holmesglen

Assessment Task – Portfolio (Learner Version)

Assessment Task 2: Threat Data Analysis

Part -1: Research Questions
Questions: Provide your responses in the boxes below each question.

Question 1.1:	How is Splunk licensed? Choose the right answer? a. Based on the amount of data ingested each day b. Based on the number of concurrent users c. Based on the number of searches executed d. Based on the number of systems sending data	**Satisfactory response** Yes ☒ No ☐
#Your response a. Based on the amount of data ingested each day		Comment:
Question 1.2:	Which Splunk system role physically stores/archives data within a deployment. a. Search head b. Indexer c. Forwarder d. Deployment server	**Satisfactory response** Yes ☒ No ☐
#Your response a. Indexer		Comment:
Question 1.3:	Both syslog and a Splunk Universal Forwarder are supported for ingesting the data you want to bring into Splunk. Which one do you prefer?	**Satisfactory response** Yes ☒ No ☐
#Your response I would prefer the Splunk Universal Forwarder due to its advantages compared to the syslog. Splunk Universal Forwarder allowing efficient data forwarding with metadata, have a better reliability to ensure no data is lost, and support encrypted data in transit by using SSL. References: https://www.splunk.com/en_us/blog/learn/splunk-universal-forwarder.html		Comment:
Question 1.4:	Your organization has two data centres and would like to ensure that no data in Splunk is lost if one location were to fail. What is the best type of clustering methodology to accomplish this goal?	**Satisfactory response** Yes ☒ No ☐
#Your response The best type of clustering is the Multisite Indexer Clustering which allows the organisation to replicate the data across multiple data centres and provide fault tolerance by replicate the data to indexers across different physical locations. References: https://docs.splunk.com/Documentation/Splunk/9.3.2/Indexer/Multisitearchitecture		Comment:
Question 1.5:	What search did you use for your table of firewall logs?	**Satisfactory response**

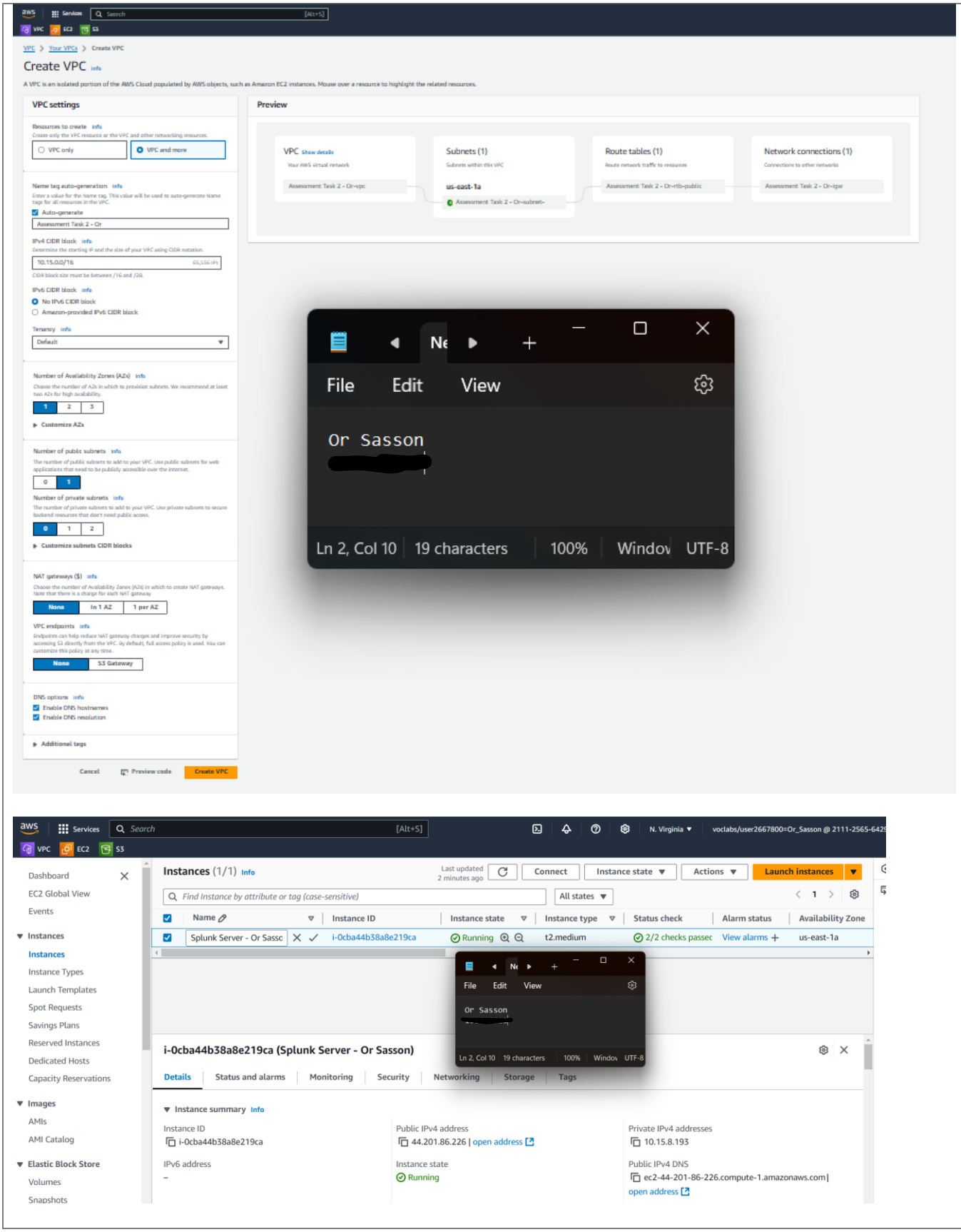
		Yes ☒ No ☐
#Your response sourcetype="linux:netfilter" | top limit=5 dest_port		**Comment**
Question 1.6:	What search did you use to build a table for your linux_secure logs?	**Satisfactory response** Yes ☒ No ☐
#Your response sourcetype=linux_secure src_ip=* eventtype=sshd_authentication | table _time,action, user, src_ip		Comment:
Question 1.7:	What is Search Processing Language?	**Satisfactory response**
#Your Response It is the main language used in Splunk for data analysis. This language allows to retrieve data from the indexes with conditions and filters. In addition, it can be used to create charts, tables, and dashboards. References: https://docs.splunk.com/Documentation/SplunkCloud/latest/Search/Aboutthesearchlanguage		Yes ☒ No ☐

