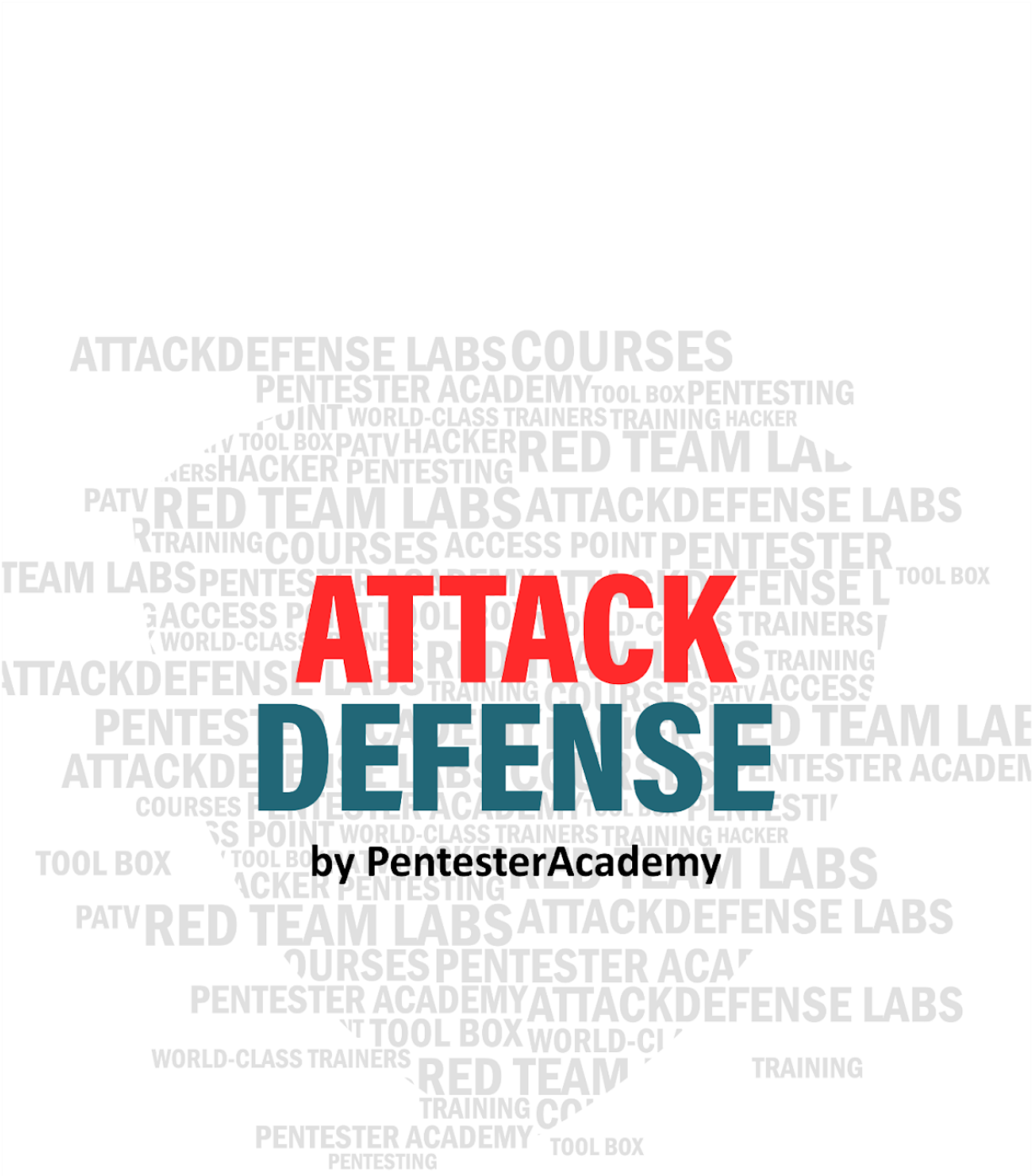

ATTACK
DEFENSE
by PentesterAcademy

Name	Kibana: Squid Log Analysis
URL	https://attackdefense.com/challengedetails?cid=1190
Type	Log Analysis : Proxy Logs

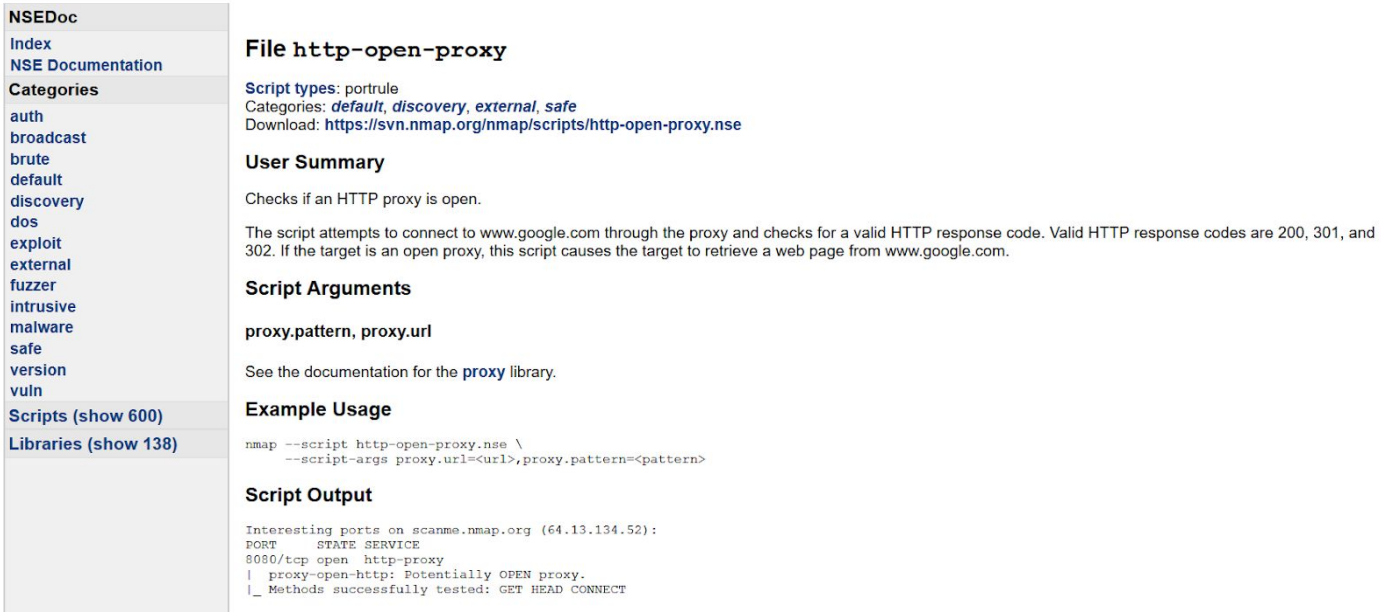
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this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

Q1. To figure out whether the squid proxy was configured without authentication or not, the attacker had used the nmap script 'http-open-proxy'. Find the IP address of the attacker machine.

