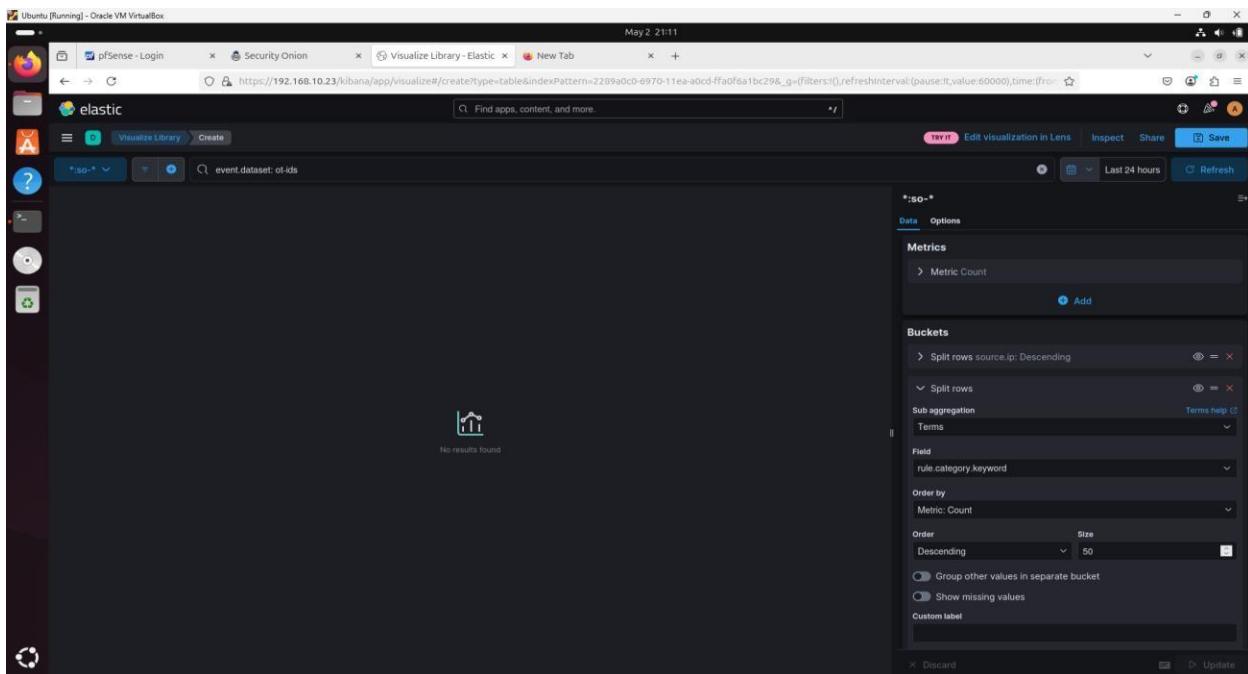
3. I added split rows for `source.ip`, `rule.category.keyword`, and `rule.name.keyword`, each with size 50.



The screenshot shows the Kibana visualization configuration interface. The URL in the browser is `https://192.168.10.23/kibana/app/visualize#/create?type=table&indexPattern=_2289a0c0-6970-11ea-a0cd-ffa0f6a1bc298_g=filterId0,refreshInterval:[pause:lt,value:60000],time:(from:now-24h, to:now)`. The main panel displays a visualization with a single bucket labeled "source.ip". The configuration pane on the right shows the following settings:

- Buckets**: Split rows, Aggregation Terms, Field source.ip, Order by Metric: Count, Order Descending, Size 50.
- Advanced** section is collapsed.

4. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with `rule.category.keyword` as the **Field** selection. Set the **Size** option to **50**.



The screenshot shows the Kibana visualization configuration interface after adding a second bucket. The URL in the browser is the same as the previous screenshot. The main panel displays a visualization with two buckets: "source.ip" and "rule.category.keyword". The configuration pane on the right shows the following settings:

- Buckets**: Split rows source.ip: Descending, Sub aggregation Terms, Field rule.category.keyword, Order by Metric: Count, Order Descending, Size 50.
- Advanced** section is collapsed.

5. Add a **Split Rows** data bucket and set **Aggregation** to **Terms**, with **rule.name.keyword** as the **Field** selection. Set the **Size** option to **50** and click **Update**.

