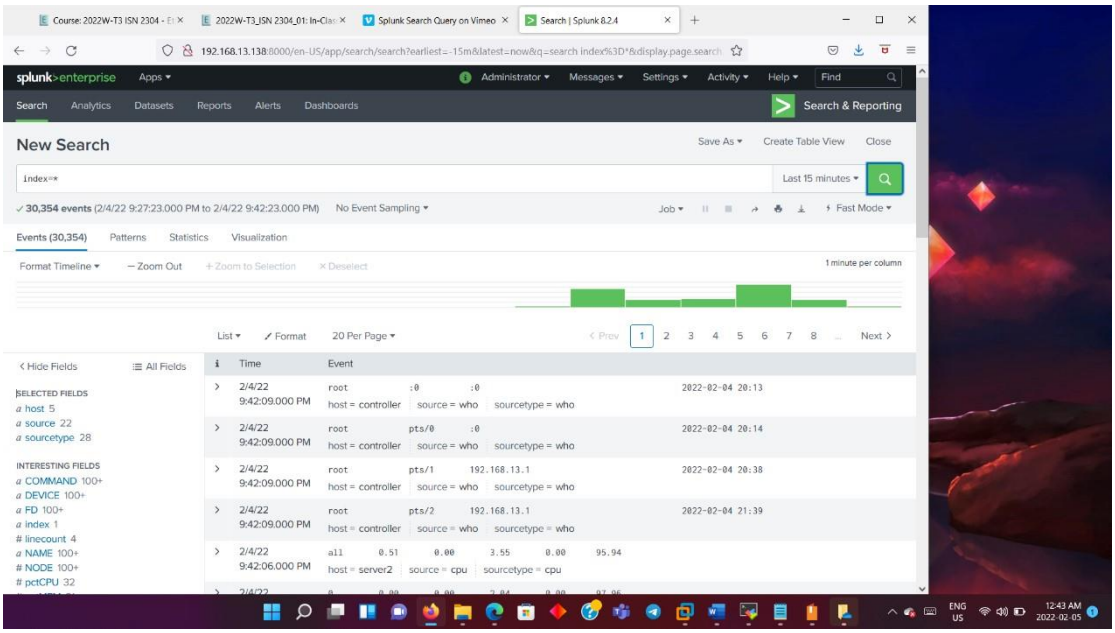
Contents

Abstract

This report describes in detail about various features of Splunk and Splunk Phantom. Such as the behavior of search queries in Splunk by executing various queries. We will also be investigating a brute-force attack and analyze some events. By installing Splunk SOAR Phantom, we will be automating incident response tasks. Also, investigate in detail about a security incident by uploading a .csv file.

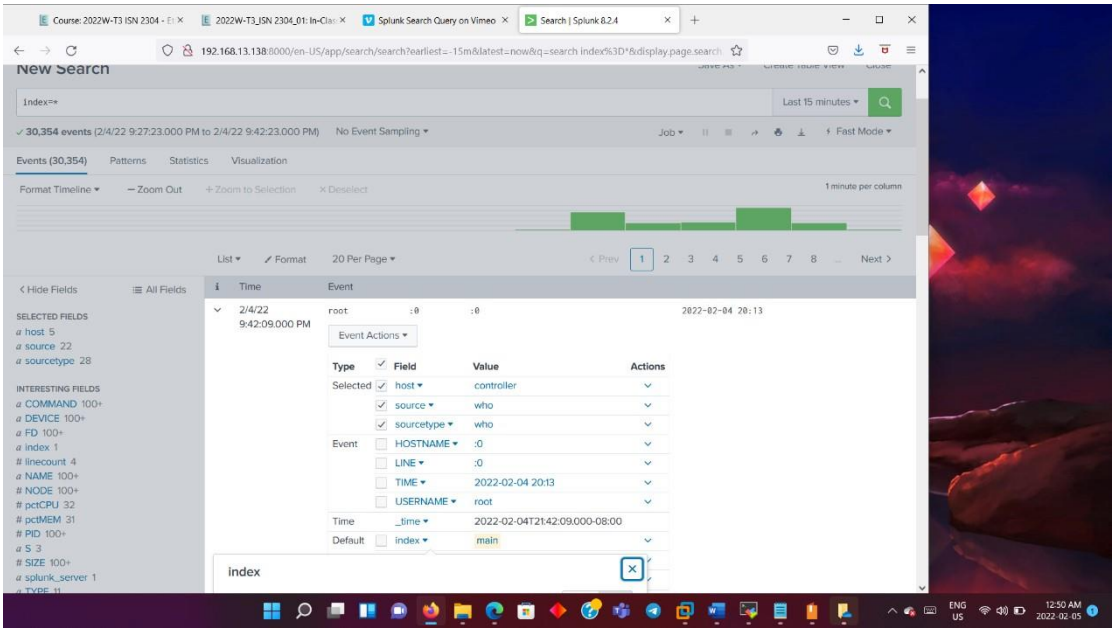
Part 1a

Splunk Search Query / Brute-Force attack

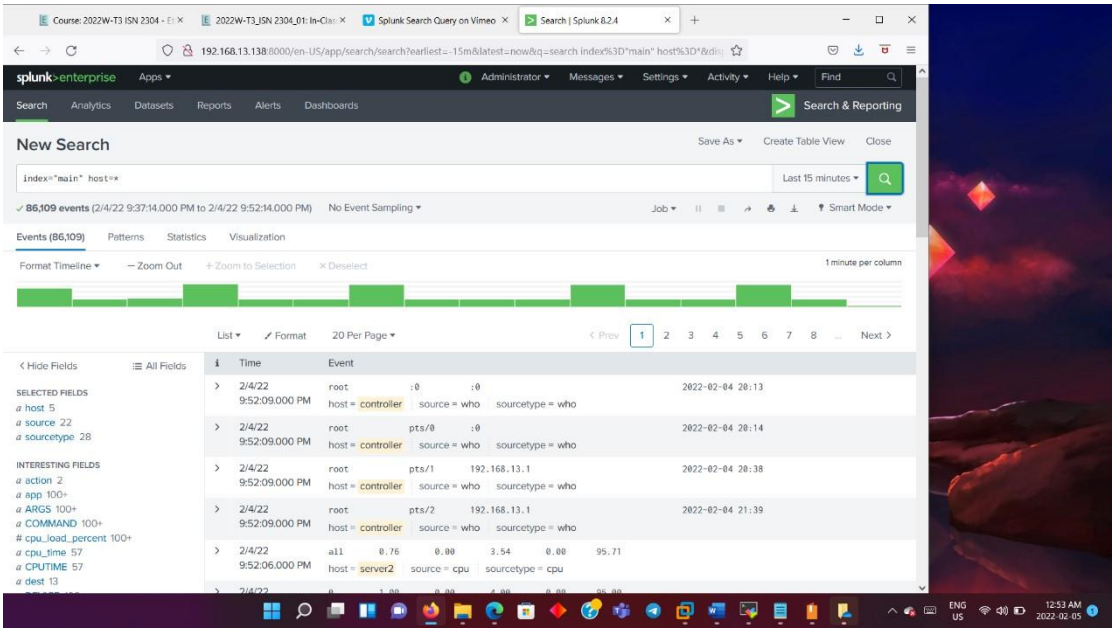


First, we restart the controller, server1 and 2 and then we check if the splunk web is functional.

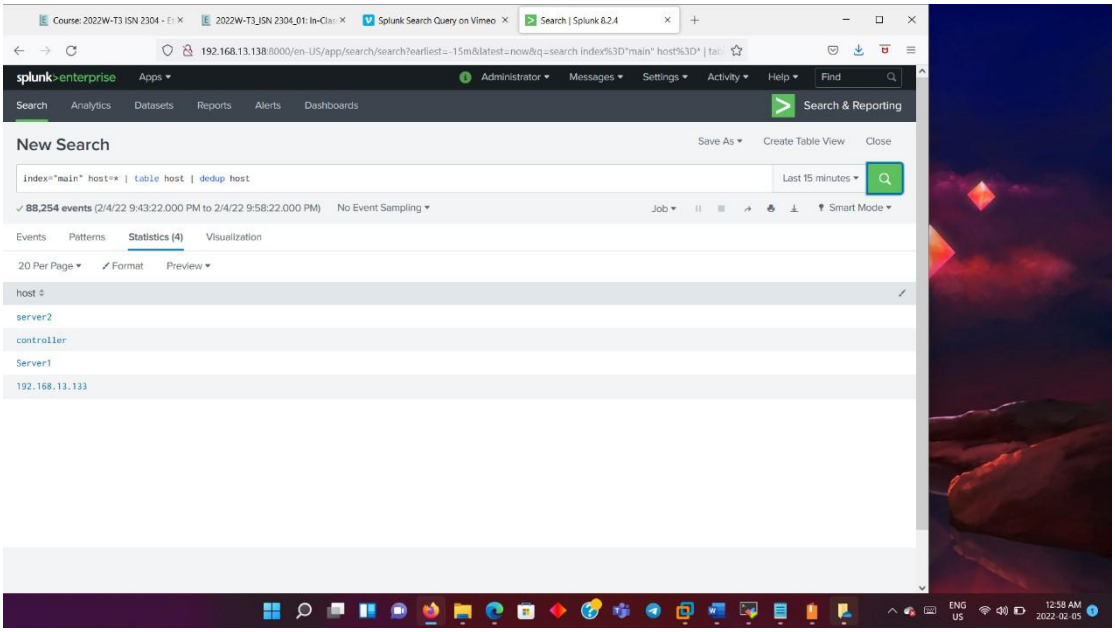
Execute query 1 index=* in search, and choose, fast or smart mode to get the output as above.



We then expand and check the index as main.

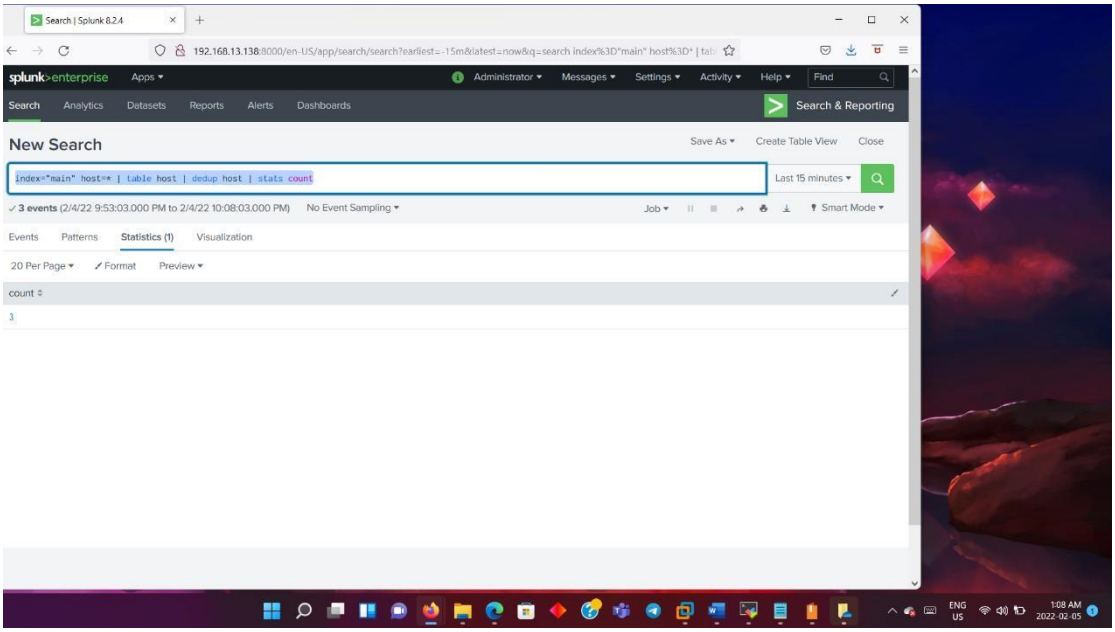


Now we run query 2: index=”main” host=* , this will return all the host as shown above.



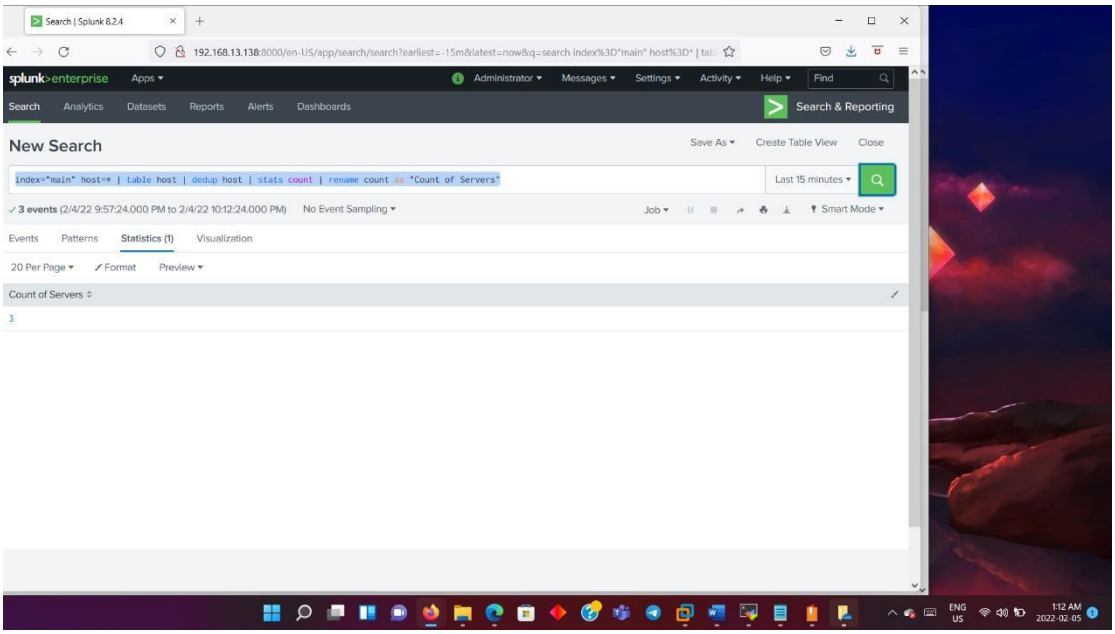
Query 3: To view in table format, we run the above command index=”main” host=* |table host | dedup host

Dedup host is used to avoid duplication.



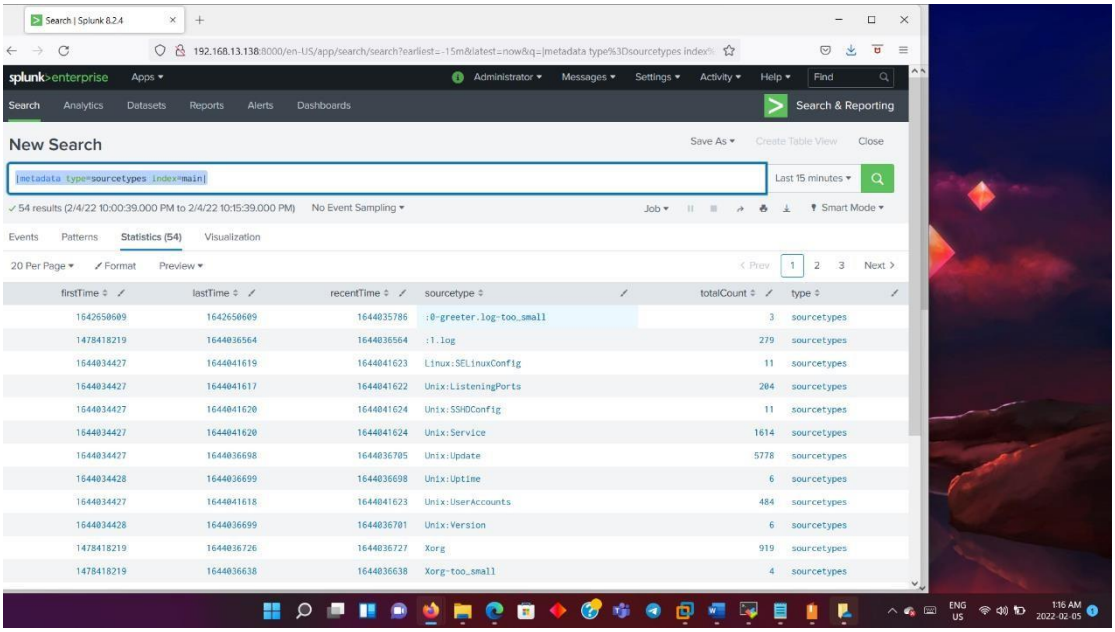
Query 4: index="main" host=* | table host | dedup host | stats count

The output gives the total number of servers. From the above picture it shows 3 servers in total.

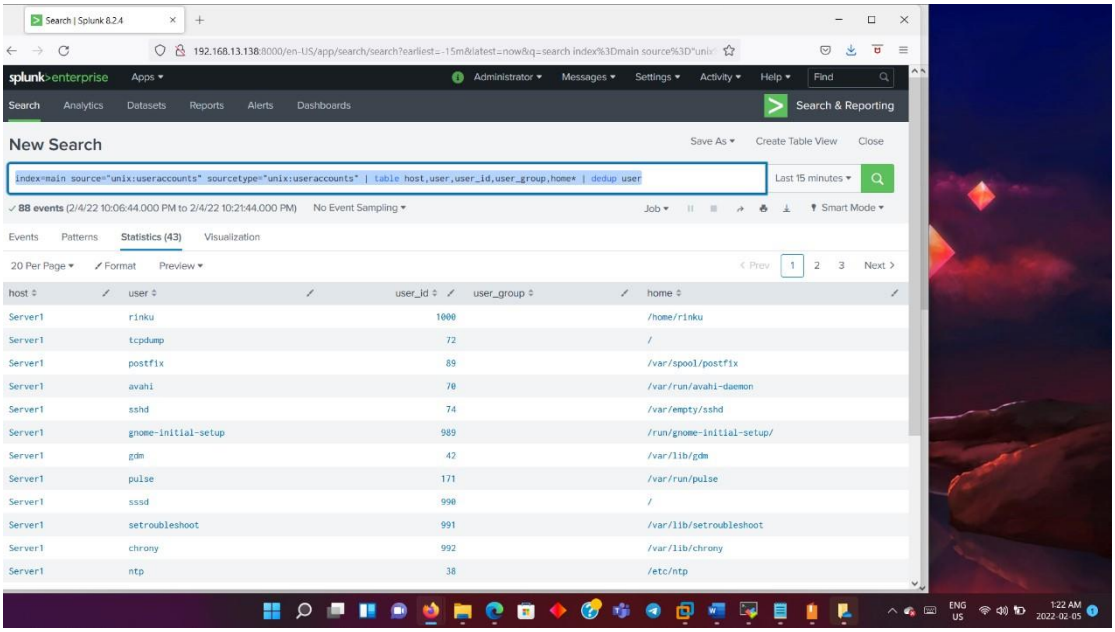


Query 5: index="main" host=* | table host | dedup host | stats count | rename count as "Count of Servers"

This changes the count to “count of servers” as shown in the picture above.

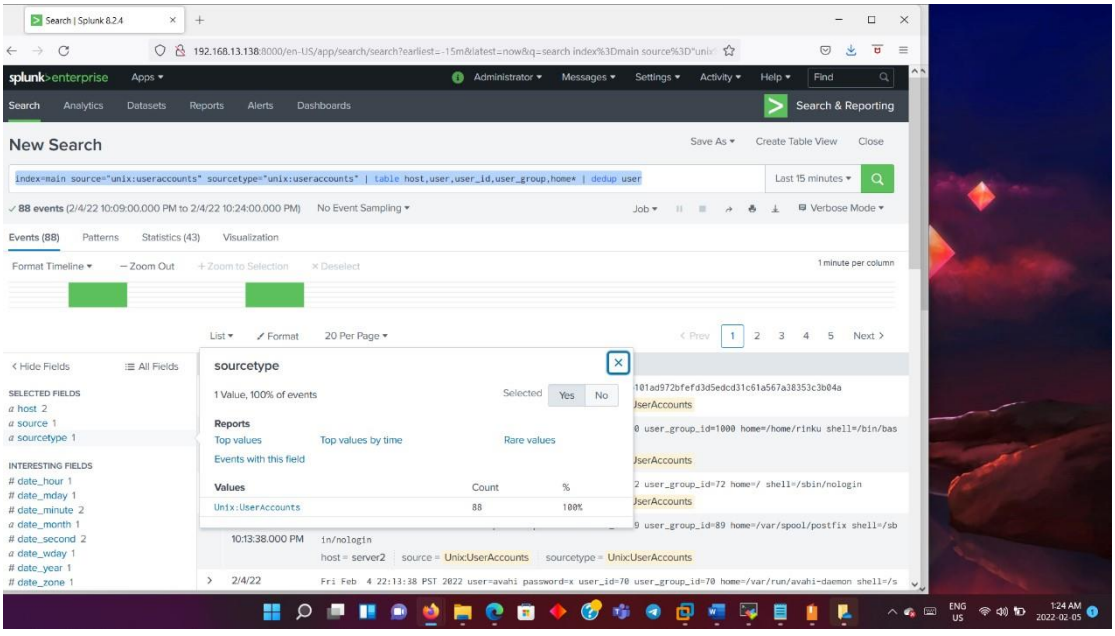


Query 6: |metadata type=sourcetypes index=main|
We get the details of source types for index pointing to main along with the count.

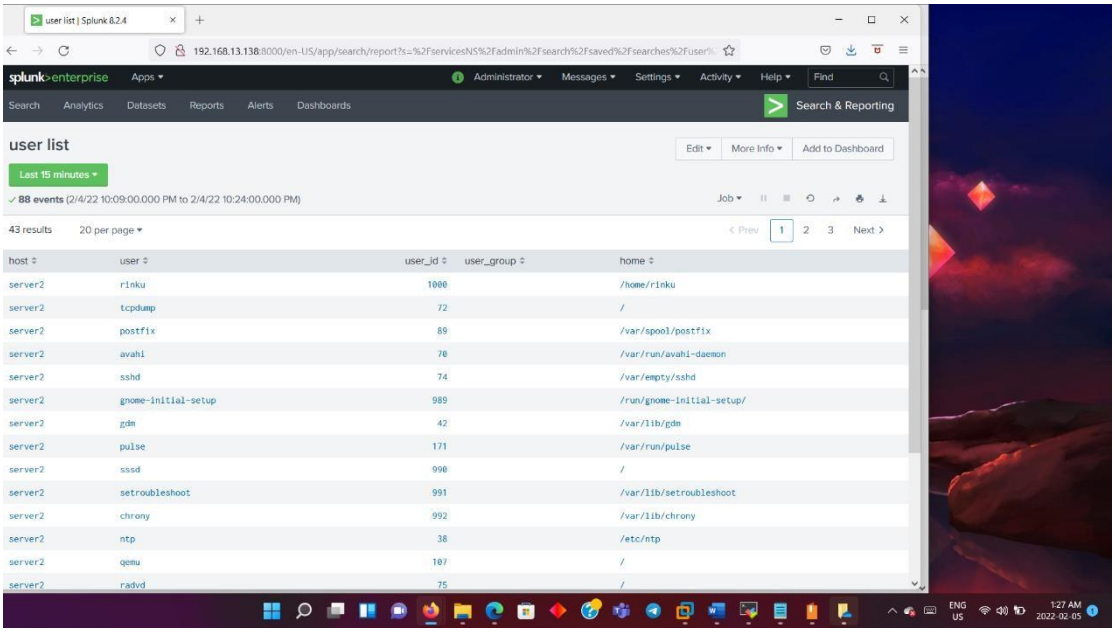


Query 7: index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | table host,user,user_id,user_group,home* | dedup user

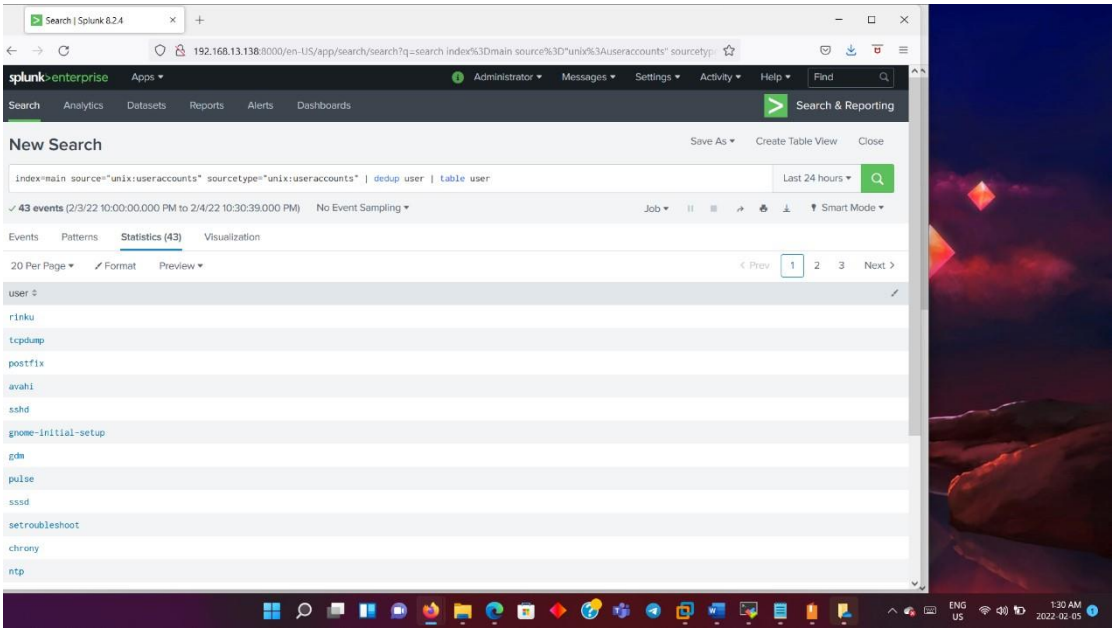
Gives the above details



Choose verbose mode and check source type.

