Cybersecurity

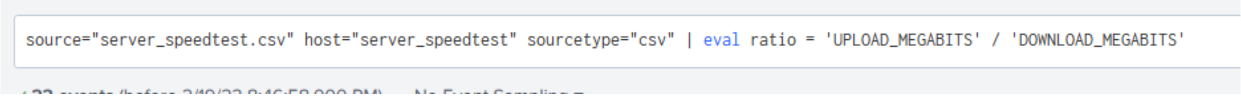
Module 19 Challenge Submission File

Let’s Go Splunking!

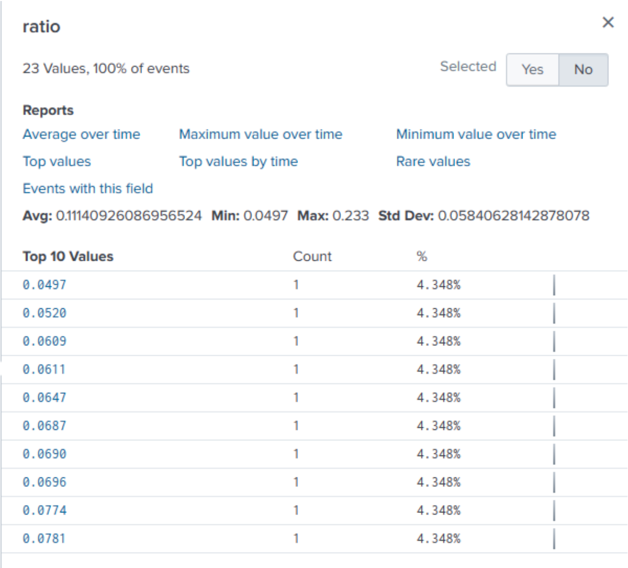
Make a copy of this document to work in, and then respond to each question below the prompt. Save and submit this completed file as your Challenge deliverable.

Step 1: The Need for Speed

2. Using the `eval` command, create a field called `ratio` that shows the ratio between the upload and download speeds.



```
source="server_speedtest.csv" host="server_speedtest" sourcetype="csv" | eval ratio = 'UPLOAD_MEGABITS' / 'DOWNLOAD_MEGABITS'
```



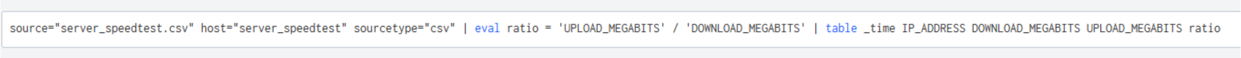
ratio

23 Values, 100% of events

Reports

Avg: 0.11140926086956524 **Min:** 0.0497 **Max:** 0.233 **Std Dev:** 0.05840628142878078

Top 10 Values	Count	%
0.0497	1	4.348%
0.0520	1	4.348%
0.0609	1	4.348%
0.0611	1	4.348%
0.0647	1	4.348%
0.0687	1	4.348%
0.0690	1	4.348%
0.0696	1	4.348%
0.0774	1	4.348%
0.0781	1	4.348%



```
source="server_speedtest.csv" host="server_speedtest" sourcetype="csv" | eval ratio = 'UPLOAD_MEGABITS' / 'DOWNLOAD_MEGABITS' | table _time IP_ADDRESS DOWNLOAD_MEGABITS UPLOAD_MEGABITS ratio
```