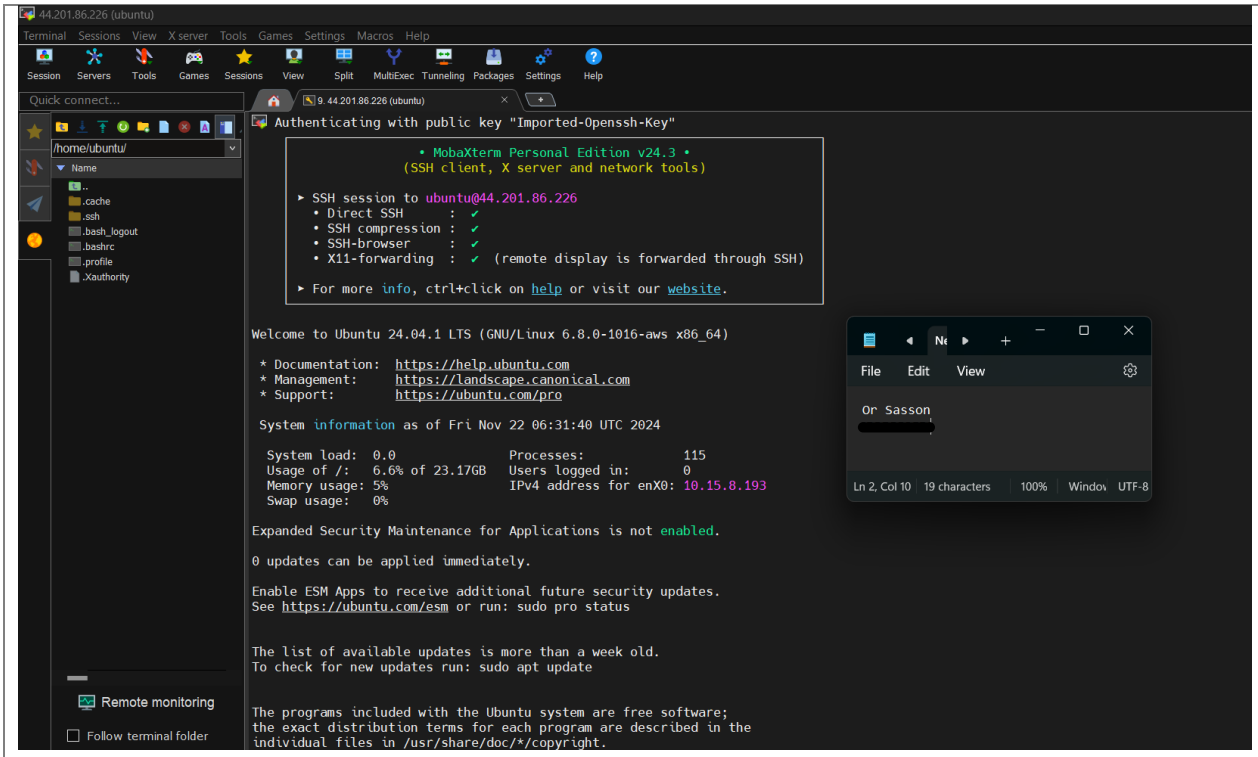
Part-2: Setup Splunk on Ubuntu VM & Splunk
Please follow these steps to complete the task-2

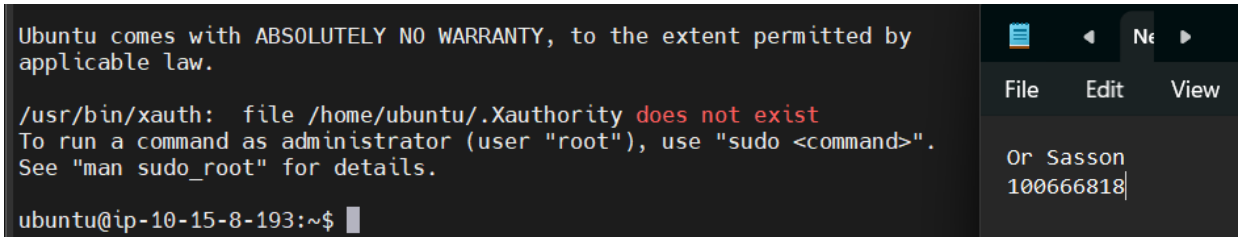
Step-1 Setup your own ubuntu Virtual Machine

Q2.1: Provide Screenshot of running ubuntu VM.

#Screenshot


Create VPC
VPC settings
Preview
VPC and more
Auto-generate
Assessment Task 2 - Or
10.15.0.0/16
No IPv6 CIDR block
Default
Number of Availability Zones (AZs)
Number of public subnets
Number of private subnets
NAT gateways ($)
None
In 1 AZ
1 per AZ
VPC endpoints
S3 Gateway
Enable DNS hostnames
Enable DNS resolution
Additional tags
Cancel
Preview code
Create VPC
File
Edit
View
Or Sasson
Ln 2, Col 10
19 characters
100%
UTF-8
Instances (1/1)
Launch instances
Splunk Server - Or Sassc
i-0cba44b38a8e219ca
Running
t2.medium
2/2 checks passec
us-east-1a
i-0cba44b38a8e219ca (Splunk Server - Or Sasson)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary
Instance ID
i-0cba44b38a8e219ca
Public IPv4 address
44.201.86.226 | open address
Private IPv4 addresses
10.15.8.193
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-44-201-86-226.compute-1.amazonaws.com | open address

