Hands-on Labs: Security Analytics

Secure your Environment: Search, Analyze, Visualize and Monitor

Introduction

Using Sumo Logic for Security Analytics

These labs will provide you hands-on experience with Sumo Logic, where you will learn basic and advanced search operators to analyze your logs and metrics.

Please Note: These labs can be completed as a standalone item; however, they are designed to complement our **Sumo Logic Security Analytics course**. If you haven't done so, for your benefit, please register and attend this webinar before completing these labs.

Accessing the Training Environment

These labs are meant to be done in our Training environment using curated sample data. However, you are welcome to use your own environment by editing the query samples to fit your data and metadata.
NOTE: Because of data variability and timing, results in the Training instance might not exactly match those listed in these pages.

To access the Sumo Logic Training environment:

1. Go to: https://service.sumologic.com
2. Use the following credentials:
   User: training+labs@sumologic.com (DO **NOT** copy/paste, as PDFs add extra characters)
   Password: Sum0Labs! (DO **NOT** copy/paste, as PDFs add extra characters)

You can also use any of these other users, with the same password:

- training+user100@sumologic.com
- training+user200@sumologic.com
- training+user300@sumologic.com

Search and Analyze

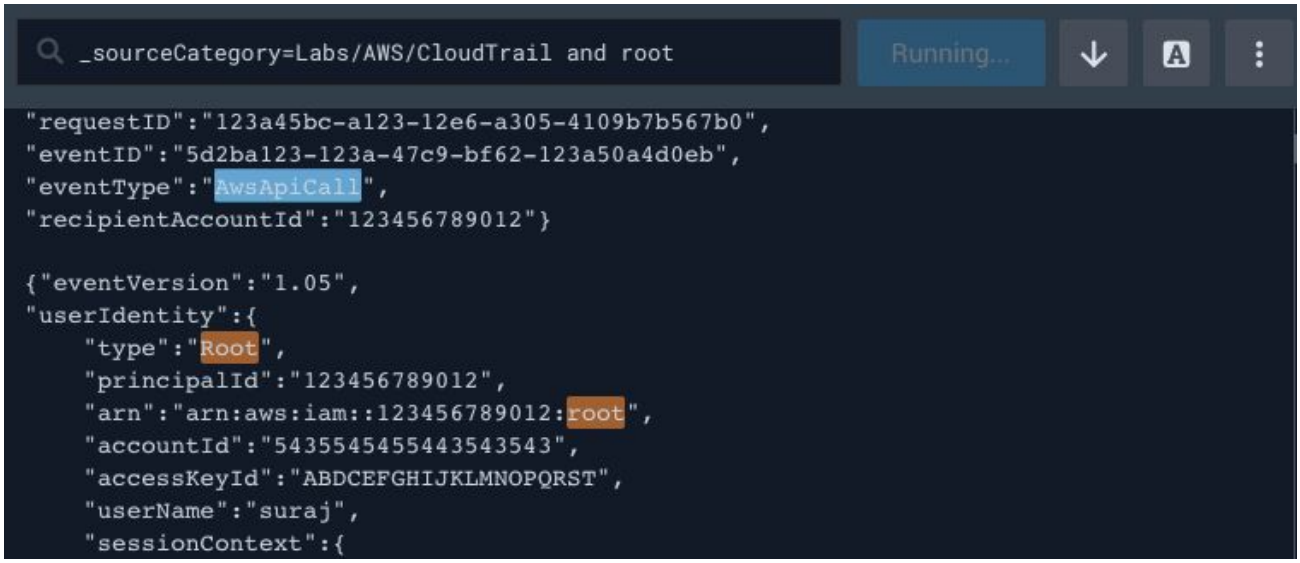
Lab 1 - Search Basics: Metadata and Keywords

In this lab, you will learn the use of metadata and keywords to narrow your search scope and improve performance.

1. Search for all messages for the last 6 hours with _sourceCategory=Labs/AWS/CloudTrail that contain the word "root".
   ```
   _sourceCategory=Labs/AWS/CloudTrail and root
   ```

2. Search for messages across all your AWS data that contain the word "root".
   ```
   _sourceCategory=Labs/AWS/* and root
   ```

3. Run a Live Tail session for the same query as #1. Highlight the words "root" and "AwsApiCall"



QUIZ: True or False

1. Keywords are case-sensitive
2. AND is implicit and OR is explicit
3. Keywords and metadata can use wildcards
4. Live Tail must contain at least one metadata tag