The screenshot shows the Kibana Visualize Library interface. A search bar at the top right contains the query "event.dataset: ol-ids". On the left, there's a sidebar with icons for pfSense, Security Onion, and elastic. The main area displays a table with two rows, both of which have a status of "No results found". To the right of the table is the configuration panel for a visualization titled "*:SO-*". The configuration includes:

- Data** tab selected.
- Metrics**: Metric Count.
- Buckets**:
 - Split rows source.ip: Descending
 - Split rows rule.category.keyword: Descending
 - Split rows
 - Sub aggregation: Terms
 - Field: rule.name.keyword
 - Order by: Metric: Count
 - Order: Descending, Size: 50
 - Group other values in separate bucket
 - Show missing values
- Options** tab available.

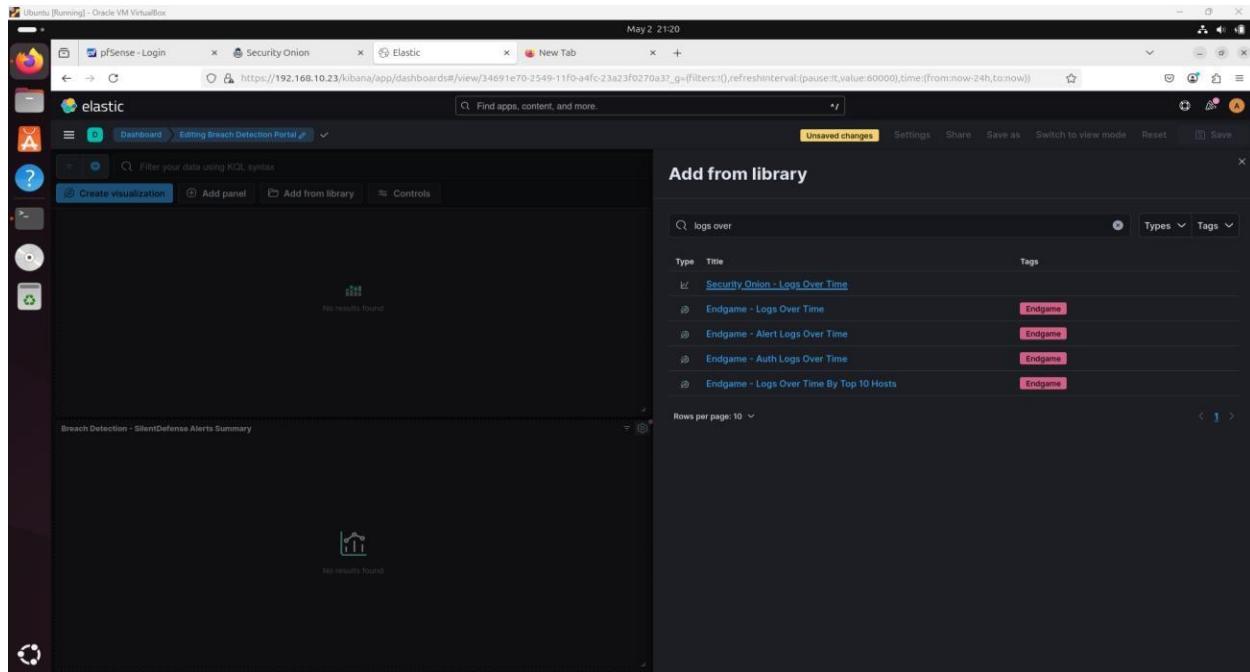
6. I saved the visualization as **Breach Detection – SilentDefense Alerts Summary**.

The screenshot shows the Kibana Visualize Library interface with a "Save visualization" dialog box open over the visualization configuration. The dialog box has the following fields:

- Title**: Breach Detection - SilentDefense Alerts Summary
- Add to dashboard**: Existing (radio button selected), dropdown menu set to "Breach Detection Portal".
- Description**: An empty text area.
- Optional**: "Add to library" checkbox is unchecked.
- Buttons**: "Cancel" and "Save and go to Dashboard" (highlighted in blue).

