**LAB - 7 REPORT**

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

Program: T433 - Cybersecurity

Term: - Winter 2024

Student Names - ID:

* Prabhjot Singh Sains – 101495218

**Use this (\_sourceCategory="Linux/system" and ("invalid user" or "disconnected")) and write queries to: [-7d]**

1. **Extract these fields month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n for all rows using nodrop option. (The nodrop option forces results also to include messages that don't match any segment of the parse expression):**

((((((((((((((((((\_sourceCategory="Linux/system" and ("invalid user" or "disconnected")))))))))))))))))))

| parse "\* \* \* \* \*[\*]: error: maximum authentication attempts exceeded for invalid user \* from \* port \* ssh2 [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnecting invalid user \* \* port \*: Too many authentication failures [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Connection closed by invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Connection reset by invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Failed password for invalid user \* from \* port \* ssh2" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnected from invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnected from authenticating user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: error: maximum authentication attempts exceeded for invalid user \* from \* port \* ssh2 [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnecting invalid user \* \* port \*: Too many authentication failures [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Invalid user \* from \* port \*" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnected from invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Disconnected from authenticating user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Failed none for invalid user \* from \* port \* ssh2" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

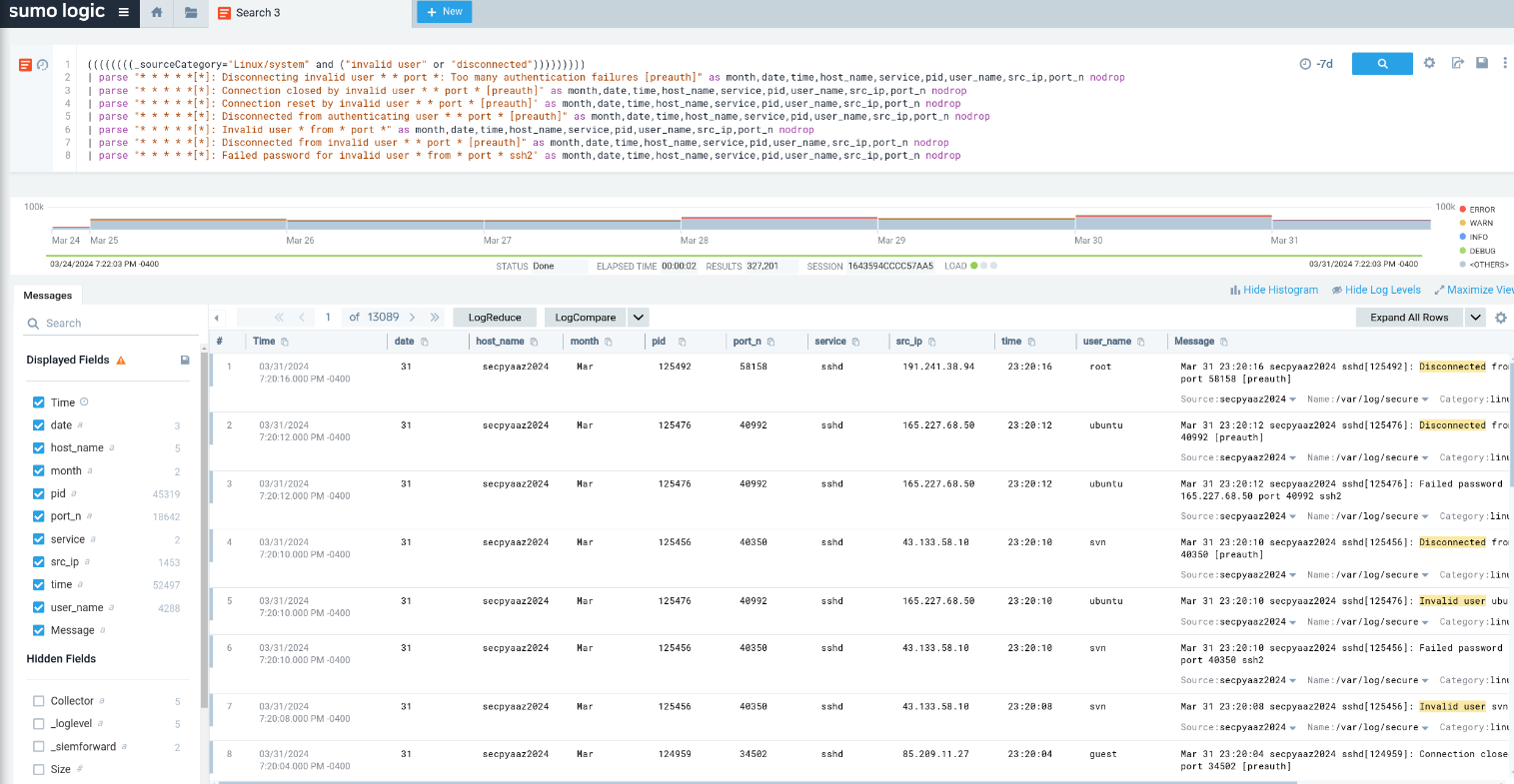
| parse "\* \* \* \* \*[\*]: Connection reset by invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Invalid user \* from \* port \*" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

| parse "\* \* \* \* \*[\*]: Failed password for invalid user \* from \* port \* ssh2" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

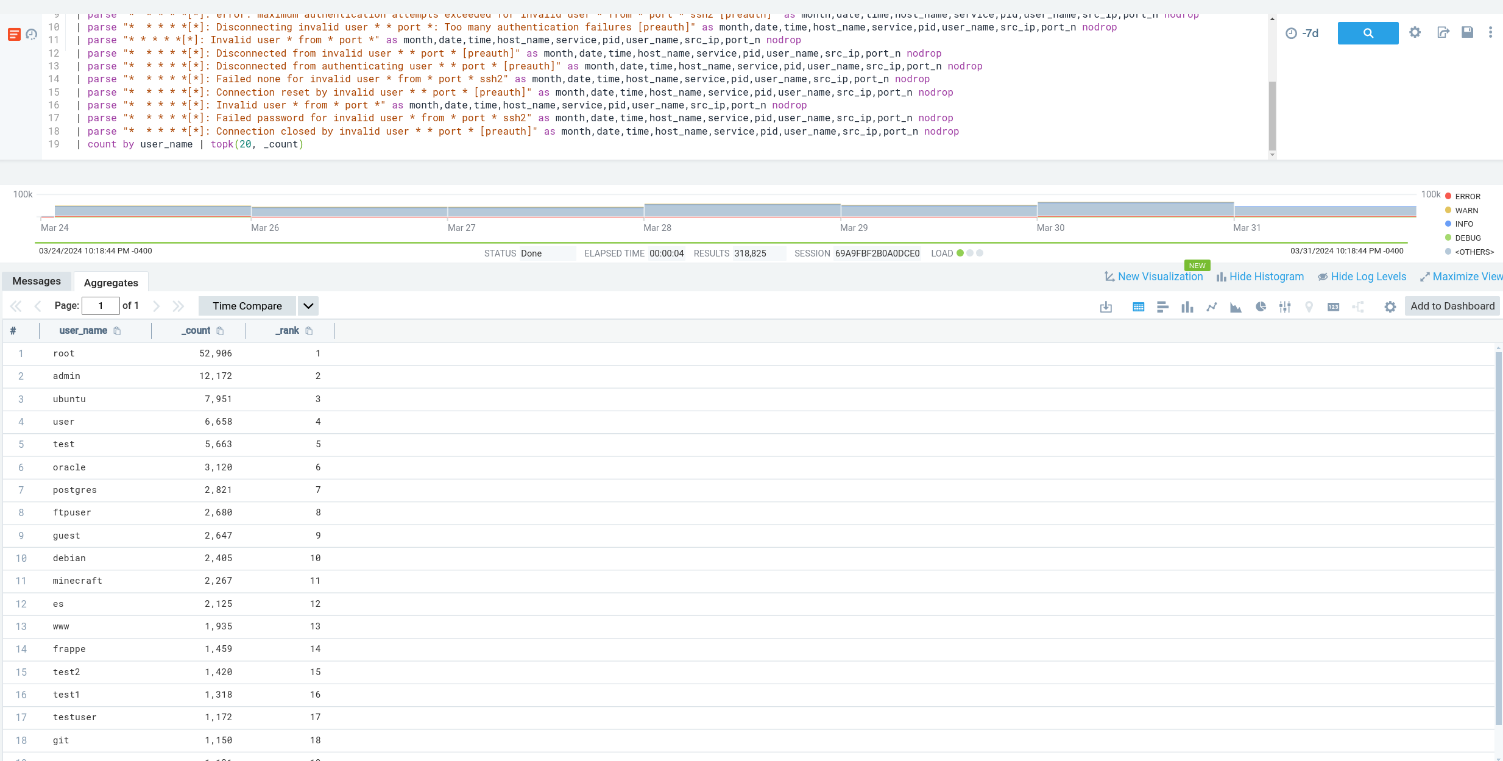
| parse "\* \* \* \* \*[\*]: Connection closed by invalid user \* \* port \* [preauth]" as month,date,time,host\_name,service,pid,user\_name,src\_ip,port\_n nodrop