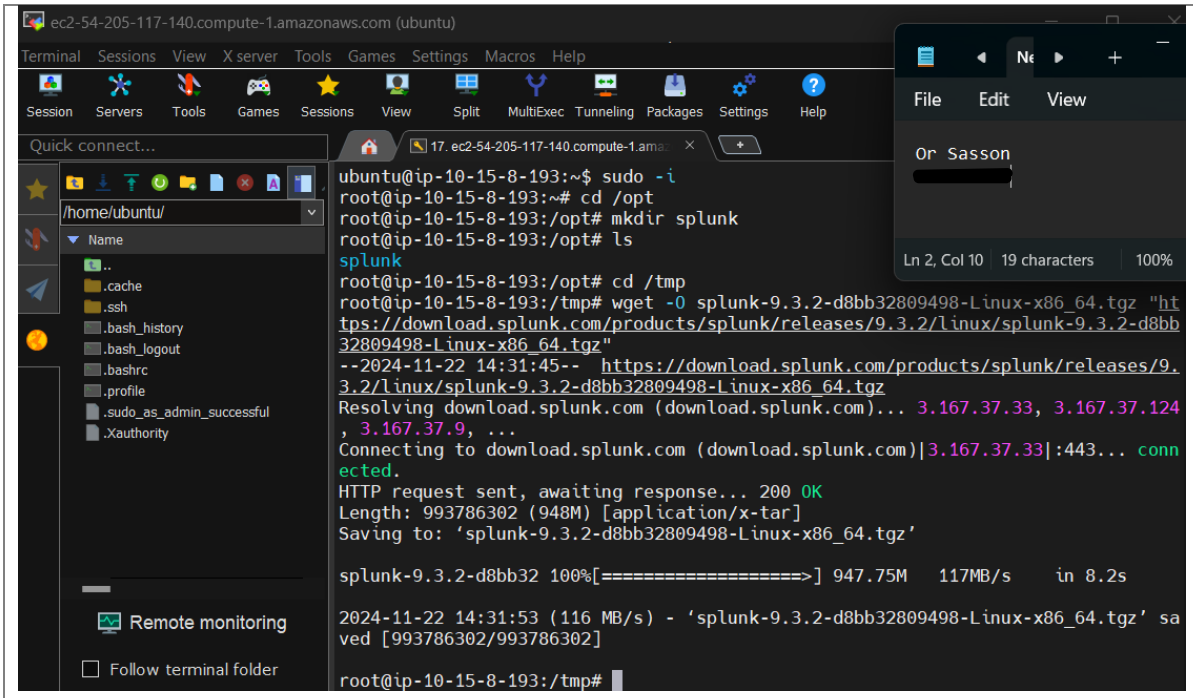


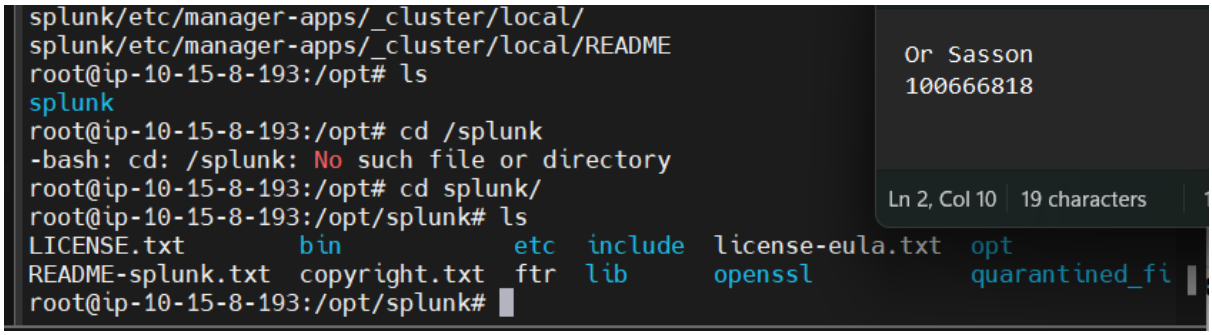


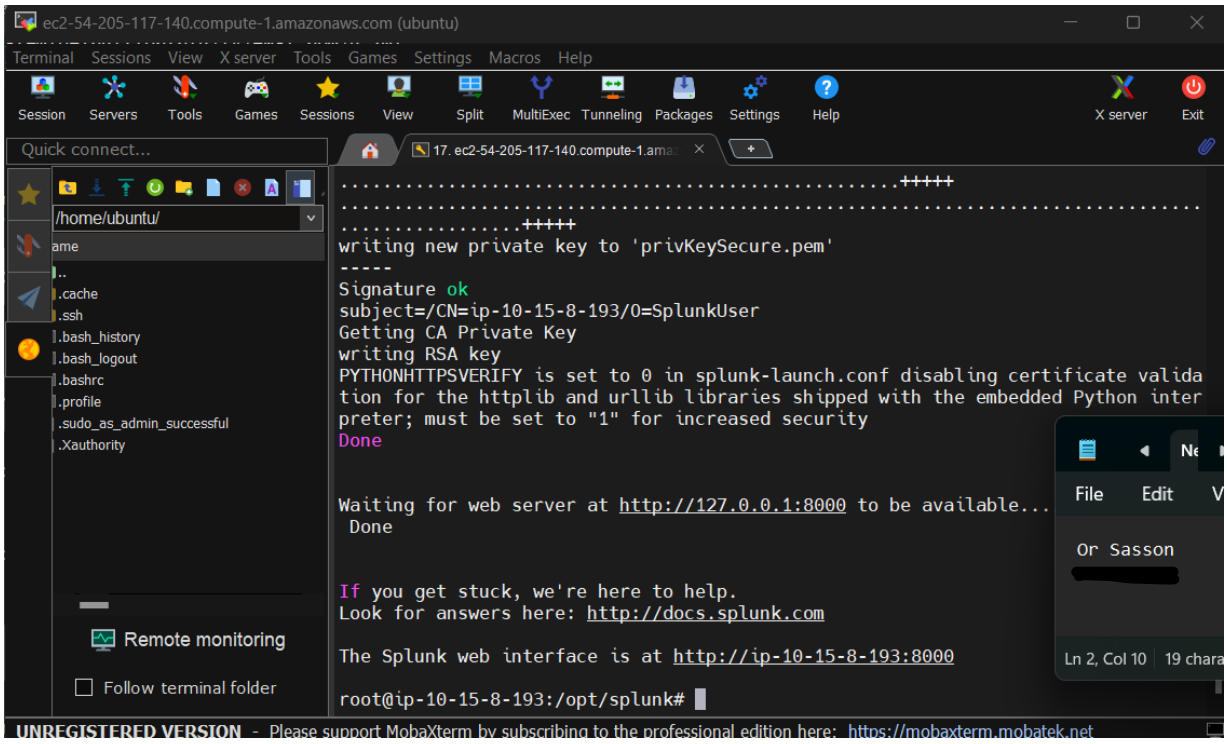
Step-2 Install Splunk on Ubuntu VM

Q2.2 Provide the Screen shot of Splunk running on Ubuntu VM

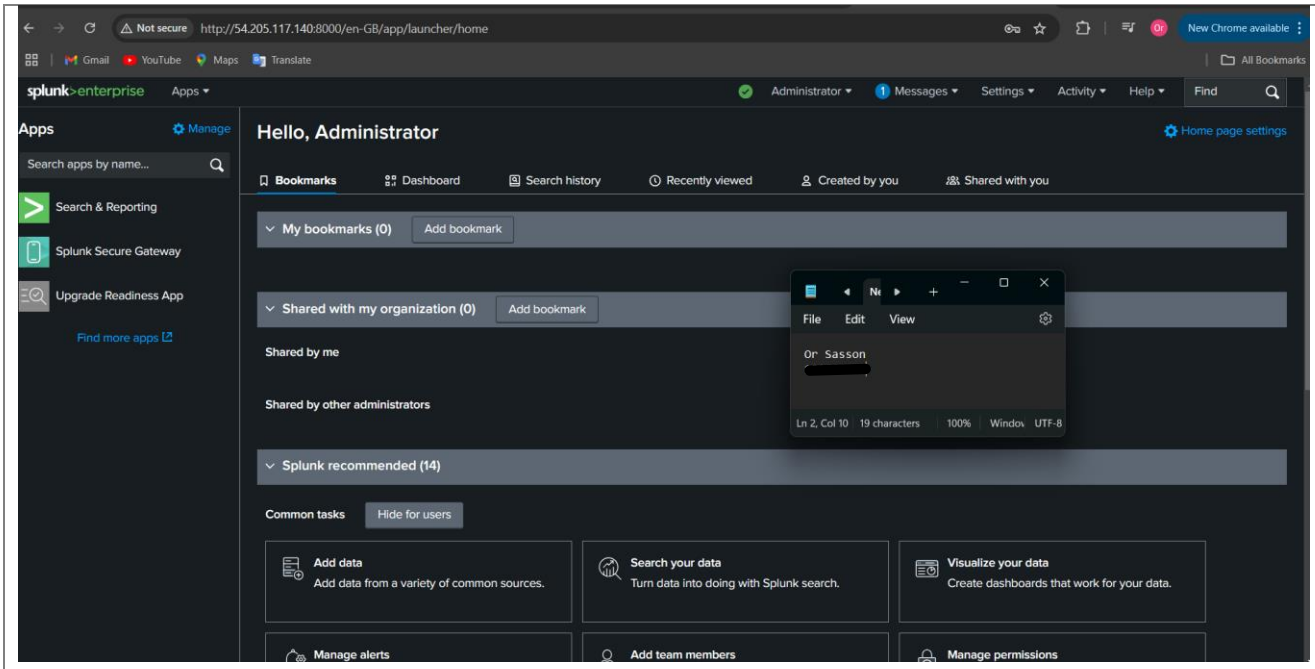
#Screenshot


ec2-54-205-117-140.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
.cache
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
Remote monitoring
Follow terminal folder
ubuntu@ip-10-15-8-193:~$ sudo -i
root@ip-10-15-8-193:~# cd /opt
root@ip-10-15-8-193:/opt# mkdir splunk
root@ip-10-15-8-193:/opt# ls
splunk
root@ip-10-15-8-193:/opt# cd /tmp
root@ip-10-15-8-193:/tmp# wget -O splunk-9.3.2-d8bb32809498-Linux-x86_64.tgz "https://download.splunk.com/products/splunk/releases/9.3.2/linux/splunk-9.3.2-d8bb32809498-Linux-x86_64.tgz"
--2024-11-22 14:31:45-- https://download.splunk.com/products/splunk/releases/9.3.2/linux/splunk-9.3.2-d8bb32809498-Linux-x86_64.tgz
Resolving download.splunk.com (download.splunk.com)... 3.167.37.33, 3.167.37.124, 3.167.37.9, ...
Connecting to download.splunk.com (download.splunk.com)|3.167.37.33|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 993786302 (948M) [application/x-tar]
Saving to: 'splunk-9.3.2-d8bb32809498-Linux-x86_64.tgz'
splunk-9.3.2-d8bb32 100%[==========>] 947.75M 117MB/s in 8.2s
2024-11-22 14:31:53 (116 MB/s) - 'splunk-9.3.2-d8bb32809498-Linux-x86_64.tgz' saved [993786302/993786302]
root@ip-10-15-8-193:/tmp#
File Edit View
Or Sasson
[redacted]