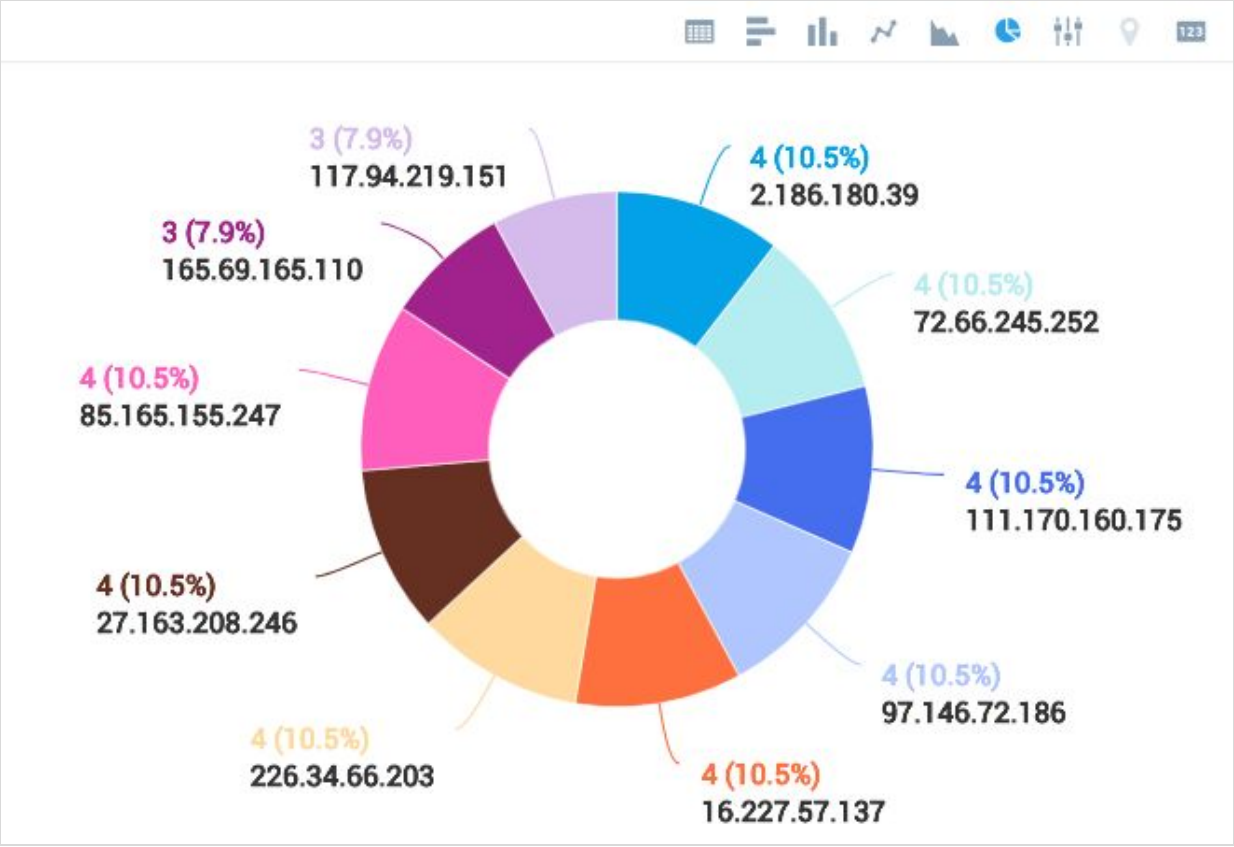
Lab 2 - Simple Parsing, Grouping, and Filtering

Learn basic operators to parse and group your search results.

1. Search CloudTrail logs to identify the top 10 IP addresses for the US West Region.

```
_sourceCategory=Labs/AWS/CloudTrail
// You can use the json operator on json-formatted logs
| json "awsRegion"
| where awsregion="us-west-1"
// You can use a mix of parsing operators in the same query
| parse regex "(?<ip_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| count by ip_address
| top 10 ip_address by _count
```

You can chart your results by choosing any of the available charting options.



Lab 3 - Parsing Options

Parsing your logs allow you to provide structure to your messages, identifying the fields that are meaningful to you.