Answer: 192.161.225.4

Solution:

Nmap Script: https://nmap.org/nsedoc/scripts/http-open-proxy.html



NSEDoc

Index
NSE Documentation

Categories

auth
broadcast
brute
default
discovery
dos
exploit
external
fuzzer
intrusive
malware
safe
version
vuln

Scripts (show 600)
Libraries (show 138)

File `http-open-proxy`

Script types: portrule
Categories: ***default***, ***discovery***, ***external***, ***safe***
Download: **https://svn.nmap.org/nmap/scripts/http-open-proxy.nse**

User Summary

Checks if an HTTP proxy is open.

The script attempts to connect to www.google.com through the proxy and checks for a valid HTTP response code. Valid HTTP response codes are 200, 301, and 302. If the target is an open proxy, this script causes the target to retrieve a web page from www.google.com.

Script Arguments

proxy.pattern, proxy.url

See the documentation for the **proxy** library.

Example Usage

```
nmap --script http-open-proxy.nse \
     --script-args proxy.url=<url>,proxy.pattern=<pattern>
```

Script Output

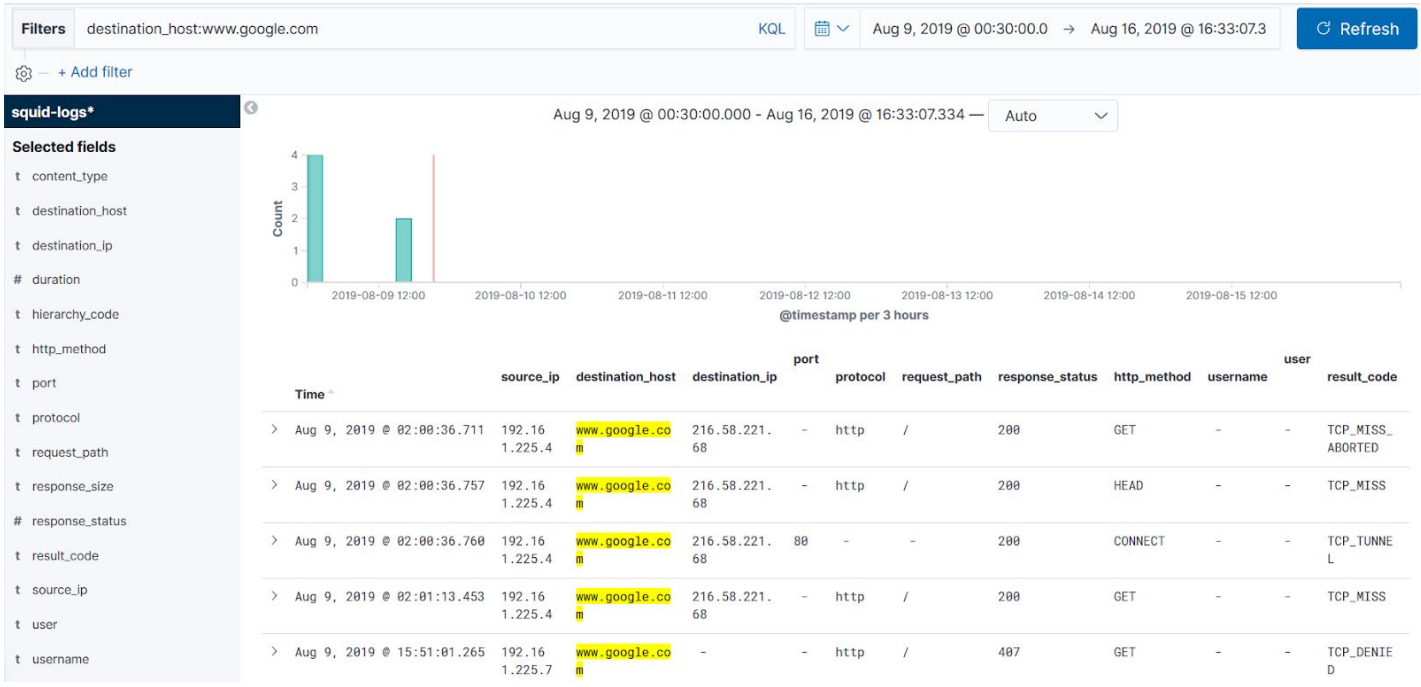
```
Interesting ports on scanme.nmap.org (64.13.134.52):
PORT     STATE SERVICE
8080/tcp open  http-proxy
|  proxy-open-http: Potentially OPEN proxy.
|_ Methods successfully tested: GET HEAD CONNECT
```

According to the description provided on the web page of the script. Nmap tries to send a request to “www.google.com” to determine whether the squid proxy is configured with authentication or not.

In the script output below, the methods which will be used are mentioned: GET, HEAD, CONNECT.

Apply a filter for destination host www.google.com

Filter: destination_host:www.google.com



The first three requests are sent to “www.google.com” in quick succession (within one second). The HTTP Method used are GET, HEAD and CONNECT same as the ones mentioned on the nmap script webpage.

The source IP is 192.161.225.4

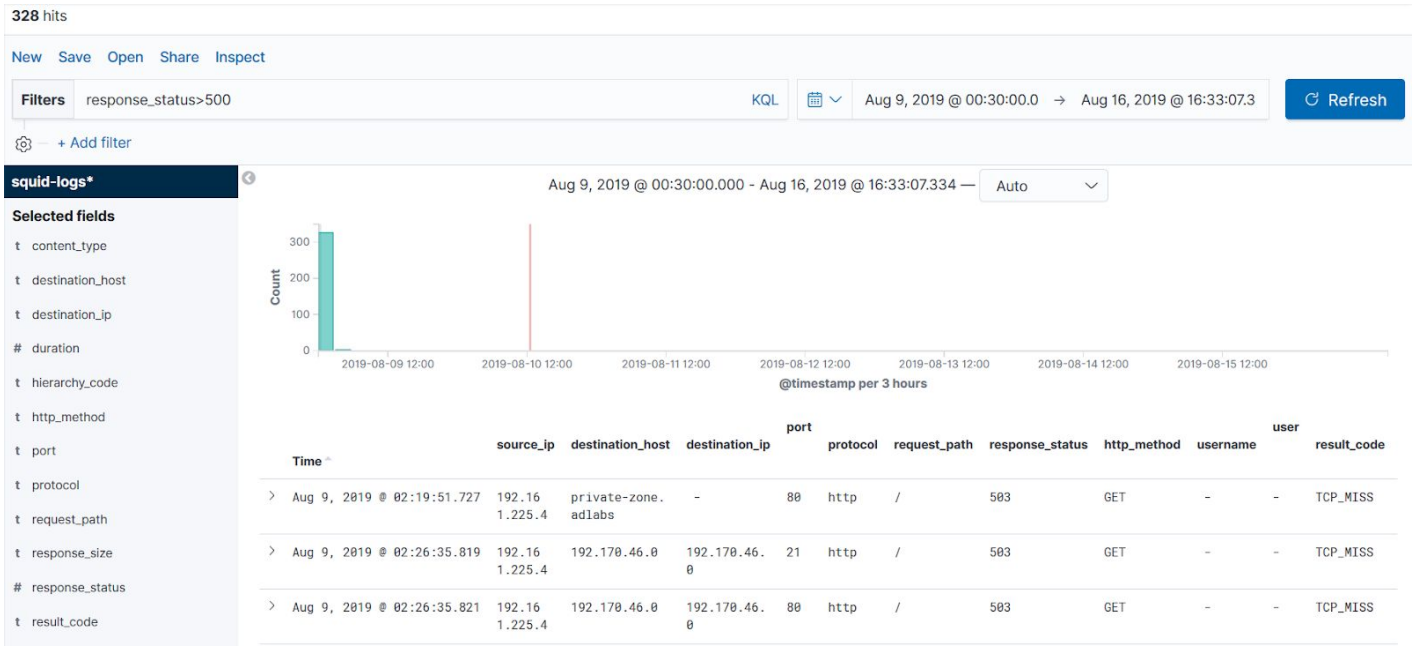
Q2. An attacker was trying to identify the live hosts on the network accessible through the squid proxy. Find the IP address of the attacker machine.

