FACULTY OF SCIENCE, ENGINEERING AND COMPUTING

School of Computer Science and Mathematics

BSc (Hons) DEGREE
IN
Cyber Security and Computer Forensics

Sahib Ghataura

K2015104

CI6280 SOC Case Study Coursework
Splunk Log File Analysis and Visualisation

Kingston University London

Table of Contents

Introduction

This project aims to provide SoftAI International with a backup plan in case they fall victim to a cyber security attack. As a Cyber Security consultant, this will be approached by implementing a log file analysis solution, which refers to a software tool that can examine log files generated from IT devices/applications.

This works by collecting log data and presenting in a graph/table format. A log could be any form, mostly a text file, and is created by applications or the operating systems logs. It will contain information that is a timeline of events executed concerning the application/operating system. Most information about the status of an IT device can be contained within log files. This helps users who need to become more familiar with raw data visualise and understand it better. Going with this approach will allow the company to monitor all the critical log files that are made on the company's network. The log analysis should include all the essential information about the event, such as connection attempts, the time, location, and destination. This will create a sense of reassurance as they will have an effective strategy to combat unauthorised access. (Watts, 2022)

Logs have significant impact, and if a user wishes to examine them for errors or system failures, they can go through several logs to find faults when troubleshooting. The easiest method to cope with this scenario is to have a systematic technique to handle these logs. All the relevant logs are gathered and kept in one place. This method is known as centralised log management. With this solution, analysis and mistake detection are made simple. (Sharif, 2022)

Technology Comparison

	Benefits	Limitations
Splunk	• Can handle large quantities of data, making it suitable for various types of consumers. • The customisable dashboards allows users to visualise and display their data so it's easy to focus on. • The search and analysation tools allow users to identify and acknowledge the problem quickly.	• The cost is decided by the amount of data that is consumed daily, some may find Splunk to be expensive. • Steep learning curve, as it will take time to learn as it requires an understanding. • Intense on computer resources due to the variety of tools when making a query and analysing logs.
Graylog	• Allows users to collect all the data the user needs from other sources, such as applications, network devices, or servers. • Offers clear data visualisation through customisable dashboards like graphs and charts. • Compresses archives and gets rid of unnecessary log files, making sure that there is not any space being taken up.	• Limited support for windows OS, official source and support is mainly concentrated at Linux-based installations. Can have issues on windows. • Limited amount of visualisation options available, so less customisable than some other tools available. • Focuses a lot more on log file management feature so may lack in analysis or visualization features
Elastic Stack	• Can handle data and can increase depending on how much the user wants. This means it can manage to change data for real-time analysis. • Architecture and extension/plug-in environment allow users to find specific needs for their tool's analysis/functionality area. • Can be set up to run automatically with a task scheduled and set up by the user.	• However, patches will need to be regularly installed for good security • Has many features which may be challenging to learn. The user may need to look at guides to start. • Hardware and resources taxing on computer; this could lead to an increase in cost. • Requires manual configuration of all the policies; this can bring about many errors and be time-consuming.

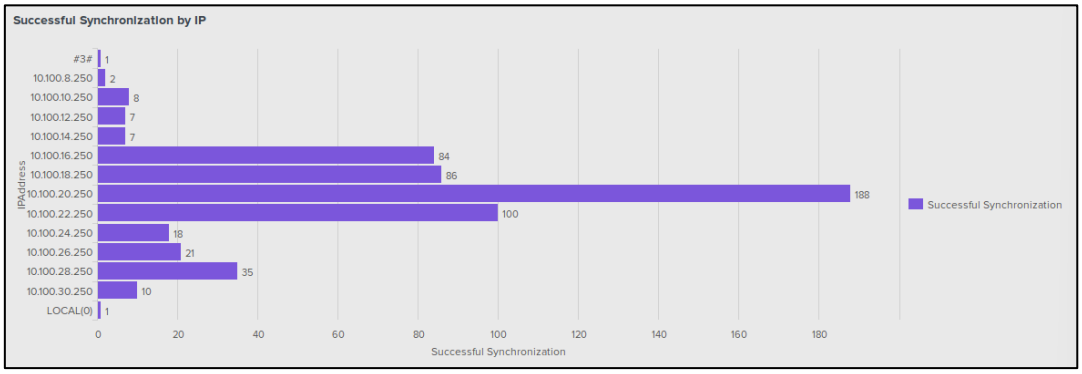
Visualizations

Thunderbird Log File

Successful Synchronizations by IP

(Refer to Appendix, 1)

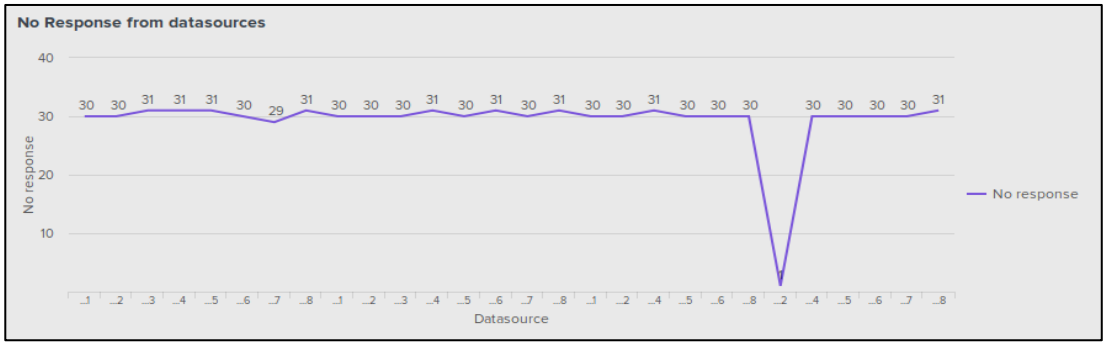
This shows the synchronizations that were successful for each IP address. In the command you can see the extracted field is synchronizedIP to filter out every non-synchronized IP. Chart count is used to tabulate all the IP addresses.



No response from datasources

(Refer to Appendix, 2)

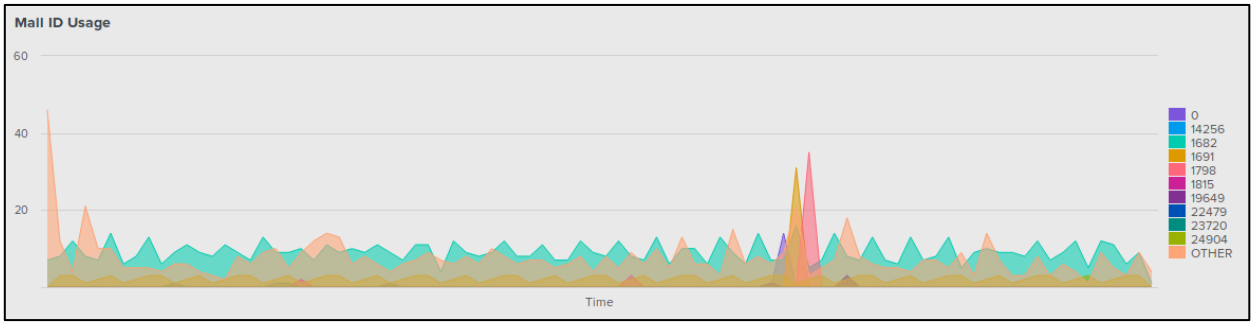
This shows each datasource and how many “No responses” they received. The extracted field No Response filters out any responses. Stats count is used to show the information in a numerical form and not in a graph before changing it to a visualisation.



Usage of all Mail_ID

(Refer to Appendix, 3)

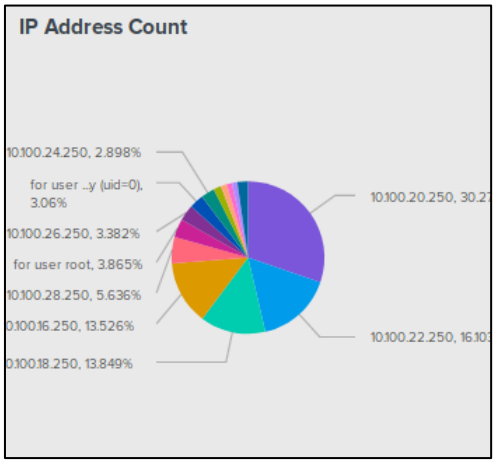
This shows each Mail ID being used throughout the entire log file. The extracted field Mail_ID is created and used in the command to only pick up Mail_ID’s. Timechart count creates a visualization for the number of events for each Mail_ID over time.



IP Address count

(Refer to Appendix, 4)

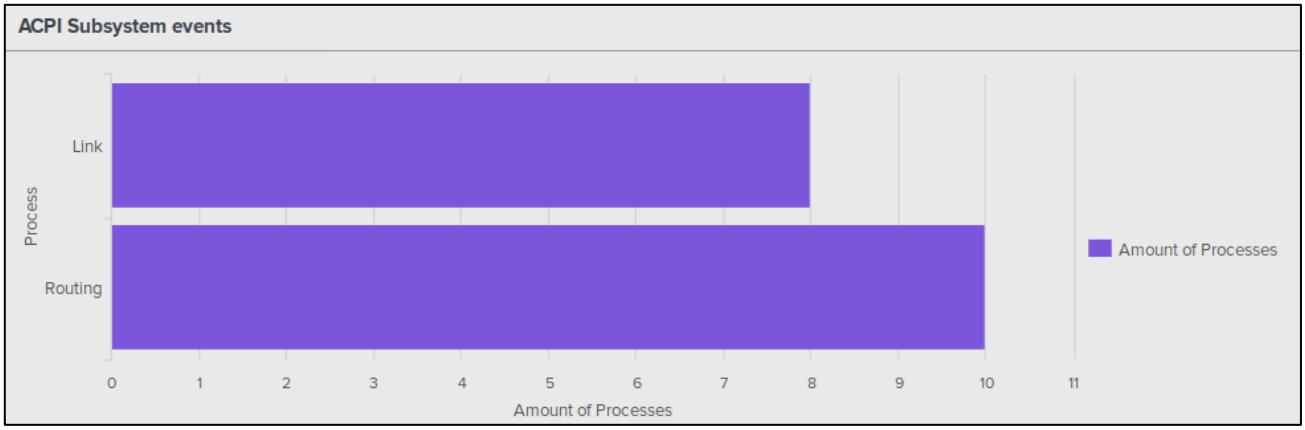
This shows how much each IP address has been used throughout this entire log file. Top limit = 20 means that it will show the top 20 used IP address based on usage.



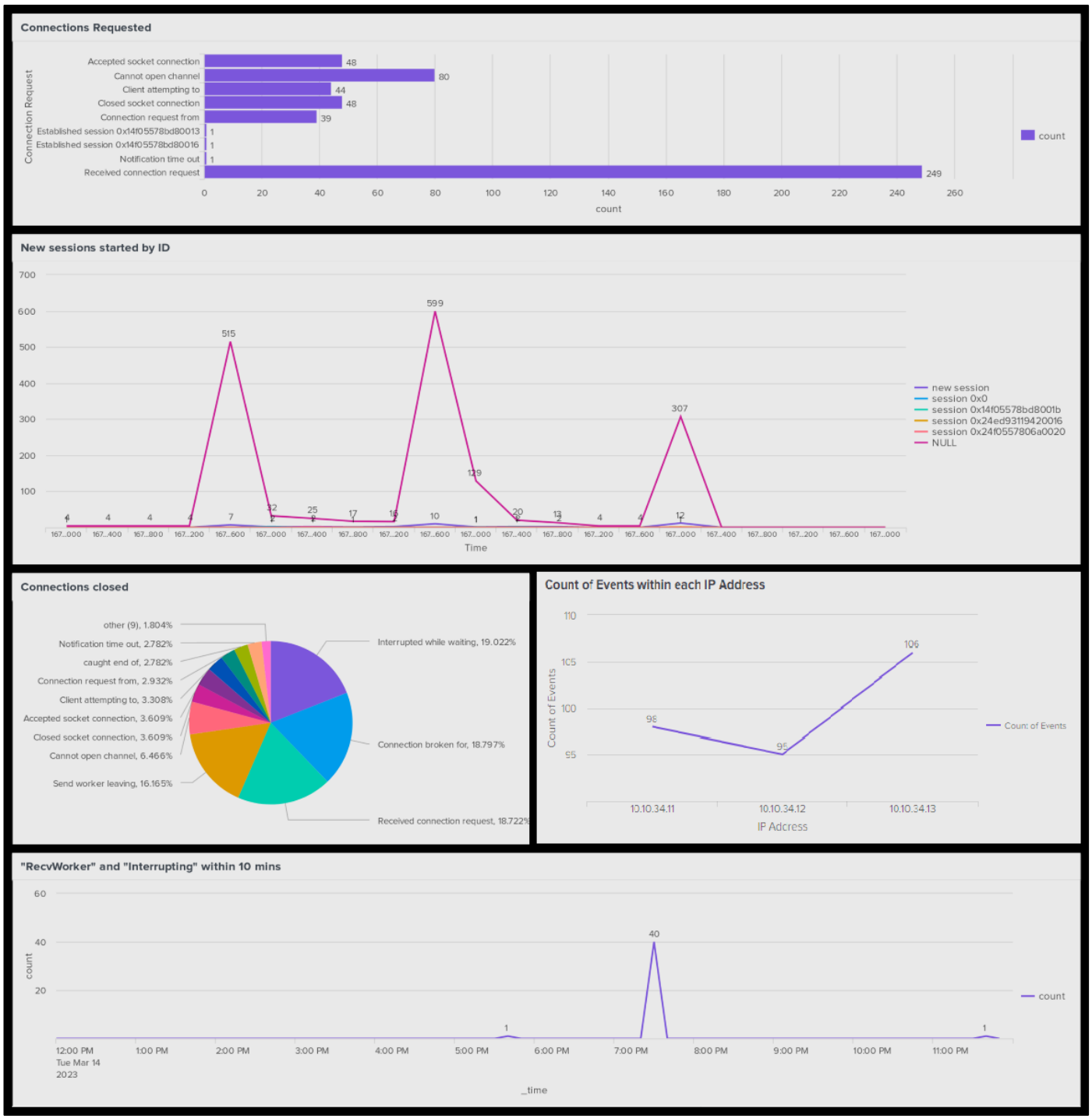
ACPI Subsystem events

(Refer to Appendix, 5)

This searches for the string “PCI interrupt” and then uses regular expression to search all the processes that are related to interrupts. The rex command extracts process names related to the interrupts and extracts them under the field “Process”. Chart count displays all the events for the “Process” field.