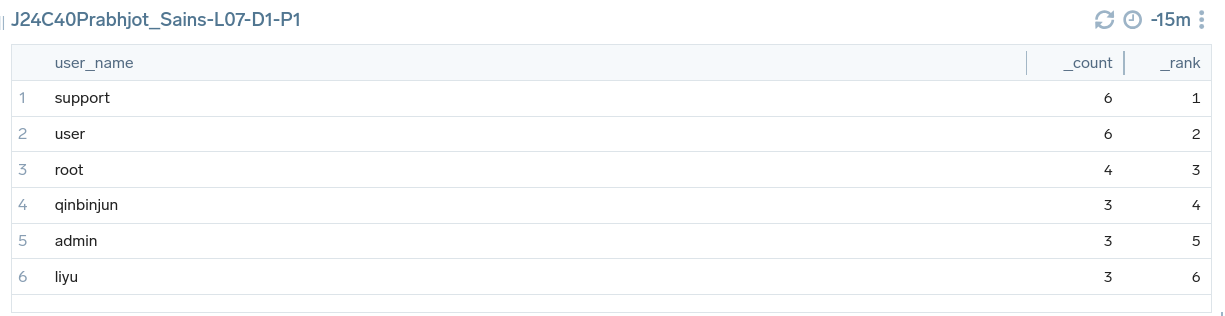
These queries are used to parse the data in the logs in the human-readable form. It helps to extract the month, date, time, hostname, service, PID, username, source IP, and port number. The no-drop option forces results also to include messages that don't match any segment of the parse expression. After using nodrop parsing is done to include messages to match.



1. **Create a table showing the most commonly tried usernames (top 20) as Panel-1:**

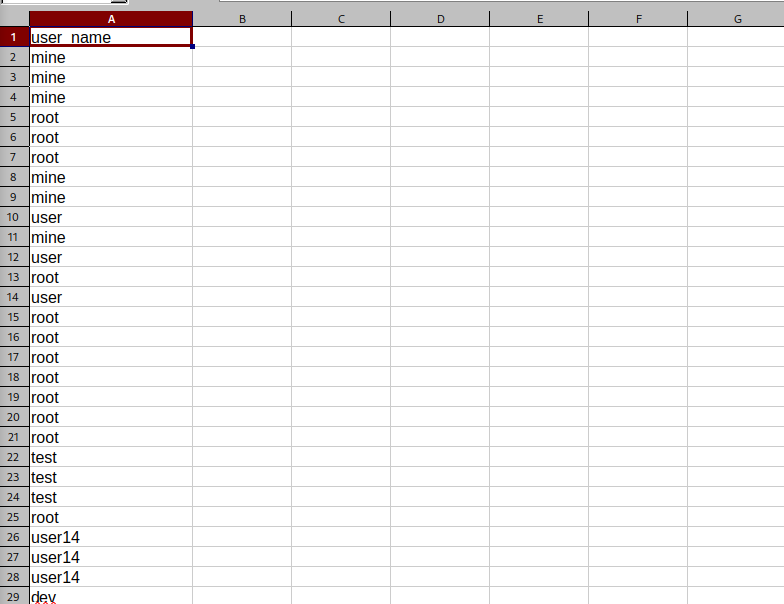
In the query from the first part add the following query to get the result “| count by user\_name | topk(20, \_count)”, This will show the top 20 most commonly tried usernames. The output is added in the dashboard as Panel-1.





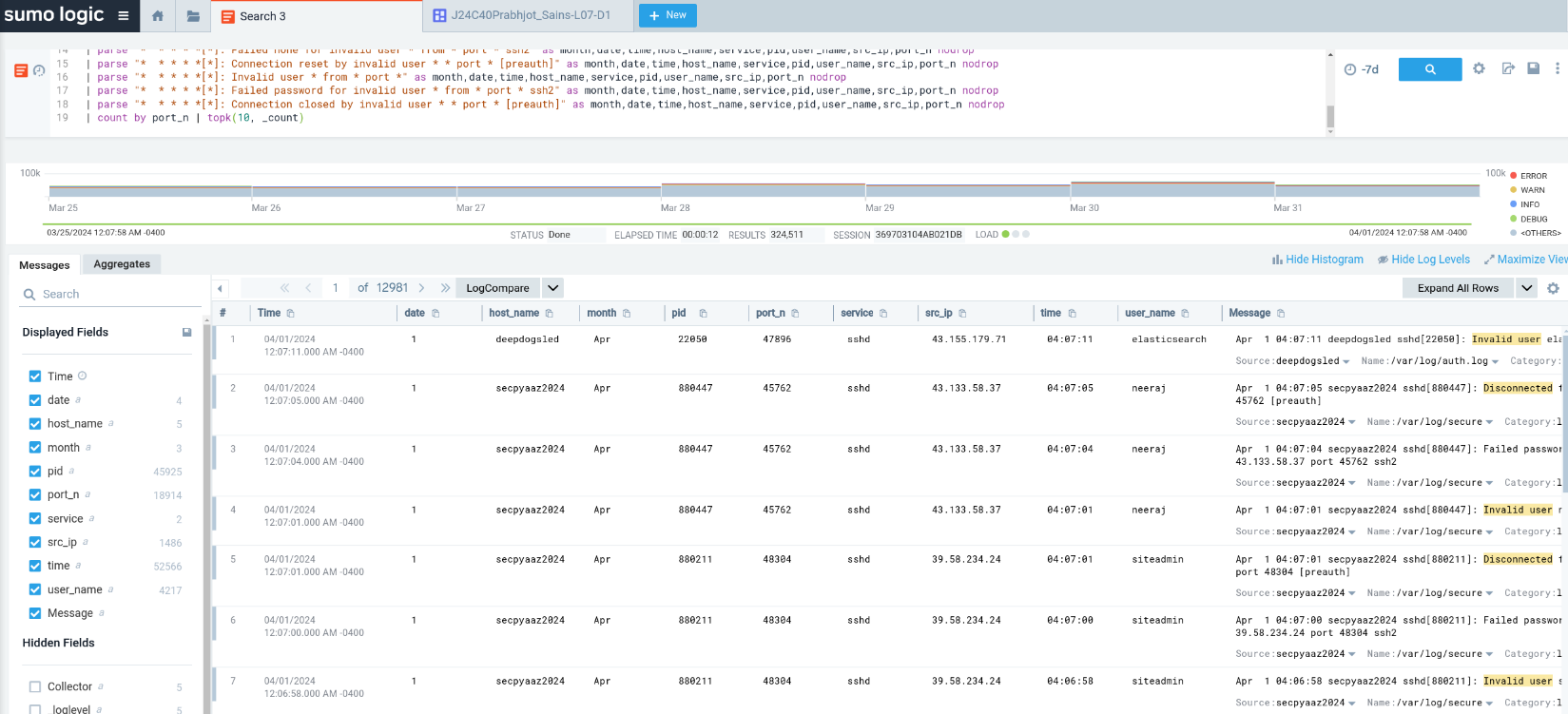
1. **Download a list of all tried users:**

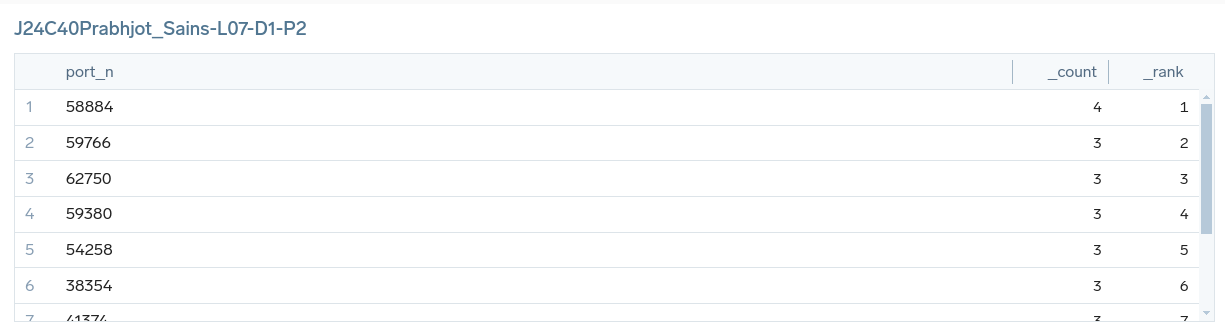
Downloaded the list in CSV format. Some of them are shown below.