1. The nodrop option for the parse operator allow users to include messages in your results that do not meet the pattern criteria or a parse statement. In this example, messages from the first parse statement will be dropped, so they can be parsed by the second parse statement.

```
_sourceCategory=Labs/Apache/Error
| parse "[client *]" as client_ip nodrop
| parse "mod_log_sql: *" as message
//| where isBlank(client_ip)
```

2. The parse field option allows you to do further parsing on an already extracted field. In this example, we want to identify the top 5 Sumo Logic committers in GitHub. We start by searching for committers in the last 30 days, and parse their email address. We then use the parse field option to further parse the email address into user and domain, select only those users we care for, and lastly, count by user and identify the top 5 committers.

```
_sourceCategory=Labs/Github and "committer"
| parse "\"email\":\"*\"" as email
| parse field=email "*@*" as users, domain
| where domain="sumologic.com"
| count by users
| top 5 users by _count
```

3. The parse multi option allows you to extract multiple occurrences of the same pattern within one message. By default, parse only extracts the first occurrence. Notice how each message is repeated for each occurrence of an ip address, allowing you to do accurate counts.

```
_sourceCategory=labs/snort
| parse regex "(?<ip_address>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})" multi
```

Pre-parsing your Messages with Field Extraction Rules

4. Field Extraction Rules extract fields at the time the log messages are ingested. You can see all FERs available (and their details) under : **Manage Data → Settings → Field Extraction Rules**.
   Taking advantage of the Apache Access rule already in place, run a search to identify the count of 404s by source ip.

```
_sourceCategory=Labs/Apache/Access and status_code=404
| count by src_ip
```

QUIZ: True or False

1. Parsing operators include csv, json, split, and keyvalue.
2. Once a field has been parsed, it cannot be parsed any further.
3. Fields parsed by the Field Extraction Rules are available in the Field Browser.

Simple Analytics

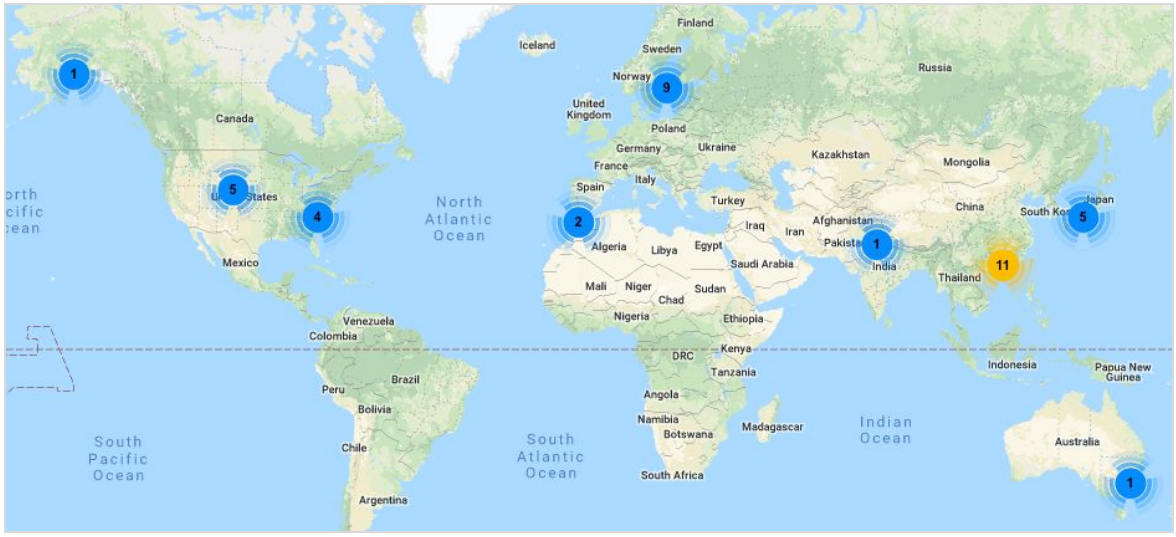
Lab 4 - Using Simple Operators in AWS Security Use Cases

Community Post: Security-Related Queries for AWS

1. From the Learn tab in Sumo Logic, select Community > Query Library and go to a post titled "*Security-related Queries for AWS*". Run each of the 3 queries in this post. Keep in mind that obtaining no results in these queries is not a bad thing. This means you do not have and security issues/breaches/potential issues in your data.
    - Lab 4A: Monitor AWS Root Account Usage
    - Lab 4B: Monitor Security Groups created with "Ingress Any" privileges
    - Lab 4C: Monitor a User's login from two different IP addresses

Bonus:

For Lab 4C, use the geo lookup operator to map locations of the IP addresses.