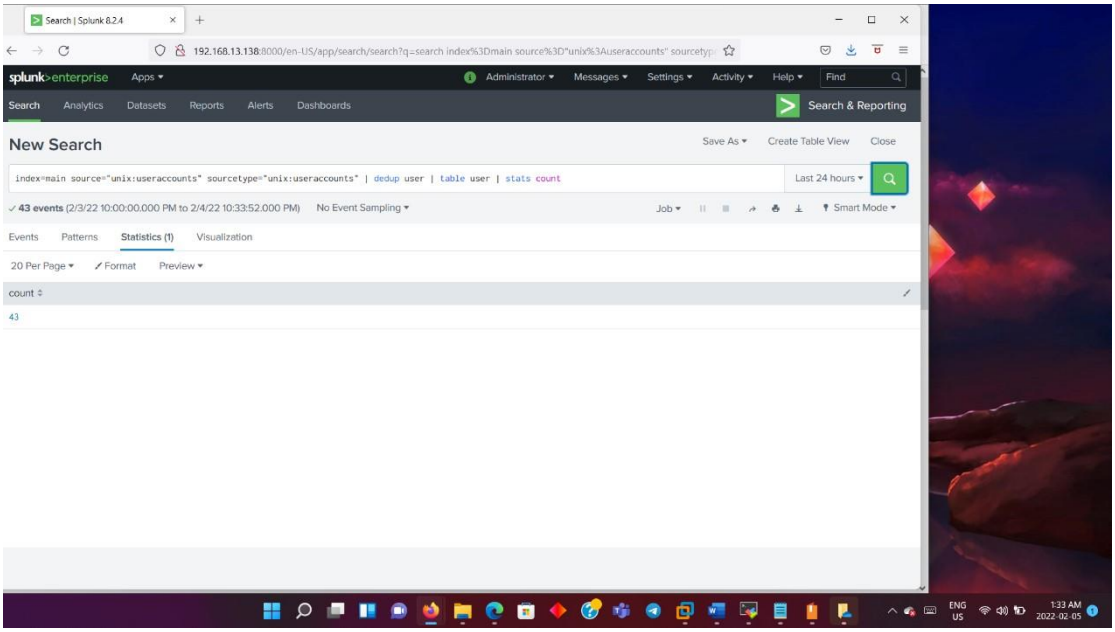


We then save the report and view.



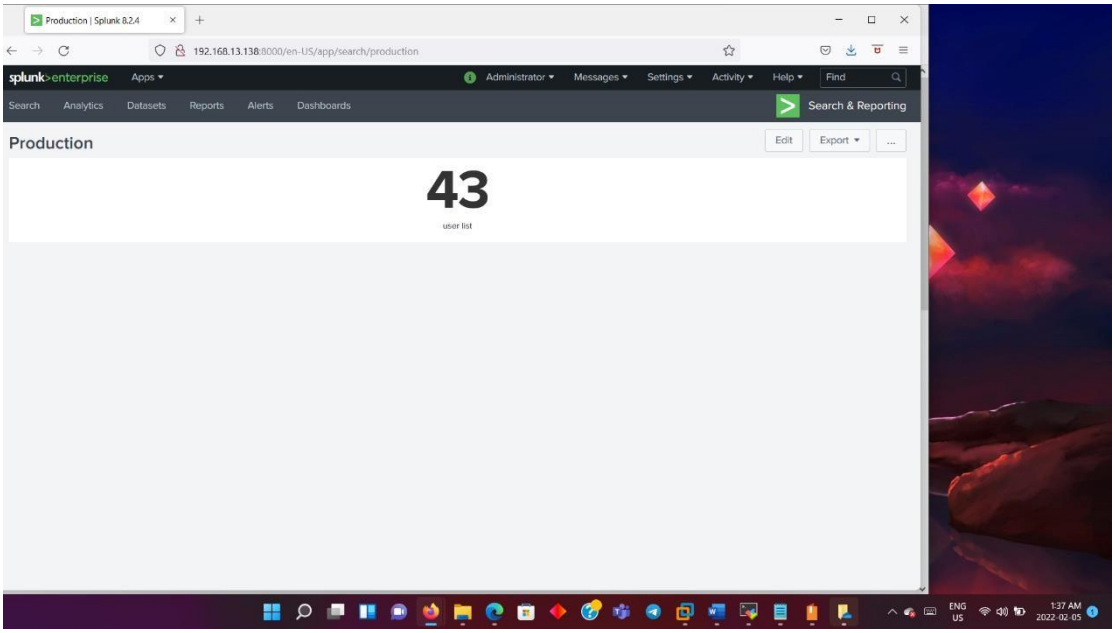
Query: index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | dedup user | table user

Shows list of users in the past 24 hours in smart mode

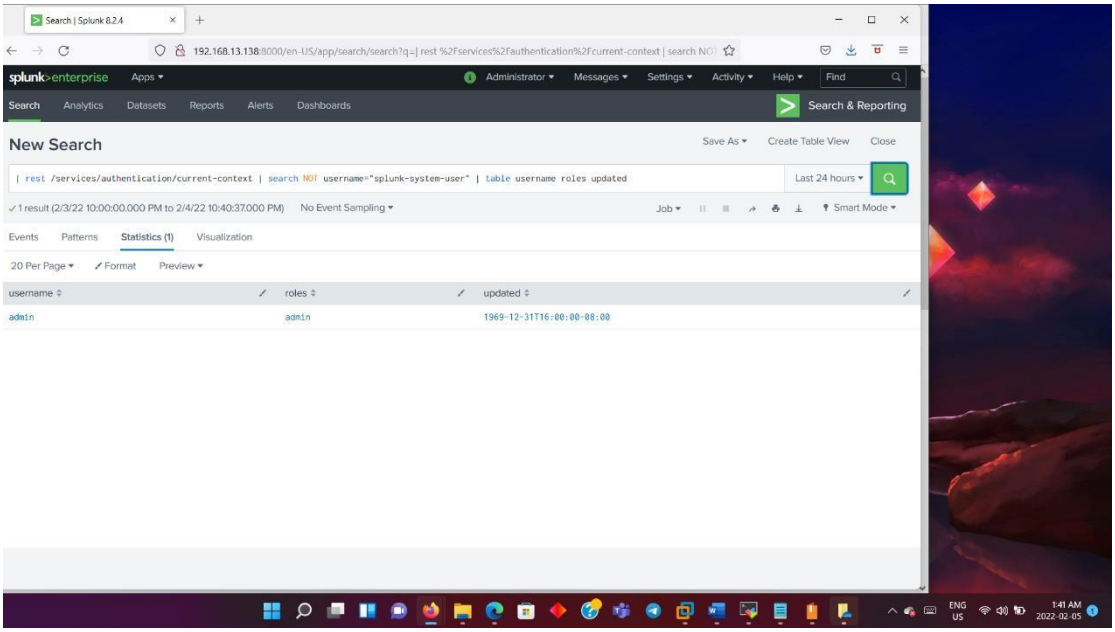


Query: index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | dedup user | table user | stats count

Gives the count of users.

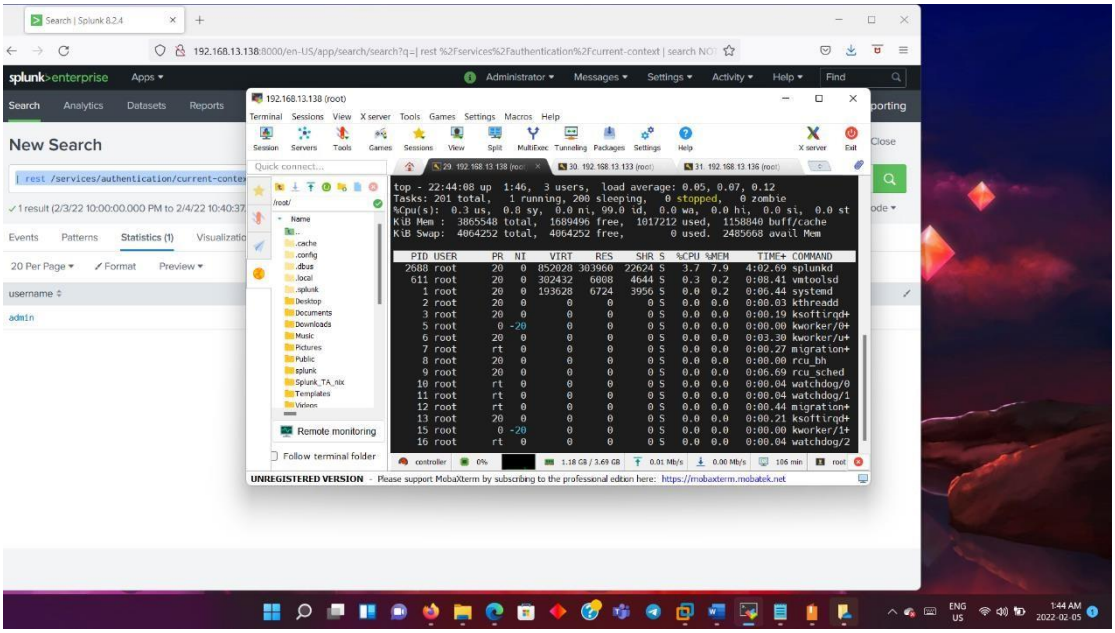


Result of production report saved in dashboard.

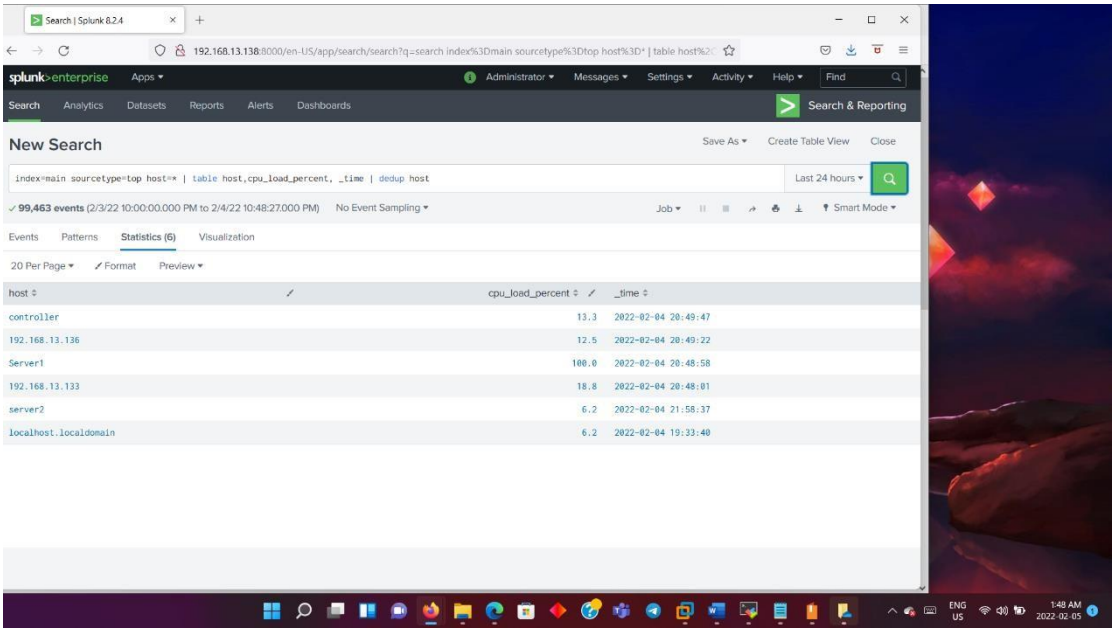


Query: | rest /services/authentication/current-context | search NOT username="splunk-system-user" | table username roles updated

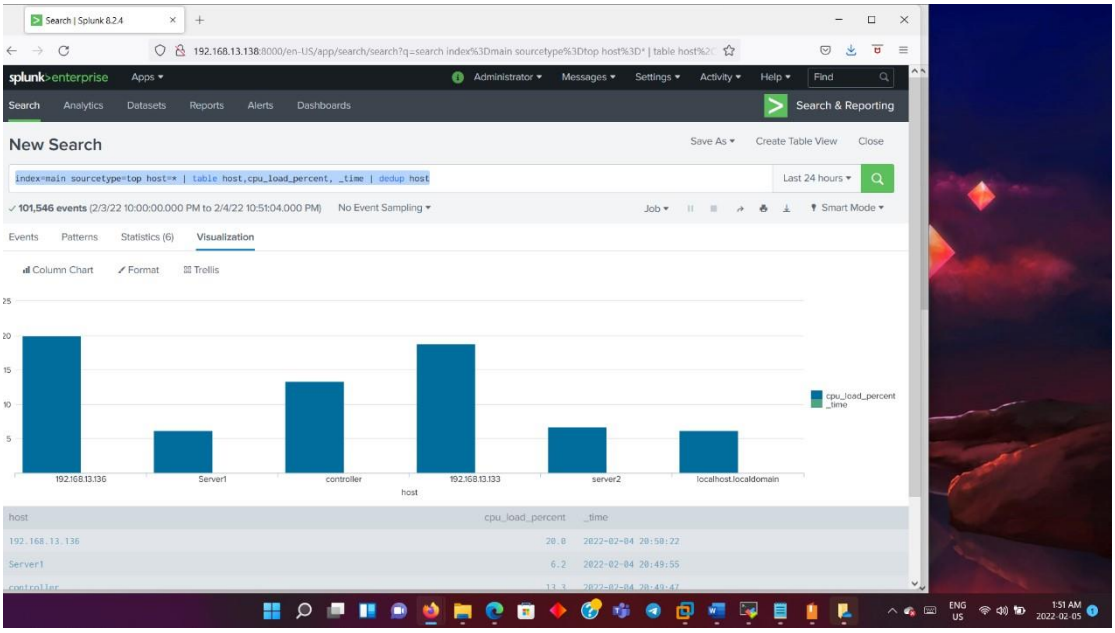
Gives the details of the current authentication user name with the roles updated



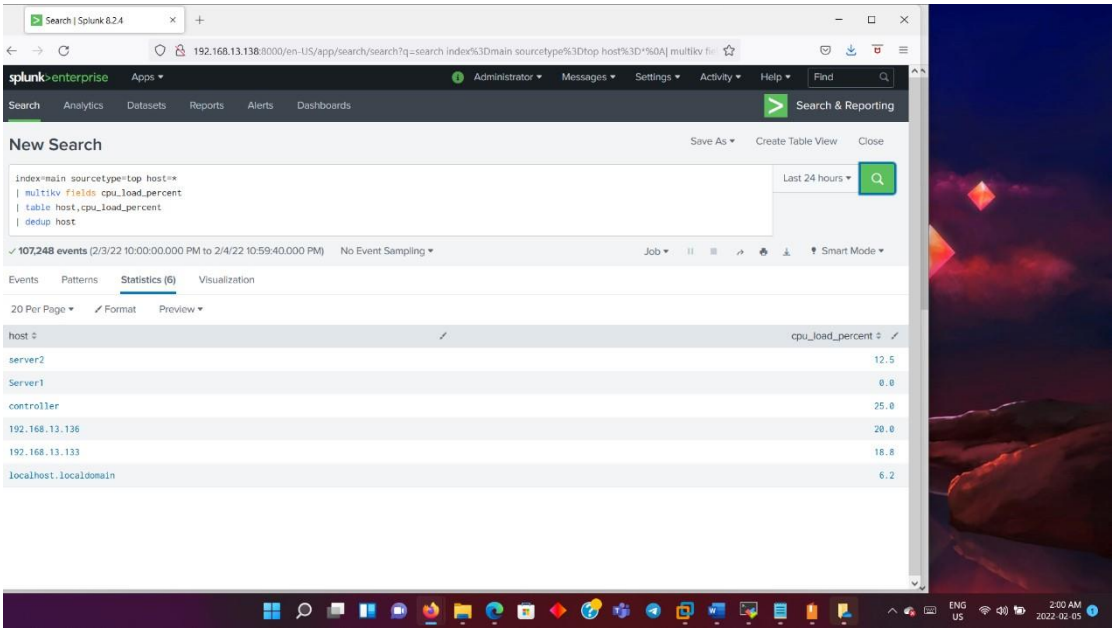
When we execute top command in the terminal, it gives the details of the CPU and time



Query: index=main sourcetype=top host=* | table host,cpu_load_percent, _time | dedup host
Gives the cpu and time details

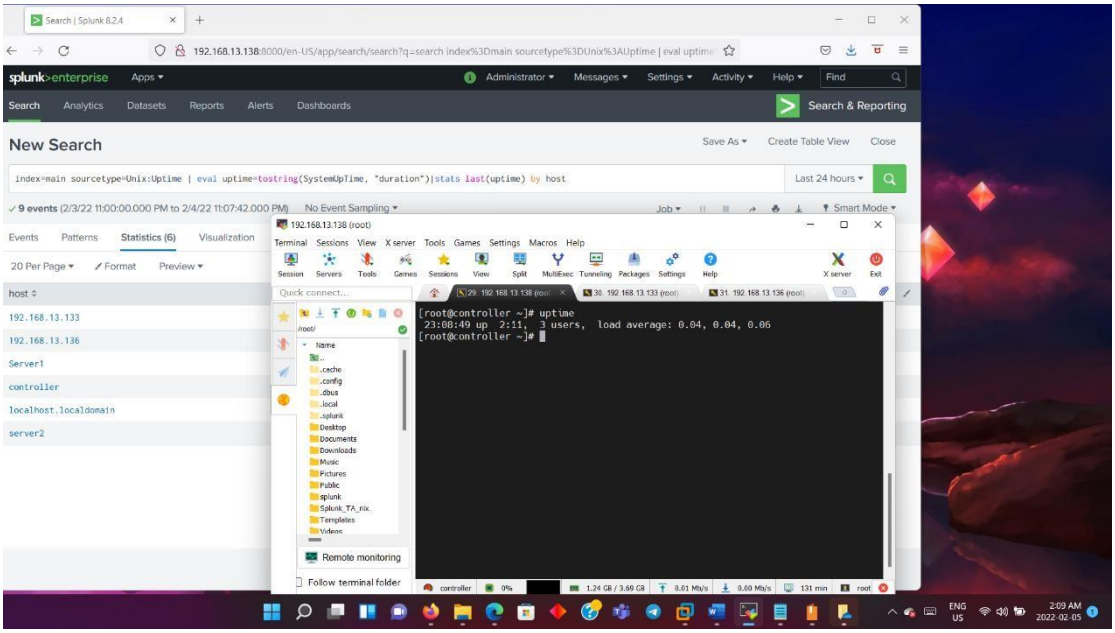


We can select and change the visualization format to pie chart, bar graph etc.



Query: index=main sourcetype=top host=*
| multikv fields cpu_load_percent
| table host,cpu_load_percent
| dedup host

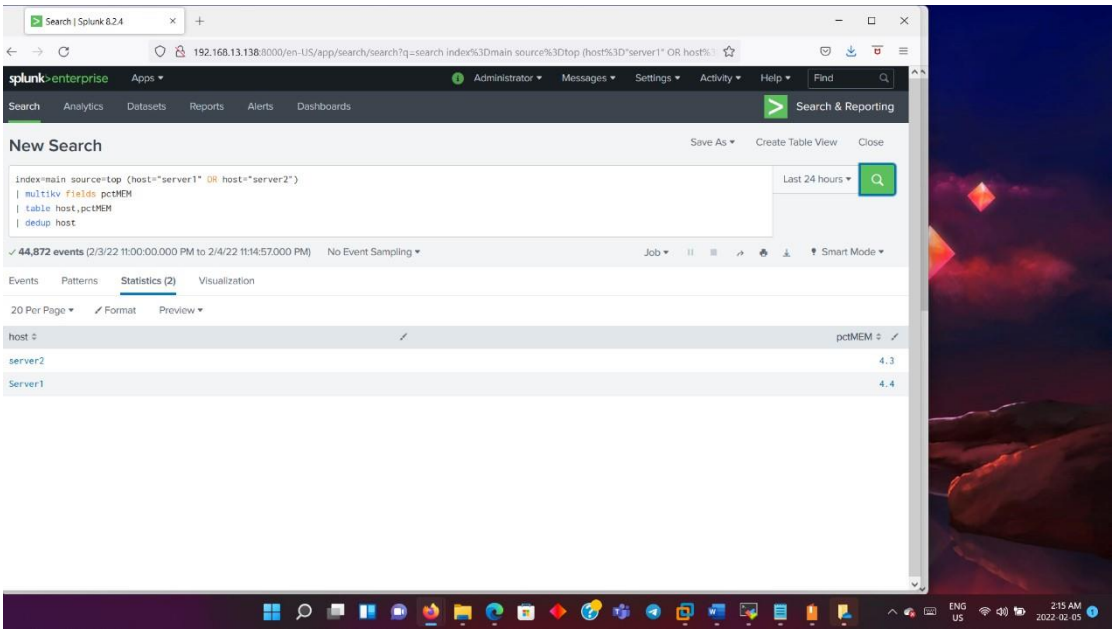
Command used to extract filed and value from the events which are table format.
The title of the table will be assigned as header.
It is a multikv field command.



Query: index=main sourcetype=Unix:Uptime | eval uptime=tostring(SystemUpTime, "duration")|stats last(uptime) by host

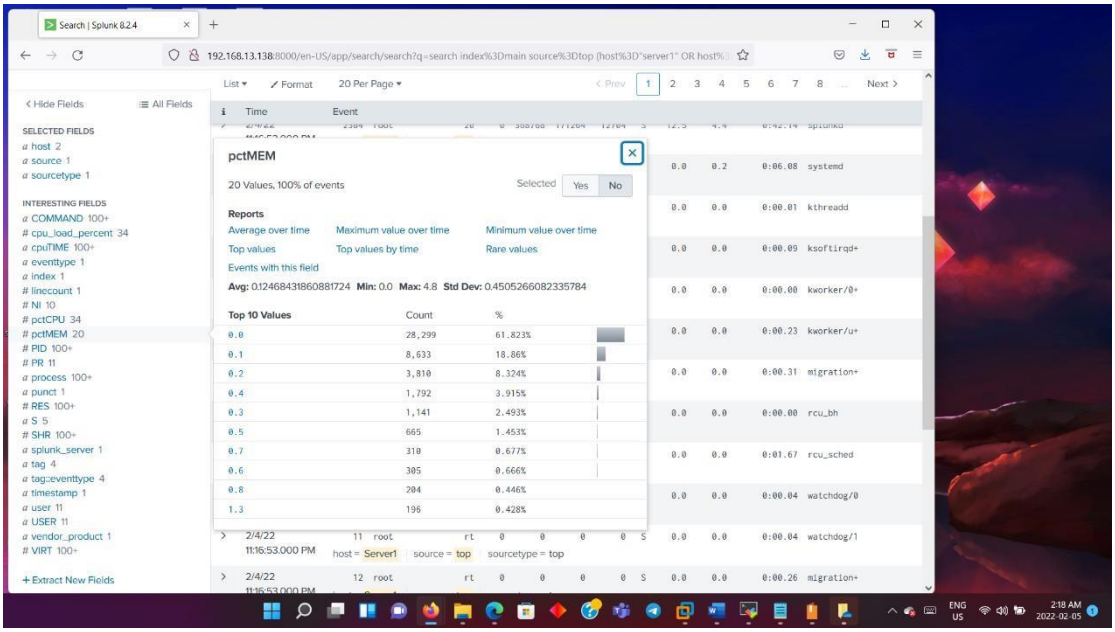
It gives the system uptime

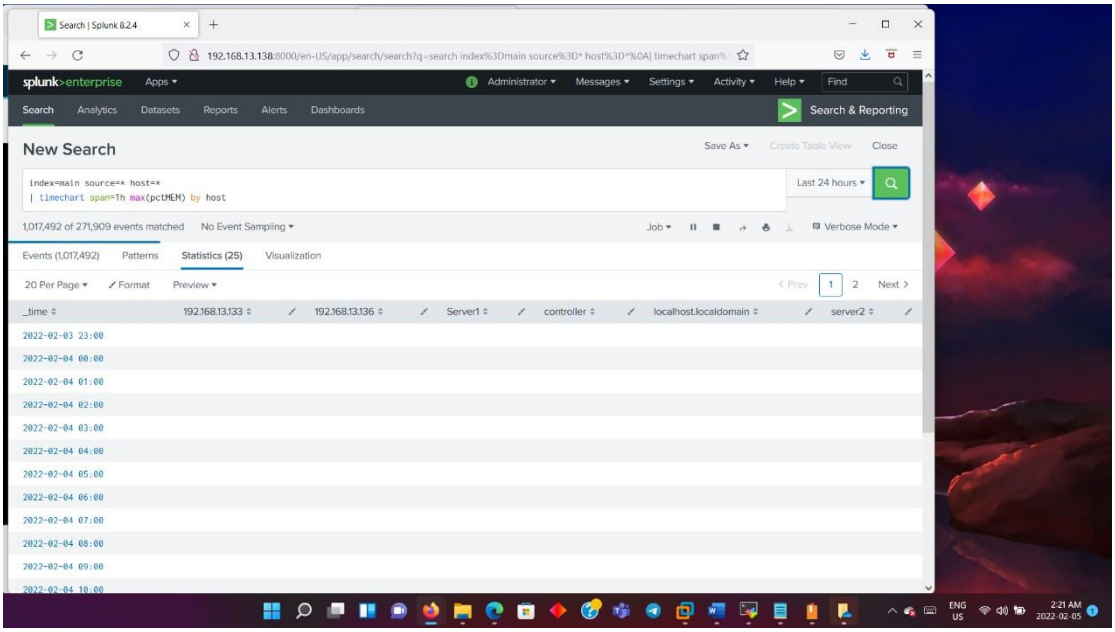
Eval -creates a function



Query: index=main source=top (host="server1" OR host="server2")
| multikv fields pctMEM
| table host,pctMEM
| dedup host