1. **Create a table of the most common tried port numbers (top 10) as Panel-2:**

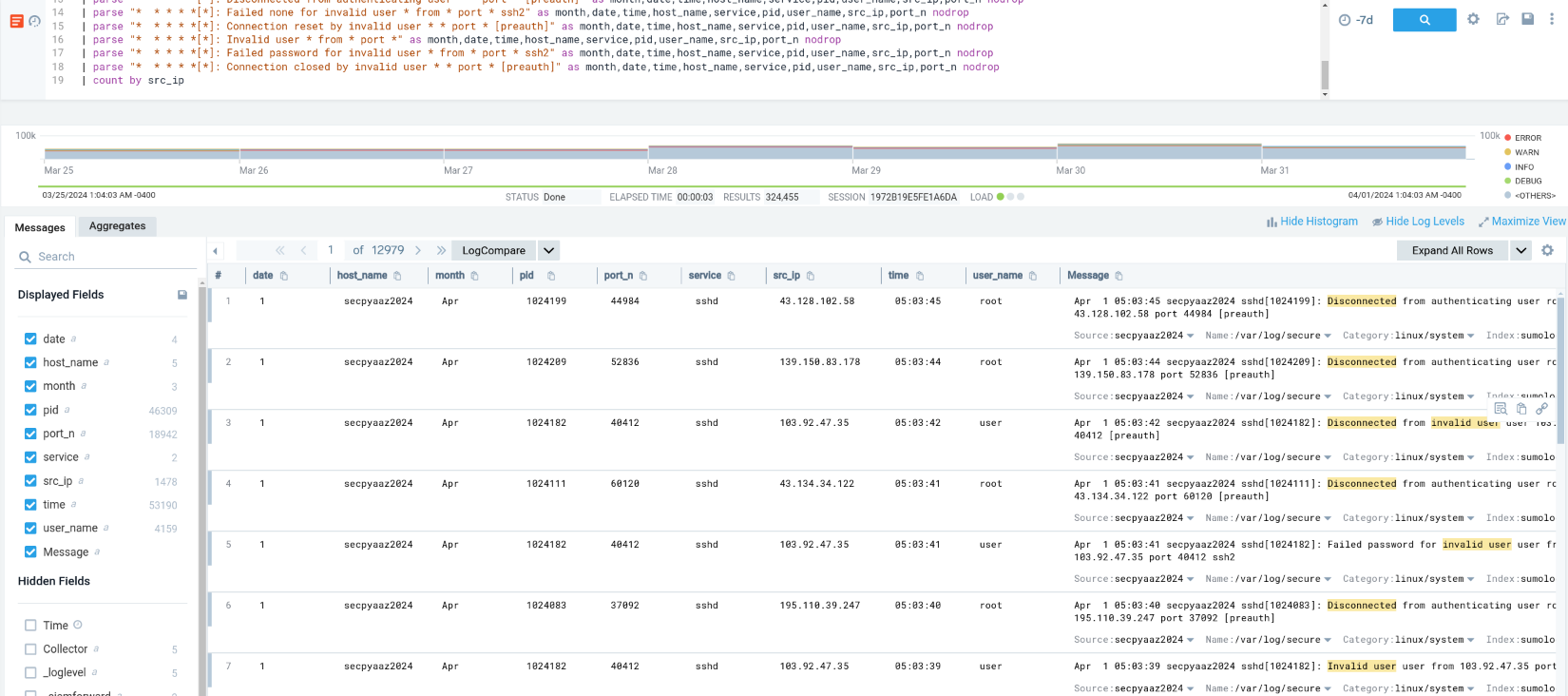
In the query from the first part add the following query to get the result “| count by port\_n | topk(10, \_count)”, This will show the top 10 most commonly tried port numbers. The output is added in the dashboard as Panel-2.

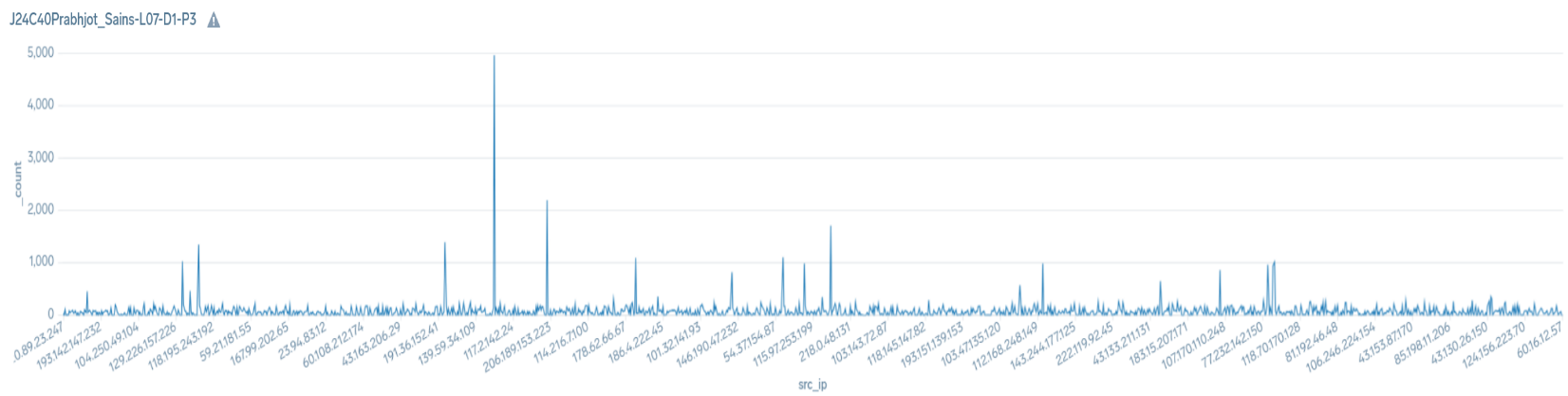




1. **Create a graph to show the number of failed tries per source IP (brute fourth) as Panel-3:**

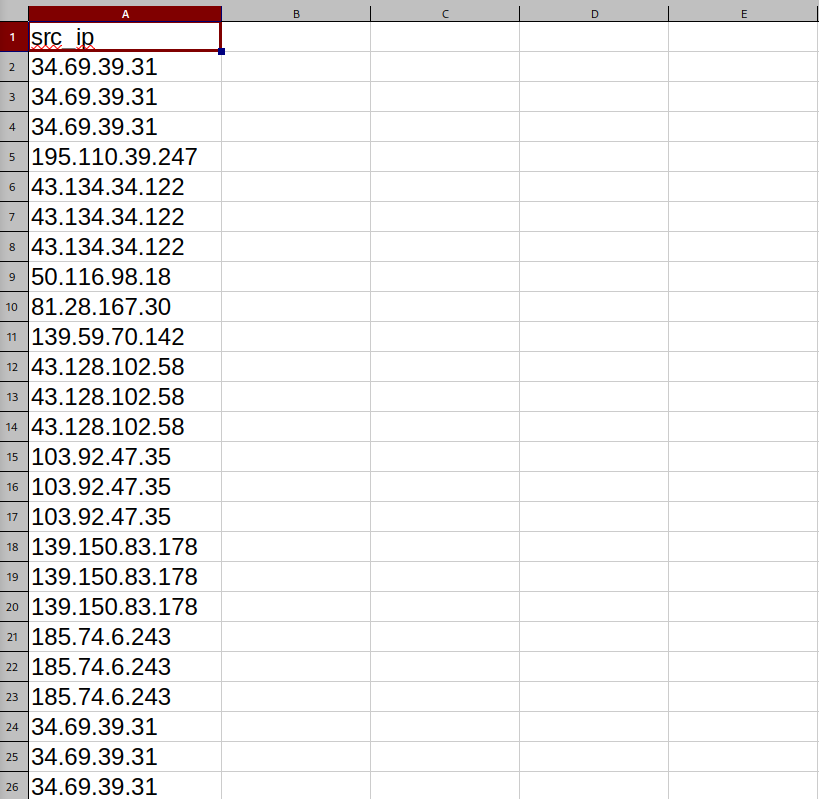
In the query from the first part add the following query to get the result “| count by src\_ip | sort by \_count desc”, This will show the number of failed tries per source IP. The output is added in the dashboard as Panel-3 in the graph format.



