**LAB - 6 REPORT**

Course Code - Course Name: - COMP4040 – Introduction to SOC

Program: T433 - Cybersecurity

Term: - Winter 2024

Student Names - ID:

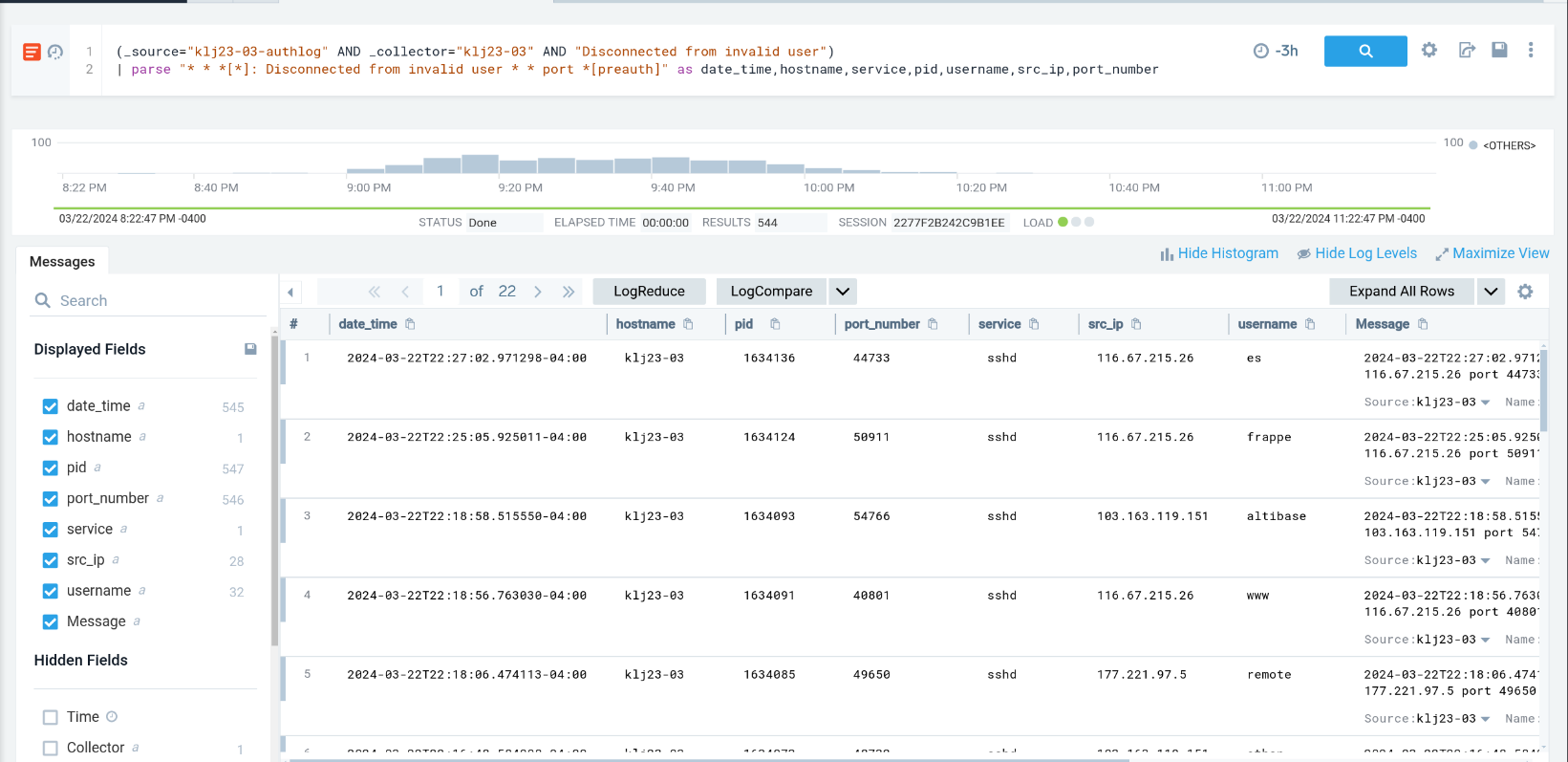
* Prabhjot Singh Sains – 101495218

1. **Try to brute force to the server cslab.softether.net via ssh port 8922 (or try to login with fake usernames including your nickname) then use (\_source="klj23-03-authlog" and \_collector="klj23-03") [3h]**

(\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "Disconnected from invalid user")

| parse "\* \* \*[\*]: Disconnected from invalid user \* \* port \*[preauth]" as date\_time,hostname,service,pid,username,src\_ip,port\_number

This is the query to get LINUX logs for the server cslab.softether.net from a source named klj23-03-authlog and a collector named klj23-03 for the past 3 hours. A filter is also been used to only get the logs of "Disconnected from invalid user". After this parsing was done on the logs to get components like date and time, hostname, service, Process Identifier (PID), username, source IP, and port number.



* 1. **Panel-1: Create a table view of the top 10 most tried usernames.**

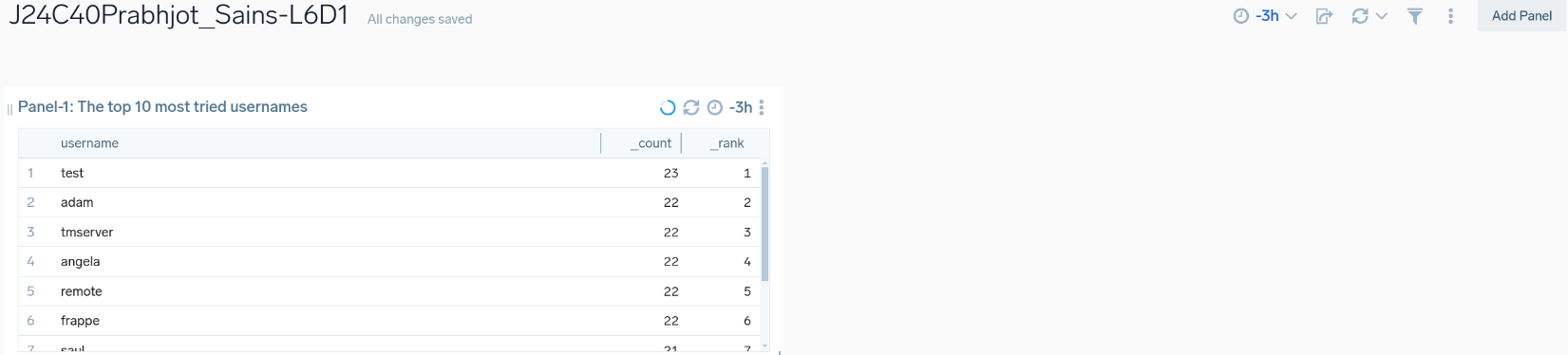
(\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "Disconnected from invalid user")

| parse "\* \* \*[\*]: Disconnected from invalid user \* \* port \*[preauth]" as date\_time,hostname,service,pid,username,src\_ip,port\_number

| count by username | topk(10, \_count)

The query “| count by username | topk(10, \_count)” is added to the query from “1.”, Due to this addition the data for the top 10 usernames which is tried on the server is shown and added to the dashboard in panel 1.





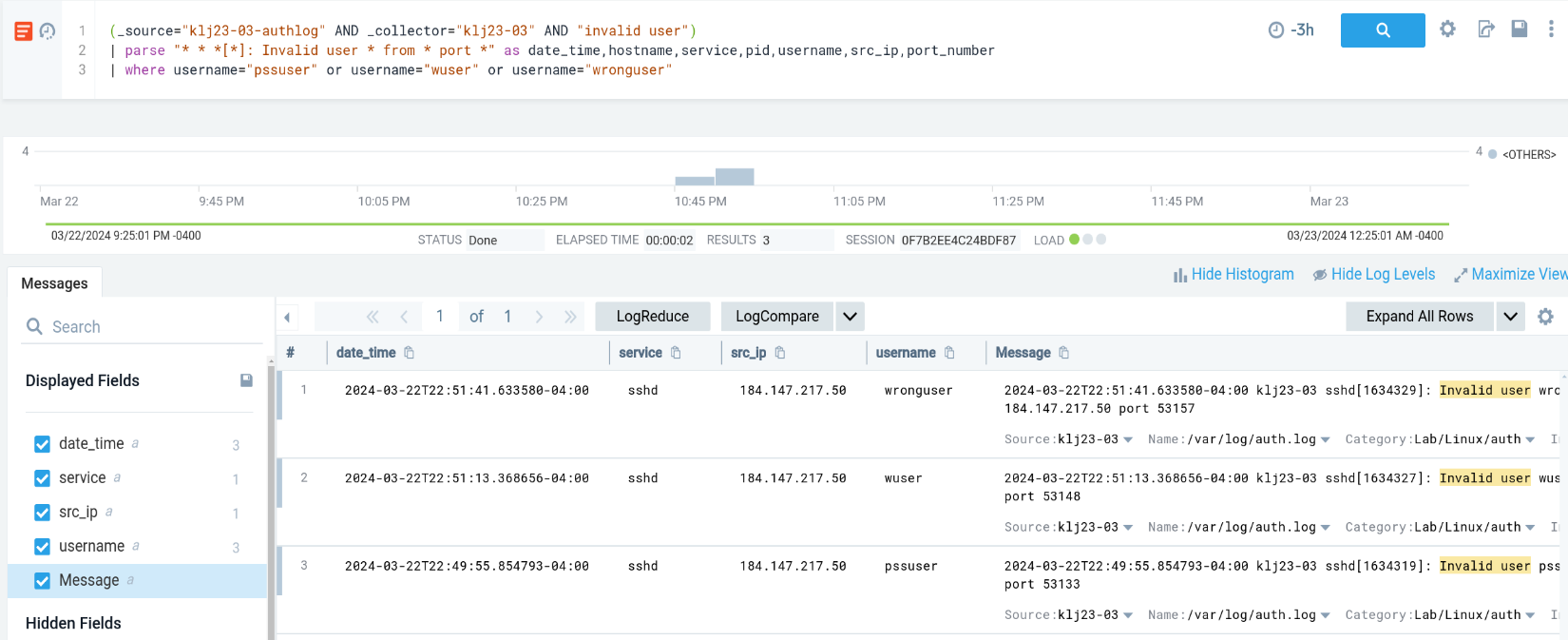
* 1. **Panel-2: Filter and find your tried usernames.**