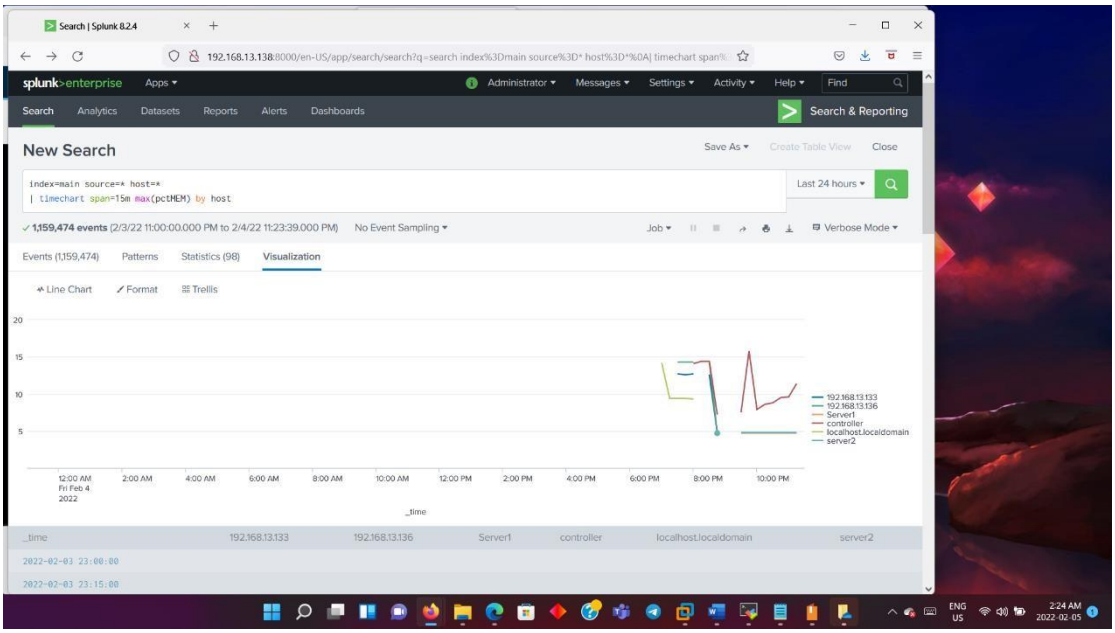
says how much RAM it is going to consume



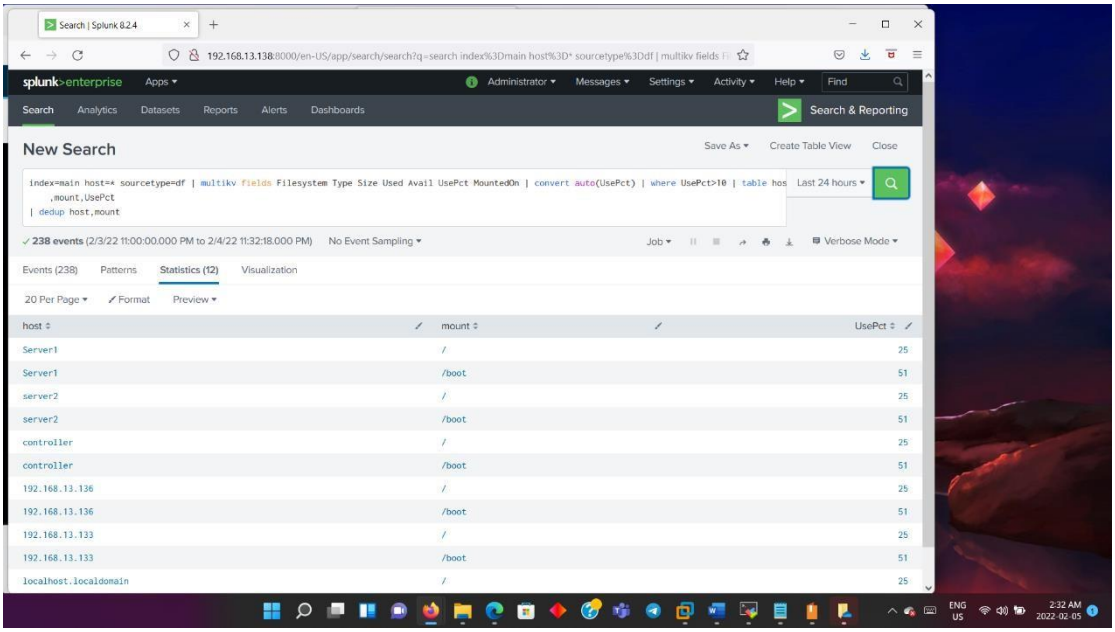


Query: index=main source=* host=*
| timechart span=1h max(pctMEM) by host

To check the memory utilization.

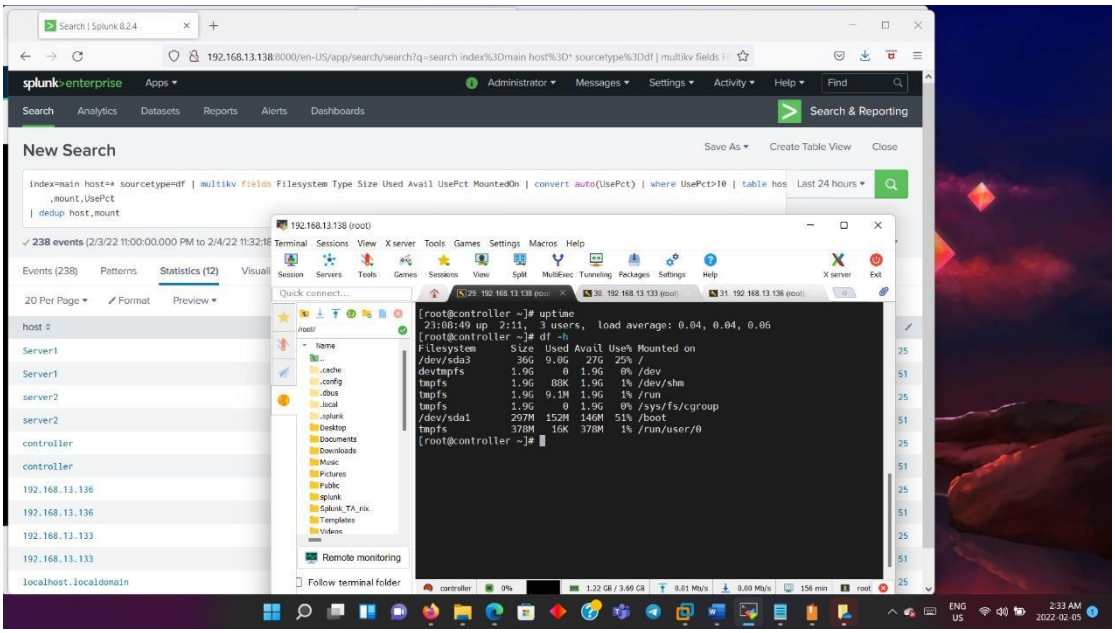


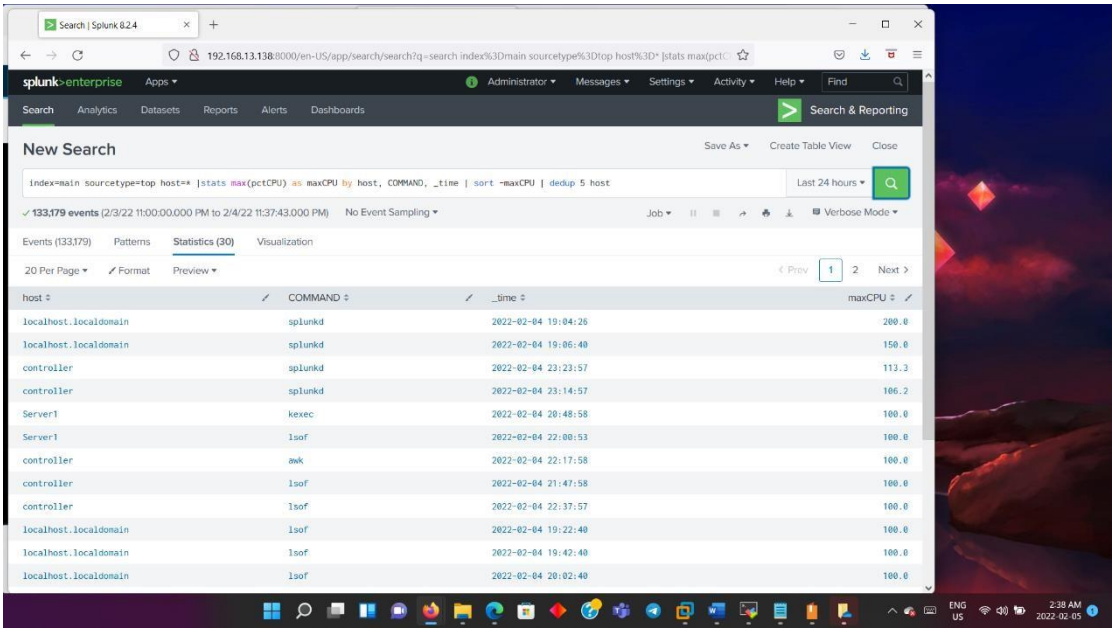
Visualization of the same query with 15 minutes.



Query: index=main host=* sourcetype=df | multikv fields Filesystem Type Size Used Avail UsePct MountedOn | convert auto(UsePct) | where UsePct>10 | table host,mount,UsePct | dedup host,mount

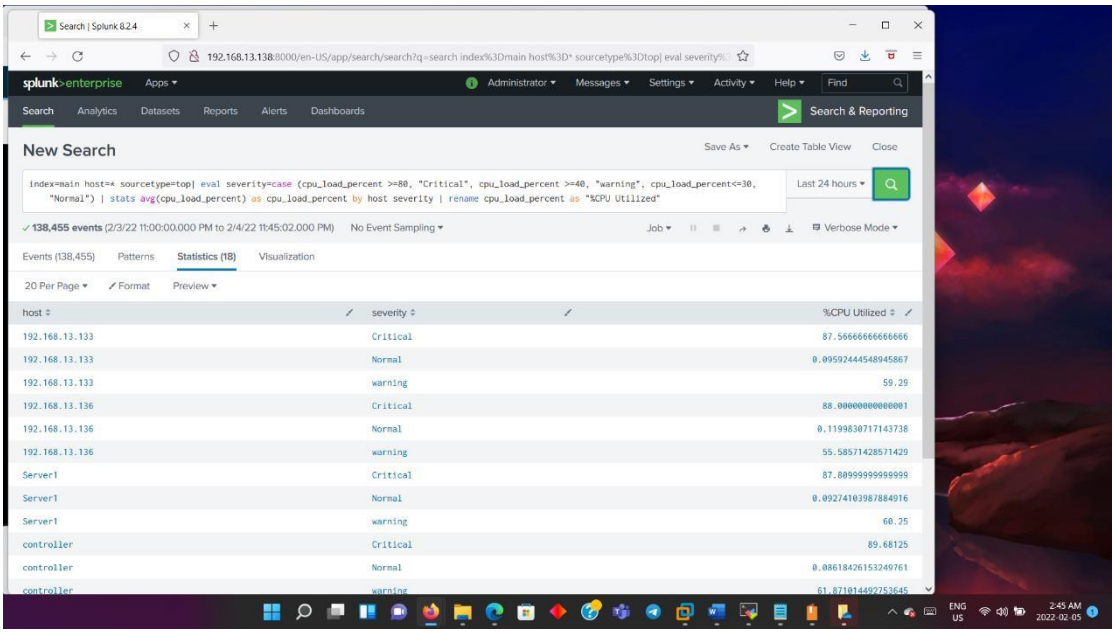
gives the used percentage



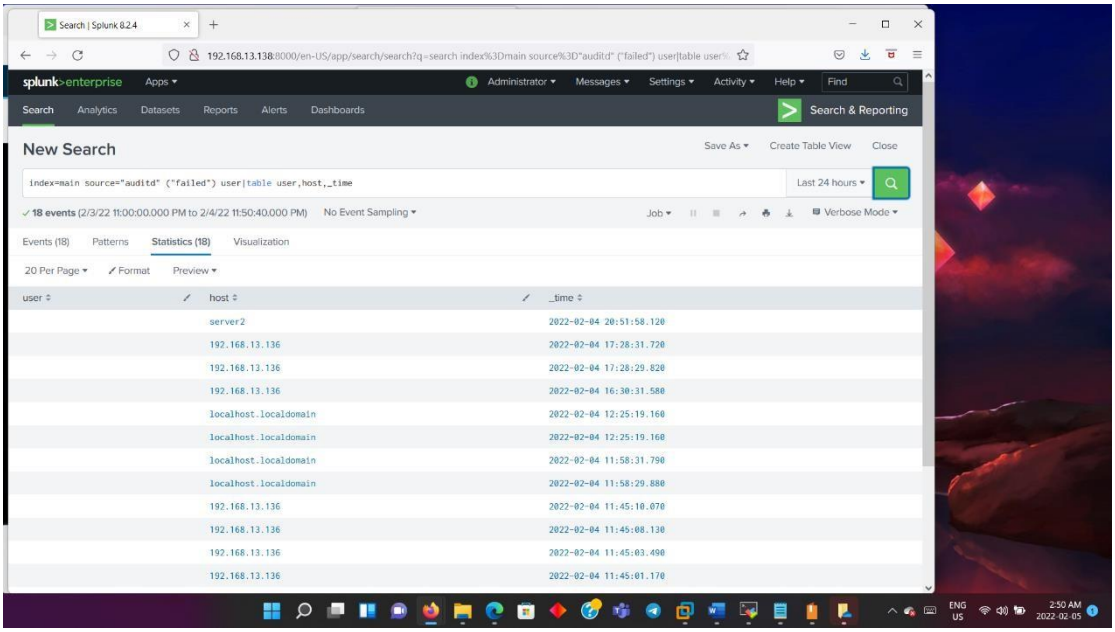


Query: index=main sourcetype=top host=* |stats max(pctCPU) as maxCPU by host, COMMAND, _time | sort -maxCPU | dedup 5 host

Top 5 cpu consuming process and will sort it.

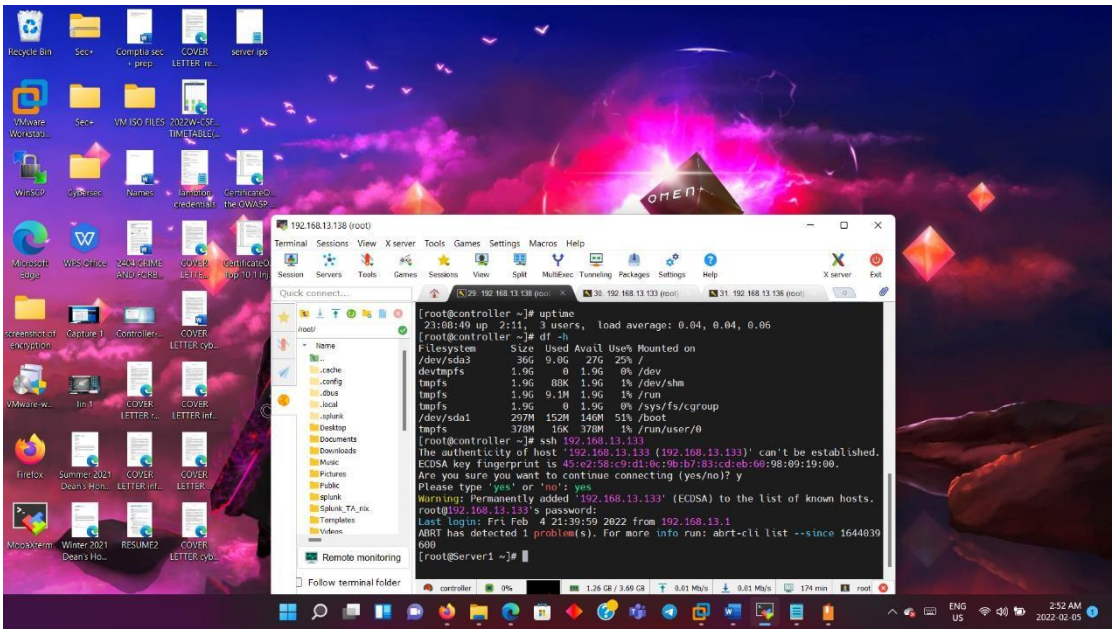


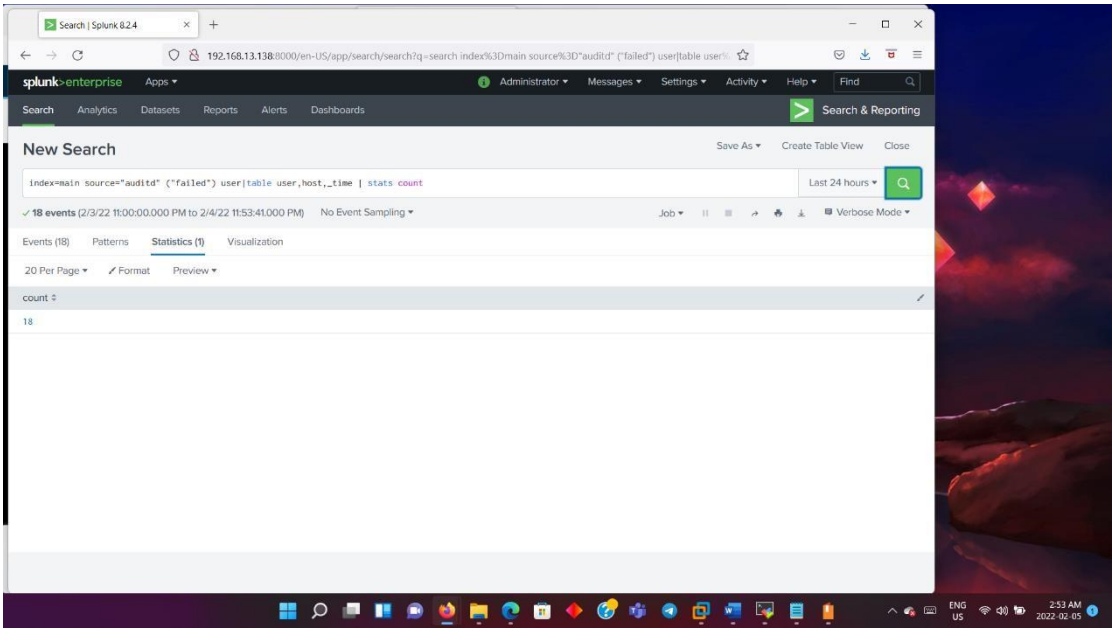
Query: index=main host=* sourcetype=top| eval severity=case (cpu_load_percent >=80, "Critical", cpu_load_percent >=40, "warning", cpu_load_percent<=30, "Normal") | stats avg(cpu_load_percent) as cpu_load_percent by host severity | rename cpu_load_percent as "%CPU Utilized"



Query: index=main source="auditd" ("failed") user|table user,host,_time

Shows failed ssh login





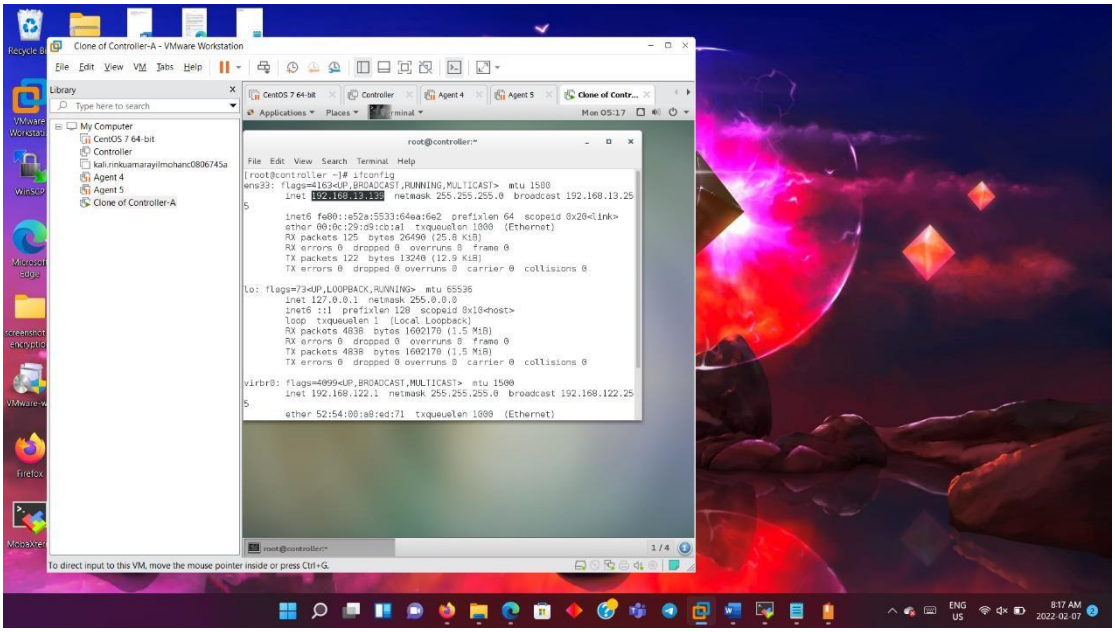
Query: index=main source="auditd" ("failed") user|table user,host,_time | stats count