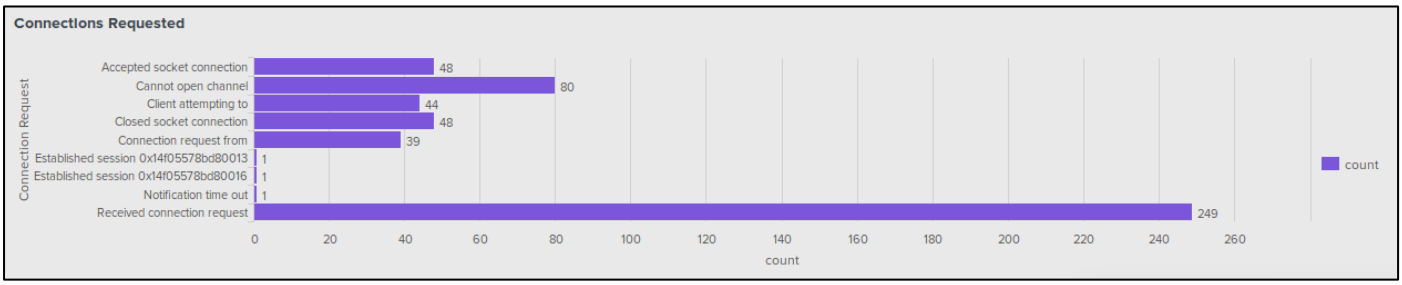
Zookeeper Log File



Connections Requested

(Refer to Appendix, 6)

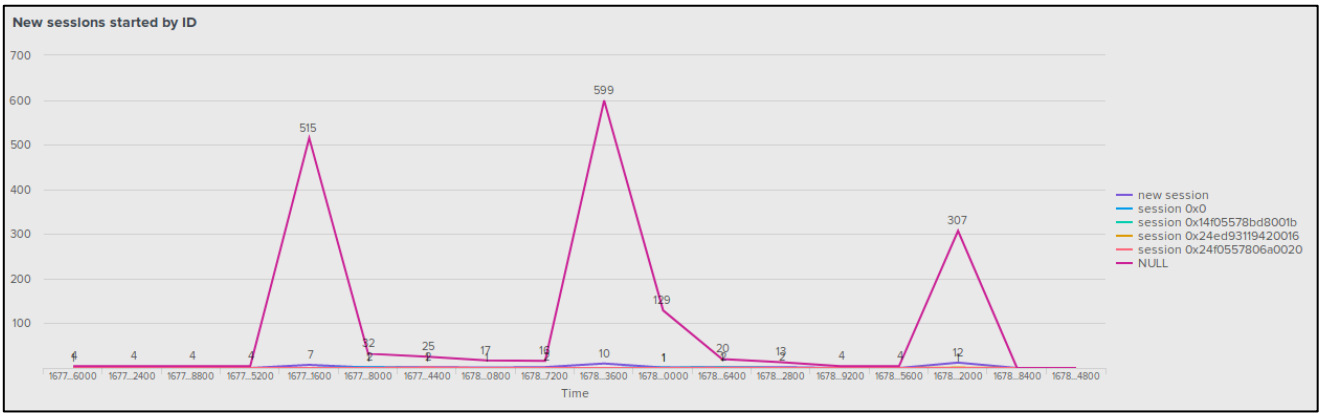
This shows all of the connections requested in this log file and all the different types of events that can occur when a connection is requested. Its all of the IP addresses within the log file. Stats count by Connection_Request means it is only going to show all of the IP address that requested a connection as stats.



New sessions started by ID

(Refer to Appendix, 7)

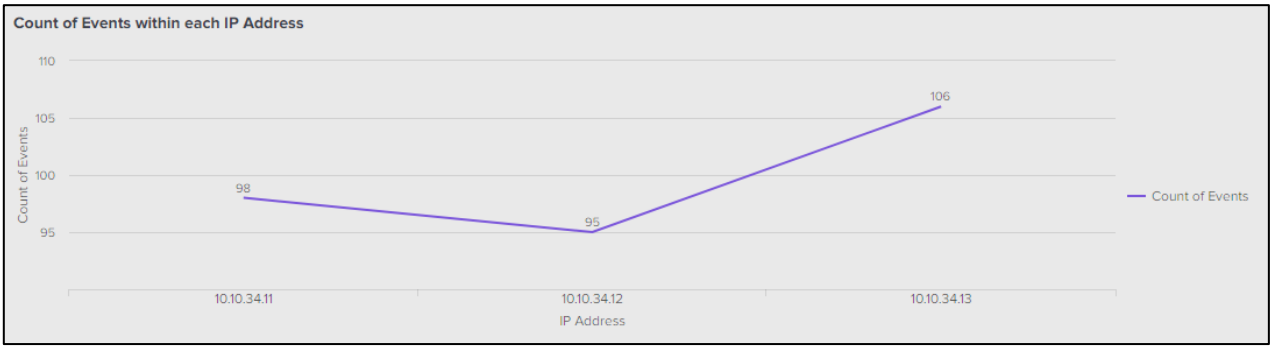
This shows all of the IDs of users and the new sessions started by them. Timechart count by New_Session limit 10 shows a time chart of how active the top 10 ID's were, however in the log file there were only 4 different ID's.



Count of Events within each IP Address

(Refer to Appendix, 8)

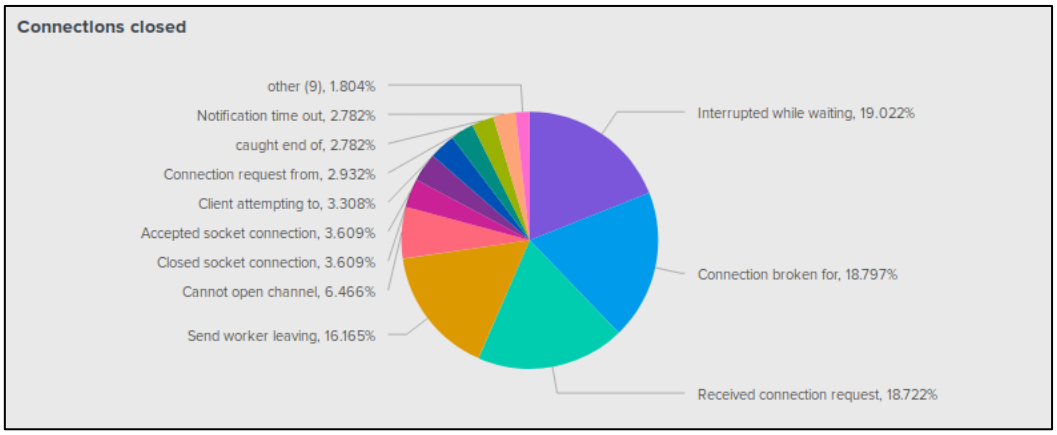
This shows the amount of events that occurred with each IP address. This regular expression command searches all events and IP addresses and puts it all into a new extracted field called ip_address. Stats count by ip_address means it is going to each IP address to see how much each one appeared in the log file.



Connections Closed

(Refer to Appendix, 9)

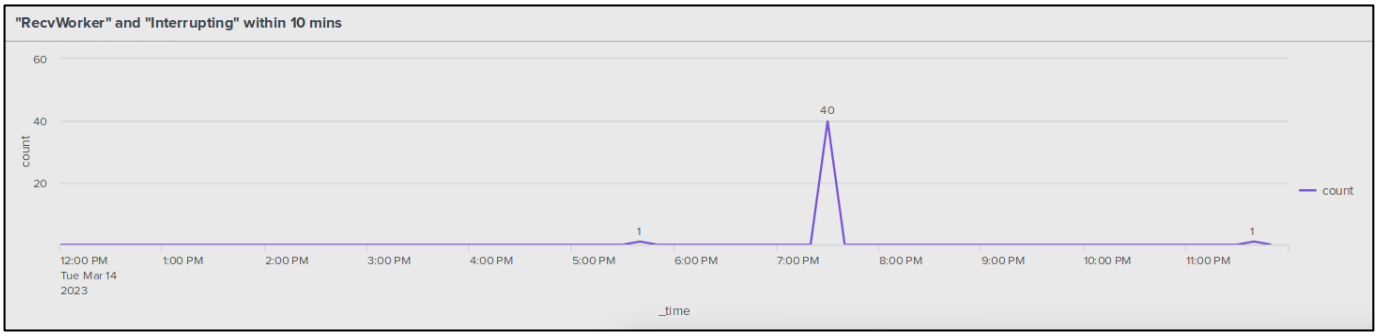
This shows all of the closed connections within the log file and the reason for them closing. Top limit=20 Closed Connection shows the top 20 types of connections being closed.

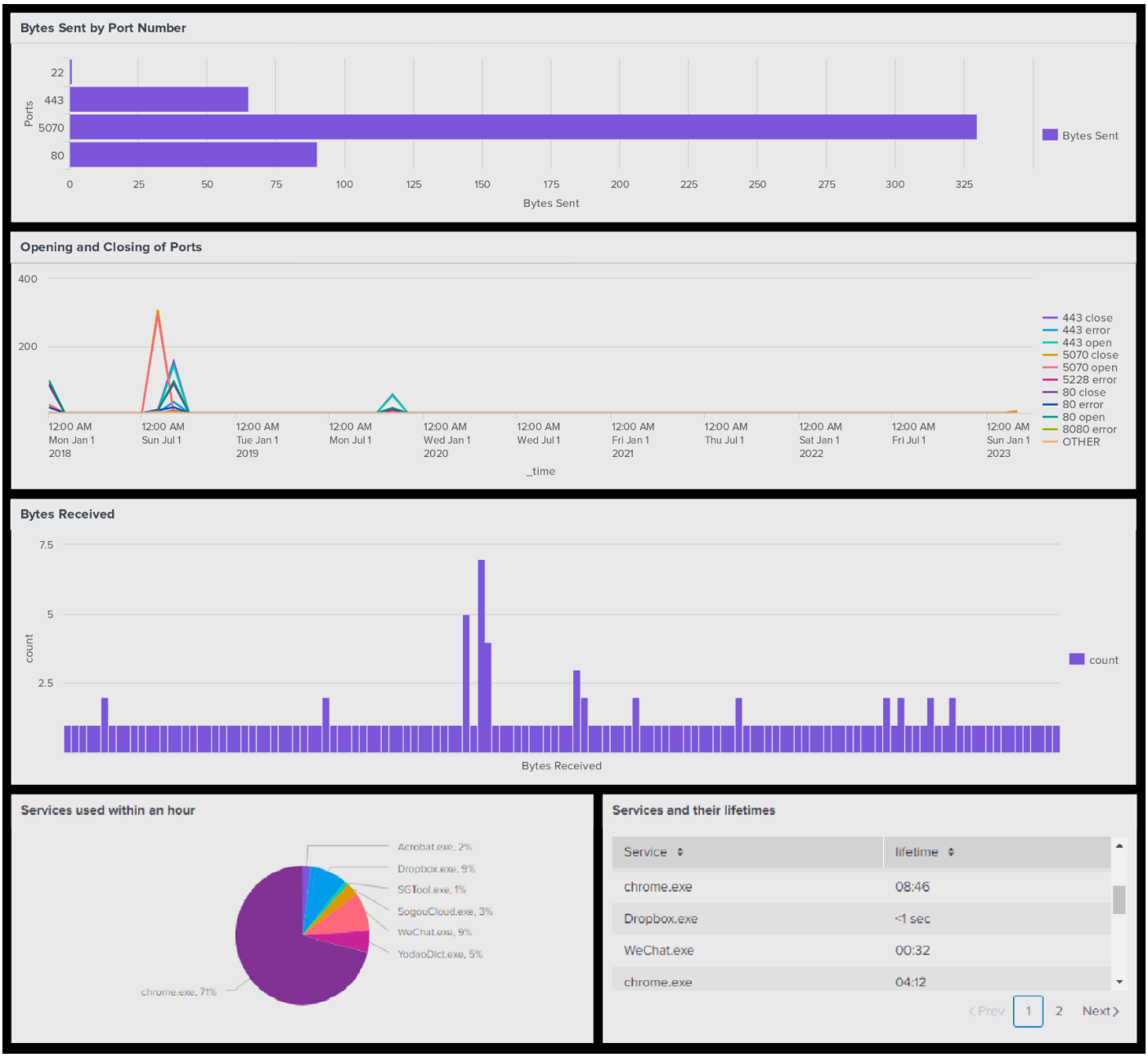


"RecvWorker" and "Interrupting" within 12 hours

(Refer to Appendix, 10)

This shows how many times the strings "RecvWorker" and "Interrupting" were present within set period of time. "RecvWorker" and "Interrupting" are in the search to filter out any logs that don't have them and earliest and latest show the time period that should be scanned. Timechart count span=12hr means that the period of time searched should be 12 hours.