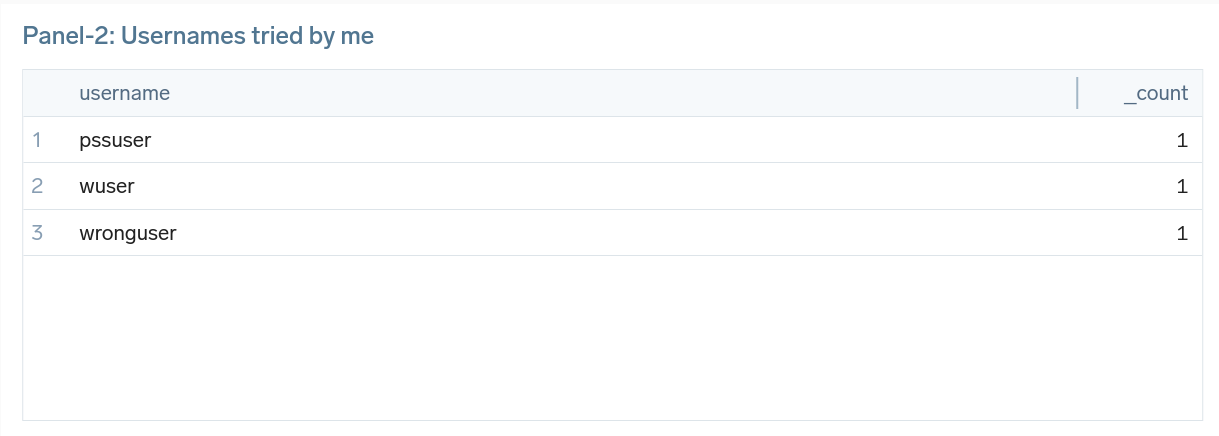
(\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "invalid user")

| parse "\* \* \*[\*]: Invalid user \* from \* port \*" as date\_time,hostname,service,pid,username,src\_ip,port\_number

| where username="pssuser" or username="wuser" or username="wronguser" | count by username

This is the query to get LINUX logs for the server cslab.softether.net from a source named klj23-03-authlog and a collector named klj23-03 for the past 3 hours. A filter is also been used to only get the logs of " invalid users". After this parsing was done on the logs to get components like date and time, hostname, service, Process Identifier (PID), username, source IP, and port number. After Parsing a where query is used to find specific usernames tried by me on the server. The output of this query is then added to the dashboard as panel 2





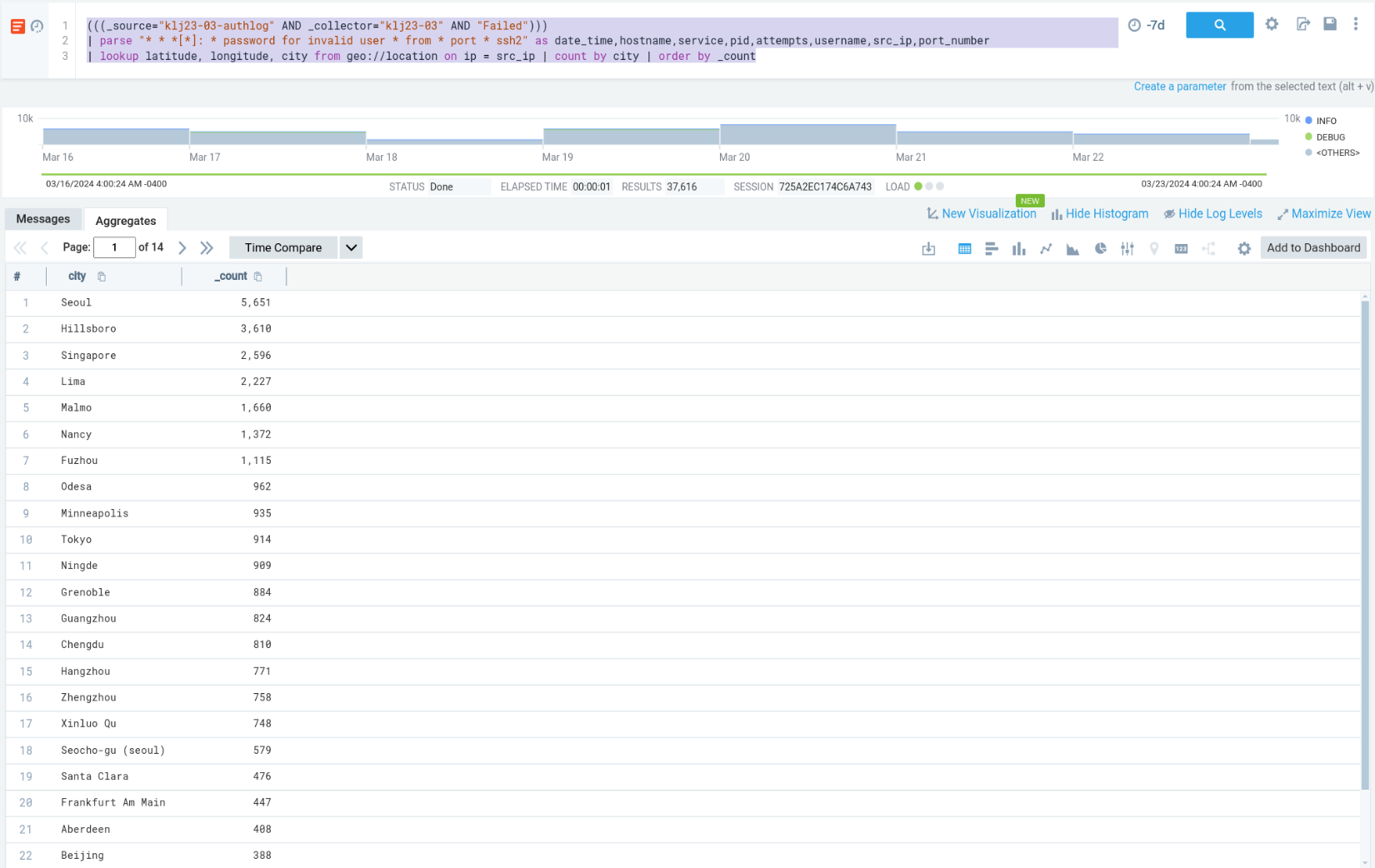
* 1. **Panel-3: Create a map to show failed users' locations.**

(((\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "Failed")))

| parse "\* \* \*[\*]: \* password for invalid user \* from \* port \* ssh2" as date\_time,hostname,service,pid,attempts,username,src\_ip,port\_number

| lookup latitude, longitude, city from geo://location on ip = src\_ip | count by city | order by \_count

This is the query to get LINUX logs to get the location of failed users. After this parsing was done on the logs to get components like date and time, hostname, service, Process Identifier (PID), attempts for login, username, source IP, and port number. After Parsing a query is used to get the geographic locations of the source IPs using latitude and longitude and display the cities. The count query is used to get the number of users in a city and then order them. The output of this query is then added to the dashboard as a map in panel 3.





