

Contents

Abstract	2
Splunk Search Query / Brute-Force attack	3
Incident response Brute-Force attack.....	18
SOAR- Phantom (Splunk)	32
Incident response	45
Conclusion	49
Achievement	50
References.....	51

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

The screenshot shows the Splunk web interface with a search results page. The search bar at the top contains the query "index=*". The results table shows 30,354 events from 2/4/22 9:27:23.000 PM to 2/4/22 9:42:23.000 PM. The table has columns for Time, Event, and Source. The first few rows of data are:

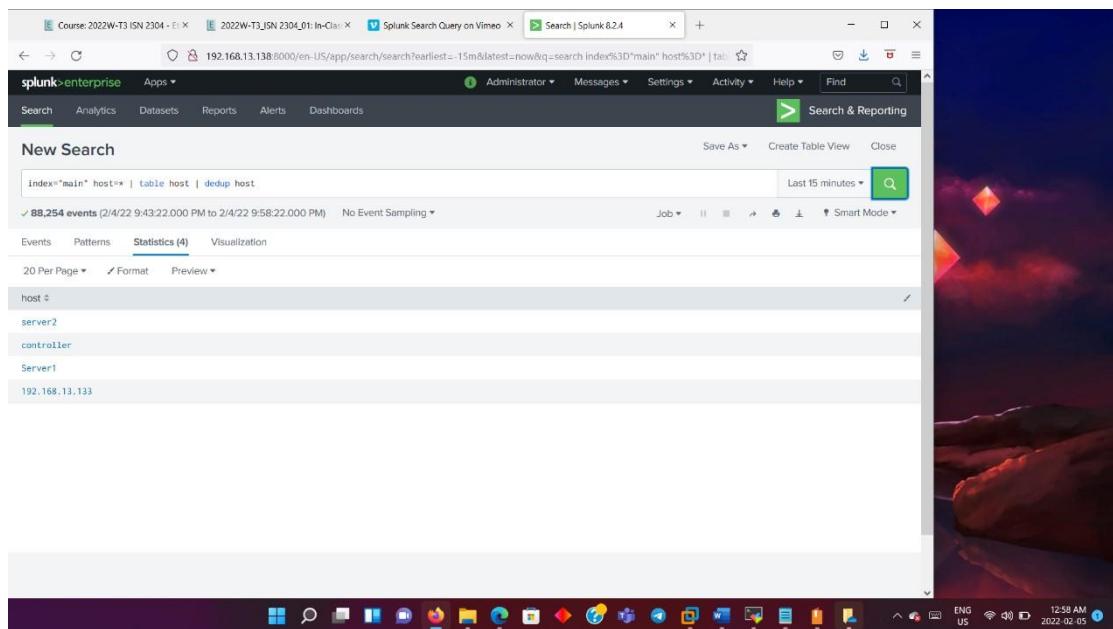
Time	Event	Source
2/4/22 9:42:09.000 PM	root :0 :0	host = controller source = who sourcetype = who
2/4/22 9:42:09.000 PM	root pts/0 :0	host = controller source = who sourcetype = who
2/4/22 9:42:09.000 PM	root pts/1 192.168.13.1	host = controller source = who sourcetype = who
2/4/22 9:42:09.000 PM	root pts/2 192.168.13.1	host = controller source = who sourcetype = who
2/4/22 9:42:06.000 PM	all 0.51 0.00 3.55 0.00 95.94	host = server2 source = cpu sourcetype = cpu

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.

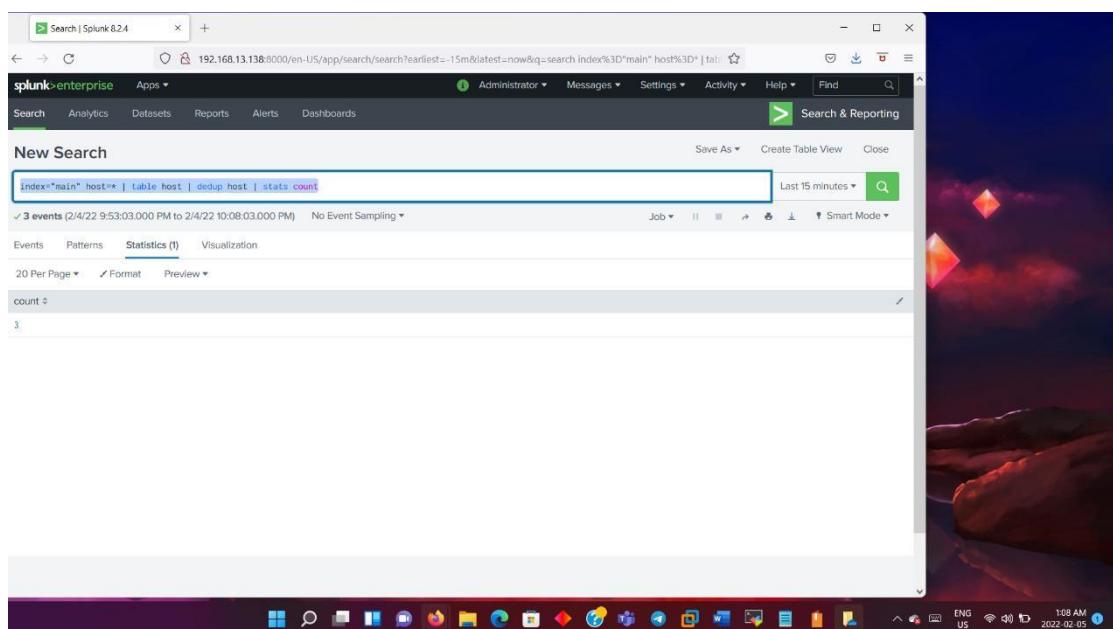
We then expand and check the index as main.

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.

A screenshot of the Splunk 8.2.4 search interface. The search bar contains the query: `index="main" host=* | table host | dedup host | stats count | rename count as "Count of Servers"`. The results section shows a single row with the value '3'. The interface includes a navigation bar with 'splunk>enterprise' and various tabs like 'Search', 'Analytics', 'Dashboards', and 'Statistics'. The status bar at the bottom right shows the date and time as 2022-02-05 11:22 AM.

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.

A screenshot of the Splunk 8.2.4 search interface. The search bar contains the query: `|metadata type=sourcetypes index=main`. The results are displayed in a table with columns: firstTime, lastTime, recentTime, sourcetype, totalCount, and type. The table lists various source types with their counts, such as 'greeter.log-too_small' (3), '1.log' (279), 'Linux:SELinuxConfig' (11), 'Unix:ListeningPorts' (284), 'Unix:SSHConfig' (11), 'Unix:Service' (1614), 'Unix:Update' (5778), 'Unix:Uptime' (6), 'Unix:UserAccounts' (484), 'Unix:Version' (6), and 'Xorg' (919). The interface includes a navigation bar with 'splunk>enterprise' and various tabs like 'Search', 'Analytics', 'Dashboards', and 'Statistics'. The status bar at the bottom right shows the date and time as 2022-02-05 11:16 AM.

firstTime	lastTime	recentTime	sourcetype	totalCount	type
1642650609	1642650609	1644035786	:0-greeter.log-too_small	3	sourcetypes
1478418219	1644036564	1644036564	:1-log	279	sourcetypes
1644034427	1644041619	1644041623	Linux:SELinuxConfig	11	sourcetypes
1644034427	1644041617	1644041622	Unix:ListeningPorts	284	sourcetypes
1644034427	1644041620	1644041624	Unix:SSHConfig	11	sourcetypes
1644034427	1644041620	1644041624	Unix:Service	1614	sourcetypes
1644034427	1644036698	1644036705	Unix:Update	5778	sourcetypes
1644034428	1644036699	1644036698	Unix:Uptime	6	sourcetypes
1644034427	1644041618	1644041623	Unix:UserAccounts	484	sourcetypes
1644034428	1644036699	1644036701	Unix:Version	6	sourcetypes
1478418219	1644036726	1644036727	Xorg	919	sourcetypes
1478418219	1644036638	1644036638	Xorg-too_small	4	sourcetypes

Query 6: `|metadata type=sourcetypes index=main|`

We get the details of source types for index pointing to main along with the count.

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | table host,user,user_id,user_group,home* | dedup user
- Results:** 88 events (2/4/22 10:06:44.000 PM to 2/4/22 10:21:44.000 PM) No Event Sampling
- Table View:** A table showing user account details for host Server1. The columns are host, user, user_id, user_group, and home. The data includes entries like rinku (user_id 1000, home /home/rinku), root (user_id 0, home /), and various system services (sshd, cron, ntp, etc.).
- Toolbar:** Includes Save As, Create Table View, Close, Job, Find, and Smart Mode buttons.
- Bottom:** Shows the Windows taskbar with various icons and system status (ENG US, 1:22 AM, 2022-02-05).

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

Gives the above details

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | table host,user,user_id,user_group,home* | dedup user
- Results:** 88 events (2/4/22 10:09:00.000 PM to 2/4/22 10:24:00.000 PM) No Event Sampling
- Table View:** A table showing user account details for host Server1. The columns are host, user, user_id, user_group, and home. The data includes entries like rinku (user_id 1000, home /home/rinku), root (user_id 0, home /), and various system services (sshd, cron, ntp, etc.).
- Toolbar:** Includes Save As, Create Table View, Close, Job, Find, and Verbose Mode buttons.
- Bottom:** Shows the Windows taskbar with various icons and system status (ENG US, 1:24 AM, 2022-02-05).

Choose verbose mode and check source type.

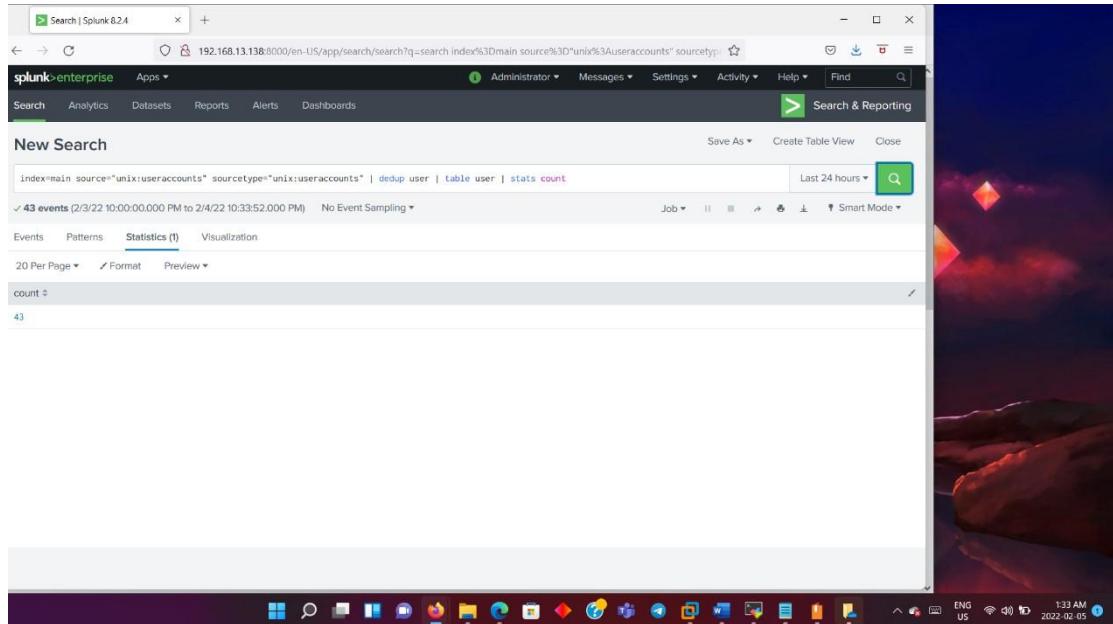
The screenshot shows a Splunk search interface titled "user list". The search bar contains the query "index=main source=\"unix:useraccounts\" sourcetype=\"unix:useraccounts\" | dedup user | table user". The results table displays 43 events from the past 15 minutes, listing users such as rinku, tcpdump, postfix, avahi, sshd, gnome-initial-setup, gdm, pulse, sssd, troubleshoot, chrony, ntp, qemu, and radvd, along with their user IDs and home directories. The interface includes a toolbar at the top with "Search & Reporting" and various navigation buttons.

We then save the report and view.

The screenshot shows a Splunk search interface titled "New Search". The search bar contains the same query as the previous screenshot. The results table displays 43 events from the past 24 hours, showing the same list of users. The interface includes a toolbar at the top with "Search & Reporting" and various navigation buttons.

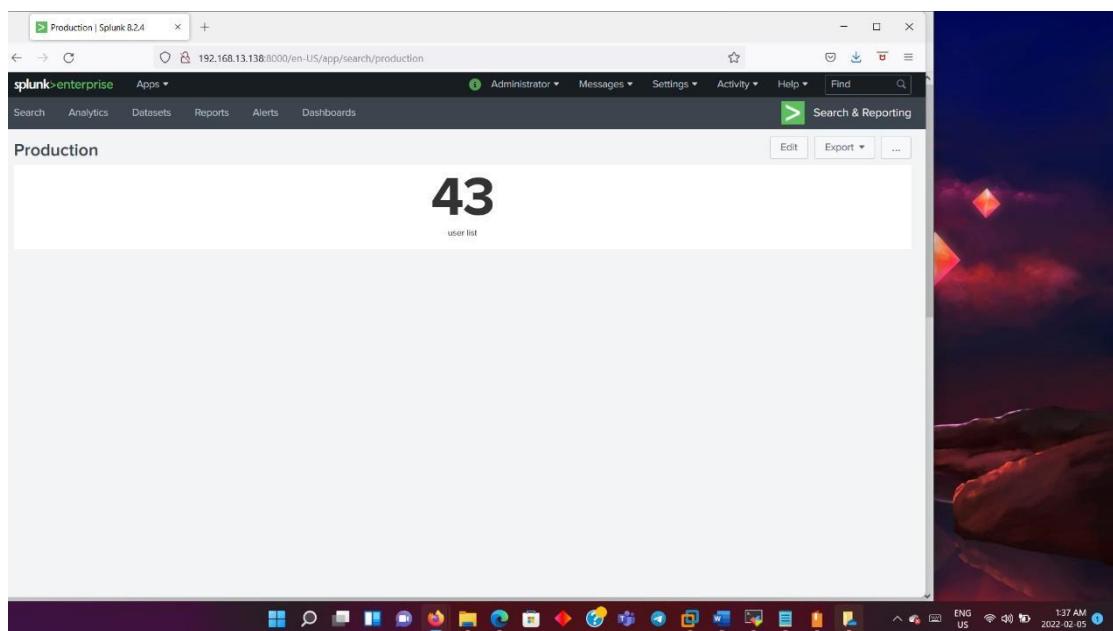
Query: `index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | dedup user | table user`

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

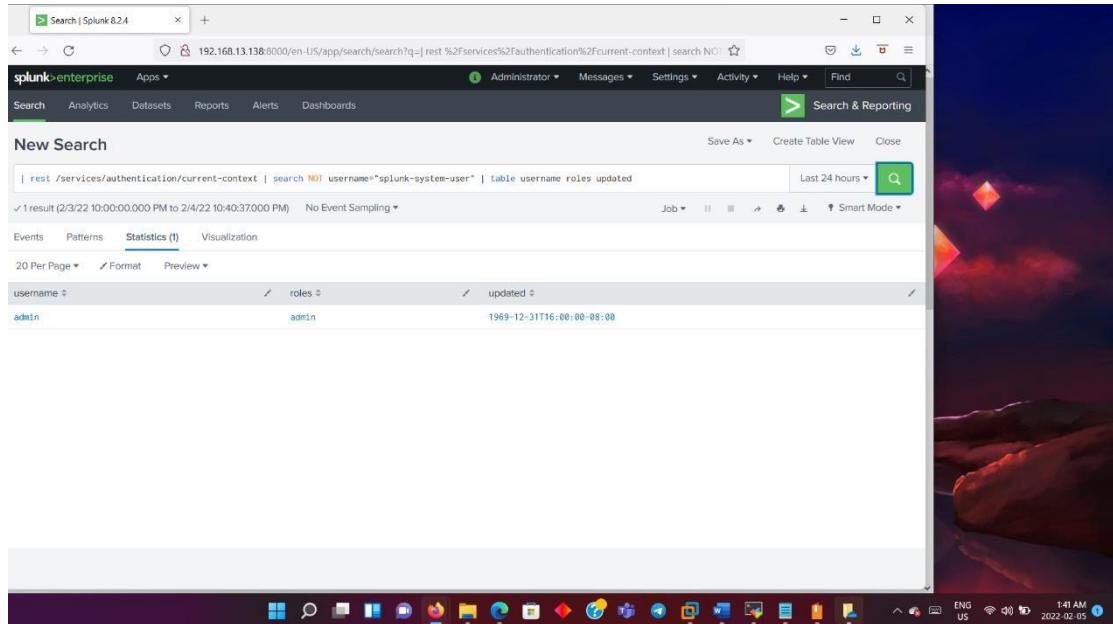


Query: `index=main source="unix:useraccounts" sourcetype="unix:useraccounts" | dedup user | table user | stats count`

Gives the count of users.

