Splunk Report

Contents

[Introduction 2](#_Toc181106453)

[Sample of splunk dashboard for MFA policy 3](#_Toc181106454)

[Query 5](#_Toc181106455)

[Panel 2 Sources with the most activity without MFA 5](#_Toc181106456)

[Query 6](#_Toc181106457)

[Panel 3 total logs vs logs without MFA 6](#_Toc181106458)

[Sample dashboard of data protection policy 7](#_Toc181106459)

[Panel 1 brute force status 8](#_Toc181106460)

[Query 8](#_Toc181106461)

[Panel 2 highest request of an IP and the frequency. 9](#_Toc181106462)

[Queries 9](#_Toc181106463)

[Panel 3 Error code frequency chart and IP details table 10](#_Toc181106464)

[Queries 11](#_Toc181106465)

[Summary 12](#_Toc181106466)

[References 13](#_Toc181106467)

## Introduction

Sutrefia (Sunken Treasure Financials Australia) are a medium to large scale business that run financial services to company’s with a wide online presence that use the online Prescence to inform and distribute their services. Any business with such an online Prescence would be a joyful target to cyber attacks because of the financial data and its value the data itself holds. As a response of improvement in their cyber security efforts the company have decided to use SEIM tool “splunk” to monitor and chart information that can makes it easier for interpretation for non-technical users like management or auditors.

After an incident the company have pushed for me to demonstrate the use of splunk effectiveness to improve their IDS and IPS. Using the BOTSv3 dataset I have made a dashboard to demonstrate as requested based off of the security **policy of access control and multi factor authentication** where all systems should require MFA to access them with an additional feature in **public access and data protection**, in specific the number of access limited to control data.

MFA monitoring can point several security concerns where event directly violate the policy. For the demonstration I have used AWS’s “cloudtrail” application in which its purpose is to monitor API communications with relations to a specific account (AWS, n.d.).

I have chosen a limited amount of fields to help build an indication to what the above mentioned problems could be. For MFA I have chosen the :

1. **“userIdentity.sessionContext.attributes.mfaAuthenticated”**

This field that I was able to find as an answer for the boss of the soc(bots) competition by splunk answers provided by medium (medium, 2024) only would equate to true or false meaning in whether if the mfa authentication is activated or not ‘true’ would mean that the user is using mfa and “false” would mean the user isn’t using mfa.

Next would be the source type that refers directly to the application in which the log has been written for in this case it would be. This field was chosen through a simple query to check the amount of logs per app using the following query  
  
and I went through some keywords and logs to find errors that would be relevant to the policy I have chosen and came across the following with 6571 logs with fields that I was able to find relevant

index="botsv3"

| stats count by sourcetype

| sort - count

| head 20

| table sourcetype, count

1. **"aws:cloudtrail"**

index="botsv3"

| stats count by sourcetype

| sort - count

| head 20

| table sourcetype, count

For the minor detail for public access and data control I chose a field to show the error message given by application to the user named   
  
**3. “errorCode” and “errorMessage”**  
  
I have only chosen one particular error message that mentions that the user is trying to access the page too many times for which the api would respond with “Client.InstanceLimitExceeded” as the error code and “You have requested more instances (1) than your current instance limit of 0 allows for the specified instance type. Please visit http://aws.amazon.com/contact-us/ec2-request to request an adjustment to this limit.” as the error message it is self explanatory what the error message is referring to but to help better understand we can think of instance as the amount of sessions the user is trying to invoke.