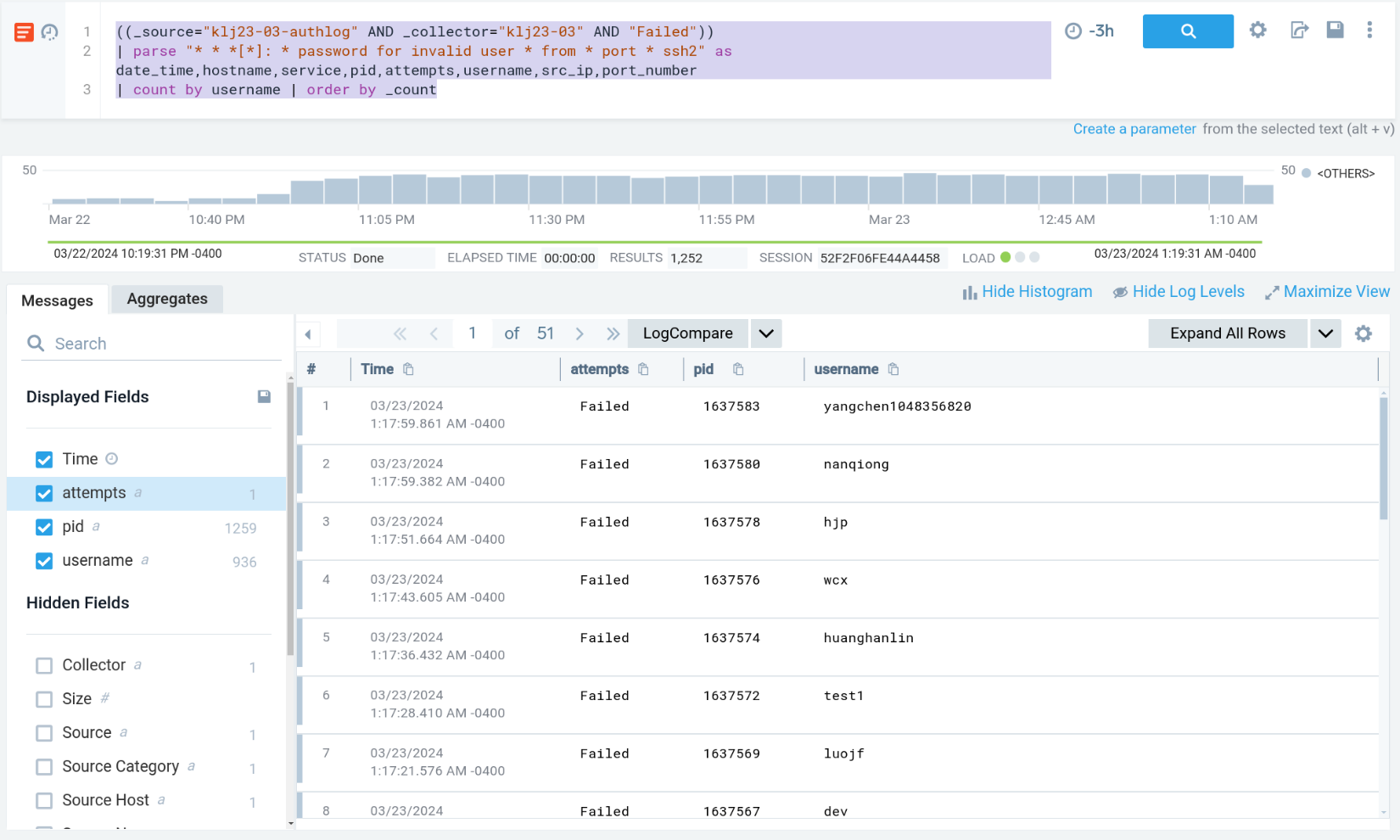
* 1. **Panel-4: Create a list of failed users.**

((\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "Failed"))

| parse "\* \* \*[\*]: \* password for invalid user \* from \* port \* ssh2" as date\_time,hostname,service,pid,attempts,username,src\_ip,port\_number

| count by username | order by \_count

This is the query to get LINUX logs for the server cslab.softether.net from a source named klj23-03-authlog and a collector named klj23-03 for the past 3 hours. A filter is also been used to only get the logs of " failed" users. After this parsing was done on the logs to get components like date and time, hostname, service, Process Identifier (PID), attempts for login,username, source IP, and port number. After Parsing a query is used to get the list of failed users. The output of this query is then added to the dashboard in panel 4.



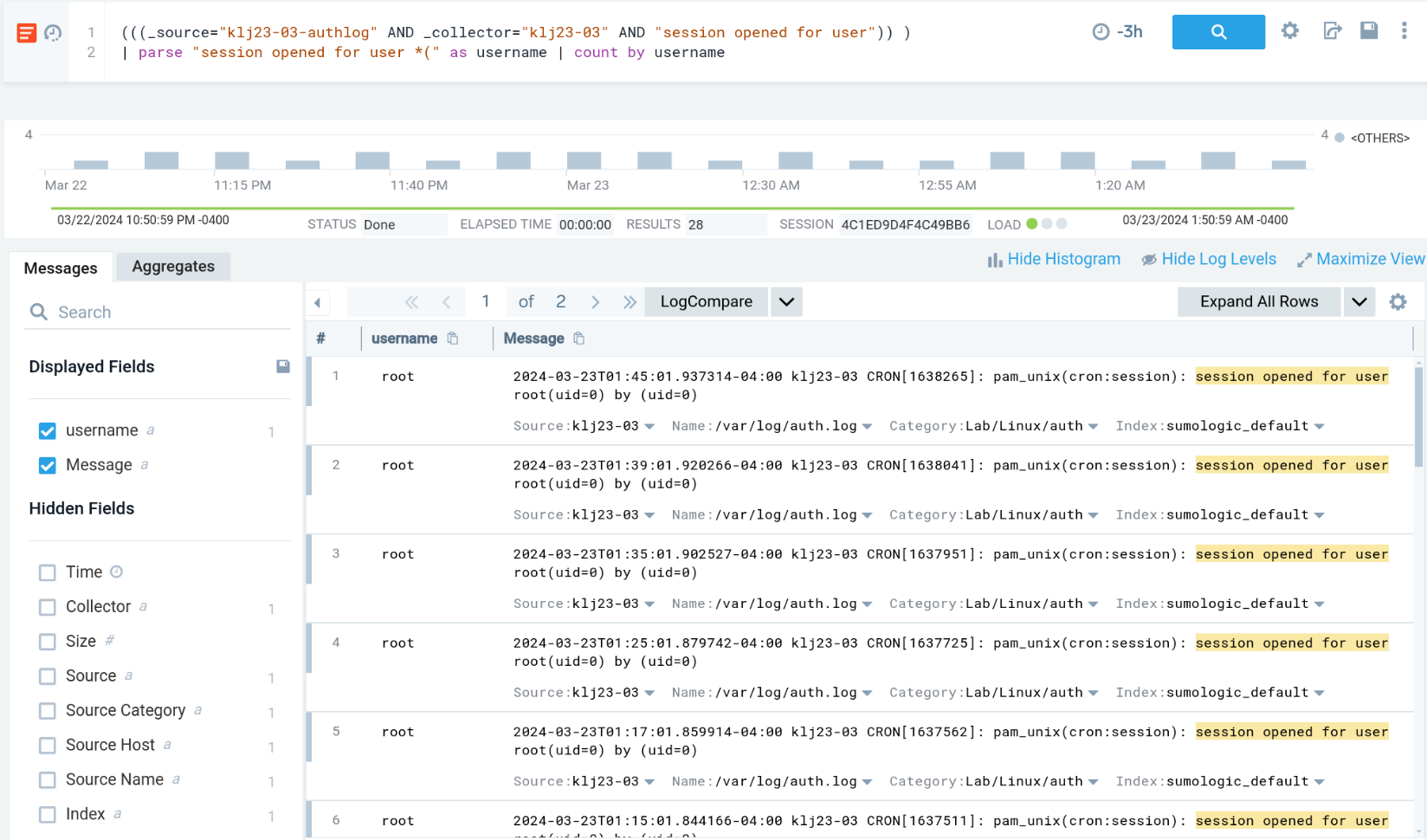


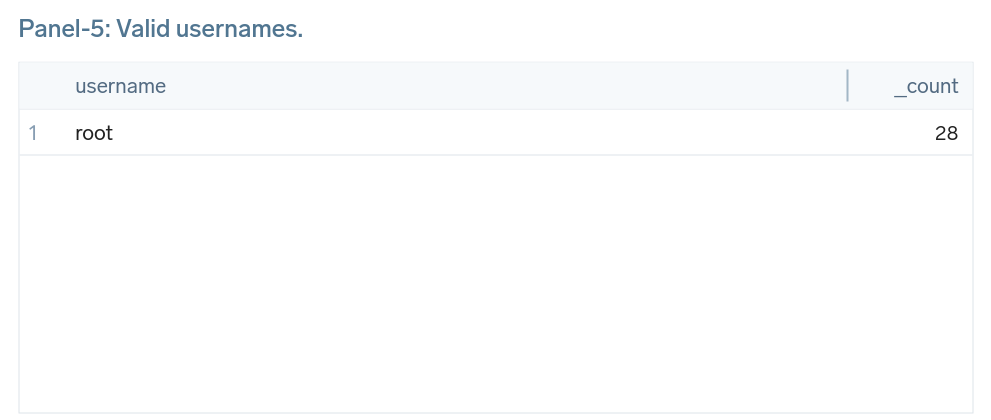
* 1. **Panel-5: Find the valid usernames.**

(((\_source="klj23-03-authlog" AND \_collector="klj23-03" AND "session opened for user")))