Ln 2, Col 10 19 characters 100%


splunk/etc/manager-apps/_cluster/local/
splunk/etc/manager-apps/_cluster/local/README
root@ip-10-15-8-193:/opt# ls
splunk
root@ip-10-15-8-193:/opt# cd /splunk
-bash: cd: /splunk: No such file or directory
root@ip-10-15-8-193:/opt# cd splunk/
root@ip-10-15-8-193:/opt/splunk# ls
LICENSE.txt bin etc include license-eula.txt opt
README-splunk.txt copyright.txt ftr lib openssl quarantined_fi
root@ip-10-15-8-193:/opt/splunk#
Or Sasson
100666818
Ln 2, Col 10 19 characters

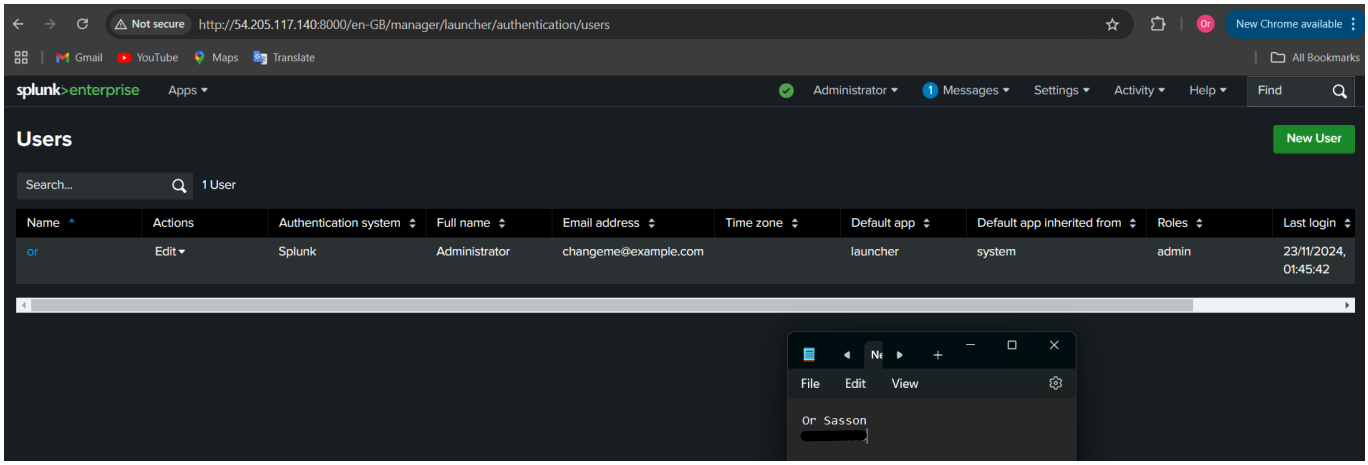

ec2-54-205-117-140.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help X server Exit
Quick connect...
/home/ubuntu/
.cache
.ssh
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.Xauthority
Remote monitoring
Follow terminal folder
writing new private key to 'privKeySecure.pem'

Signature ok
subject=/CN=ip-10-15-8-193/O=SplunkUser
Getting CA Private Key
writing RSA key
PYTHONHTTPSVERIFY is set to 0 in splunk-launch.conf disabling certificate validation for the httplib and urllib libraries shipped with the embedded Python interpreter; must be set to "1" for increased security
Done
Waiting for web server at http://127.0.0.1:8000 to be available...
Done
If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com
The Splunk web interface is at http://ip-10-15-8-193:8000
root@ip-10-15-8-193:/opt/splunk#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
File Edit
Or Sasson
[redacted]
Ln 2, Col 10



Q2.3 Provide screenshot of Splunk Users.