Answer: 192.161.225.4

Solution:

There might be few live hosts on the network, every other host will be down . The Response status code for such machines will be 503 (service unavailable). If multiple 503 response is generated for requests for many IP addresses or domains, it can be concluded that an attempt is being made to enumerate live hosts.

Filter: response_status>500



Aug 9, 2019 @ 02:26:46.144	192.16 1.225.4	192.170.46.1	192.170.46. 1	139	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.148	192.16 1.225.4	192.170.46.1	-	443	https	/	501	GET	-	-	TAG_NONE
Aug 9, 2019 @ 02:26:46.151	192.16 1.225.4	192.170.4	192.170.46. 1	445	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.155	192.16 1.225.4	192.170.46.1	192.170.46. 1	143 3	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.170	192.16 1.225.4	192.170.46.1	192.170.46. 1	152 1	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.203	192.16 1.225.4	192.170.46.1	192.170.46. 1	172 3	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.206	192.16 1.225.4	192.170.46.1	192.170.46. 1	338 9	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.209	192.16 1.225.4	192.170.46.1	192.170.46. 1	808 0	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.213	192.16 1.225.4	192.170.46.1	192.170.46. 1	910 0	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.216	192.16 1.225.4	192.170.46.2	192.170.46. 2	21	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.219	192.16 1.225.4	192.170.46.2	192.170.46. 2	80	http	/	503	GET	-	-	TCP_MISS
Aug 9, 2019 @ 02:26:46.222	192.16 1.225.4	192.170.46.2	192.170.46. 2	139	http	/	503	GET	-	-	TCP_MISS

Multiple 503 response status code is received for the IP range 192.170.46.0/24. The attack originated from 192.161.255.4

Q3. The attacker was sending requests to particular ports in order to identify live hosts on the network. How many ports were being scanned?. Provide the list of all ports.

Answer: 11

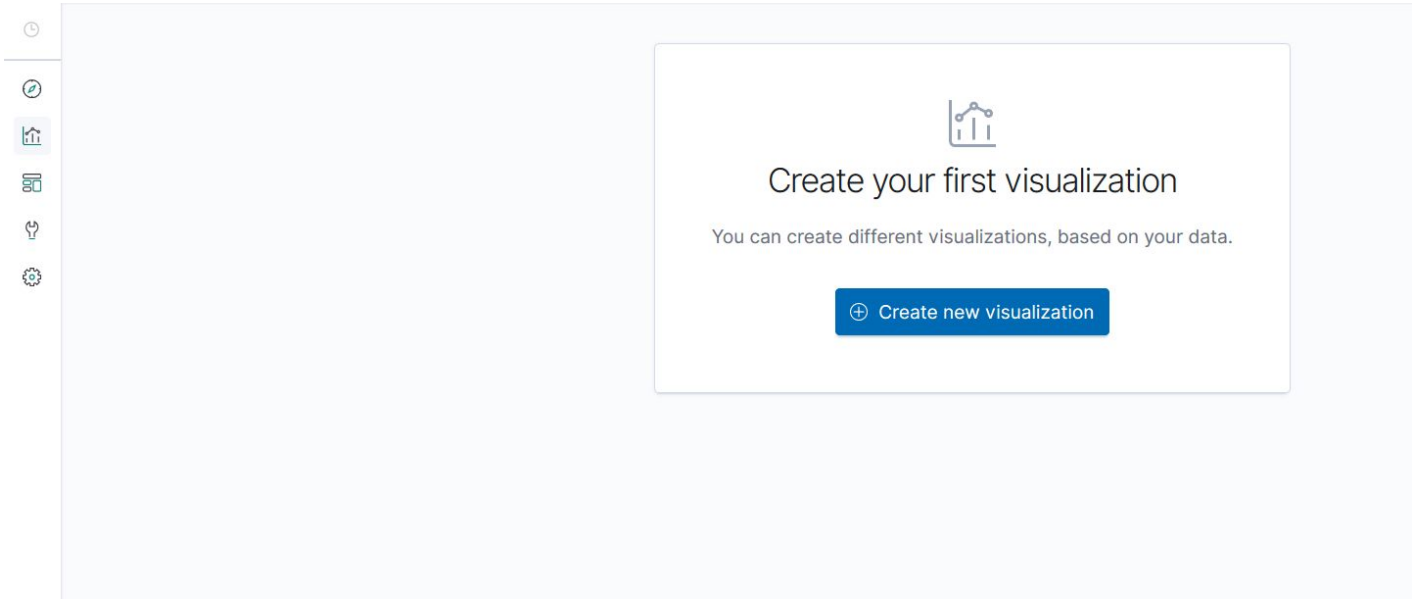
Scanned ports: 21, 80, 139, 443, 1433, 1521, 1723, 3389, 445, 8080, 9100

Solution:

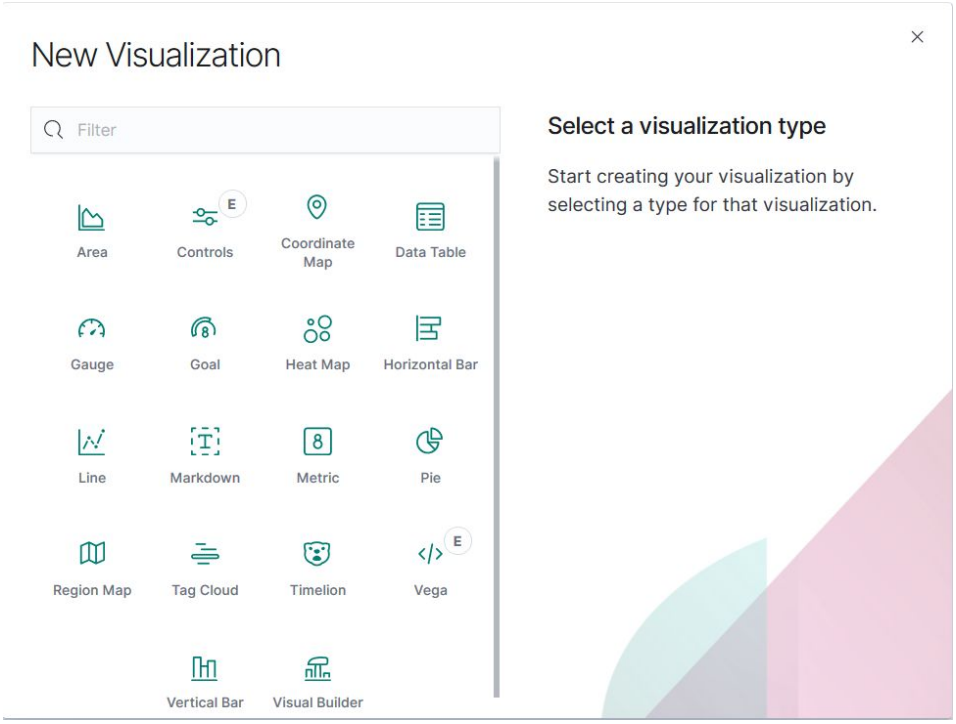
Create a visualization with the number of requests made to ports which originated from the IP address 192.161.225.4 and received a 503 response status code.

Navigate to Visualize section.

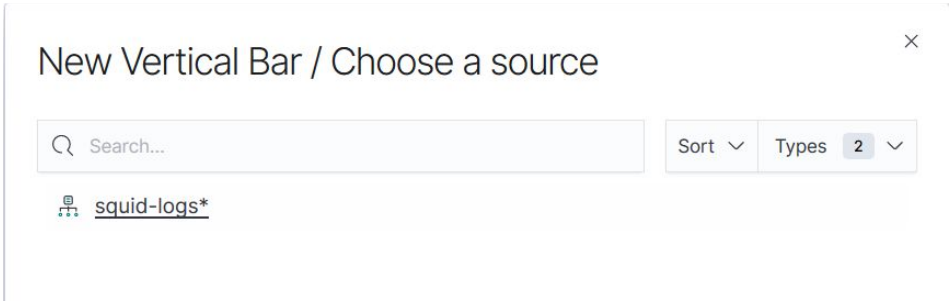
Click on the second icon on the left panel.



Click on “Create new Visualization”.

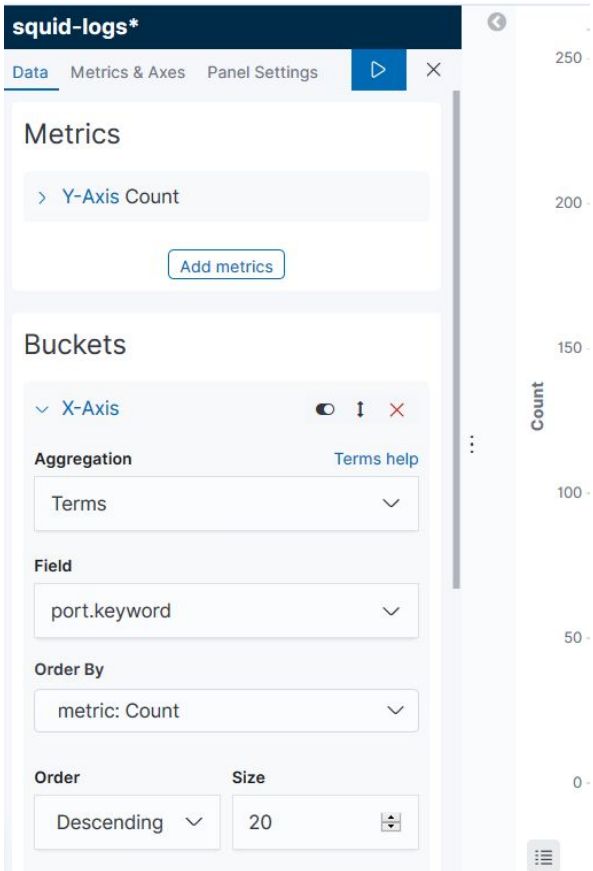


Select Vertical Bar:



Select squid-logs:

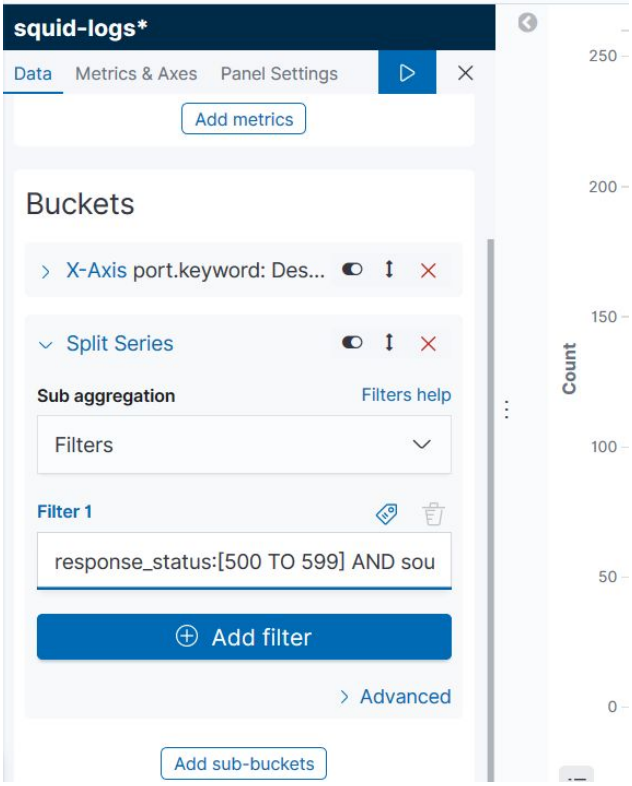
In X-axis, specify “terms” as aggregation and select “port.keyword” in the field. Increase the size to 20



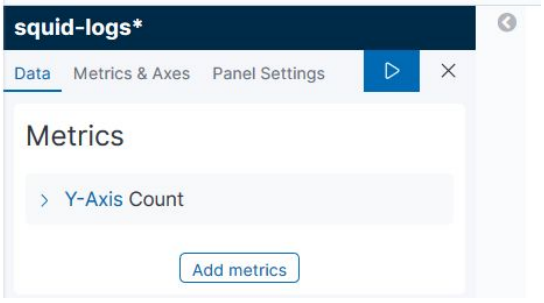
Add a “Split Series” sub basket:

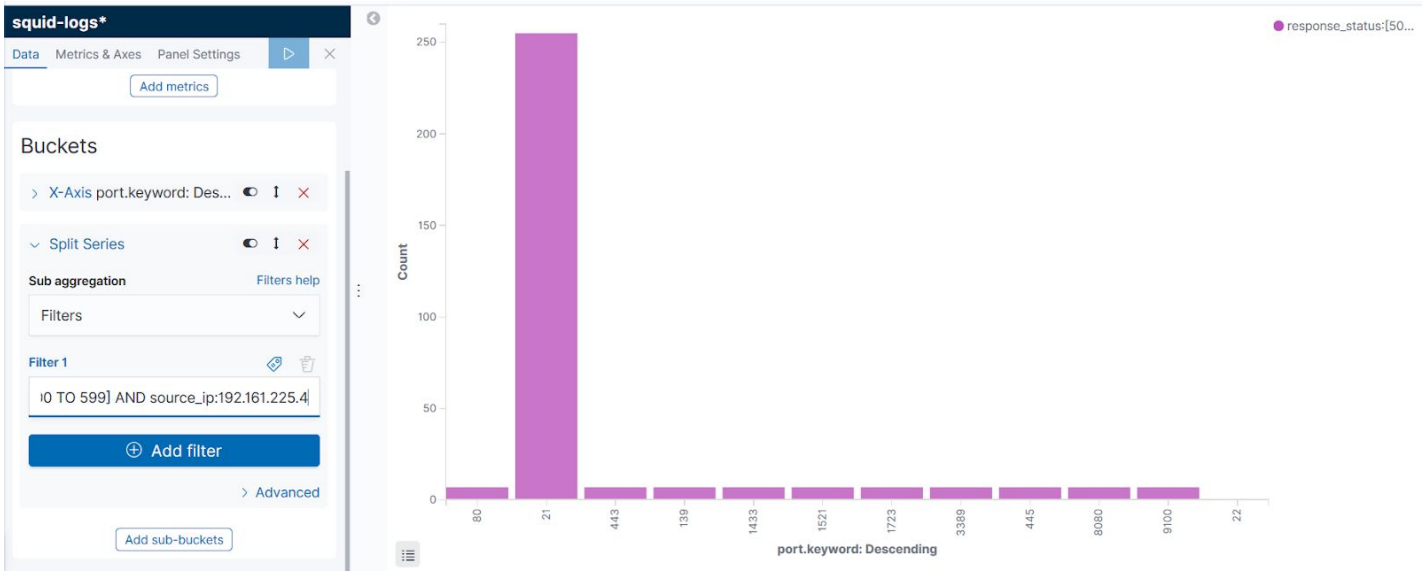
Select “Filters” in Sub aggregation and enter the following filter:

Filter: response_status:[500 TO 599] AND source_ip:192.161.225.4



Click on the Run button on the top right corner.





Total of 11 ports were scanned on 7 machines and port 21 was scanned on all of the machines.

The particular ports on which the scan was directed were: 21, 80, 139, 443, 1433, 1521,1723, 3389, 445, 8080, 9100.

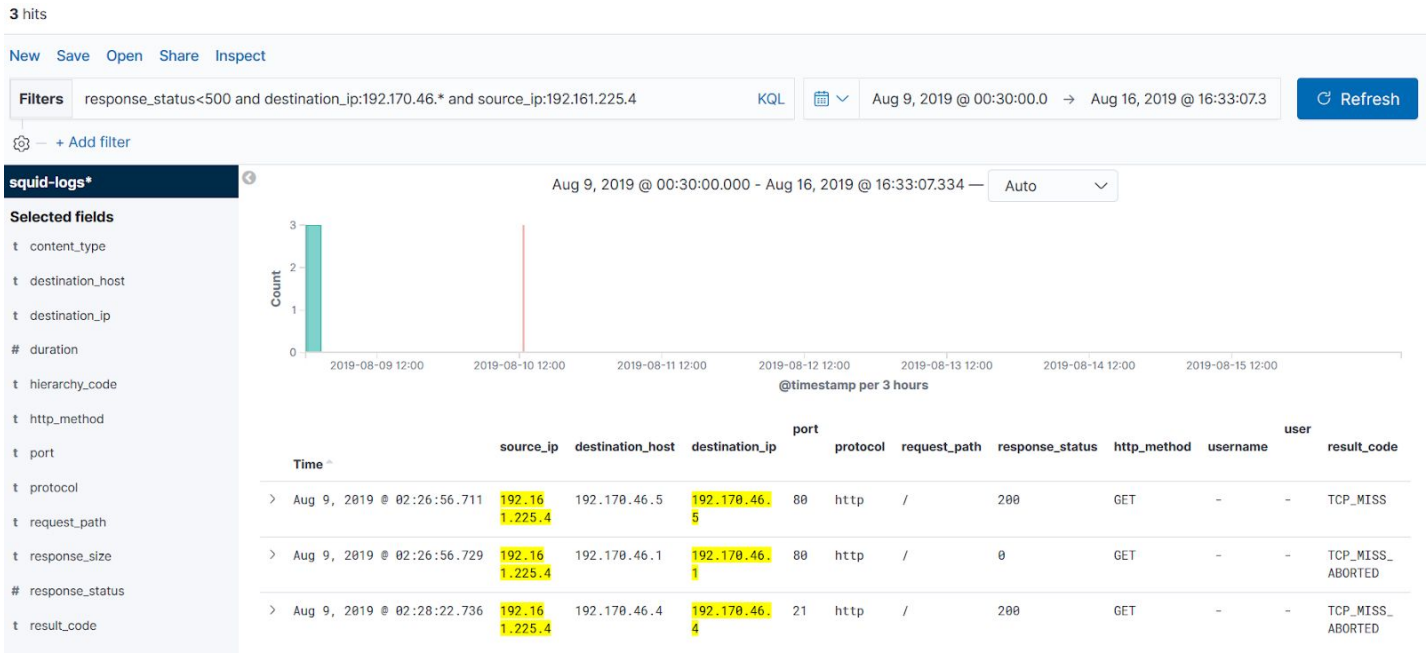
Q4. How many live hosts were detected in the scan performed by the attacker machine identified in the second question?

Answer: 3

Solution:

Apply the filter to look for requests made from the attacker IP to the network behind squid proxy where the response status is not 503 (Service unavailable).

Filter: response_status < 500 and destination_ip:192.170.46.5 and source_ip:192.161.225.4



Three live hosts were detected. The live hosts were 192.170.46.1, 192.170.46.4, 192.170.46.5.

The host 192.170.46.1 is live but since the response status code is 000, it cannot be concluded that port 80 was open or not.
The host 192.170.46.5 was alive and port 80 was open.
The host 192.170.46.5 was alive and port 21 was open.

Q5. An attacker had used a popular scanning tool on the web server running on one of the machines. Find the IP address of the attacker machine.