It gives the stat count of the failed ssh login

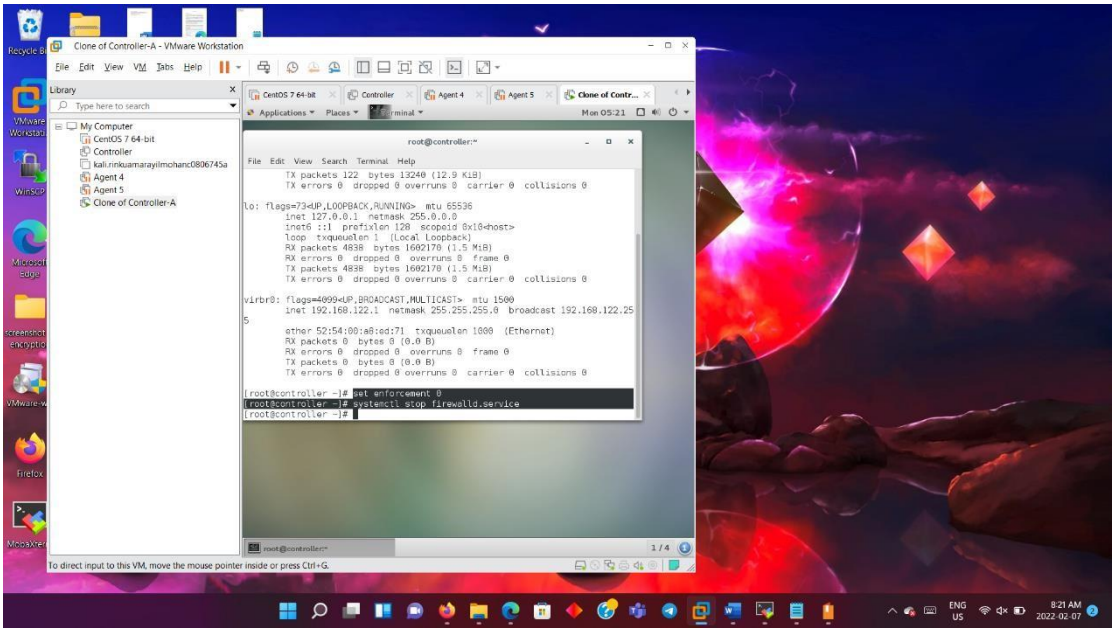
\

Part 1b

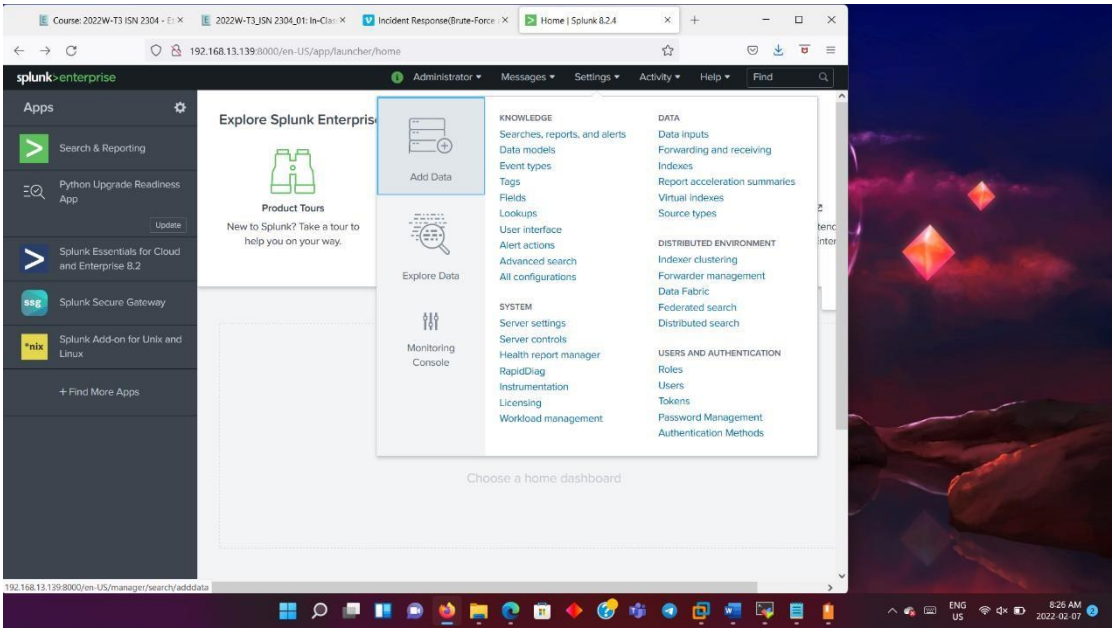
Incident response Brute-Force attack

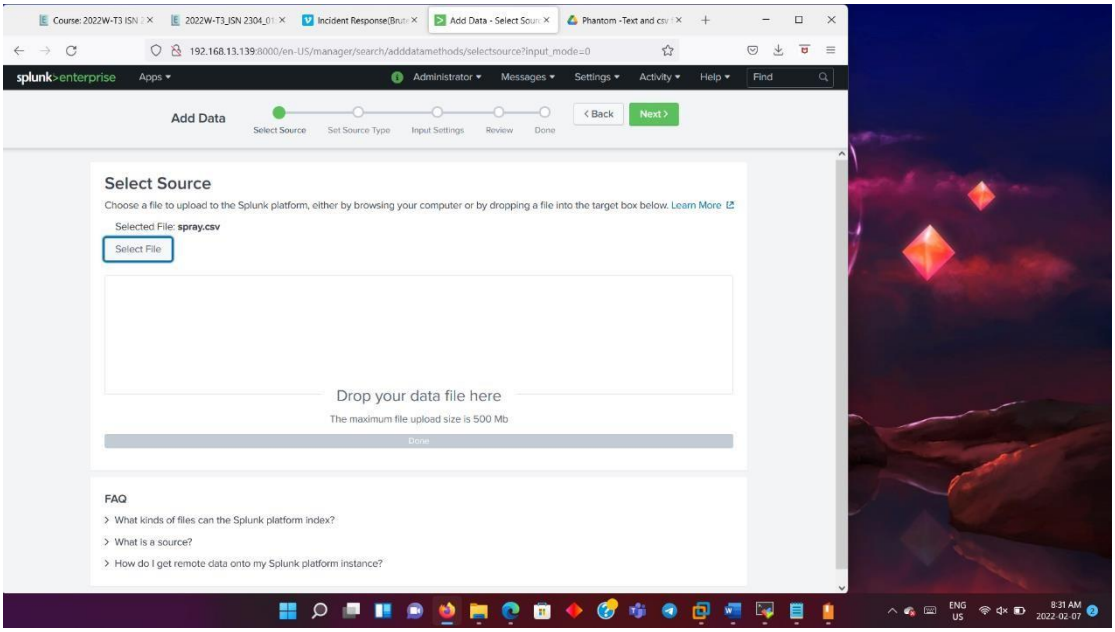


Created clone of controller and named in Clone of controller A
Opened a new terminal and checked the Ip address - 192.168.13.139
We also ensure that the firewall is stopped by typing: # set enforcement 0
systemctl stop firewalld.service

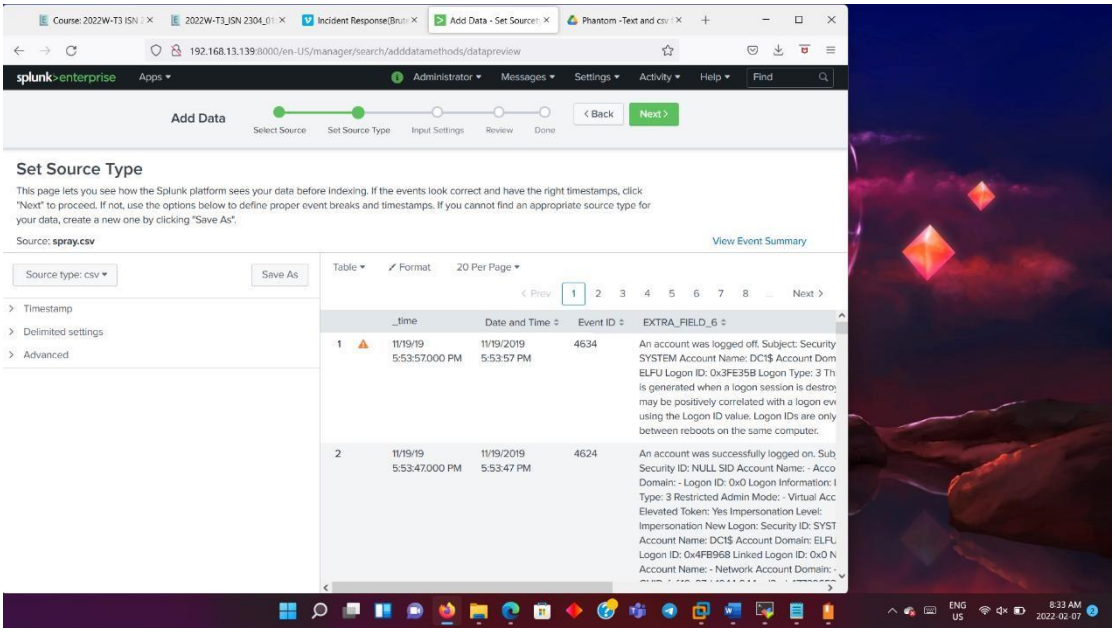


Then login to Splunk we with the newly cloned controller A Ip address - 192.168.13.139
And then add data manually.

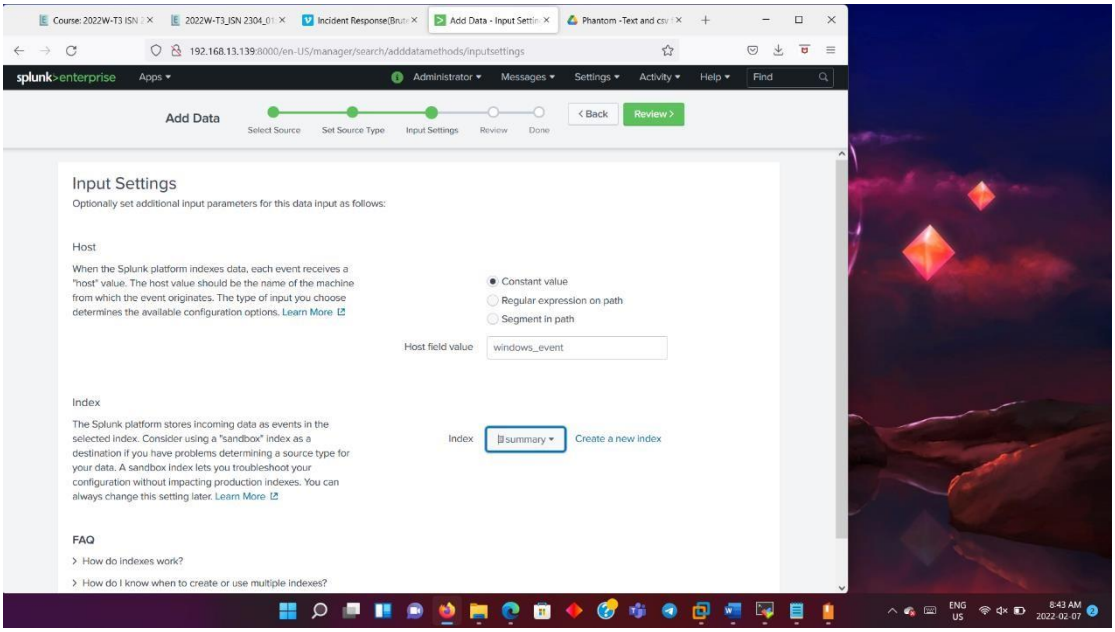




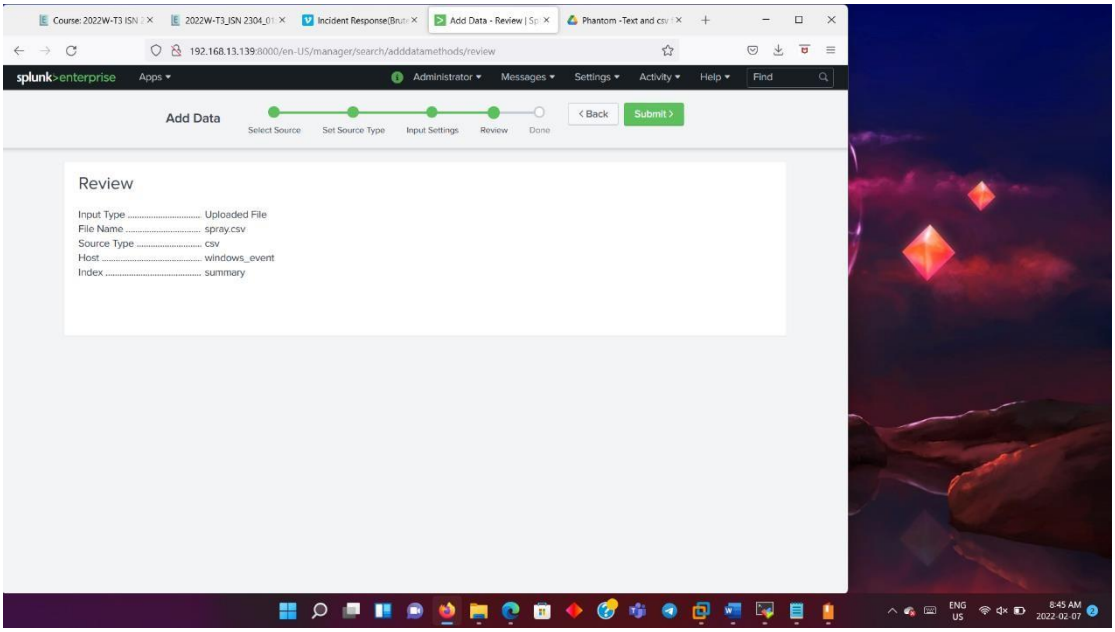
Now we have uploaded the data spray.csv file and then click on next.



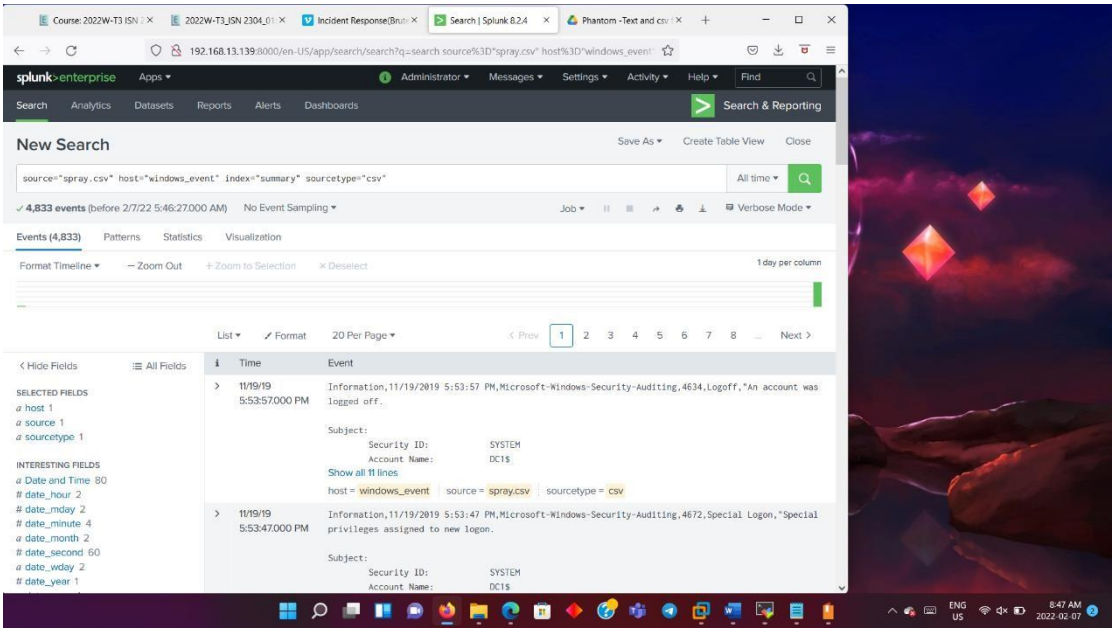
In this we get the source type, timestamp details. Now we click on next.

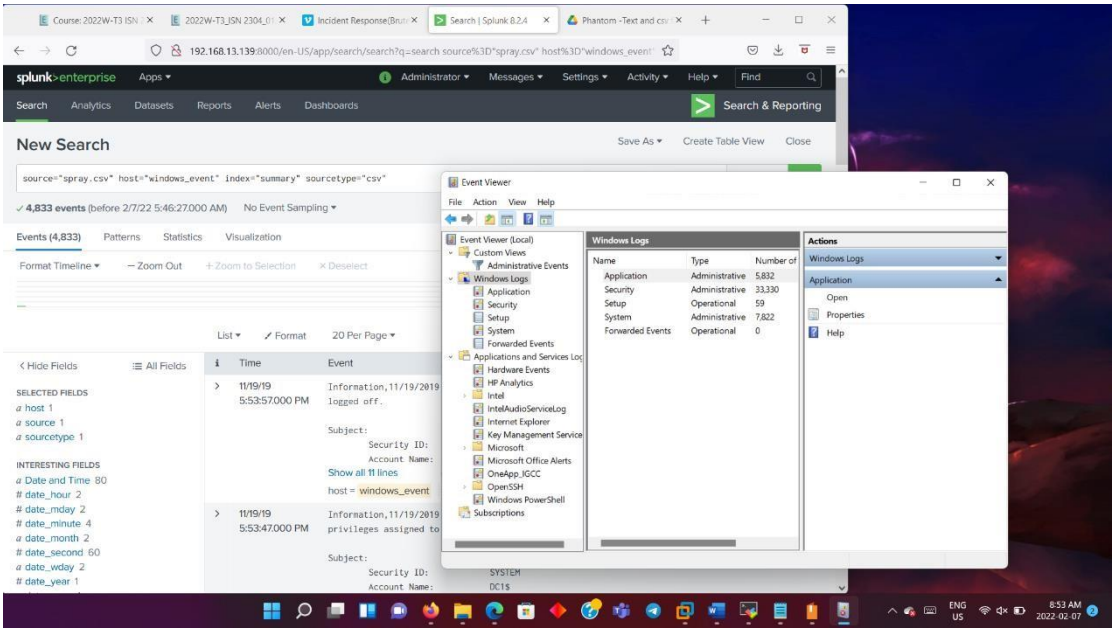


Enter the Host field value as windows_event and select the index as summary and click on review.



We get the above details, we then click on submit and then select start search, we get the page shown below.

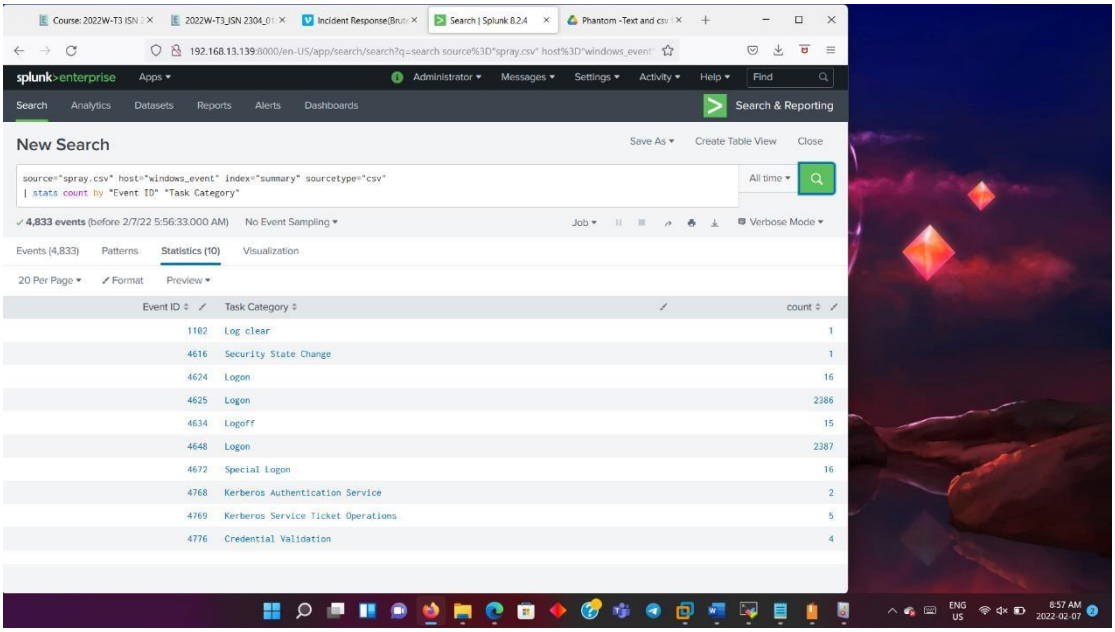




We can review with the windows event, event viewer, gives the same details.

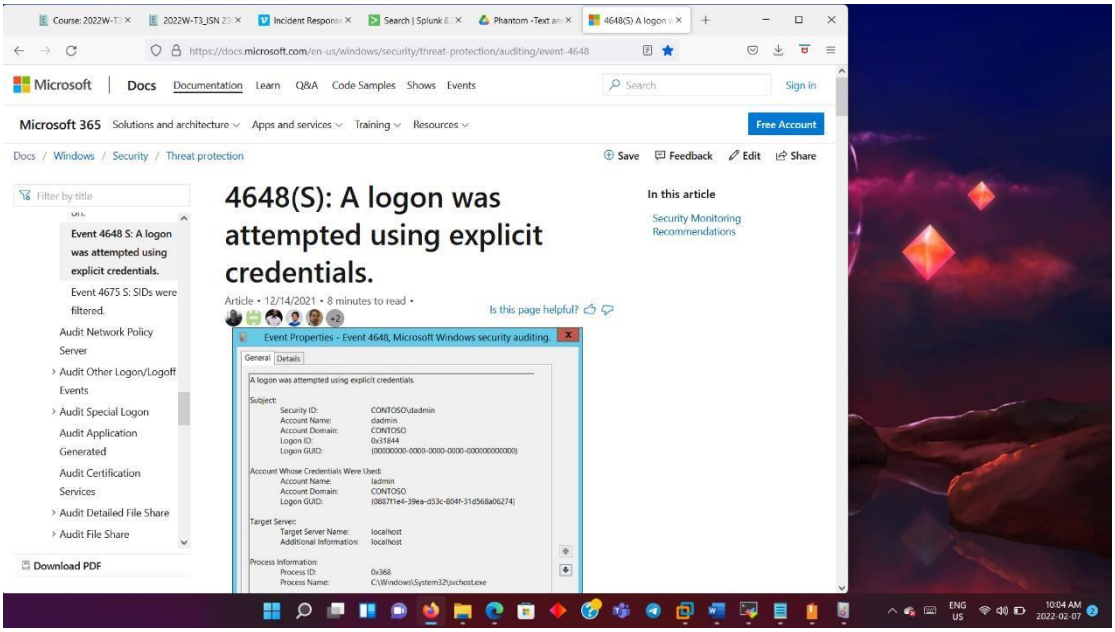
Now search for query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" | stats count by "Event ID" "Task Category"

Search the above query selecting all time.

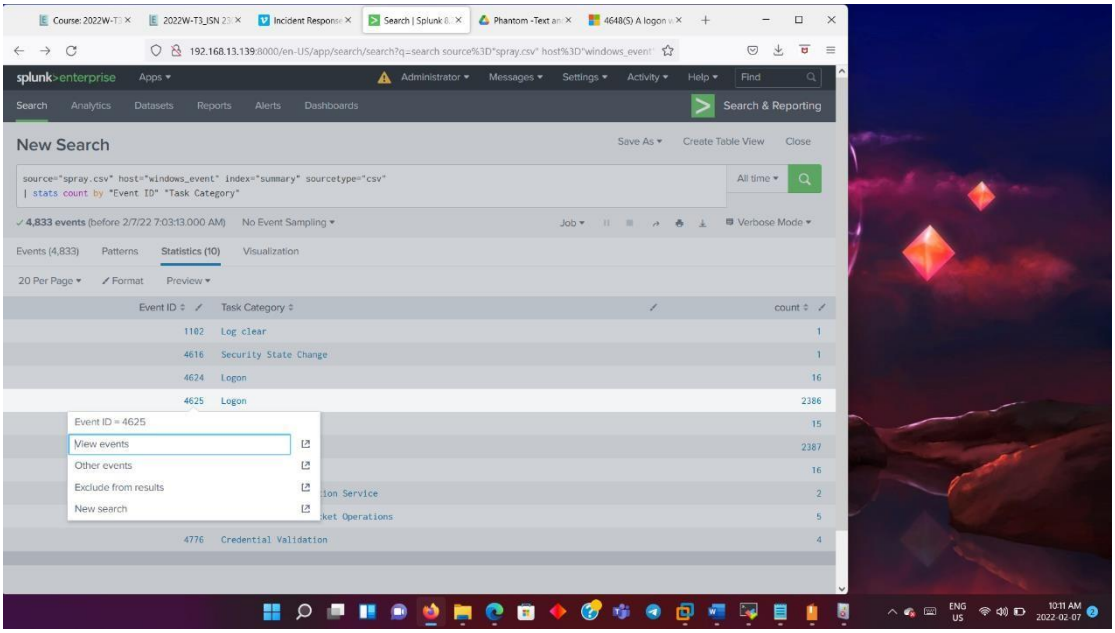


We can see each event ID along with the task category and count.

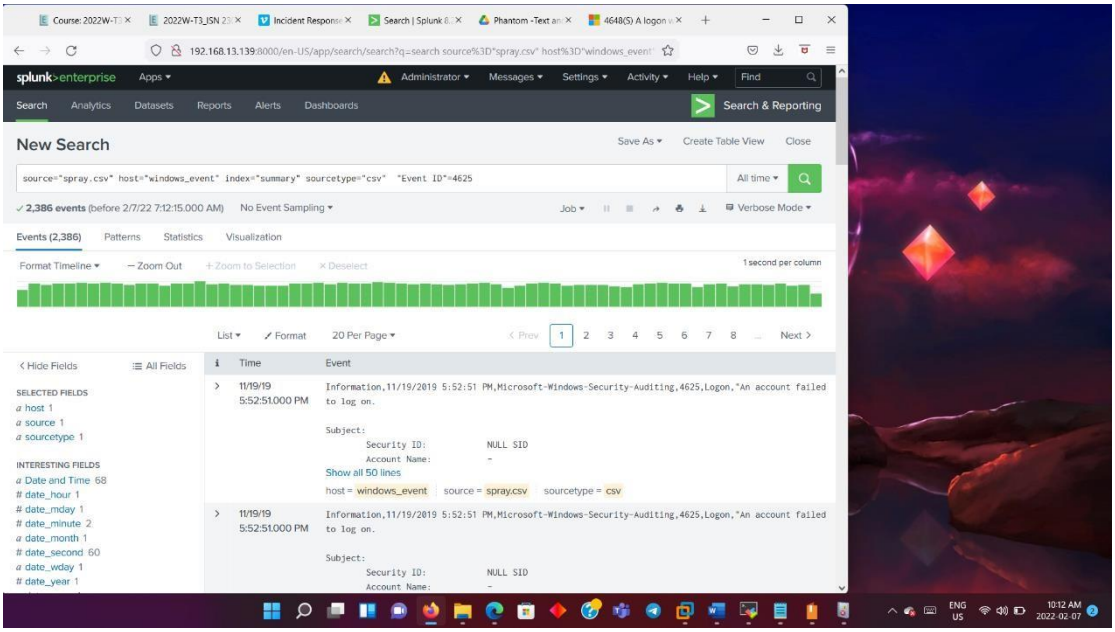
Event ID is explained in detail in the page below along with the recommendation.