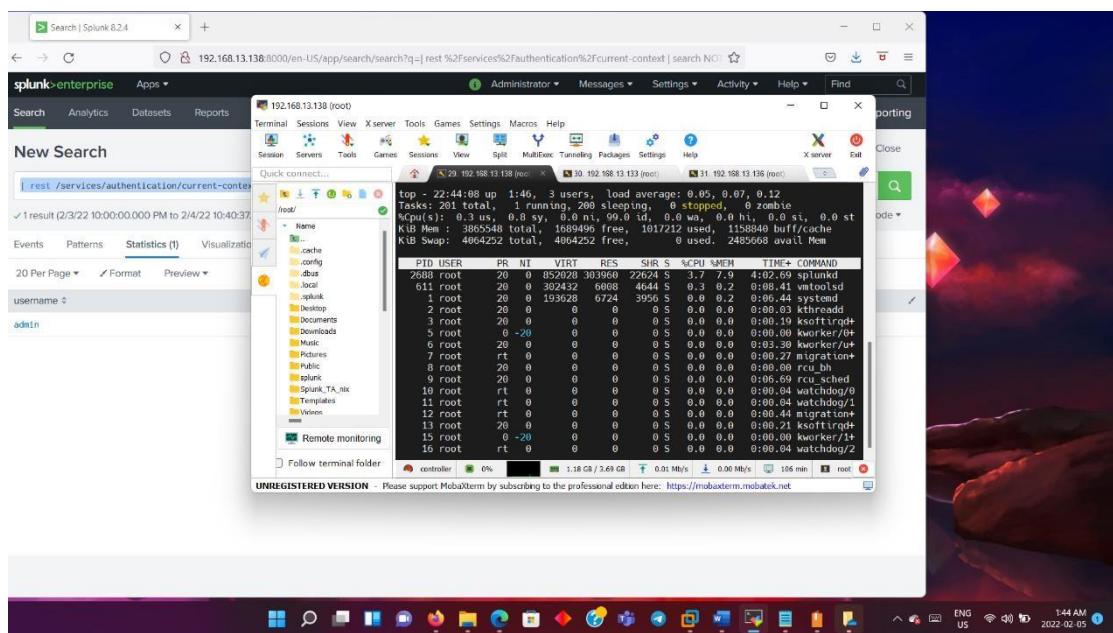


Result of production report saved in dashboard.



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

The screenshot shows the Splunk interface with a search bar containing the query: `index=main sourcetype=top host=* | table host,cpu_load_percent, _time | dedup host`. The results show 99,463 events from February 4, 2022, between 10:00:00.000 PM and 2:42:27.000 PM. The table includes columns for host, cpu_load_percent, and _time. The host column lists five hosts: controller, 192.168.13.136, Server1, 192.168.13.133, server2, and localhost.localdomain. The cpu_load_percent values range from 6.2 to 188.0. The _time column shows the timestamp for each event.

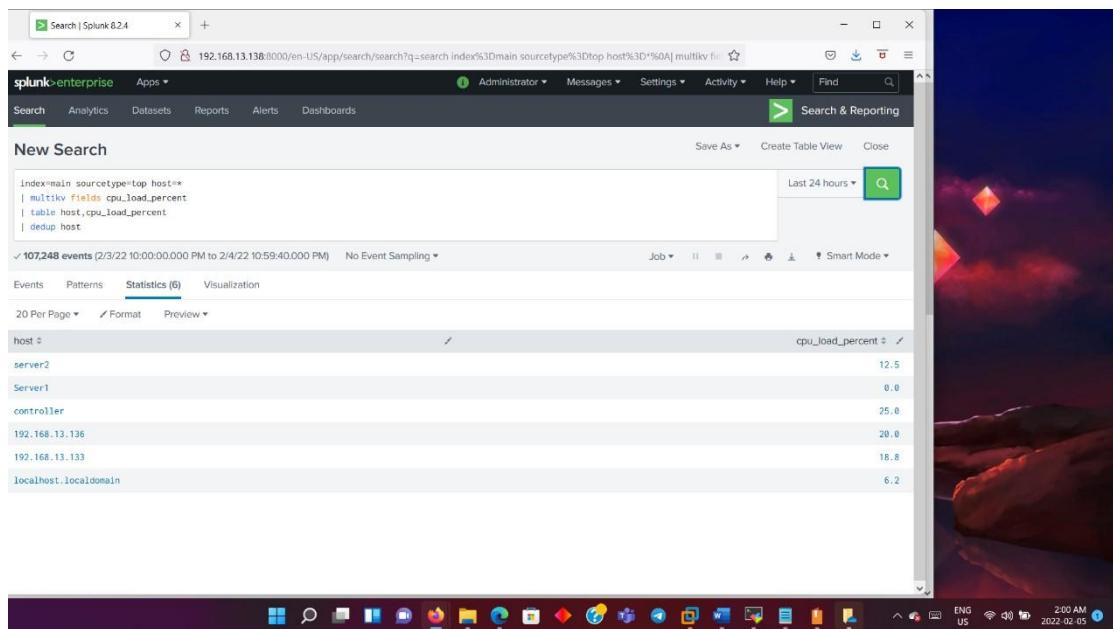
host	cpu_load_percent	_time
controller	13.3	2022-02-04 20:49:47
192.168.13.136	12.5	2022-02-04 20:49:22
Server1	188.0	2022-02-04 20:48:58
192.168.13.133	18.8	2022-02-04 20:48:01
server2	6.2	2022-02-04 21:58:37
localhost.localdomain	6.2	2022-02-04 19:33:40

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

The screenshot shows the same search results as the previous one, but the visualization is set to a bar chart. The chart displays the CPU load percentage for each host. The host names are on the x-axis, and the y-axis represents the CPU load percent, ranging from 0 to 25. The chart shows the following data points:

host	cpu_load_percent
192.168.13.136	20.8
Server1	6.2
controller	13.3
192.168.13.133	18.8
server2	6.2
localhost.localdomain	6.2

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



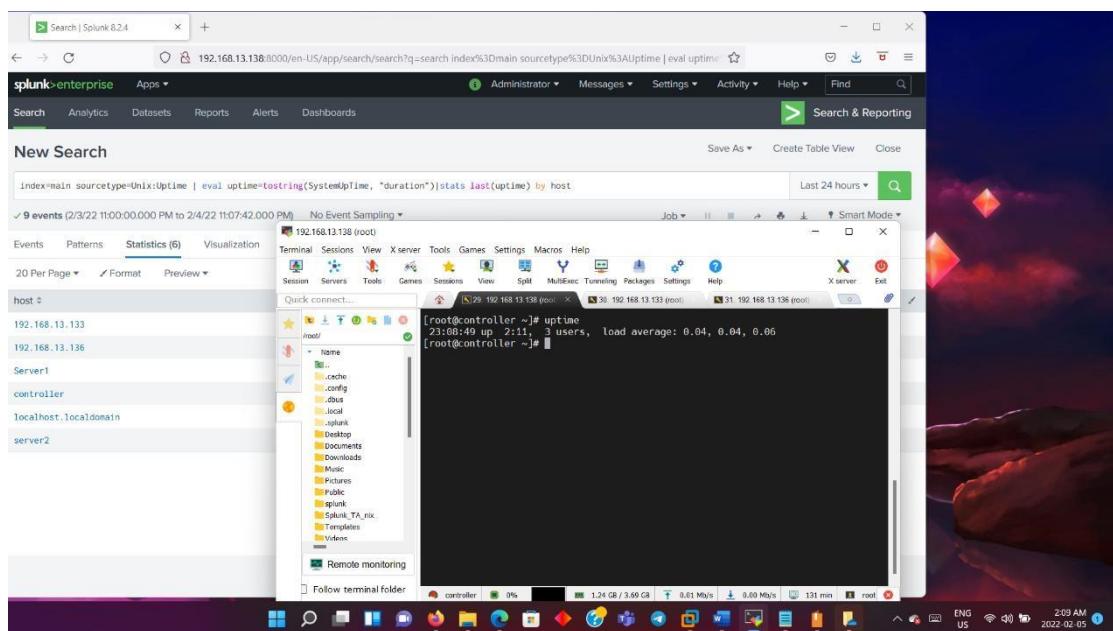
Query: `index=main sourcetype=top host=*`

```
| multikv fields cpu_load_percent
| table host,cpu_load_percent
| dedup host
```

Command used to extract field 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

The screenshot shows the Splunk 8.2.4 search interface. The search bar contains the query: `index=main source=top (host="server1" OR host="server2") | multikv fields pctMEM | table host,pctMEM | dedup host`. The results show 44,872 events from March 2, 2022, to April 2, 2022. The Statistics tab is selected, displaying a table with two rows: server2 (pctMEM: 4.3) and Server1 (pctMEM: 4.4). The desktop background is visible on the right.

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

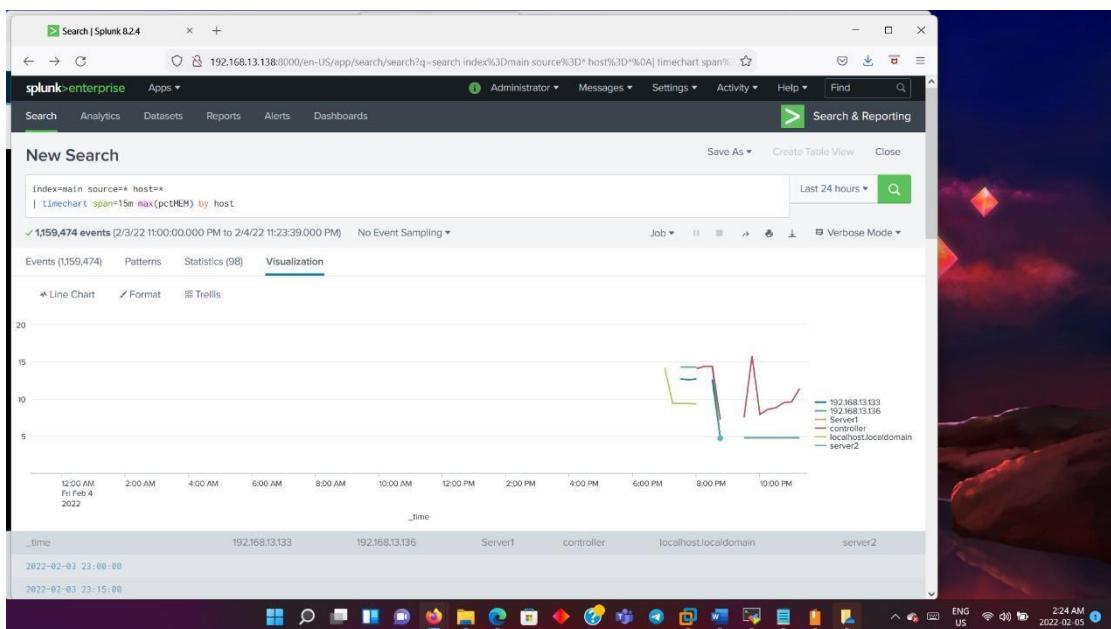
The screenshot shows the Splunk 8.2.4 search interface with the same query as above. The Statistics tab has been switched to a table view. The table header is "pctMEM" and it shows 20 values. The first few rows of the table are:

	Time	Event	0.0	0.2	0:06.08	systemd
20 Values, 100% of events			0.0	0.0	0:00.01	kthreadd
Reports			0.0	0.0	0:00.09	ksoftirqd+
Average over time			0.0	0.0	0:00.00	kworker/u+
Maximum value over time			0.0	0.0	0:00.23	kworker/u+
Minimum value over time			0.0	0.0	0:00.31	migration+
Top values			0.0	0.0	0:00.00	rcu_bh
Top values by time			0.0	0.0	0:00.04	watchdog/0
Rare values			0.0	0.0	0:00.00	
Events with this field			0.0	0.0	0:00.00	
Avg: 0.12468431860881724 Min: 0.0 Max: 4.8 Std Dev: 0.4505266082335784			0.0	0.0	0:00.00	
Top 10 Values			0.0	0.0	0:00.00	
Count			0.0	0.0	0:00.00	
%			0.0	0.0	0:00.00	

The screenshot shows the Splunk 8.2.4 search interface. The search bar contains the query: `index=main source=* host=* | timechart span=1h max(pctMEM) by host`. The results table shows 1,017,492 events matched over the last 24 hours. The table includes columns for _time, host, and max(pctMEM). The host column lists several hosts: 192.168.13.133, 192.168.13.136, Server1, controller, localhost.localdomain, and server2. The max(pctMEM) values range from 0.00 to 100.00. The interface also includes a visualization tab, a toolbar with various icons, and a system tray at the bottom.

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.

The screenshot shows a Splunk search interface with the following search 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
```

The search results table displays the following data:

host	Mount	UsePct
Server1	/	25
Server1	/boot	51
server2	/	25
server2	/boot	51
controller	/	25
controller	/boot	51
192.168.13.136	/	25
192.168.13.136	/boot	51
192.168.13.133	/	25
192.168.13.133	/boot	51
localhost.localdomain	/	25

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

The screenshot shows a Splunk search interface with the same search query as the previous screenshot. An X server terminal window is overlaid on the search results, displaying the output of the command:

```
[root@controller ~]# df -h
```

The terminal output shows the following disk usage:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	36G	9.0G	27G	25%	/dev
devpts	1.0M	0K	1.0M	0%	/dev/pts
/tmp	1.9G	0K	1.9G	0%	/tmp
tmpfs	1.9G	9.1M	1.9G	1%	/run
tmpfs	1.9G	0	1.9G	0%	/sys/fs/cgroup
/dev/sda1	297M	152M	146M	51%	/boot
tmpfs	378M	16K	378M	1%	/run/user/0

The screenshot shows a Splunk Enterprise search interface. The search bar contains the query: `index=main sourcetype=top host=* | stats max(pctCPU) as maxCPU by host, COMMAND, _time | sort -maxCPU | dedup 5 host`. The results table displays 133,179 events from 2/3/22 11:00:00.000 PM to 2/4/22 11:37:43.000 PM. The table has columns for host, COMMAND, _time, and maxCPU. The top five entries are:

host	COMMAND	_time	maxCPU
localhost.localdomain	splunkd	2022-02-04 19:04:26	200.0
localhost.localdomain	splunkd	2022-02-04 19:06:40	150.0
controller	splunkd	2022-02-04 23:23:57	113.3
controller	splunkd	2022-02-04 23:14:57	106.2
Server1	kexec	2022-02-04 20:48:58	100.0

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.

The screenshot shows a Splunk Enterprise search interface. The search bar contains the 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"`. The results table displays 138,455 events from 2/3/22 11:00:00.000 PM to 2/4/22 11:45:02.000 PM. The table has columns for host, severity, and %CPU Utilized. The top five entries are:

host	severity	%CPU Utilized
192.168.13.133	Critical	87.56666666666666
192.168.13.133	Normal	0.09592444548945867
192.168.13.133	warning	59.29
192.168.13.136	Critical	88.09000000000001
192.168.13.136	Normal	0.1199838717143738

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"`

The screenshot shows the Splunk Enterprise search interface. The search bar contains the query: `index=main source="auditd" ("failed") user|table user,host,_time`. The results table shows 18 events from February 3, 2022, to February 4, 2022. The columns are user, host, and _time. The data includes various hosts like server2, 192.168.13.136, and localhost.localdomain, with timestamps ranging from 2022-02-04 11:45:01 to 2022-02-04 17:28:31.

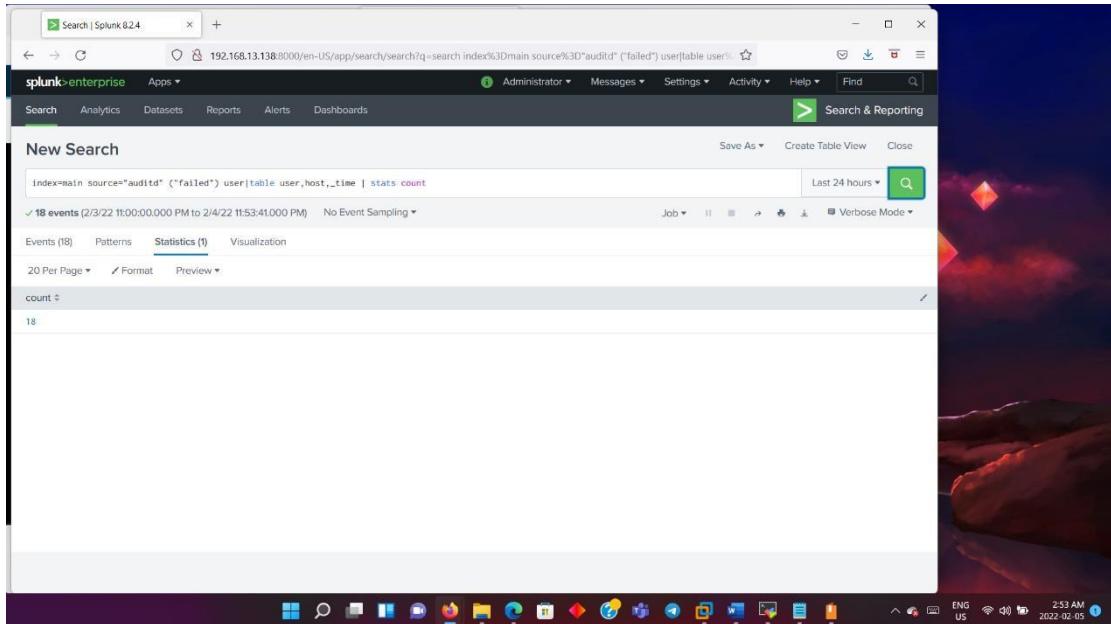
user	host	_time
	server2	2022-02-04 20:51:58.128
	192.168.13.136	2022-02-04 17:28:31.728
	192.168.13.136	2022-02-04 17:28:29.828
	192.168.13.136	2022-02-04 16:38:31.588
	localhost.localdomain	2022-02-04 12:25:19.168
	localhost.localdomain	2022-02-04 12:25:19.168
	localhost.localdomain	2022-02-04 11:58:31.798
	localhost.localdomain	2022-02-04 11:58:29.888
	192.168.13.136	2022-02-04 11:45:10.078
	192.168.13.136	2022-02-04 11:45:08.138
	192.168.13.136	2022-02-04 11:45:03.498
	192.168.13.136	2022-02-04 11:45:01.178

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

Shows failed ssh login

The screenshot shows a Linux terminal session on a root account. The user is attempting to connect to another host via SSH. The terminal output shows:

```
[root@controller ~]# uptime
[23:39:49 up 21:11, 1 user,  load average: 0.04, 0.04, 0.06
[root@controller ~]# ssh 192.168.13.133
Warning: Permanently added '192.168.13.133' (EDSA) to the list of known hosts.
ECDSA key fingerprint is 45:e2:59:c9:d1:0c:90:b7:83:cde8:60:96:09:19:00.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.13.133' (EDSA) to the list of known hosts.
root@192.168.13.133's password:
Last login: Fri Feb 4 21:39:59 2022 from 192.168.13.1
abrt has detected 1 problem(s). For more info run: abrt-cli list --since 1644039
690
[root@Server1 ~]#
```



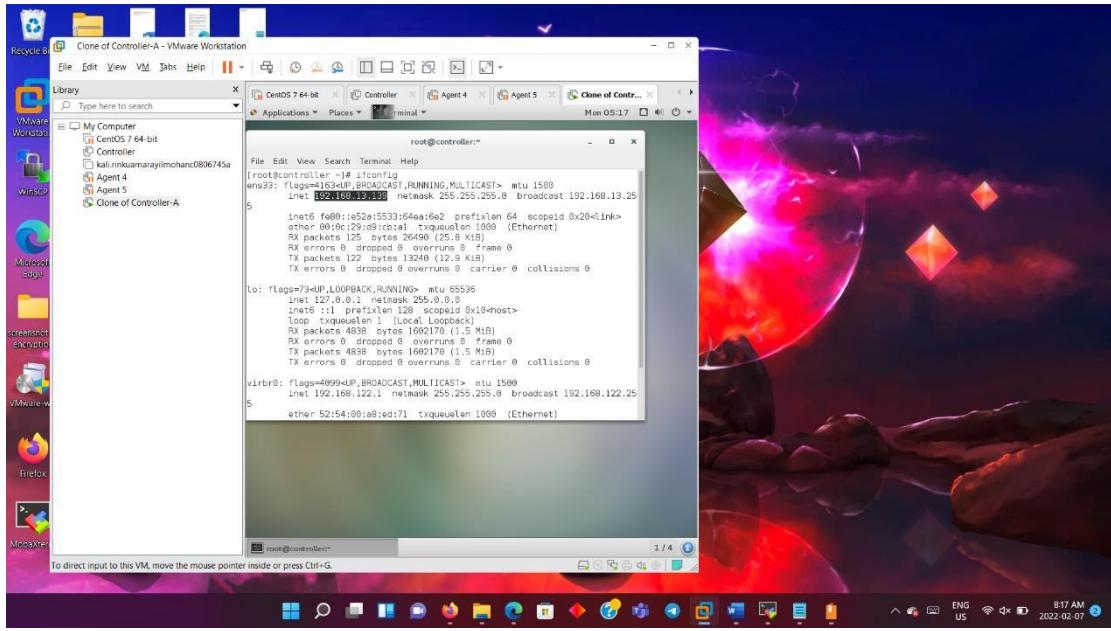
Query: index=main source="auditd" ("failed") user|table user,host,_time | stats count

It gives the stat count of the failed ssh login

\

Part 1b

Incident response Brute-Force attack

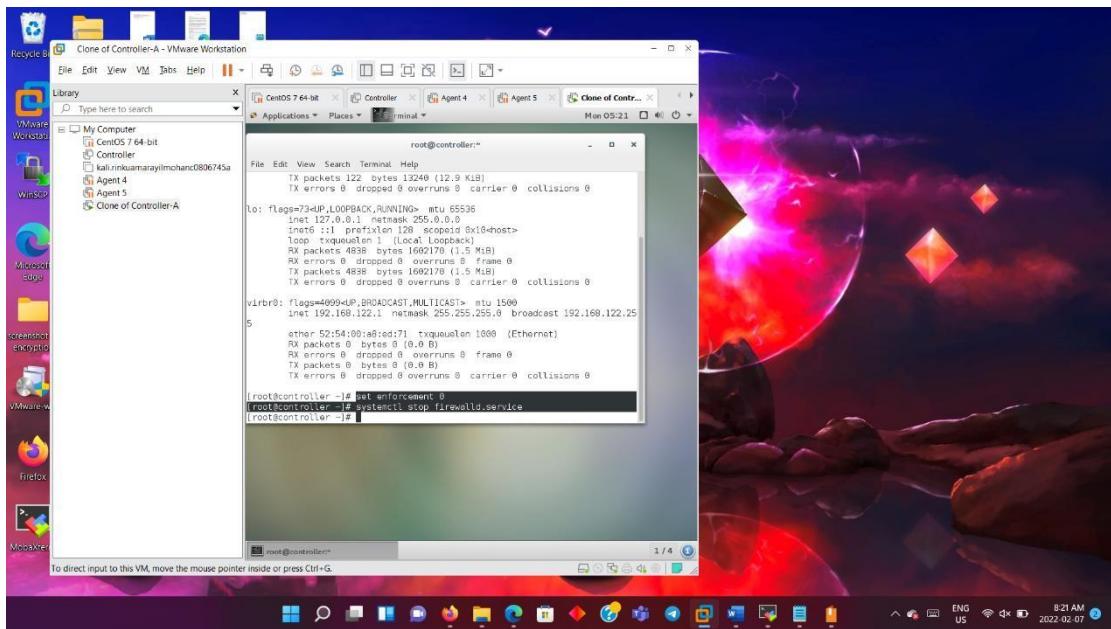


Created clone of controller and named in Clone of controller A

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

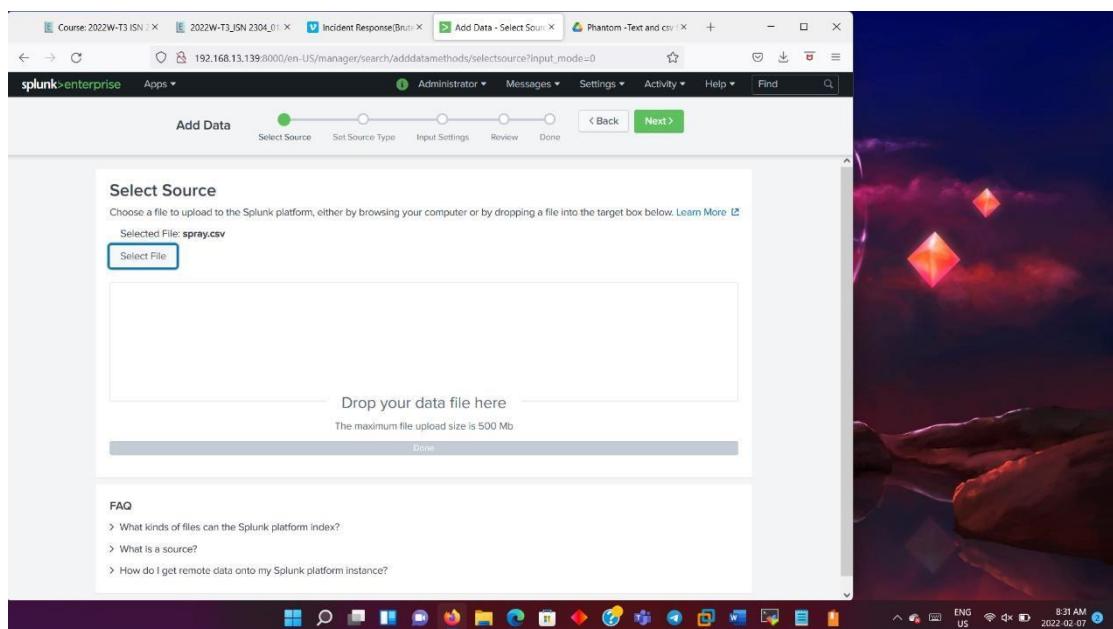
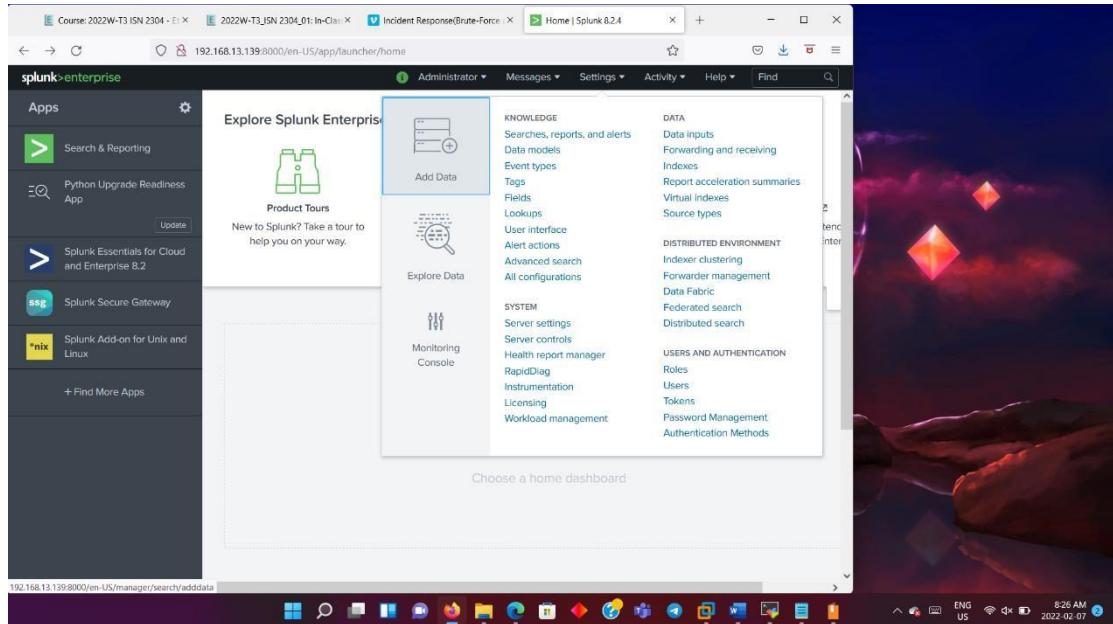
systemctl stop firewalld.service



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.

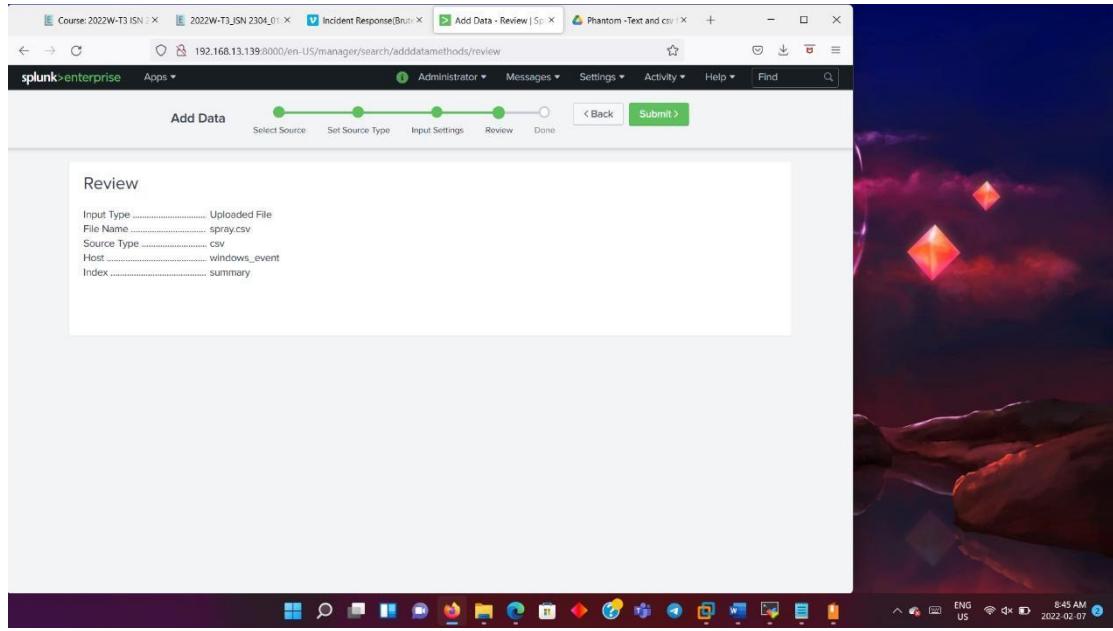
The screenshot shows the Splunk Add Data - Set Source Type interface. The source is set to 'spray.csv'. The interface displays two log entries:

	_time	Date and Time	Event ID	EXTRA_FIELD_6
1	11/19/19 5:53:57 2000 PM	11/19/2019 5:53:57 PM	4634	An account was logged off. Subject: Security SYSTEM Account Name: DCIS Account Dom EFLU Logon ID: 0x3F2E2B Logon Type: 3 This is generated when a logon session is destroyed; may be positively correlated with a logon event using the Logon ID value. Logon IDs are only between reboots on the same computer.
2	11/19/19 5:53:47 2000 PM	11/19/2019 5:53:47 PM	4624	An account was successfully logged on. Subject: Security ID: NULL SID Account Name: - Account Domain: - Logon ID: 0x0 Logon Information: I Type: 3 Restricted Admin Mode: - Virtual Account Elevated Token: Yes Impersonation Level: Impersonation New Logon: Security ID: SYSTEM Account Name: DCIS Account Domain: EFLU Logon ID: 0x4FB906 Linked Logon ID: 0x0 N Account Name: - Network Account Domain: -

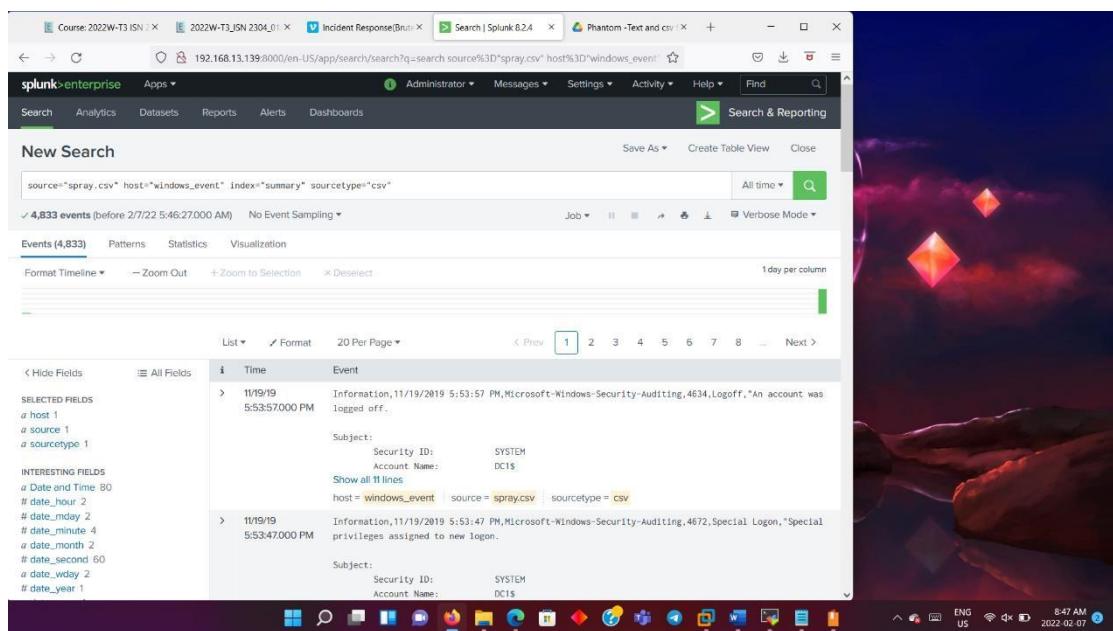
In this we get the source type, timestamp details. Now we click on next.

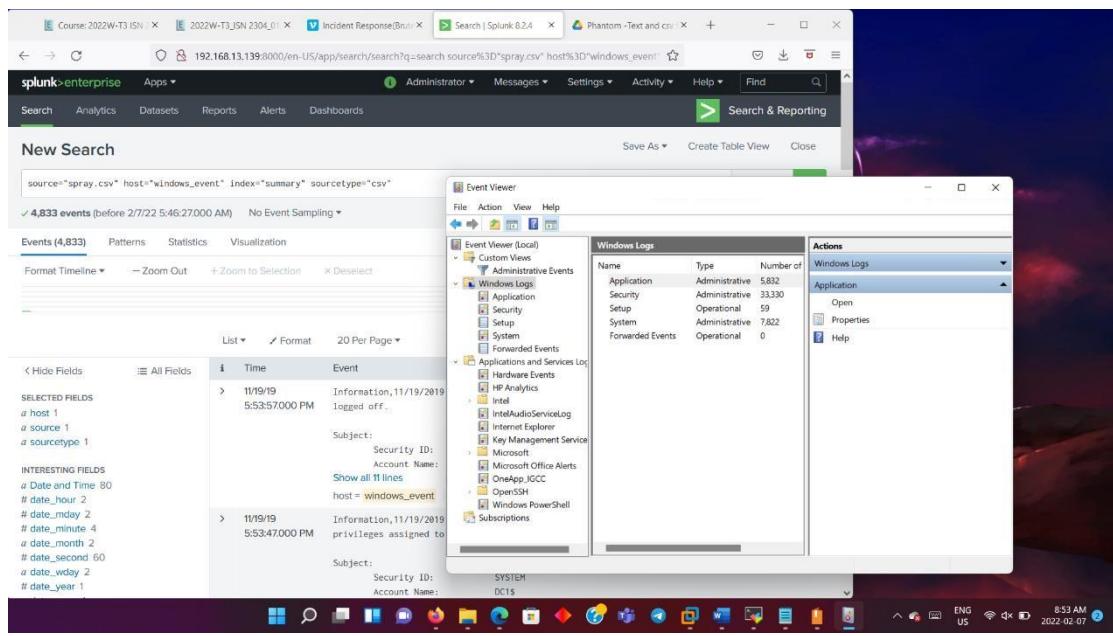
The screenshot shows the Splunk Add Data - Input Settings interface. Under 'Host', the 'Constant value' option is selected, and the 'Host field value' is set to 'windows_event'. Under 'Index', the 'summary' index is selected. The 'FAQ' section includes links to 'How do indexes work?' and 'How do I know when to create or use multiple indexes?'. The status bar at the bottom right shows '8:43 AM' and '2022-02-07'.

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.

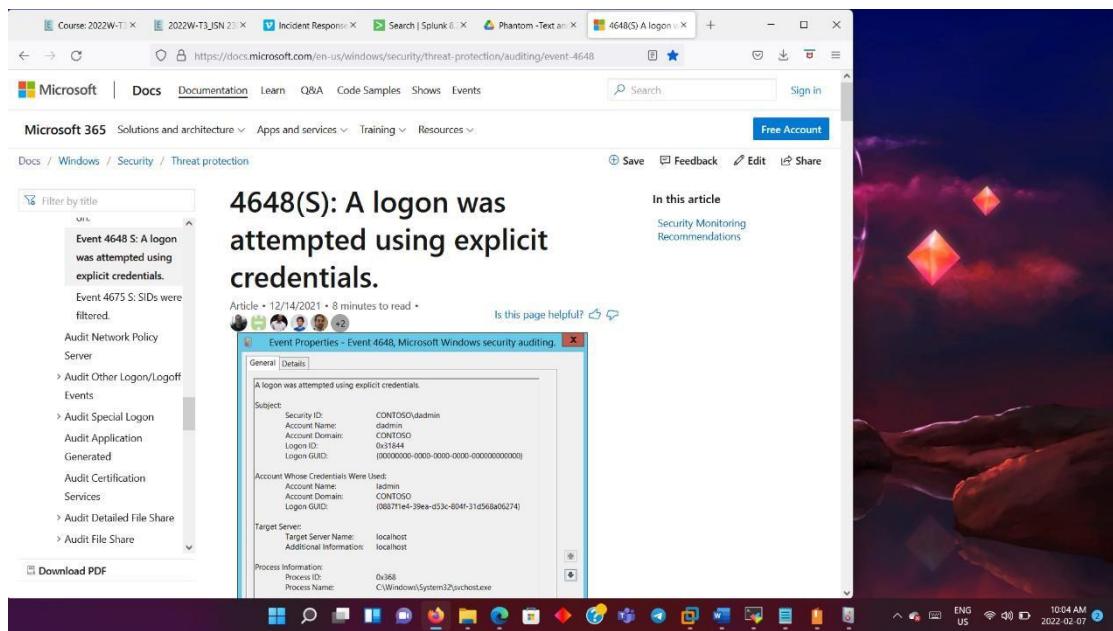
Now search for query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" | stats count by "Event ID" "Task Category"

Search the above query selecting all time.

Event ID	Task Category	count
1102	Log clear	1
4616	Security State Change	1
4624	Logon	16
4625	Logon	2386
4634	Logoff	15
4648	Logon	2387