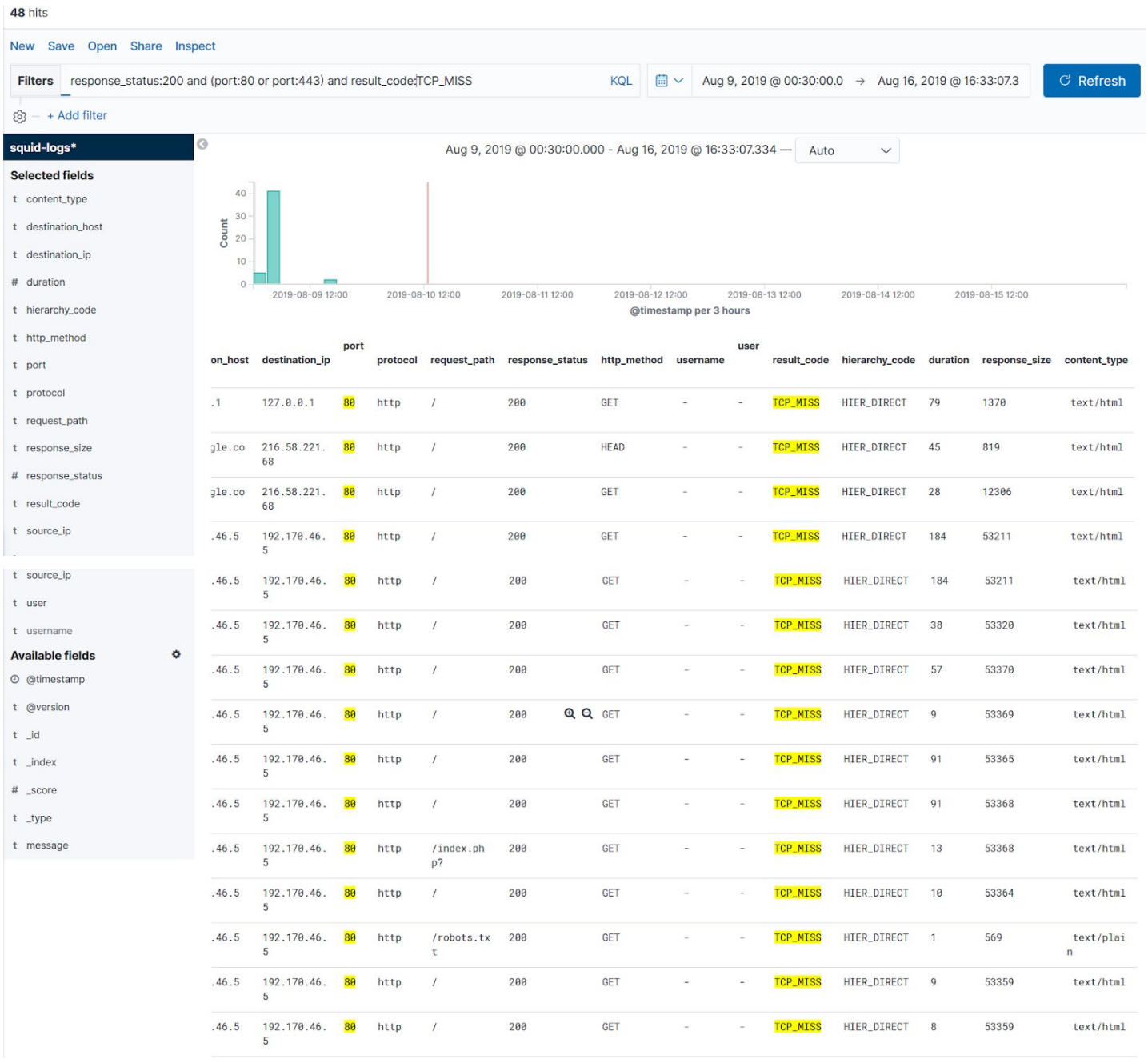
Answer: 192.161.225.5

Solution:

In the previous question, Port 80 was detected as open on target machine with IP address 192.170.46.5.

Verify whether the content type was “text/html” for the request which received 200 response status. Also apply TCP_MISS in result code to filter data obtained directly from the web server.

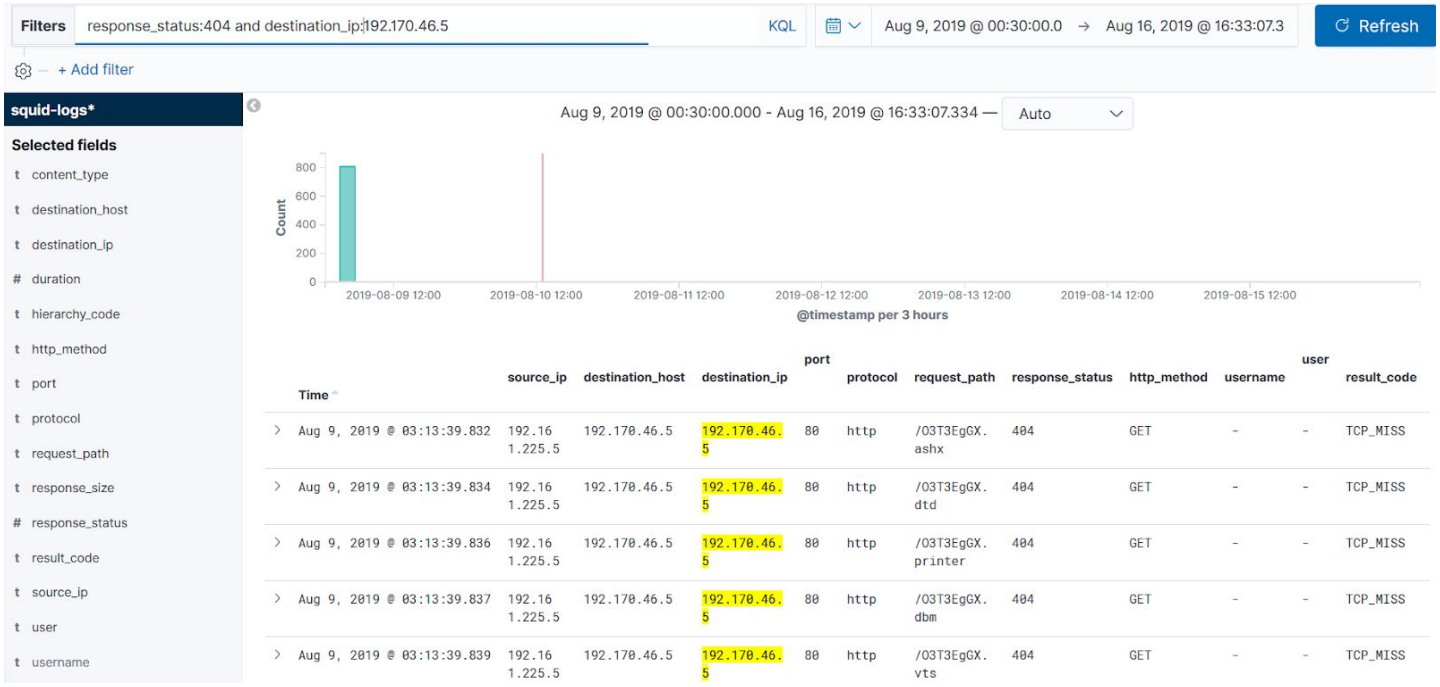
Filter: response_status:200 and (port:80 or port:443) and result_code:TCP_MISS



on_host	destination_ip	port	protocol	request_path	response_status	http_method	username	user	result_code	hierarchy_code	duration	response_size	content_type
.1	127.0.0.1	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	79	1370	text/html
gle.co	216.58.221.68	80	http	/	200	HEAD	-	-	TCP_MISS	HIER_DIRECT	45	819	text/html
gle.co	216.58.221.68	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	28	12306	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	184	53211	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	184	53211	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	38	53320	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	57	53370	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	9	53369	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	91	53365	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	91	53368	text/html
.46.5	192.170.46.5	80	http	/index.php?	200	GET	-	-	TCP_MISS	HIER_DIRECT	13	53368	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	10	53364	text/html
.46.5	192.170.46.5	80	http	/robots.txt	200	GET	-	-	TCP_MISS	HIER_DIRECT	1	569	text/plain
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	9	53359	text/html
.46.5	192.170.46.5	80	http	/	200	GET	-	-	TCP_MISS	HIER_DIRECT	8	53359	text/html

The webserver is running on the host with IP address 192.170.46.5

Since a scanner is used against the webserver, there are high chances many requests would have resulted in a 404 response. Apply a filter for 404 response status code for the corresponding webserver.