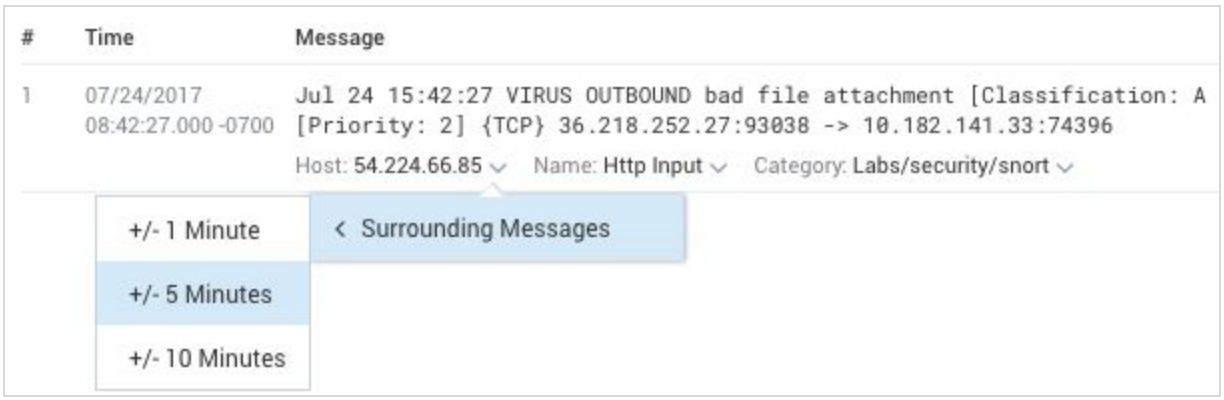
Advanced Analytics

Lab 5 - Find the "needle in the haystack"

Explore the functionality of LogReduce, which allows you to distill unique messages from the noise by identifying recurring Signatures in your data.

1. Run LogReduce on your Snort security data to identify unusual activity (i.e. intrusions) in the last 60 minutes.
```
_sourceCategory=labs/snort
| logreduce
```

2. Sort your results by count to identify those that happen only once. Click on the count (1) to view the unusual message. Now click on the host to view surrounding messages to identify the context of the intrusion.



Lab 6 - Compare Activity from Different Periods

Explore the functionality of LogCompare, which allows you to compare log activity from two different time periods, providing you insight on how your current time compares to a baseline. In this case, use **LogCompare** to identify when signature messages deviate by more than 25% from the baseline.

1. First, review summarized signatures for Snort messages in the last 60 minutes (Use LogReduce)
```
_sourceCategory=labs/snort
| logreduce
```

2. Now use LogCompare to run a summarized query for a baseline 24 hours ago (Click on LogCompare button)
```
_sourceCategory=labs/snort
| logcompare timeshift -24h
```

3. To view only those results where Delta Percentage is more than 25%, add a where clause for _deltaPercentage, which is one of the hidden fields available.
```
_sourceCategory=labs/snort
| logcompare timeshift -24h
| where abs(_deltaPercentage) > 25
```

4. To view results where there is a new Signature in the current time period, add a where clause for _isNew:
```
_sourceCategory=labs/snort
| logcompare timeshift -24h
| where (_isNew)
```

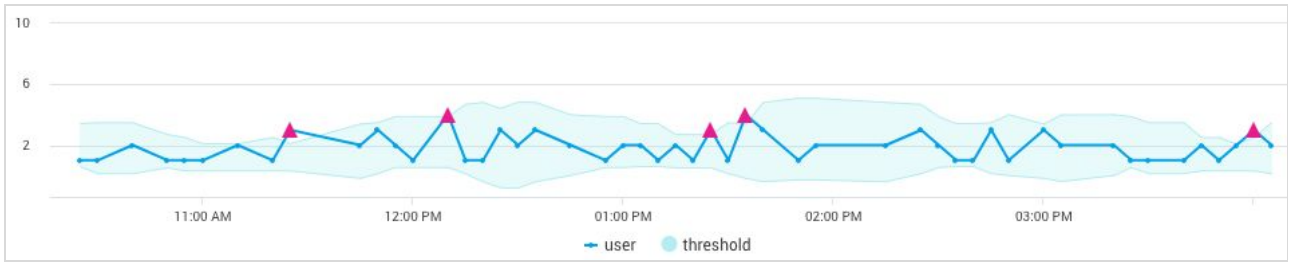
Lab 7 - Identify "out of the ordinary" Events

Explore the functionality of the outlier operator, which allows you to identify events outside of a threshold.