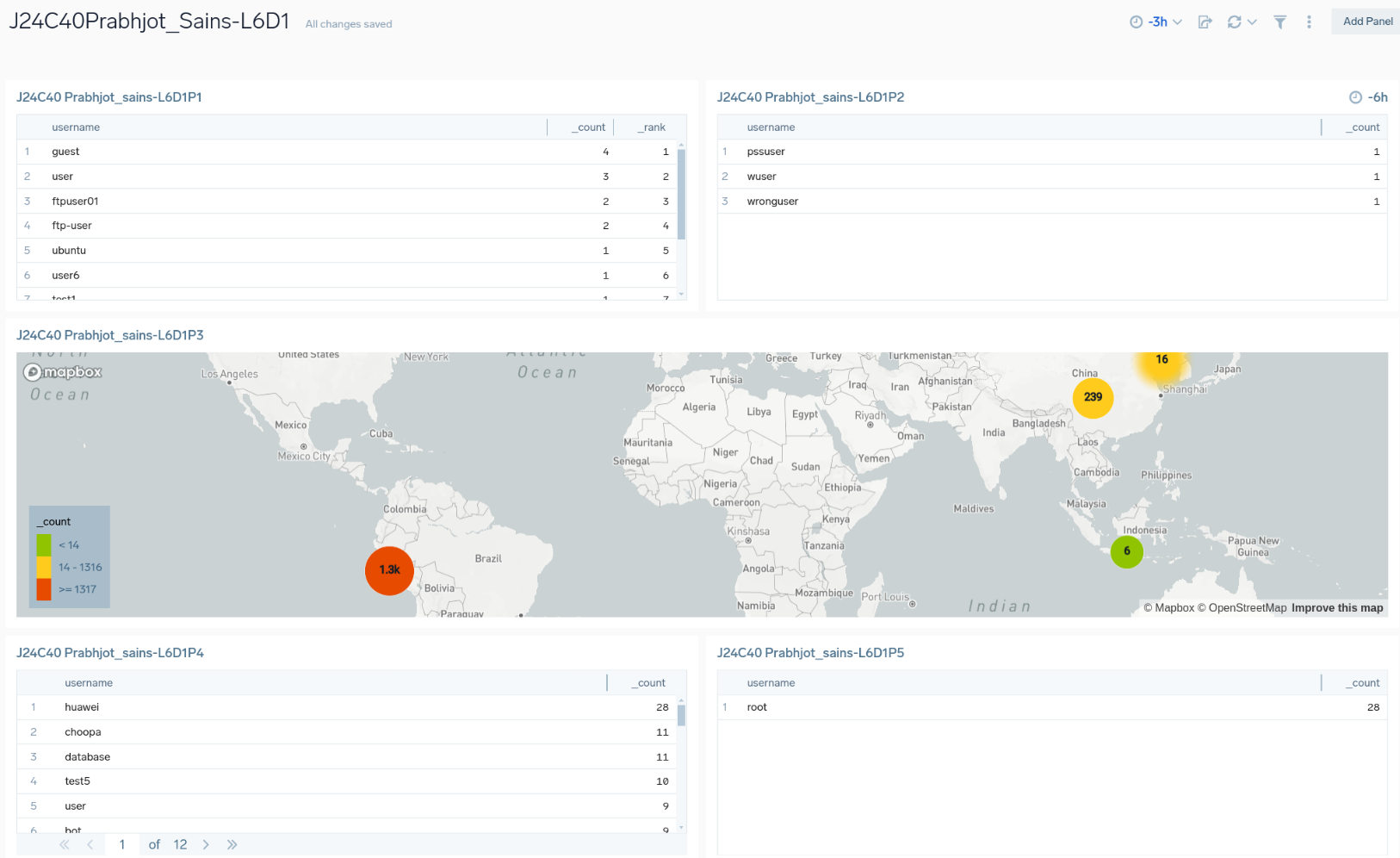
| parse "session opened for user \*(" as username | count by username

This is the query to get LINUX logs for the server cslab.softether.net from a source named klj23-03-authlog and a collector named klj23-03 for the past 3 hours. A filter is also been used to only get the logs of "session opened for user “, this is for filtering valid users. After this parsing was done on the logs to get the usernames of the users. After Parsing users are grouped by using count. The output of this query is then added to the dashboard in panel 4.





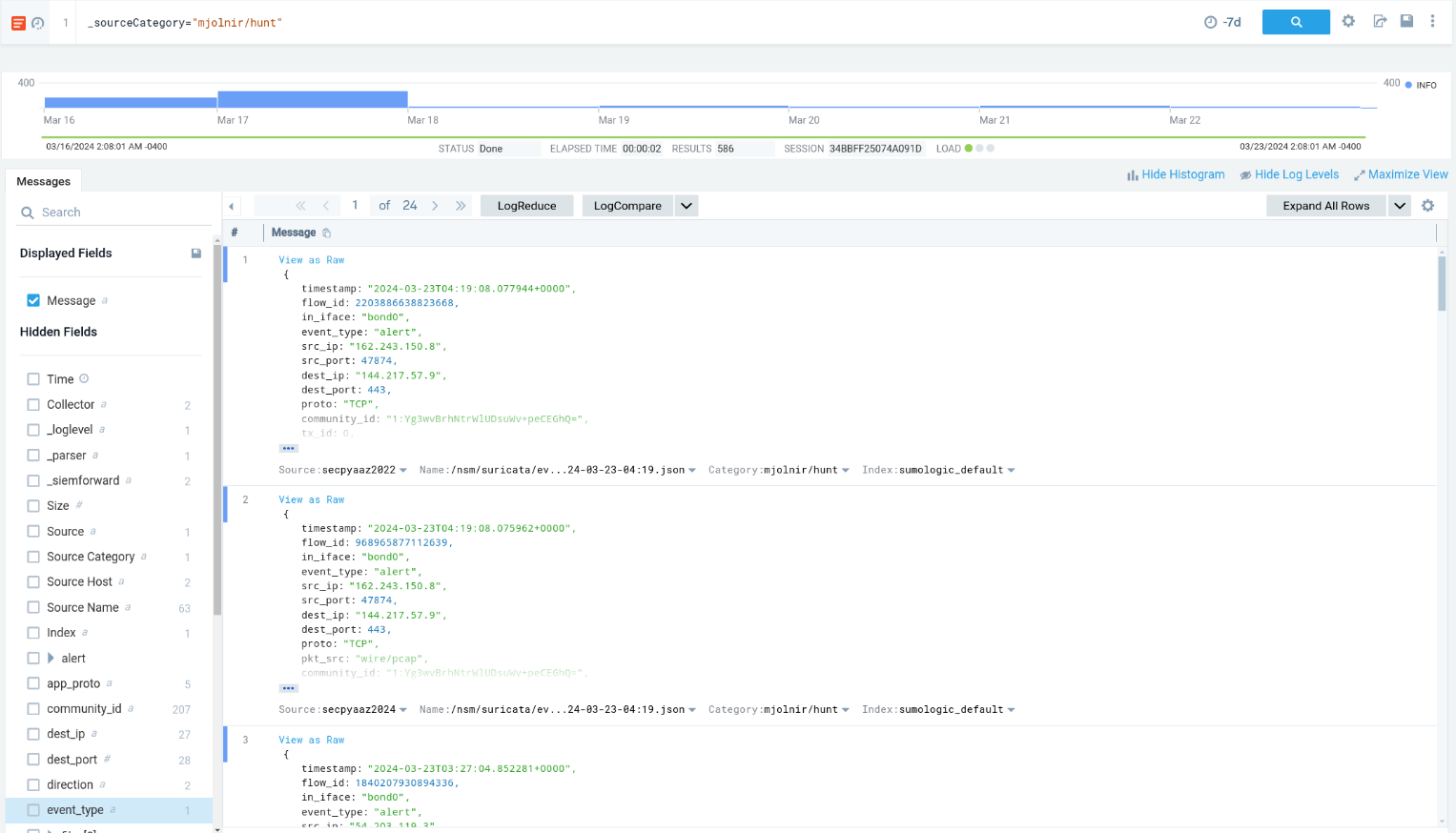
* 1. **Create a dashboard and add these 5 Panels to it.**



* 1. **Analyze the result of each dashboard.**
     1. **Panel-1:** It shows the top 10 most tried usernames on the server. As shown user named test have the most login attempts of 23.
     2. **Panel-2:** It shows the usernames that I tried with the wrong passwords.
     3. **Panel-3:** It shows a map of users who are not able to login to the system and have failed login attempts on the system.
     4. **Panel-4:** It shows the list of users who failed to get into the system.
     5. **Panel-5:** It shows the valid user who logged into the system named root.