## A screenshot of a graph Description automatically generatedSample of splunk dashboard for MFA policy

## 

With the use of the sample dataset I have created a dashboard that showcases real time information relating to the MFA policies with this dashboard non technical users wouldn’t have to wait for experts to inform them of the overview of the compliance of the MFA polices. Here I have used the application as cloudtrail for branding image of where the data is mainly derived from. This dashboard highlights the IP with the highest usage without MFA having it referrable and visible whenever needed making it easy point out any concerning users, if there is a 100% compliance then the IP wouldn’t show. We also have a more detail dense tile containing a pie chart of the count of the sources without MFA which would be ideal for support teams or any other teams to refer to when they need to know who to reach out to and which sources are the concern and the bottom tile is where we have comparison of the total number of activity with the number of logs where MFA isn’t activated. This makes a more efficient way of showing the compliance rate the users in their network. The dashboard is constructed to demonstrate that any sort of information that we require with the needed depth can be added into a simple viewing arrangement which makes decision making, planning, data gathering and management more accessible without the need for certified personal even more so since the data is updated in real time lifting the more cumbersome job off any cyber employees. This dashboard can also aid with post disaster information if in any case where there was a social engineering attack where we can be alerted as when a person without MFA credential attempts to get into the system.  
  
  
Panel 1 IP with the most activity without MFA

A number with numbers on it

Description automatically generated

The function of the panel is have an updated one glance view of the IP who has the most activity in the company network without a MFA. The panel would update once move onto the next most active IP without MFA the view of the panel is to create urgency and importance on where to draw most attention without crowding it with too much information which can be unpleasant and confusing for non technical viewers. This shows that splunk can be customized to fit most requirements.

### Query

index="botsv3" sourcetype="aws:cloudtrail" userIdentity.sessionContext.attributes.mfaAuthenticated="false"

| stats count as frequency by sourceIPAddress

| table sourceIPAddress

| sort - frequency

| head 1

The query starts by searching for events where MFA was not authenticated, indicated by the mfaAuthenticated="false" field. It then aggregates these events by counting the number of occurrences for each sourceIPAddress using the stats command. The results are sorted in descending order based on the frequency of occurrences, so the IP address with the highest count of MFA failures appears first. Finally, the query displays a table with the top result, showing the IP address and its corresponding frequency of MFA failures.

## Panel 2 Sources with the most activity without MFA

A screenshot of a graph

Description automatically generated

The purpose of this panel is to show more details to a more broader group of data while maintaining viewing convenience this panels is meant to couple with the first panel to show the group of Ips that follow behind that are actively using the network without MFA this can help internal teams to delegate work. This also emphasizes the severity which in this case how much more of a security risk this ip is and we can use this as a reference point in case there are any suspicions of unauthorized users grabbing company data since it only has the most obvious option of the users who don’t have MFA which in an ideal DRP would be the first people you would suspect.

### Query

index="botsv3" sourcetype="aws:cloudtrail" userIdentity.sessionContext.attributes.mfaAuthenticated="false"

| stats count as frequency by sourceIPAddress

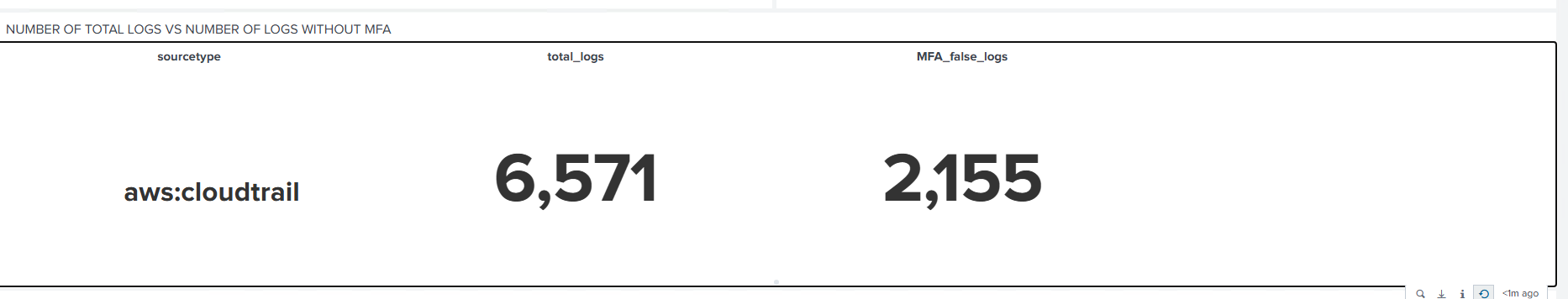
| table sourceIPAddress, frequency

| sort - frequency

| head 10

This query provides a table of the 10 IP addresses with the highest number of failed MFA authentication events, sorted by frequency. This is useful for identifying potential security risks related to failed authentication attempts from specific IP addresses.