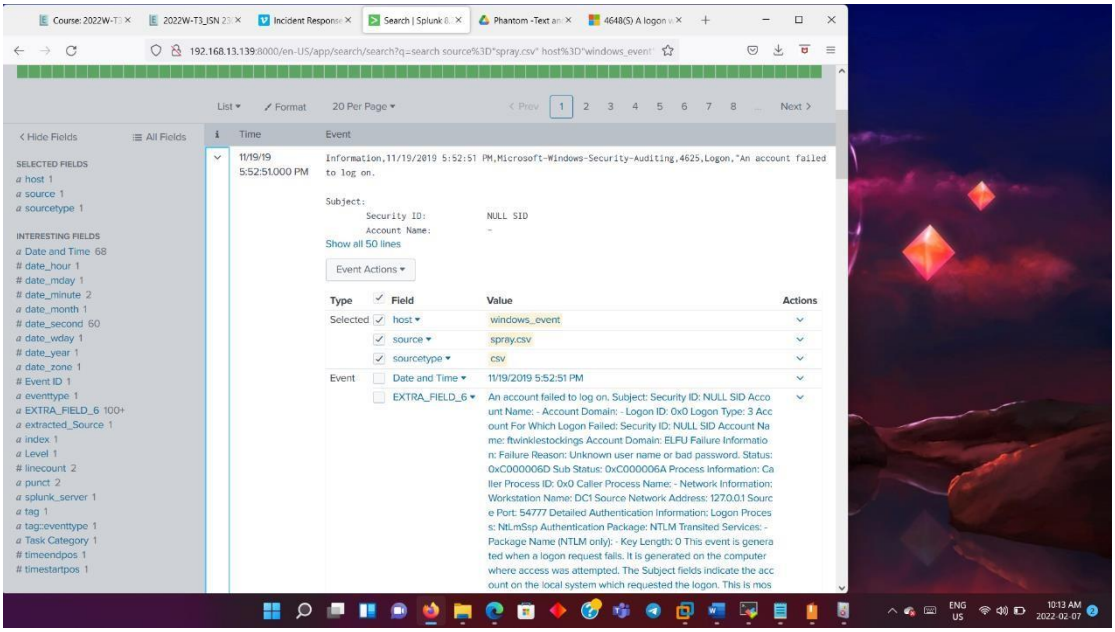


We need to monitor all windows security event like logon activity 4624, 4625- logon failure. If account is compromised or not. To check if brute force incident has occurred or no.

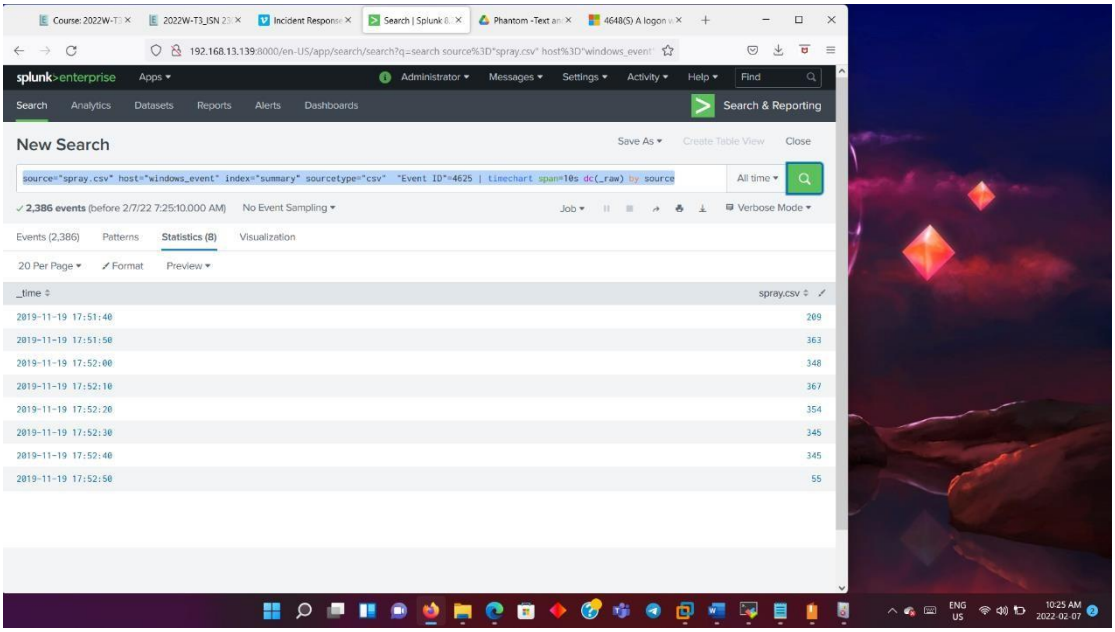


For further details, hover mouse on the event ID 4625 and select view events.



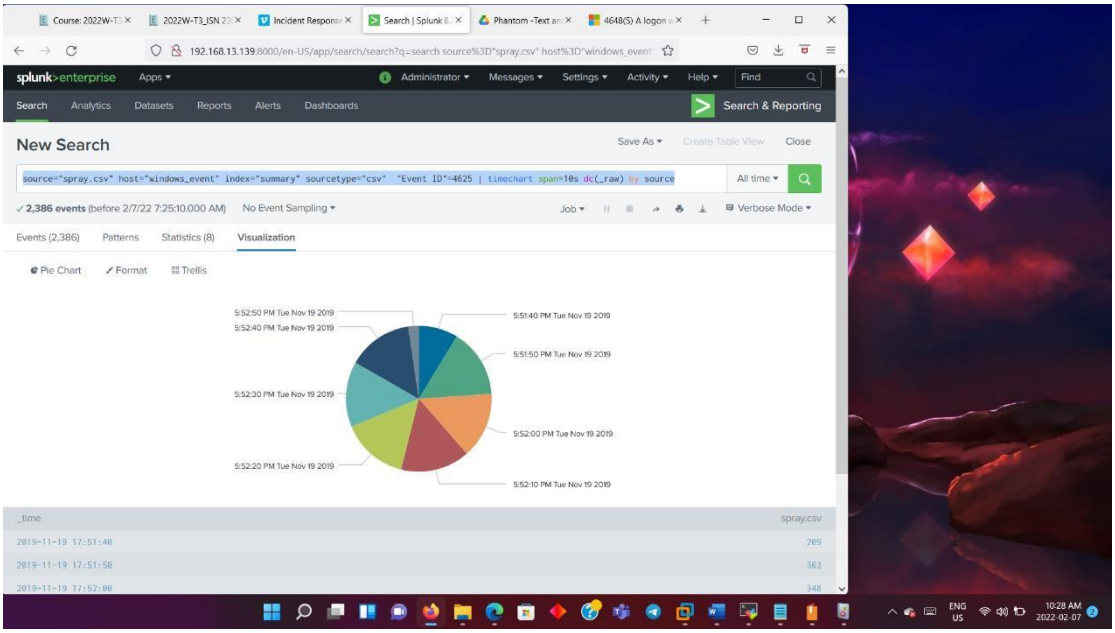


We get the details of host, source, source type etc.

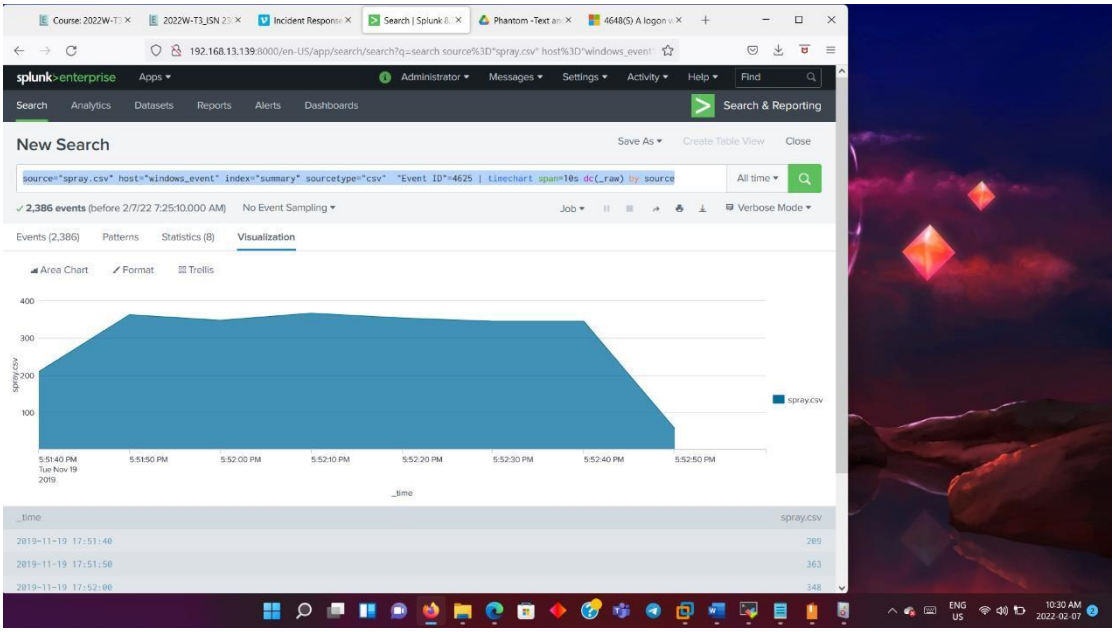


Query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" "Event ID"=4625 | timechart span=10s dc(_raw) by source

Gives the details of the brute force attack that took place in the event id given.

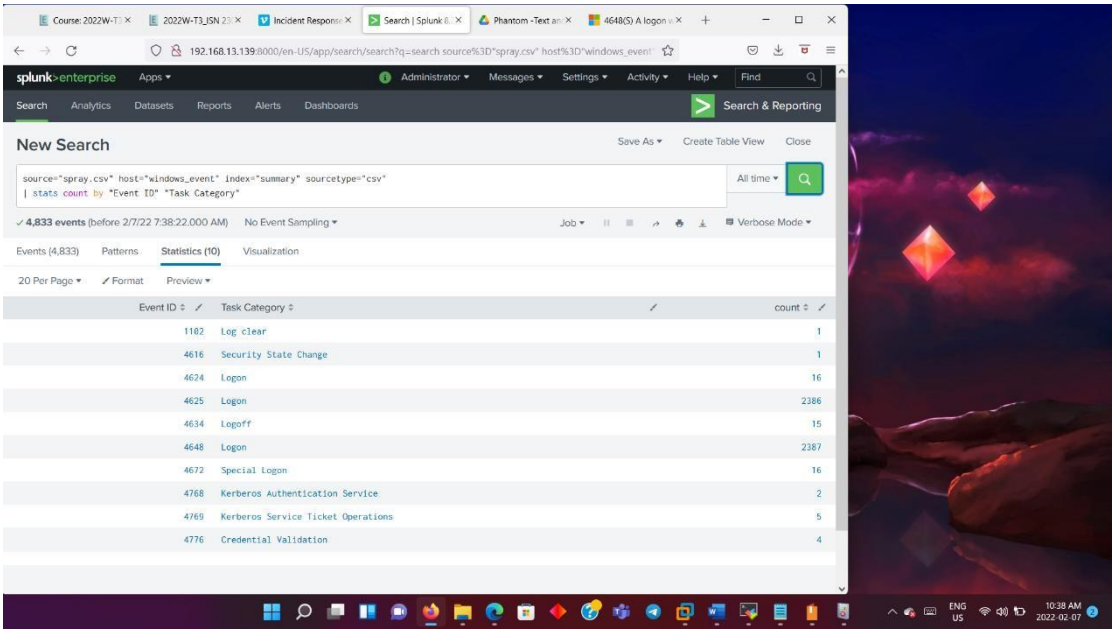


We can see the result in the visualization as pie chart as well.



Above shows the same view in area chart to show the progress of brute force attack.

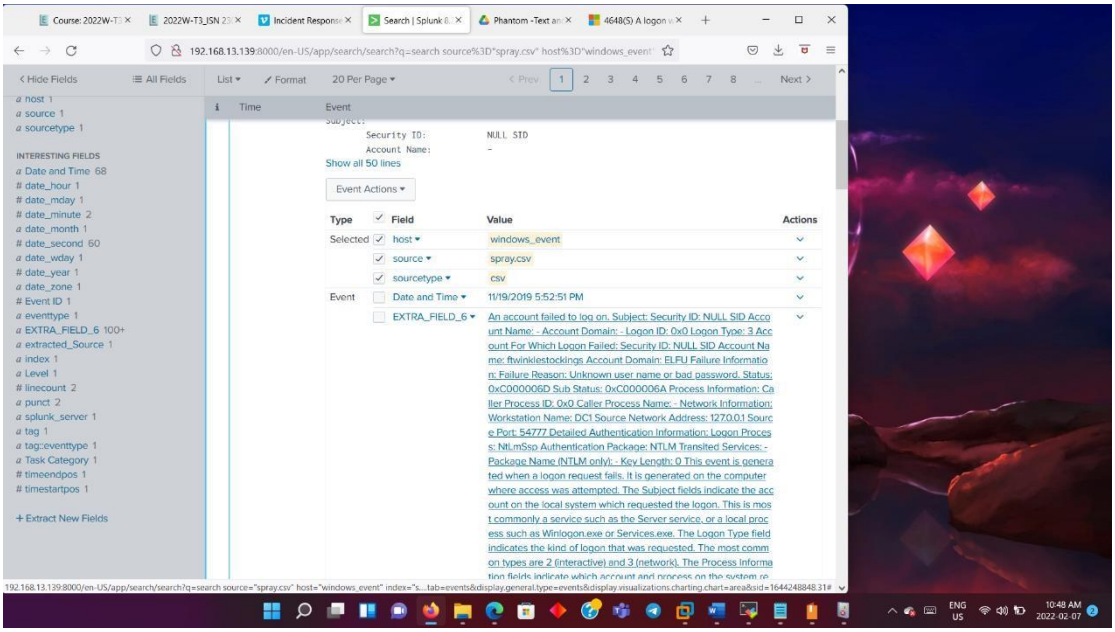
We have observed the count of high number of failed login event compared with success logon event.



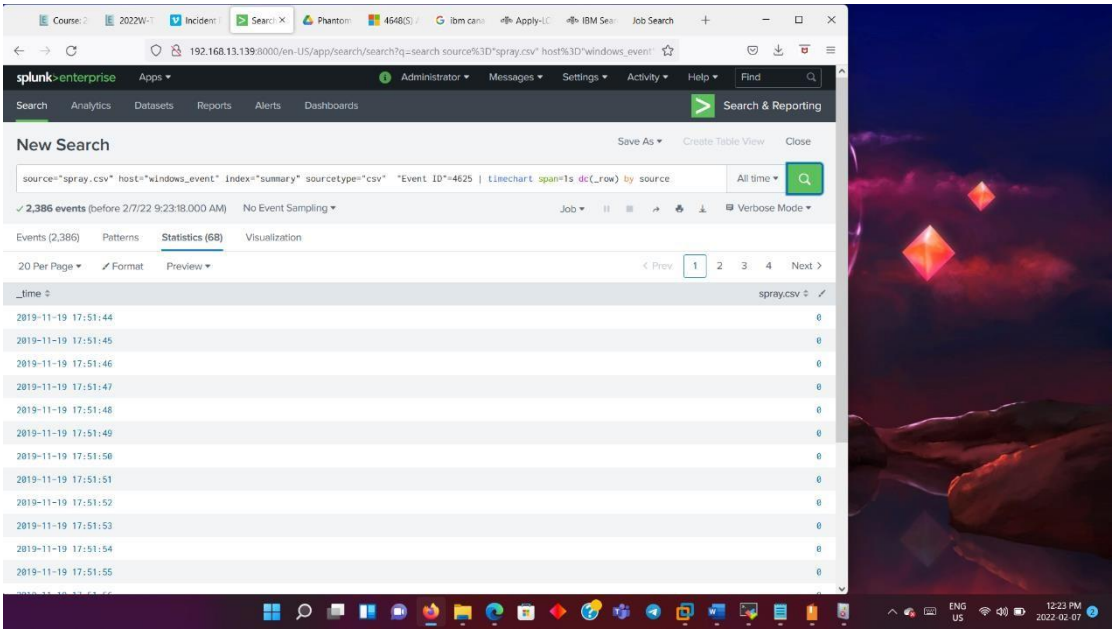
Query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv"
| stats count by "Event ID" "Task Category"

We see that the successful even was much less compared to failed login. Also failed logon seems to be automated script because events are distributed over short time span.

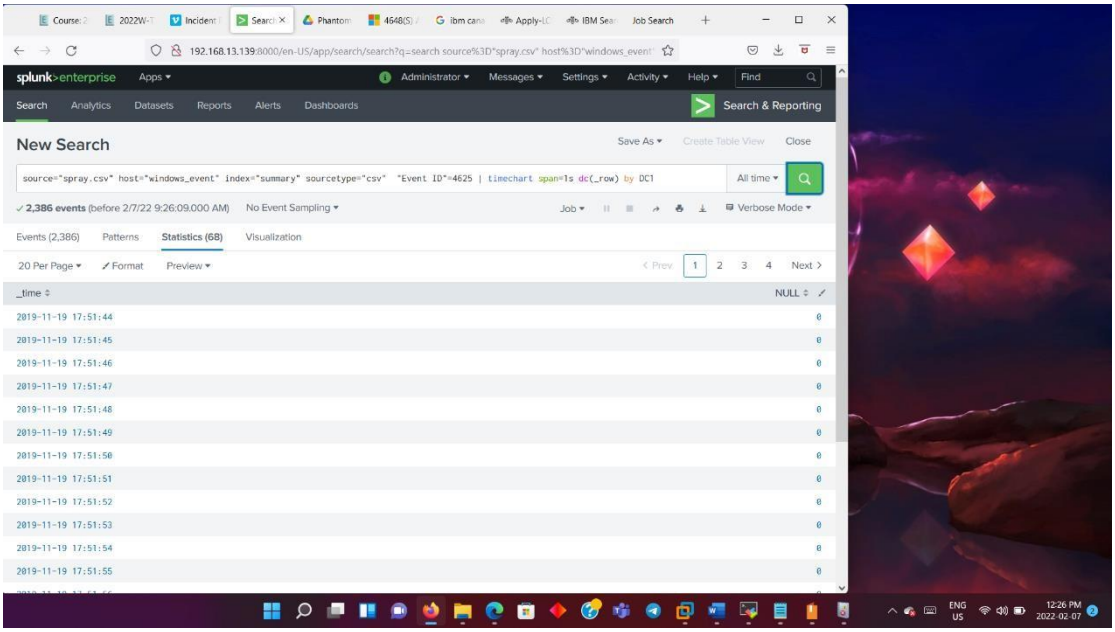
We can also see the source of attack



The source of the attack (device)is given in the box: DC1.
We must also do an RCA root cause analysis to make sure that it does not happen again.

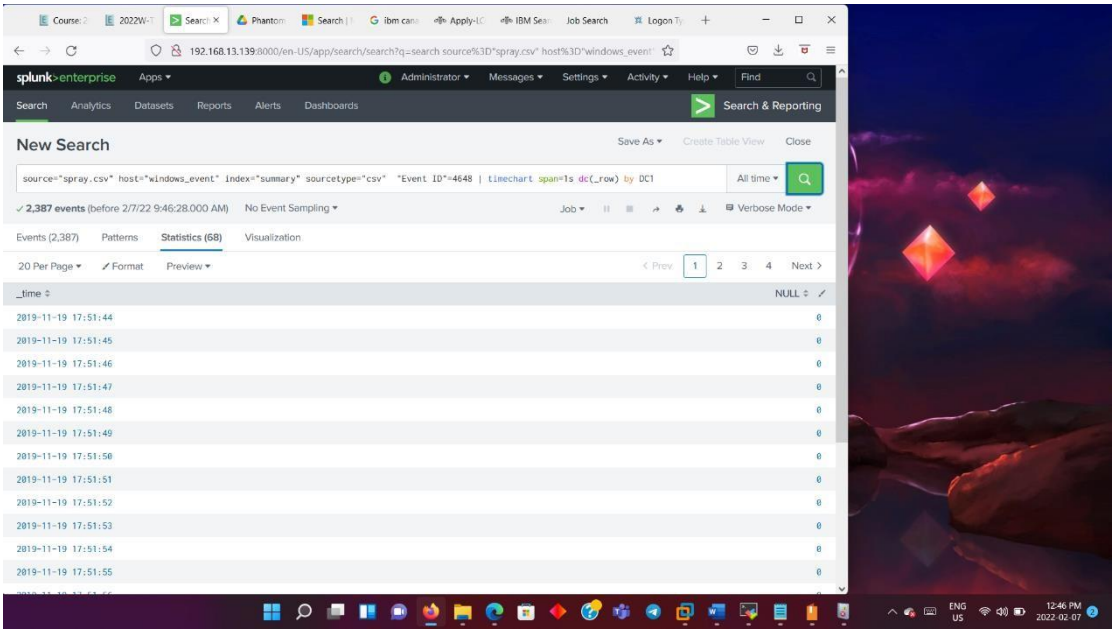


For every second it shows the attack.

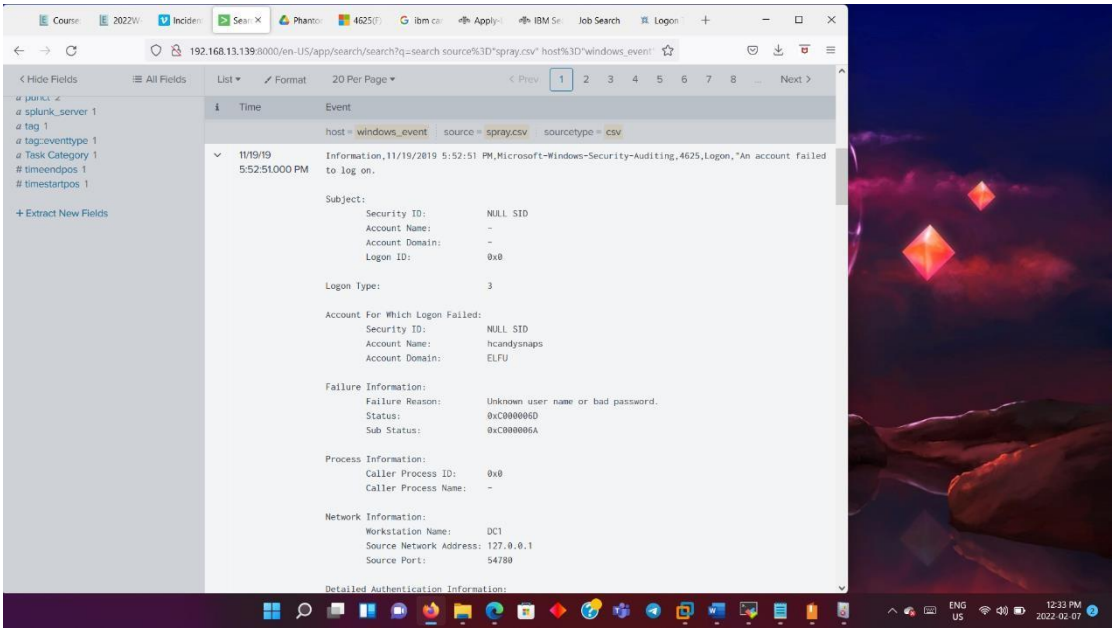


Query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" "Event ID"=4625 | timechart span=1s dc(_row) by DC1

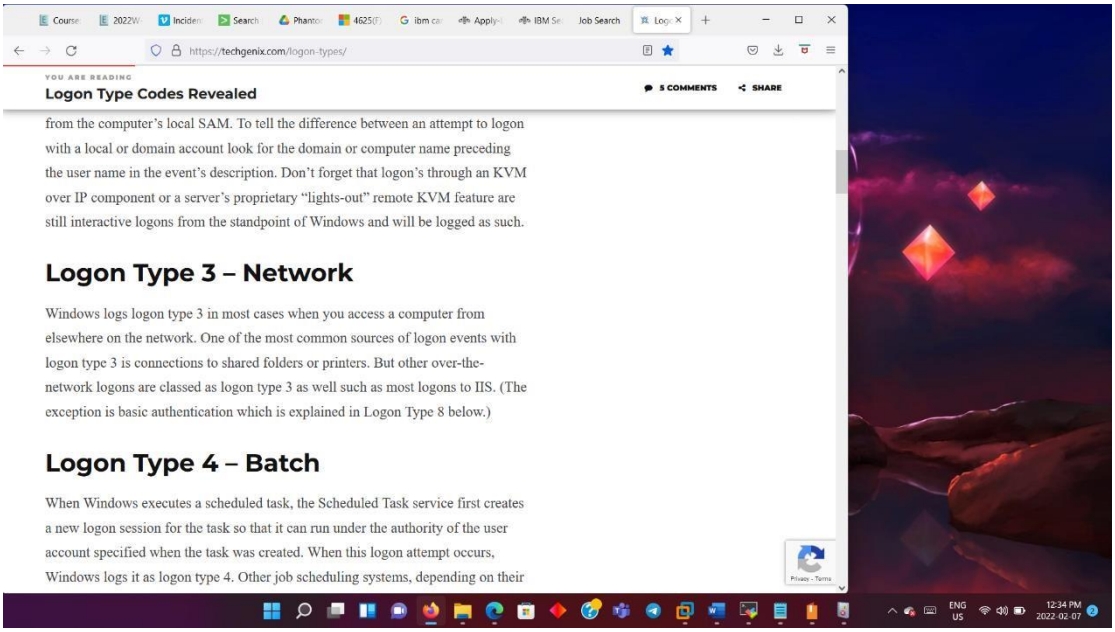
It also gives the details by workstation name: DC1 which is attacked.