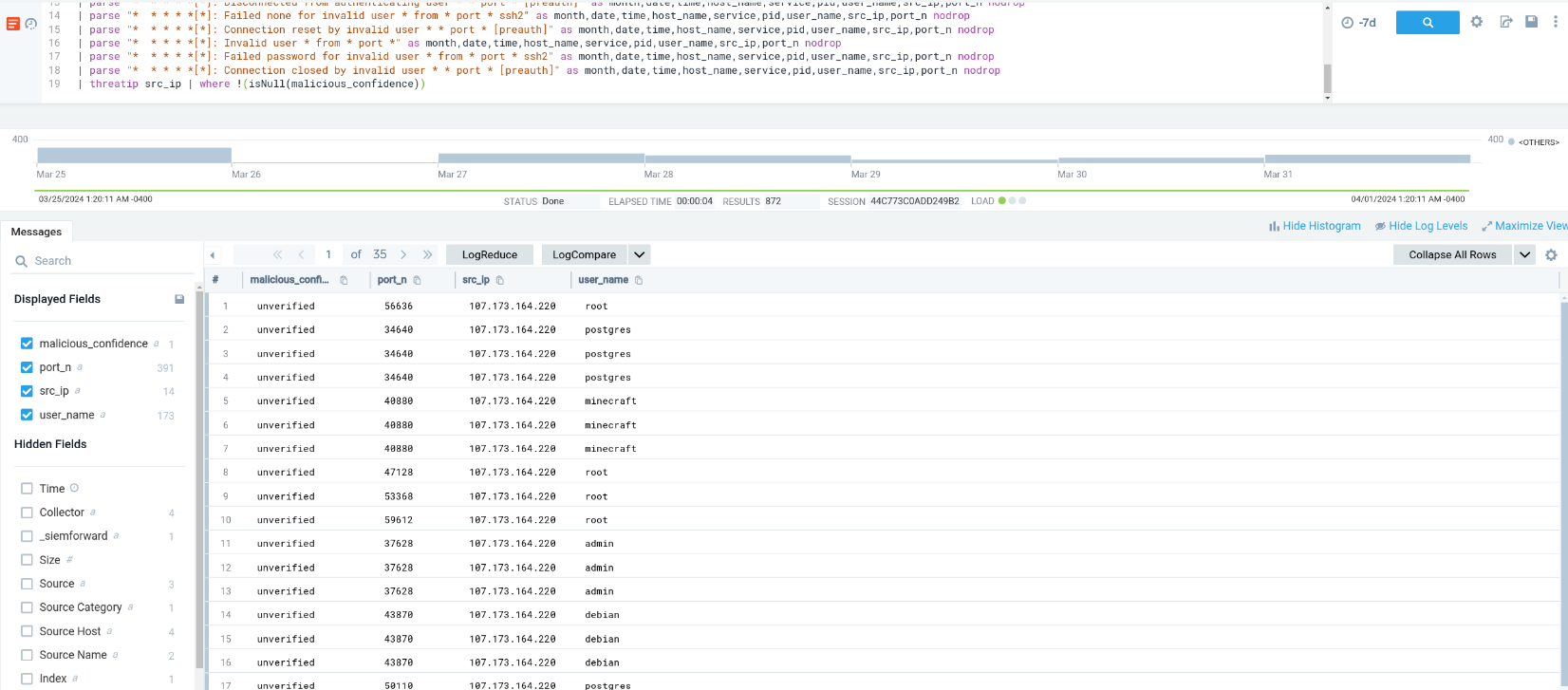


1. **Download the list of Attacker Ips:**

Downloaded the list in CSV format. Some of them are shown below.

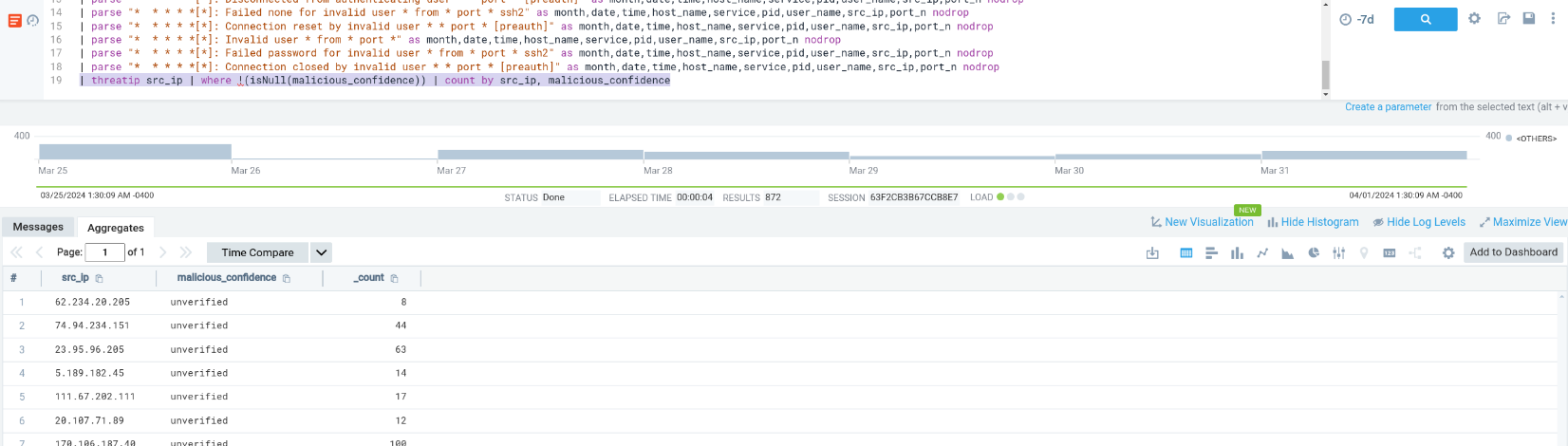


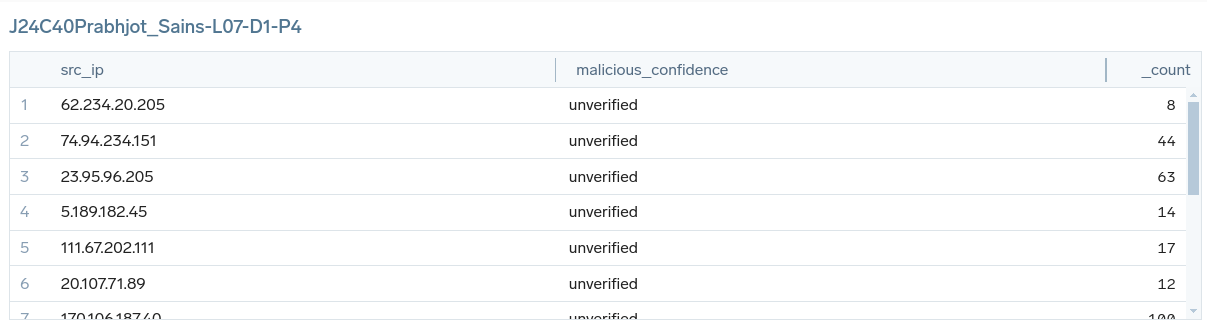
1. **Check malicious\_confidence of src\_ip using “| threatip src\_ip” function and “| where !(isNull(malicious\_confidence))” condition:**



1. **Create a table of src\_ip and their malicious\_confidence level as Panel-4:**

In the query from the first part add the following query to get the result “| threatip src\_ip | where !(isNull(malicious\_confidence)) | count by src\_ip, malicious\_confidence”, This will show the table of source Ip and their Malicious confidence. The output is added in the dashboard as Panel-4.





1. **Visualize the location of src\_ips by malicious\_confidence low or above. Panel-5:**