4672	Special Logon	16
4768	Kerberos Authentication Service	2
4769	Kerberos Service Ticket Operations	5
4776	Credential Validation	4

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.

A screenshot of the Splunk Enterprise interface showing search results for Windows events. The search query is "source='spray.csv' host='windows_event' index='summary' sourcetype='csv' | stats count by *Event ID* *Task Category*". The results table lists several events, with Event ID 4625 selected. A context menu for Event ID 4625 is open, with "View events" highlighted. The desktop taskbar at the bottom shows various icons.

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

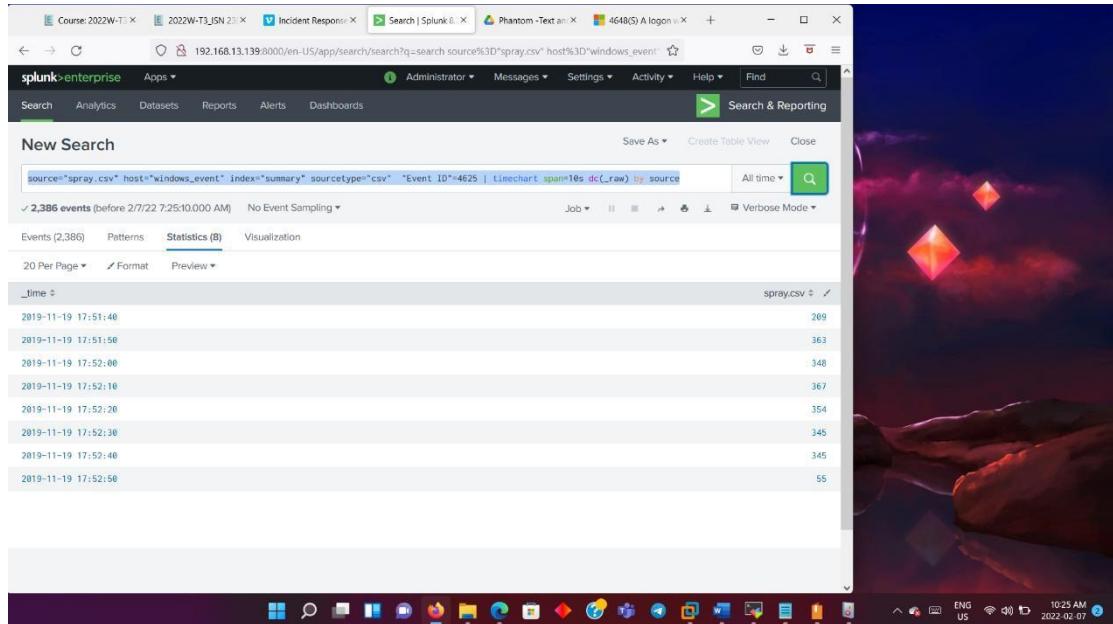
The screenshot shows the Splunk Enterprise interface with a search results page. The search query is: `source="spray.csv" host="windows_event" index="summary" sourcetype="csv" "Event ID":4625`. The results show 2,386 events from 2/7/22 7:12:15:000 AM. The event details are displayed in a table:

	Time	Event
>	11/19/19 5:52:51:000 PM	Information,11/19/2019 5:52:51 PM,Microsoft-Windows-Security-Auditing,4625,Logon,"An account failed to log on.
		Subject: Security ID: NULL SID Account Name: - Show all 50 lines
>	11/19/19 5:52:51:000 PM	host = windows_event source = spray.csv sourcetype = csv Information,11/19/2019 5:52:51 PM,Microsoft-Windows-Security-Auditing,4625,Logon,"An account failed to log on.
		Subject: Security ID: NULL SID Account Name: -

The screenshot shows the same Splunk interface with the Event Actions panel expanded. The selected fields are: host, source, and sourcetype. The event details are identical to the previous screenshot, but the expanded panel shows the following actions:

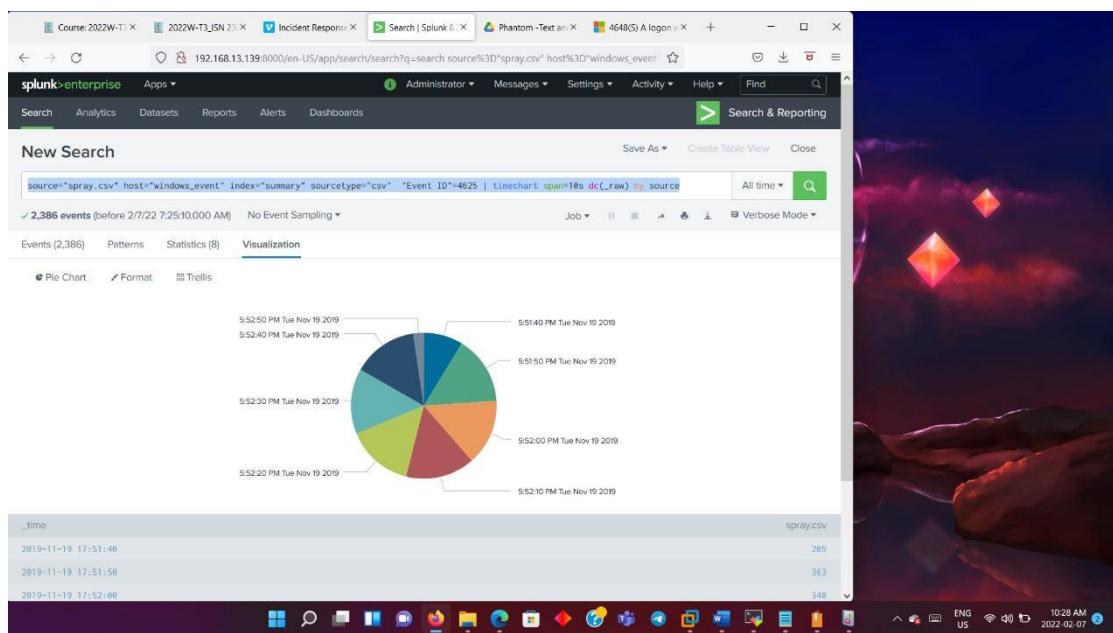
Type	Field	Value	Actions
Selected	host	windows_event	<input type="button" value="▼"/>
	source	spray.csv	<input type="button" value="▼"/>
	sourcetype	csv	<input type="button" value="▼"/>
Event	Date and Time	11/19/2019 5:52:51 PM	<input type="button" value="▼"/>
	EXTRA_FIELD_6	An account failed to log on. Subject: Security ID: NULL SID Account Name: - Account Domain: - Logon ID: 0x0 Logon Type: 3 Account For Which Logon Failed: Security ID: NULL SID Account Name: ffwinklesstockings Account Domain: ELFU Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xC000006D Sub Status: 0xC000006A Process Information: Caller Process ID: 0x0 Caller Process Name: - Network Information: Workstation Name: DC1 Source Network Address: 127.0.0.1 Source Port: 54777 Detailed Authentication Information: Logon Process: NtLmSpn Authentication Package: NTLM Transited Services: - Package Name (NTLM only): - Key Length: 0 This event is generated when a logon request fails. It is generated on the computer where access was attempted. The Subject fields indicate the account on the local system which requested the logon. This is mos	<input type="button" value="▼"/>

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

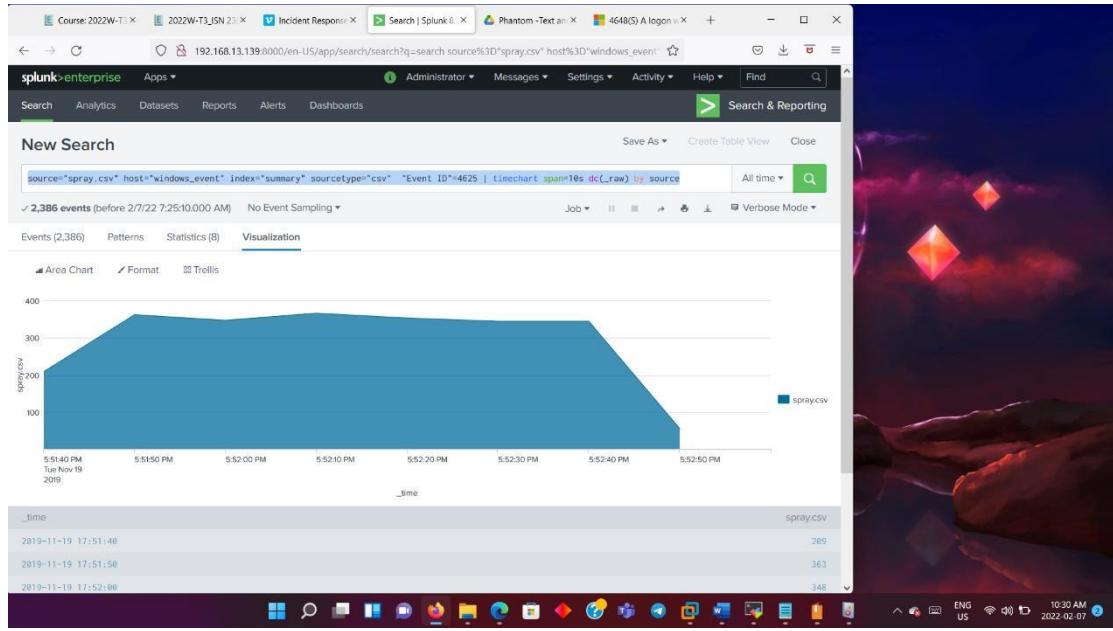


Query: source="spray.csv" host="windows_event" index="summary" sourcetype="csv" "Event ID"=4625 | timechart span=10s dc(_raw) by source