1. **Use (\_sourceCategory="mjolnir/hunt") for the last 7d**

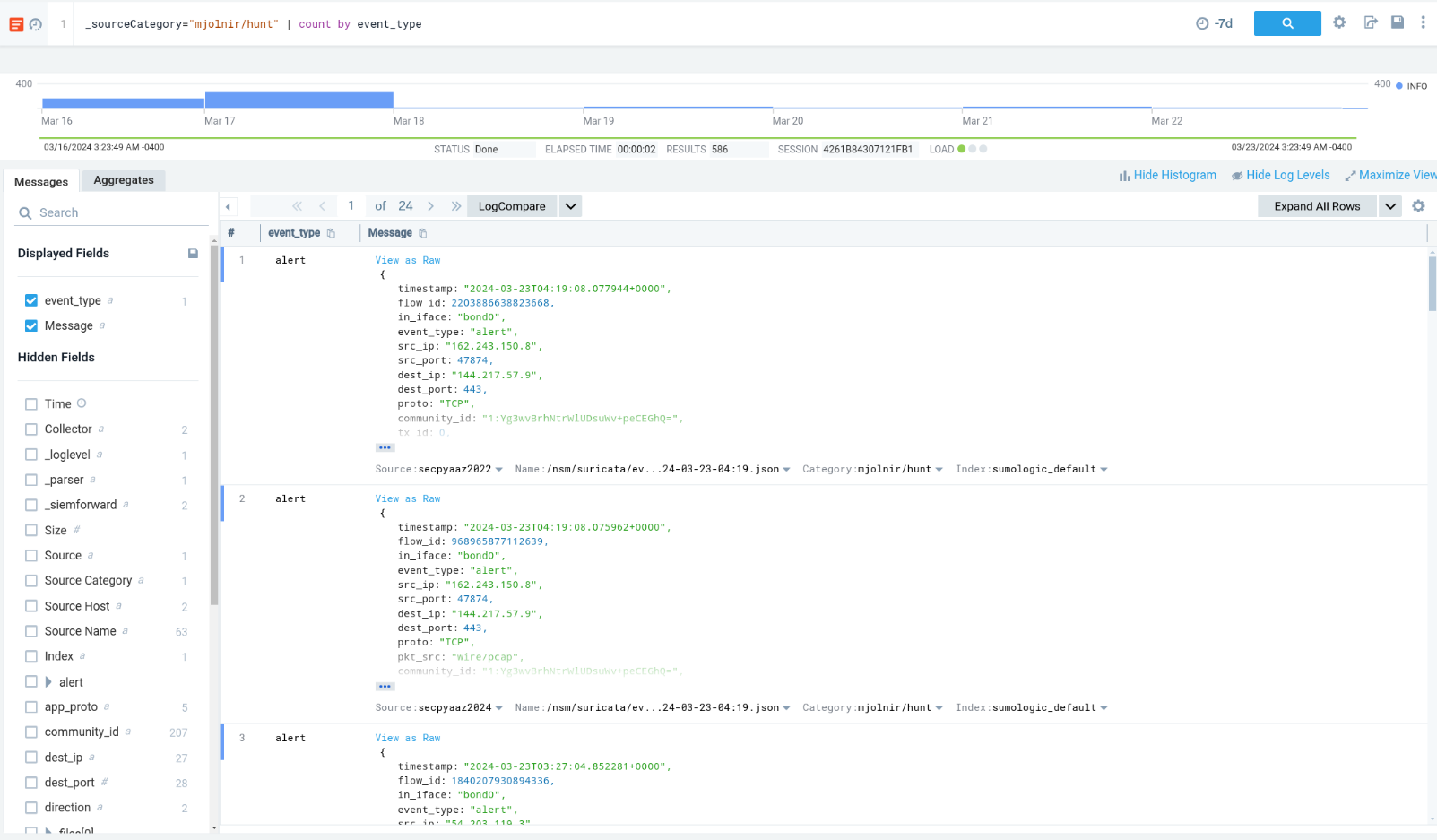
A query to get logs from the source with a category named “mjolnir/hunt” for the past seven days.



* 1. **Panel-1: Create a Pi chart based on the event type**

\_sourceCategory="mjolnir/hunt" | count by event\_type

This query is used to get the logs from a source category called ="mjolnir/hunt" and then event\_type grouped by using the count. The output of this query is added to the dashboard in panel 1 as a pie chart.



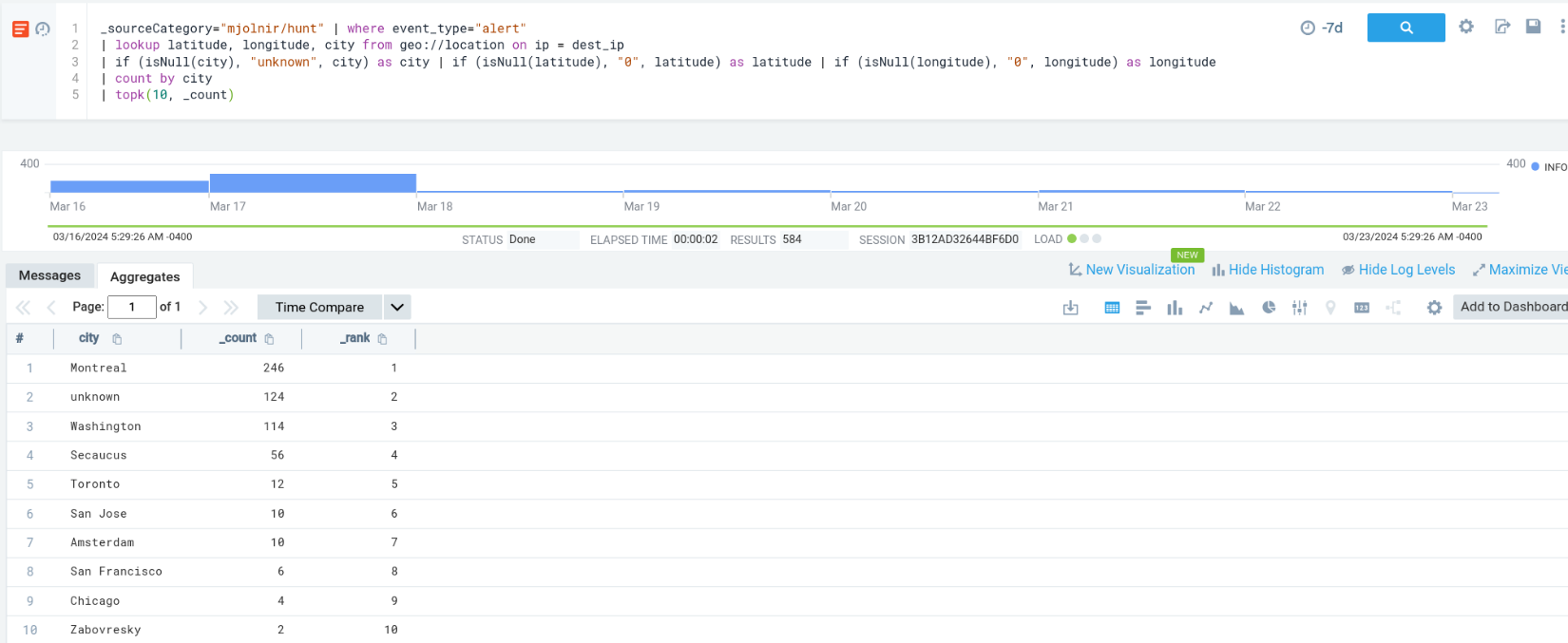


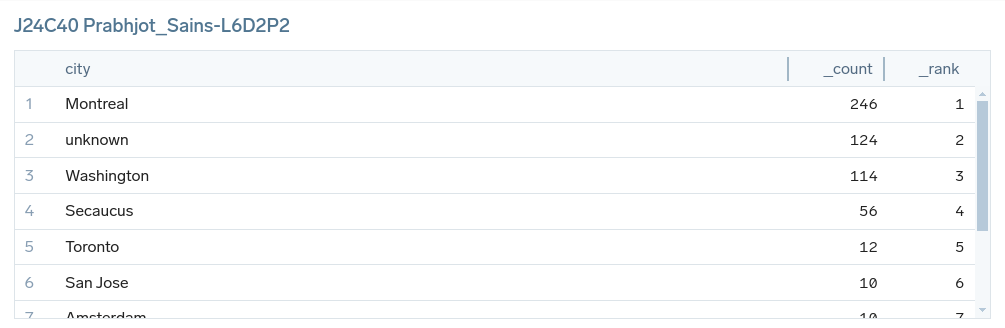
* 1. **Panel-2: Create a table for the top 10 destinations under attack (event\_type = "alert")**

\_sourceCategory="mjolnir/hunt" | where event\_type="alert"

| lookup latitude, longitude, city from geo://location on ip = dest\_ip | count by city | topk(10, \_count)

This is the query to get the logs to get the location for the top 10 destinations under attack. A query is used to get the geographic locations based on the destination IP using latitude and longitude and display the cities. The count query is used to get the number of users in a city and then order them. The output of this query is added to the dashboard in panel 2 as a list.