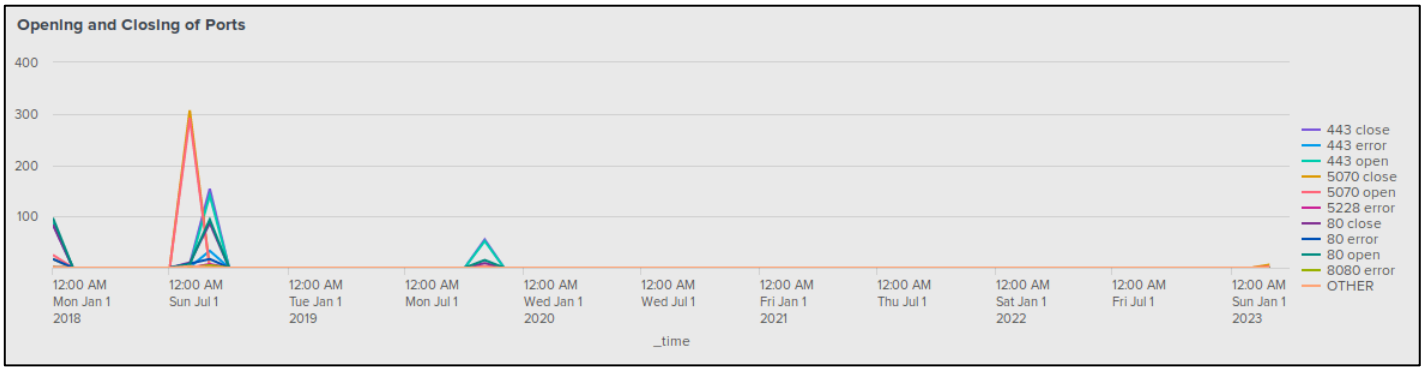



Bytes Sent by Port Number
Ports
22
443
5070
80
0
25
50
75
100
125
150
175
200
225
250
275
300
325
Bytes Sent
Bytes Sent
Opening and Closing of Ports
400
200
12:00 AM Mon Jan 1 2018
12:00 AM Sun Jul 1
12:00 AM Tue Jan 1 2019
12:00 AM Mon Jul 1
12:00 AM Wed Jan 1 2020
12:00 AM Wed Jul 1
12:00 AM Fri Jan 1 2021
12:00 AM Thu Jul 1
12:00 AM Sat Jan 1 2022
12:00 AM Fri Jul 1
12:00 AM Sun Jan 1 2023
_time
443 close
443 error
443 open
5070 close
5070 open
5228 error
80 close
80 error
80 open
8080 error
OTHER
Bytes Received
7.5
5
2.5
count
Bytes Received
count
Services used within an hour
Acrobat.exe, 2%
Dropbox.exe, 9%
SGTool.exe, 1%
SogouCloud.exe, 3%
WeChat.exe, 9%
YodaoDict.exe, 5%
chrome.exe, 71%
Services and their lifetimes
Service
lifetime
chrome.exe
08:46
Dropbox.exe
<1 sec
WeChat.exe
00:32
chrome.exe
04:12
Prev
1
2
Next

Opening and Closing of Ports

(Refer to Appendix, 11)

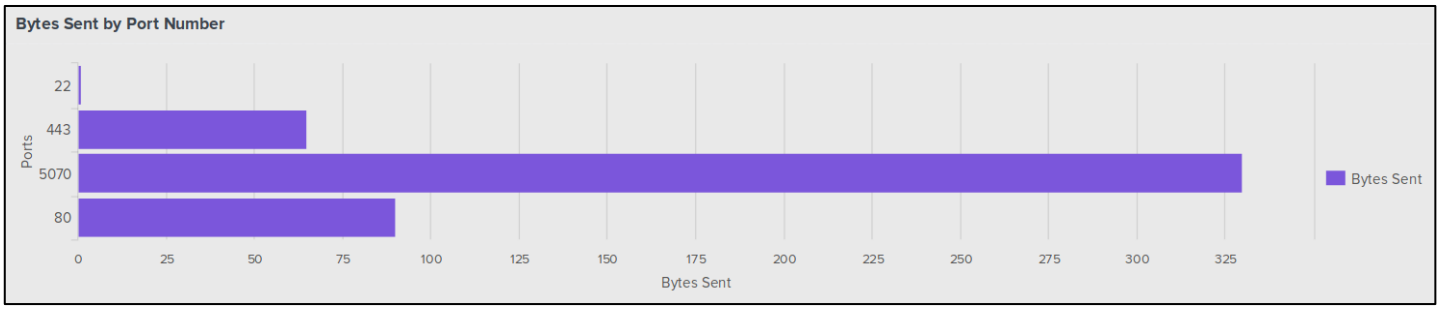
This shows the amount of times the top 10 ports were opened and closed. Timechart count by Open_Close limit=10 shows all the ports opening and closing within the 3 years of the log existing.



Bytes sent by Port Number

(Refer to Appendix, 12)

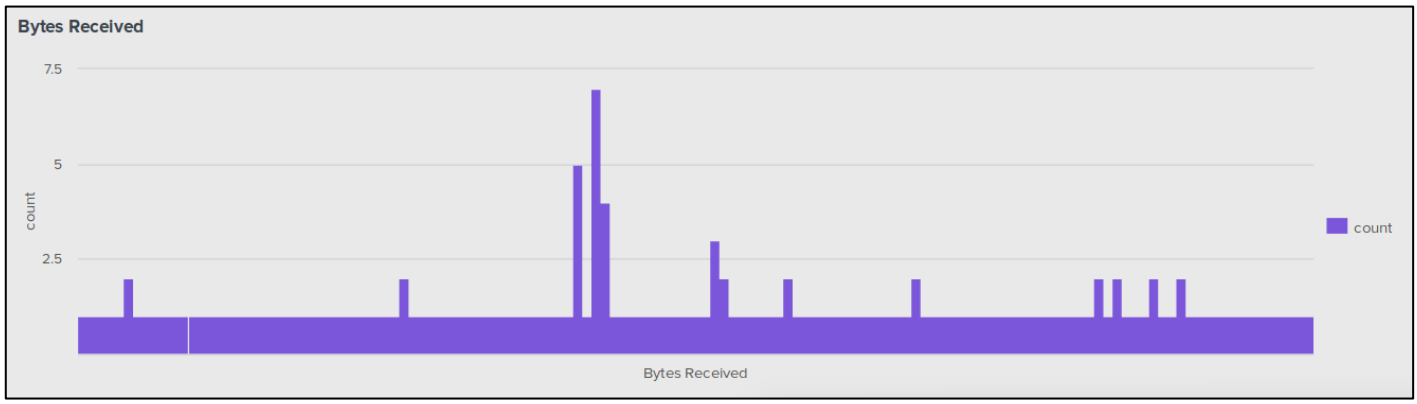
This shows the bytes sent by every port number. The extracted field Bytes_Sent creates a field indicating all of the bytes sent from every port number throughout the entire log. Stats count by Ports was used to search every port with the field Bytes sent acting as a filter.



Bytes Received

(Refer to Appendix, 13)

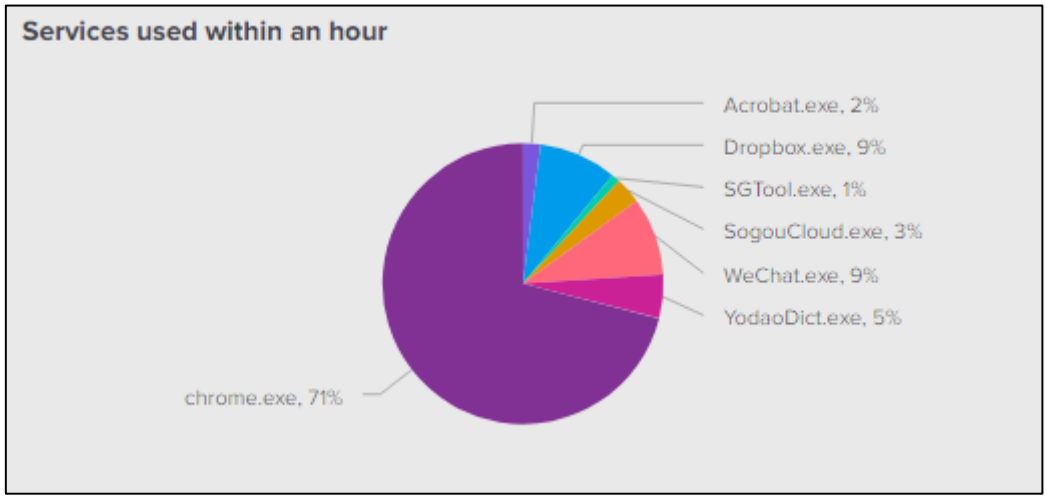
This shows all Bytes received from every port within the log file. The extracted field “Bytes _Received” filters out everything that isn’t receiving bytes. Stats count by Bytes_Received shows it in a numerical format.



Services used within an hour

(Refer to Appendix, 14)

This shows all the services that are being used within a certain timeframe in the log file. In the search a random hour is selected with the earliest and latest fields in the search area of the command, this will make it so that only that timeframe is searched. Stats count by service shows all of the services that were used.



Services and their lifetimes

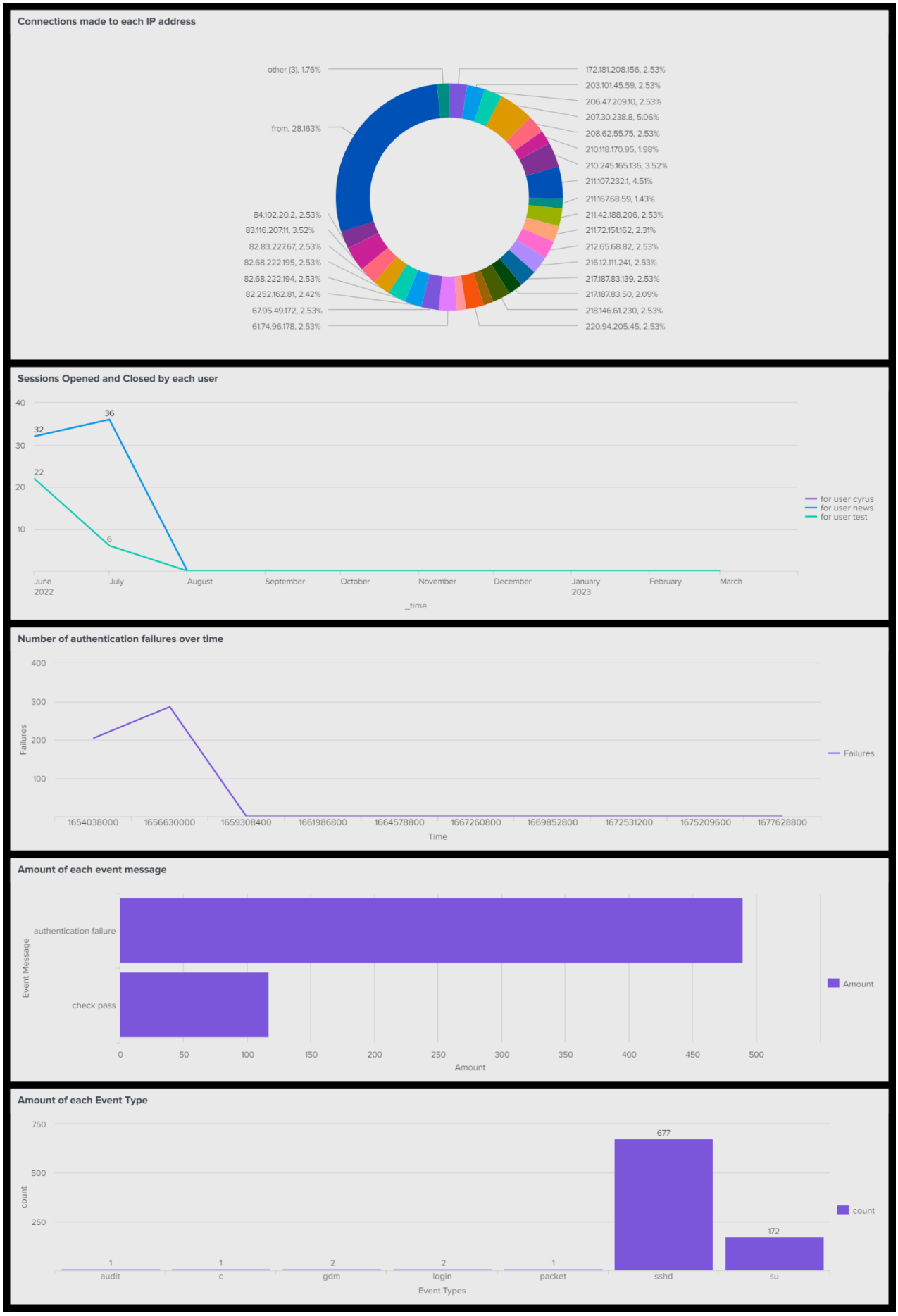
(Refer to Appendix, 15)

This shows all of the services and their lifetimes in a timeframe. The rex command is used to extract the field called “lifetime” which would track how active each service is for within this period of time. The rest makes any empty fields not shown. Table Service,lifetime is used to present the information in a table.