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

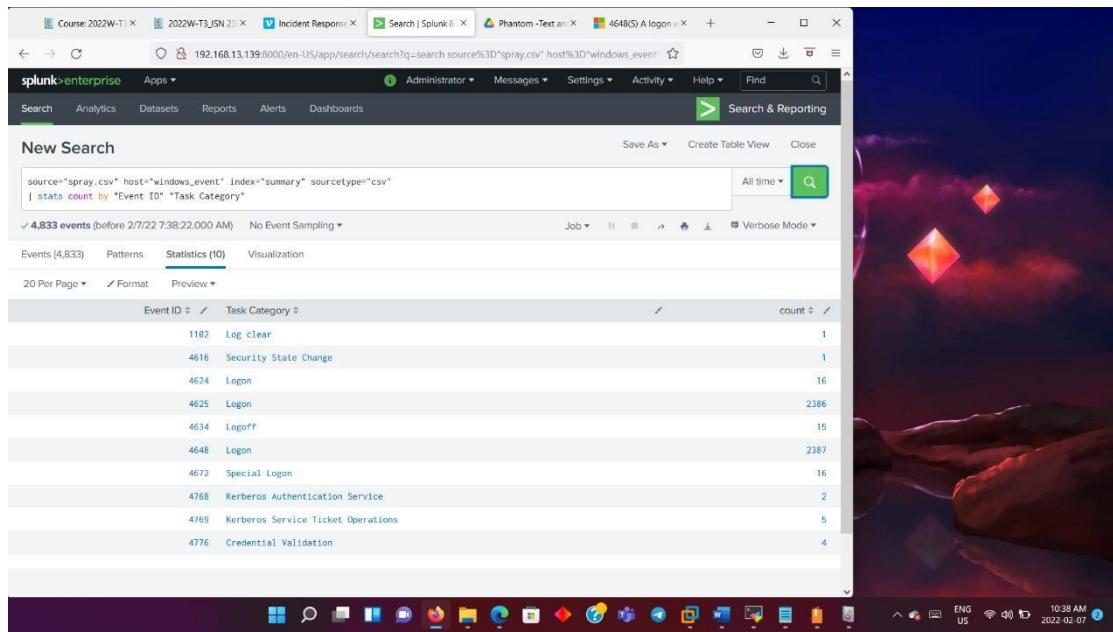


We can see the result in the visualization as pie chart as well.



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 screenshot shows a Splunk search interface with the following details:

- Search Query:** `source="spray.csv" host="windows_event" index="spray.csv" sourcetype="csv" EventID=4625`
- Event Details:**
 - Time: 11/19/2019 5:52:51 PM
 - Subject: Security ID: NULL SID
 - Account Name: -
 - Description: An account failed to log on. Subject: Security ID: NULL SID Account Name: - Account Domain: - Logon ID: 0x0 Logon Type: 3 Account For Which Logon Failed: Security ID: NULL SID Account Name: ftnwkhlestockings Account Domain: ELFU Failure Information: Failure Reason: Unknown user name or bad password. Status: 0xC000006D Sub Status: 0xC000006A Process Information: Caller Process ID: 0x0 Caller Process Name: - Network Information: Workstation Name: DC1 Source Network Address: 127.0.0.1 Source Port: 54777 Detailed Authentication Information: Logon Process: Ntlmssp Authentication Package: NTLM Transited Services: Package Name (NTLM only): - Key Length: 0 This event is generated when a logon request fails. It is generated on the computer where access was attempted. The Subject fields indicate the account on the local system which requested the logon. This is most commonly a service such as the Server service, or a local process such as Winlogon.exe or Services.exe. The Logon Type field indicates the kind of logon that was requested. The most common types are 2 (interactive) and 3 (network). The Process Information fields indicate which account and process on the system received the logon request.
- Event Actions:** A dropdown menu is open, showing options like "Selected", "host", "source", "sourcetype", and "EXTRA_FIELD_6".
- Bottom Status Bar:** Shows system information: ENG US, 10:49 AM, 2022-02-07.

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.

The screenshot shows a Splunk search interface with the following details:

- Search Query:** `source="spray.csv" host="windows_event" index="spray.csv" sourcetype="csv" EventID=4625`
- Visualizations:** A timechart is displayed, showing the count of events per second over a 1-second span. The chart shows a single data series named "spray.csv" with values of 0 for each second from 2019-11-19 17:51:44 to 2019-11-19 17:51:55.
- Bottom Status Bar:** Shows system information: ENG US, 12:23 PM, 2022-02-07.

For every second it shows the attack.

The screenshot shows the Splunk Enterprise interface with a search results table. The search query is:

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

The results table has the following columns: _time, _index, and _count. The data shows 2,386 events from November 19, 2019, at 17:51:44 to 17:51:55. The count column is all zeros.

_time	_index	_count
2019-11-19 17:51:44		0
2019-11-19 17:51:45		0
2019-11-19 17:51:46		0
2019-11-19 17:51:47		0
2019-11-19 17:51:48		0
2019-11-19 17:51:49		0
2019-11-19 17:51:50		0
2019-11-19 17:51:51		0
2019-11-19 17:51:52		0
2019-11-19 17:51:53		0
2019-11-19 17:51:54		0
2019-11-19 17:51:55		0