* 1. **Panel-3: Visualize on a map source of attack (event\_type = "alert")**

\_sourceCategory="mjolnir/hunt"

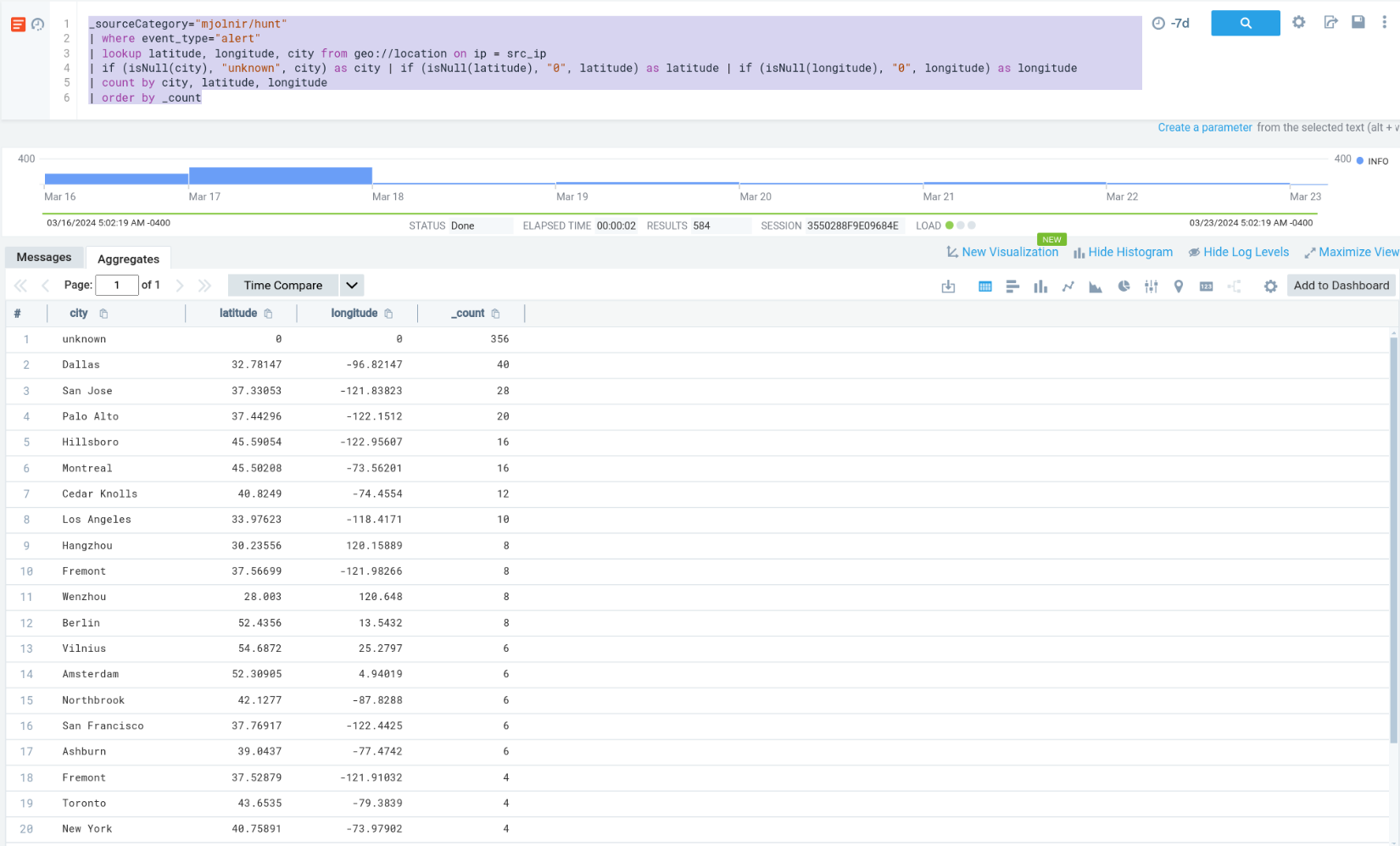
| where event\_type="alert"

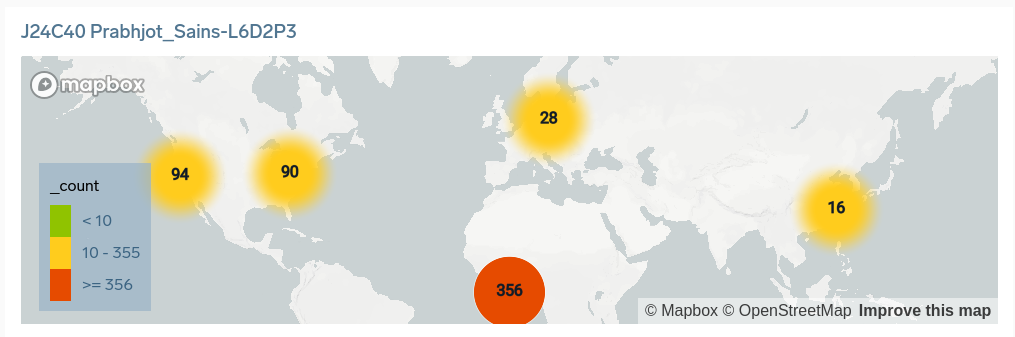
| lookup latitude, longitude, city from geo://location on ip = src\_ip

| if (isNull(city), "unknown", city) as city | if (isNull(latitude), "0", latitude) as latitude | if (isNull(longitude), "0", longitude) as longitude

| count by city, latitude, longitude | order by \_count

This is the query to get the logs to get the location for the top 10 destinations under attack. A query is used to get the geographic locations based on the destination IP using latitude and longitude and display the cities. The isNull query is used because there were logs that do not have any latitude and longitude which is why there can be no city, do we just give them values as unknown and 0. The count query is used to get the number of users in a city and then order them. The output of this query is added to the dashboard in panel 3 as a map.



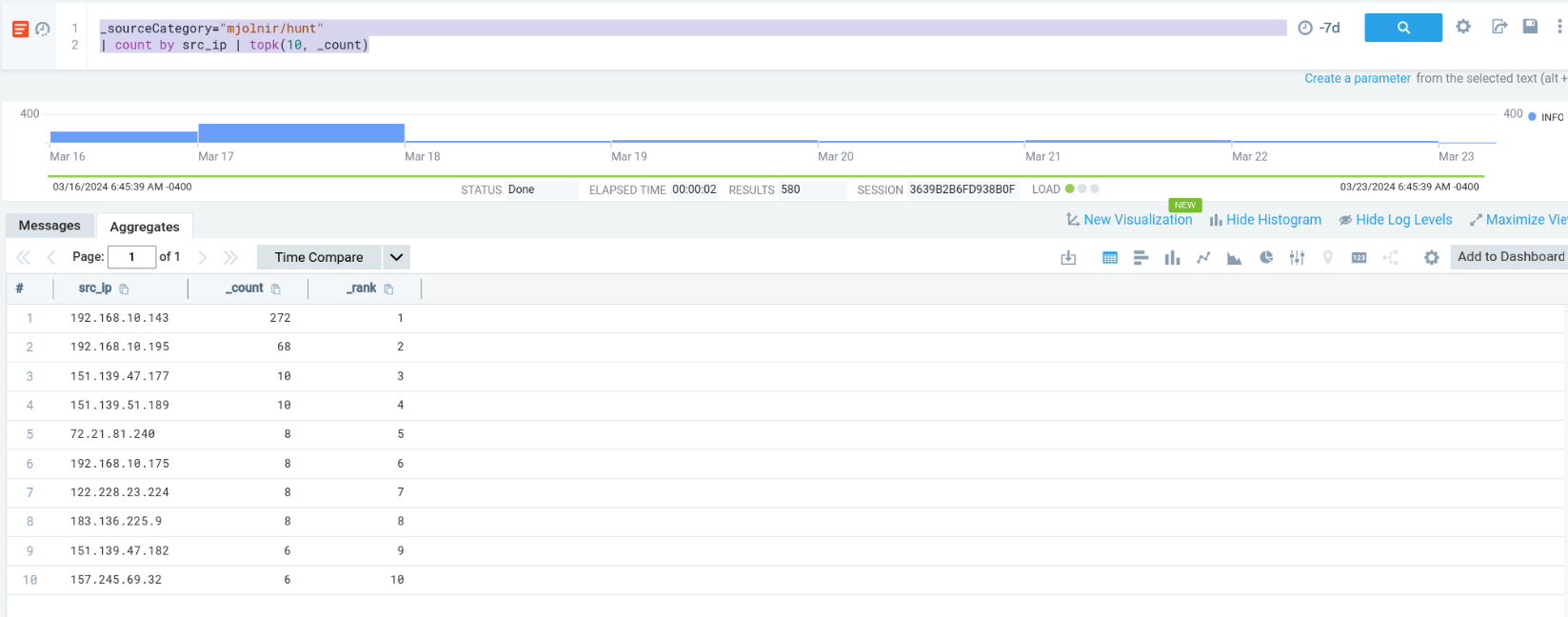


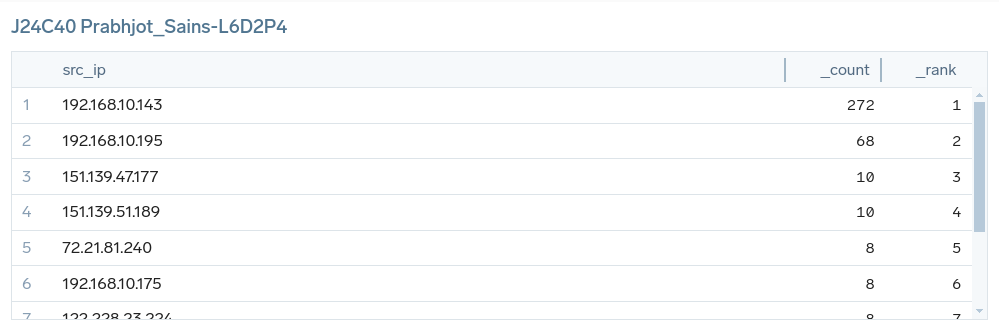
* 1. **Panel-4: Show the table of the top 10 attackers**

\_sourceCategory="mjolnir/hunt"

| count by src\_ip

| topk(10, \_count)