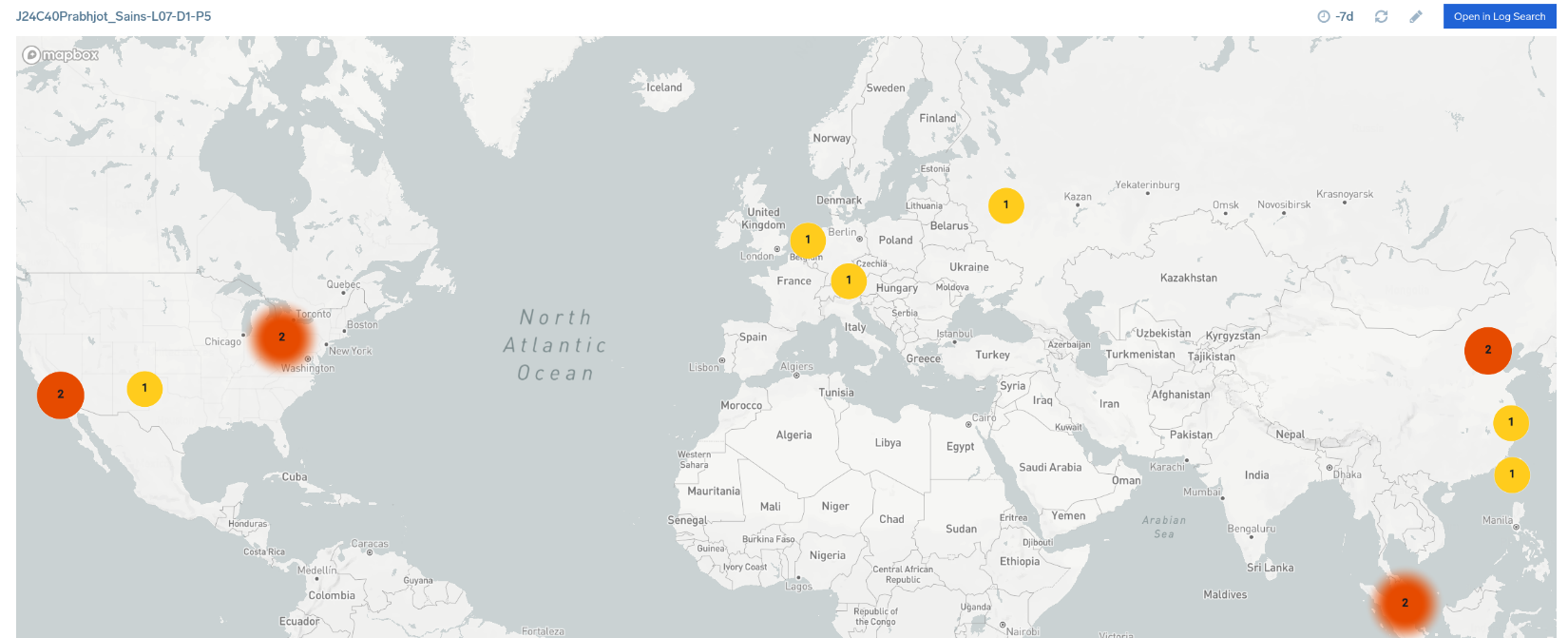
| threatip src\_ip | where !(isNull(malicious\_confidence)) | count by src\_ip, malicious\_confidence

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

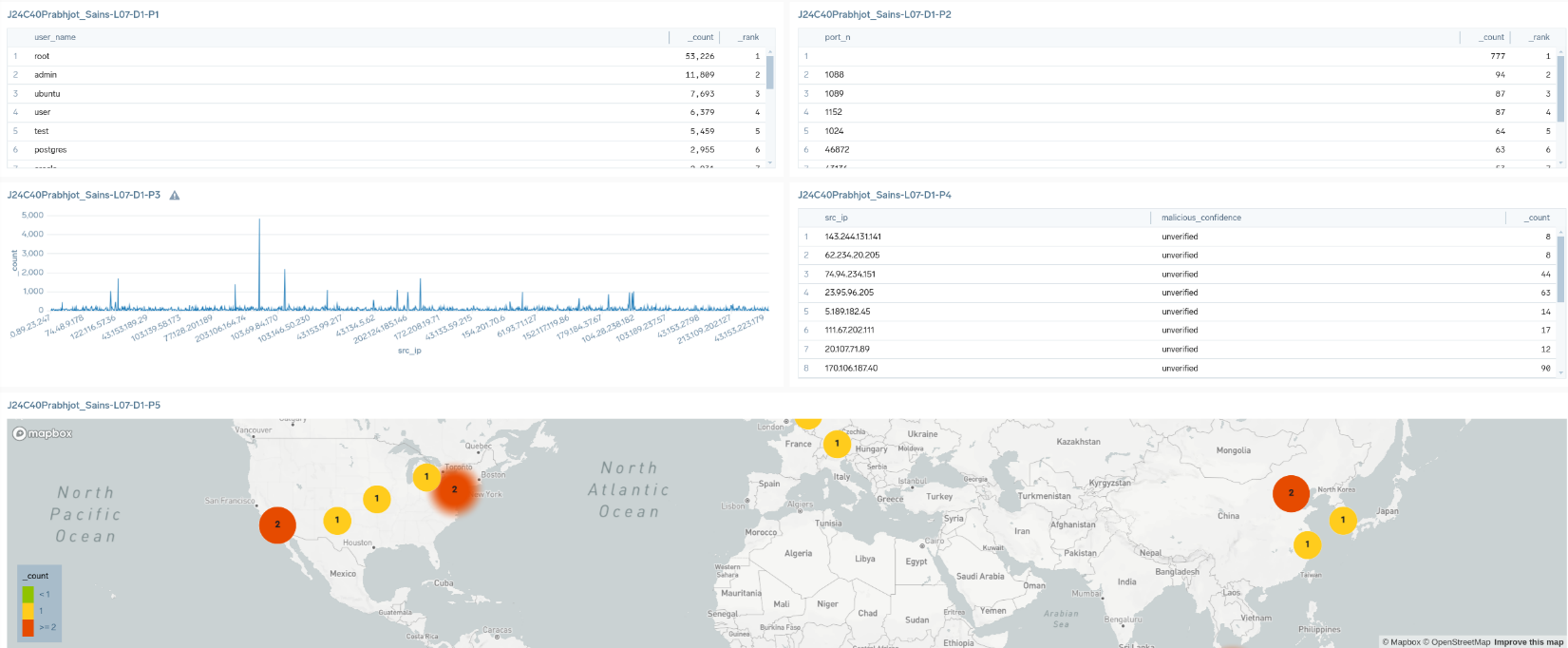
| fields src\_ip, malicious\_confidence, latitude, longitude

| where !(isNull(latitude) OR isNull(longitude))

| count by latitude, longitude



1. **Dashboard:** Shared on sumo logic.

****

1. **Panels Analysis:**
   1. In the first panel the top 20 most commonly tried usernames are shown from which Root users have tried most of the time with 53,226 total counts.
   2. In the second panel the most common top 10 tried port numbers are shown from which port numbers which are not parsed is maximum with 777 counts and port number 1088 is second most common with 94 counts.
   3. In the third panel a graph shows the number of failed tries per source IP (brute fourth) in which IP 103.69.84.170 is the one with the maximum number of failed tries with almost 5,000 counts.
   4. In the fourth panel a table of src\_ip and their malicious\_confidence level are shown and every source IP has unverified malicious confidence.
   5. In the fifth panel the location of src\_ips by malicious\_confidence is shown.

**Browse http://cslab.softether.net:8989/ from your host machine then try to connect to this URL with both default and customized User Agent "your nickname" (user agent spoofing). You can use different browser features to do this or use an assessment tool like Nikto:**

**nikto -h http://cslab.softether.net:8989/ -user-agent "your nickname"**

**then Use this (\_source="klj23-03-apache-access") source and write queries to:**

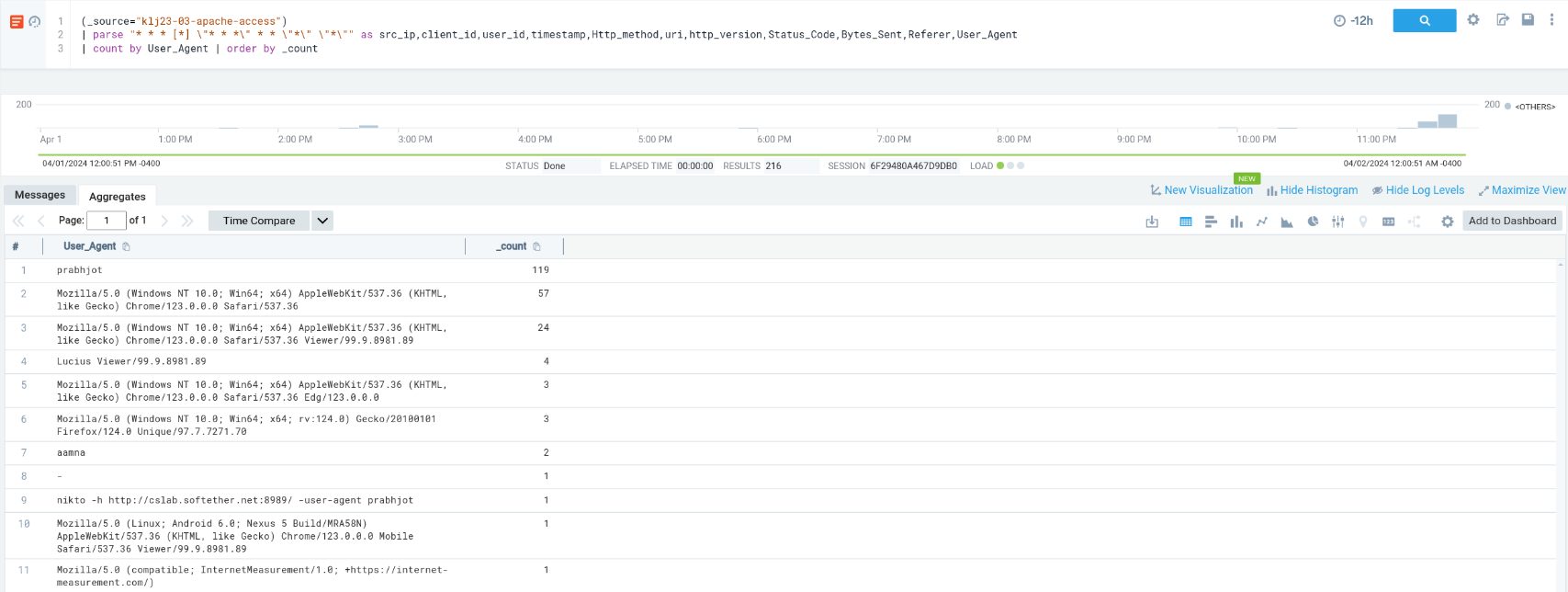
1. **Extract different information and create a table view to show different user agents and count each user agent [12h] Panel-1:**