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.

The screenshot shows the Splunk Enterprise interface with a search results table. The search query is:

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

The results table has the following columns: _time, _index, and _count. The data shows 2,387 events from November 19, 2019, at 17:51:44 to 17:51:55. The count column is all zeros.

_time	_index	_count
2019-11-19 17:51:44		0
2019-11-19 17:51:45		0
2019-11-19 17:51:46		0
2019-11-19 17:51:47		0
2019-11-19 17:51:48		0
2019-11-19 17:51:49		0
2019-11-19 17:51:50		0
2019-11-19 17:51:51		0
2019-11-19 17:51:52		0
2019-11-19 17:51:53		0
2019-11-19 17:51:54		0
2019-11-19 17:51:55		0

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

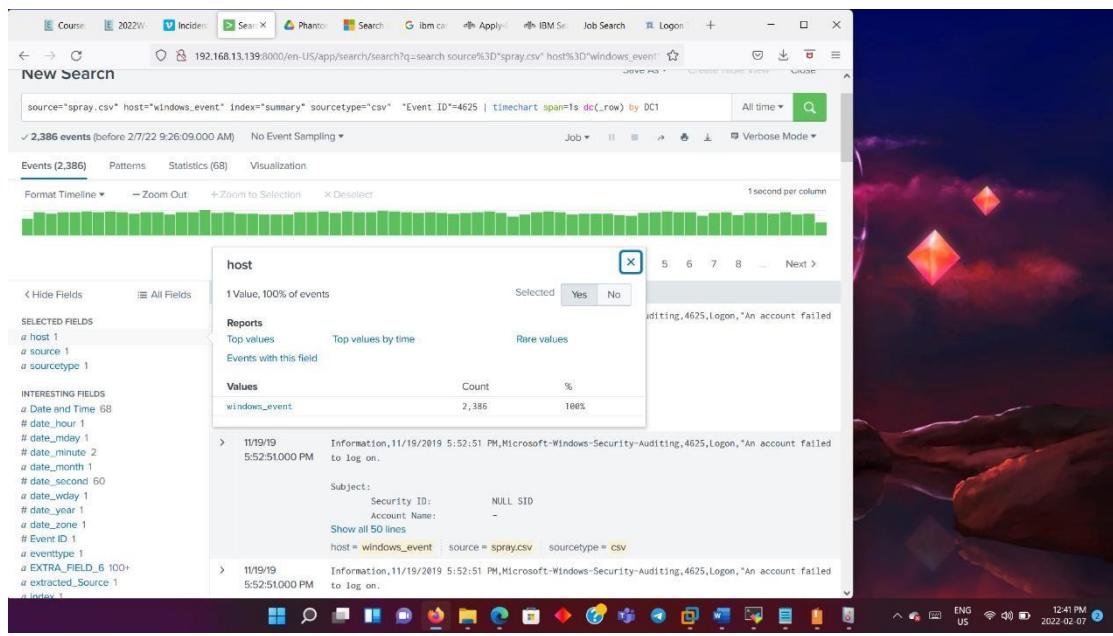
Failure Reason: Unknown user name or bad password.

Status: 0xC000006D

Sub Status: 0xC000006A

Logon type 3 is justified above.

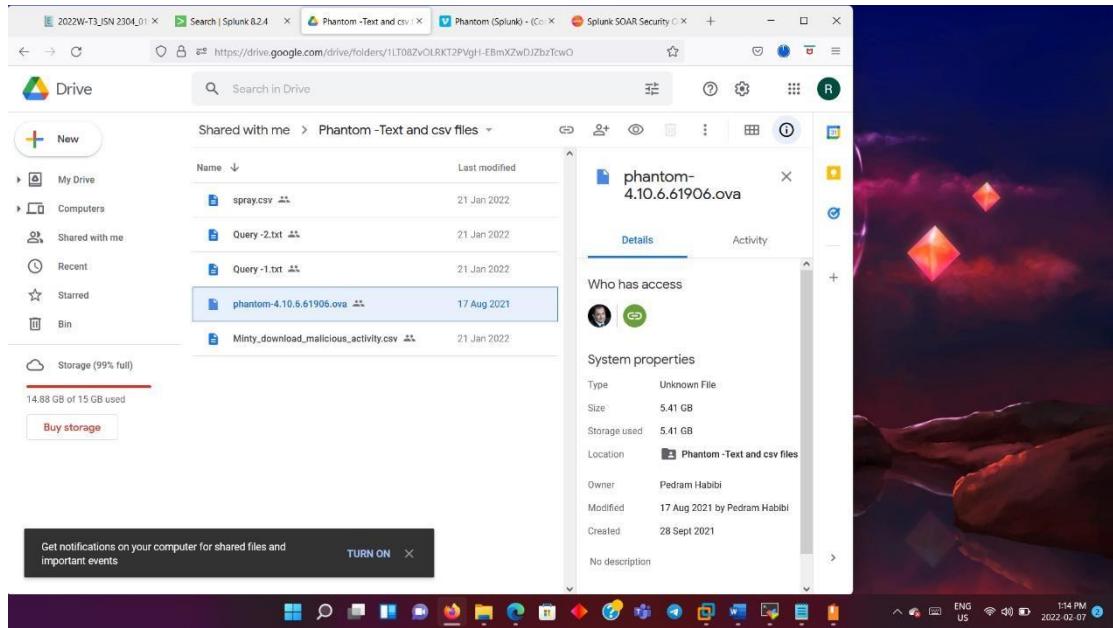
We can also get the details by checking the status in Microsoft docs.



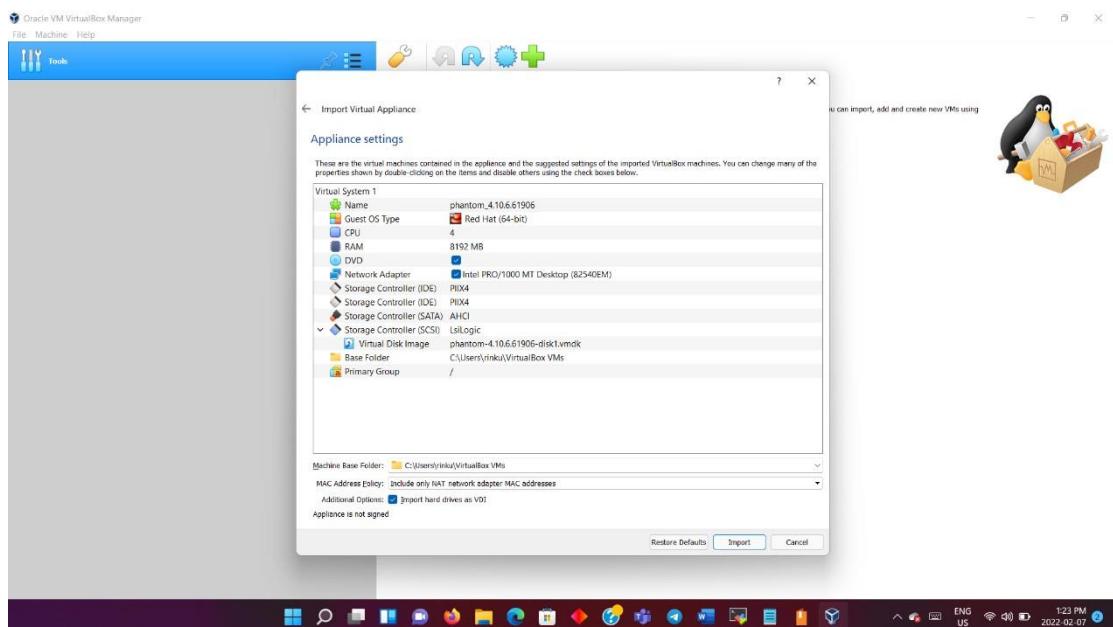
Finally, we can also get many details like ghost name, source, source type etc.

Part 2a

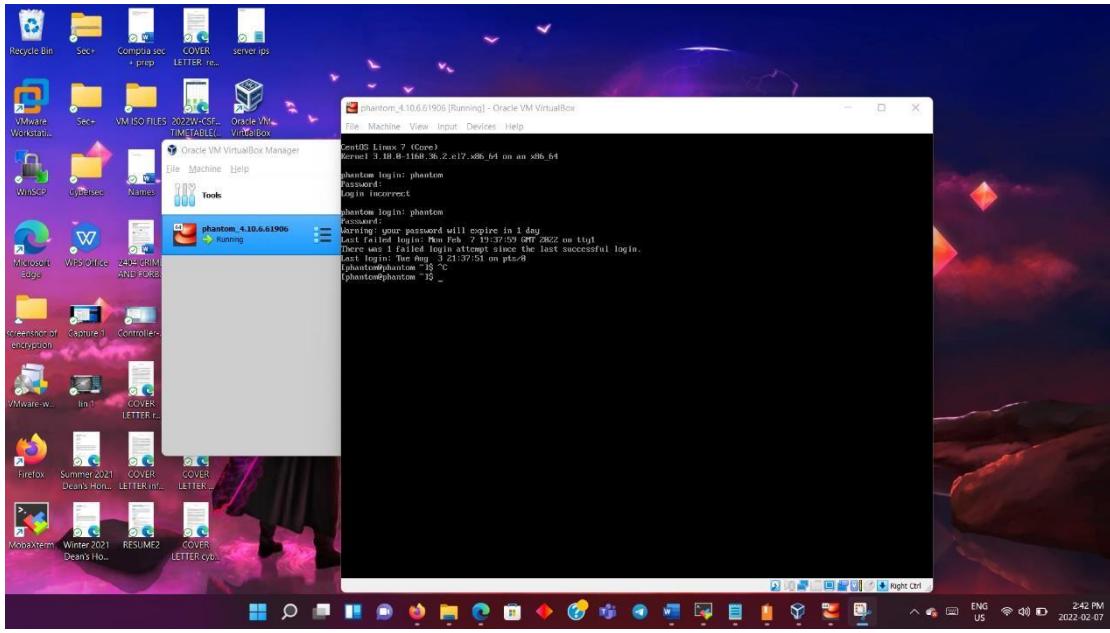
SOAR- Phantom (Splunk)



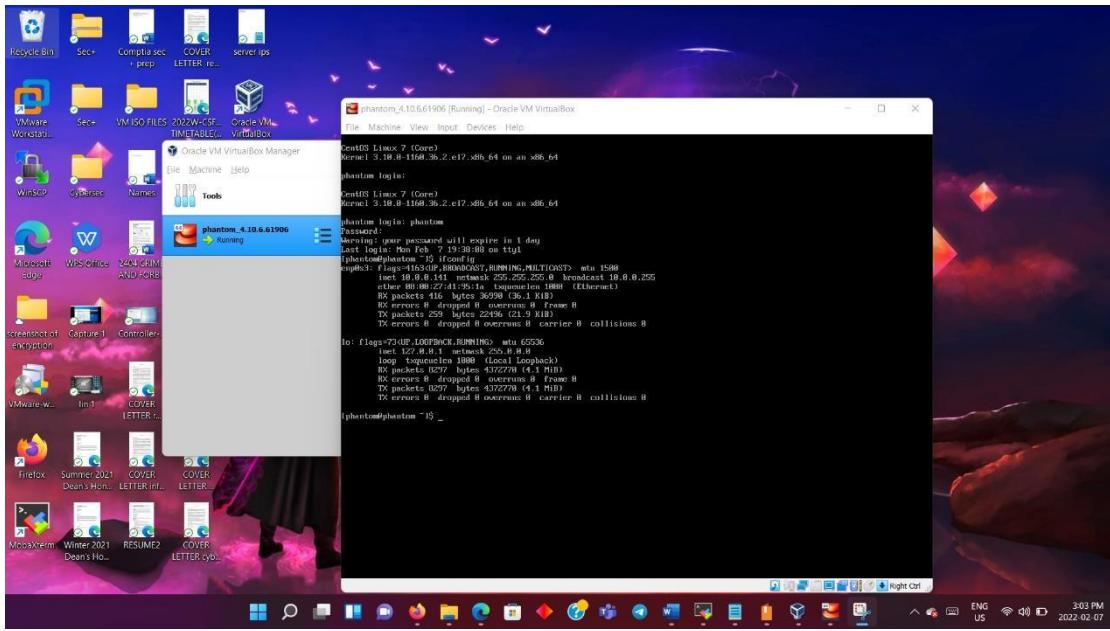
We first download the phantom community edition from google. (Splunk SOAR Security Orchestration & Automation)



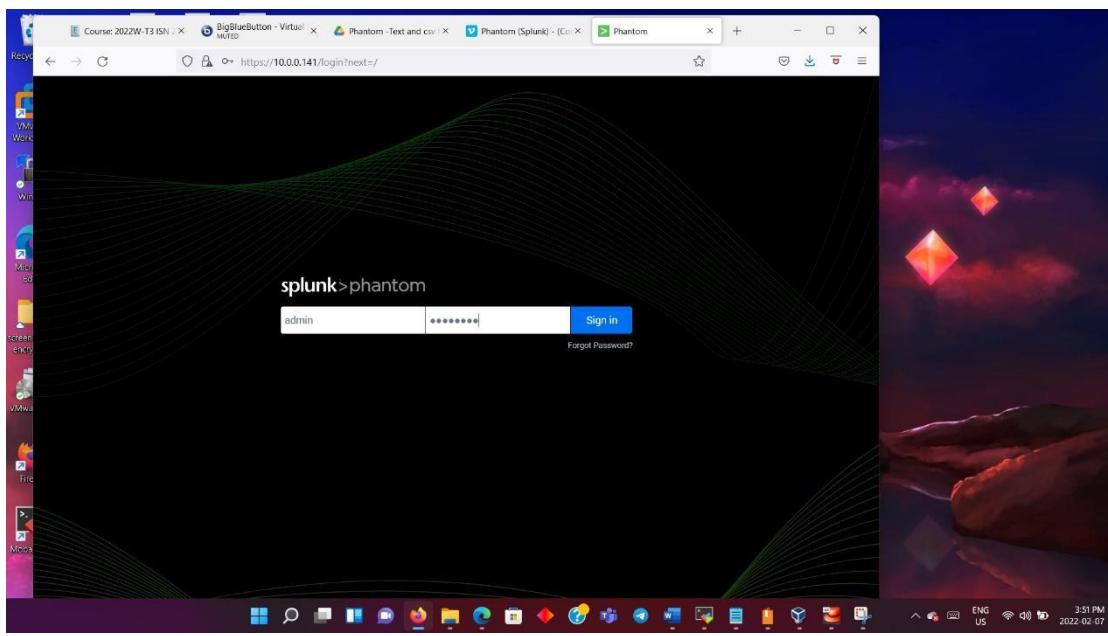
We first import the virtual application.



Logged in using username: phantom
Password: password



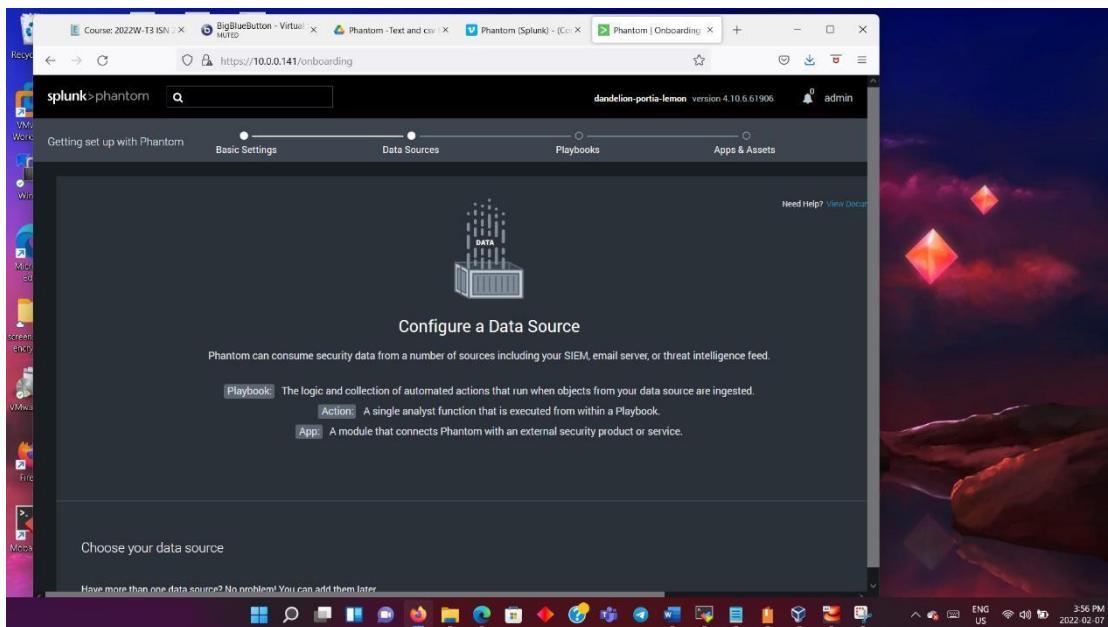
The IP address of phantom is: 10.0.0.141



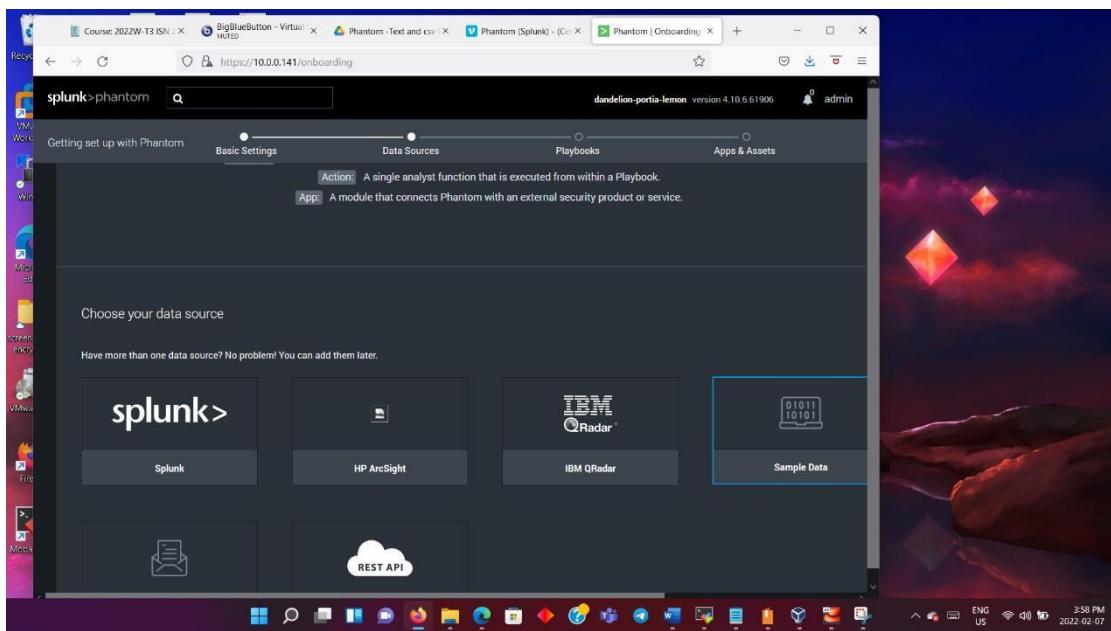
Logged in to the website with the phantom IP address, so Splunk phantom web page got launched.

Logged in using the username: admin
Password: password

Later login and change the credentials.



Now set the data source.



Select sample data

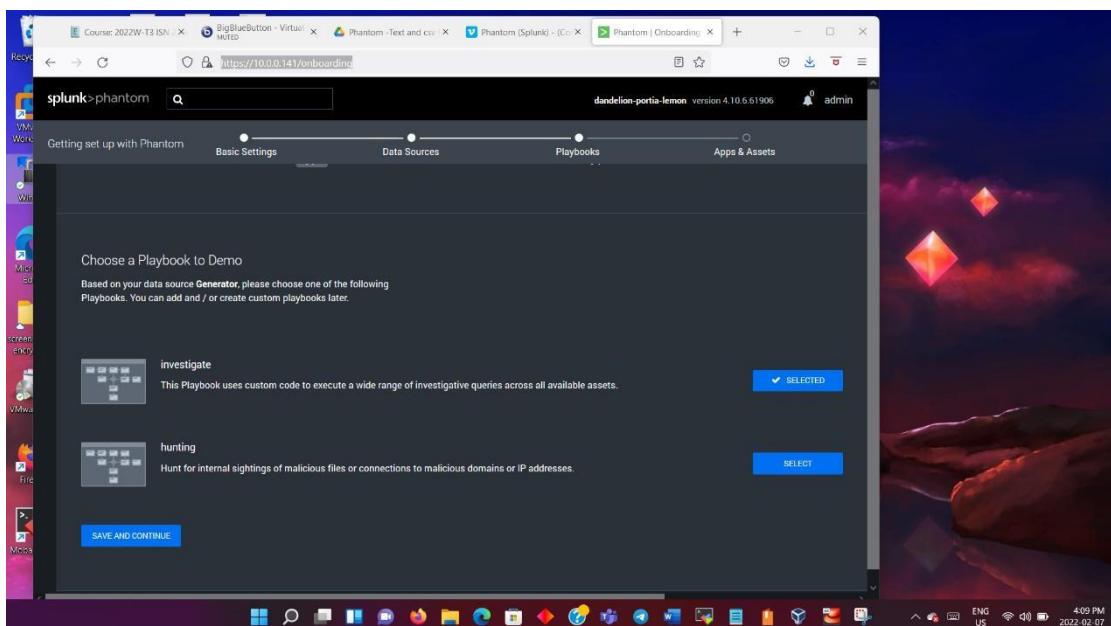
Provide the asset name as sample data

Events to generate :20

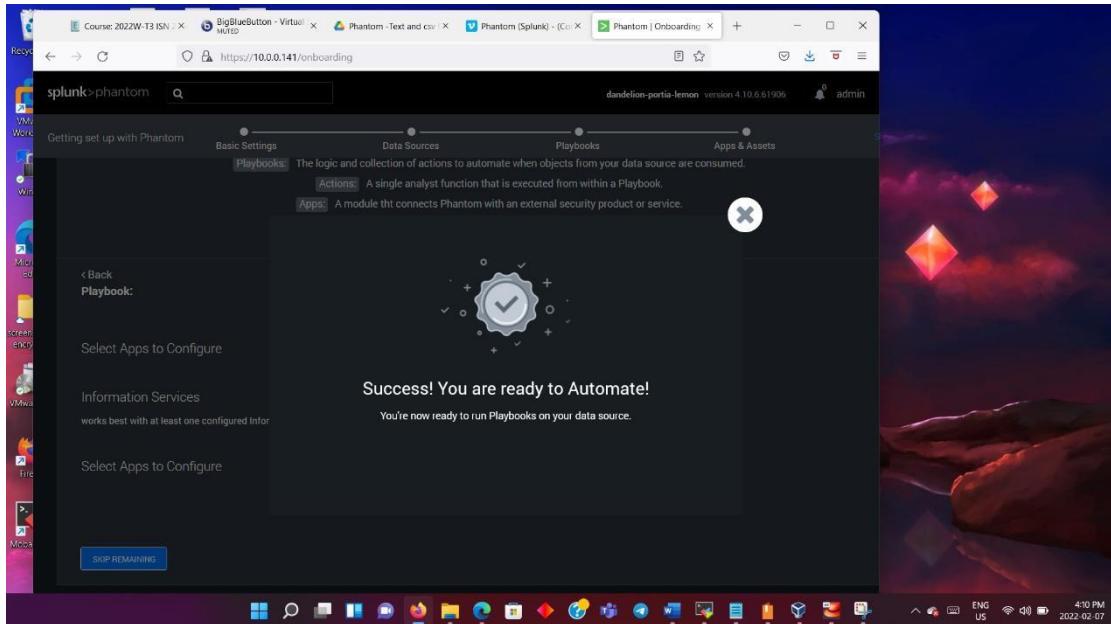
Artifacts per events to generate :2

Container label: email

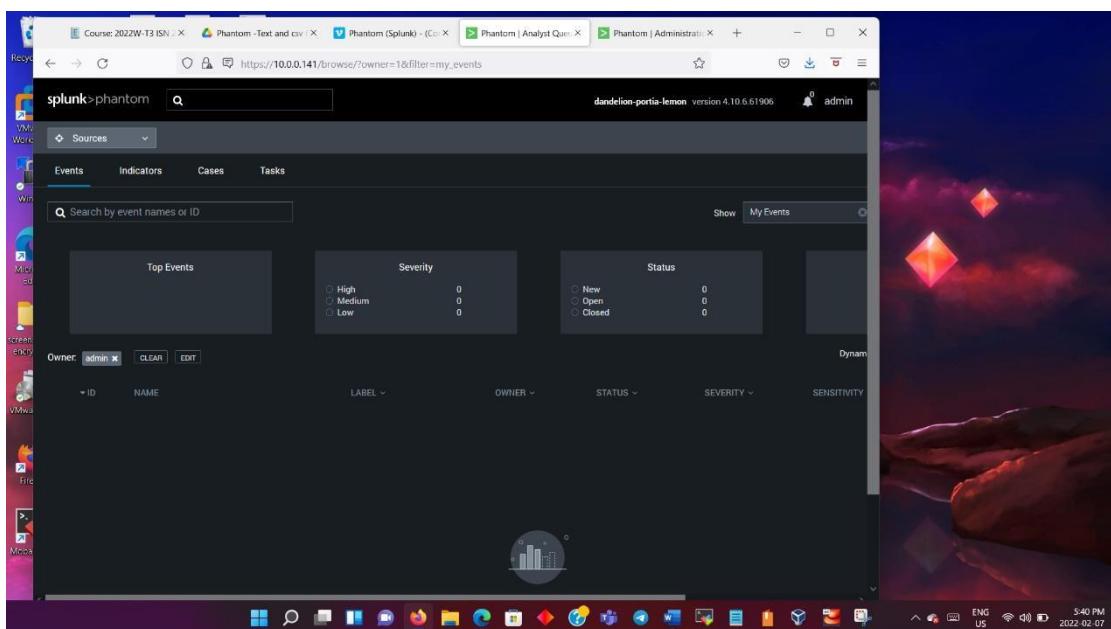
And click on save and continue



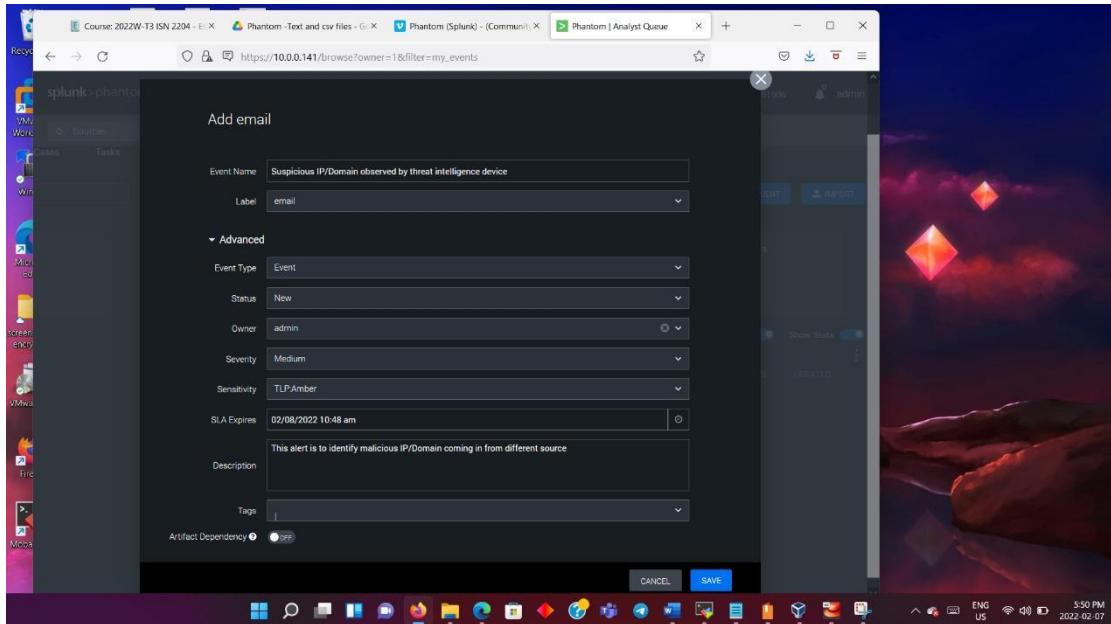
Select investigate and click on save and continue.



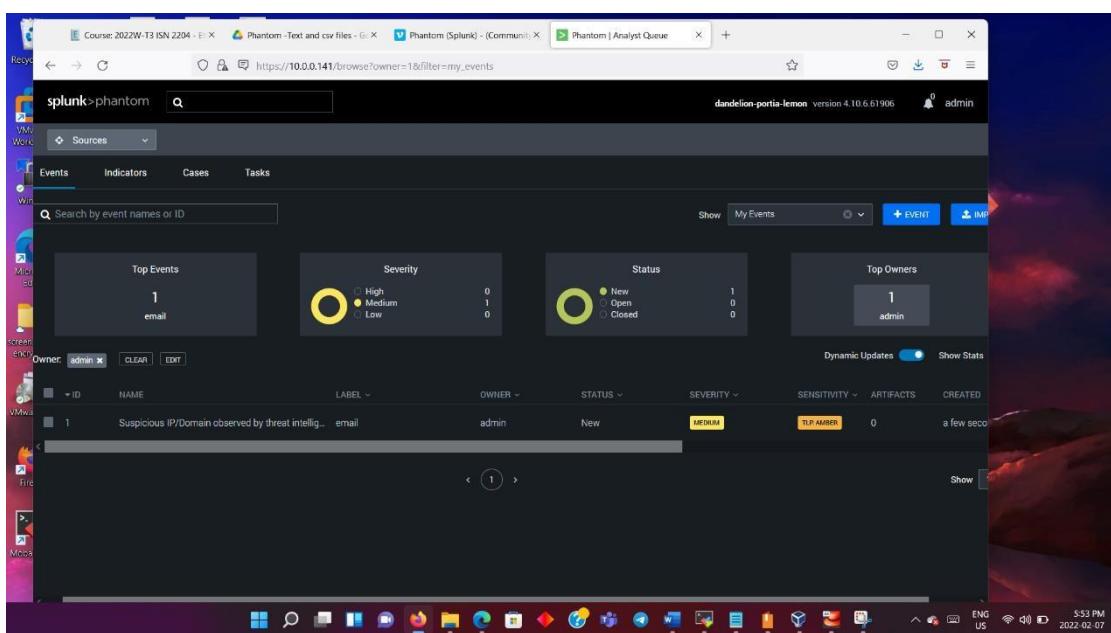
Click on skip remaining.



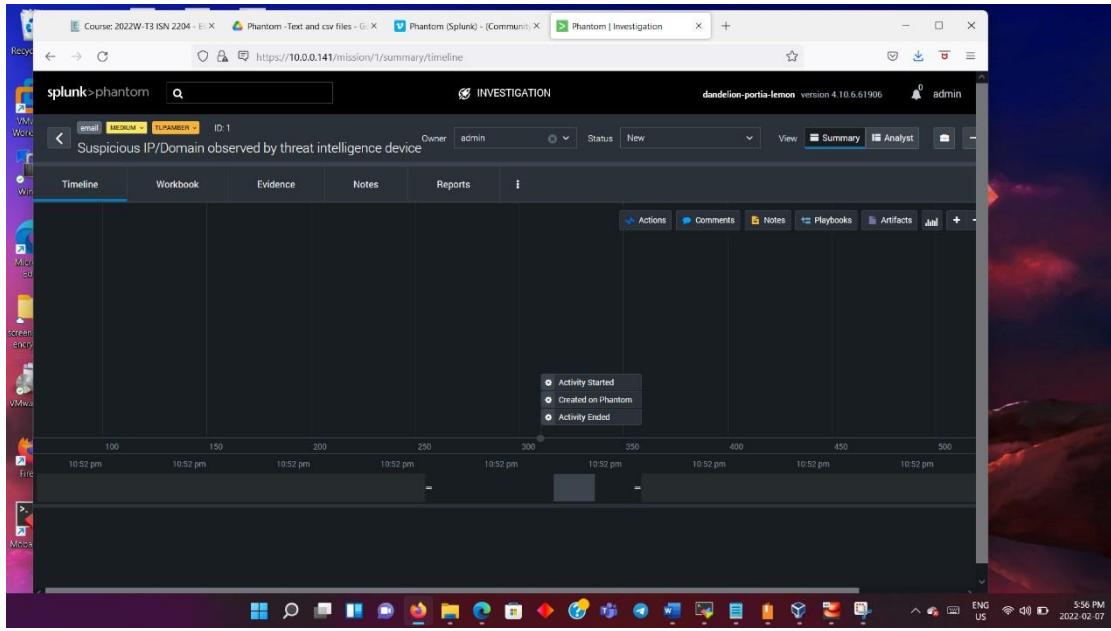
Click on source >my event



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

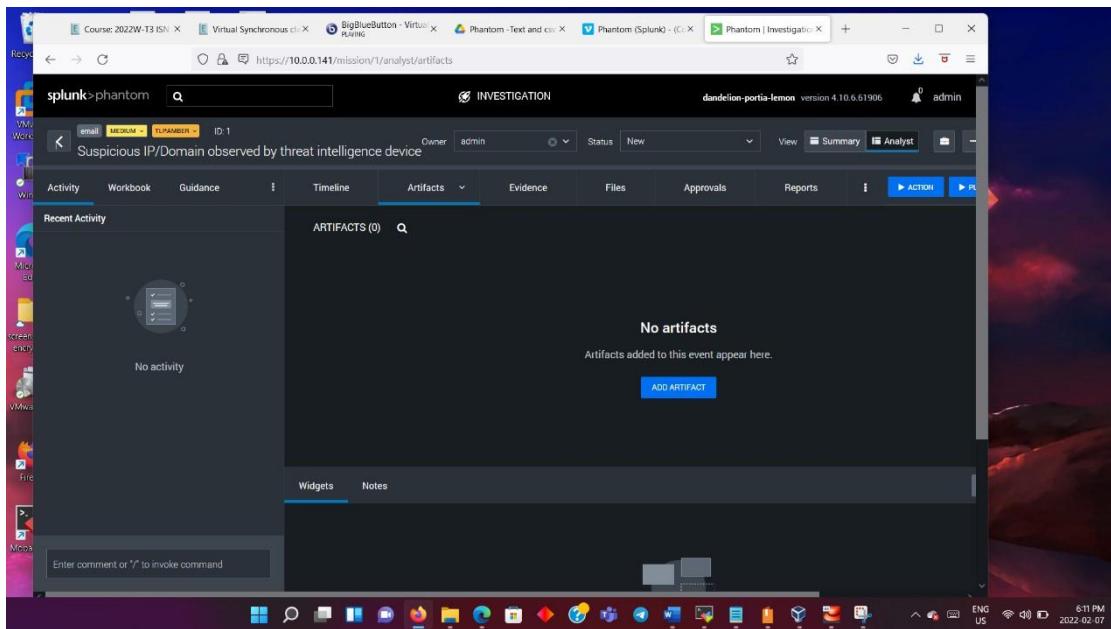


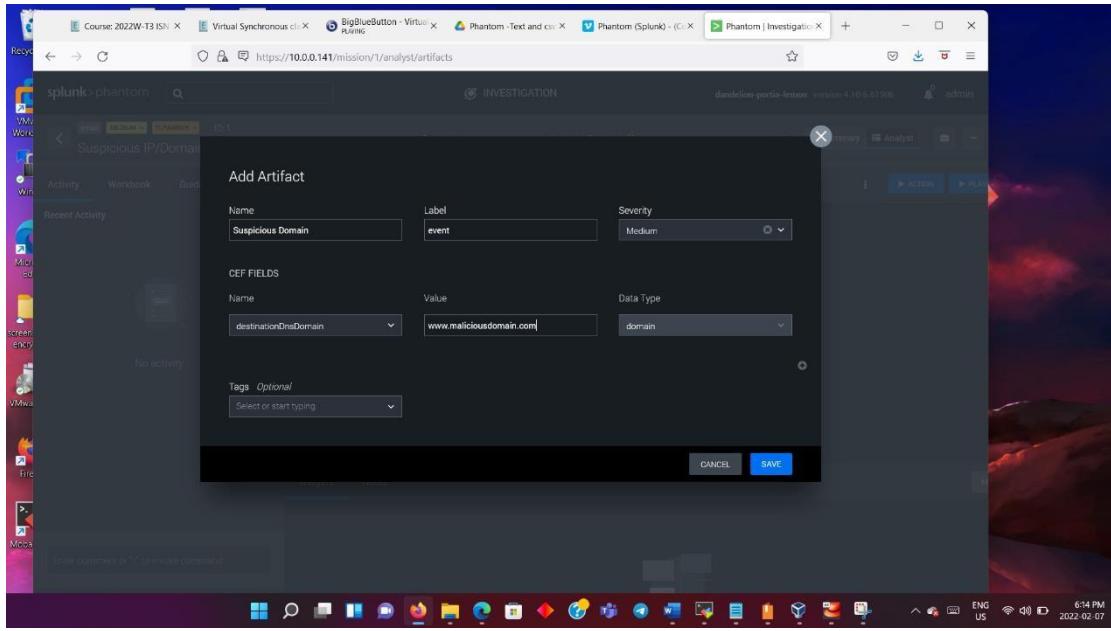
From the above image we are able to see the event created, severity provided, status etc.



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

The screenshot shows the Splunk Phantom Investigation interface. At the top, it displays "splunk>phantom" and "INVESTIGATION". The mission ID is 1, owned by "admin" and status is "Open". The main area shows "Suspicious IP/Domain observed by threat intelligence device". The "Artifacts" tab is selected, showing two artifacts: "Suspicious IP" and "Suspicious Domain", both created by "admin" with medium severity. On the left sidebar, under "MISSION GUIDANCE", there is no expert listed. Under "PLAYBOOKS", one playbook is completed. Under "ACTIONS", five actions are listed, with two completed: "lookup_ip google_dns", "lookup_domain google_dns", and "geolocate_ip maxmind". A "MAXMIND" map widget is present, indicating the IP 185.65.98.24 is located in France. The bottom status bar shows "1°C Mostly cloudy" and the date "10-02-2022".