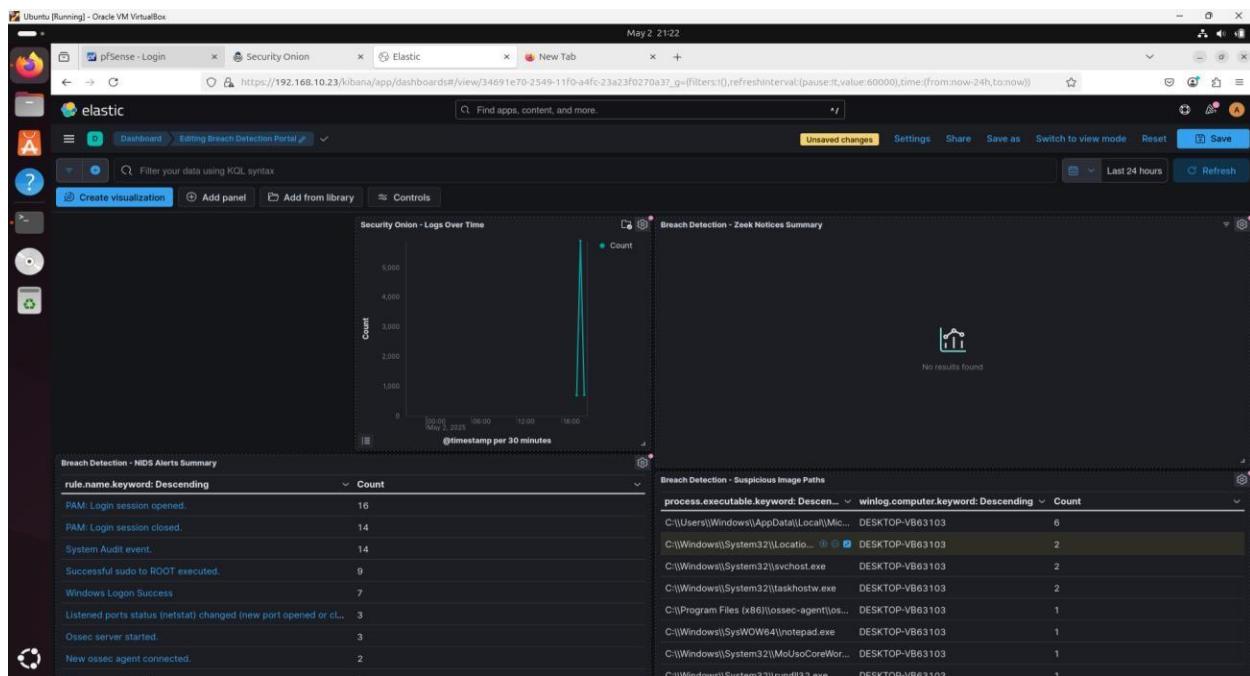
8.13: Finishing up the dashboard

- I clicked **Add from library** and inserted the **Security Onion – Logs Over Time** visualization.



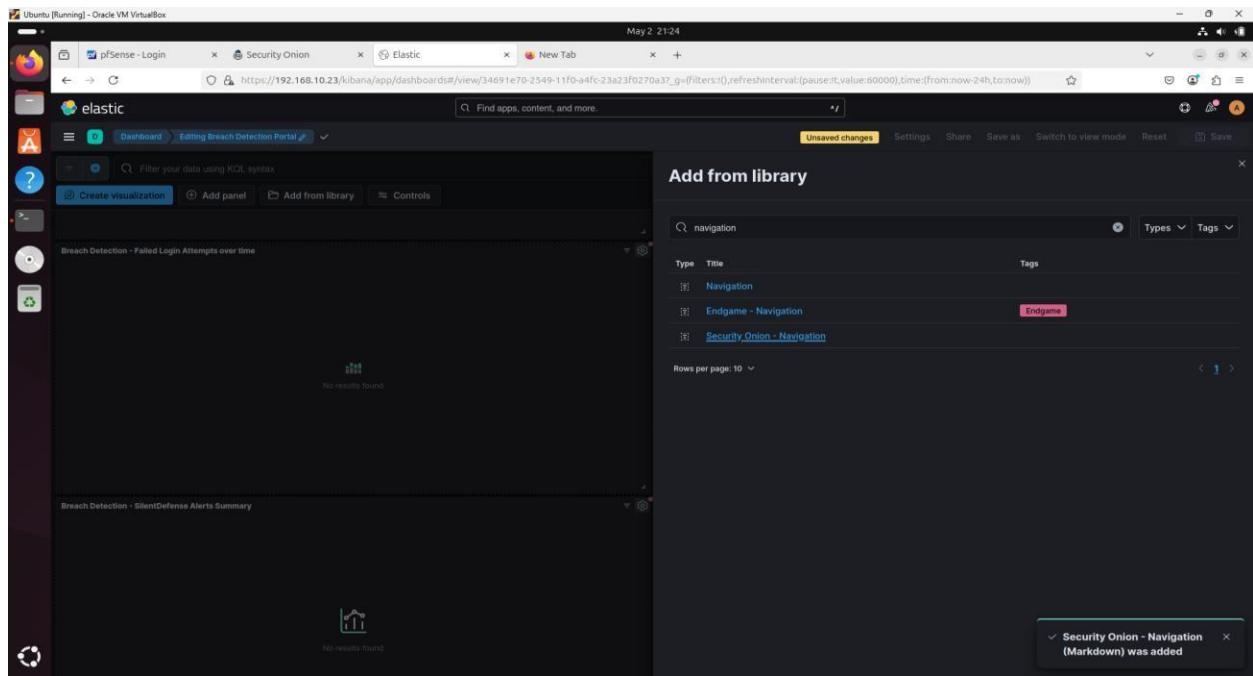
The screenshot shows the Kibana interface on an Ubuntu desktop. A search bar at the top right contains the query "logs over". Below it, a table lists several visualizations, each with a preview icon, title, type, and tags. The first item, "Security Onion - Logs Over Time", is highlighted. The interface includes a sidebar with icons for pfSense Login, Security Onion, and Elastic, and a main panel showing a visualization titled "Breach Detection - SilentDefense Alerts Summary" which displays "No results found".