Services and their lifetimes

Service	lifetime
chrome.exe	05:20
chrome.exe	08:46
Dropbox.exe	<1 sec
WeChat.exe	00:32
chrome.exe	04:12
chrome.exe	00:30
chrome.exe	00:19

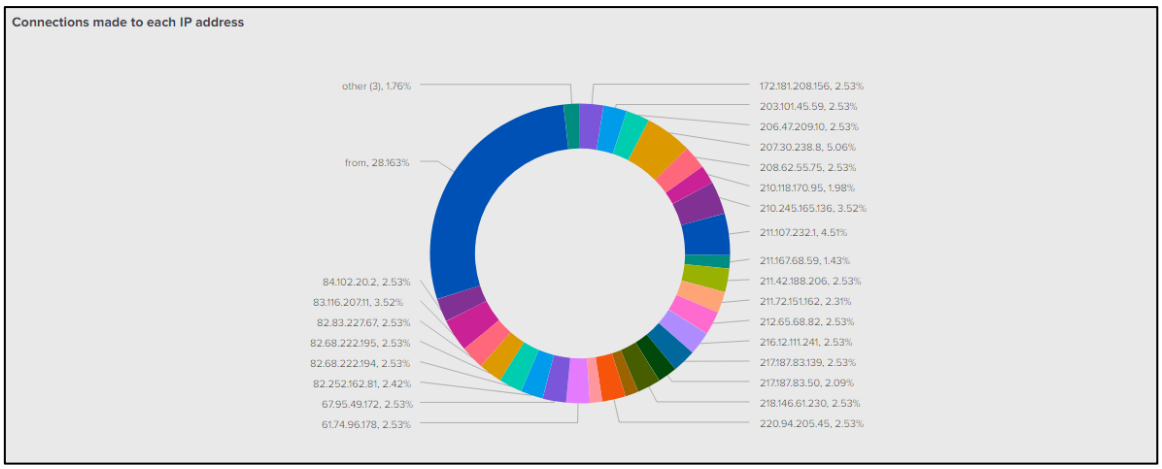
< Prev 1 2 Next >


Connections made to each IP address
other (3), 1.76%
from, 28.163%
84.102.20.2, 2.53%
83.116.207.11, 3.52%
82.83.227.67, 2.53%
82.68.222.195, 2.53%
82.68.222.194, 2.53%
82.252.162.81, 2.42%
67.95.49.172, 2.53%
61.74.96.178, 2.53%
172.181.208.156, 2.53%
203.101.45.59, 2.53%
206.47.209.10, 2.53%
207.30.238.8, 5.06%
208.62.55.75, 2.53%
210.118.170.95, 1.98%
210.245.165.136, 3.52%
211.107.232.1, 4.51%
211.167.68.59, 1.43%
211.42.188.206, 2.53%
211.72.151.162, 2.31%
212.65.68.82, 2.53%
216.12.111.241, 2.53%
217.187.83.139, 2.53%
217.187.83.50, 2.09%
218.146.61.230, 2.53%
220.94.205.45, 2.53%
Sessions Opened and Closed by each user
40
30
20
10
32
36
22
6
June 2022
July
August
September
October
November
December
January 2023
February
March
_time
for user cyrus
for user news
for user test
Number of authentication failures over time
400
300
200
100
Failures
1654038000
1656630000
1659308400
1661986800
1664578800
1667260800
1669852800
1672531200
1675209600
1677628800
Time
Failures
Amount of each event message
Event Message
authentication failure
check pass
0
50
100
150
200
250
300
350
400
450
500
Amount
Amount
Amount of each Event Type
750
500
250
count
1
audit
1
c
2
gdm
2
login
1
packet
677
sshd
172
su
Event Types
count

Connections made to each IP address

(Refer to Appendix, 16)

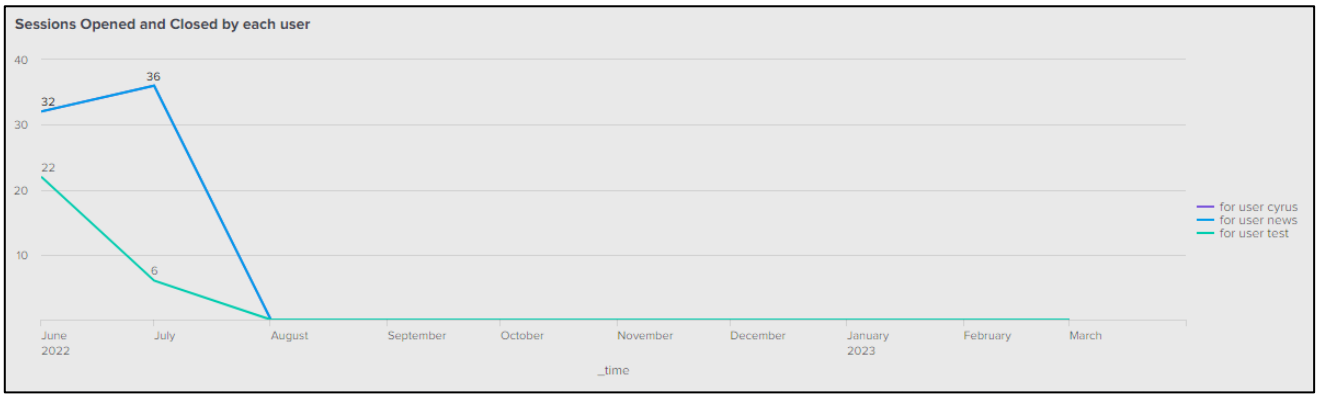
This shows the amount of connections made for each IP address in the log file. The extracted field Connections made is used to filter out anything that isn't making a connection. Stats count by IP_Address presents all of the Ips as statistics.



Sessions opened and closed by each user

(Refer to Appendix, 17)

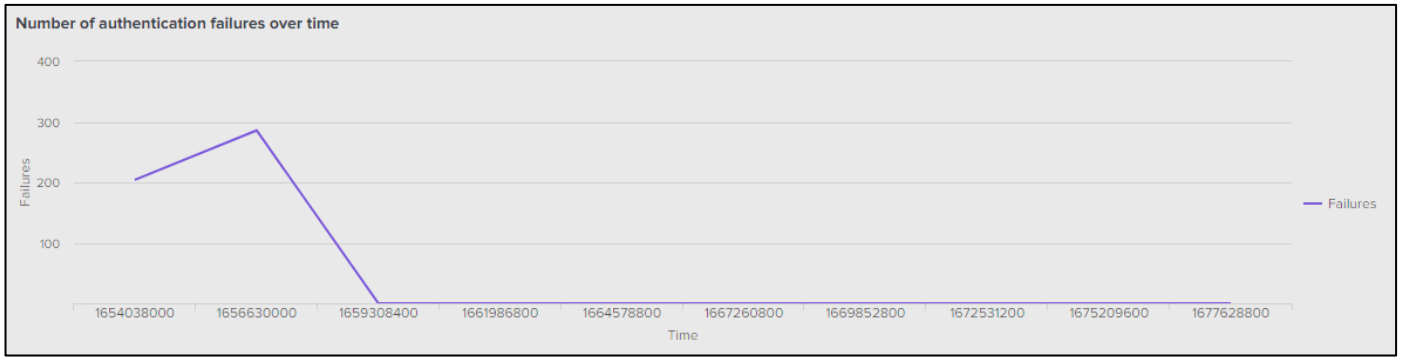
This shows the amount of sessions opened and closed for each user. The extracted field Sessions_Open_Closed shows everything to do with sessions opening and closing in the log file. The rest removes anything that doesn't indicate each user's opening and closing. Timechart count by users is used because its easiest to display this information in a time chart format.



Number of Authentication failures overtime

(Refer to Appendix, 18)

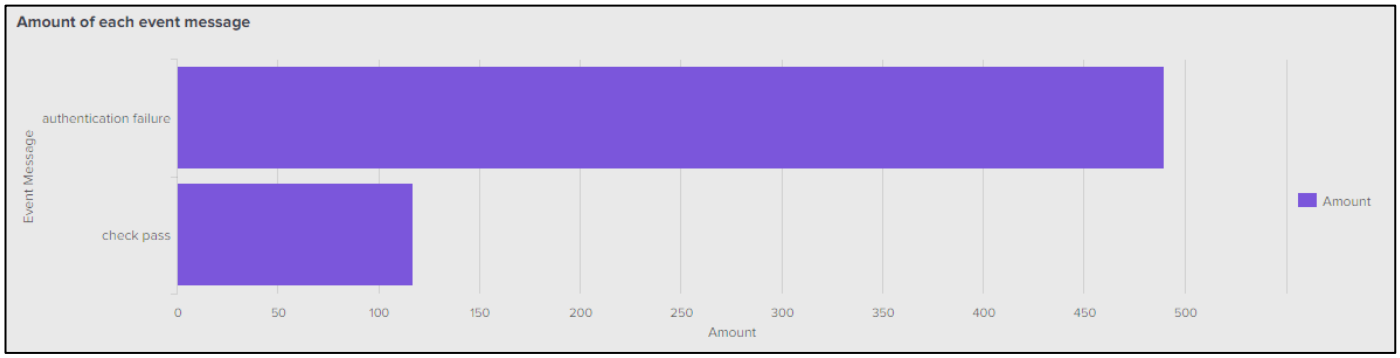
This shows the number of authentication failures overtime in the log file. The expression creates a timechart that shows where the term "authentication failure" appears. Eval uses searchmatch to check if the term "authentication failure" appears.



Amount of each event message

(Refer to Appendix, 19)

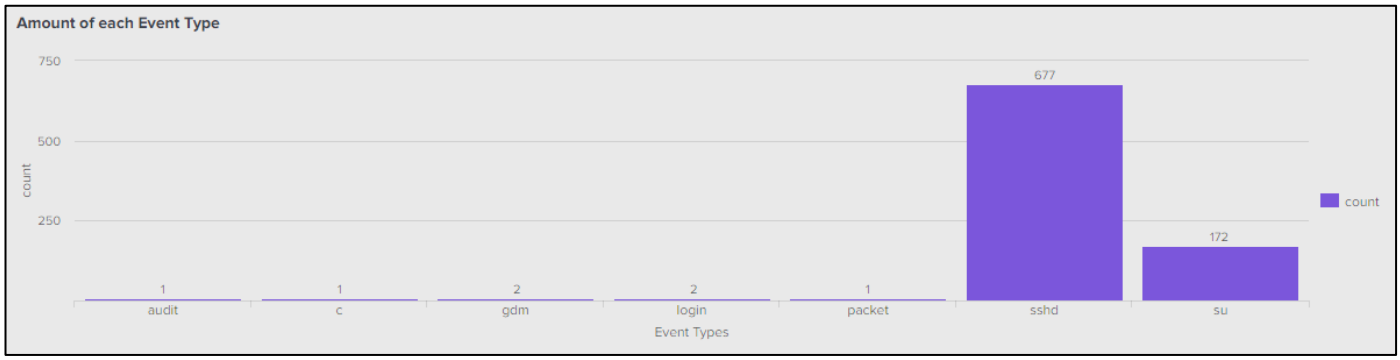
This shows the amount of each event message that appears in the log file. The regular expression command is used to created the extracted field event_message from the log file. Chart count shows the information as a chart so its good for this command.



Amount of each event type

(Refer to Appendix, 20)

This shows the amount of each event type that occurred in the log file. Regular expression is used to extract the field event_type from the log file. The expression used it stats count so it shows as a statistic.

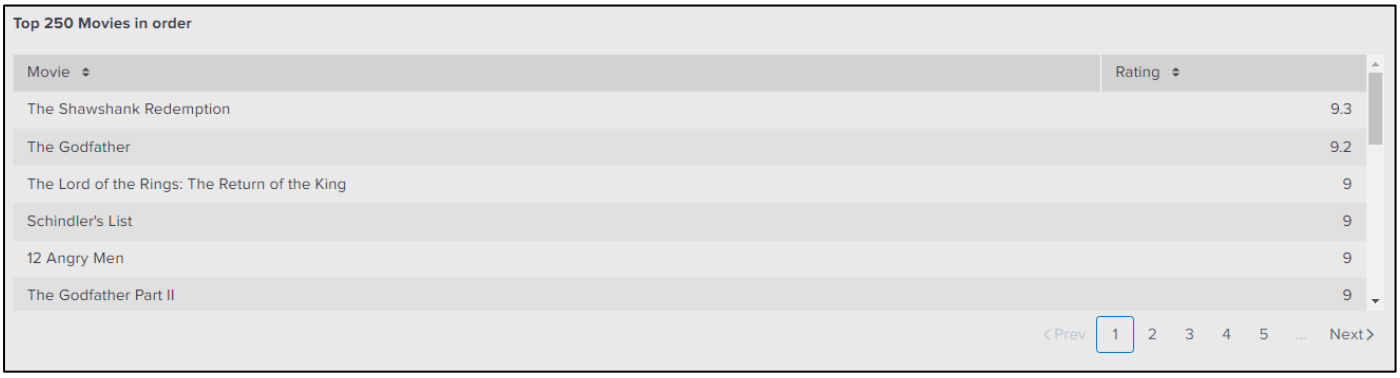


IMDB Top 250 Movies Dataset

Top 250 Movies in order

(Refer to Appendix, 21)

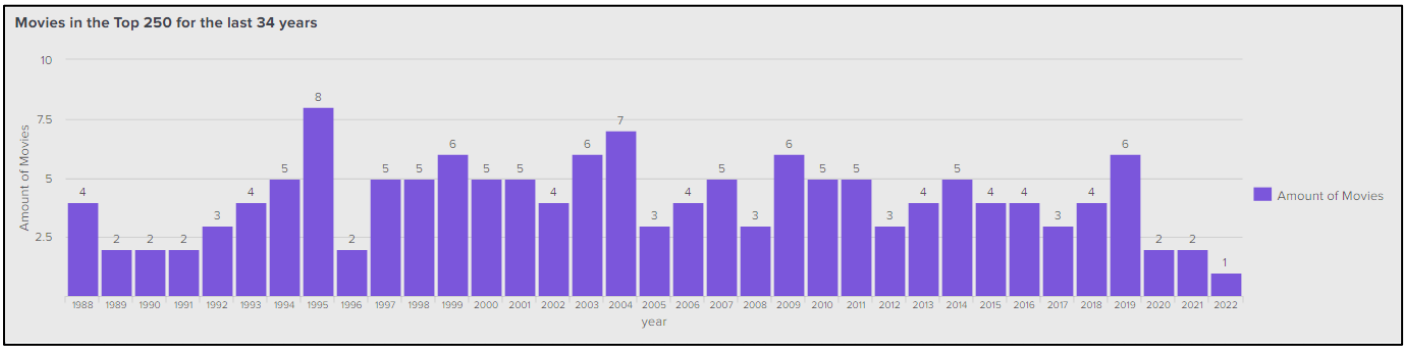
This shows all the movies in the top 250 and their ratings in descending order. Table to put all information into a tabulated form. Sort - rating is how you declare that you want it in descending order.