#Screenshot



Part-3 Working with Security Logs

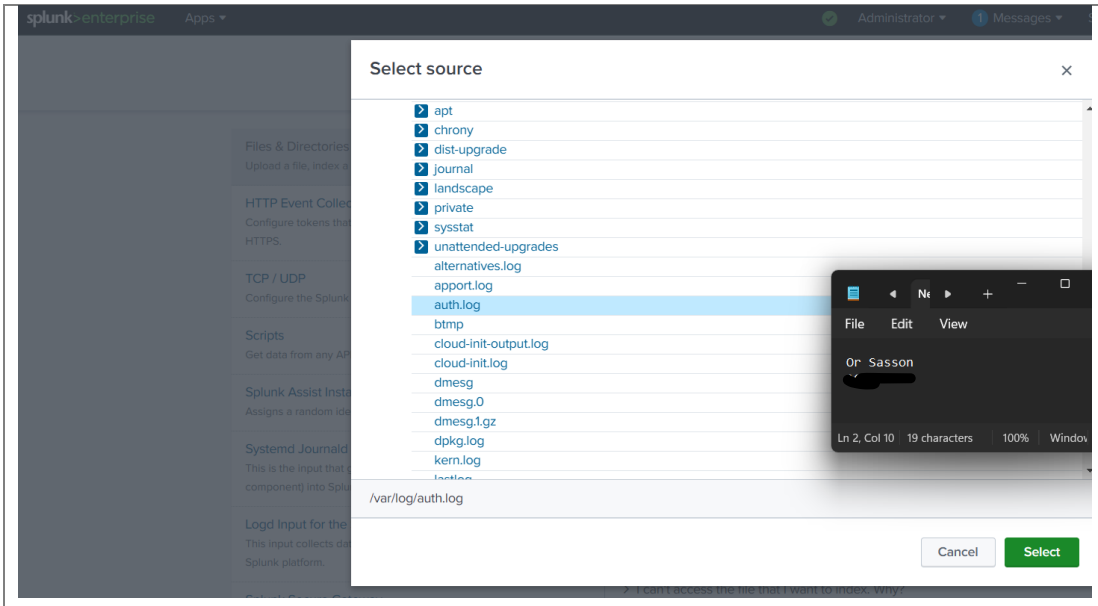
In this task, you will onboard the auth.log file from your Ubuntu VM into Splunk. In Ubuntu, this file is located at /var/log/auth.log - if you are using a different distribution this file may be located in a different location or have a different name.

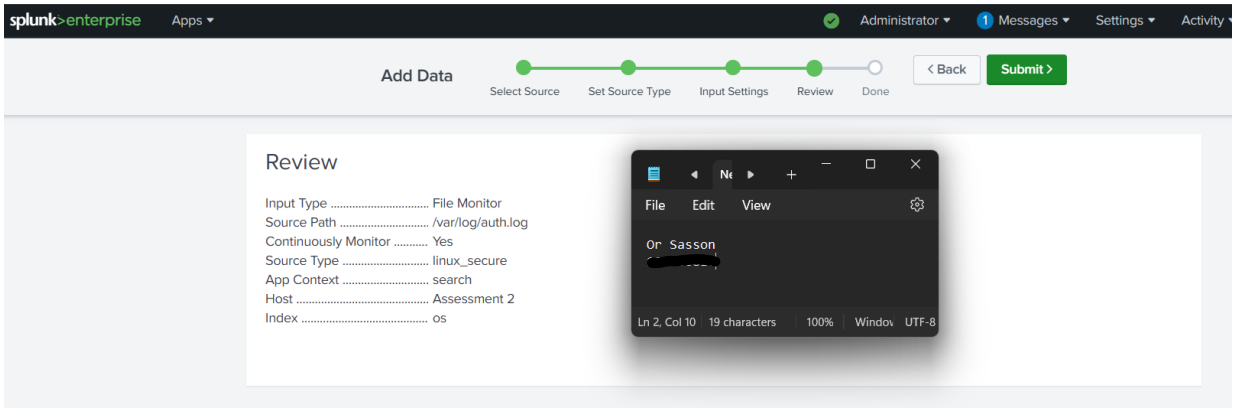
- Create an index named os

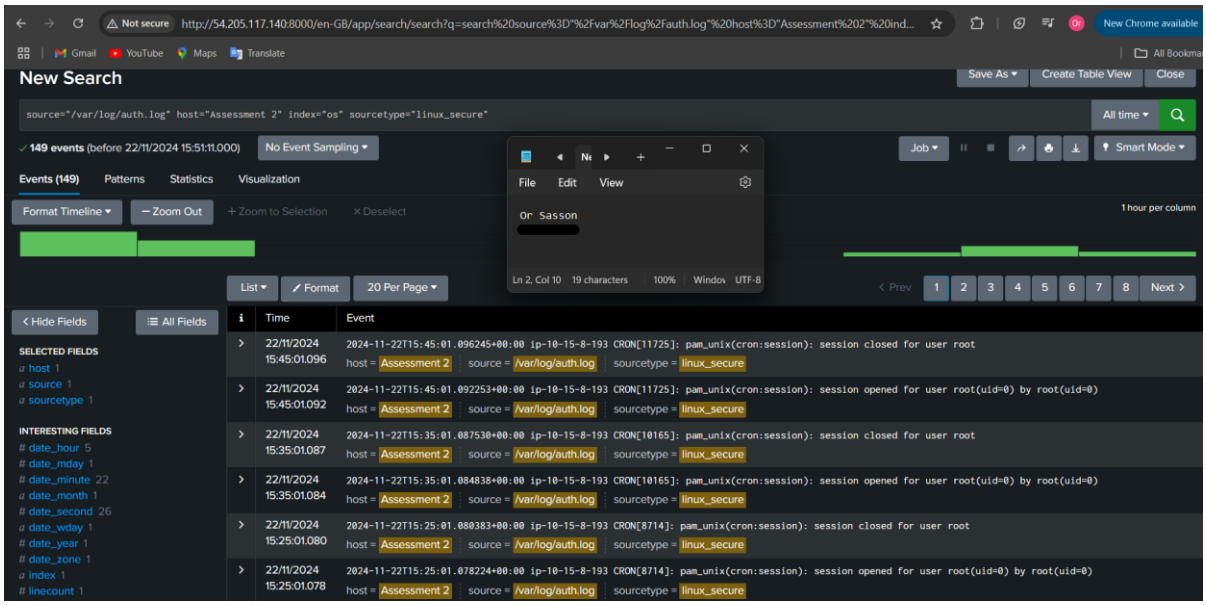
-Using the Add Data wizard, create a configuration to continuously monitor the /var/log/auth.log file. Use the linux_secure sourctype, and store this data in the os index.

Q3.1. Provide Screenshots of your first of the data you just onboard. How many events do you have?

