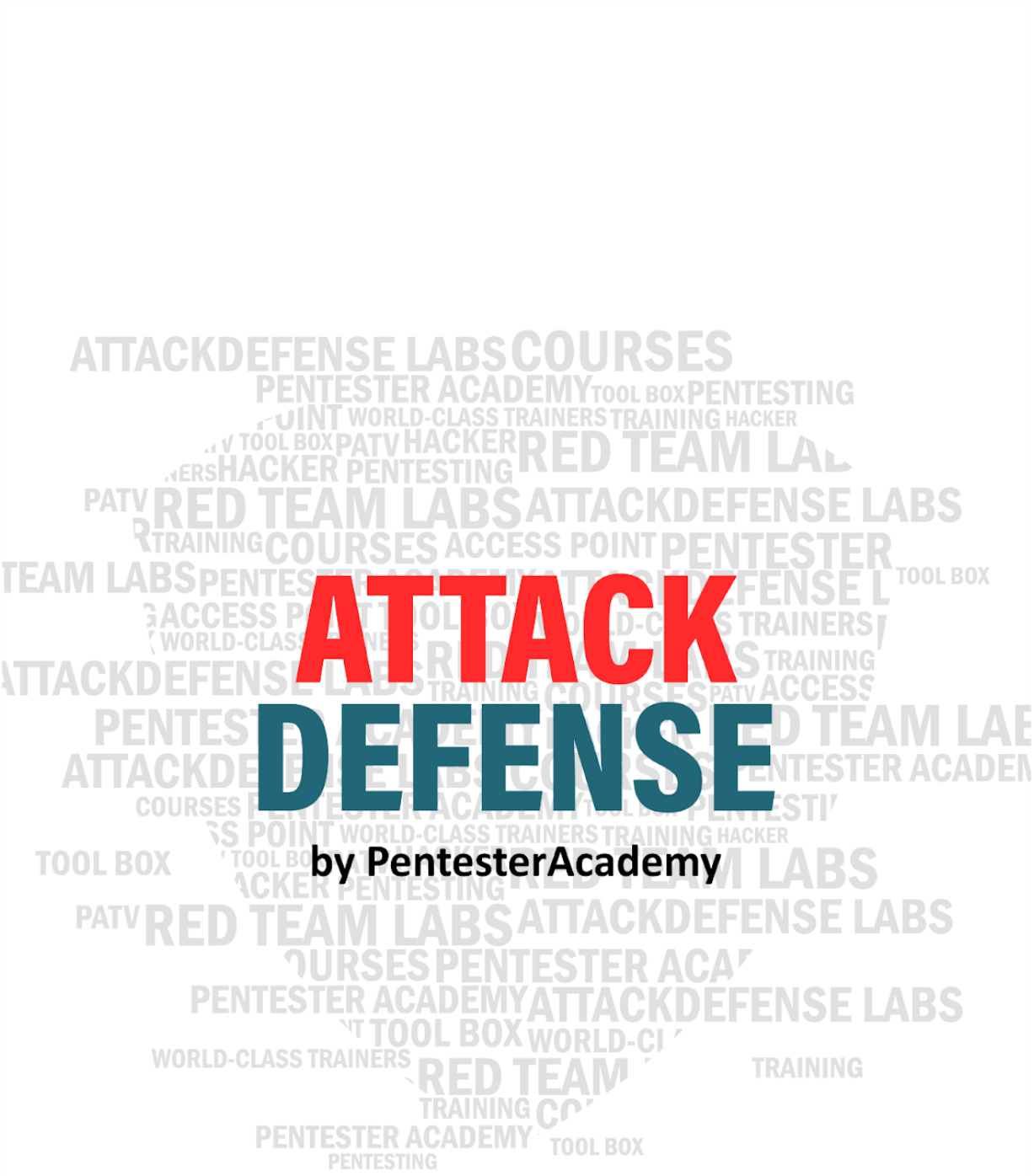

ATTACK
DEFENSE
by PentesterAcademy

Name	Kibana: Apache Log Analysis
URL	https://attackdefense.com/challengedetails?cid=1181
Type	Log Analysis : Webserver Logs

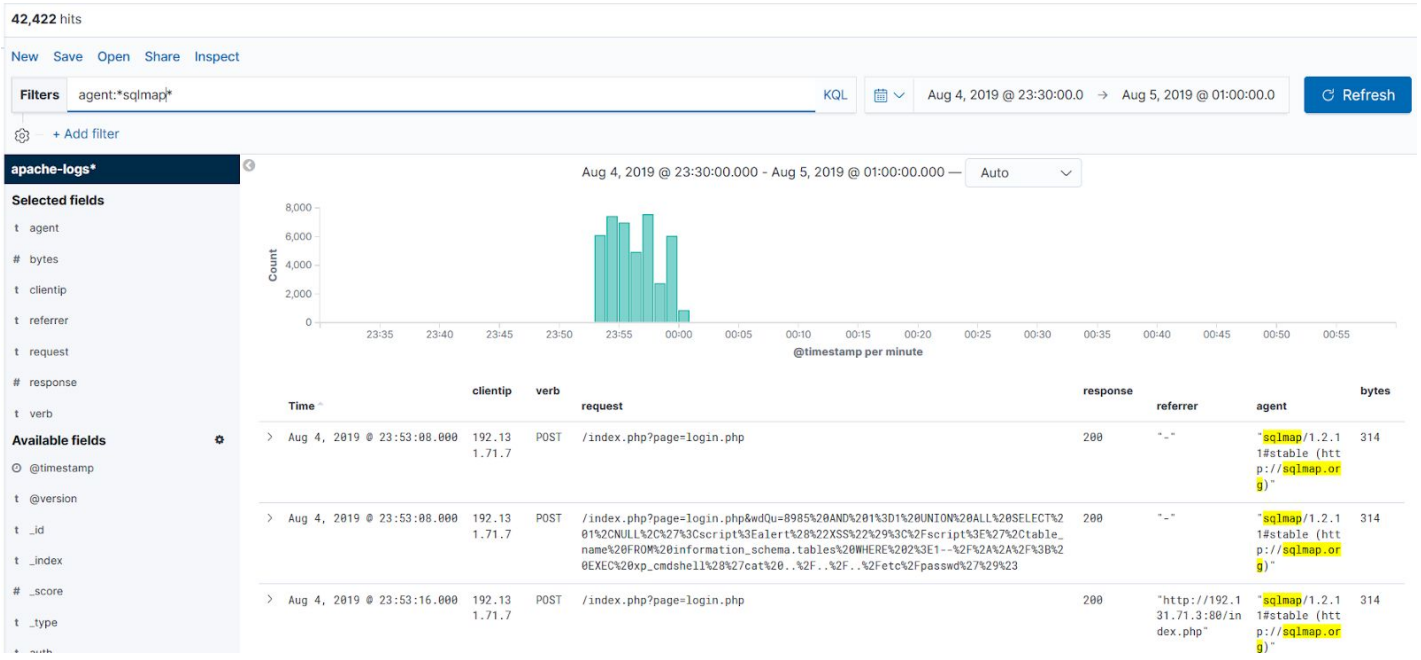
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Q1. SQLmap is used to perform SQL injection attack against a particular webpage. Find the path of the vulnerable webpage.

Answer: /index.php

Solution:

Apply filter: agent:*sqlmap*

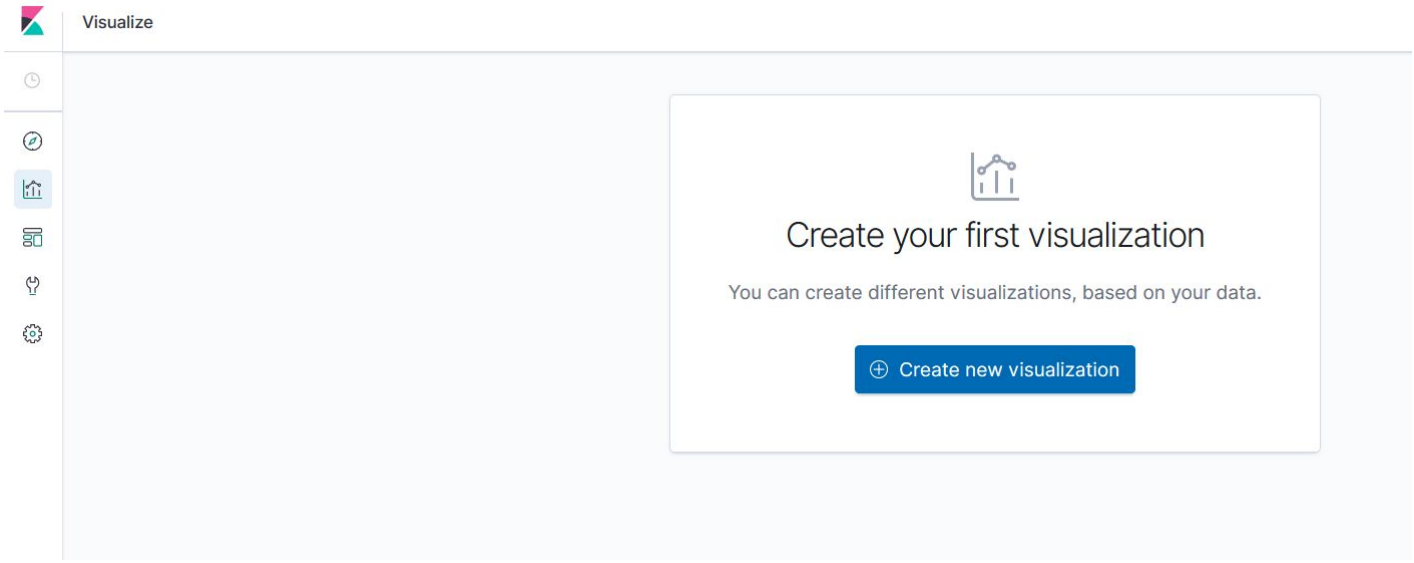


SQLmap is used to perform sql injection attack on the web page “/index.php”

There are a total of 42,422 hits.