Movies in the Top 250 for the last 34 years

(Refer to Appendix, 22)

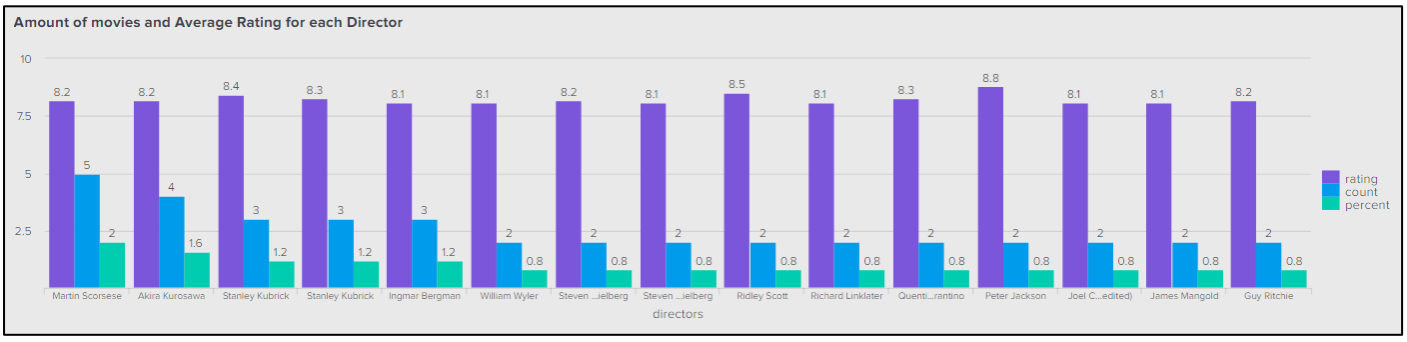
This shows the amount of movies that appear in the top 250 by each year from 1988 to 2022. Earliest and Latest insert the time period that I would like to be searched. Chart count by year puts all the years in a chart format.



Amount of movies and average Rating for each Director

(Refer to Appendix, 23)

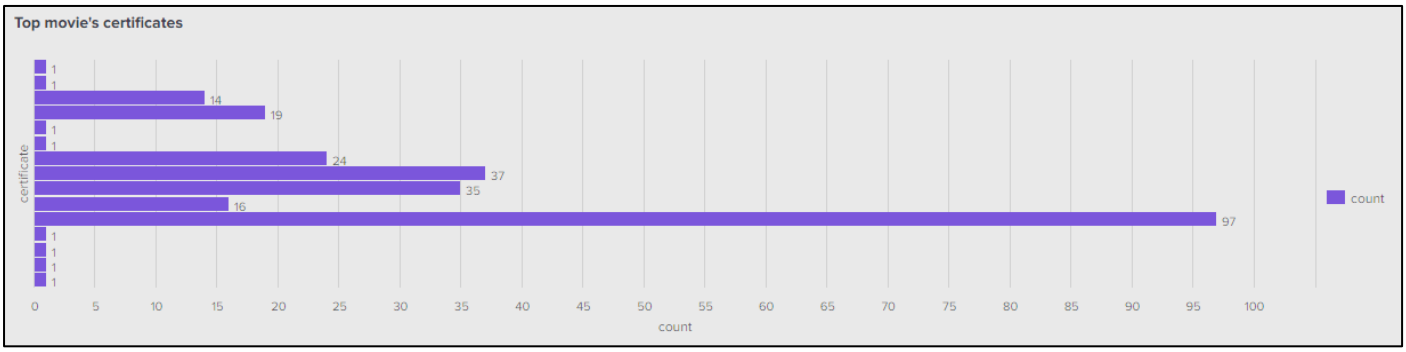
This shows the the average rating for each directing and the amount of movies they have within the top 250. Top limit=15 directors and ratings only displays the top 15 of both fields.



Top movie's certificates

(Refer to Appendix, 24)

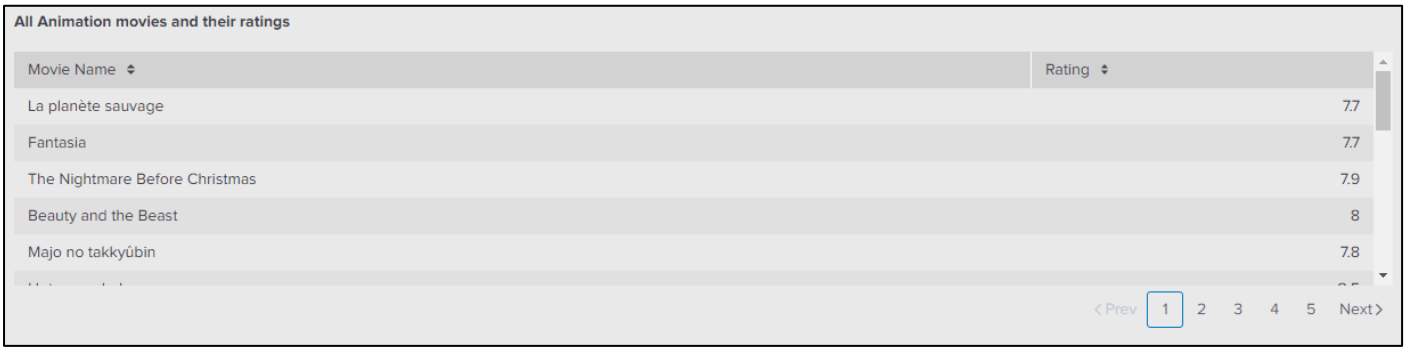
This shows the amount each certificate appears in the top 250. Stats count by certificate displays each certificate in a statistical manner.



All animation movies and their ratings

(Refer to Appendix, 25)

This shows the animation movies and their ratings in the top 250. The extracted field animation filters out anything that isn't an animation. Table movie,imdb rating displays everything but with the extracted field it will only display the animation genre.



Summary of Outcomes

The report included analysation of four log files and one dataset using Splunk to demonstrate and discuss how to use the log analysis tool. Various commands and expressions were included to exhibit a proper understanding of Splunk. One analysis included five commands put into a dashboard and labelled visualisation. It was important that each command output interesting queries related to the information at hand. The level of complexity varied throughout the entire report because it was necessary to go into more detail to show more complex commands.

Overall, the visualisations went well, and various commands were exhibited to demonstrate Splunk's capabilities. There is undoubtedly a learning curve, but when the format of each command is understood, it becomes easier for the user to comprehend.

Wow factor

Demonstration of how to utilise universal forwarder to analyse logs on splunk

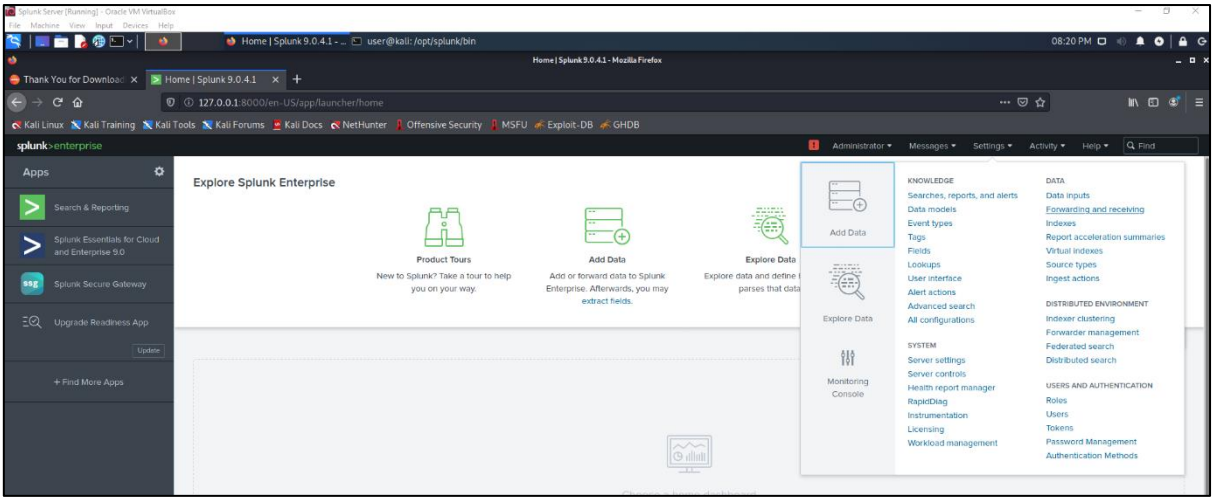


Figure 1: Splunk Enterprise Server set up on a Linux VM is used as a Reciever

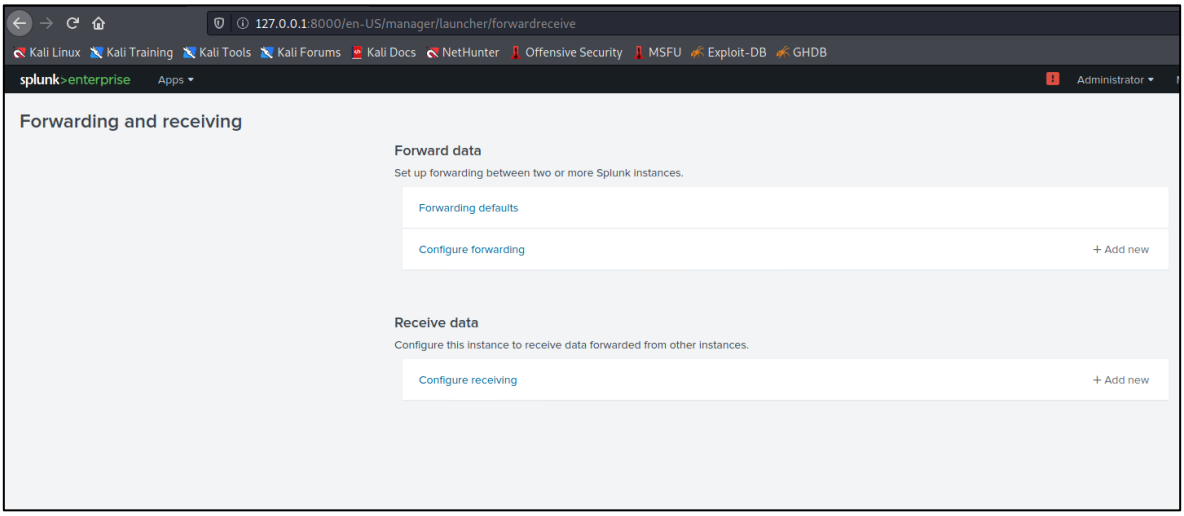


Figure 2: “Add New” next to “Configure receiving” to make Splunk detect certain ports