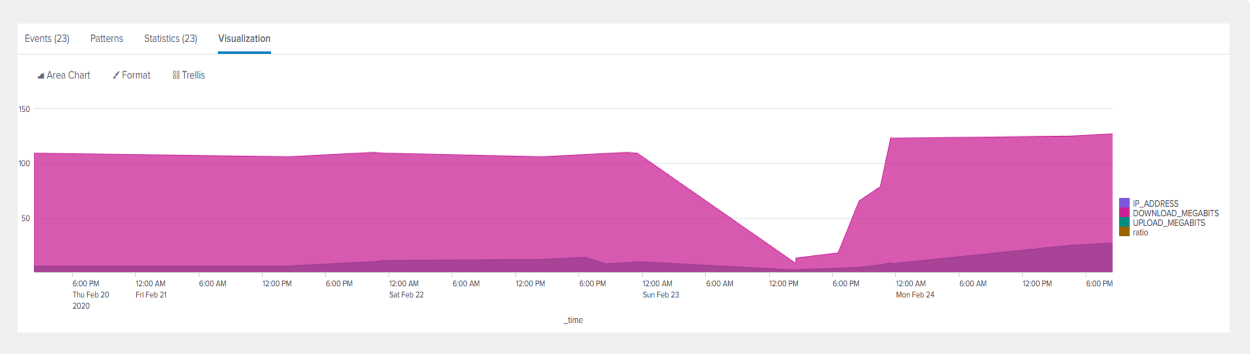
1. Based on the report you created, what is the approximate date and time of the attack?

```
02/23/2020 at 2:30 PM.
```

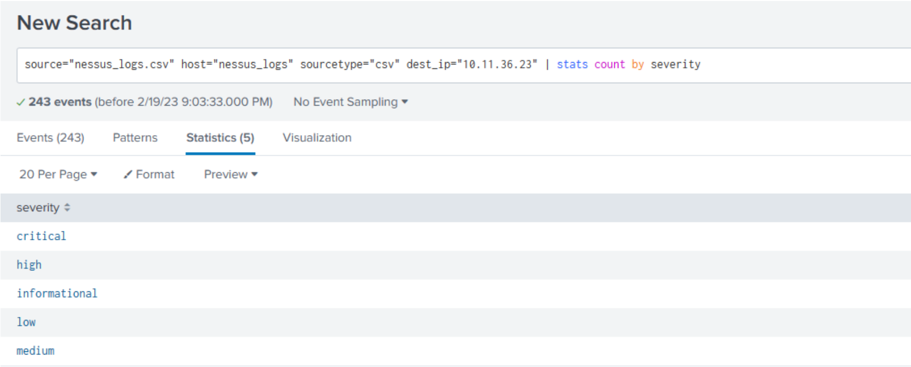
2. How long did it take your systems to recover?

```
11:30 PM or 9 hours until speeds recovered.
```

Provide a screenshot of your report:



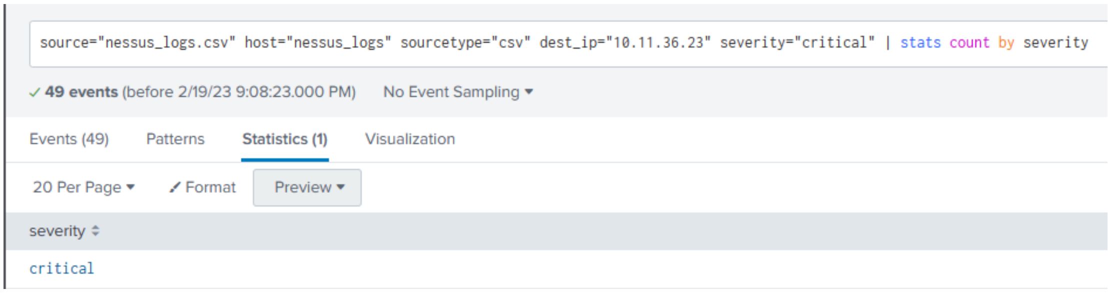
Step 2: Are We Vulnerable?



```
source="nessus_logs.csv" host="nessus_logs" sourcetype="csv" dest_ip="10.11.36.23" | stats count by severity
```