404 response status was received for the requests made to random paths on the webserver. The requests originated from the machine with IP address 192.161.225.5

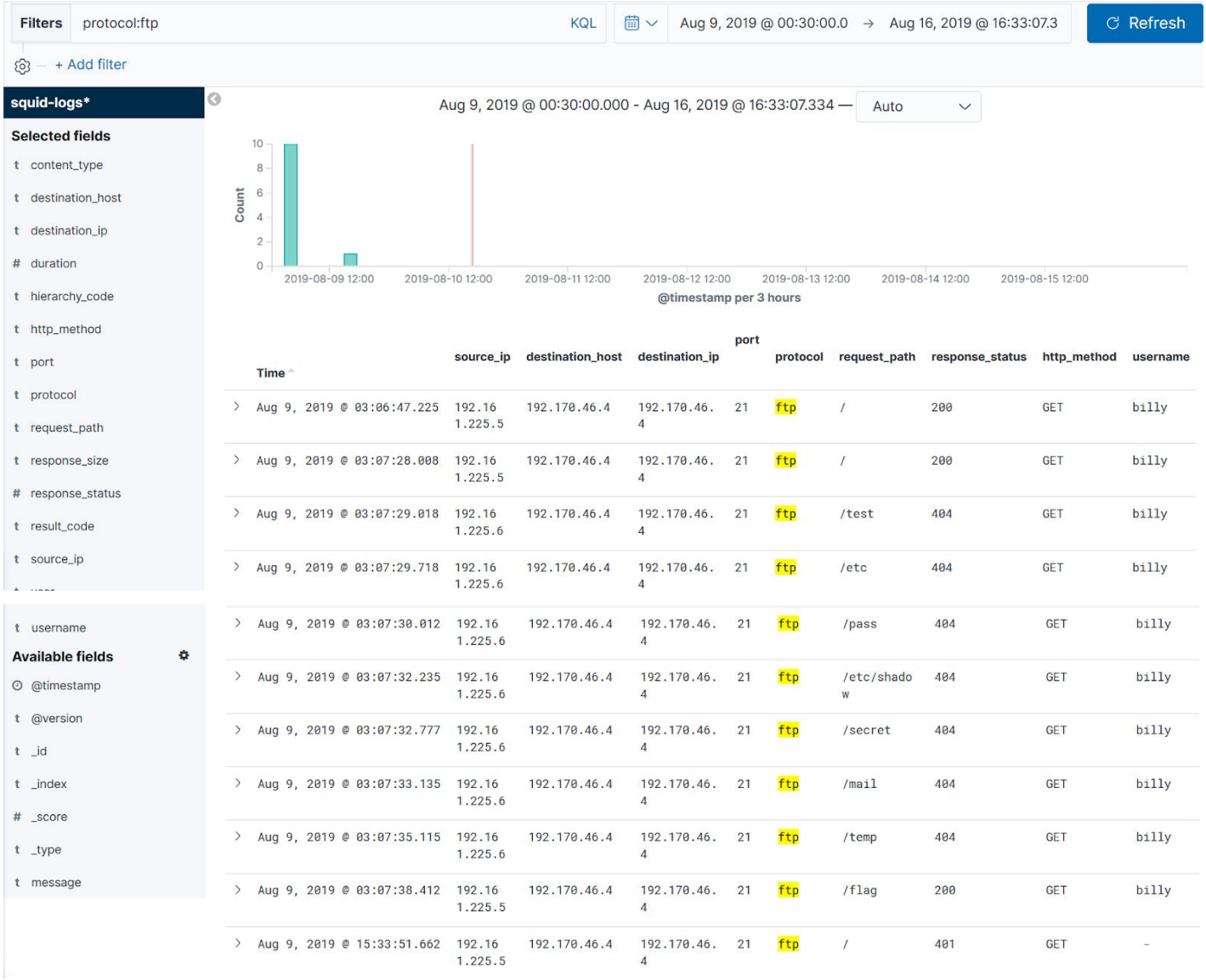
Q6. An attacker had logged in to the FTP server to retrieve a file. Find the FTP username used by the attacker to access the FTP server.

Answer: billy

Solution:

Apply a filter for FTP protocol and check the username field.