1. Search your Labs/OS/Linux/Security logs looking for anomalies in sudo users in the last 24 hours. In particular, you are looking for users that are not part of the whitelisted "sudoers". "Timeslicing" your results allow you to see a trend over time. Lastly, Plot your failed attempts on a line graph to provide a visual to easily spot outliers.

```
_sourceCategory=Labs/OS/Linux/Security and " user NOT in sudoers"
| parse regex "sudo:\s+(?<src_user>\S+)\s+:\s+user NOT in sudoers" nodrop
| timeslice 5m
| count (src_user) as user by _timeslice
| outlier user window=5, consecutive=1, threshold=2, direction=+-
```

2. Bonus: Edit settings for window, consecutive, threshold, and direction to see the change in behavior.



Lab 8 - Comparing Over Time

Check your firewall logs to identify a 2-fold increase in denied traffic. In this case, we will use the time compare operator, along with the timeshift option, to compare the current results of denied traffic to a base. Search for denied traffic for the last 24 hours, and compare it to the average count of denied traffic for the last 3 days.

```
_sourceCategory=Labs/PaloAltoNetworks and ",TRAFFIC," and action="deny"
| count action
| compare with timeshift -1d 3 avg
| if(isNull(_count), 0, _count) as _count
| if(isNull(_count_3d_avg), 0, _count_3d_avg) as _count_3d_avg
//Uncomment the following line to identify a count 2x that of your avg.
//| where _count > (2 * _count_3d_avg)
```

Lookups and Data Correlation

Lab 9 - Testing the Threat Intel Lookup with Sample Indicators Of Compromise (IOCs)

When using the Threat Intel app (which you will install in a later lab), no results is a good thing, as this means you do not have malicious threats in your logs. However, in order to test the lookup functionality, the Threat Intel FAQs provide samples for each type of IOC. Let's run a simple query with an actual IOC. you can replace the IP address with any other IOC in the FAQs.

```
* | limit 1 // these lines allow us to query just one result
| "158.69.196.112" as my_threat
| lookup type, actor, raw, threatlevel as malicious_confidence
from sumo://threat/cs on threat = my_threat
| where type in ("ip_address","email_address", "domain", "url", "file_name") and
!isNull(malicious_confidence)
```

Lab 10 - Creating a Query Template for the Threat Intel Lookup

Using the same query from the previous lab, let's create a simple query template for future use. We will create a parameter in our query so we can easily input an IOC for testing.

1. Highlight the IP address (including the double quotes) and click **Create a parameter**

2. In the dialog box, enter a Parameter Name and Description and click on Save

3. You can now use this Search Template to test any of the sample IOCs listed, or any of your own.

4. Lastly, don't forget to share this template with your team. Closing the query box allows you to only present the Parameters box, making it easier for non-technical users to simply enter a parameter and get query results.

Lab 11 - Creating Your Own Lookup

Using the save and lookup operators, you are able to create your own custom list and run lookups against this list. In this lab, you will create a list of blacklisted IP addresses. This list is populated with IP addresses that Snort has identified with a Priority 1 alert.

1. Use the save operator to store the list of IP addresses in your custom list

```
_sourceCategory=labs/snort and "[Priority: 1]"
| parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| count src_ip
| src_ip as blacklisted
| fields - src_ip
| save append myfolder/<your_name>
```

2. To verify the creation of your list, run the following query:

```
cat myfolder/<your_name>
```

3. To achieve more realistic results in step 4, use the blacklist provided here:

```
cat shared/snort_alerts
```

4. Use the lookup operator to do a lookup against a custom file:

```
_sourceCategory=labs/snort "[Priority: 1]"
| parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| lookup blacklisted from shared/snort_alerts on blacklisted=src_ip
| where !isEmpty(blacklisted)
```

Lab 12 - Correlation using Transaction

1. The transaction operator allows you to analyze related sequences of messages based on a unique transaction identifier such as a SessionID or IP Address. Transaction uses the unique identifier you specify to group related messages together and arrange them based on states which you define. In this lab, use the transaction operator to identify source IPs in your web apache logs that correlate to IPs that Snort (network intrusion detection) has flagged as related to a Web Application Attack.

```
((_sourceCategory=labs/snort "[Classification: Web Application Attack]") or
_sourceCategory=Labs/Apache/Access)
| parse "{TCP} *:* -> *:*" as src_ip, src_port, dest_ip, dest_port nodrop
| parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| transaction on src_ip
  with states %"labs/snort", %"Labs/Apache/Access" in _sourceCategory
| where %"labs/snort">0 and %"Labs/Apache/Access">0
```

2. Now that you have identified these IPs, use the Threat Intel lookup to see if these are IOCs. Keep in mind that no results simply means that they are not flagged as malicious IP addresses in the CrowdStrike database.

```
((_sourceCategory=labs/snort "[Classification: Web Application Attack]") or
_sourceCategory=Labs/Apache/Access)
| parse "{TCP} *:* -> *:*" as src_ip, src_port, dest_ip, dest_port nodrop
| parse regex "(?<src_ip>\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})"
| transaction on src_ip
  with states %"labs/snort", %"Labs/Apache/Access" in _sourceCategory
| where %"labs/snort">0 and %"Labs/Apache/Access">0
| lookup type, actor, raw, threatlevel as malicious_confidence
  from sumo://threat/cs on threat=src_ip
| where !isEmpty(type)
```