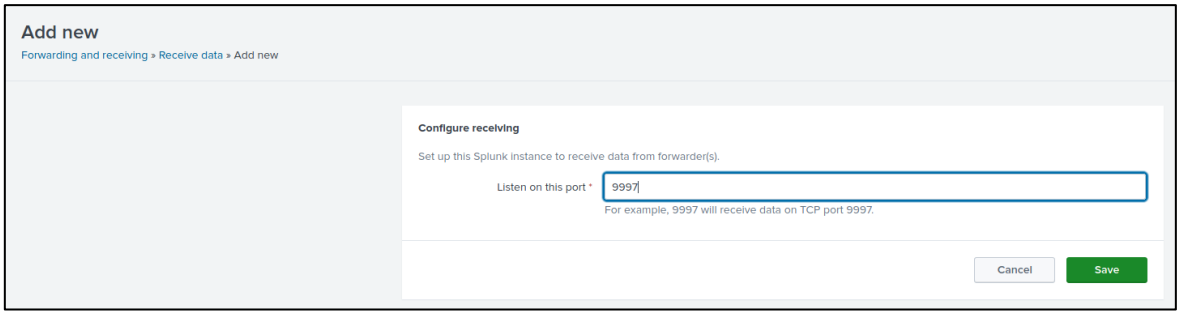


Figure 3: TCP port “9997” as the receiving port number in this example

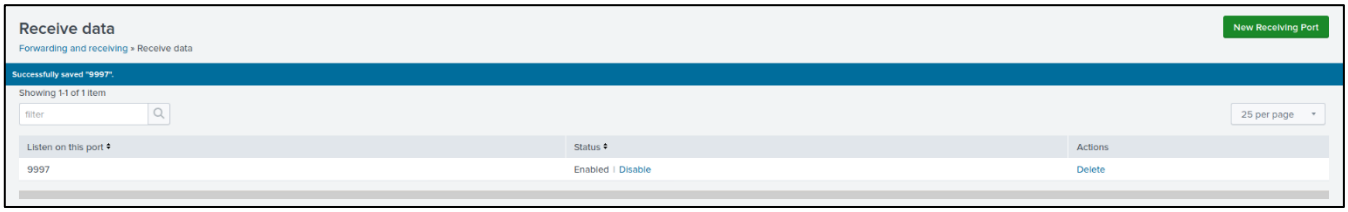


Figure 4: Shows that Receiving data settings were successfully saved

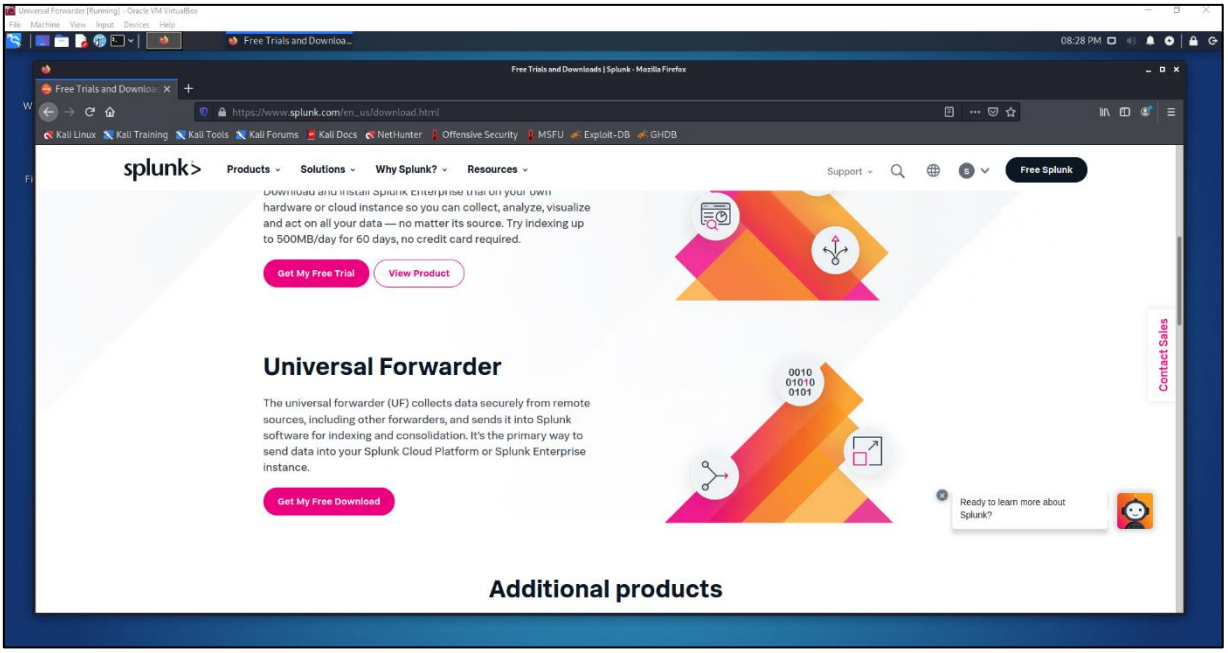


Figure 5: Another Linux VM is used as the universal forwarder

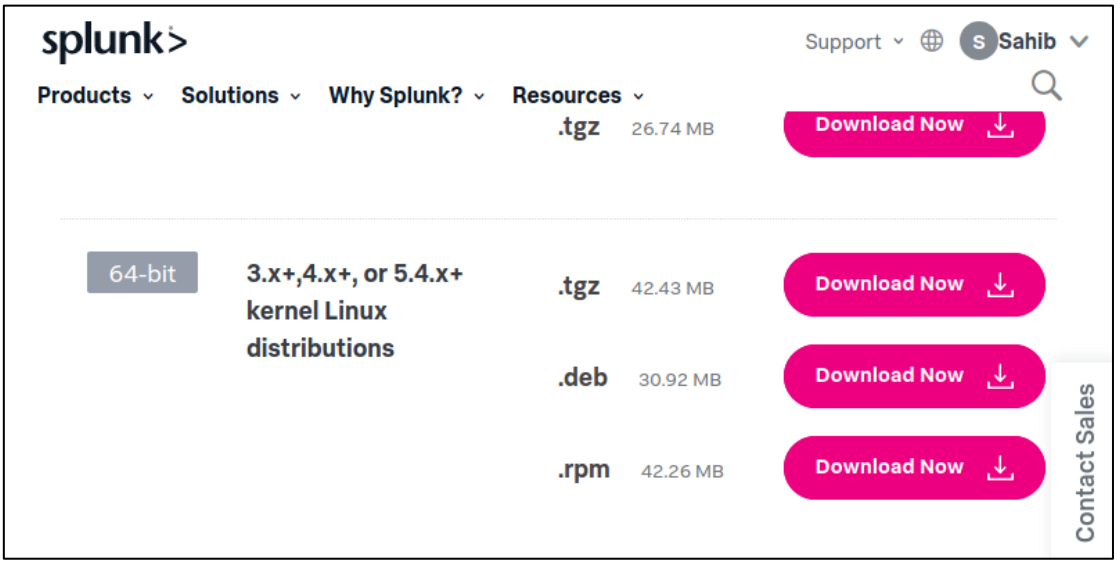


Figure 6: Download correct version of universal forwarder for Linux VM

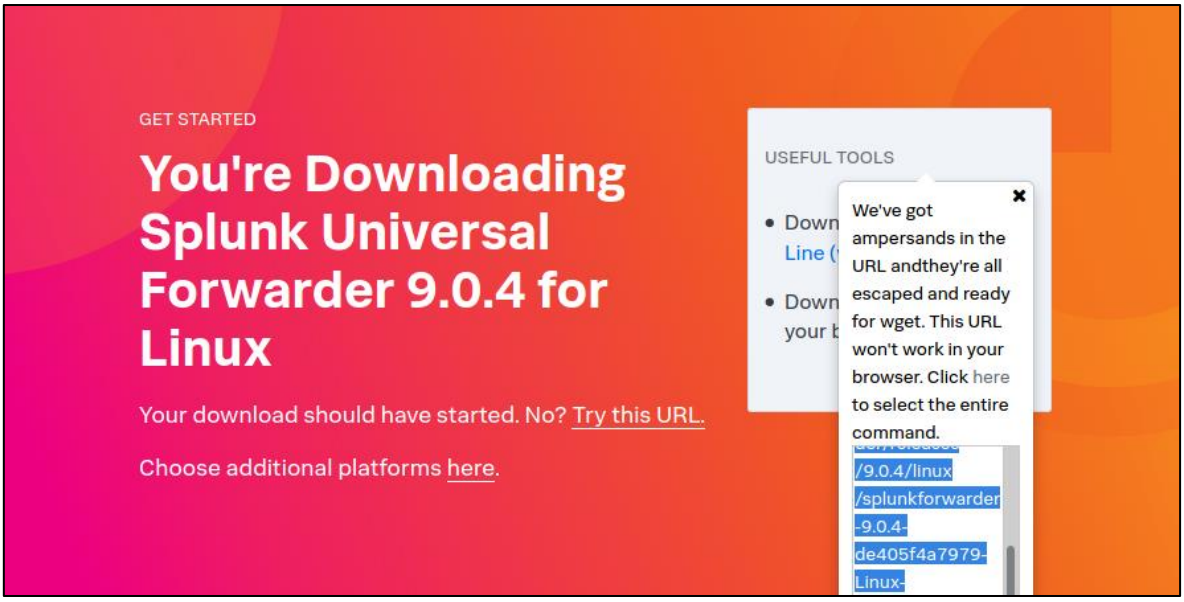


Figure 7: wget command is provided if the user prefers to download the universal forwarder via command line

```
┌──(user㉿kali)-[~]
└─$ sudo wget -O splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz "
https://download.splunk.com/products/universalforwarder/releases/9.0.4
/linux/splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz"
[sudo] password for user:
--2023-04-03 21:43:42--  https://download.splunk.com/products/universalforwarder/relea
ses/9.0.4/linux/splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 143.204.191.78, 143.204.191.70,
 143.204.191.57, ...
Connecting to download.splunk.com (download.splunk.com)|143.204.191.78|:443... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 44489067 (42M) [binary/octet-stream]
Saving to: 'splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz'

splunkforwarder-9.0.4 100%[===================>]  42.43M  40.0MB/s    in 1.1s

2023-04-03 21:43:43 (40.0 MB/s) - 'splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz
' saved [44489067/44489067]
```

Figure 8: The wget command is entered into a terminal and download ensues from there

```
┌──(user㉿kali)-[~]
└─$ ls
Desktop    Music      splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz
Documents  Pictures   Templates
Downloads  Public     Videos

┌──(user㉿kali)-[~]
└─$ sudo tar -xvzf splunkforwarder-9.0.4-de405f4a7979-Linux-x86_64.tgz -C /opt
splunkforwarder/
splunkforwarder/swidtag/
splunkforwarder/swidtag/splunk-UniversalForwarder-primary.swidtag
splunkforwarder/ftr
splunkforwarder/splunkforwarder-9.0.4-de405f4a7979-linux-2.6-x86_64-manifest
splunkforwarder/openssl/
splunkforwarder/openssl/misc/
splunkforwarder/openssl/misc/c_info
splunkforwarder/openssl/misc/tsget
splunkforwarder/openssl/misc/c_issuer
splunkforwarder/openssl/misc/CA.sh
```

Figure 9: It is decompressed using the tar command and moved into the /opt directory

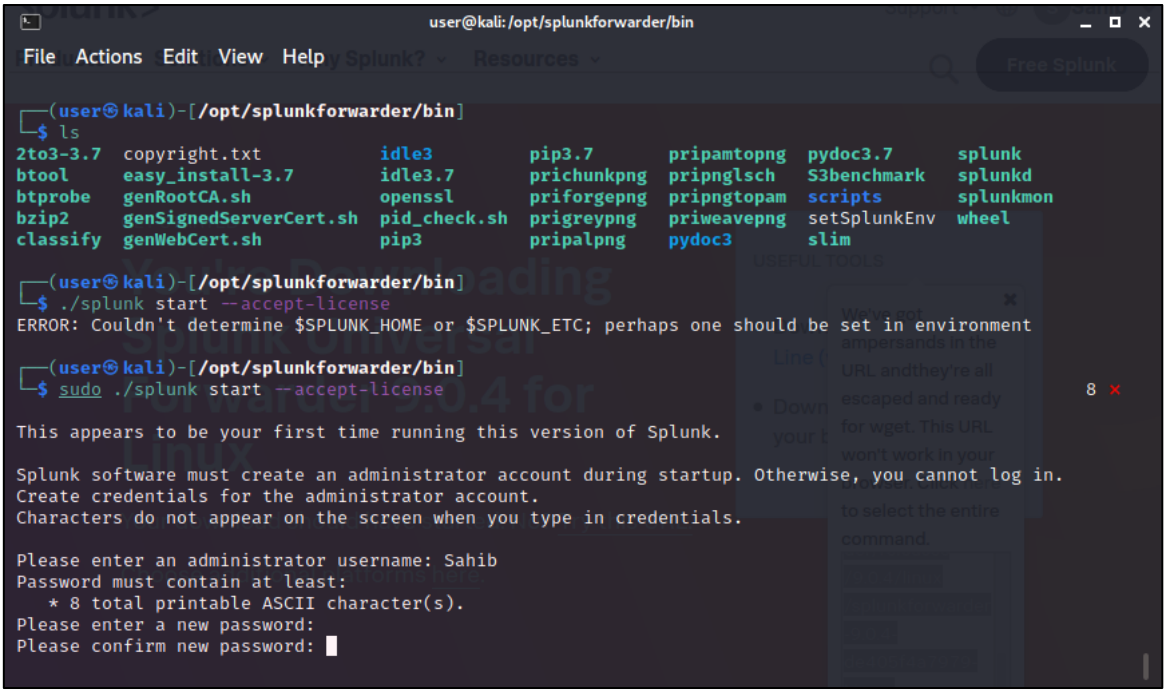


Figure 10: Navigate to the bin directory and automatically accept the license argument using the sudo ./start –accept-license command. It will then prompt the user to create an admin username and password.

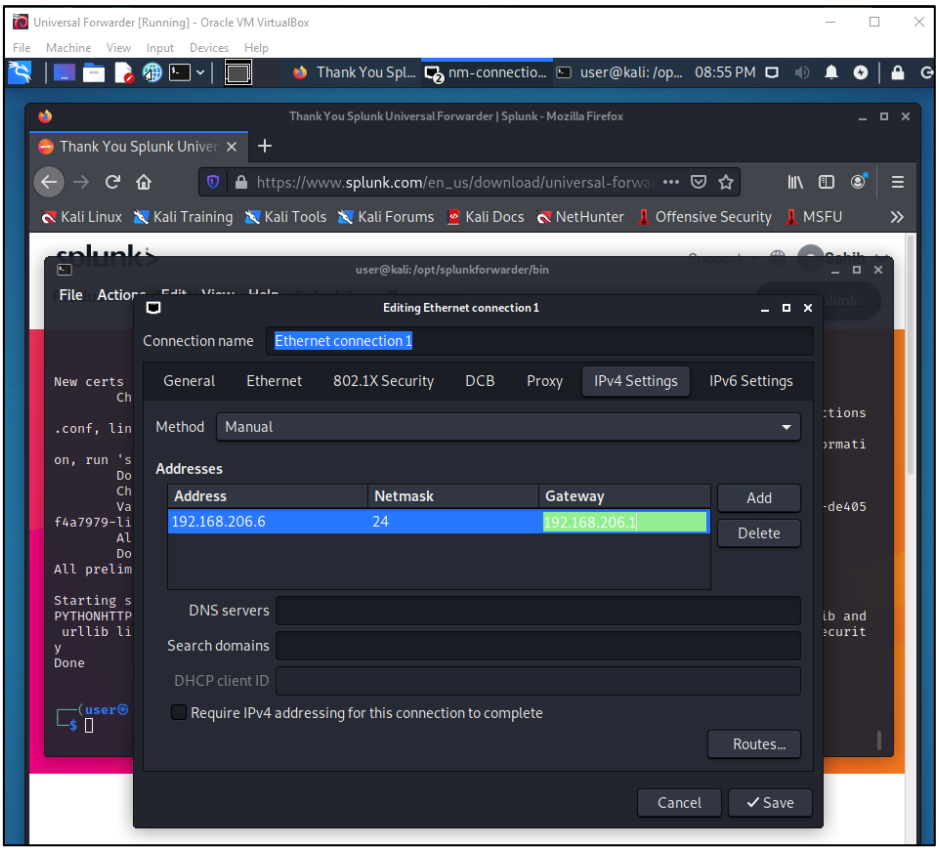


Figure 11: The networks are both changed to “Internal Network” and they are both given an IP address so they can connect to eachother. This figure shows the IP address of the Universal Forwarder VM being changed.