* 1. **Panel-5: Show the top 10 attackers on the map.**

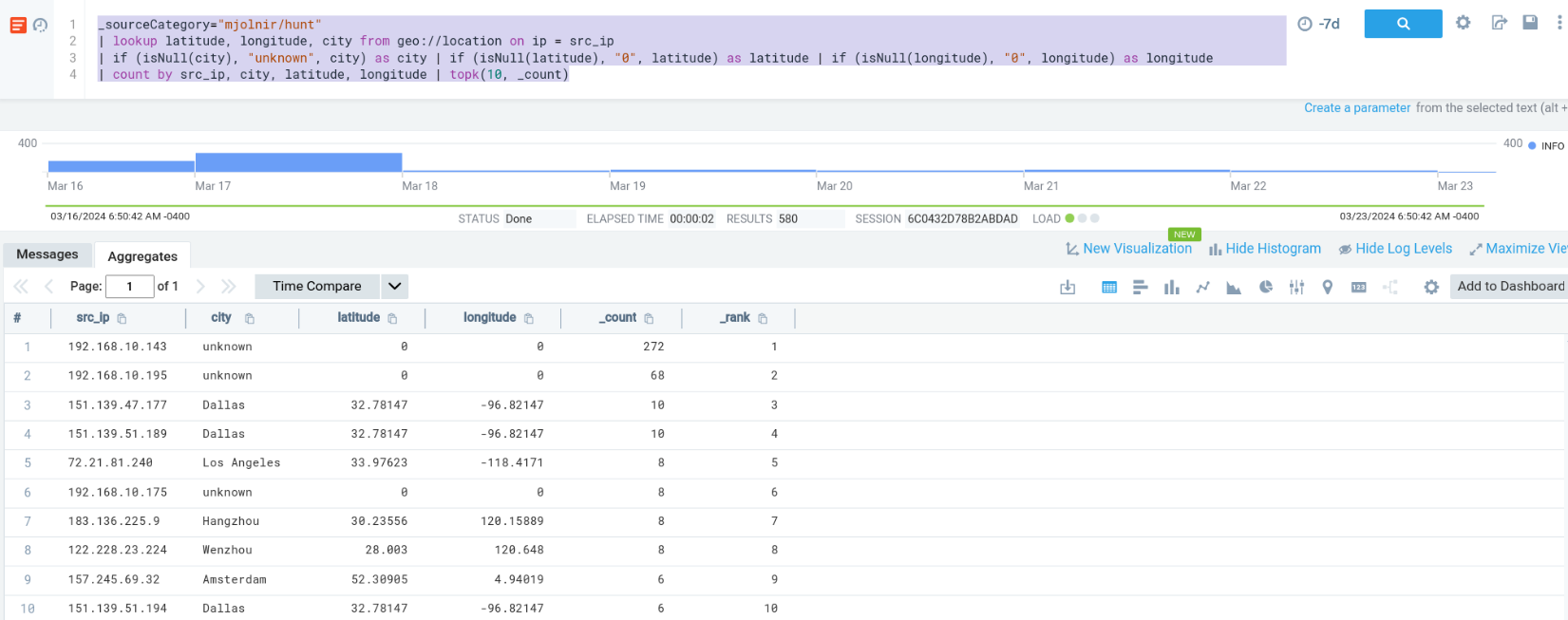
\_sourceCategory="mjolnir/hunt"

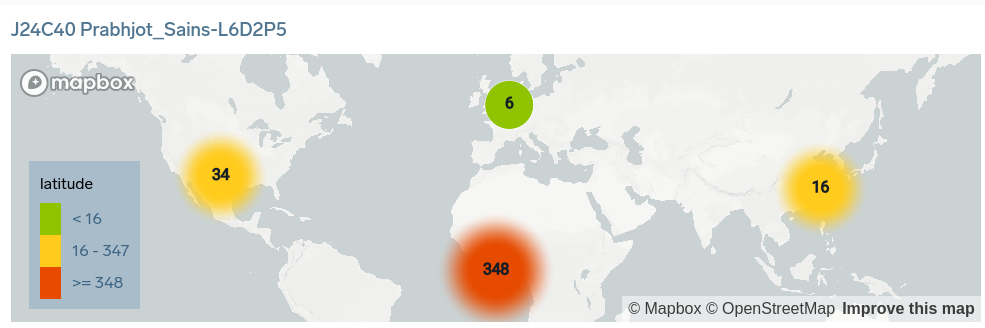
| lookup latitude, longitude, city from geo://location on ip = src\_ip

| if (isNull(city), "unknown", city) as city | if (isNull(latitude), "0", latitude) as latitude | if (isNull(longitude), "0", longitude) as longitude

| count by src\_ip, city, latitude, longitude | topk(10, \_count)

This query shows the logs of the top 10 attackers and based on the source IP latitude and longitude are found and with the help of both of them, a location is found which is used to visualize logs in a map.



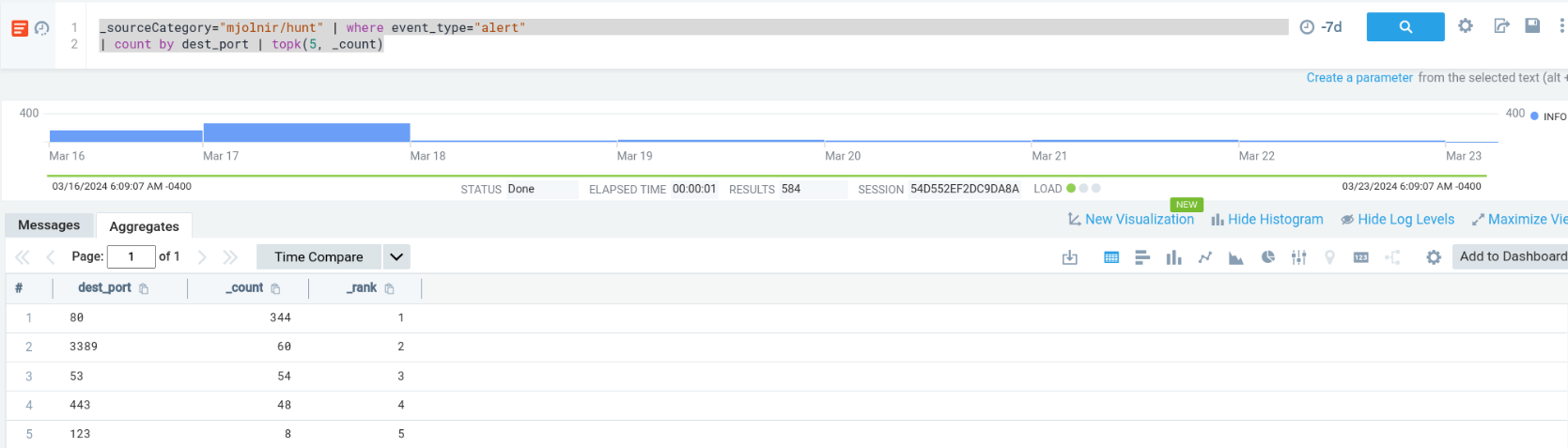


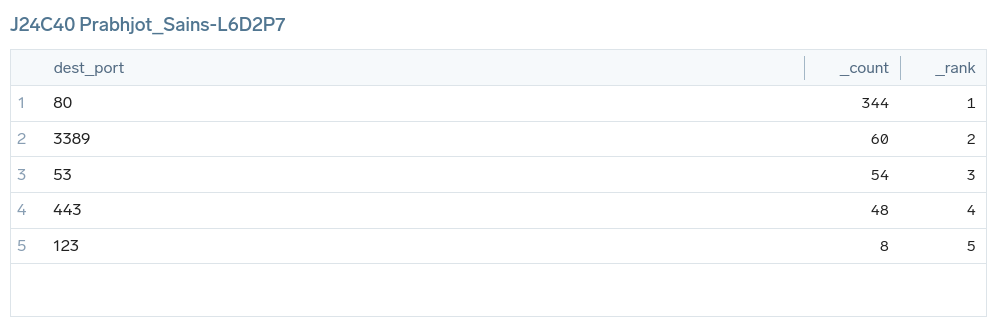
* 1. **Panel-6: Create a table of the top 5 destination ports.**

\_sourceCategory="mjolnir/hunt" | where event\_type="alert"

| count by dest\_port | topk(5, \_count)