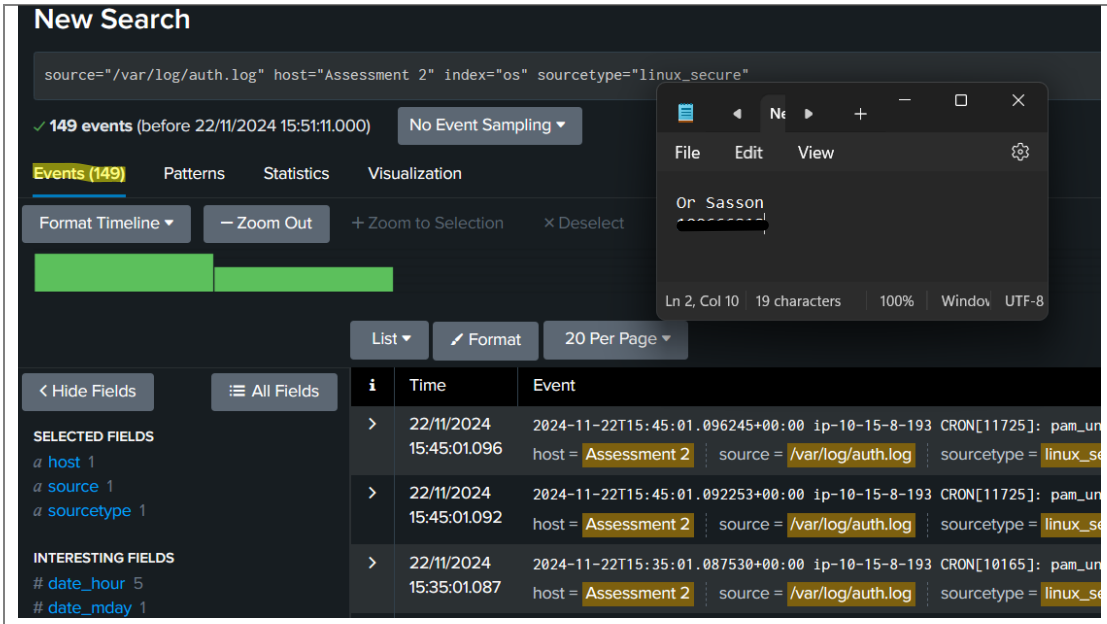
#Screenshot







#Your response
No. of events : 149



Q3.2. What do you notice about the fields in this newly-onboarded data?

#Your response:

I noticed that the number of logs are increasing regularly.

Part-4 Create a Report and Dashboard

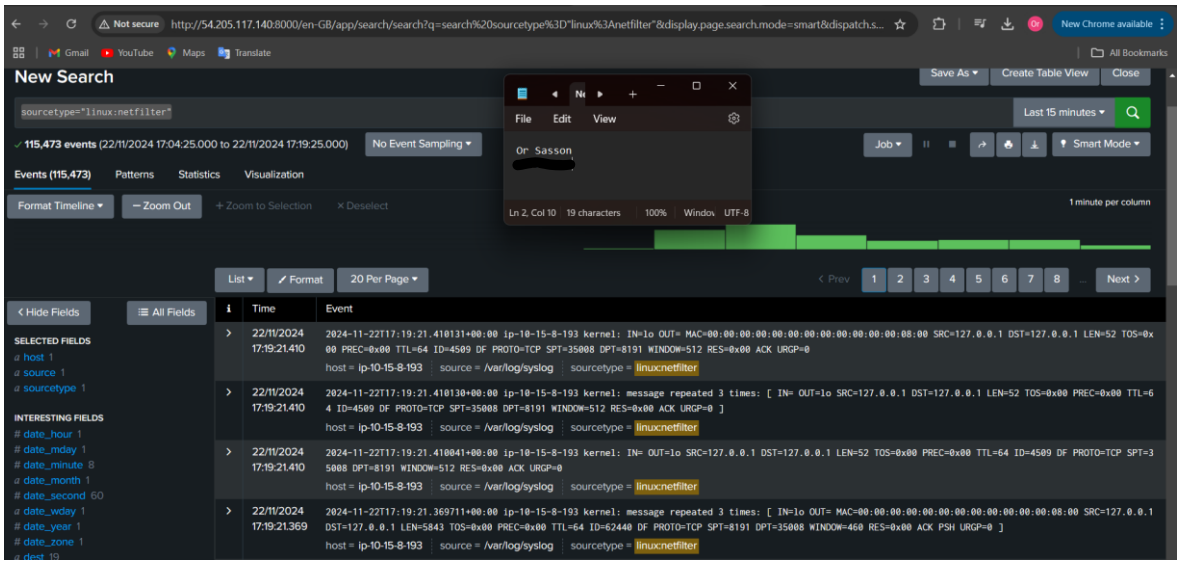
Create a search showing the top TCP/UDP ports passing through the firewall from your machine over the past 15 minutes:

- Experiment with different visualizations and limits on the search
- Save this search as a report
- Save As -> Report
- Title: <Your Name> Top Services

Q.4.1 What search did you develop for your report?