This screenshot shows the same Splunk Phantom Investigation interface as the first one, but with a different focus. The "Widgets" tab is selected, displaying a "splunk>" search results table for the query "whois ip 185.65.98.24". The table has columns: IP, ASN, ASN CIDR, ASN COUNTRY CODE, ASN DATE, ASN REGISTRY, and ABUSE EF. It lists two entries for the IP 185.65.98.24, both from RIPE NCC on July 29, 2014. The rest of the interface is identical to the first screenshot.

Creating a custom list of malicious IP addresses

The screenshot shows the Splunk Phantom interface. In the top navigation bar, there are tabs for 'Playbooks', 'Custom Functions', and 'Custom Lists'. The 'Custom Lists' tab is selected. Below it, a sub-menu titled 'Custom Lists > Malicious IP address' is open. A table displays the following data:

	Malicious IP 1	Malicious IP 2
1	145.56.32.54	
2	78.59.65.41	
3		

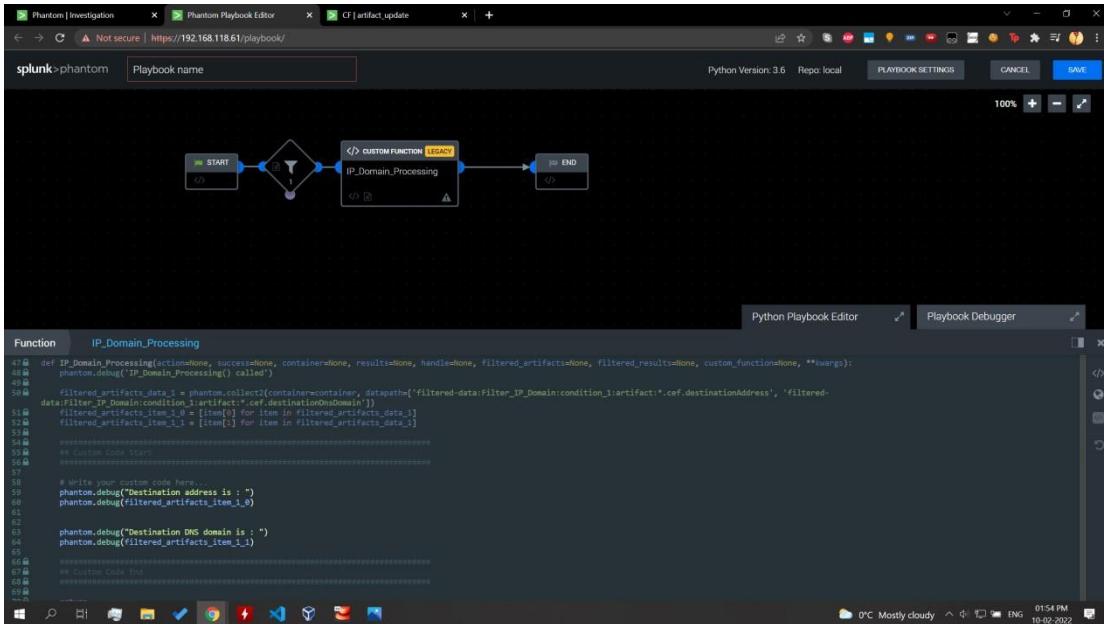
Create a new playbook in Splunk with a custom function

The screenshot shows the Phantom Playbook Editor. The main workspace displays a workflow diagram with the following steps:

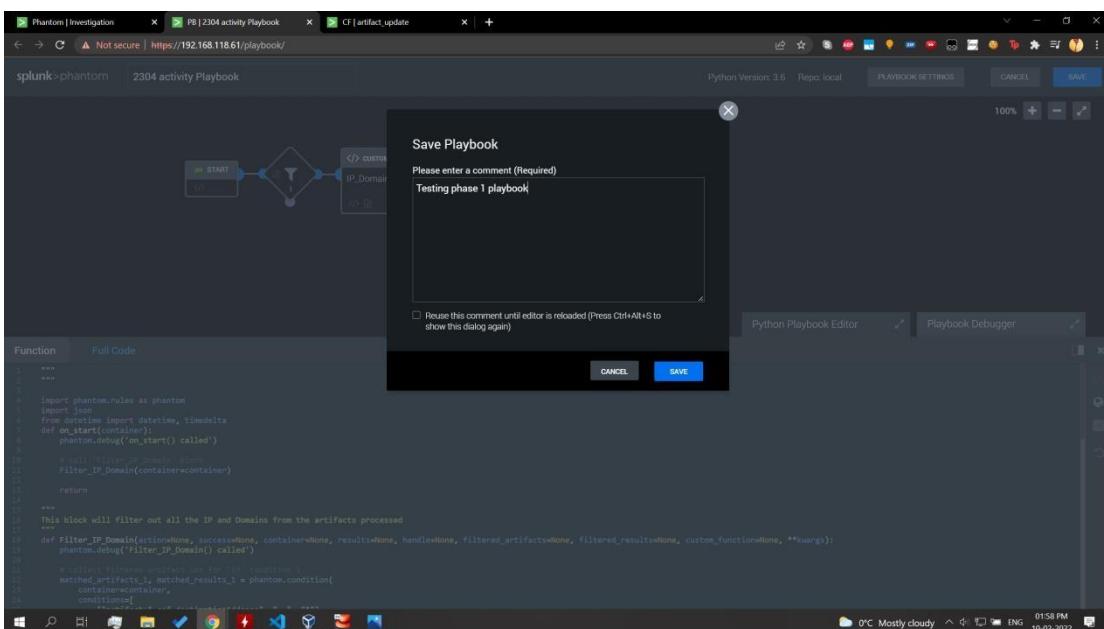
```
graph LR; START((START)) --> Decision{Decision}; Decision --> Function[custom function 1]; Function --> END((END))
```

The 'custom function 1' step is highlighted with a yellow background and labeled 'CUSTOM FUNCTION LEGACY'. The editor interface includes tabs for 'Python Playbook Editor' and 'Playbook Debugger' at the bottom.

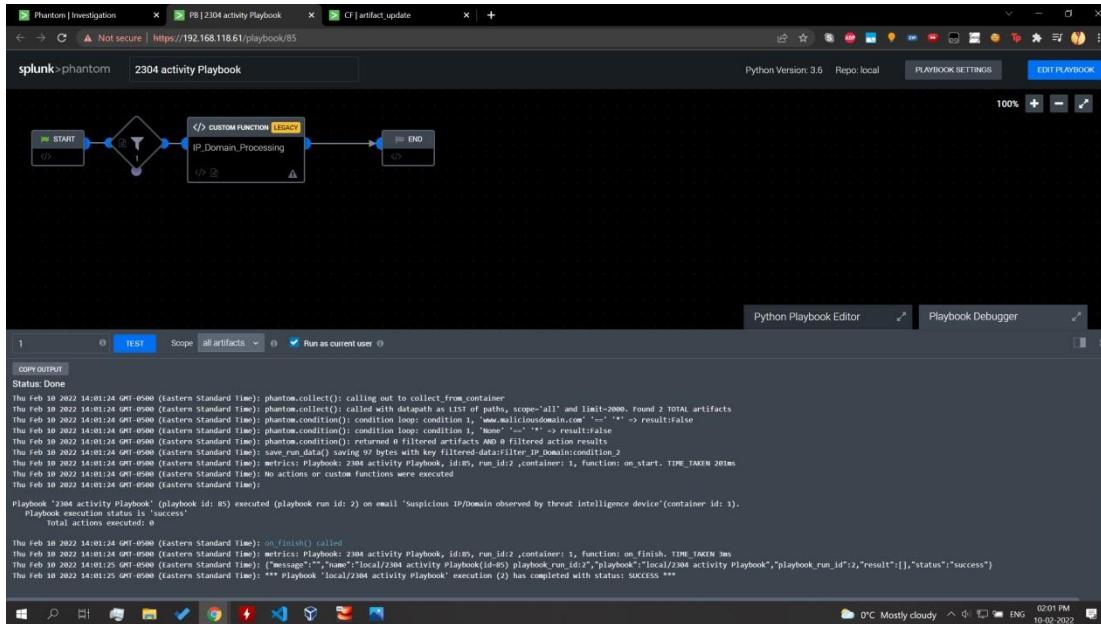
Modify the custom function to detect malicious IPs (as defined previously in artifacts)



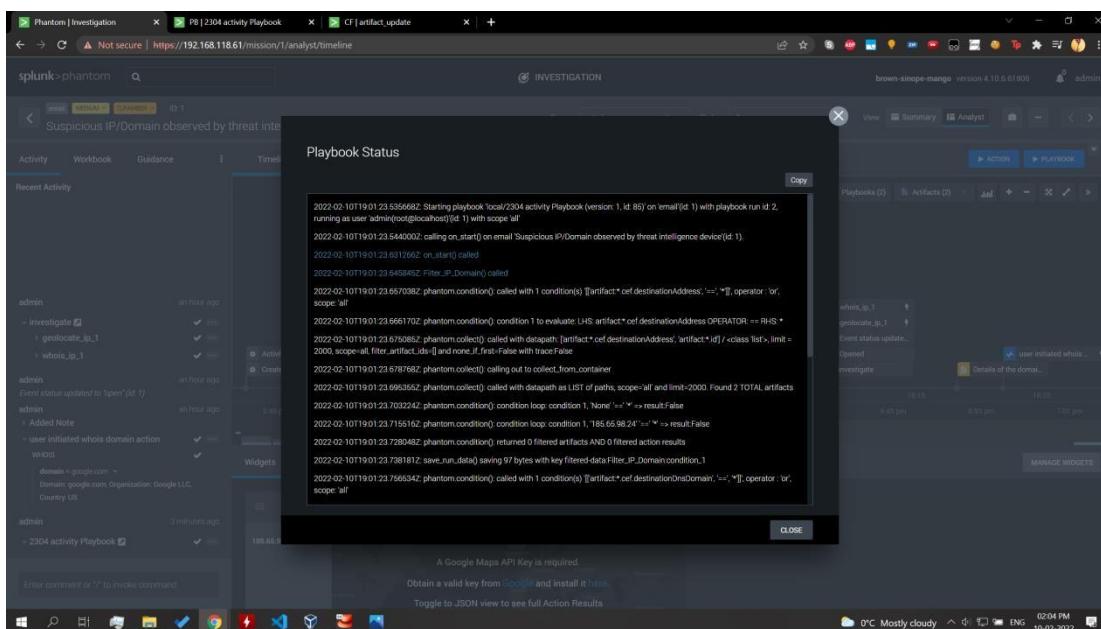
Save the playbook for testing later

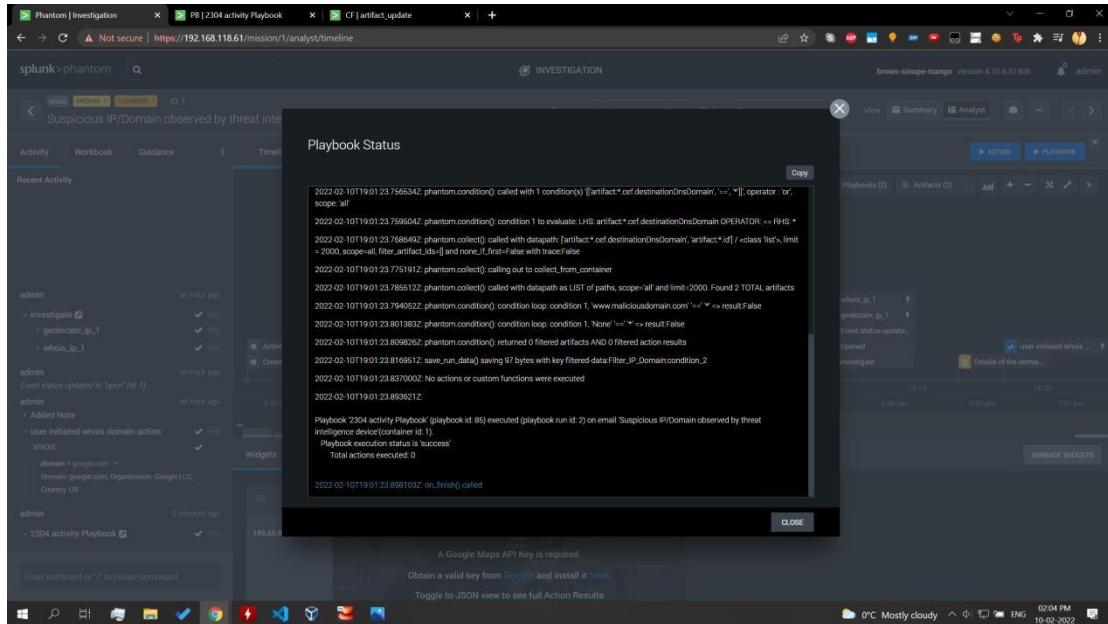


Test the playbook with event ID as 1



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





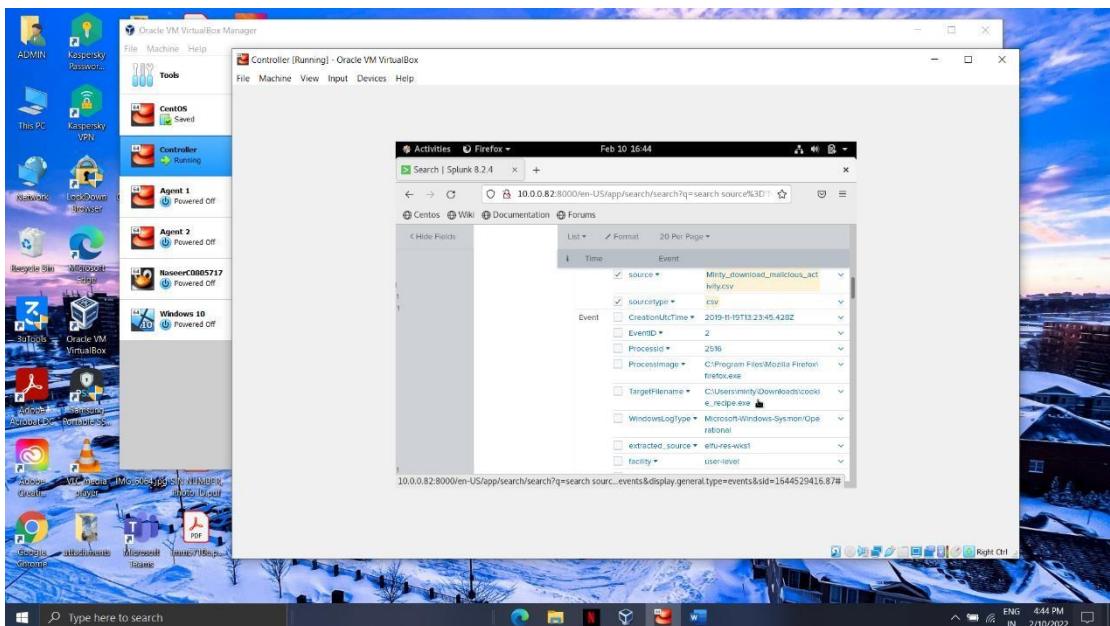
As it can be seen the 2 artifact conditions are listed in the debug log

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.

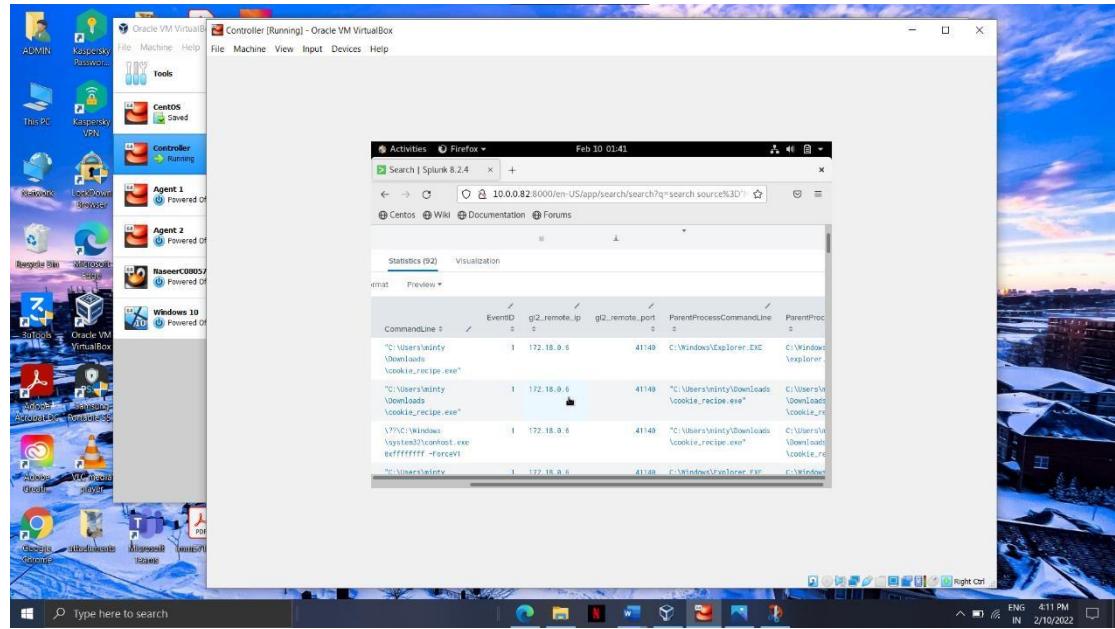
```
source="Minty_download_malicious_activity.csv" host="localhost.localdomain"
index="main" sourcetype="csv" firefox cookie
```



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

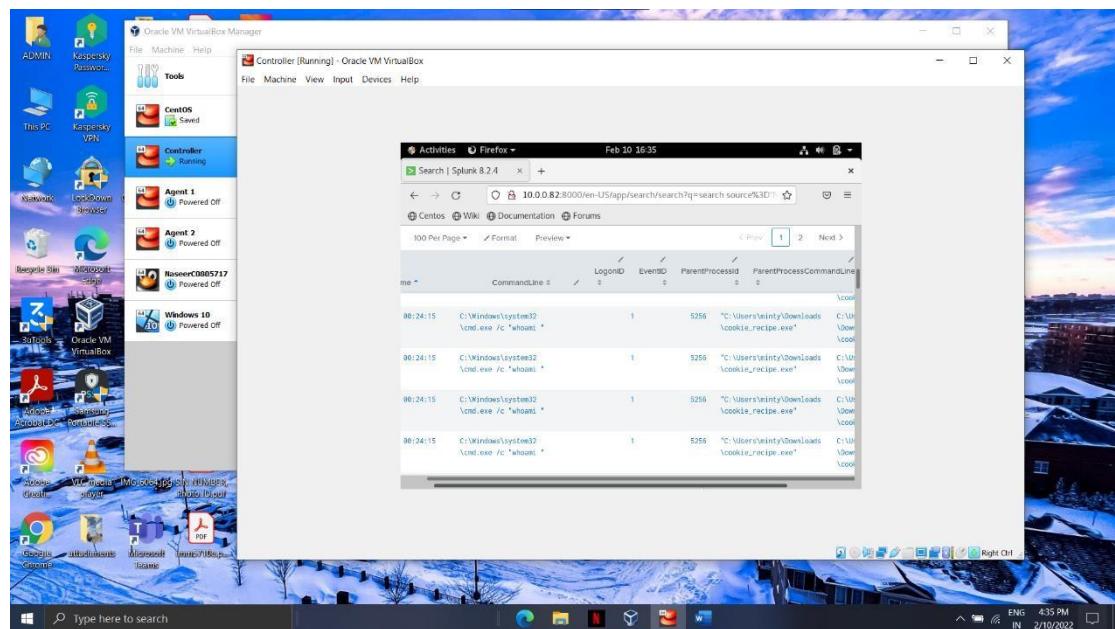
Question 2: The malicious file downloaded and executed by Minty gave the attacker remote access to his machine. What was the ip:port the malicious file connected to first?

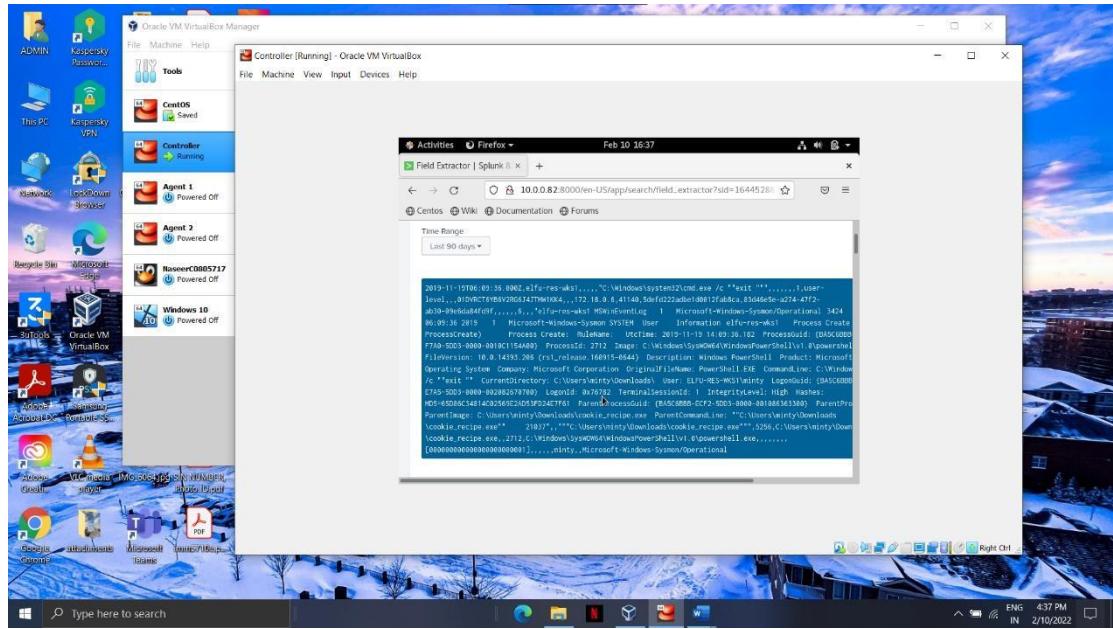
```
source="Minty_download_malicious_activity.csv" host="Minty"
index="main" sourcetype="csv" cookie_recipe.exe | table _time
CommandLine EventID gl2_remote_ip gl2_remote_port
ParentProcessCommandLine ParentProcessImage
```



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
```