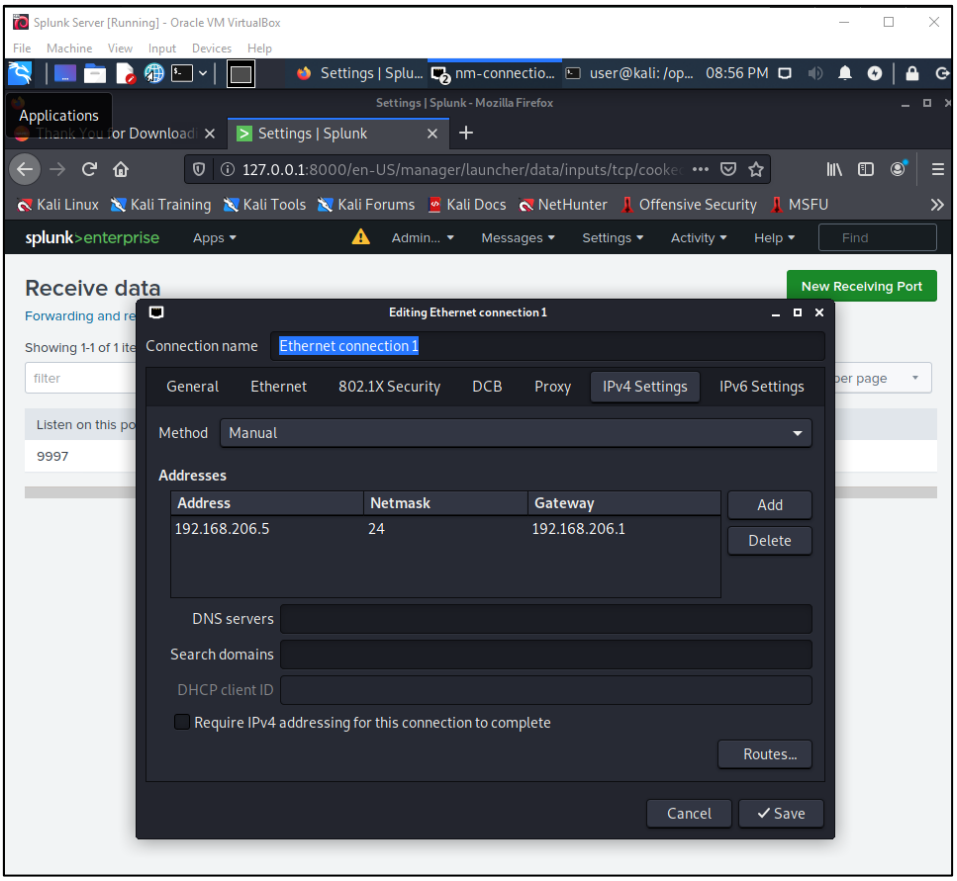


Figure 12: This figure shows the IP address of the receiving indexer VM being changed.

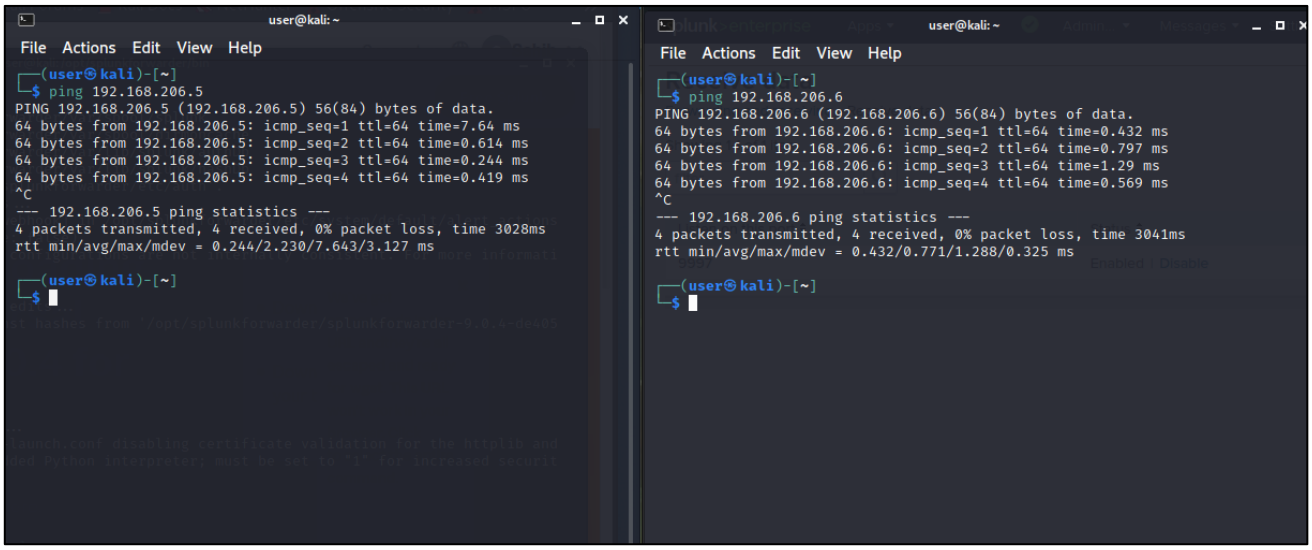


Figure 13: This figures shows that both VMs can successfully ping eachother so they are able to connect to eachother.

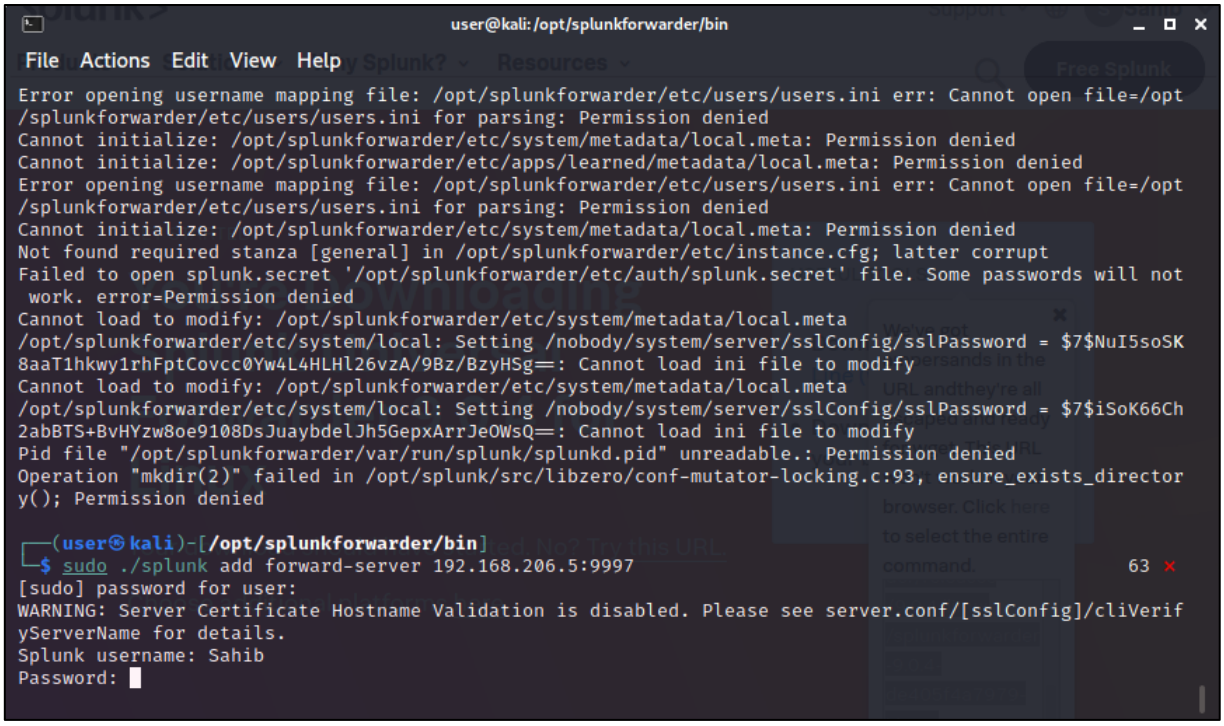


Figure 14: Configure the universal forwarder to send data to the receiving indexer. The command “add forward-server” is used alongside the IP address and port number of the receiving indexer. This port number is the one created in Figure 3.

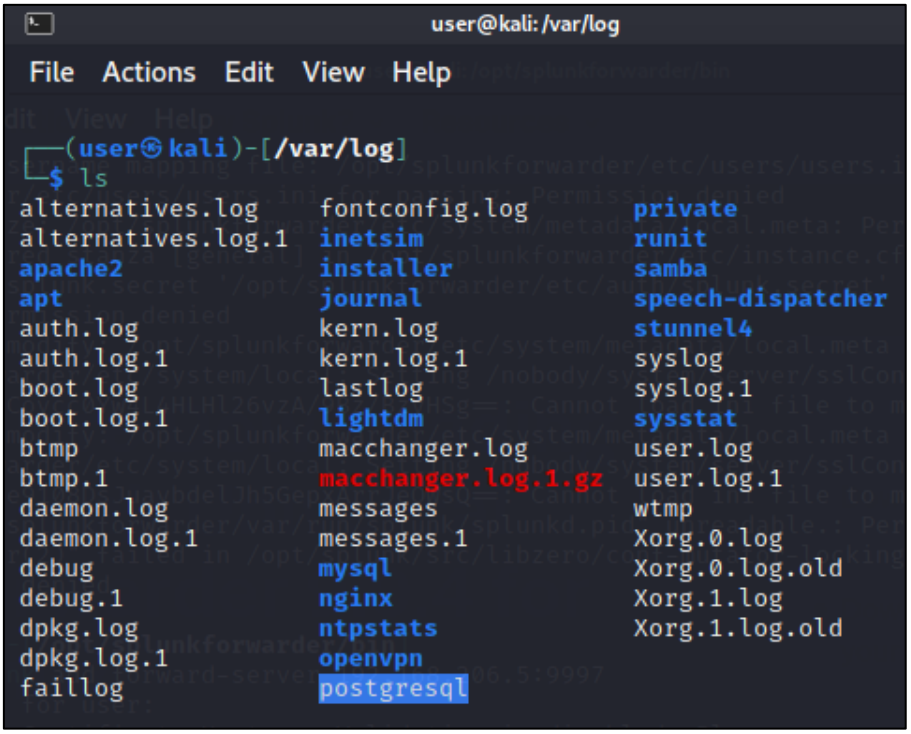


Figure 15: These are all the potential log files that can be sent to the indexer.

```
┌──(user㉿kali)-[/opt/splunkforwarder/bin]
└─$ sudo ./splunk add monitor -auth Sahib:abcd1234 /var/log/auth.log          22 ×
WARNING: Server Certificate Hostname Validation is disabled. Please see server.conf/[sslConfig]/cliVerif
yServerName for details.
Added monitor of '/var/log/auth.log'.
```

Figure 16: The “add monitor” command is used to tell the forwarder what data you have chosen to send. In this case the auth.log file was selected.

```
┌──(user㉿kali)-[/opt/splunkforwarder/bin]
└─$ sudo ./splunk add monitor -auth Sahib:abcd1234 /var/log/boot.log          63 ×
WARNING: Server Certificate Hostname Validation is disabled. Please see server.conf/[sslConfig]/cliVerif
yServerName for details.
Added monitor of '/var/log/boot.log'.
```

Figure 17: The boot.log file was also selected to be sent to splunk.