Query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" "Event ID"=4648 | timechart span=1s dc(_row) by DC1

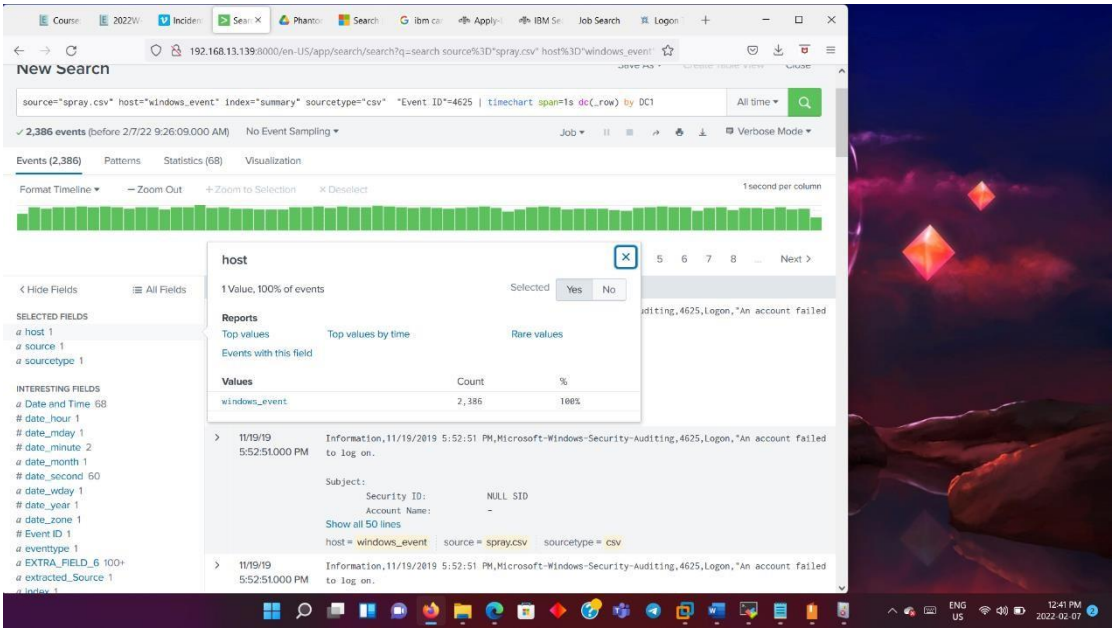


From the above image it shows logon type 3.
Account name: hcandysnaps
Failure Reason: Unknown user name or bad password.
Status: 0xC000006D
Sub Status: 0xC000006A



Logon type 3 is justified above.

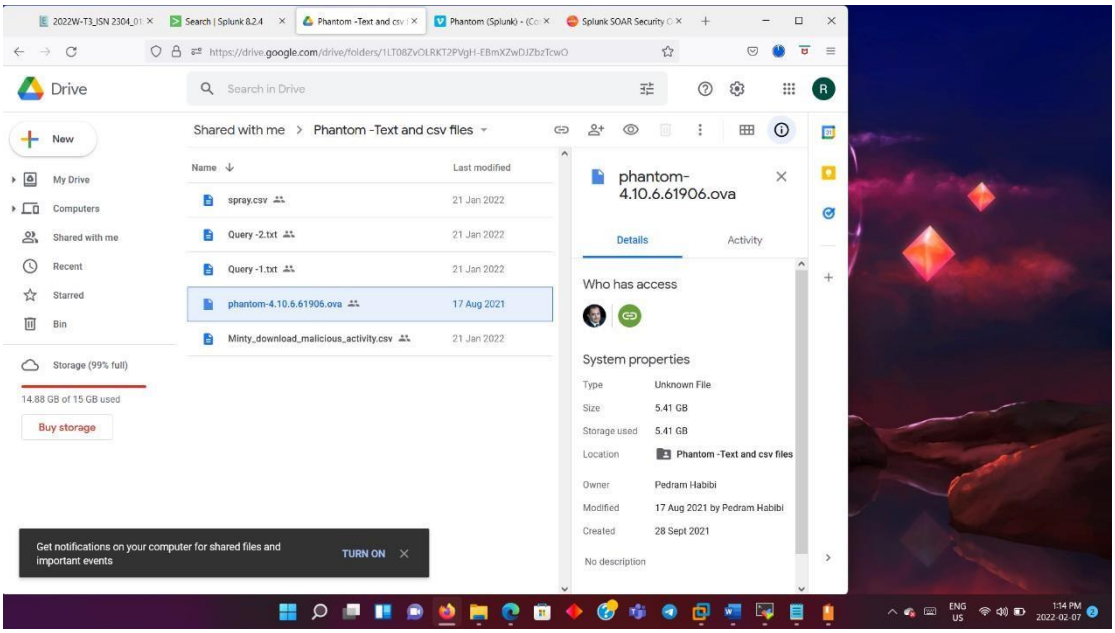
We can also get the details by checking the status in Microsoft docs.



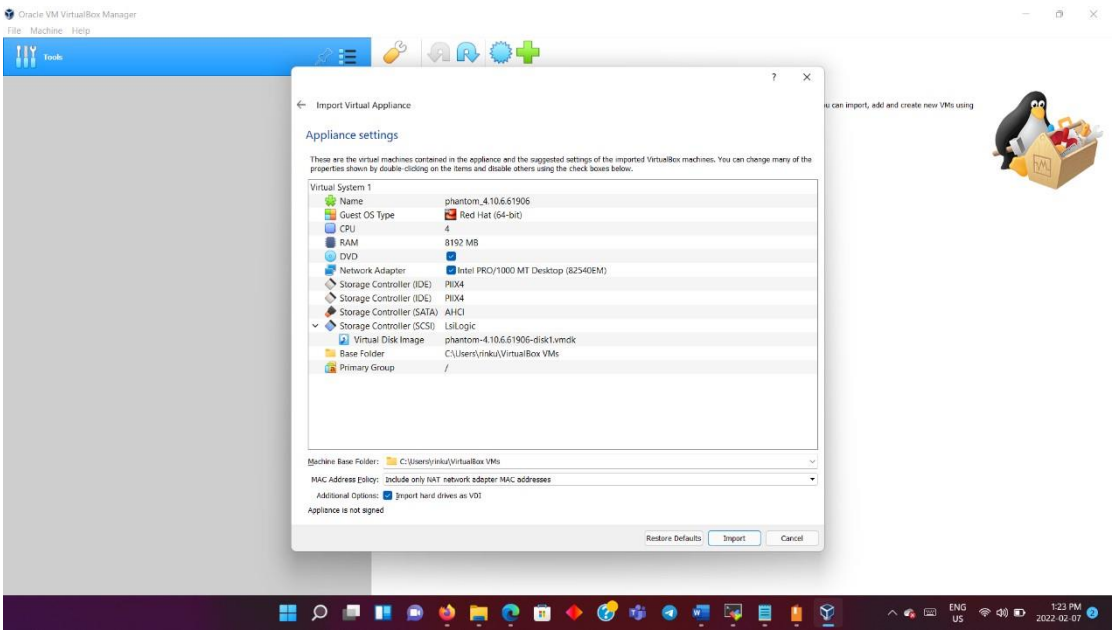
Finally, we can also get many details like ghost name, source, source type etc.

Part 2a

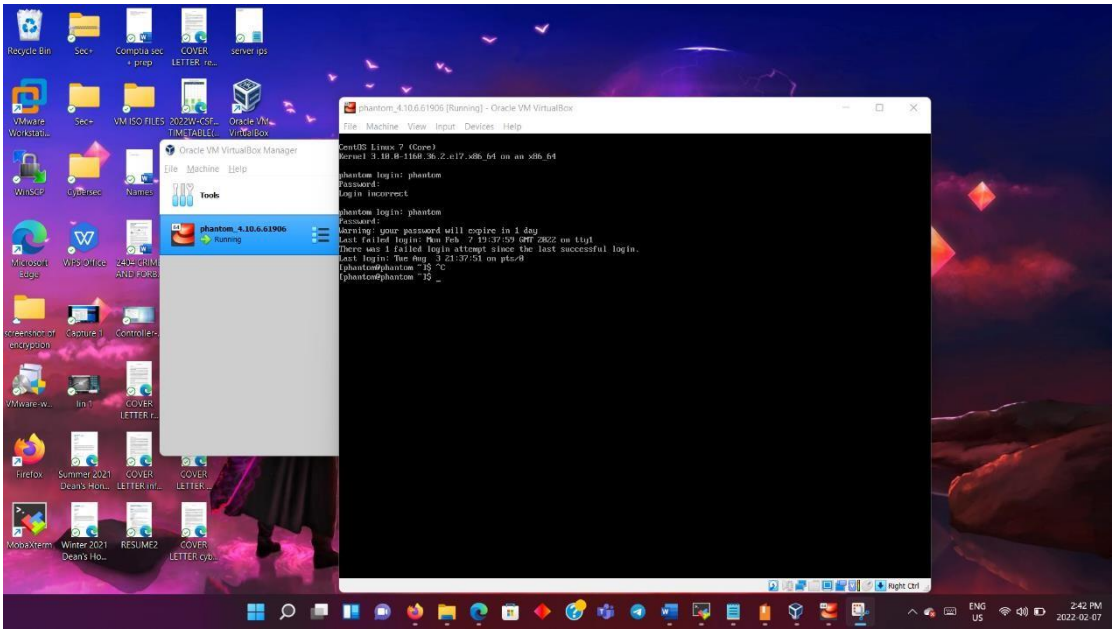
SOAR- Phantom (Splunk)



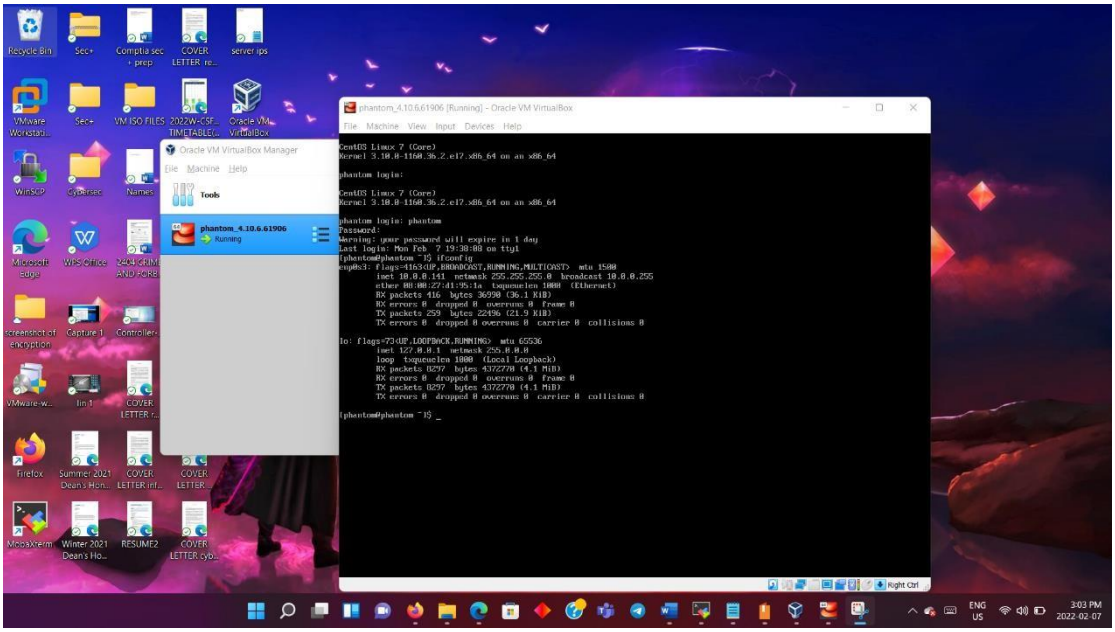
We first download the phantom community edition from google. (Splunk SOAR Security Orchestration & Automation)



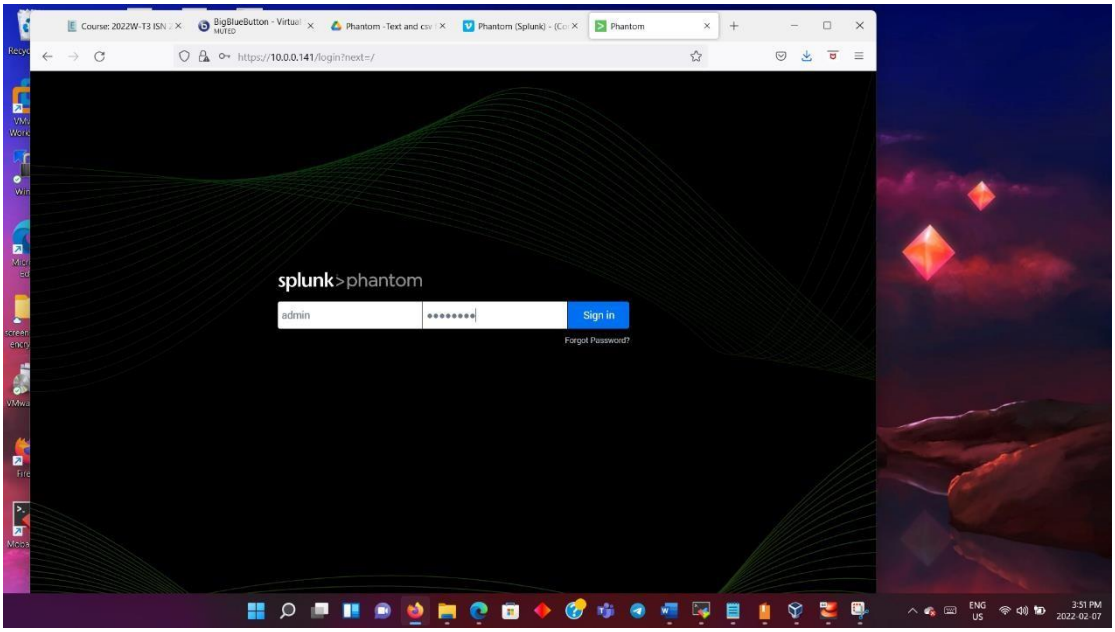
We first import the virtual application.



Logged in using username: phantom
Password: password



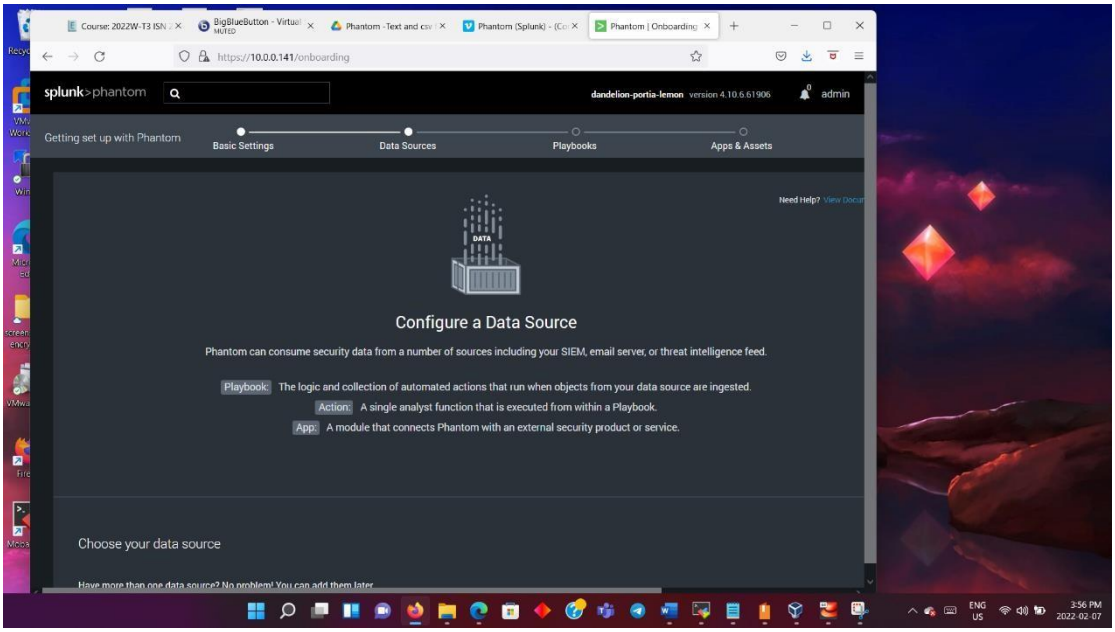
The IP address of phantom is: 10.0.0.141



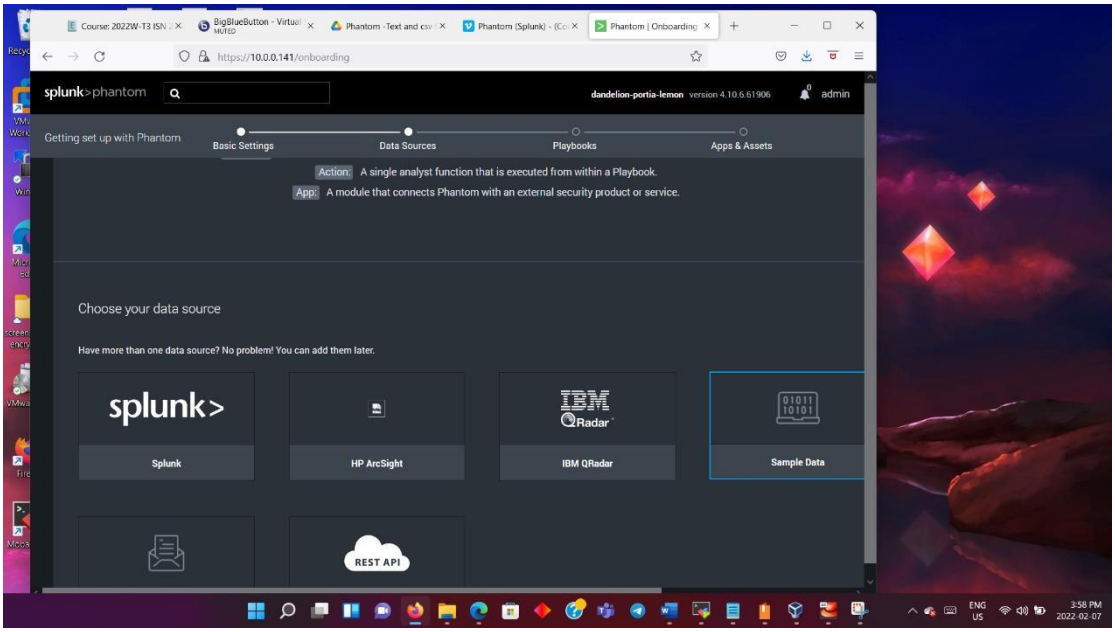
Logged in to the website with the phantom IP address, so Splunk phantom web page got launched.

Logged in using the username: admin
Password: password

Later login and change the credentials.

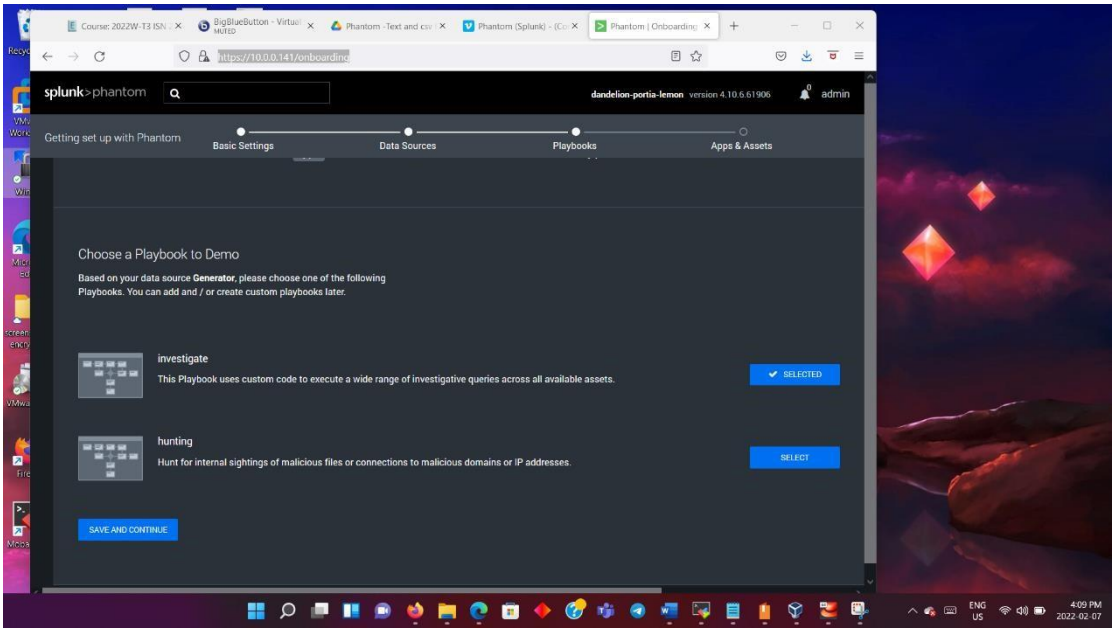


Now set the data source.

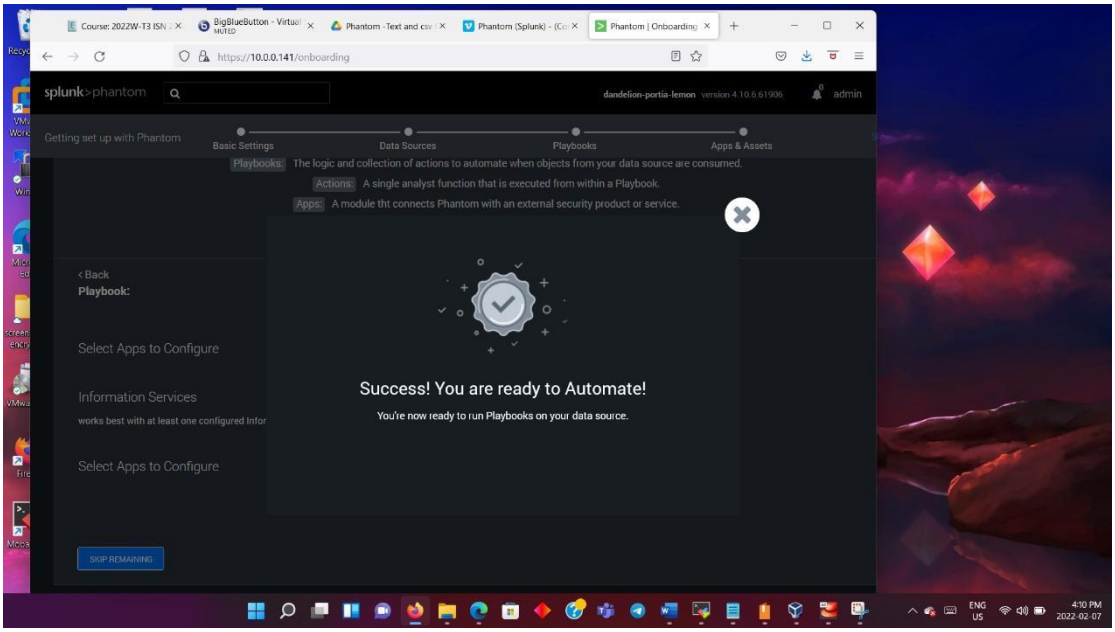


Select sample data

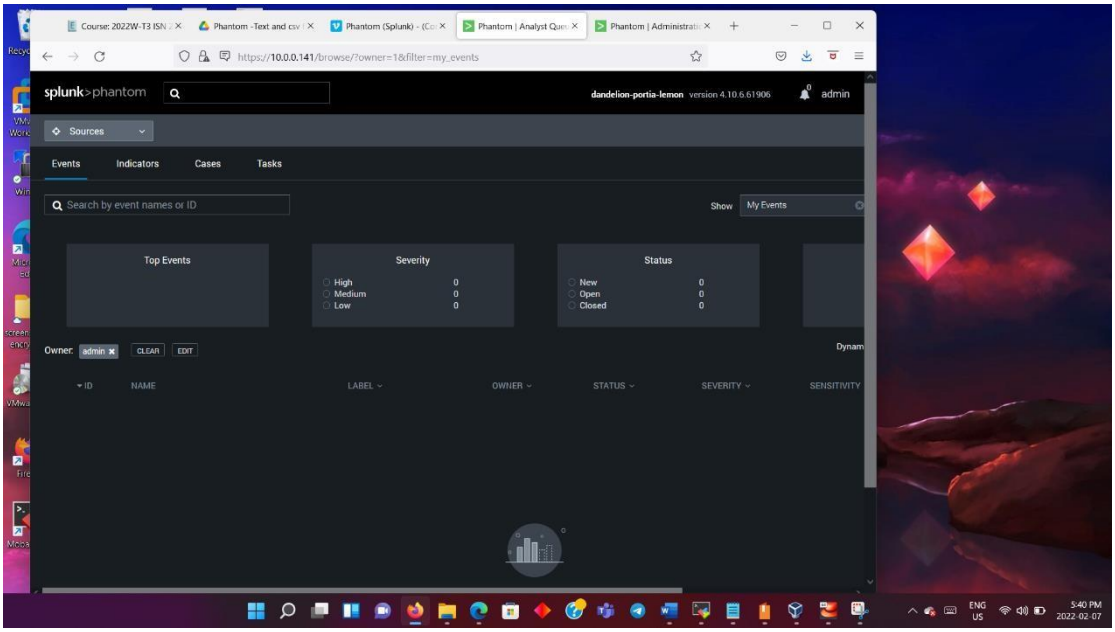
Provide the asset name as sample data
Events to generate :20
Artifacts per events to generate :2
Container label: email
And click on save and continue



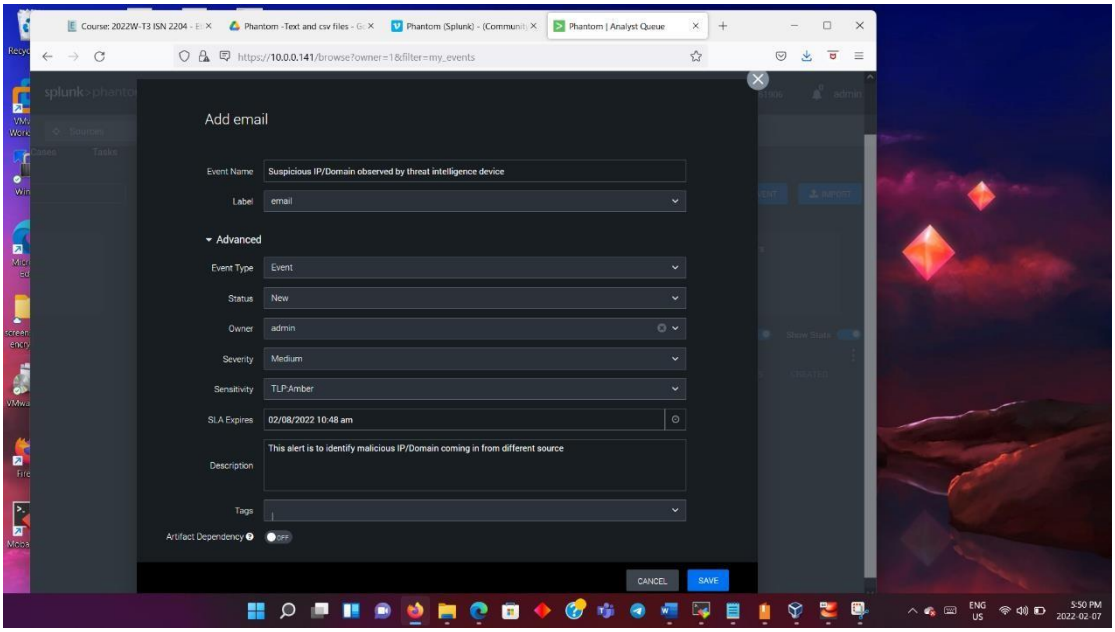
Select investigate and click on save and continue.