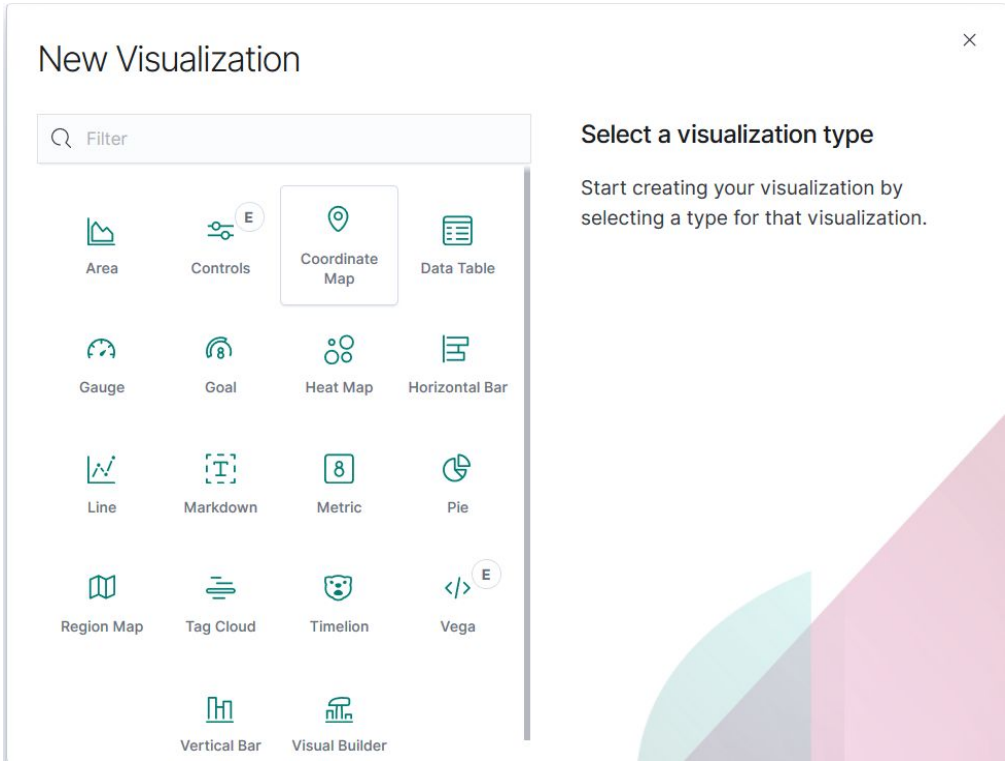
Create a Pie visualization and check whether all hits are made on “/index.php” web page.

Click on the Visualize button on the left panel.



Click on the “Create new visualization button”.

Select Pie.



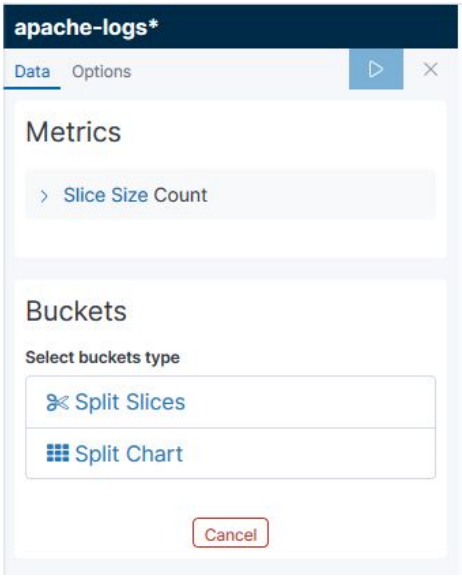
Select apache-logs.

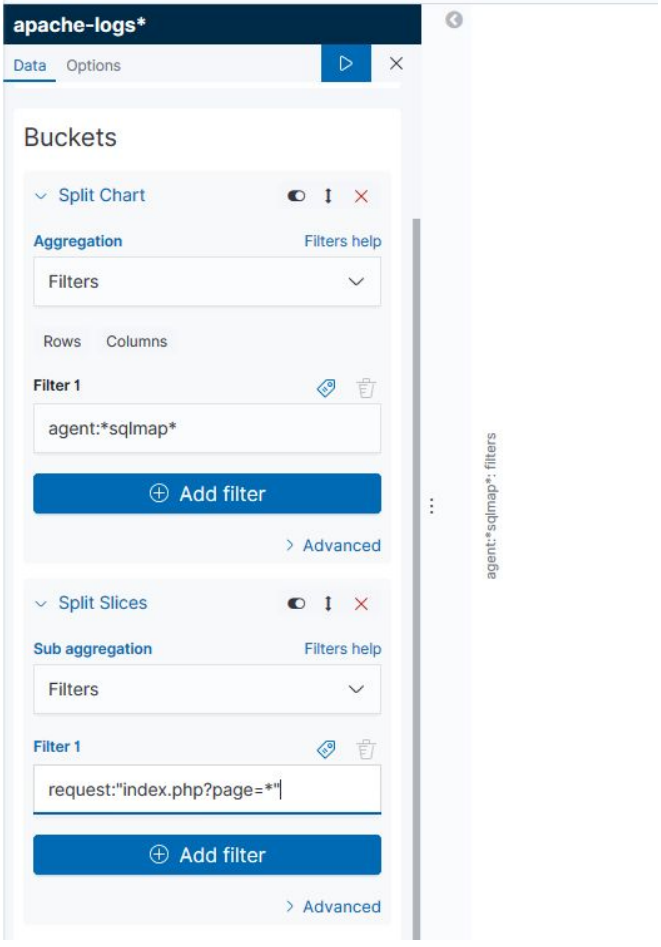
On the side panel click on Split Chart

Apply the filter: agent:*sqlmap*

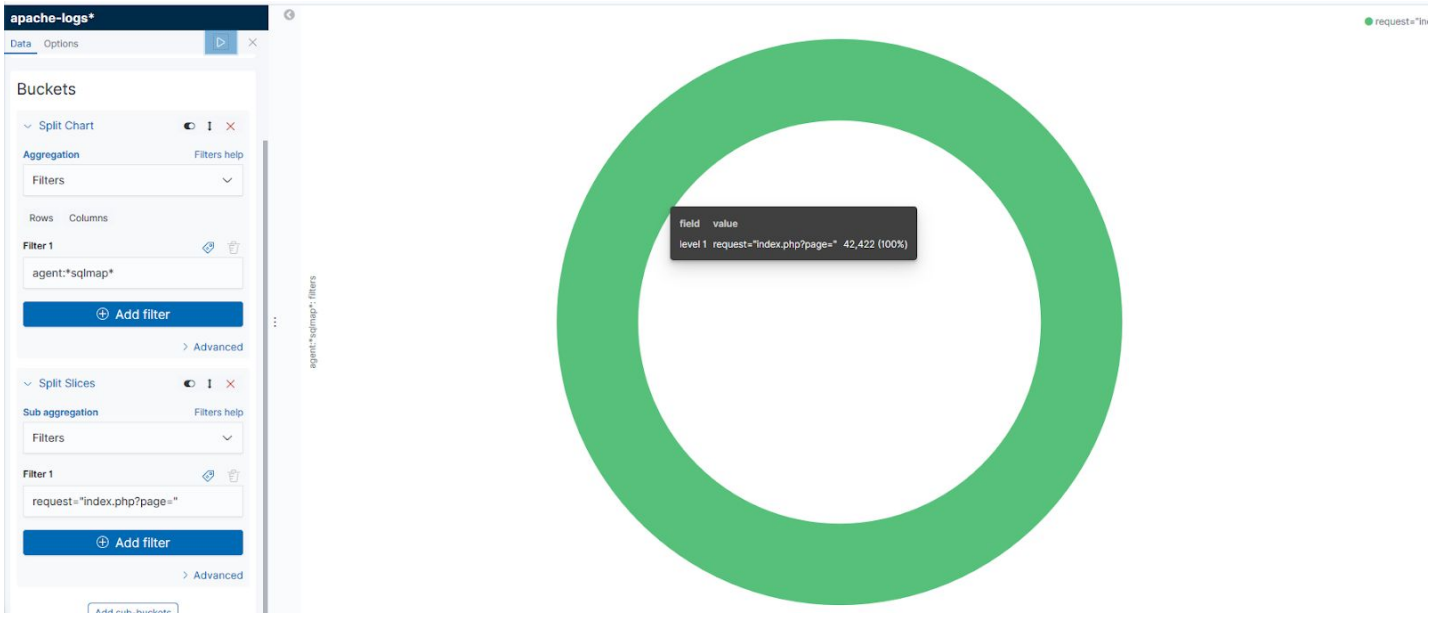
Click on Split Slices

Apply the filter: request:"index.php?page=*"


apache-logs*
Data
Options
Metrics
Slice Size Count
Buckets
Select buckets type
Split Slices
Split Chart
Cancel


apache-logs*
Data
Options
Buckets
Split Chart
Aggregation
Filters help
Filters
Rows
Columns
Filter 1
agent:*sqlmap*
Add filter
Advanced
Split Slices
Sub aggregation
Filters help
Filters
Filter 1
request:"index.php?page=*"
Add filter
Advanced
agent:*sqlmap*: filters

Click on the Run button on the top corner of the left panel



Verify the number of matched requests. There were a total of 42,422 request made to “index.php” webpage page which had user agent “sqlmap”.

Since the number of hits in discover and visualize are the same. It can be concluded, that all requests sent by sqlmap were on the web page “/index.php”.

Q2. Find the IP address of the machines which are performing directory enumeration attack on the web application. Which client performed the highest number of attempts?

Answer:

192.131.71.2
192.131.71.6
192.131.71.5

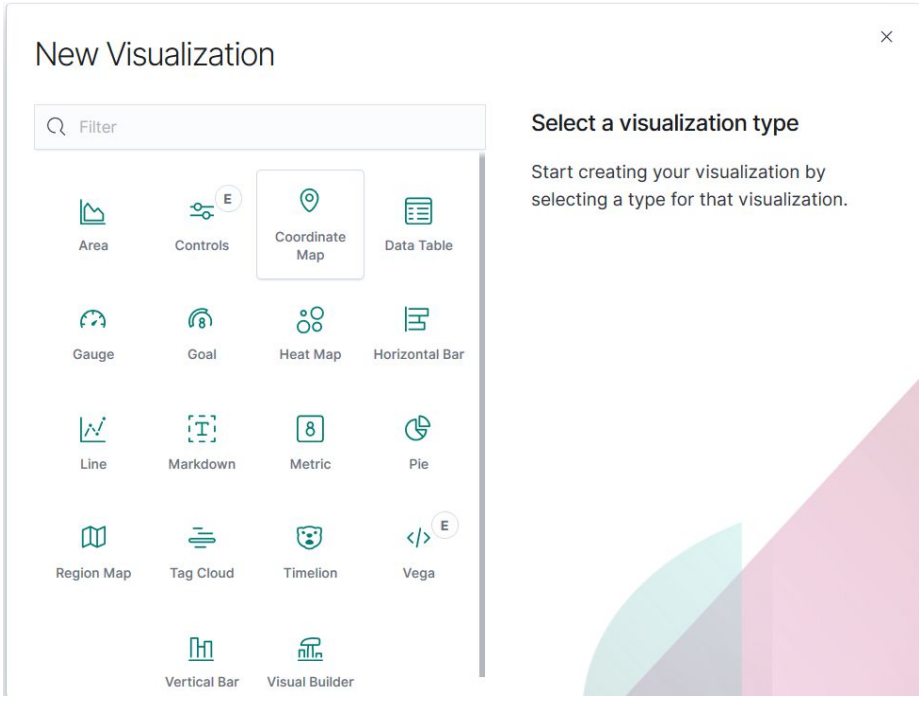
Client 192.131.71.6 performed highest number of attempts.