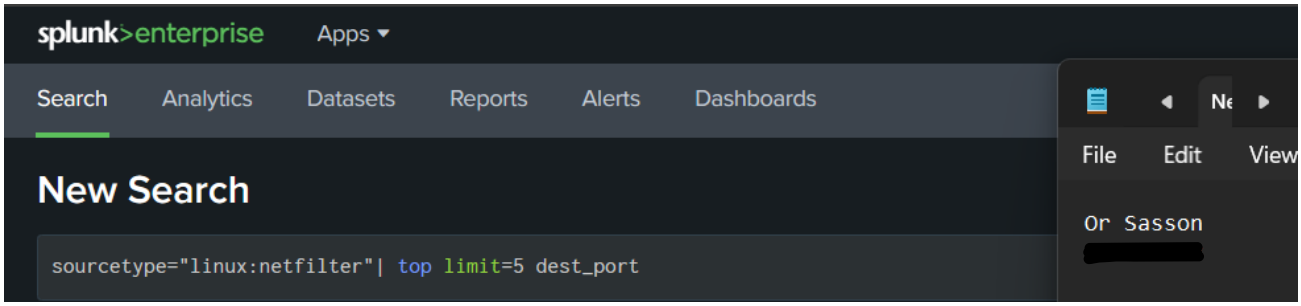
#Screenshot

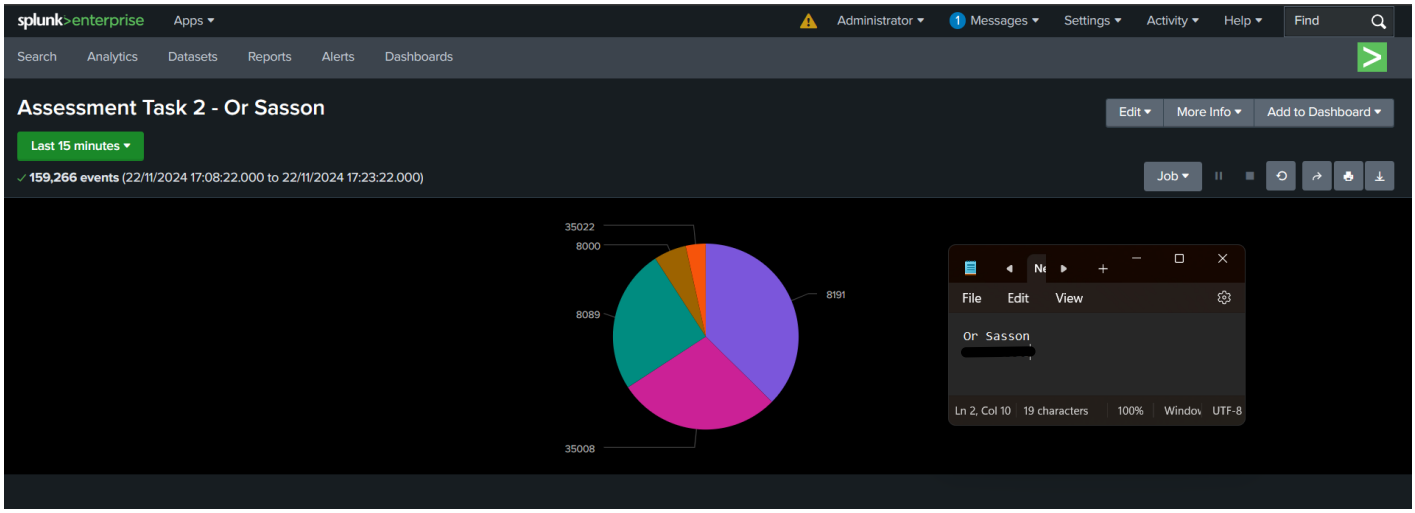
sourcetype="linux:netfilter" from the last 15 minutes



Q.4.2 Provide a screenshot of your saved report

#Screenshot

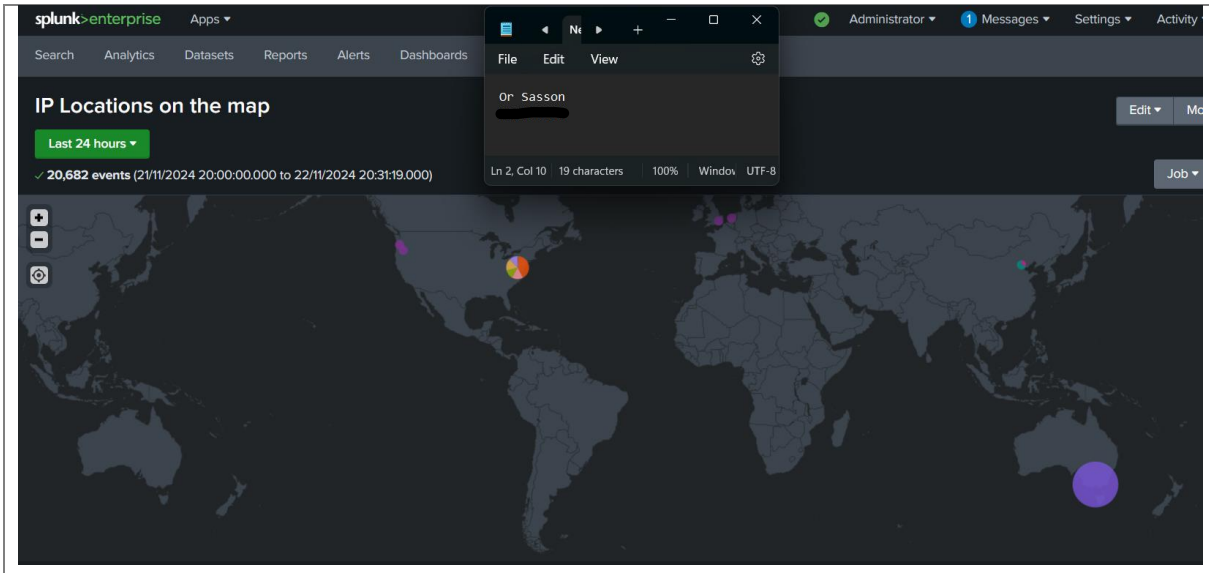


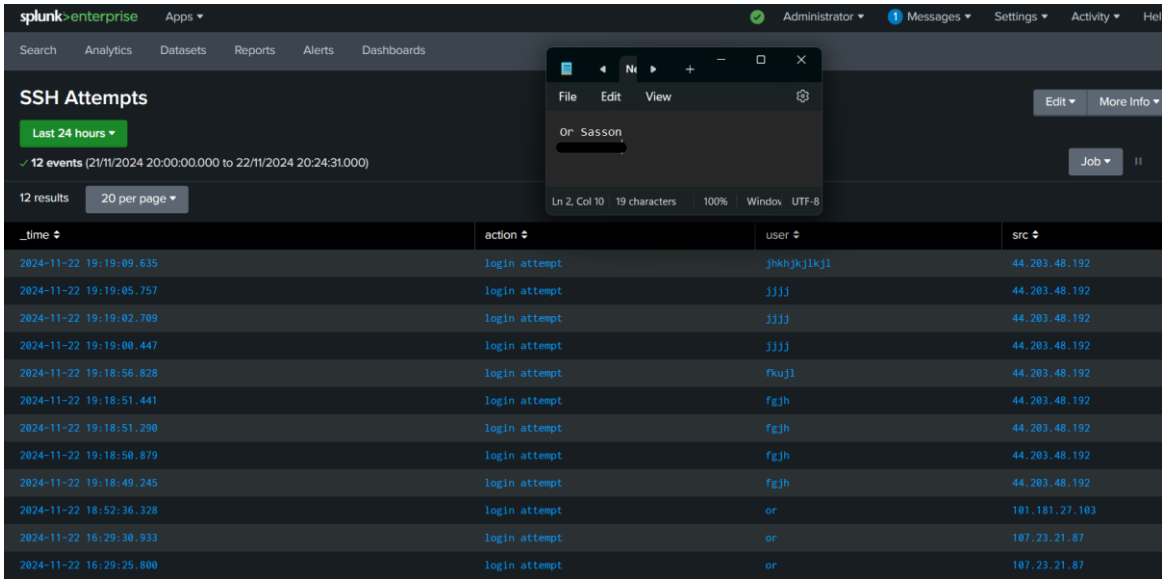


Build a dashboard using the report created earlier
Create new reports using the firewall and auth logs and add them to the dashboard
Adjust the look and feel of the dashboard
Re-arrange panels
Rename panels