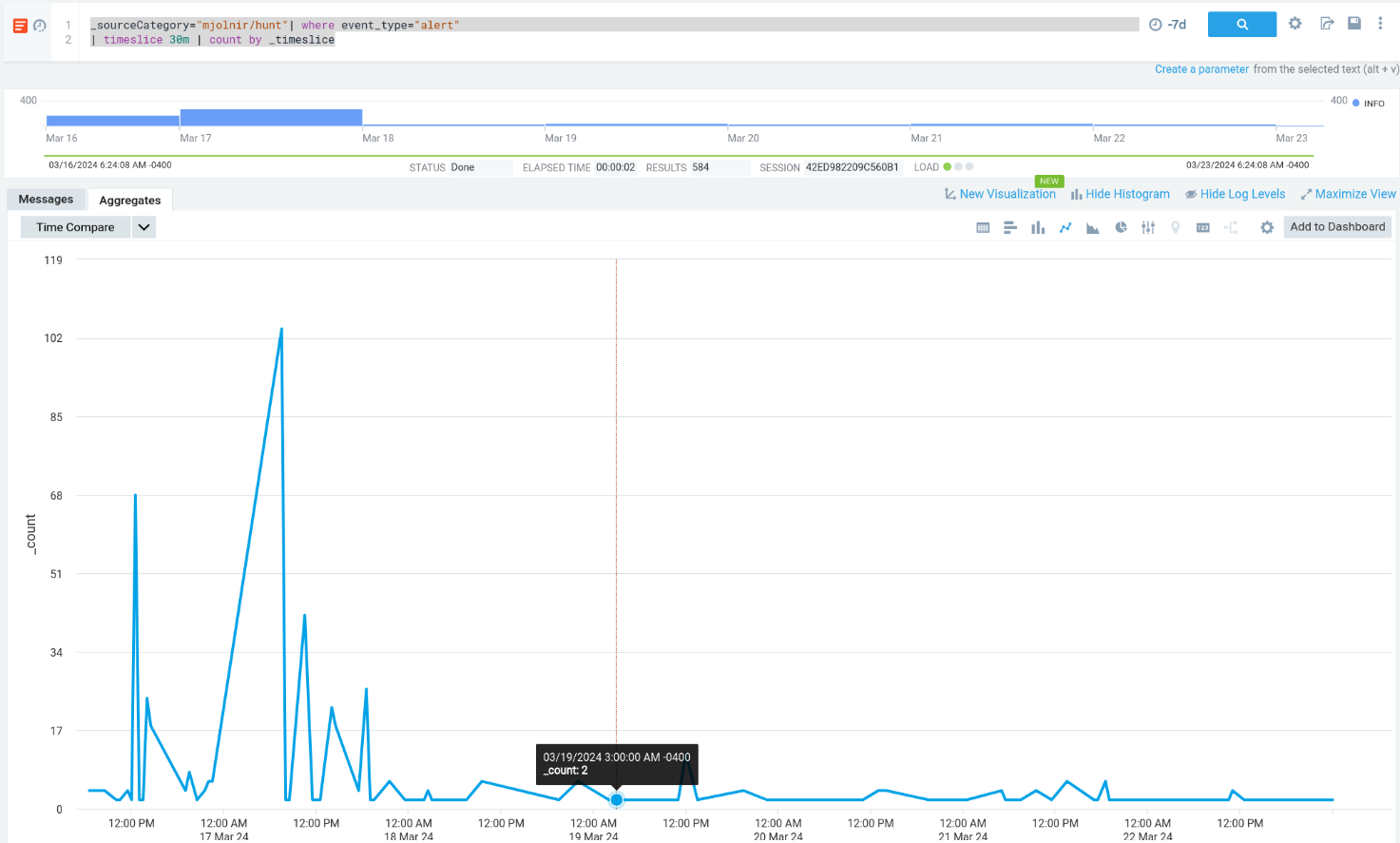
This query is used to filter logs based on destination ports and find the top five destination ports.

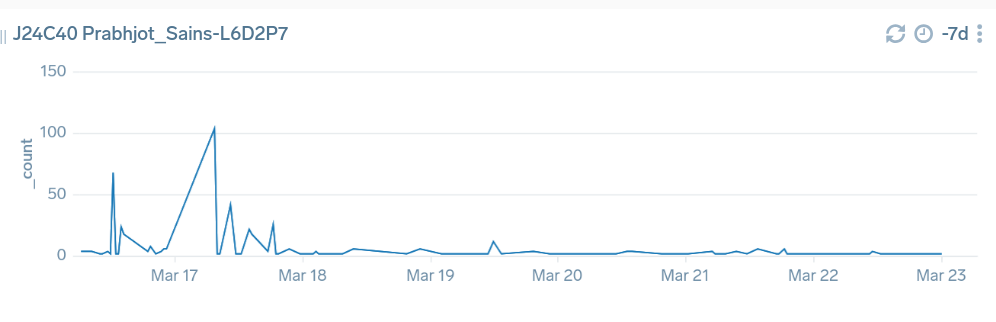




* 1. **Panel-7: Create a graph to visualize event trends every 30 minutes (time slice 30m)**

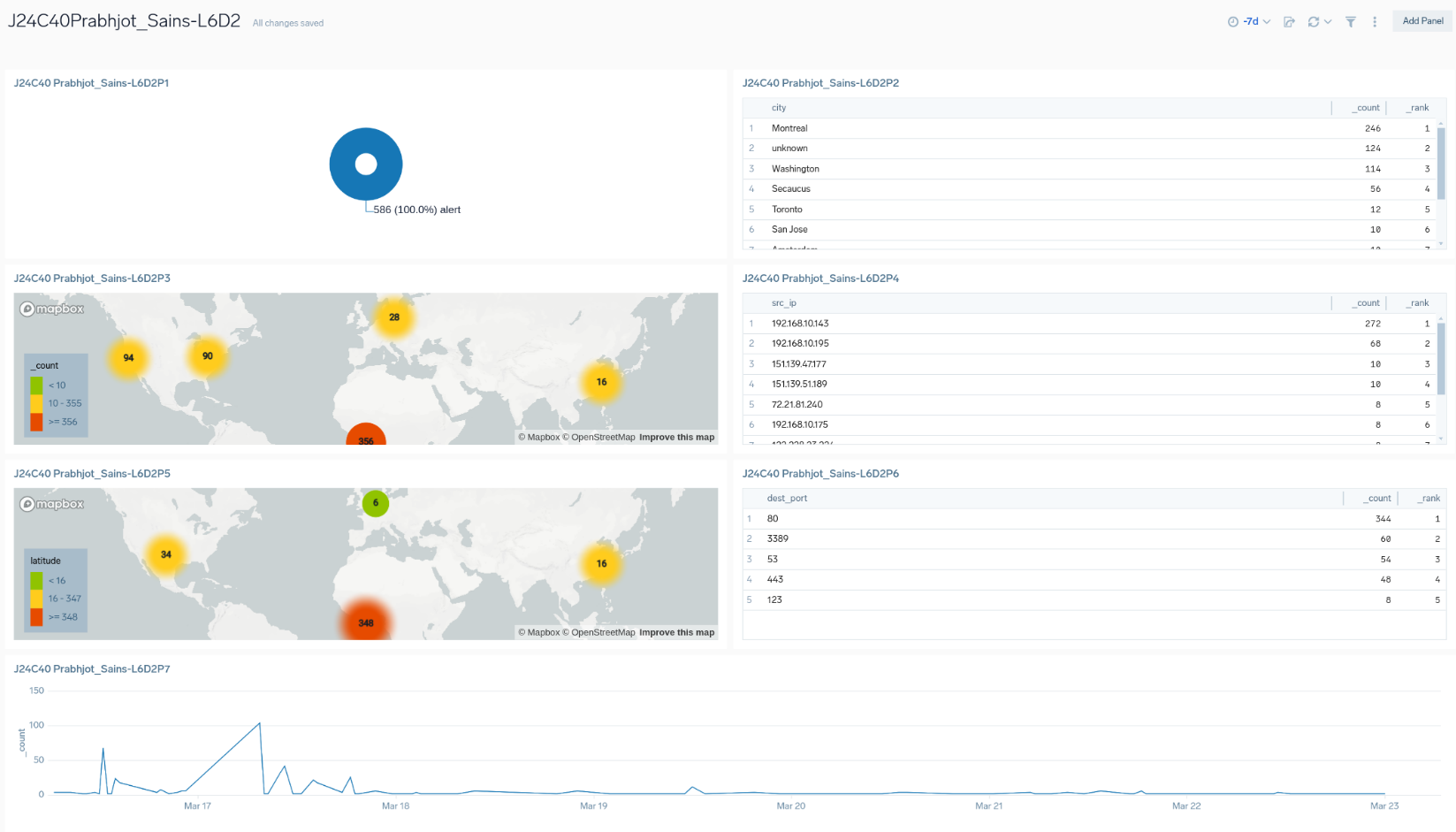
\_sourceCategory="mjolnir/hunt"| where event\_type="alert" | timeslice 30m | count by \_timeslice





* 1. **Create a dashboard and add these 7 Panels to it.**

Shared on sumo logic named as “J24C40Prabhjot\_Sains-L6D2”



* 1. **Analyze the result of each dashboard**
     1. **Panel-1:** Shows a Pi chart based on the event types in which alert is maximum at 100%.
     2. **Panel-2:** Shows the top 10 destinations under the attack of event type alert, the number one destination under attack is Montreal with 246 attacks in the previous 7 days.
     3. **Panel-3:** A map showing the source of an attack of event type alert, with the maximum number of attacks coming from an unknown source.
     4. **Panel-4:** Show the table of the top 10 attackers, the IP address 192.168.10.143 with 272 counts is the attacker with the most attacks.
     5. **Panel-5:** Show the top 10 attackers on the map, with 348 attackers from the African region.
     6. **Panel-6:** Shows Top 5 Destination Ports with first port number 80 with 344 counts.
     7. **Panel-7:** Shows a graph to visualize event trends every 30