Solution:

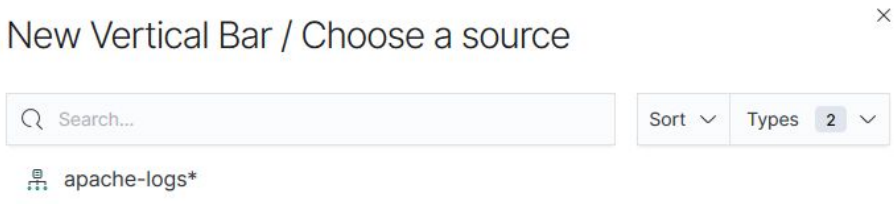
Since directory enumeration is attempted. Many 404 responses would have been logged.

Create a visualization with number of 404 responses against the Client IP address.

From the new visualization menu. Select Vertical bar.



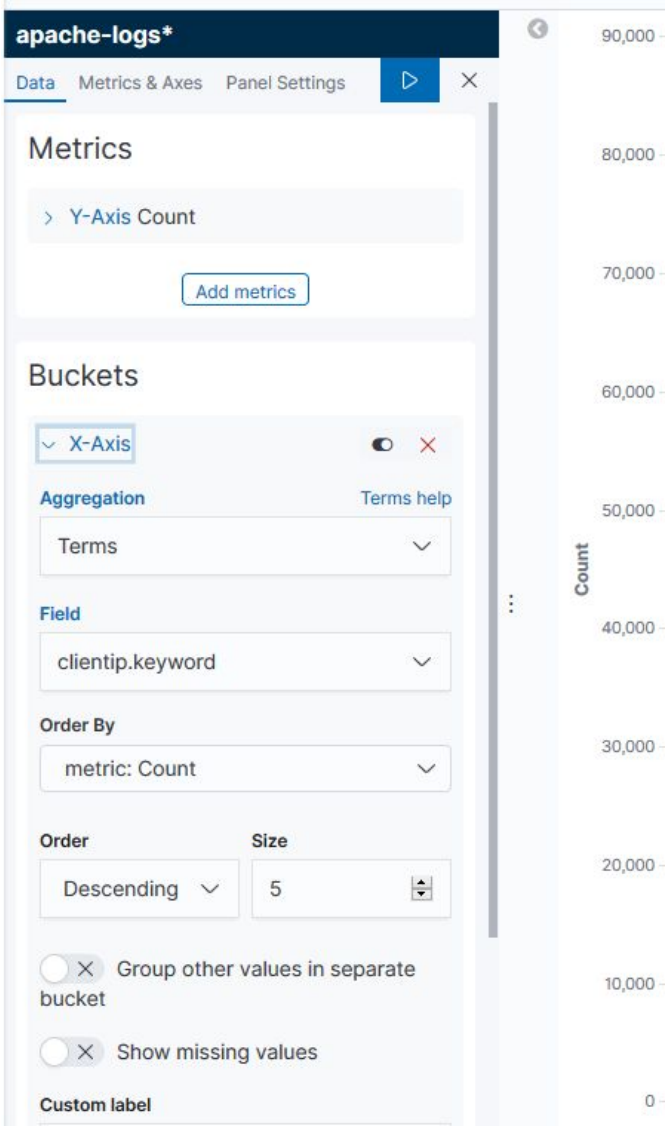
Select apache-logs as the source.



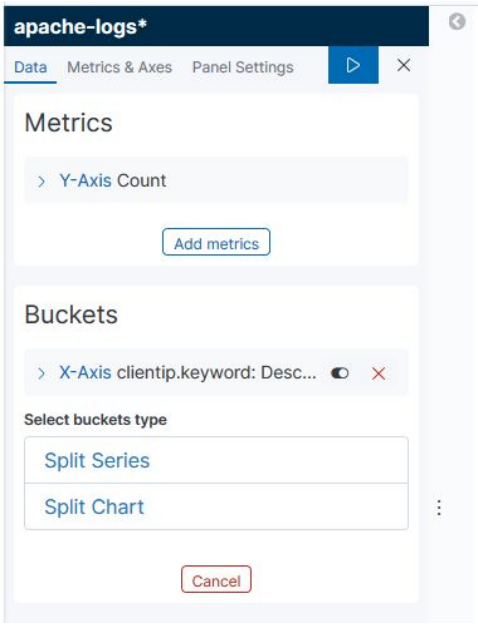
On the left panel. Select X-Axis Bucket

Select the following:

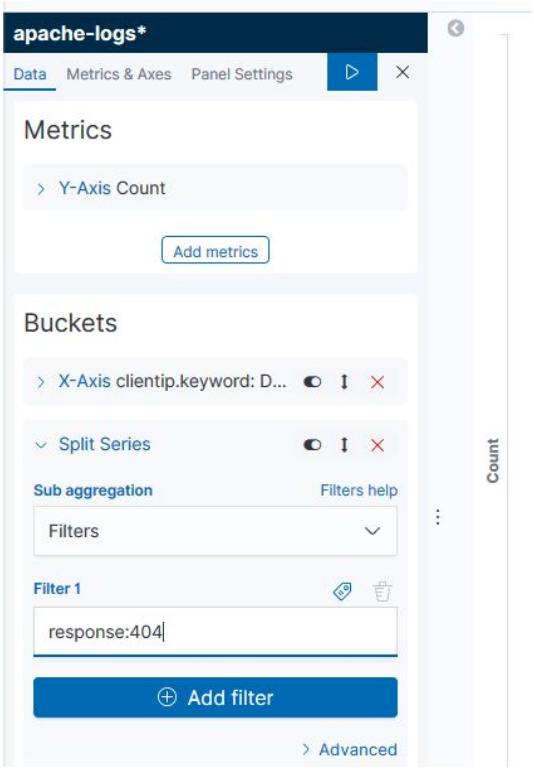
Aggregation: Terms
Field: clientip.keyword



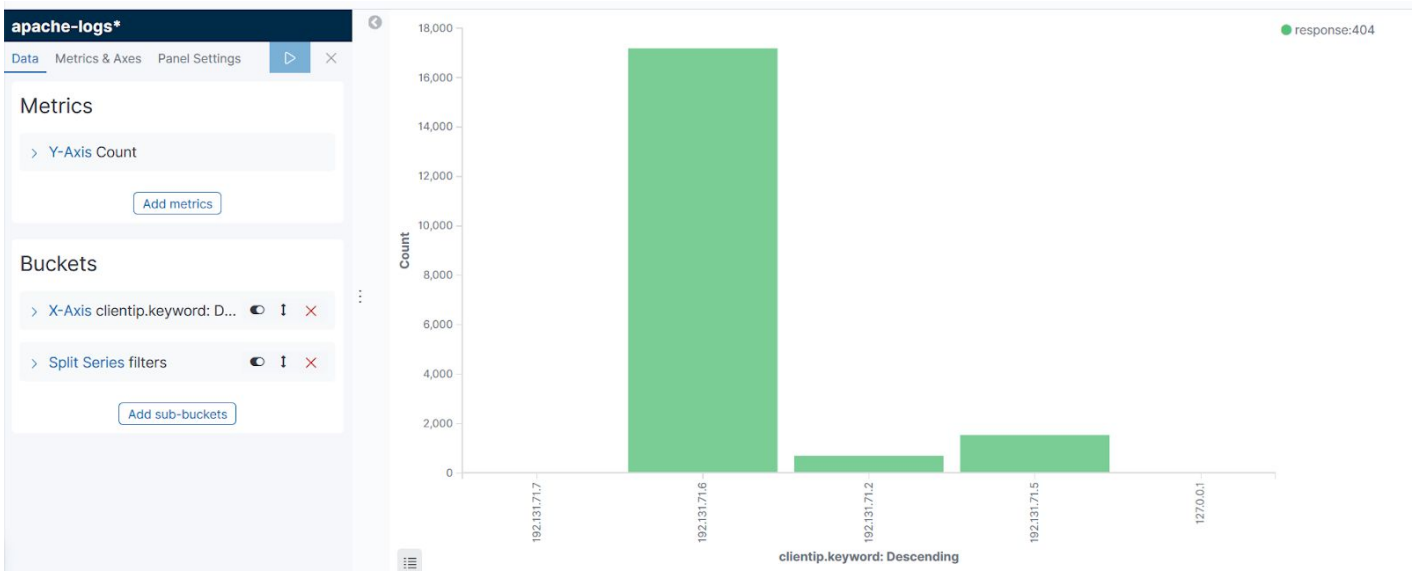
Add one more bucket by clicking on “Split Series” option.



Select “Filters” in sub-aggregation and enter “response:404” as filter.