✓ **243 events** (before 2/19/23 9:03:33.000 PM)

severity
critical
high
informational
low
medium

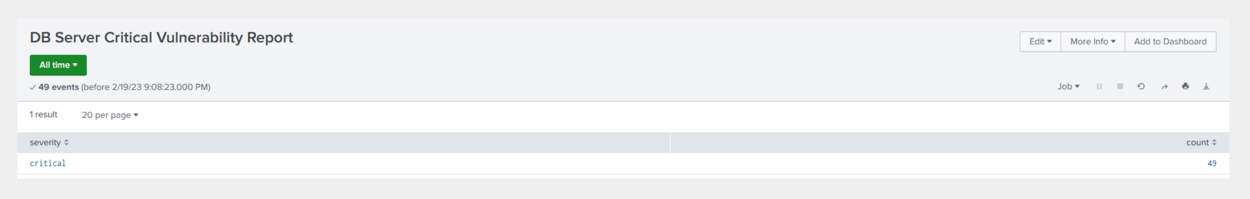


```
source="nessus_logs.csv" host="nessus_logs" sourcetype="csv" dest_ip="10.11.36.23" severity="critical" | stats count by severity
```

✓ **49 events** (before 2/19/23 9:08:23.000 PM)

severity
critical

Provide a screenshot of your report:



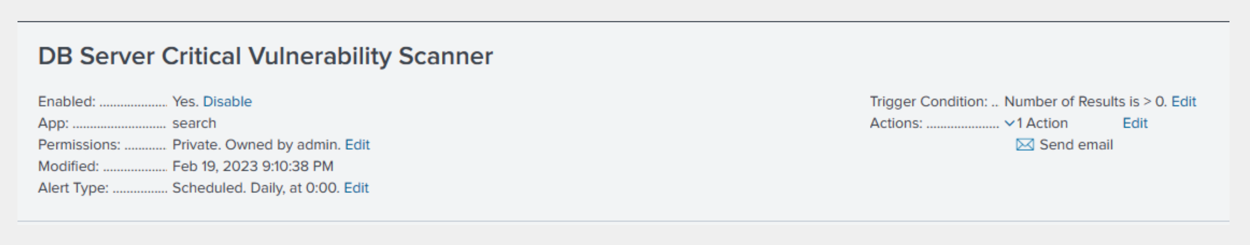
DB Server Critical Vulnerability Report

✓ **49 events** (before 2/19/23 9:08:23.000 PM)

1 result

severity ⇕	count ⇕
critical	49

Provide a screenshot showing that the alert has been created:

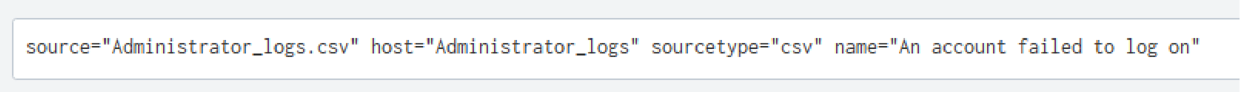


DB Server Critical Vulnerability Scanner

Enabled: .... Yes. Disable
App: .... search
Permissions: .... Private. Owned by admin. Edit
Modified: .... Feb 19, 2023 9:10:38 PM
Alert Type: .... Scheduled. Daily, at 0:00. Edit

Trigger Condition: .. Number of Results is > 0. Edit
Actions: .... ⌄1 Action Edit
✉ Send email