Lab 13 - Correlation using Transactionize

1. Different from the transaction operator which aggregates results, the transactionize operator correlates and provides detail, allowing you to display more fields/detail in your results. In this lab, use the transactionize operator to identify source IP addresses for which you've had normal traffic (type="TRAFFIC" and action="allow), " but also received traffic that has been flagged as a THREAT.

```
((_sourceCategory=Labs/PaloAltoNetworks ",THREAT,") or
(_sourceCategory=Labs/PaloAltoNetworks ",TRAFFIC," action=allow))
| concat(dest_ip,":", dest_port) as destination
| transactionize src_ip (merge type, destination, src_ip takeFirst)
| where type matches "*TRAFFIC*" and type matches "*THREAT*"
// Optionally, you can use these last 2 lines to clean up your results
//| count src_ip, type, destination
//| fields - _count
```

Note that this same query can be used to correlate your Snort data (or any other threat detection data) to your network data by replacing the source categories in the top 2 lines.

Lab 14 - Correlation using Subqueries

Subqueries help you correlate events across different Sumo queries, by allowing one query to pass results back to another query to narrow down the set of messages that are searched in that query. In this lab, identify the web server traffic that has also been flagged as a Web Application Attack in your Snort data.

```
_sourceCategory=Labs/Apache/Access
[subquery:
_sourceCategory=labs/snort "[Classification: Web Application Attack]"
| parse "{TCP} *:* -> *:*" as src_ip, src_port, dest_ip, dest_port nodrop
| compose src_ip
]
| count src_ip, method, status_code, url
| sort _count
```

You have seen various ways to correlate your data. Here is a summary to help you know which operator to use:

Transaction	Transaction allows you to correlate messages (from a single source or multiple sources) based on one or more common keys (IP Addresses, Session ID's, etc). **It performs an "outer join" and produces an aggregate result**. Its main use case in the security space is to check the existence of a values across several data sources. For example, when various security tools alert on the same IP address.
Transactionize	Transactionize allows you to correlate messages (from a single source or multiple sources) based on one or more common keys (IP Addresses, Session ID's, etc). **It performs an "outer join", but operates on the raw messages.** Combined with merge, you can merge raw messages or different extracted fields across messages into a single row in the result set.
Subquery	Subquery lets you filter data from one result set based on the result of another query (or multiple queries), within one or across several datasets. It **performs an "inner join" and returns raw data.** Its main use case is to find data in one dataset that can be found in another dataset. For example, show all Windows Event Logs for hosts that have been flagged by and Endpoint protection system.