Click on the Run button on the top corner of the left panel

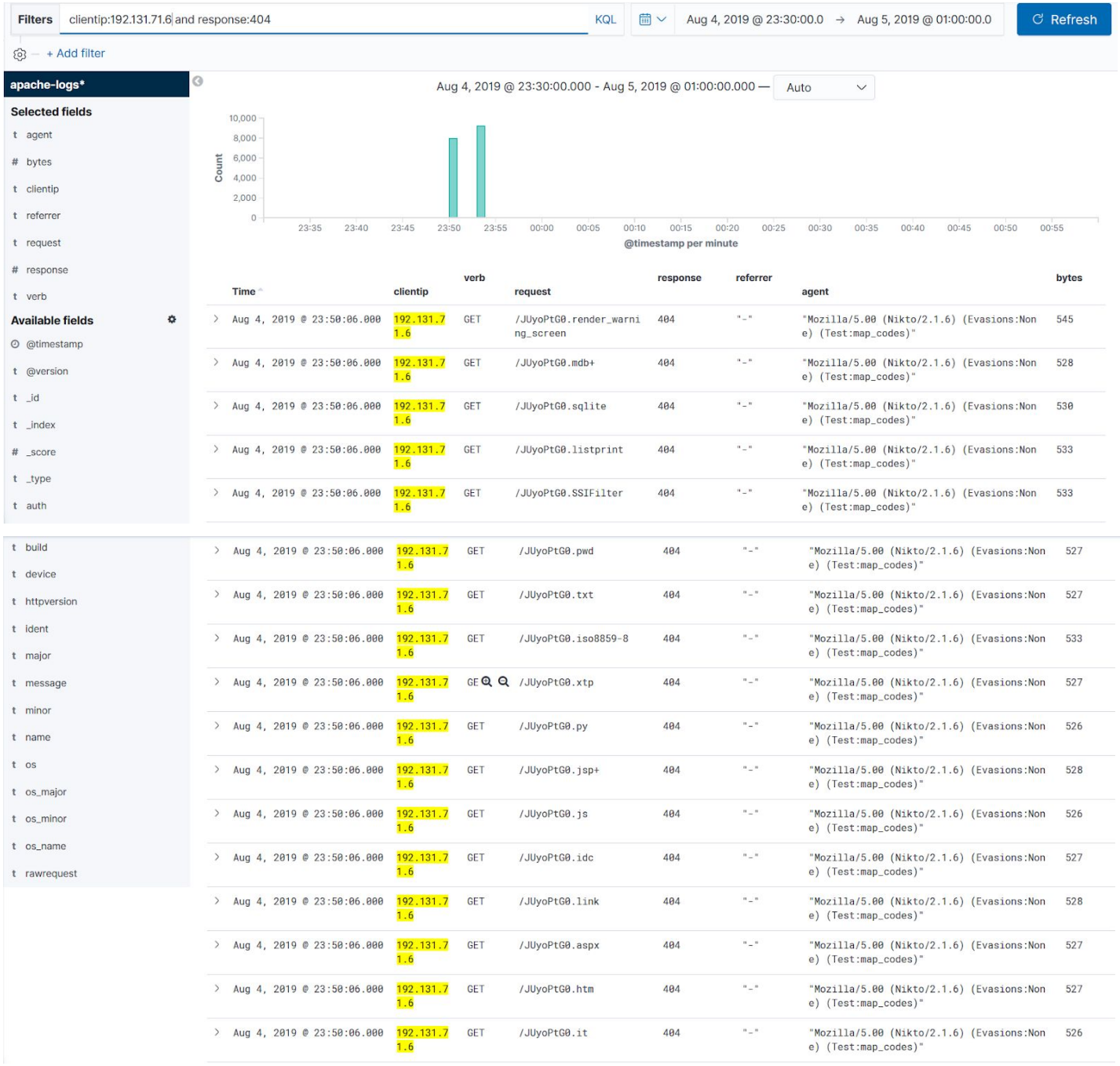


The clients 192.131.71.6, 192.131.71.2 and 192.131.71.5 had made many requests which resulted in 404 responses

To verify that directory enumeration attack was performed. Check the time difference between consecutive request. If the time difference is very less it can be concluded that directory enumeration attack was attempted.

Navigate to discover and apply the filter for client “192.131.71.6” and 404 response.

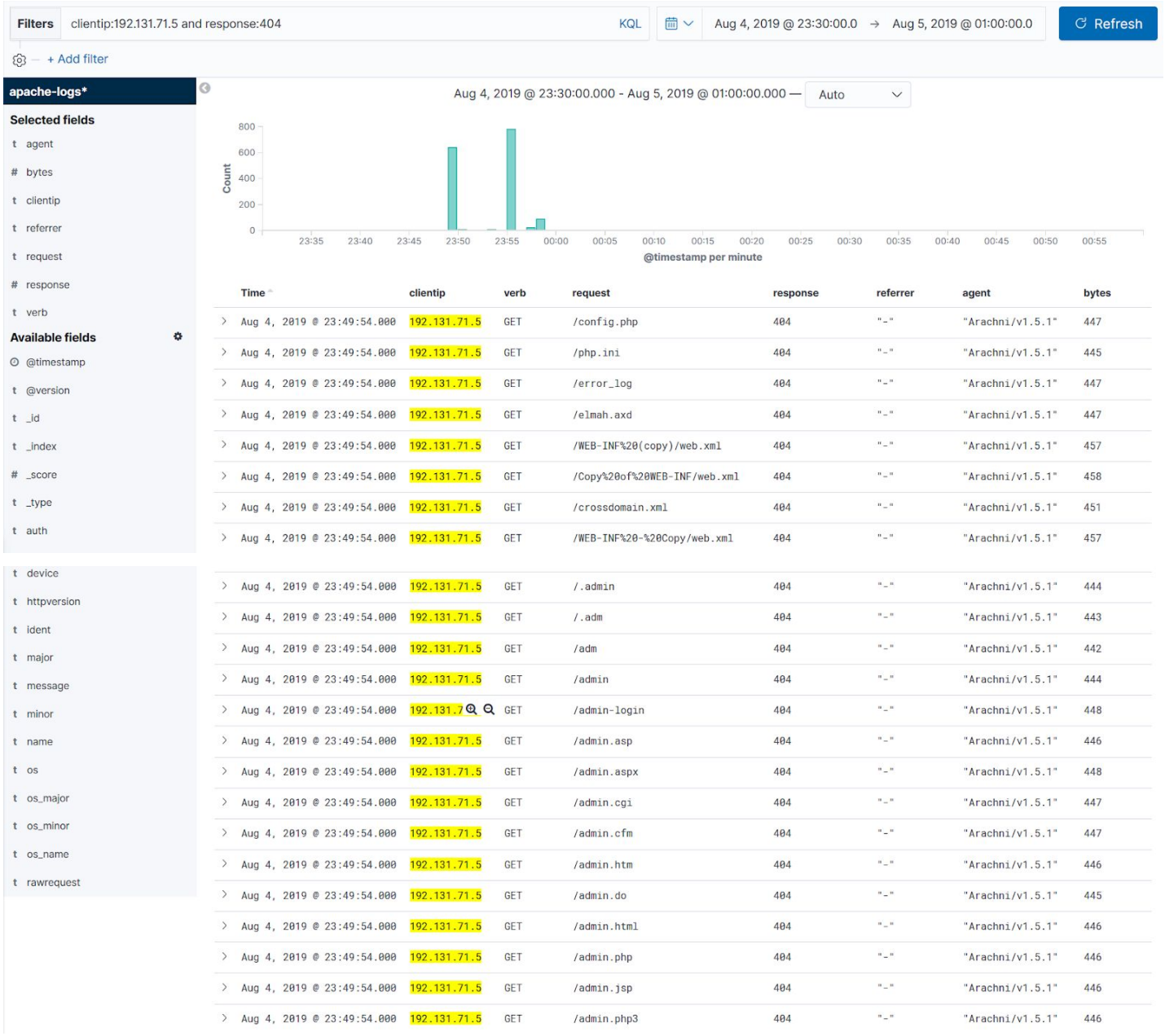
Filter: clientip:192.131.71.6 and response:404



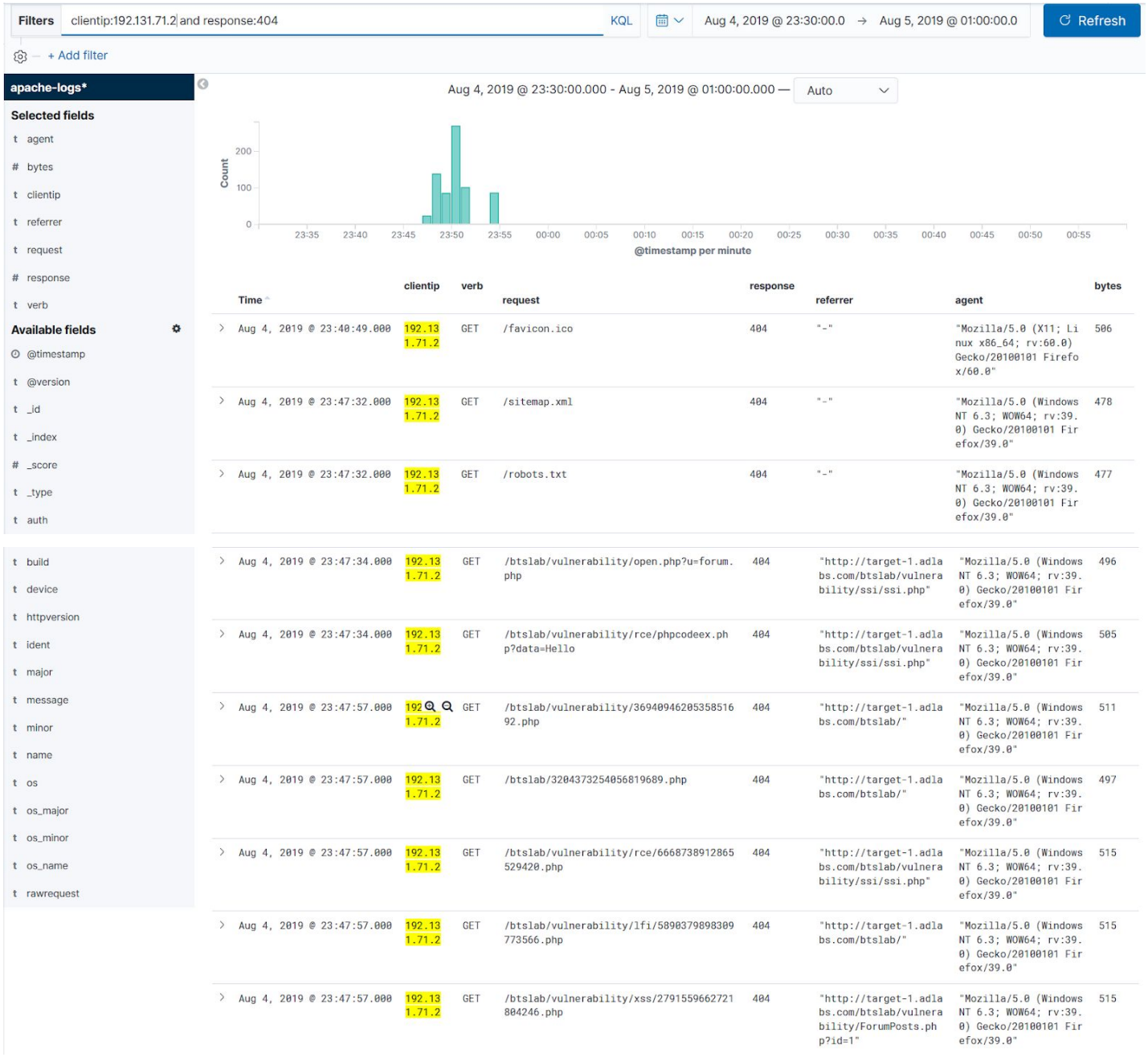
Many requests are sent within one second.