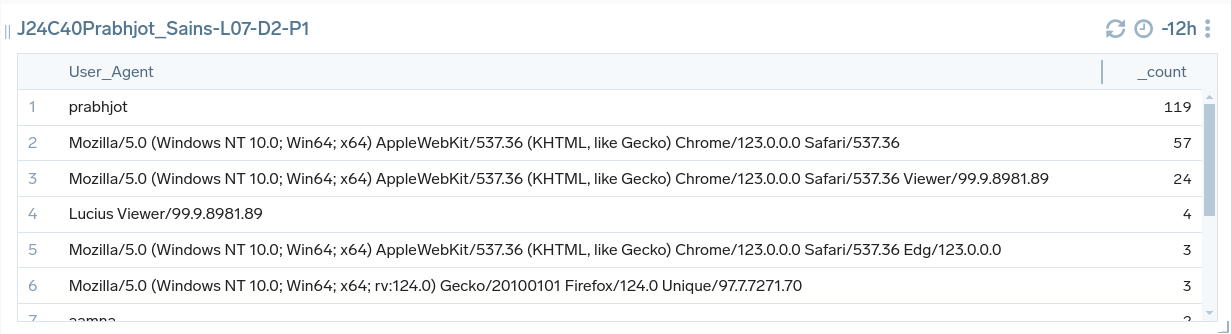
(\_source="klj23-03-apache-access")

| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip,client\_id,user\_id,timestamp,Http\_method,uri,http\_version,Status\_Code,Bytes\_Sent,Referer,User\_Agent

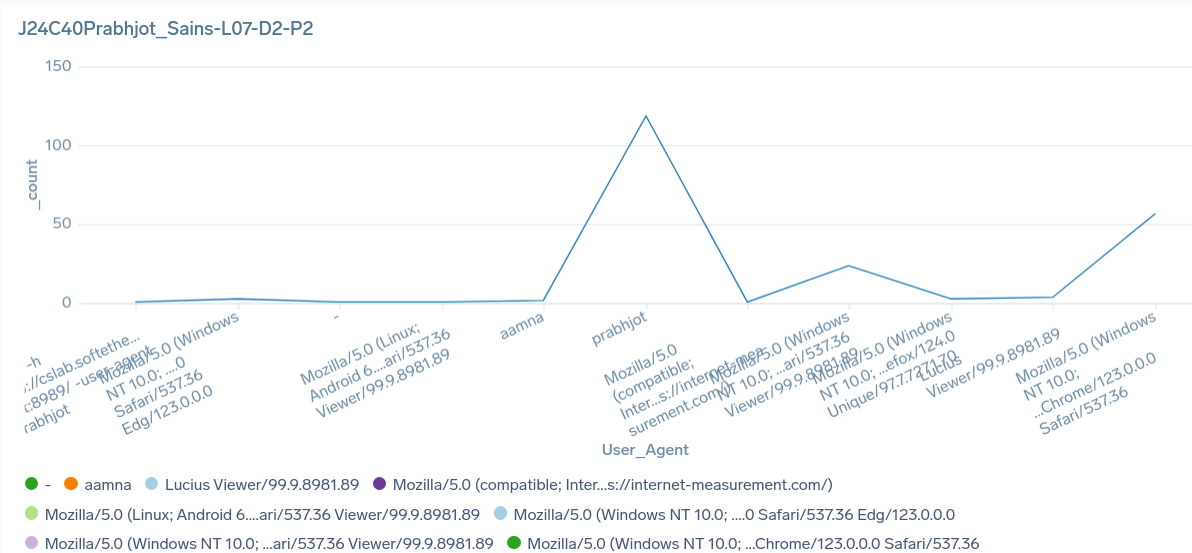
| count by User\_Agent | order by \_count

In this query, parsing is done to extract the values from the logs. Manily we are extracting User Agent and count and order the User Agent.





1. **Create a graph to visualize this information. Panel-2:**

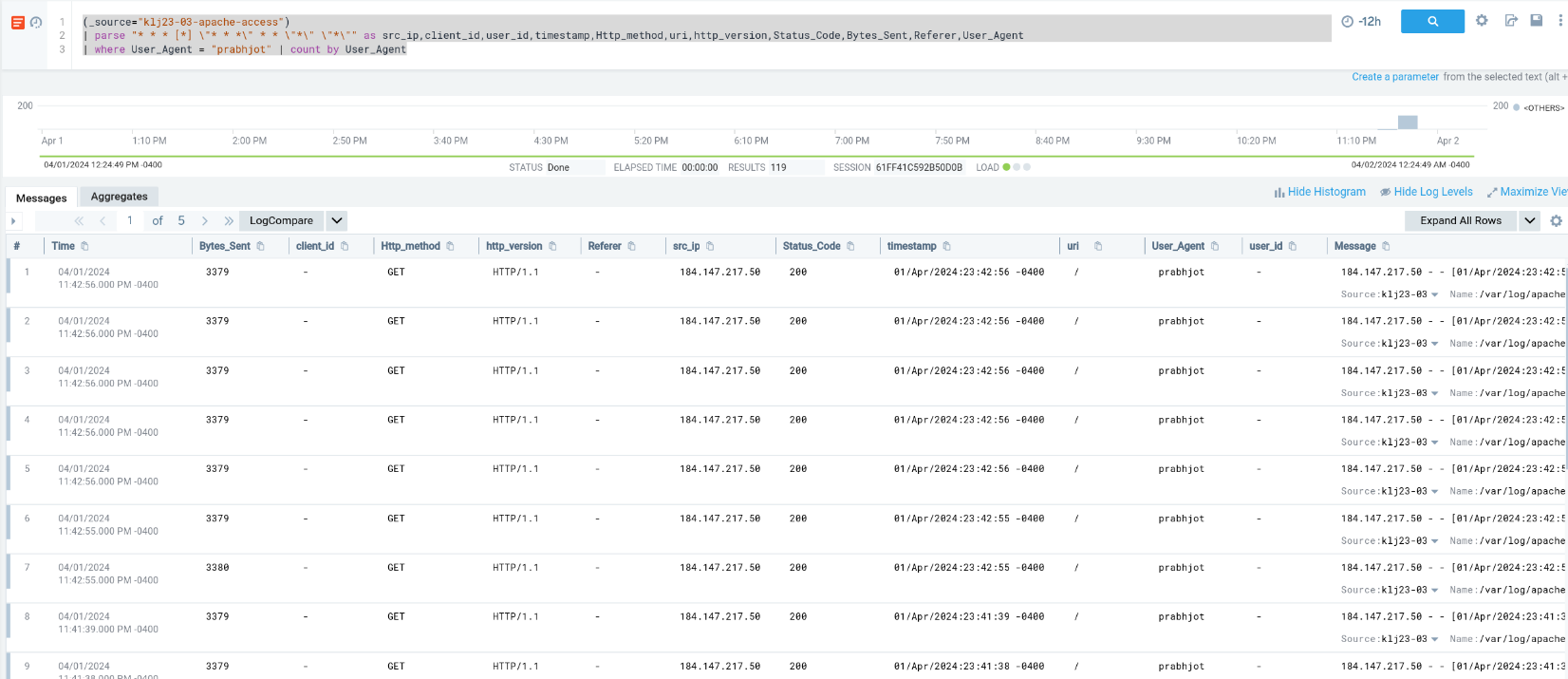
****

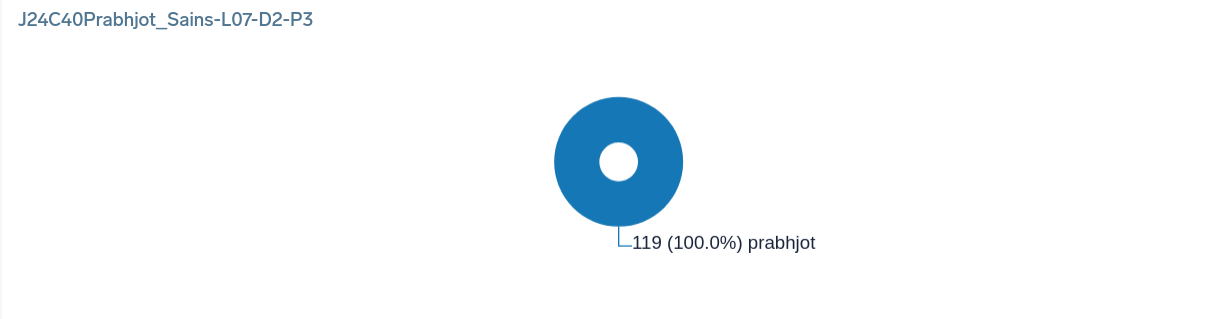
1. **Use a query to search your custom user agent. Panel-3:**