- I moved this chart to the top center of the dashboard. screen.

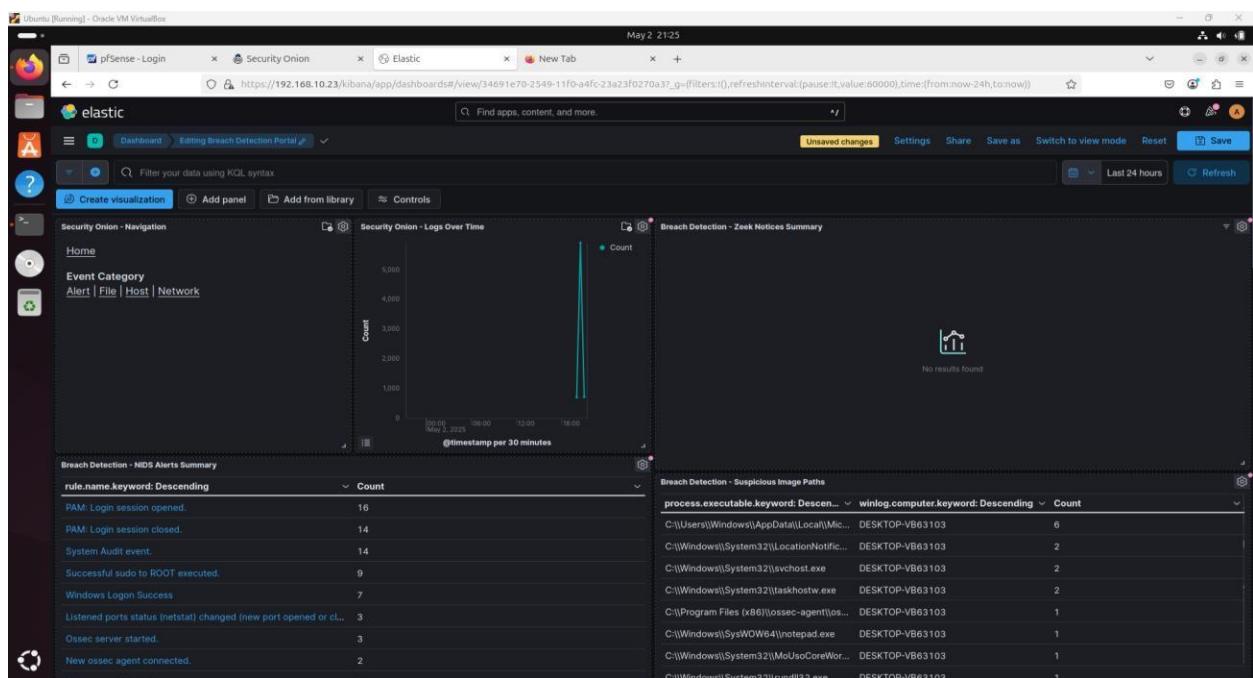


The screenshot shows the Kibana dashboard after the chart has been moved. The top center features the "Logs Over Time" visualization with the title "Security Onion - Logs Over Time" and subtitle "Count" on the Y-axis and "@timestamp per 30 minutes" on the X-axis. To the left is a table for "Breach Detection - NIDS Alerts Summary" and to the right is a table for "Breach Detection - Suspicious Image Paths". At the bottom, there is a table for "Breach Detection - Zesk Notices Summary". The interface includes a sidebar with icons for pfSense Login, Security Onion, and Elastic, and a main panel showing a visualization titled "Breach Detection - SilentDefense Alerts Summary" which displays "No results found".