Similarly for other clients:

Filter: clientip:192.131.71.5 and response:404



Filter: clientip:192.131.71.2 and response:404



Q3. How many Path Traversal attacks were attempted to include the content of "/etc/shadow" file?

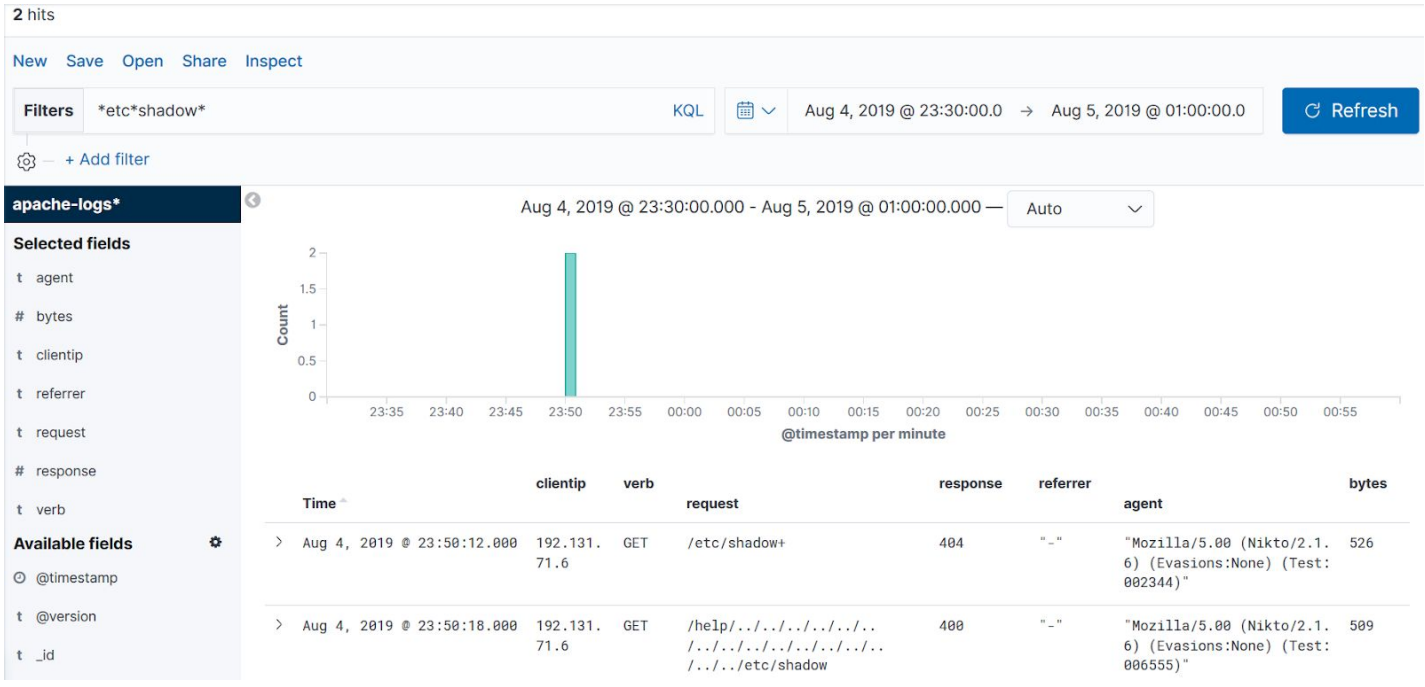
Answer: 2

Solution:

To include the file /etc/shadow, a request could have been made with path traversal, URL encoding or normal LFI payload.

Searching for files

Apply filter: *etc*shadow*



The number of hits are 2 (Top left corner)

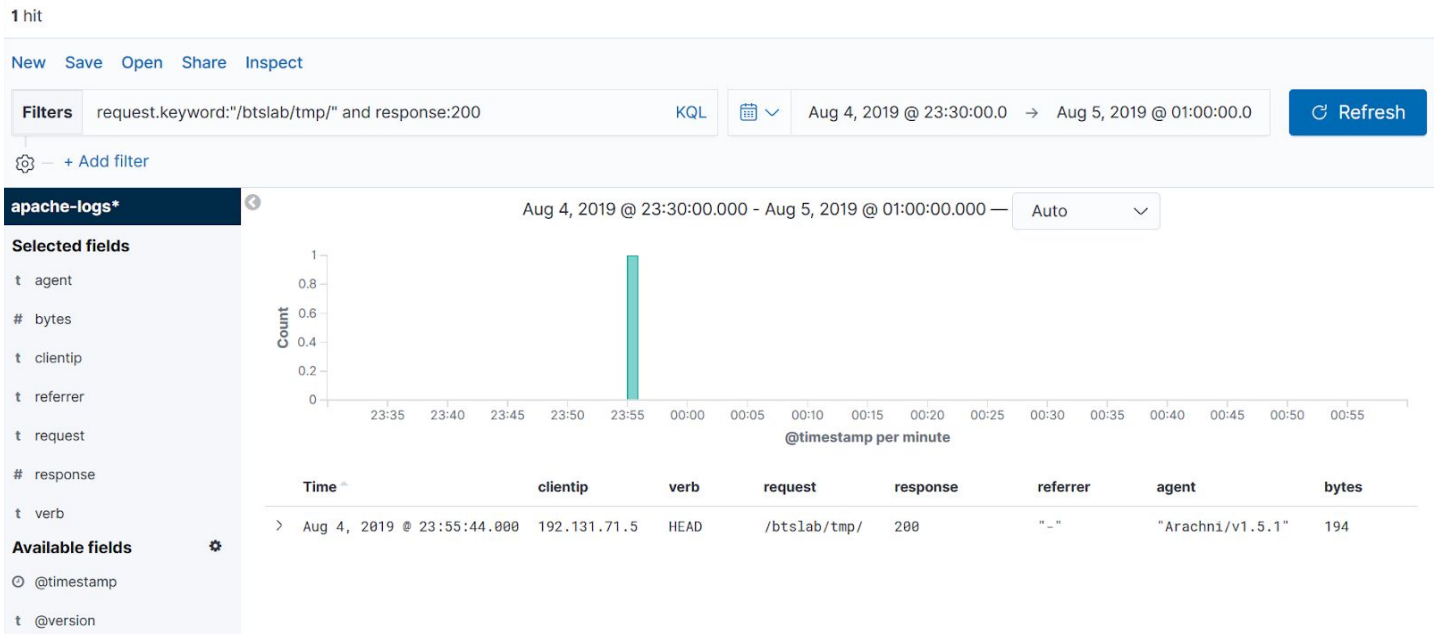
Q4. Which attacker machine was able to find out the existence of directory "/btslab/tmp/"? Provide the IP address.

Answer: 192.131.71.5

Solution:

Apply filter for request to the specified directory and 200 response.

Filter: request.keyword:"/btslab/tmp/" and response:200



Client 192.131.71.5 was able to find /btslab/tmp directory.

Q5. How many union based SQL injection attacks were attempted on the web application?

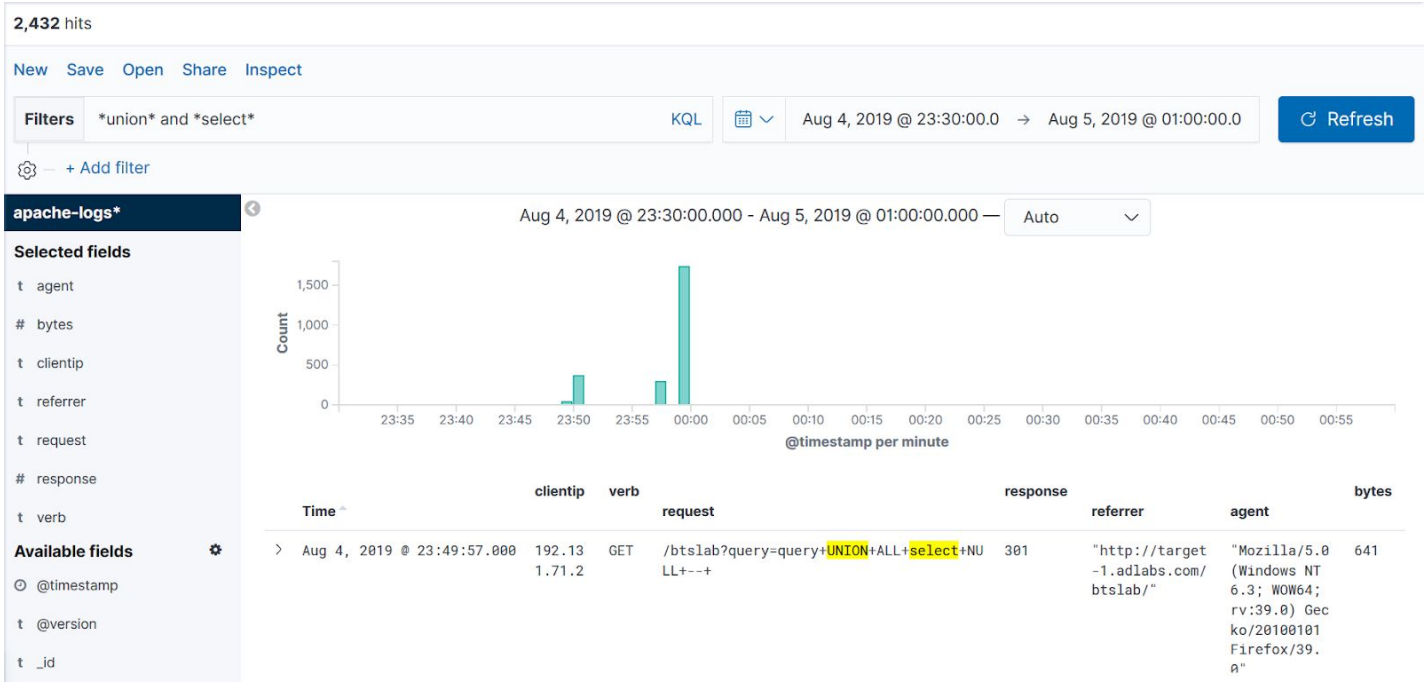
Answer: 2432

Solution:

Select statement is used with union in a SQL query.

A filter can be applied to look for “select” and “union” inside all fields.