Visualizing your Data

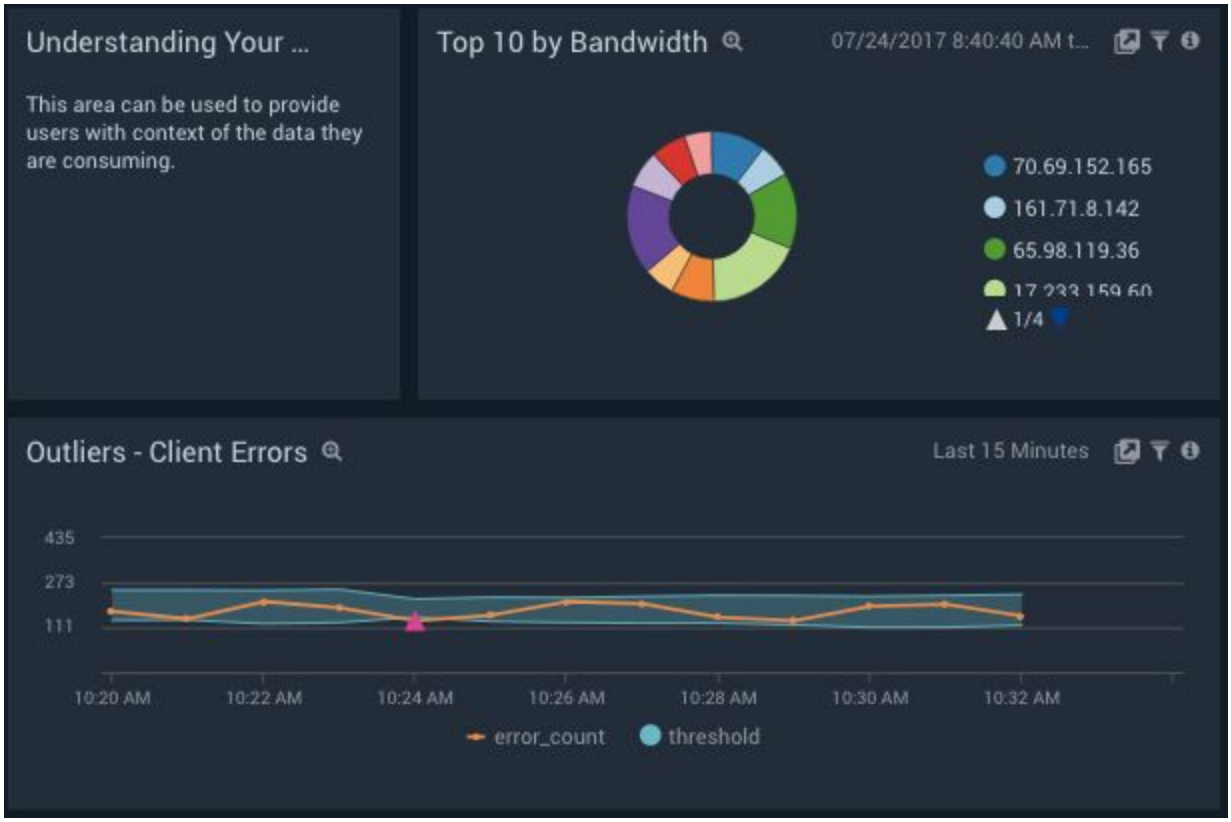
Lab 15 - Create and Publish a Dashboard

Dashboards allow you to group your queries and charts and share with others.

1. Re-run Lab 2 to obtain a pie chart.
2. In the chart area, click Add to Dashboard.
3. You can name your new Panel: *Top 10 IPs in US West*.
4. You can name your Interactive Dashboard: *<Your Name> Training Dashboard*
5. Once created, you are able to resize your Panel, toggle themes, add filters, or share your newly created Dashboard.
6. Note that by clicking the "Show in Search" button, you can go back to the query behind the chart, which allows you to make any changes needed to the query and re-save by clicking "Update Dashboard".

Lab 16 - Add Panels to your Dashboard

1. Re-run Lab 7 and create an outlier line graph.
2. Save this as a Panel to your existing dashboard.
3. Add a Text panel to your Dashboard to simulate the screenshot below.
4. Lastly, share your dashboard with other users in your organization.



Monitoring Critical Events

Lab 17 - Alerting on New Security Attacks

Using LogCompare, which allows you to compare log activity from two different time periods, alert on log messages which exist in the last 60 minutes, but did not exist for the same 60 minutes time period, but 24 hours ago.

1. Search your Snort data for the last 60 minutes. Click LogCompare to compare current signatures to signatures from 24 hours ago.
```
_sourceCategory=labs/snort
| logcompare timeshift -24h
```

2. To view results where there are new Signatures in the current time period that did not exist 24 hours ago, add a where clause for _isNew:
```
_sourceCategory=labs/snort
| logcompare timeshift -24h
| where (_isNew)
```

3. Using your email address, you can now create a Scheduled Search to Alert when this query has results.
   **** NOTE****
   If you did create a Scheduled Search, delete it after testing, otherwise, you will be emailed/alerted periodically.

Lab 18 - Create Alerts with Context

In this lab, rather than alerting on simple error counts with a static threshold, which can yield false positives (Fig. 1 below), learn to create an alert that will notify you when your errors increase at a higher rate than your overall traffic (Fig. 2). For a more in depth explanation, check out this blog post on creating meaningful alerts.

Fig. 1

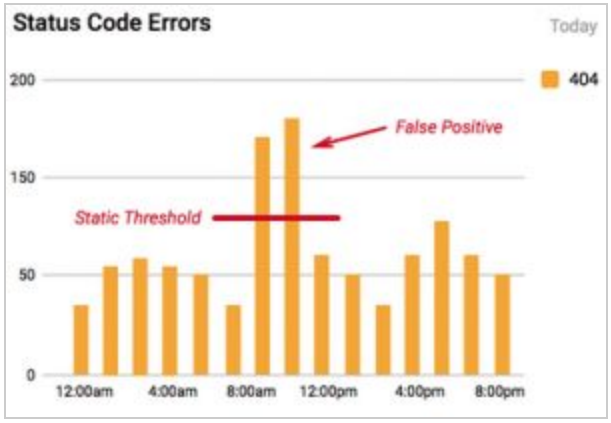
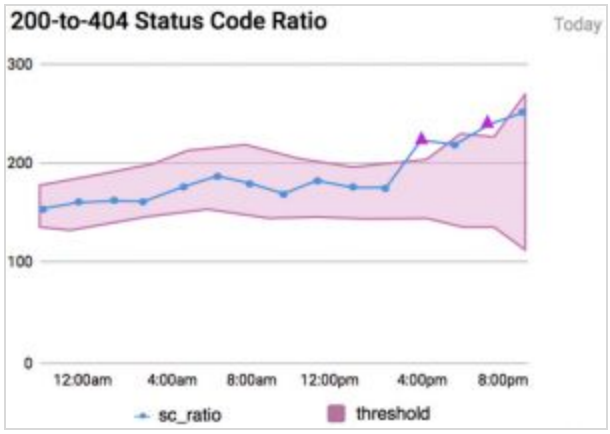