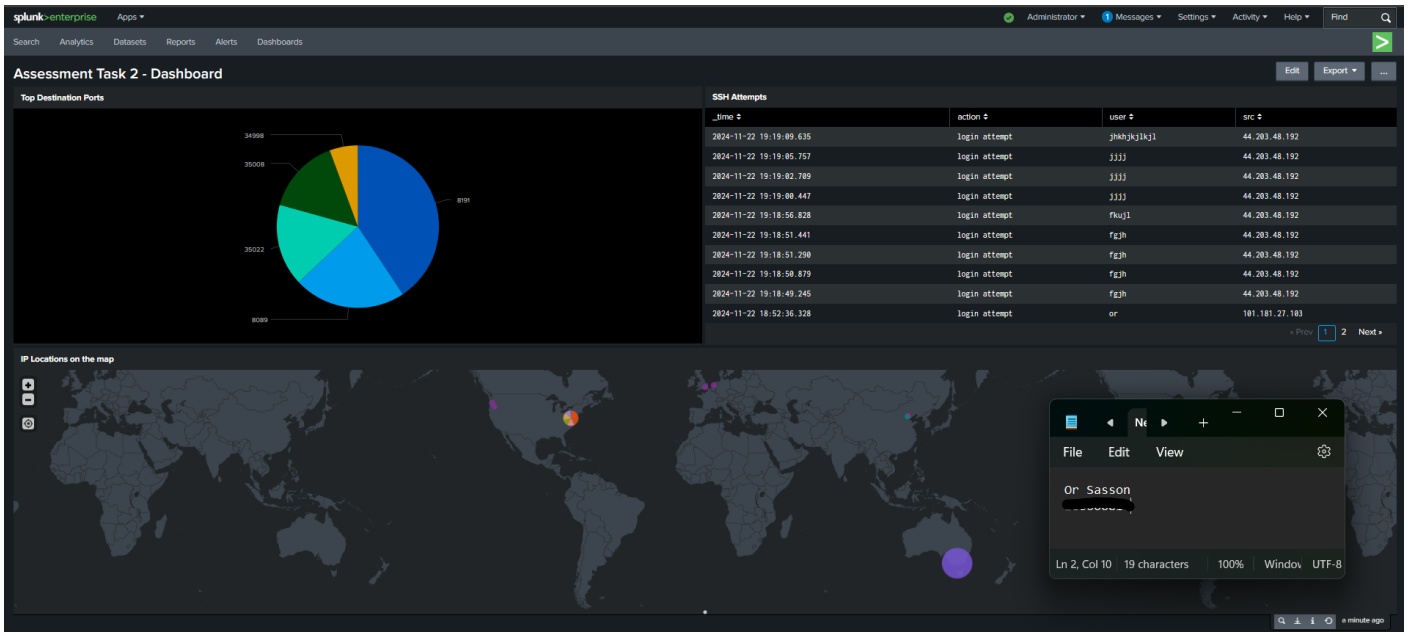
Q4.3 Once you have built your dashboard and are happy with the design, provide a screenshot of your dashboard:

Screenshot





My dashboard:



Part-5: Strategies to detect data patterns using Python

Q.5.1 Which python module can be used to read and match patterns in the log file?

#Your response

The re module.

It is a library with strong patter machine capabilities which can assist with the extraction of data from strings.

Q5.2. Identify the line which is used to match pattern in the following python code?

```
1  import re
2
3  def analyze_log(log_file):
4      with open(log_file, 'r') as file:
5          log_data = file.readlines()
6
7      # Count the number of log entries
8      num_entries = len(log_data)
9      print(f"Number of log entries: {num_entries}")
10
11     # Find and print all error messages
12     error_pattern = r"\bERROR\b"
13     error_messages = [line.strip() for line in log_data if re.search(error_pattern, line)]
14     print("Error messages:")
15     for i, message in enumerate(error_messages, start=1):
16         print(f"Line {i}: {message}")
17
18     # Find and print all warning messages
19     warning_pattern = r"\bWARNING\b"
20     warning_messages = [line.strip() for line in log_data if re.search(warning_pattern, line)]
21     print("Warning messages:")
22     for i, message in enumerate(warning_messages, start=1):
23         print(f"Line {i}: {message}")
24
25     # Find and print all suspicious IP addresses
26     ip_pattern = r"\b(?:\d{1,3}\.){3}\d{1,3}\b"
27     ip_addresses = [re.search(ip_pattern, line).group() for line in log_data if re.search(ip_pattern, line)]
28     suspicious_ips = set()
29     for ip in ip_addresses:
30         # Add your own conditions for determining suspicious IP addresses
31         if ip.startswith("192.168.") or ip.startswith("10.") or ip == "127.0.0.1":
32             suspicious_ips.add(ip)
33     print("Suspicious IP addresses:")
34     for ip in suspicious_ips:
35         matching_lines = [i+1 for i, line in enumerate(log_data) if ip in line]
36         print(f"IP: {ip}, Lines: {matching_lines}")
37
38 # Provide the path to the syslog file
39 log_file_path = '/var/log/syslog'
40 analyze_log(log_file_path)
```

#Your response:

Line 26 with the IP pattern for searching digits that are 1-3 in length, separated by periods to identify the IP addresses.

There are also two more patterns for error and warning on line 12 and 19.

It checks the patterns in the syslog file.

Q5.3 Can you change the above script and add your own matching pattern for any logfile using re python module. Provide the screenshot of your script.

#Your response:

```python
import re

log = """
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: rsyslog.service: Sent signal SIGHUP to main process 382 (rsyslogd) on client request.
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: e2scrub_reap.service: Deactivated successfully.
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: Finished Remove Stale Online ext4 Metadata Check Snapshots.
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: Started Dispatcher daemon for systemd-networkd.
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: logrotate.service: Deactivated successfully.
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: Finished Rotate log files.
Sep  5 04:33:42 ip-10-15-12-140 systemd-hostnamed[407]: Hostname set to <ip-10-15-12-140> (static)
Sep  5 04:33:42 ip-10-15-12-140 systemd[1]: ec2-instance-connect.service: Deactivated successfully.
Sep  5 04:33:42 ip-10-15-12-150 systemd[1]: Finished EC2 Instance Connect Host Key Harvesting.
Sep  5 04:33:42 ip-10-15-12-160 systemd[1]: Starting OpenBSD Secure Shell server...
"""

# Regular expression to match IP addresses in the format ip-10-15-12-140
pattern = r"ip-(\d{1,3}-\d{1,3}-\d{1,3}-\d{1,3})"

# Find all matches
ip_addresses = re.findall(pattern, log)

# Convert the found format to standard dotted decimal notation
ip_addresses = [ip.replace('-', '.') for ip in ip_addresses]

# Remove duplicates by converting the list to a set and then back to a list
unique_ips = list(set(ip_addresses))

print(unique_ips)
```