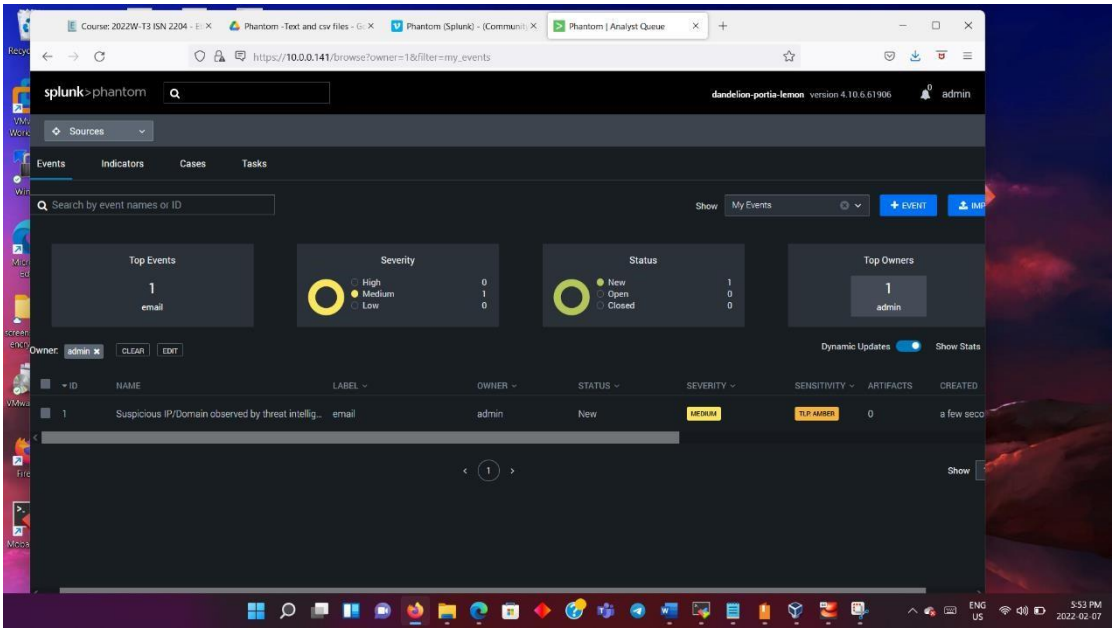
Click on skip remaining.



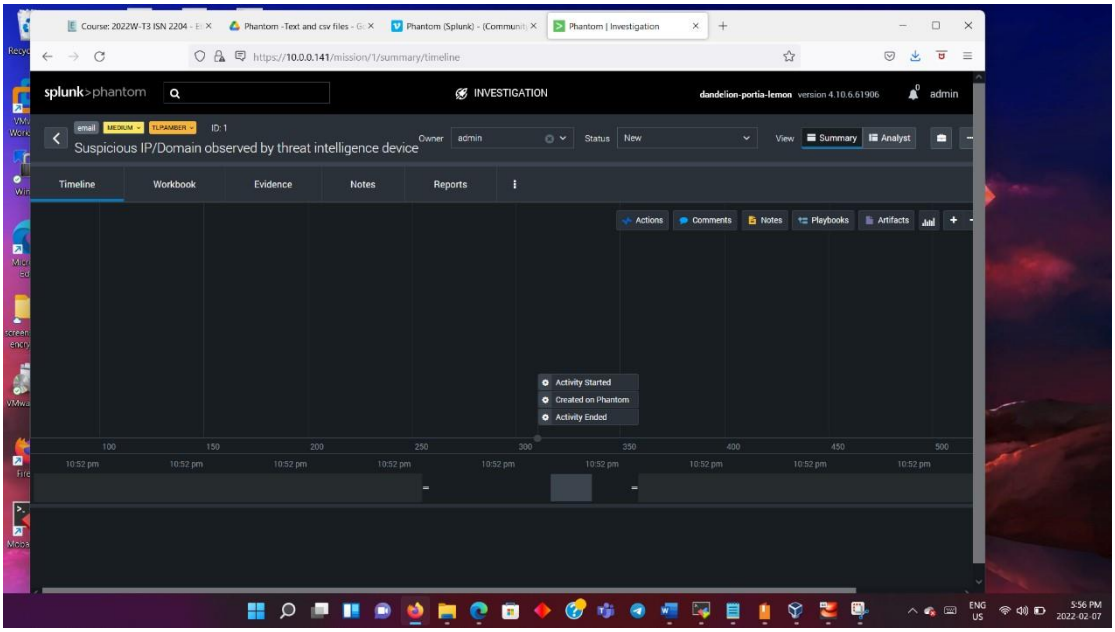
Click on source >my event



Select add event and create new event details as above and also tag suspicious IP and click on save.

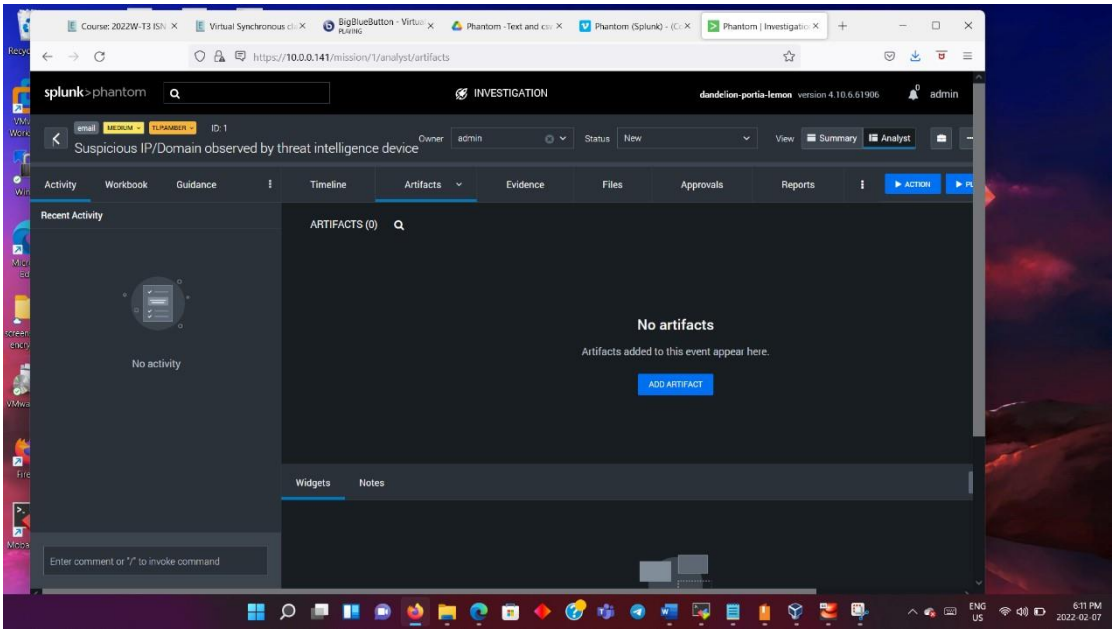


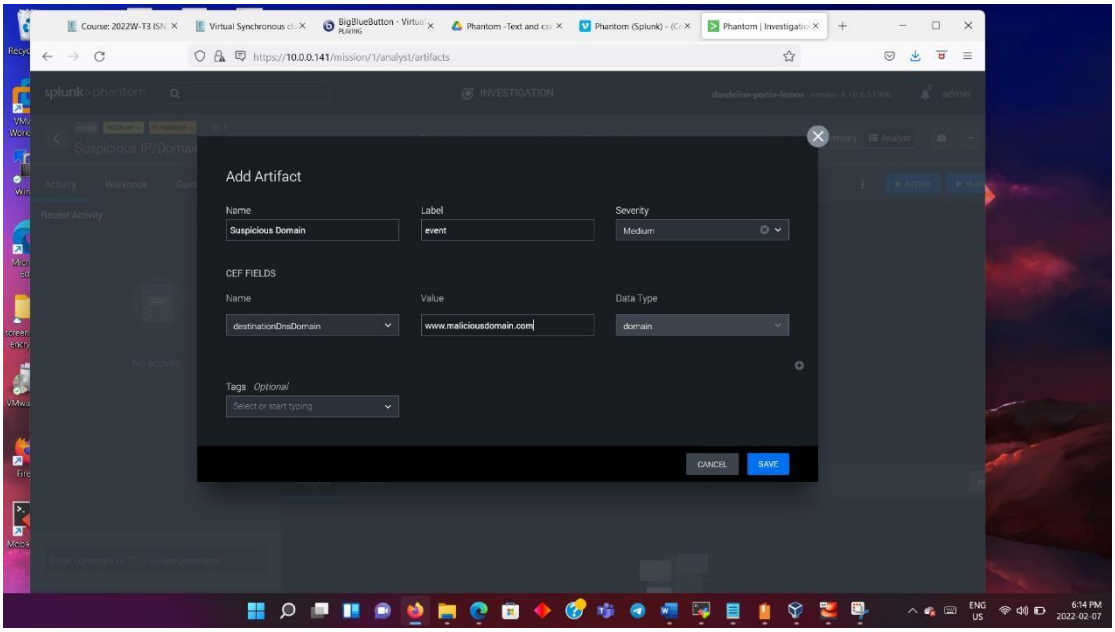
From the above image we are able to see the event created, severity provided, status etc.



Select on the event name created, it will show the above page, which gives the details of when it was created, ended etc.

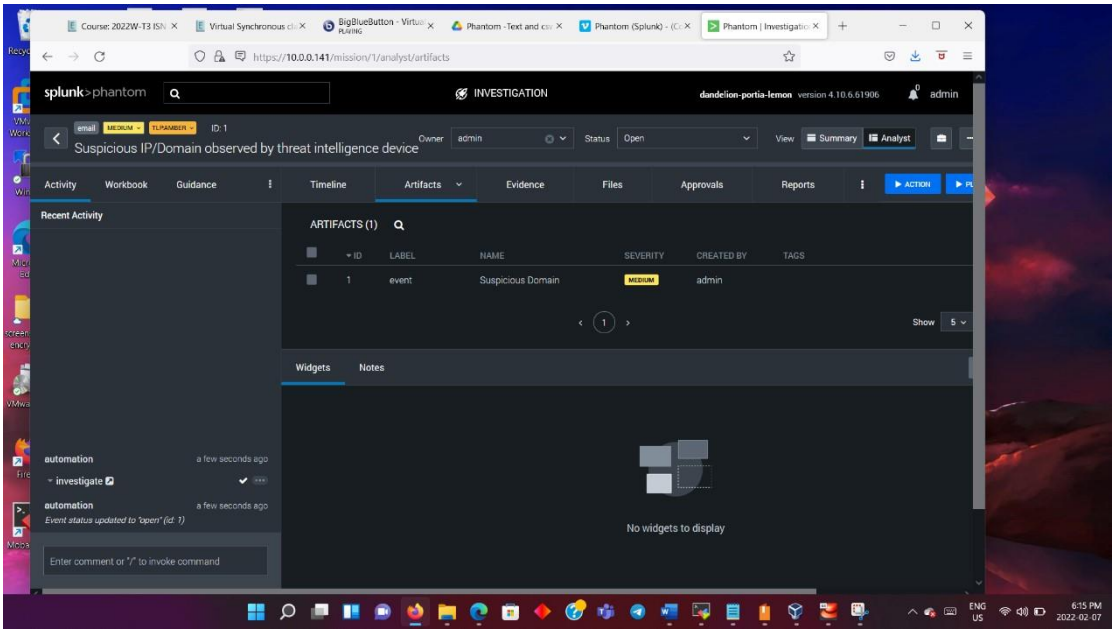
Now select Artifact > change to analyst view > select artifact > select add artifact.



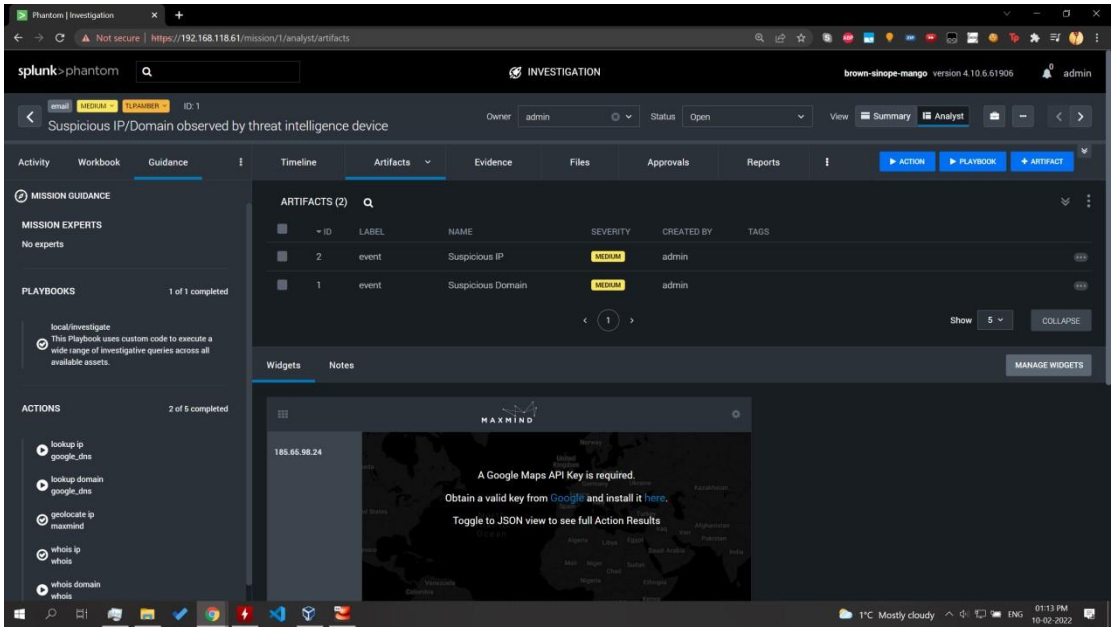


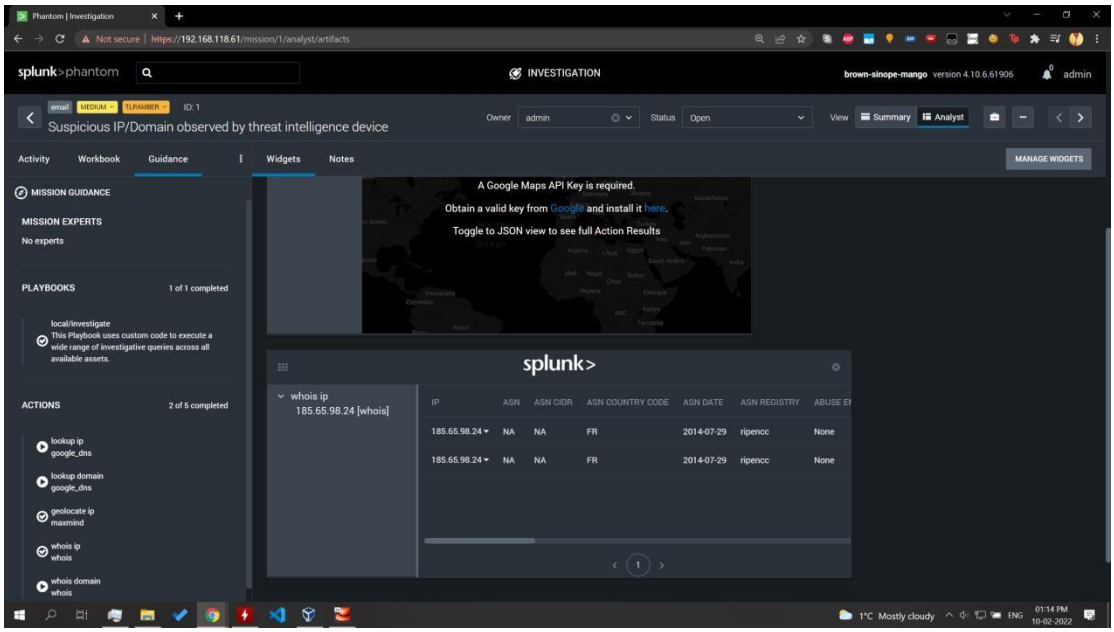
Type in the details as above and click on save.

It creates the first artifact as shown below.

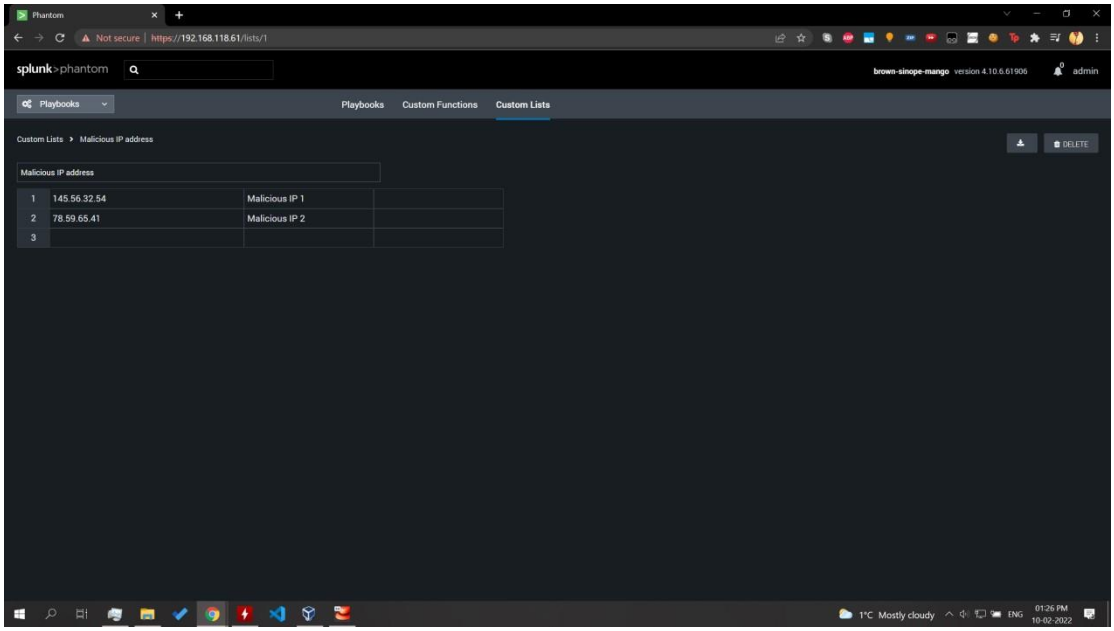


Similarly create the 2nd artifact container should appear like this afterwards

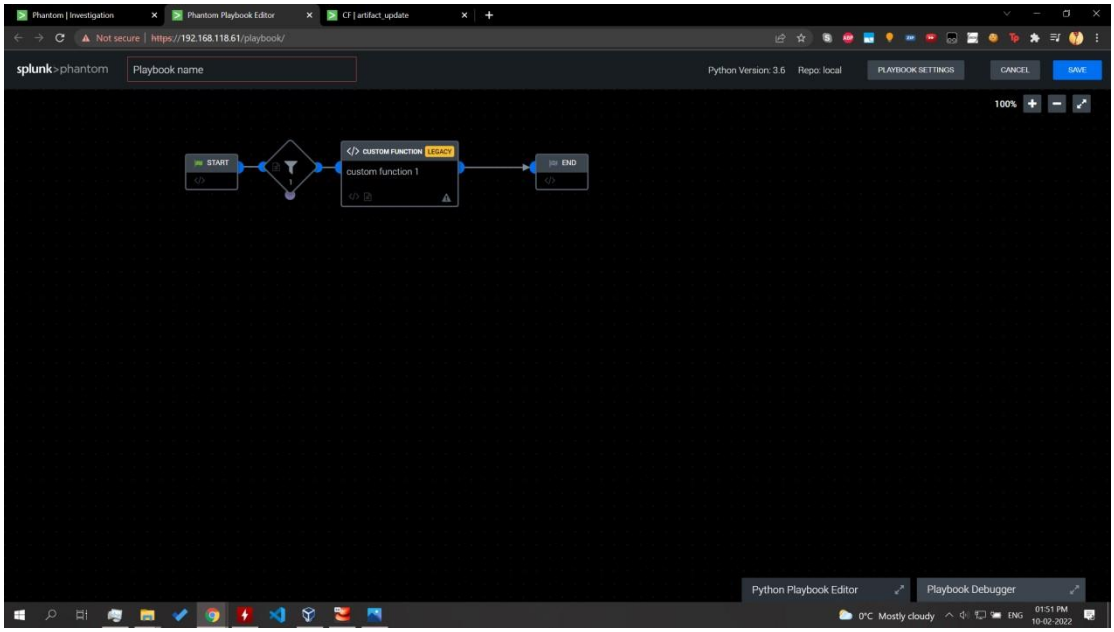




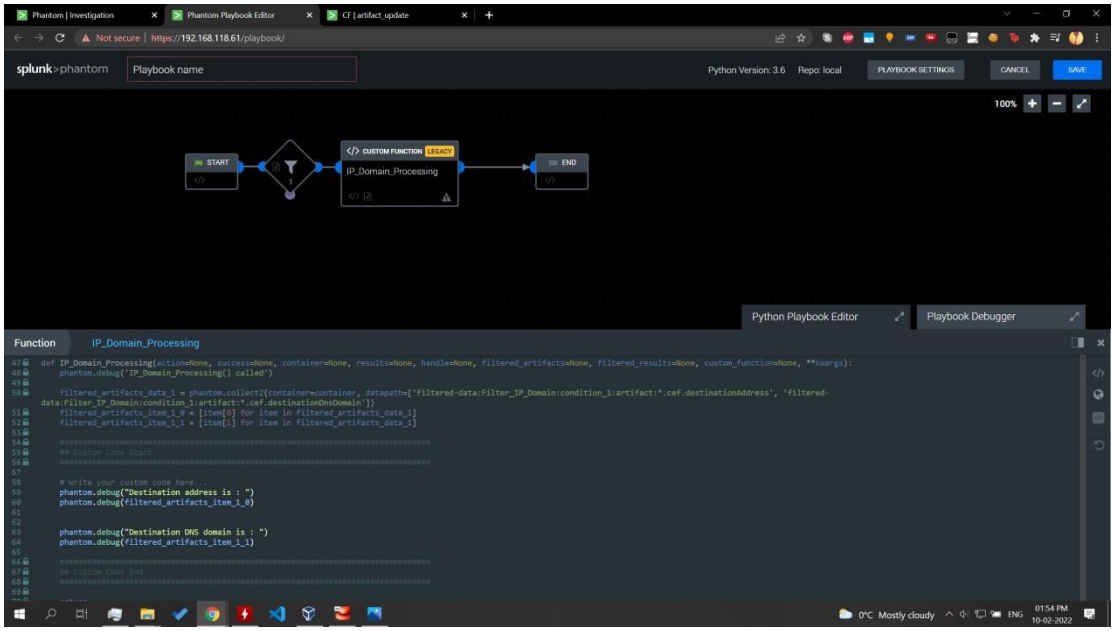
Creating a custom list of malicious IP addresses