3. I added the **Security Onion – Navigation Panel** and placed it to the top-left.



4. Then I added the **Security Onion – All Logs saved search** for deeper context.

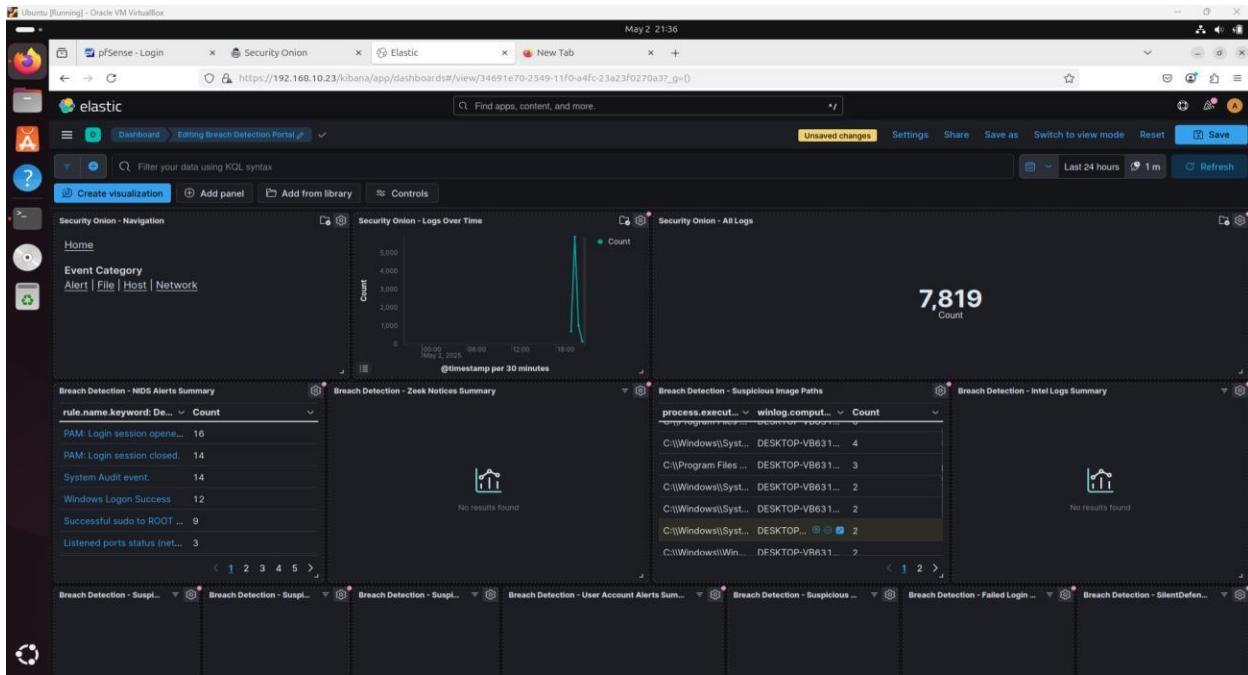


5. I configured the dashboard to show a 24-hour window and refresh every minute.

The screenshot shows the Kibana interface with the 'Add from library' modal open. The modal lists several pre-built dashboards and visualizations, all tagged with 'Endgame'. A message at the bottom right indicates that 'Security Onion - All Logs' (Metric) was added. The background shows the main Kibana dashboard area with some panels and search bars.

6. Set the dashboard to show a 24-hour timespan and automatically refresh every minute. Once you are happy with how things look, save the dashboard.

The screenshot shows a saved Kibana dashboard. The top navigation bar displays 'Last 24 hours' and '1m' as the time range. The dashboard contains three main panels: a chart titled 'Security Onion - Logs Over Time' showing a count of 7,600, a table titled 'Breach Detection - NIDS Alerts Summary' listing various alerts, and a table titled 'Breach Detection - Zeek Notices Summary' listing notices. The left sidebar shows navigation links like 'Home', 'Event Category', 'Alert', 'File', 'Host', and 'Network'.



- After reviewing the layout and confirming it looked clean and functional, I saved the full dashboard setup.

This dashboard can function as a starter to get a **security operations center (SOC)** view in place. This will likely change, so add and modify the dashboard to make it fit your needs, but this is a great starting point for your journey into a holistic security monitoring approach.

NAME: Phanindhar Reddy Karnati
ID: 001667635