## Panel 3 total logs vs logs without MFA



This panel is has 3 main heads of information. First is the source type meaning the application in which we get the logs from in this case it is cloudtrail from aws the purpose is to remind any interested partys who don’t use this dashboard a lot like the executives or higher management to remind and of which application is sending us this information and since in this case it an aws service we can prove its worth by putting it onto the dashboard as a result of the logs it sends and if in any case the application is changed or the dashboard is used with other applications or sources it will be displayed so that the viewers know data is based from. The next is the total logs this is a number logs that have happened over the course of the deployment of the application this particular number can show company milestones or the current activity which can be also to decide whether we would need upgrade any hardware if need the final panel is the total count of users without MFA this is a crucial piece of information for the company operations to remind them of the progress they would be making which would concern the users compliance for enable MFA this would not only keep the relavant parties that are working on the compliance management if they have any more users to reach out to or even a reference of a standard they would need to maintain to keep the system up and running without breaking policies to a sever extend

#### Query

index="botsv3" sourcetype="aws:cloudtrail"

| stats count as total\_logs by sourcetype

| append [

search index="botsv3" sourcetype="aws:cloudtrail" userIdentity.sessionContext.attributes.mfaAuthenticated="false"

| stats count as MFA\_false\_logs by sourcetype

]

| stats sum(total\_logs) as total\_logs, sum(MFA\_false\_logs) as MFA\_false\_logs by sourcetype

| table sourcetype, total\_logs, MFA\_false\_logs

The query starts by counting all CloudTrail logs and stores this count as total\_logs. It then appends a subsearch that filters for logs where MFA was not authenticated (mfaAuthenticated="false") and stores this count as MFA\_false\_logs. Afterward, it aggregates the results from both searches, summing up the total number of logs and the number of MFA failures per sourcetype. Finally, it displays a table showing the sourcetype, total number of logs, and the number of MFA failures (MFA\_false\_logs), providing a comparison between overall CloudTrail activity and MFA failures.

## Sample dashboard of data protection policy

A screenshot of a white screen

Description automatically generated

This dashboard is meant to represent another policy of which is data protection policy. This particular dashboard is meant to showcase the number of instance requested by an IP meaning the how often a user would request to visit the page it normally usual for users to request around 5 to 10 times how if in which users request more than 20 times it would raise suspicion because the practical need to request so frequently. The possible reason for frequent request is that the user could possibly trying to force any sort of malware or a form an DOS attack where users would overload the sever with requests forcing it to crash. This is just few of the many possibilities for reasons. The dashboard is built in a manner where it would show an alert in case of an abnormal activities in the network or even alert or warn hosts of a rise in malicious activity before the attack has taken place. This particular dashboard shows more functionality and more information showing how much control we have with splunk terms of information density and comfortability.

### Panel 1 brute force status



This panel boasts the ability of splunk to not only show data but also alert when a specified trigger is met. In this case we can see the prompt showing there is a brute force attempt off the trigger which will shown later in the report. This could immensely useful in time of IRP where it would give non only technicians a HeadStart but also a heads up for non technicians of the company.

### Query

index="botsv3" sourcetype="aws:cloudtrail" errorCode=\*

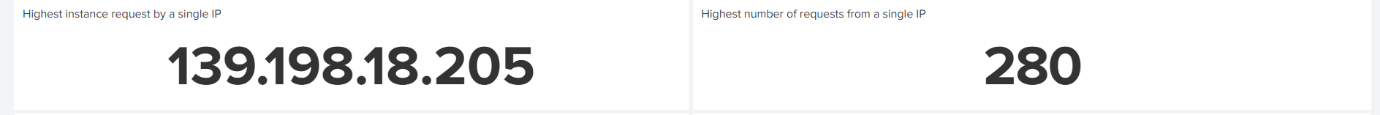
| stats count as error\_count by errorCode