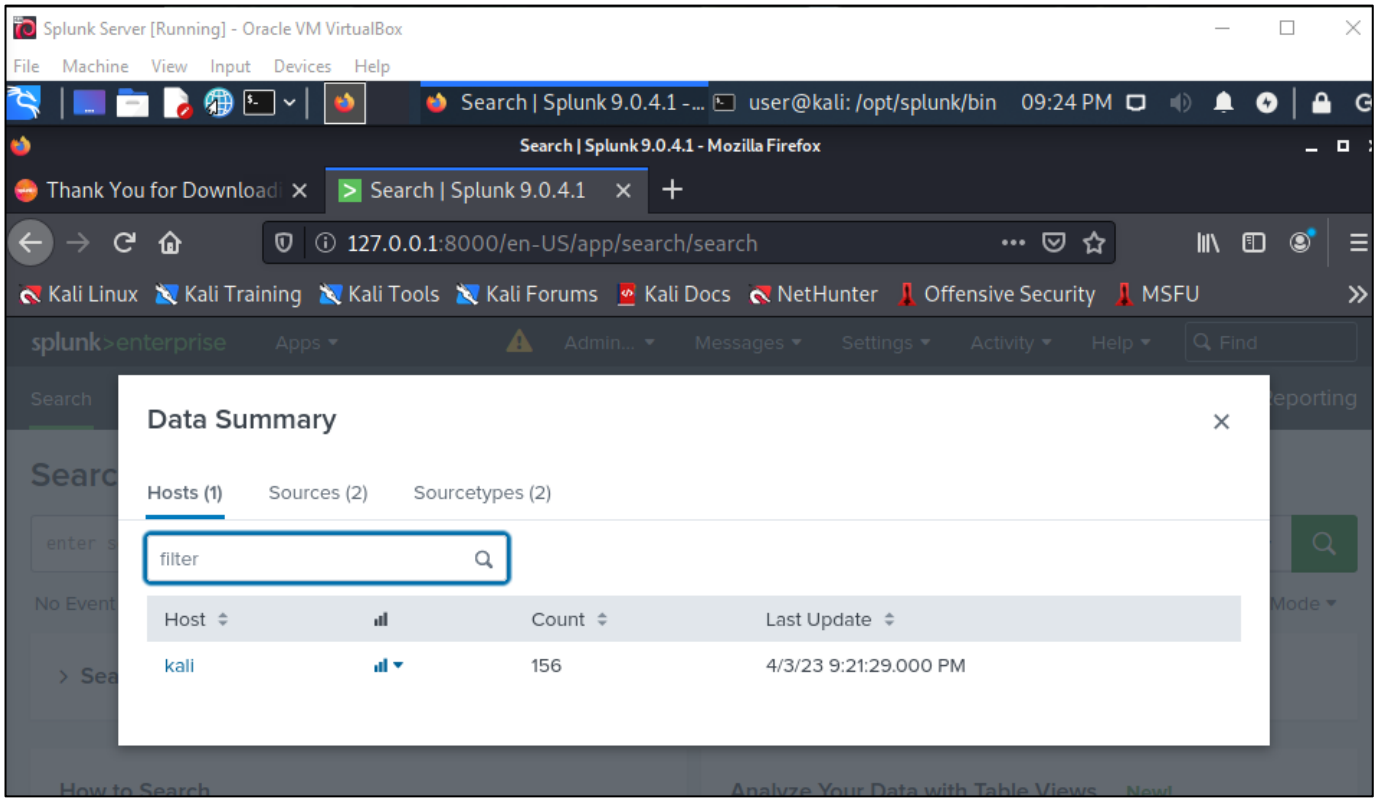


Figure 18: After returning to the indexer and clicking on the data summary on the search and report section of splunk, you may notice that there is a new host. Meaning that the data successfully indexed.

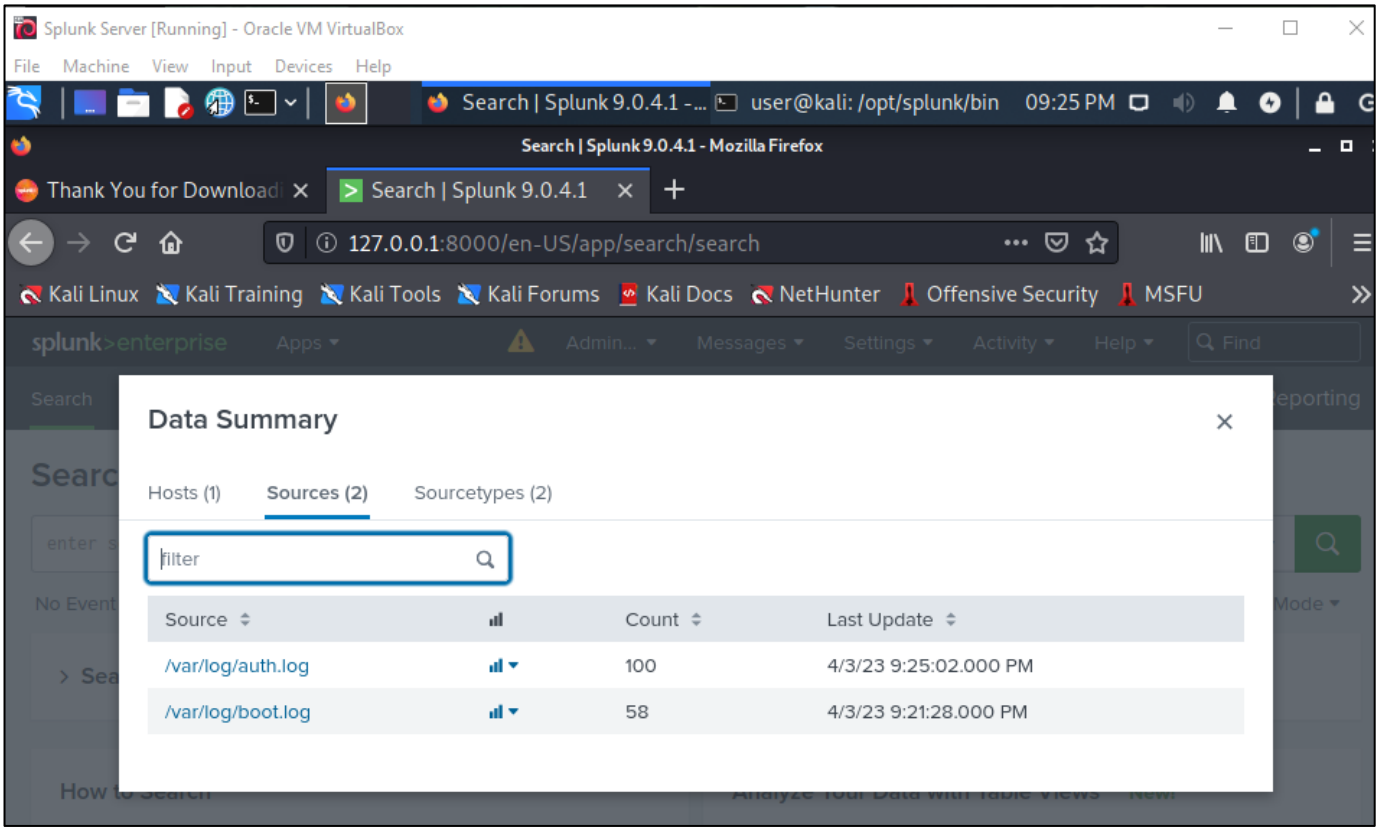


Figure 19: Clicking on Sources, you'll see that both the log files that were selected have appeared.

Figure 20: Example of the auth.log file working in splunk

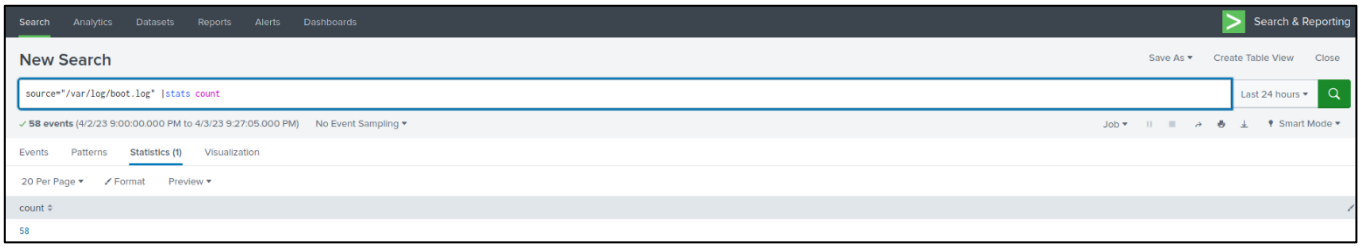


Figure 21: Example of the boot.log working in splunk

Conclusion

After completing a full assessment of the log management tool Splunk in action, the goal of creating a proper understanding and awareness of what log file management tools are capable of was created. A deep dive into the analysation and management side of the log file and data shows how much of a significant impact for various types of users, from individuals to the organisation. No matter the customer, this type of security will help somehow. By implementing a comprehensive strategy, users can benefit from increased system performance, early detection of issues, and valuable insights for decision-making. For organisations, it is a priority to reduce downtime. Having the correct log management tool to keep track of system trends will ensure that their systems remain efficient and secure.

Box Link for video

https://kingston.box.com/s/ul2idb9clpr93hzhssbgyajbdka2wc28

References

Amazon Web Services (2021). *What are Log Files? - Log Files Explained - AWS*. [online] Amazon Web Services, Inc. Available at: https://aws.amazon.com/what-is/log-files/ [Accessed 28 Mar. 2023].

Elastic Stack (2021). *Elastic Stack and Product Documentation | Elastic*. [online] www.elastic.co. Available at: https://www.elastic.co/guide/index.html [Accessed 28 Mar. 2023].

Graylog (2023). *Graylog Documentation*. [online] go2docs.graylog.org. Available at: https://go2docs.graylog.org/5-0/home.htm [Accessed 28 Mar. 2023].

Sharif, A. (2022). *Log Files: Definition, Types, and Importance | CrowdStrike*. [online] crowdstrike.com. Available at: https://www.crowdstrike.com/cybersecurity-101/observability/log-file/ [Accessed 28 Mar. 2023].

Watts, S. (2022). *Log Management: A Useful Introduction*. [online] Splunk-Blogs. Available at: https://www.splunk.com/en_us/blog/learn/log-management.html [Accessed 28 Mar. 2023].

Appendix

Thunderbird Log File Commands

1. index=_* OR index=* sourcetype=Thunderbird SynchronizedIP=synchronized | chart count by IPAddress | rename count as "Successful Synchronization"
2. index=_* OR index=* sourcetype=Thunderbird No_Response="got not answer"| stats count by Datasource | rename count as "No response"
3. index=_* OR index=* sourcetype=Thunderbird Mail_ID="*"| timechart count by Mail_ID |rename _time AS "Time", avg(Mail_ID) AS Usage
4. index=_* OR index=* sourcetype=Thunderbird| top limit=20 IPAddress

5. index=_* OR index=* sourcetype=Thunderbird "PCI Interrupt"| rex "Interrupt\s(?<Process>\w+)\s" | chart count by Process | rename count AS "Amount of Processes"

Zookeeper Log File Commands

6. index=_* OR index=* sourcetype="Zookeeper Log" | stats count by Connection_Request| rename Connection_Request as "Connection Request"

7. index=_* OR index=* sourcetype="Zookeeper Log"| timechart count by New_session limit=10 |rename "_time" as Time

8. source="Zookeeper_2k.log.txt" sourcetype="Zookeeper Log File" "Received connection request"| rex "(?<ip_address>\d+\.\d+\.\d+\.\d+)" |stats count by ip_address |rename ip_address as "IP Address"|rename count as "Count of Events"

9. index=_* OR index=* sourcetype="Zookeeper Log" | top limit=20 Closed_Connection

10. index=_* OR index=* sourcetype="Zookeeper Log" "RecvWorker" "Interrupting" earliest = "03/14/2023:12:00:00" latest = "03/14/2023:24:00:00" | timechart count span=12hr

Proxifier Log File Commands

11. index=_* OR index=* sourcetype="Proxifier Log File"| timechart count by Open_Close limit=10

12. index=_* OR index=* sourcetype="Proxifier Log File" Bytes_Sent="*"| stats count by Ports |rename count as "Bytes Sent"

13. index=_* OR index=* sourcetype="Proxifier Log File" Bytes_Received="*"|stats count by Bytes_Received |rename Bytes_Received as "Bytes Received"

14. index=_* OR index=* sourcetype=Proxifier earliest = "01/18/2018:13:49:38" latest = "01/18/2018:14:49:29" | stats count by Service

15. index=_* OR index=* sourcetype=Proxifier earliest = "01/18/2018:14:15:38" latest = "01/18/2018:14:49:29" | rex field=_raw "lifetime\s(?<lifetime>.*$$)" | fields Service, lifetime | where NOT isnull(Service) AND NOT isnull(lifetime) | table Service,lifetime

Linux Log File Commands

16. index=_* OR index=* sourcetype="Linux Log" Connections_Made=connection | stats count by IP_Address |rename IP_Address as "IP Address"

17. index=_* OR index=* sourcetype="Linux Log" Sessions_Open_Closed="*" Users!="timed out after" Users!="section contains no" Users!="opened for user" Users!="closed for user" Users!="LOGIN FROM 84" | timechart count by Users

18. index=_* OR index=* sourcetype="Linux Log" | timechart count(eval(searchmatch("authentication failure"))) as Failures |rename _time as "Time"

19. index=_* OR index=* sourcetype="Linux Log" | rex "(?<=\:)(?<event_message>[\w\s]+)(?=;)" | chart count by event_message | rename event_message as "Event Message", count as "Amount"

20. index=_* OR index=* sourcetype="Linux Log" | rex "(?P<event_type>\w+)\(" | stats count by event_type

IMDB Top 250 movies Dataset Commands

21. source="archive (1).zip:*" |table name, rating | sort - rating | rename name as "Movie", rating as "Rating"

22. source="archive (1).zip:*" earliest=04/19/1988:00:00:00 latest=04/27/2022:00:00:00 |chart count by year | rename count as "Amount of Movies"

23. source="archive (1).zip:*"| top limit=15 directors,rating

24. source="archive (1).zip:*" certificate="*" | stats count by certificate

25. index=_* OR index=* sourcetype=csv genre_1=Animation | table movie,imdb_rating |rename movie as "Movie Name", imdb_rating as "Rating"