(\_source="klj23-03-apache-access")

| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip,client\_id,user\_id,timestamp,Http\_method,uri,http\_version,Status\_Code,Bytes\_Sent,Referer,User\_Agent

| where User\_Agent = "prabhjot" | count by User\_Agent



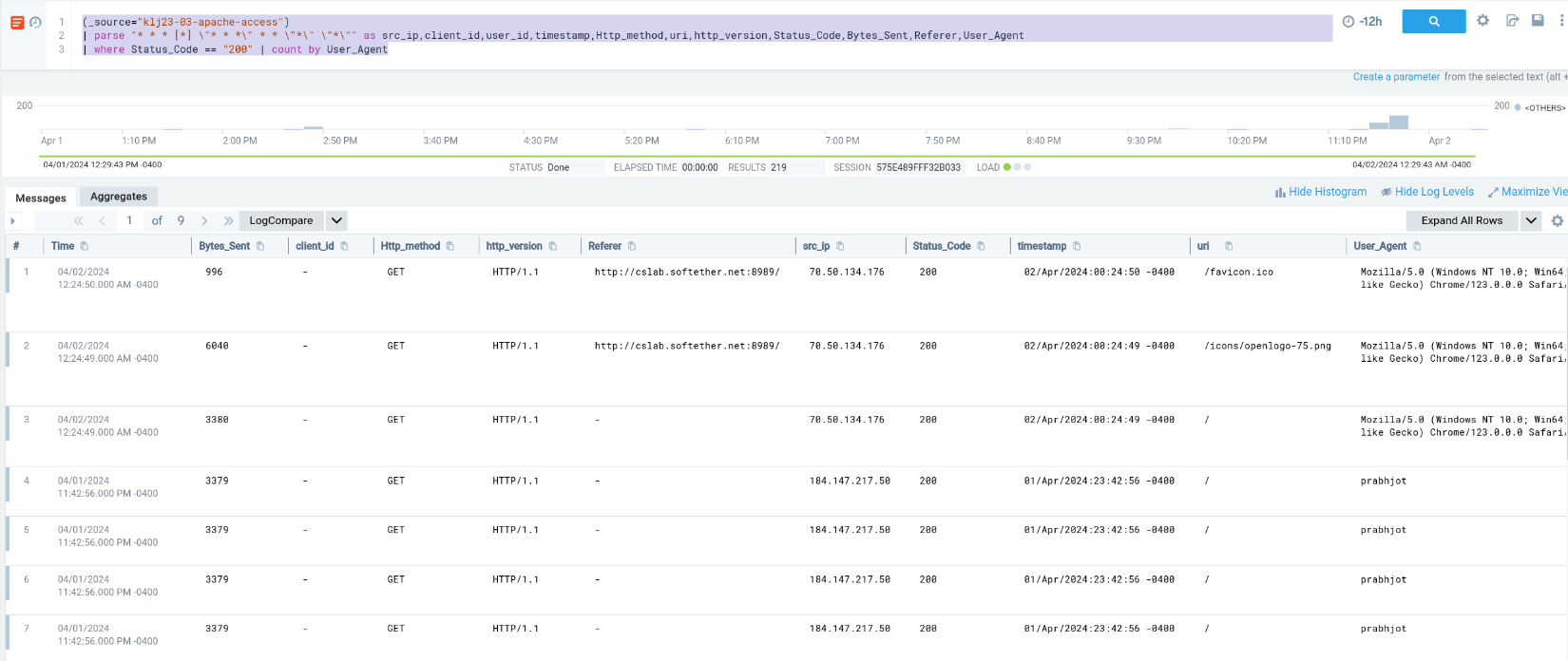


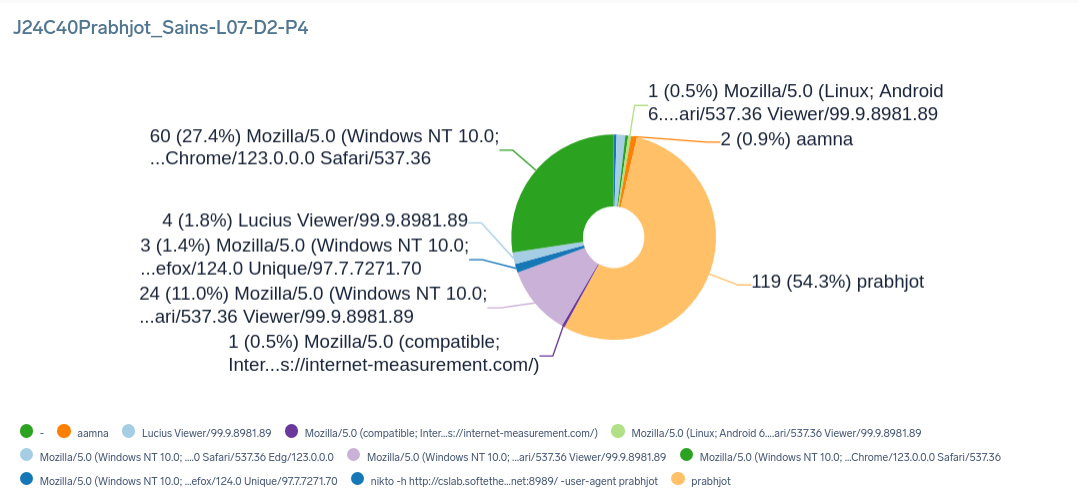
1. **Write a query to visualize the number of successful accesses per user agent. Panel-4:**

(\_source="klj23-03-apache-access")

| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip,client\_id,user\_id,timestamp,Http\_method,uri,http\_version,Status\_Code,Bytes\_Sent,Referer,User\_Agent

| where Status\_Code == "200" | count by User\_Agent





1. **Show source IP locations on a map. Panel-5:**

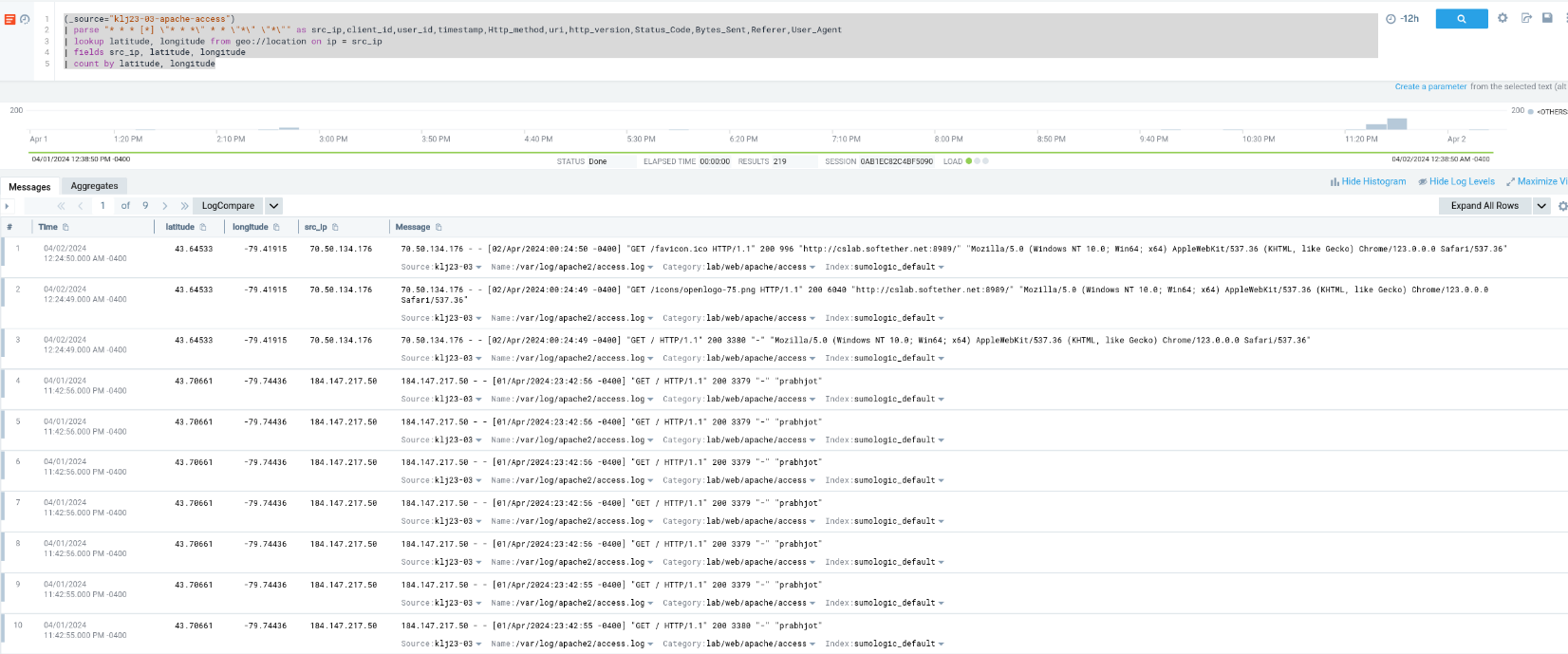
(\_source="klj23-03-apache-access")

| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip,client\_id,user\_id,timestamp,Http\_method,uri,http\_version,Status\_Code,Bytes\_Sent,Referer,User\_Agent