| eval message=if(error\_count > 15, "Brute force attempts detected: " . frequency, "BRUTE FORCE ALERT")

| table message, error\_count

query analyzes AWS CloudTrail logs to identify potential brute force attempts based on error codes. It first filters logs with any errorCode and counts the occurrences of each error, storing this count as error\_count. Then, using the eval function, it checks if the error\_count exceeds 15 which can be tailored to the company preference this feature could also save company resources and show where technicians could close down certain request or daily maintaince any networking, and if so, it generates a message indicating "Brute force attempts detected" along with the error count. If the error count does not exceed 15, it assigns a default message, "BRUTE FORCE ALERT". finally, the query presents a table with the message and error\_count, highlighting possible brute force activity based on error frequency.query presents a table with the message and error\_count, highlighting possible brute force activity based on error frequency.

## Panel 2 highest request of an IP and the frequency.



These 2 panels show the culprit IP that needs to be concerned about and shows how much work he has already done and maybe the capabilities. The object of this panel is to show in times either the rising culprit or the current culprit of a possible DOS attack for instance. This panel can be seen easily and people can know if in case the IP does not change for a certain period of time or if the frequency keep increasing too fast it would give an to the host to mitigate the issue and block the IP from the sever making it convenient as well as audible for non technical personnel.

### Queries

index="botsv3" sourcetype="aws:cloudtrail" errorCode="Client.InstanceLimitExceeded"

| stats count as frequency by sourceIPAddress

| table sourceIPAddress, frequency

| sort - frequency

| head 1

query is designed to identify the source IP address responsible for the highest number of occurrences of the Client.InstanceLimitExceeded error in AWS CloudTrail logs. It filters the logs from the botsv3 index and the aws:cloudtrail sourcetype, specifically where the error code is Client.InstanceLimitExceeded. The query then counts the number of occurrences (frequency) for each sourceIPAddress, displaying the results in a table with the IP address and its corresponding frequency. The results are sorted in descending order by frequency, and the query returns only the top IP address with the most occurrences of the error.

index="botsv3" sourcetype="aws:cloudtrail" errorCode="Client.InstanceLimitExceeded"

| stats count as frequency by sourceIPAddress

| table frequency

| sort - frequency

| head 1

query is designed to find the highest occurrence of the Client.InstanceLimitExceeded error in AWS CloudTrail logs, but it focuses only on the frequency of occurrences, not the specific source IP addresses. It filters the logs from the botsv3 index with the aws:cloudtrail sourcetype, specifically looking for events where the error code is Client.InstanceLimitExceeded. The query then counts the number of times this error appears and stores the count as frequency. It sorts the results in descending order by frequency and returns only the top result, showing the highest occurrence of this error. However, it omits the sourceIPAddress, focusing purely on the number of error occurrences.

Panel 3 Error code frequency chart and IP details table  
  
A screenshot of a computer

Description automatically generated

Here we have a demonstration of how much of information we can saturate using splunk making it not only useful for non technicians but technicians as well on the left we have a bar chart with the errorcode and the frequency shown in real time this can not only provide information on malicious attacks but also point out any bugs or other problems in the system and network giving it great value for the space it takes this chart can ignite meetings for functional purposes this also can point to any other attacks besides DOS attacks like:

* **Volume-based attacks** (e.g., UDP floods, ICMP floods): These attacks flood the network with massive amounts of traffic to saturate the bandwidth.
* **Protocol attacks** (e.g., SYN flood, Ping of Death): These exploit weaknesses in network protocols to overwhelm the system.
* a**pplication layer attacks** (e.g., HTTP GET/POST flood): These target specific applications, overwhelming them with requests.