Filter: *union* and *select*



The number of hits are 2432 (Top left corner)

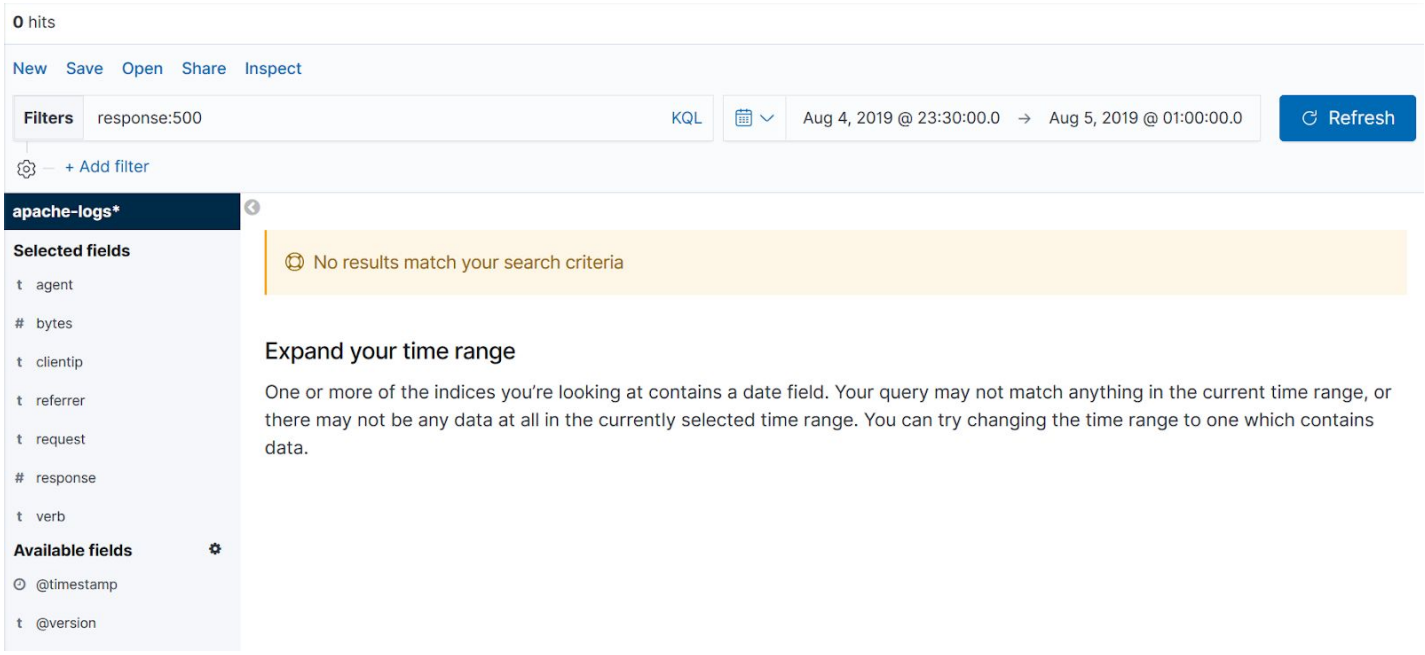
Q6. Does any of the request results in 500 internal error?

Answer: No

Solution:

Apply filter for 500 response

Filter: response:500



No result was found for 500 response code.

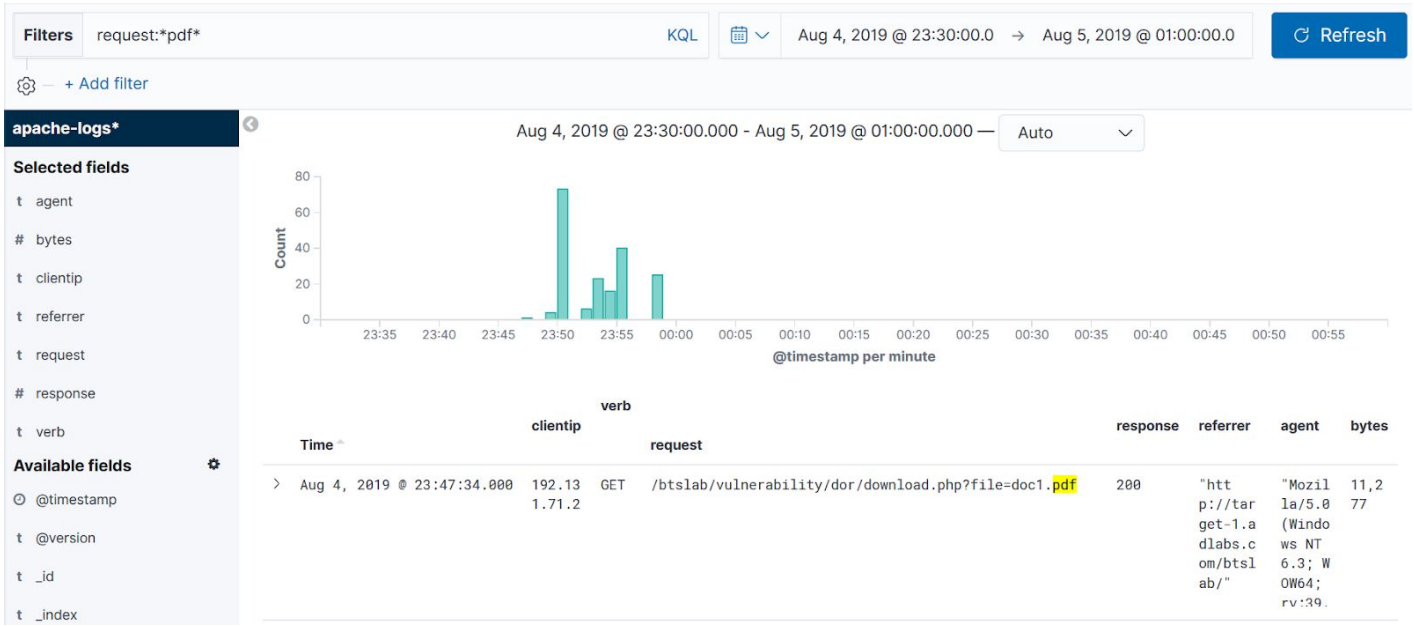
Q7. An HTTP GET request was made to a web page to download a PDF file. The same web page was vulnerable to PATH traversal attack and was exploited to download /etc/passwd file. Find the approximate size of /etc/passwd file in KB.

Answer: 1

Solution:

Apply filter to search for “pdf” in request.

Filter: request:*pdf*

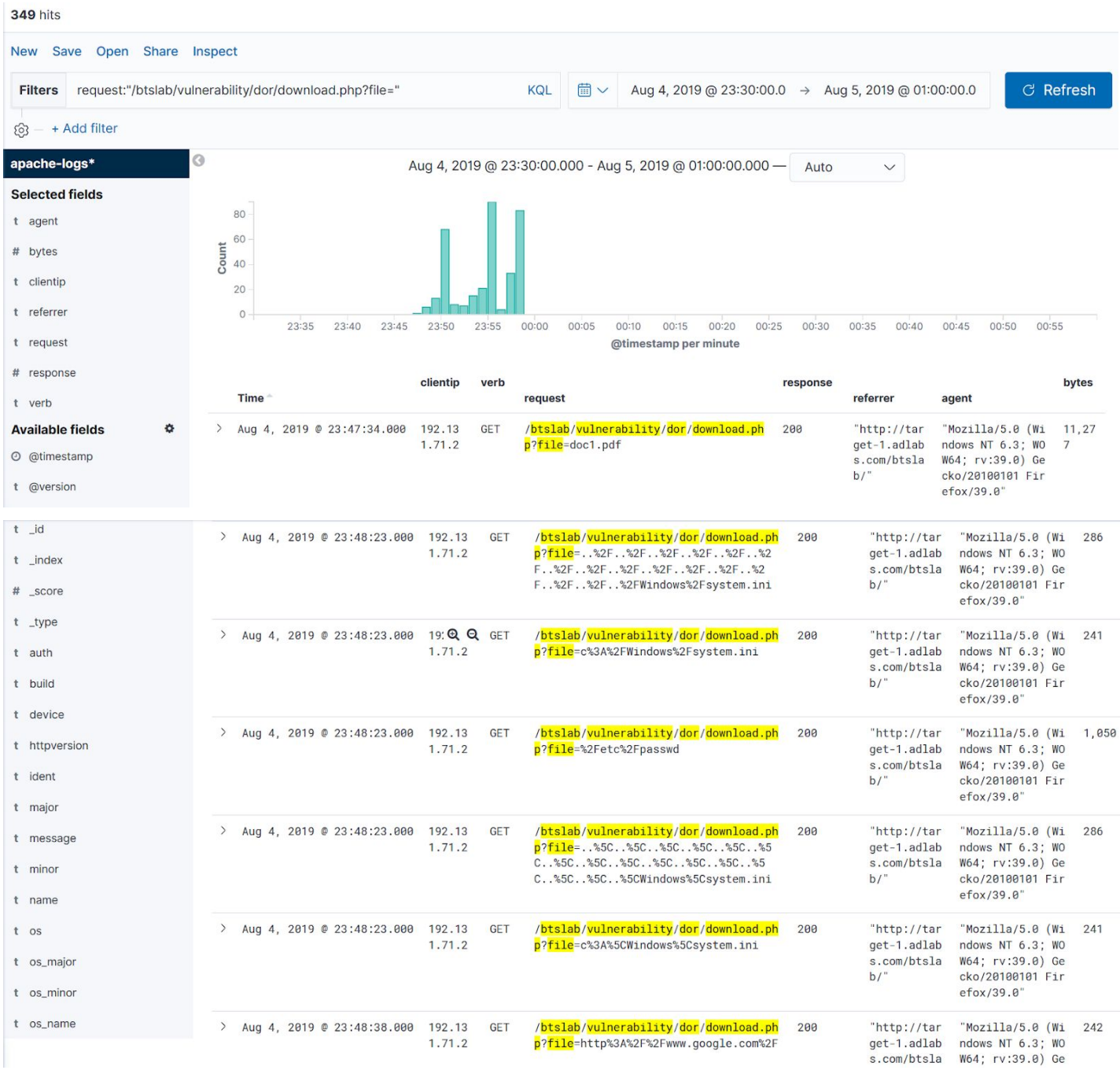


The request sent to download the PDF file was "/btslab/vulnerability/dor/download.php?file=doc1.pdf".

The response length was 11,277 bytes

The value of file parameter can be altered to download various files. Check the response length for various requests to identify the approximate response length when the file fails to download.