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

| fields src\_ip, latitude, longitude

| count by latitude, longitude





1. **Find your default User Agent and compare it with your browser. Panel-6:**

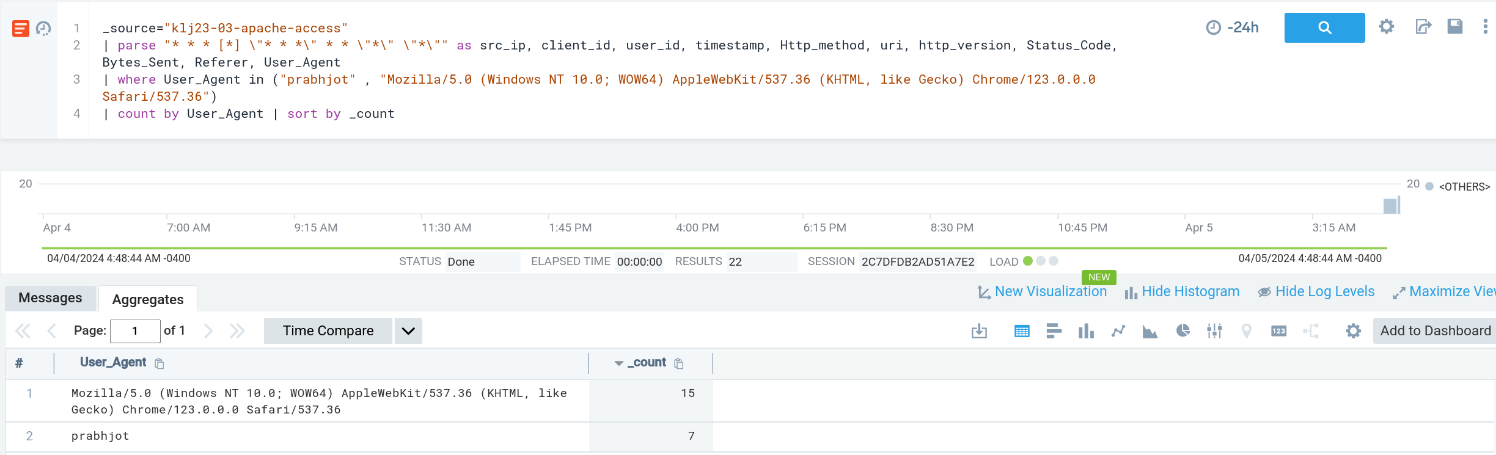
\_source="klj23-03-apache-access"

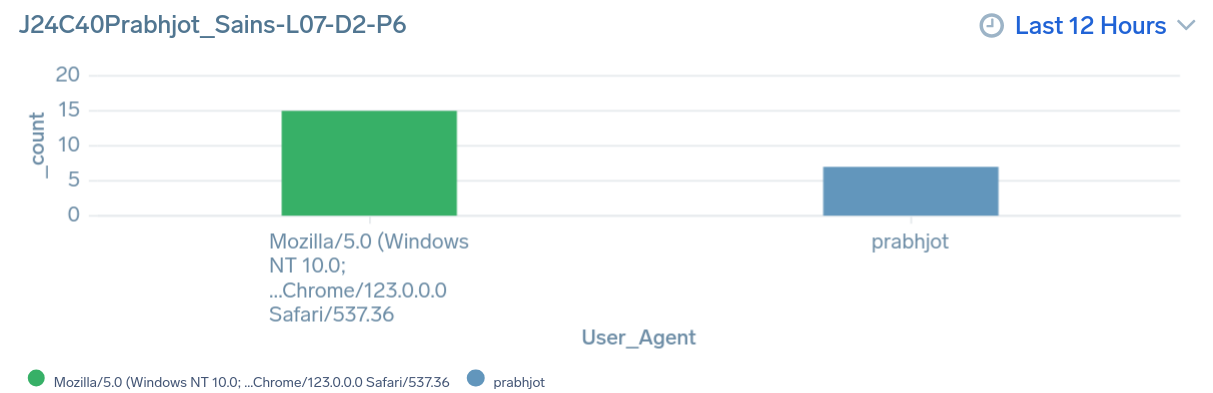
| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip, client\_id, user\_id, timestamp, Http\_method, uri, http\_version, Status\_Code, Bytes\_Sent, Referer, User\_Agent

| where User\_Agent in ("prabhjot" , "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36")

| count by User\_Agent | sort by \_count

This query compares the occurrence of two user agents in the logs with each other. One is the browser user agent and the other one is the default user agent.



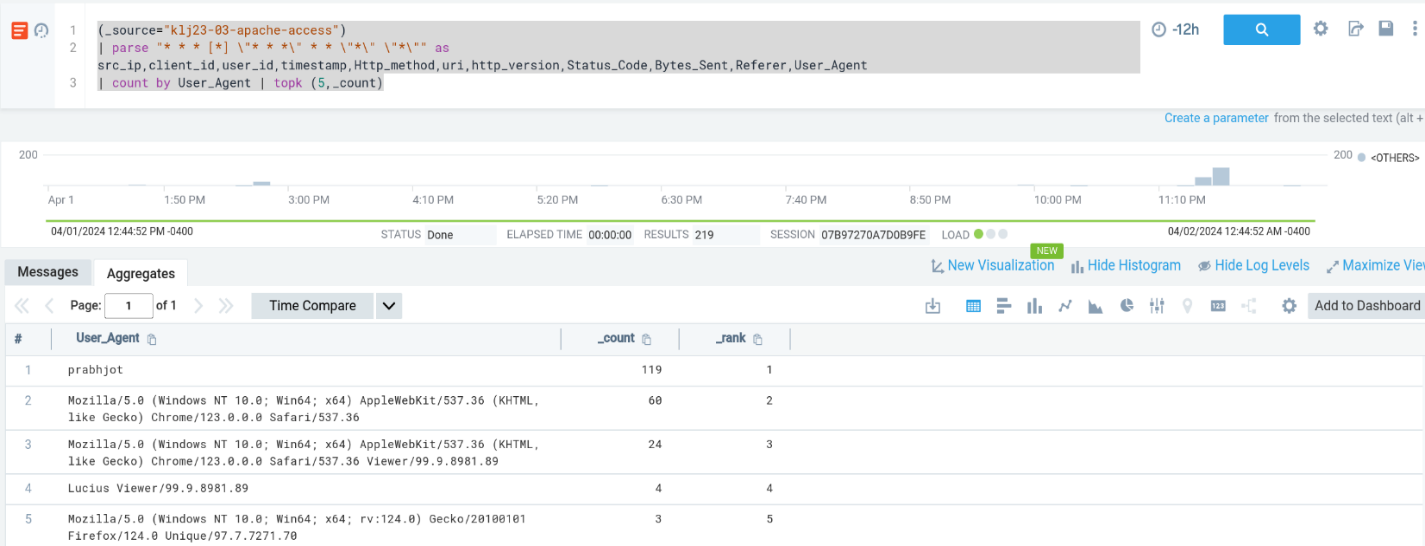


1. **Analyze top 5 user agents (search about client browser) took topk(5,\_count):**

(\_source="klj23-03-apache-access")

| parse "\* \* \* [\*] \"\* \* \*\" \* \* \"\*\" \"\*\"" as src\_ip,client\_id,user\_id,timestamp,Http\_method,uri,http\_version,Status\_Code,Bytes\_Sent,Referer,User\_Agent

| count by User\_Agent | topk (5,\_count)



* The browser for the first and fourth user agents is not mentioned.
* The second and third user agents in the list use the Chrome browser.
* The fifth user agent in the list is using Firefox.

1. **Dashboard:** Shared on sumo logic.