With more information and testing the splunk dashboard can update the y and x axis to show specialised information to inform a specific attack. On the right hand side we have the table which can more information once again showing splunks versatility to tailor information as we need. In the table we have deeper information such as the account id, region, the event version and the ip address to tie them into relevance this table is more used for any technicians if they would have look up certain details when they are either defending against an attack or if they catch an eventversion that would suggest lower security standards or even simply to use as point of reference compared to having it on a piece of paper where the information would manually have to updated or busy computer which would be too disruptive to look upon for the details needed.

### Queries

index="botsv3" sourcetype="aws:cloudtrail" errorCode=\*

| stats count as error\_count by errorCode

| table errorCode, error\_count

| sort - error\_count

| head 10

query analyzes AWS CloudTrail logs to identify the most frequent error codes. It starts by searching for any events where an errorCode is present, using the errorCode=\* filter. The query then counts the number of occurrences of each unique errorCode and stores this count as error\_count. The results are organized in a table that displays the errorCode alongside the corresponding error\_count. The query sorts the results in descending order based on the error count and returns the top 10 most frequent error codes. This helps in identifying the most common errors within the CloudTrail logs, potentially highlighting recurring issues

index="botsv3" sourcetype="aws:cloudtrail" errorCode="Client.InstanceLimitExceeded"

| stats count as frequency by sourceIPAddress, awsRegion, userIdentity.userName, userIdentity.accountId, eventVersion

| sort - frequency

| head 5

| table sourceIPAddress, frequency, awsRegion, userIdentity.userName, userIdentity.accountId, eventVersion

query is designed to analyze AWS CloudTrail logs specifically for events where the error code is Client.InstanceLimitExceeded, which indicates that AWS instance limits have been reached. The query counts the number of occurrences (frequency) of this error and groups the results by sourceIPAddress, AWS region (awsRegion), the username and account ID associated with the event (userIdentity.userName and userIdentity.accountId), and the eventVersion of the log. It then sorts the results in descending order by the frequency of occurrences and displays the top 5 results in a table

## Summary

The dashboard for Sutrefia (Sunken Treasure Financials Australia) is a crucial tool for managing security policy compliance, particularly in the context of the financial services industry, which is highly susceptible to cyber threats. Given the sensitive nature of the financial data handled by Sutrefia The dashboard can include panels that track and visualize MFA authentication status across the organization For example, the query examining AWS CloudTrail logs for mfaAuthenticated="false" allows the security team to identify instances where MFA is not being utilized. This is critical, as MFA adds an essential layer of security, making it more difficult for unauthorized users to gain access to sensitive systems. Another essential aspect of the dashboard is to monitor and enforce data protection policies through access controls. The use of queries that identify the frequency of specific error codes, such as Client.InstanceLimitExceeded, can help track and manage access limitations. High frequencies of these errors might indicate improper access attempts or over-utilization of resources, prompting a review of user access levels. By understanding which users, IP addresses, and regions are frequently encountering access issues, Sutrefia can refine its access controls and ensure that only authorized personnel have access to sensitive data

**Overall Benefits**

1. **Enhanced Visibility**: The combination of these panels provides security teams with real-time visibility into compliance with security policies, enabling them to identify potential vulnerabilities quickly.
2. **Proactive Risk Management**: By monitoring for MFA compliance and access violations, Sutrefia can proactively address security gaps before they lead to incidents.
3. **Data-Driven Decisions**: The analytics generated from the dashboard empower Sutrefia to make informed decisions about security policy adjustments and resource allocation based on actual usage patterns and compliance status.
4. **Regulatory Compliance**: Given the nature of financial services, the dashboard will help Sutrefia comply with industry regulations regarding data protection and access controls, thus avoiding potential penalties and reputational damage.

# References

AWS. (n.d.). *AWS CloudTrail FAQs*. Retrieved from aws.amazon: https://aws.amazon.com/cloudtrail/faqs/#:~:text=CloudTrail%20enables%20auditing%2C%20security%20monitoring,%2C%20analysis%2C%20and%20remediation%20actions.

medium. (2024, 2 18). *Splunk 3*. Retrieved from medium.com: https://medium.com/@enyel.salas84/splunk-3-a9bebd668771