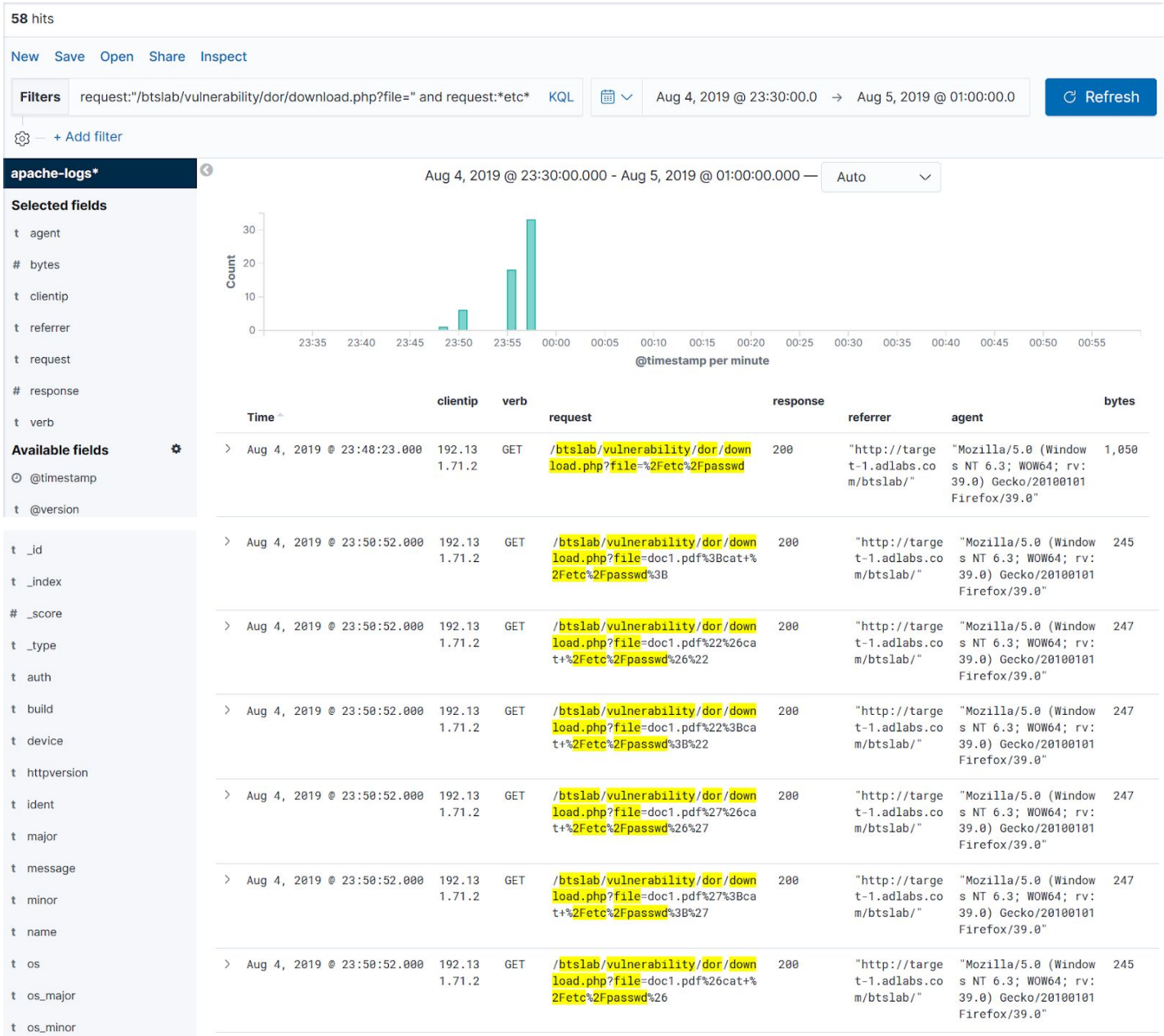
Filter: request:"/btslab/vulnerability/dor/download.php?file="



By observing the response length for various requests, it can be concluded that the approximate length of the response is around 240 bytes when a file fails to download.

Apply a filter to filter out request which contains “/etc/passwd” string.

Filter: request:"/btslab/vulnerability/dor/download.php?file=" and request:*etc* and request:*passwd*



The response length of the first request is more than 4 times the approximate response length. Therefore it can be assumed that, /etc/passwd file was downloaded in response of the first request.

The approximate size of /etc/passwd file is 1KB.

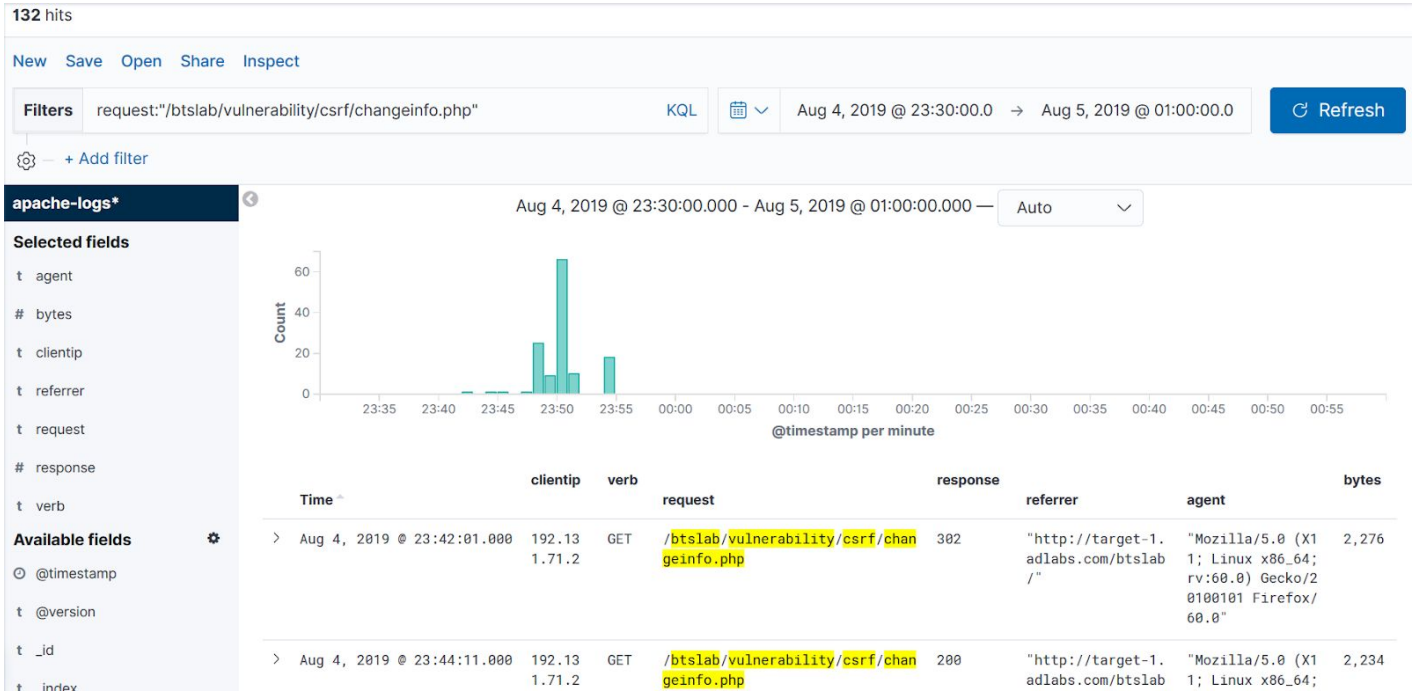
Q8. A CSRF attack was performed on the webpage "/btslab/vulnerability/csrf/changeinfo.php". Find the IP address of the machine from which the attack originated.

Answer: 192.131.71.2

Solution:

Apply filter for the request and the referrer for a valid request.

Filter: request:"/btslab/vulnerability/csrf/changeinfo.php"

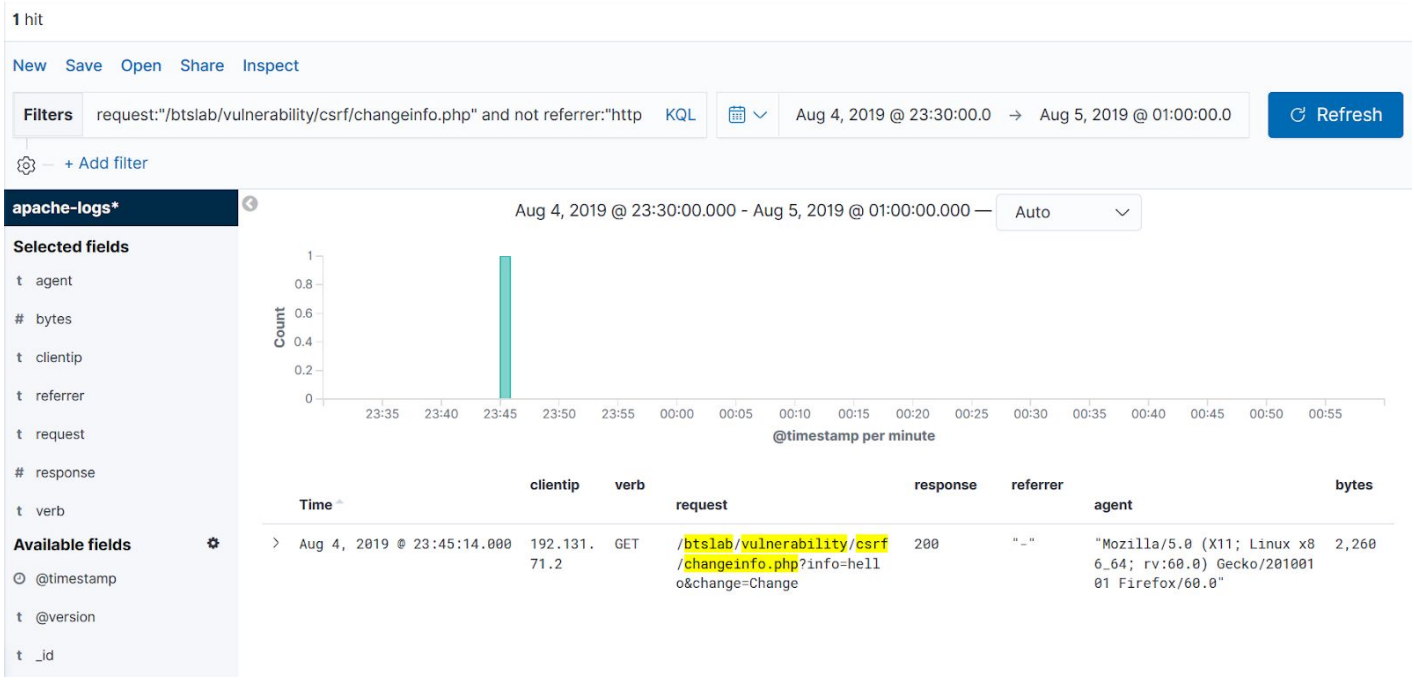


Most of the request have “http://target-1.adlabs.com/btslab/” in the refer.