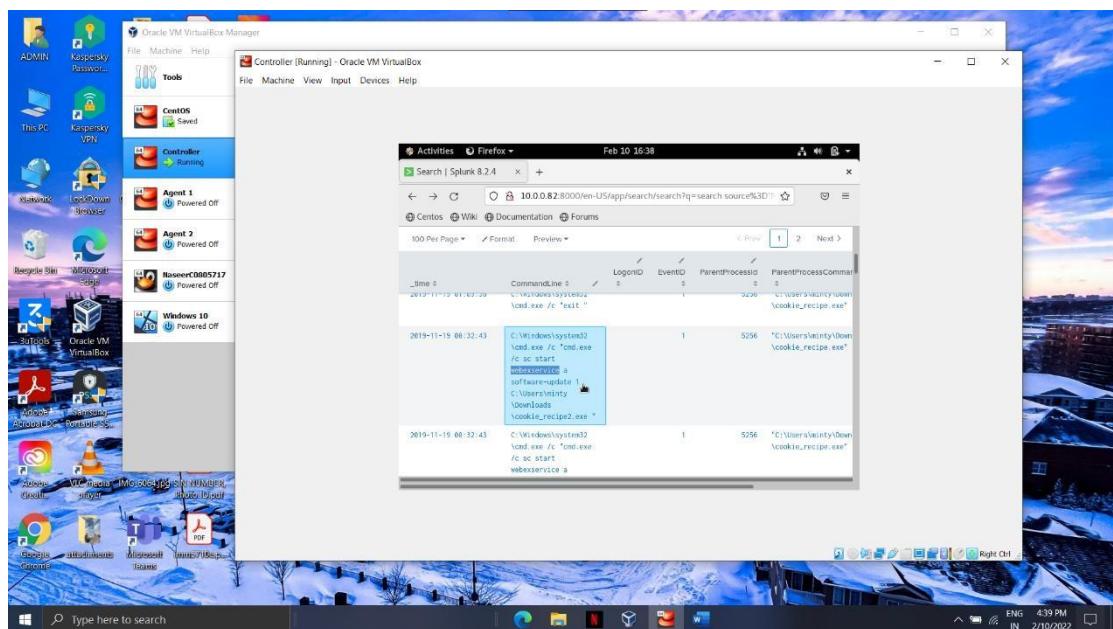
Answer: whoami

LogonId: 0x76782

ParentProcessId: 5256

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

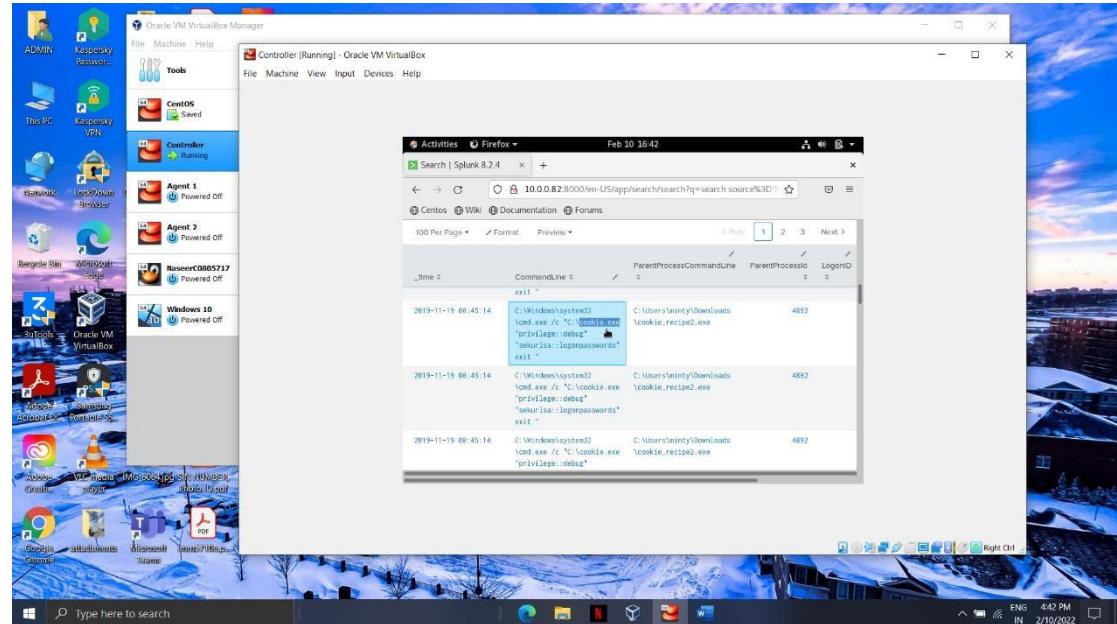
```
source="Minty_download_malicious_activity.csv" host="Minty"
index="main" sourcetype="csv" cookie_recipe.exe | table _time
CommandLine EventID ParentProcessCommandLine ParentProcessImage
```



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/1LT08ZvOLRKT2PVgHEBmXZwDJZbzTcwO?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:

Student name:

All members of the group have participated in this activity.