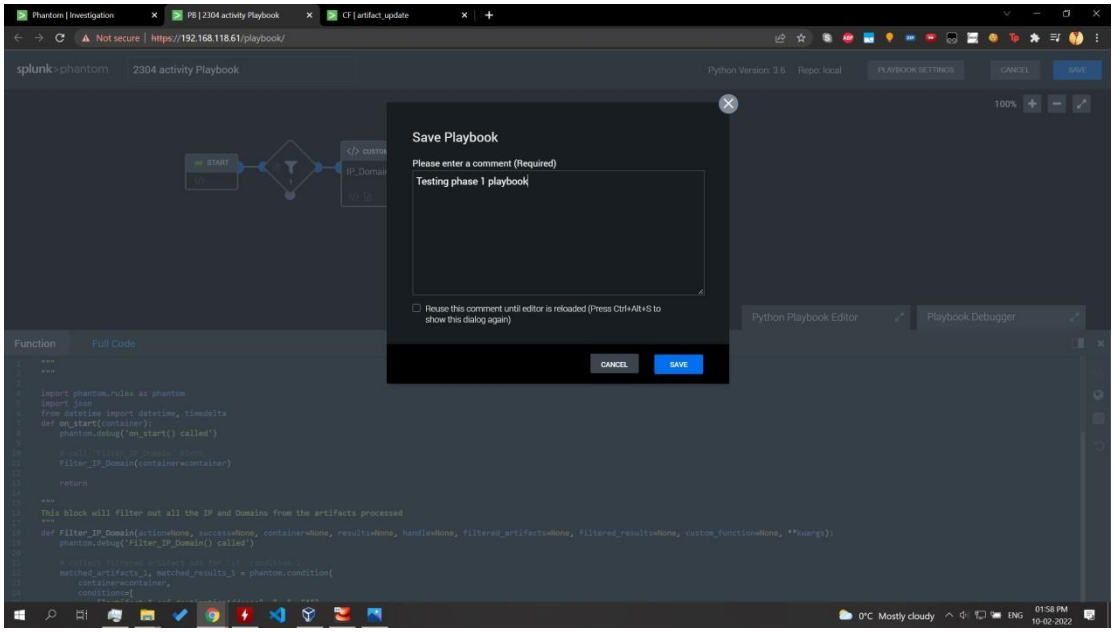
Create a new playbook in Splunk with a custom function



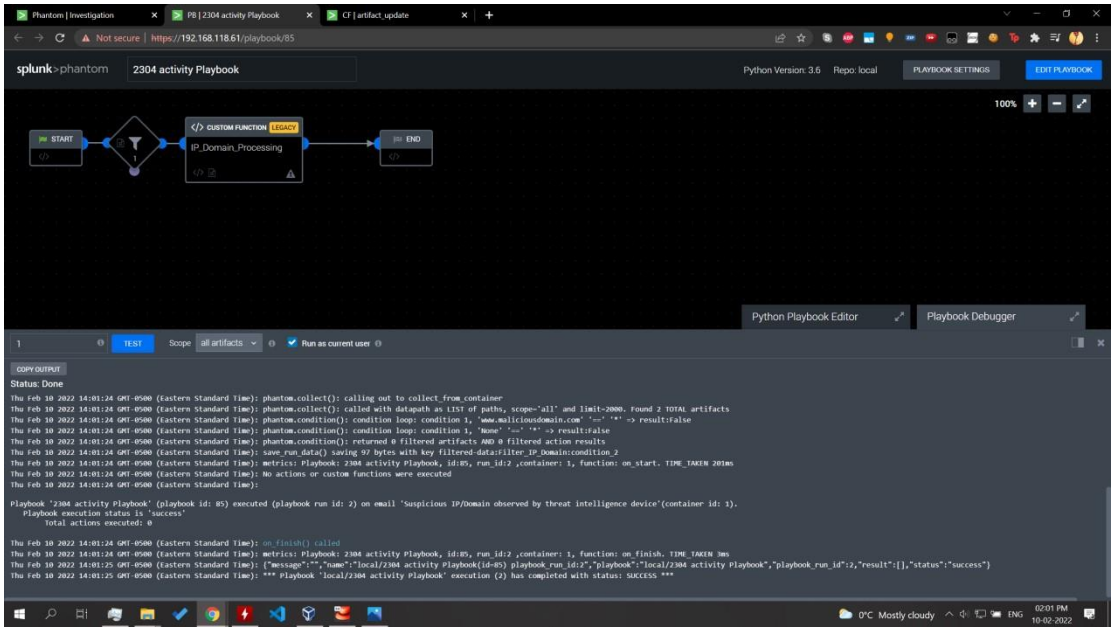
Modify the custom function to detect malicious IPs (as defined previously in artifacts)



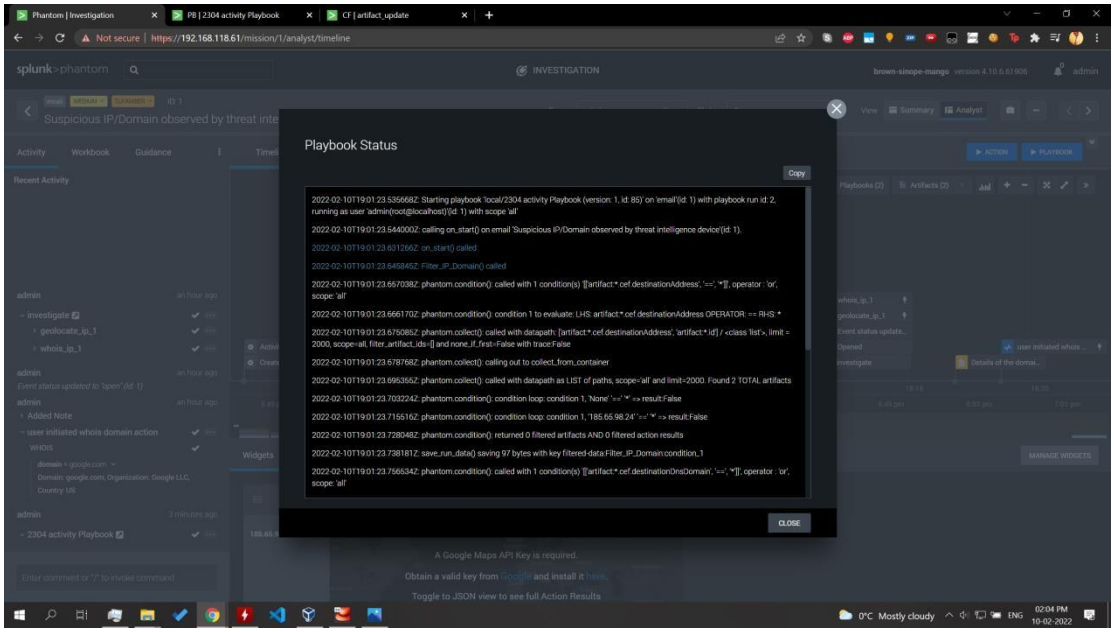
Save the playbook for testing later

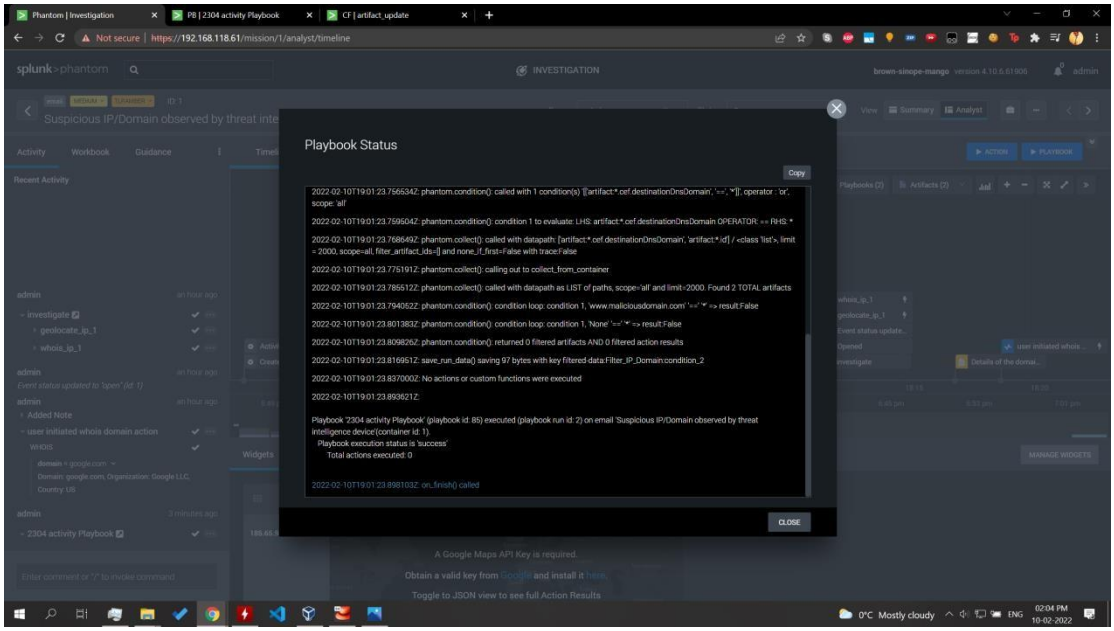


Test the playbook with event ID as 1



Open Debug log from container window to verify that conditions specified in custom function were fulfilled. Output looks as follows





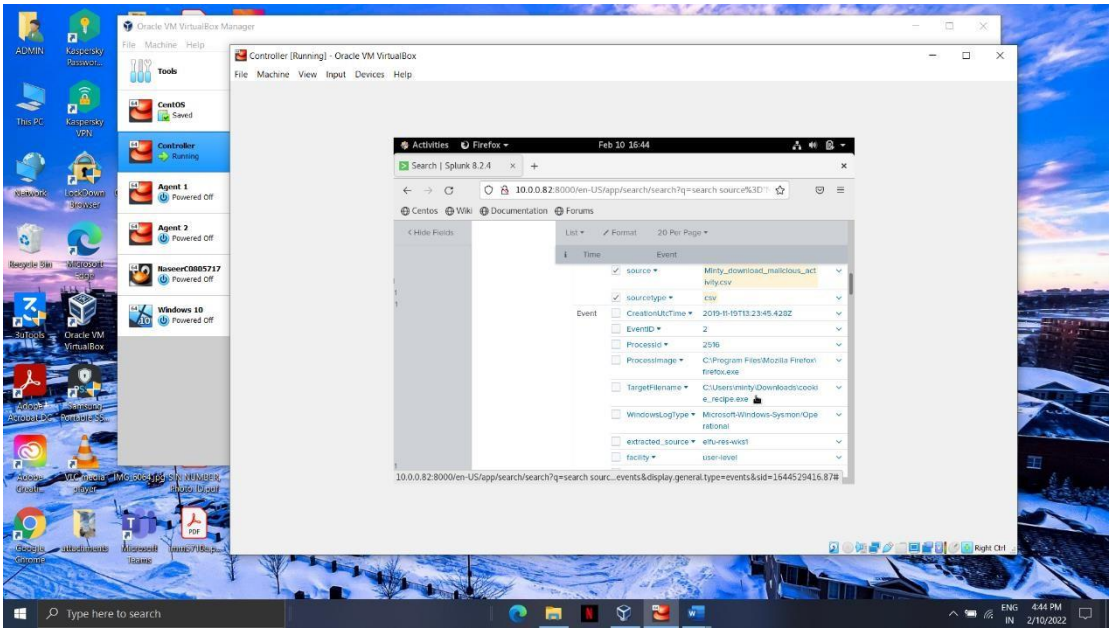
As it can be seen the 2 artifact conditions are listed in the debug log

Part 2b

Incident response

Question 1: Minty CandyCane reported some weird activity on his computer after he clicked on a link in Firefox for a cookie recipe and downloaded a file.

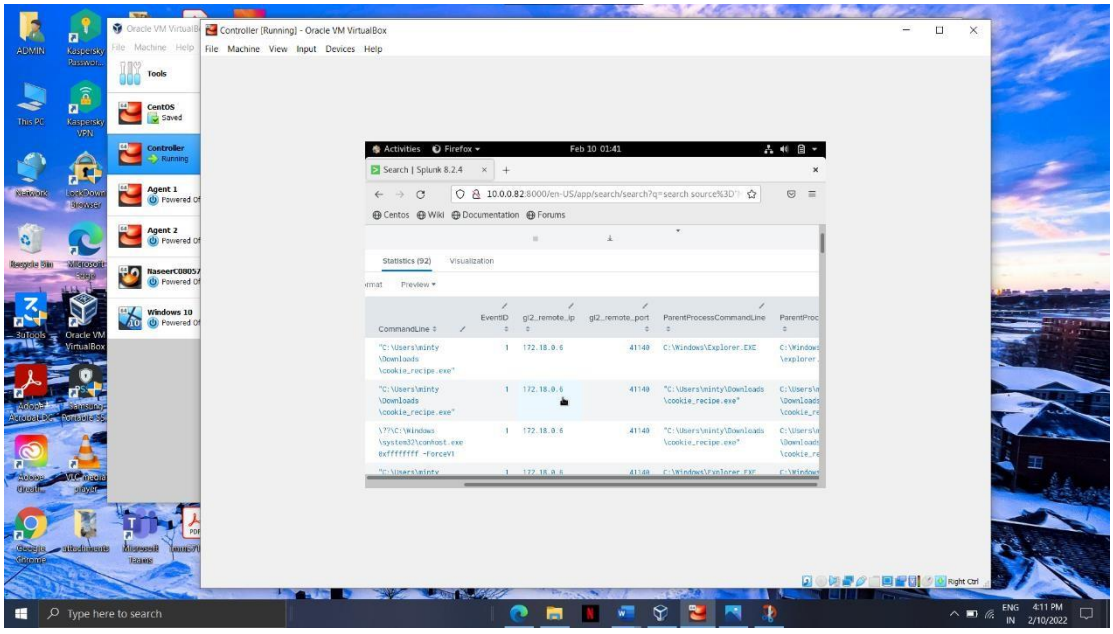
source="Minty_download_malicious_activity.csv" host="localhost.localdomain" index="main" sourcetype="csv" firefox cookie



Answer: C:\Users\minty\Downloads\cookie_recipe.exe

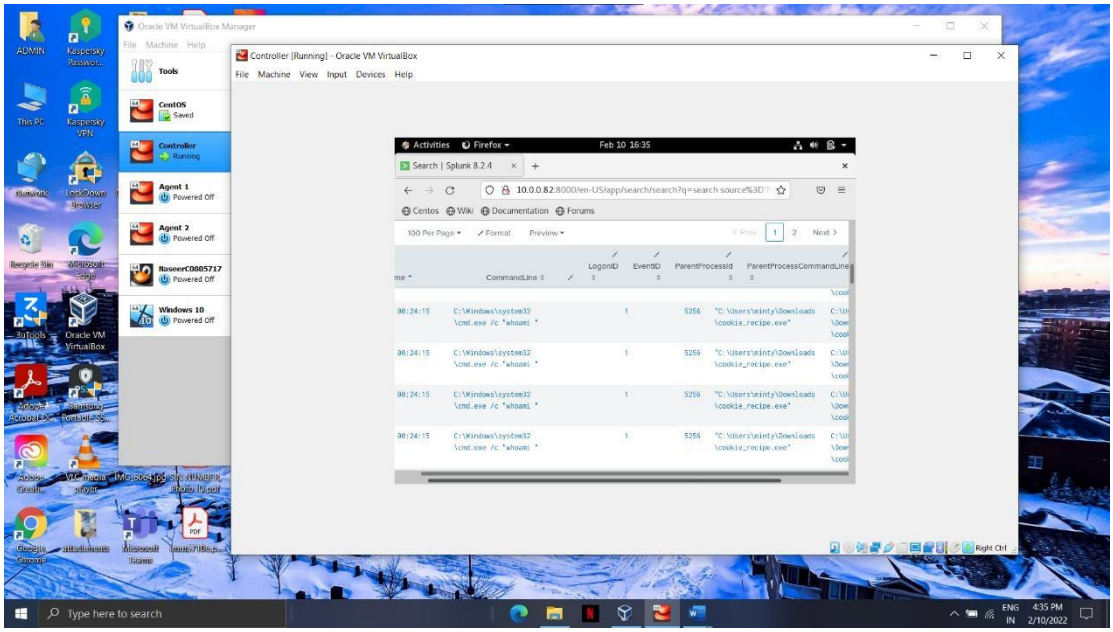
Question 2: The malicious file downloaded and executed by Minty gave the attacker remote access to his machine. What was the ip:port the malicious file connected to first?

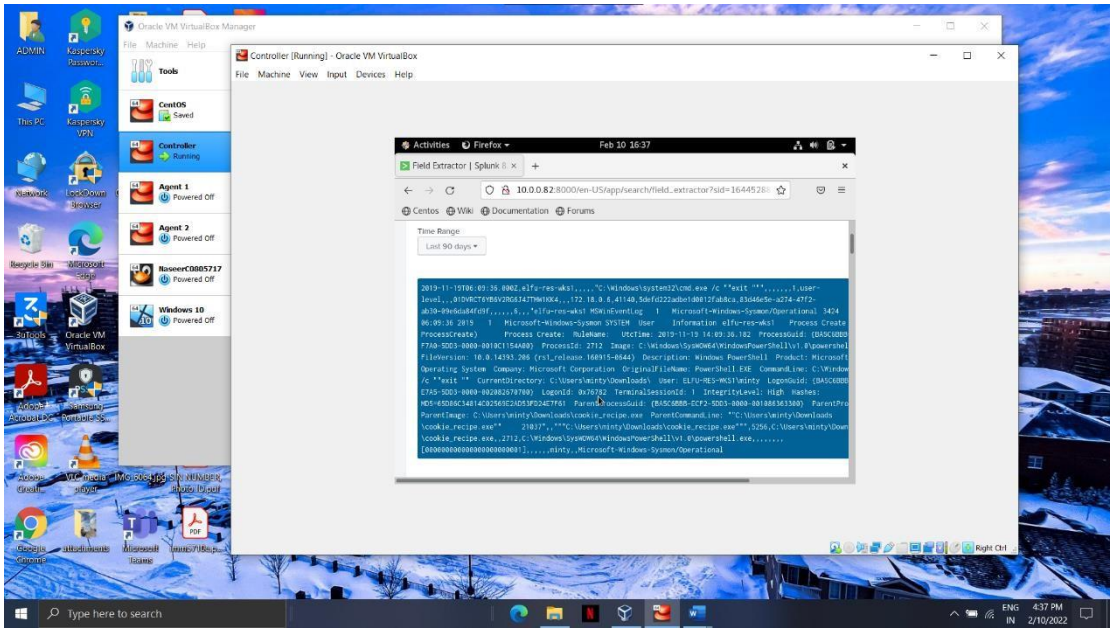
source="Minty_download_malicious_activity.csv" host="Minty" index="main" sourcetype="csv" cookie_recipe.exe | table _time CommandLine EventID gl2_remote_ip gl2_remote_port ParentProcessCommandLine ParentProcessImage



Question 3: What was the first command executed by the attacker?

source="Minty_download_malicious_activity.csv" host="Minty" index="main" sourcetype="csv" cookie_recipe.exe | table _time CommandLine EventID gl2_remote_ip gl2_remote_port ParentProcessCommandLine ParentProcessImage

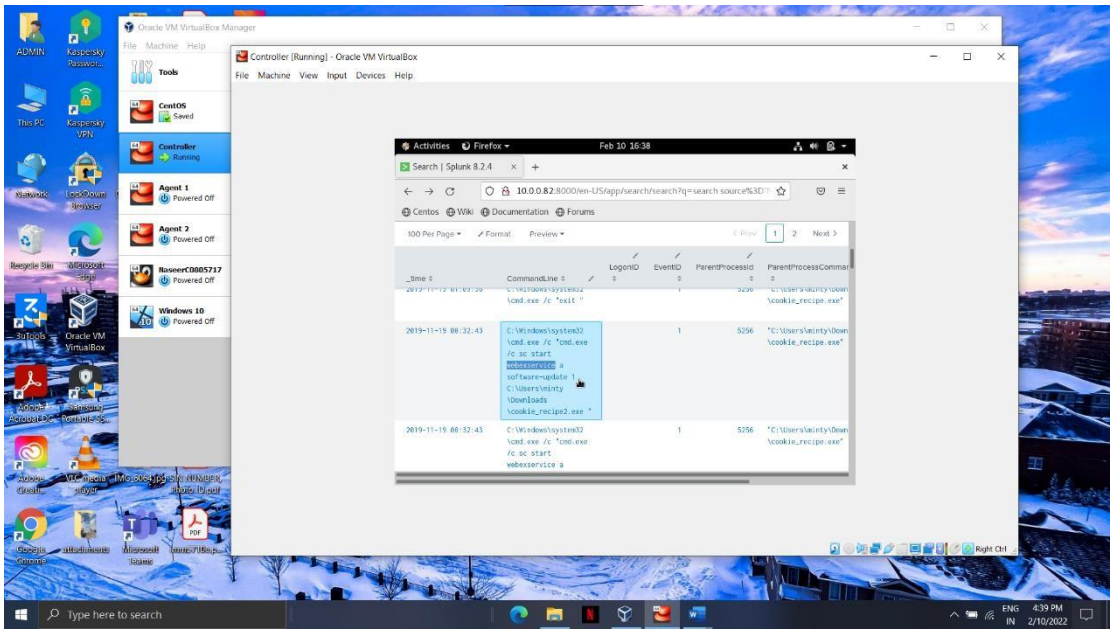




Answer: whoami
LogonId: 0x76782
ParentProcessId: 5256

Question 4: What is the one-word service name the attacker used to escalate privileges?

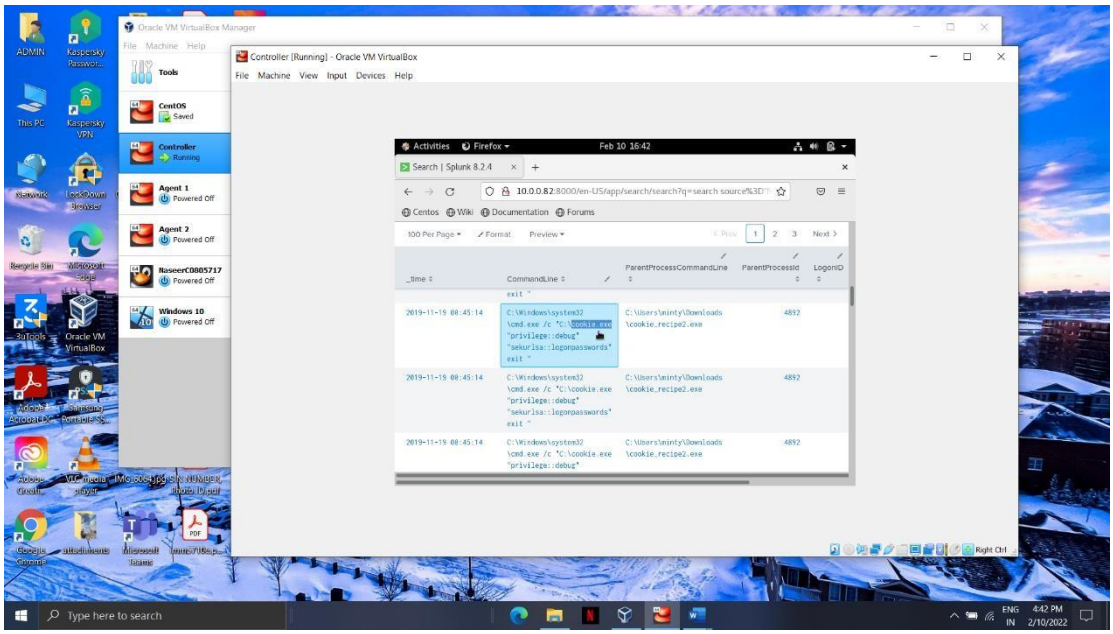
source="Minty_download_malicious_activity.csv" host="Minty" index="main" sourcetype="csv" cookie_recipe.exe | table _time CommandLine EventID ParentProcessCommandLine ParentProcessImage



Answer: webexservice

Question 5: What is the file-path + filename of the binary ran by the attacker to dump credentials?

source="Minty_download_malicious_activity.csv" host="Minty" index="main" sourcetype="csv" cookie_recipe2.exe | table _time CommandLine EventID ParentProcessCommandLine ParentProcessImage



Answers: C:\cookie.exe

Conclusion

To conclude, we can say that Splunk can be beneficial in a real-world use in various scenarios such as to analyze a security incident from its root. To automate tasks and generate an easy-to-understand statistical report for events/logs. By using Splunk Phantom, we can drastically reduce our incident response time due to its ability to investigate and respond to threats faster. This can reduce malware footprint in the system.

Achievement

By completing this exercise, we have learned how to query in Splunk to get wellversed with its behavior. Investigated a brute force attack and analyzed various EventIDs. Also, by using Splunk Phantom we learned to automate incident responses. At last, we learned how to investigate a security incident using Splunk by uploading a .csv file. All these exercises made us industry-ready to operate and execute tasks using Splunk in real-world environments.

References

1. *Phantom -Text and csv files - Google Drive*. (n.d.). Google Drive. Retrieved February 11, 2022, from https://drive.google.com/drive/folders/1LT08ZvOLRKT2PVgH-EBmXZwDJZbzTcwO?usp=sharing
2. *Private video on Vimeo*. (n.d.-a). Vimeo. Retrieved February 11, 2022, from https://vimeo.com/602202629
3. *Private video on Vimeo*. (n.d.-b). Vimeo. Retrieved February 11, 2022, from https://vimeo.com/602943288
4. *Private video on Vimeo*. (n.d.-c). Vimeo. Retrieved February 11, 2022, from https://vimeo.com/605272297
5. V. (n.d.). *Private video on Vimeo*. Vimeo. Retrieved February 11, 2022, from https://vimeo.com/609171980

Name of students who has not participated in the assignment.
Student name:
Student name:
Student name:
Student name:
Student name:

All members of the group have participated in this activity.