Fig. 2



1. Using Labs/Apache/Access data, search only for messages with status code 200 or 404. 404 messages are your errors, and 200 messages give you a sense of the overall traffic.
2. Count 200 messages as Successes and 404 messages as Fails.
3. Sum Successes and Fails to get a count by timeslice to identify a trend over time.
4. Create a ratio of fails to successes
5. Use **outlier** operator to identify anomalies in the ratio

```
_sourceCategory=Labs/Apache/Access (status_code=200 or status_code=404)
| timeslice 1m
| if (status_code="200", 1, 0) as successes
| if (status_code="404", 1, 0) as fails
| sum(successes) as success_cnt, sum(fails) as fail_cnt by _timeslice
| fail_cnt/success_cnt as failure_rate
| sort _timeslice desc
| outlier failure_rate window=5, threshold=3, consecutive=1, direction=+
```

6. Adding the following where clause allows you to filter out only outliers (where ration increase is higher than normal) . Using your email address, you can now create a Scheduled Search to Alert when this query has results.

```
| where failure_rate_violation > 0
```

**** NOTE****
If you did create a Scheduled Search, delete it after testing, otherwise, you will be emailed/alerted periodically.

Security Apps

Lab 19 - Installing the AWS CloudTrail App

Sumo Logic Apps allow you to take advantage of out-of-the-box-content, providing you with popular queries and Dashboards for common data Sources.

1. In the App Catalog, search for the AWS CloudTrail app
2. Once selected, click Add to Library
3. For _sourceCategory, select **Labs/AWS/CloudTrail**
4. As a best practice, create a new folder named Apps. Later on, you can share the entire folder and its content.
5. Add the AWS CloudTrail app to your new folder in your Library.
6. Now view your new Dashboards in your Personal/Apps/AWS CloudTrail folder. This new folder includes all pre-built queries and Dashboards for that source.
7. Review the new content, both queries and dashboards.
8. Notice that you can click on any Dashboard panel to view the query behind it. If you make changes to the query, you can always click Update Dashboard to save your changes to the original dashboard.
9. If you want to share this content with other users, from the Library, select the AWS CloudTrail folder. Clicking on the 3 stacked dots to the right opens a menu of actions, including Share.

Lab 20 - Installing the Threat Intel App for OSSEC data

The Threat Intel App allows you to correlate CrowdStrike threat intelligence data with your own log data, for real-time security analytics to detect threats. In particular, it scans for threats based on filename, URL, domain, Hash 256 and email.
Although you could install the Threat Intel app for your entire data set, this is not recommended for performance reasons. Best practice is to install the app for a given source. If necessary, you can install the app numerous times for the different sources you are looking to correlate. In this lab, we will install the Threat Intel app for our Ossec data (host-based intrusion detection system).

1. In the App Catalog, search for the Threat Intel Quick Analysis app
2. Once selected, click Add to Library
3. For App Name, enter: **Threat Intel - OSSEC**
4. For _sourceCategory, enter: **Labs/OSSEC**
5. Under Advanced, select Personal > Apps, and finally click Add to Library.
6. This created a new folder Personal/Apps/Threat Intel - OSSEC with queries and Dashboards for that source.
7. Open the **Threat Intel Quick Analysis - Overview** dashboard. Notice that most panels have no data to display.
8. Is that a bad thing? NO - No data means there are no threats correlated to your current OSSEC data. As you can see in the top right Panel, numerous records were scanned, but no threats were returned. In a future lab, we will walk you through an exercise that actually gets a hit on an IOC.
9. Notice that you can click on any Dashboard panel to view the query behind it. If you make changes to the query, you can always click Update Dashboard to save your changes to the original dashboard.
10. If you want to share this content with other users, from the Library, select the Threat Intel folder. Clicking on the 3 stacked dots to the right opens a menu of actions, including Share.