The CSRF attack would have been originated from a different site than “http://target-1.adlabs.com/btslab/”.

Apply a filter and search where referrer does not match “http://target-1.adlabs.com/btslab/”.

Filter: request:"/btslab/vulnerability/csrf/changeinfo.php" and not referrer:"http://target-1.adlabs.com/btslab/"



Only one request made to the URL does not have a matching referer.

The attack originated from the machine with IP address 192.131.71.2

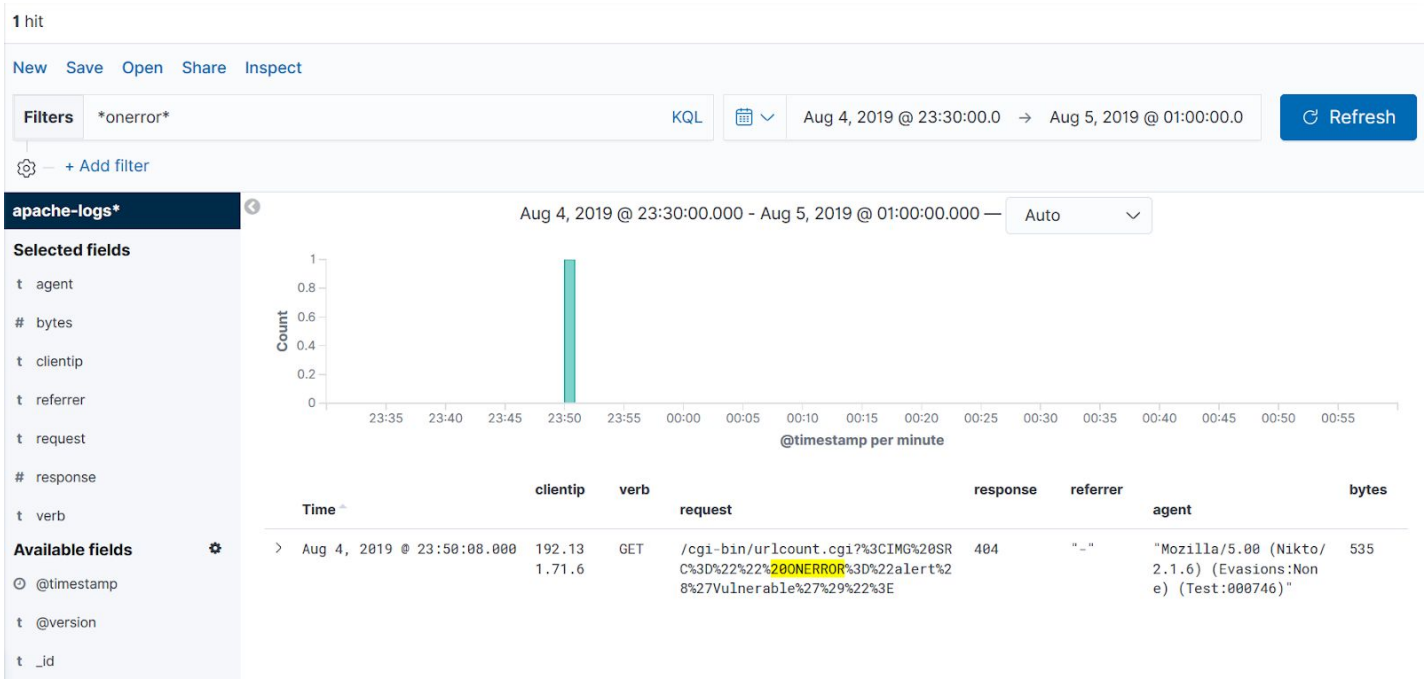
Q9. How many onerror based XSS attacks were attempted on the web application?

Answer: 1

Solution:

Apply filter and search for “onerror” string in request.

Filter: *onerror*



The number of hits are 1 (Top left corner)

Q10. Find the duration for which SQLmap was used to attack the web application. Provide the duration in seconds.

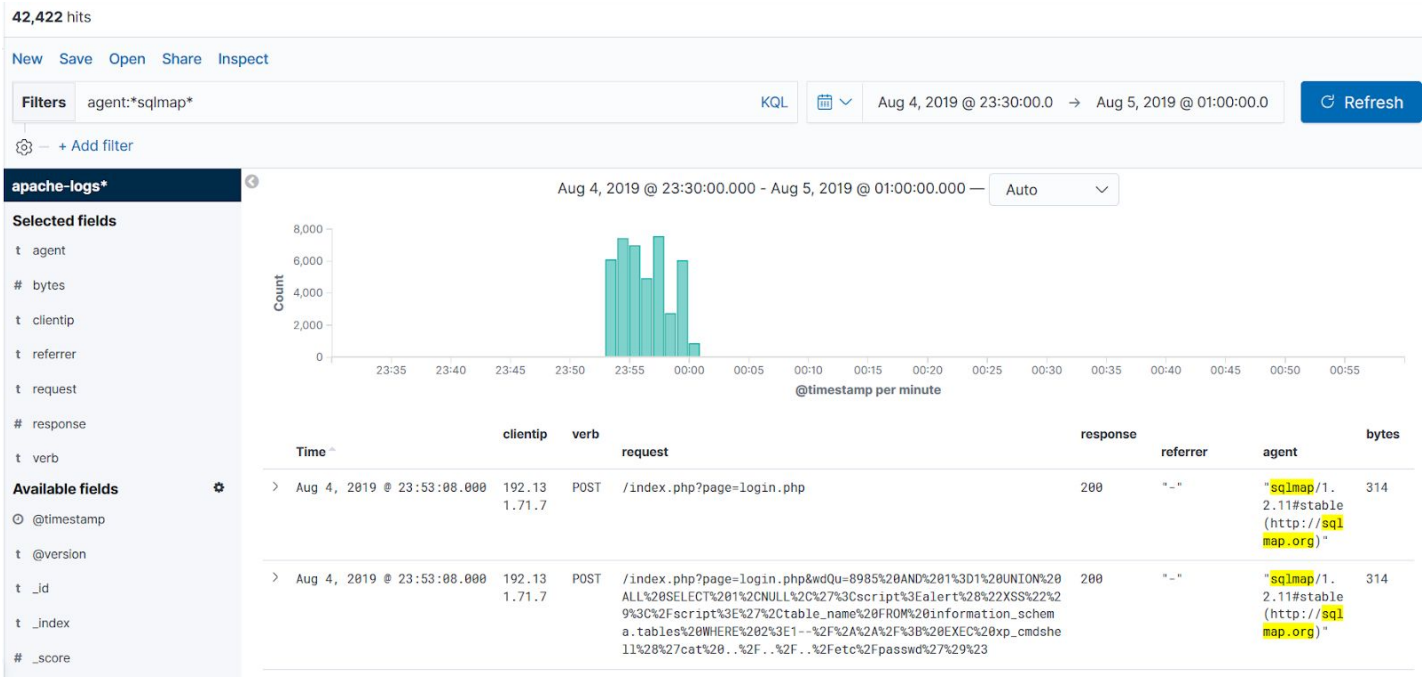
Answer: 434

Solution:

Apply filter for “sqlmap” user agent.

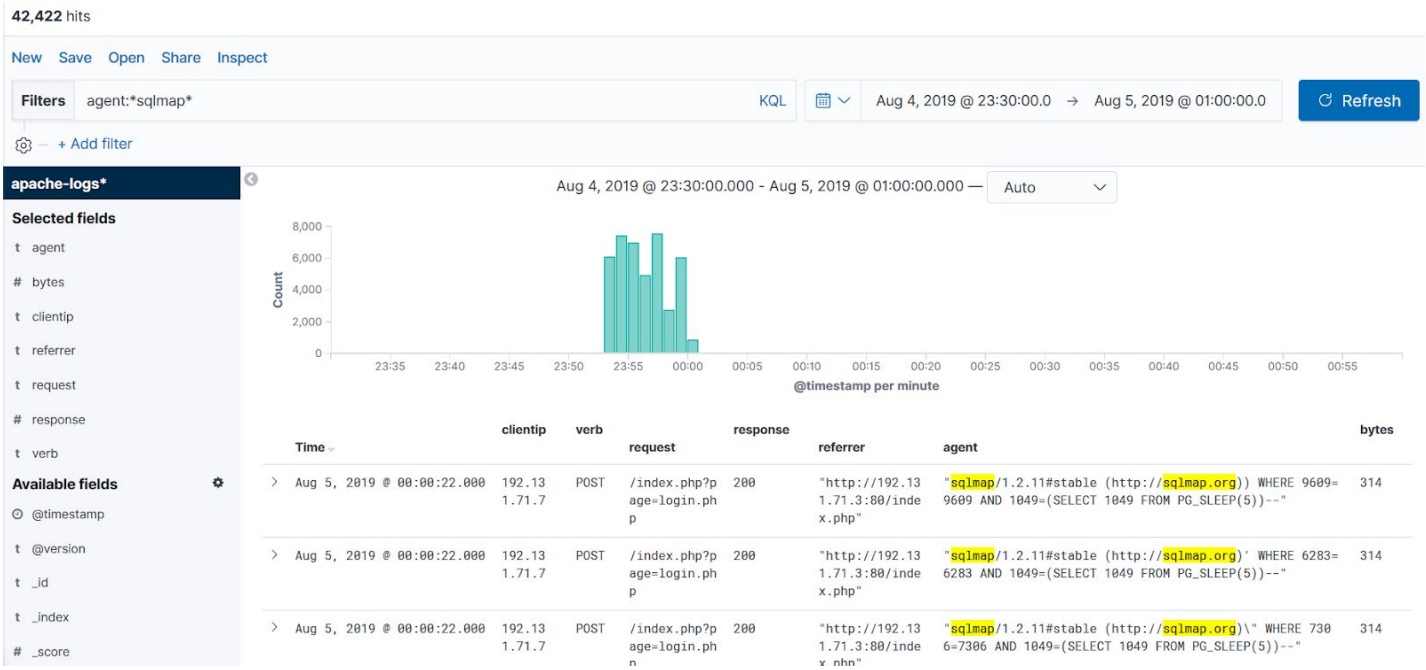
Filter: agent:*sqlmap*

Check the time for the first request.



The first request was received on 23:53:08.000

Sort the time in descending order.



The last request was received on 00:00:22.000

The time difference in seconds is: 434 seconds

References:

1. Kibana (https://www.elastic.co/products/kibana)
2. Kibana Query Language (https://www.elastic.co/guide/en/kibana/7.2/kuery-query.html)
3. Lucene Query Language (https://lucene.apache.org/core/2_9_4/queryparsersyntax.html)