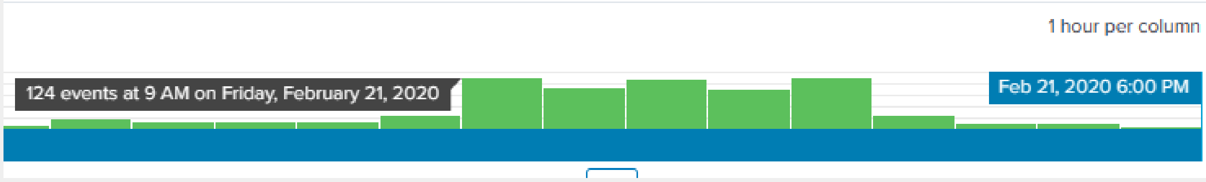
Step 3: Drawing the (Base)line



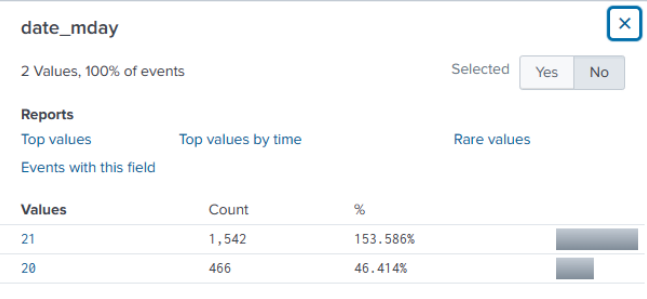
```
source="Administrator_logs.csv" host="Administrator_logs" sourcetype="csv" name="An account failed to log on"
```

1. When did the brute force attack occur?

```
02/21/2020 from 9AM - 1PM.
```



2. Determine a baseline of normal activity and a threshold that would alert if a brute force attack is occurring:



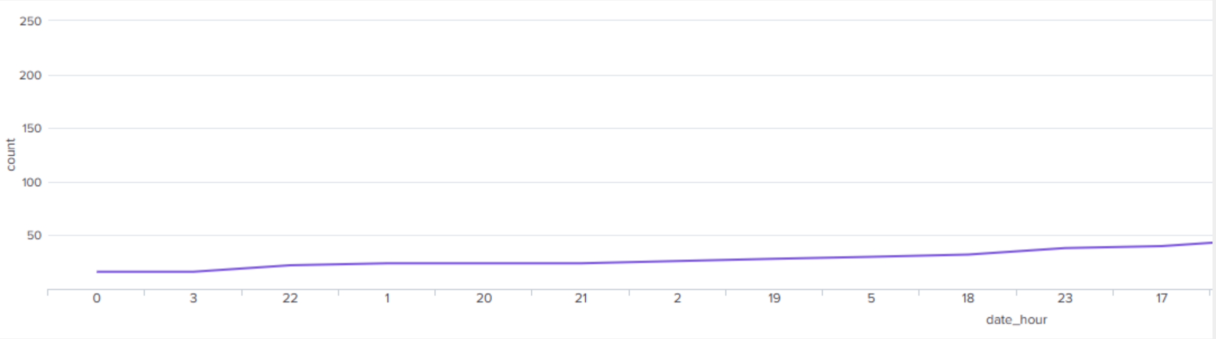
date_mday

2 Values, 100% of events

Reports

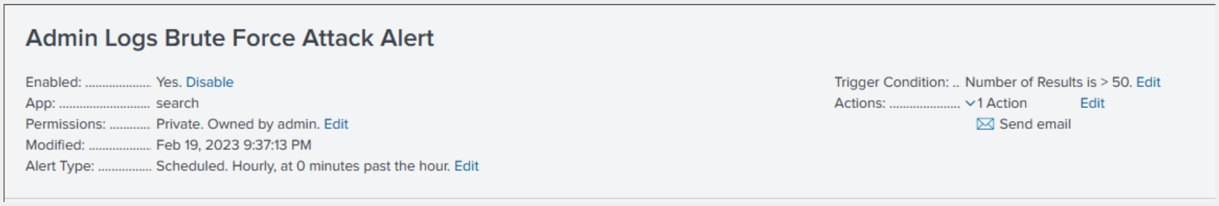
Top values Top values by time Rare values
Events with this field

Values	Count	%
21	1,542	153.586%
20	466	46.414%



Using this data, we can determine that less than 50 an hour or less than 500 a day would work as a baseline. Since the response needs to be swift, we will use the hourly baseline to set up our alert.

3. Provide a screenshot showing that the alert has been created:



Admin Logs Brute Force Attack Alert

Enabled: .... Yes. Disable
App: .... search
Permissions: .... Private. Owned by admin. Edit
Modified: .... Feb 19, 2023 9:37:13 PM
Alert Type: .... Scheduled. Hourly, at 0 minutes past the hour. Edit

Trigger Condition: .. Number of Results is > 50. Edit
Actions: .... ⌄1 Action Edit
✉ Send email