Filter: protocol:ftp

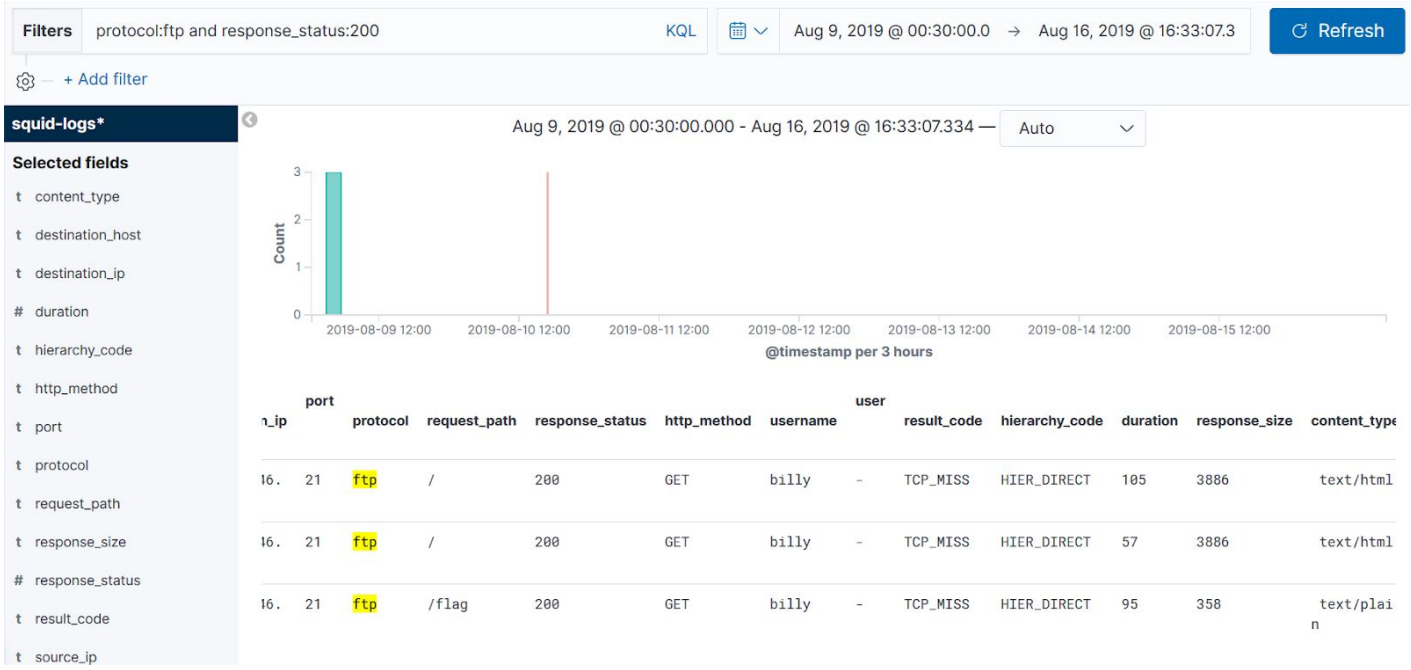


Time	source_ip	destination_host	destination_ip	port	protocol	request_path	response_status	http_method	username
Aug 9, 2019 @ 03:06:47.225	192.16 1.225.5	192.170.46.4	192.170.46. 4	21	ftp	/	200	GET	billy
Aug 9, 2019 @ 03:07:28.008	192.16 1.225.5	192.170.46.4	192.170.46. 4	21	ftp	/	200	GET	billy
Aug 9, 2019 @ 03:07:29.018	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/test	404	GET	billy
Aug 9, 2019 @ 03:07:29.718	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/etc	404	GET	billy
Aug 9, 2019 @ 03:07:30.012	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/pass	404	GET	billy
Aug 9, 2019 @ 03:07:32.235	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/etc/shado w	404	GET	billy
Aug 9, 2019 @ 03:07:32.777	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/secret	404	GET	billy
Aug 9, 2019 @ 03:07:33.135	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/mail	404	GET	billy
Aug 9, 2019 @ 03:07:35.115	192.16 1.225.6	192.170.46.4	192.170.46. 4	21	ftp	/temp	404	GET	billy
Aug 9, 2019 @ 03:07:38.412	192.16 1.225.5	192.170.46.4	192.170.46. 4	21	ftp	/flag	200	GET	billy
Aug 9, 2019 @ 15:33:51.662	192.16 1.225.5	192.170.46.4	192.170.46. 4	21	ftp	/	401	GET	-

Multiple requests were sent to access certain files on the FTP server, since the response status code received for many requests was 404, it can be concluded that malicious activity is going on.

The response status code for few request was 200

Filter: protocol:ftp and response_status:200



The content type of the response received for the request sent to /flag is text/plain.

The flag file was retrieved by the attacker machine.

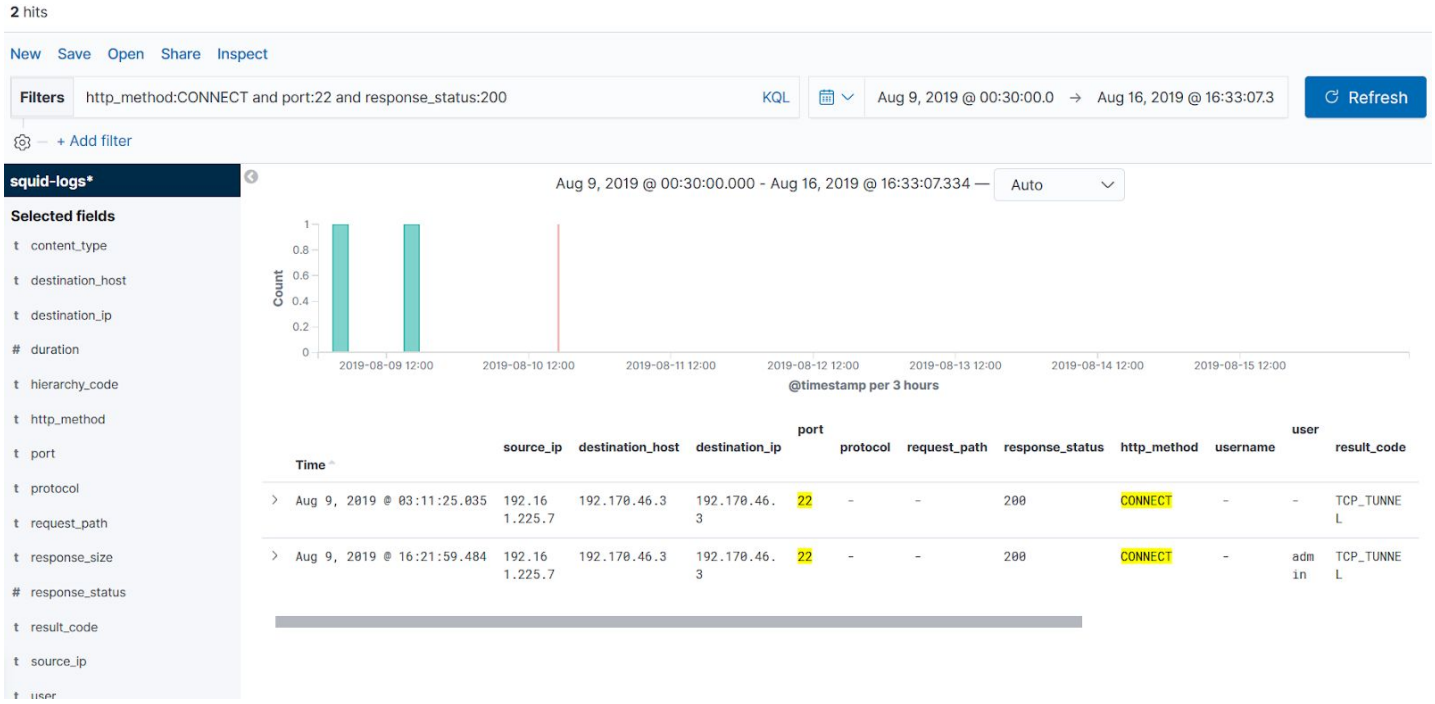
The FTP username used by the attacker to retrieve the file was “billy”.

Q7. An attacker machine had leveraged the open squid proxy to obtain an SSH session on one of the machines. Find the IP address of the SSH server.

Answer: 192.170.46.3

An SSH session requires a HTTP tunnel to be created with the help of HTTP CONNECT method. Since the SSH session was successful, the response status code will be 200.

Filter: http_method:CONNECT and port:22 and response_status:200



While accessing open squid proxy server, the user field will be empty.

The IP address of the SSH server is 192.170.46.3

Q8. An attacker was performing a dictionary attack on the FTP server using hydra. Identify the IP address of the attacker machine.

Answer: 192.161.225.6

Solution:

Hydra package description link: https://tools.kali.org/password-attacks/hydra

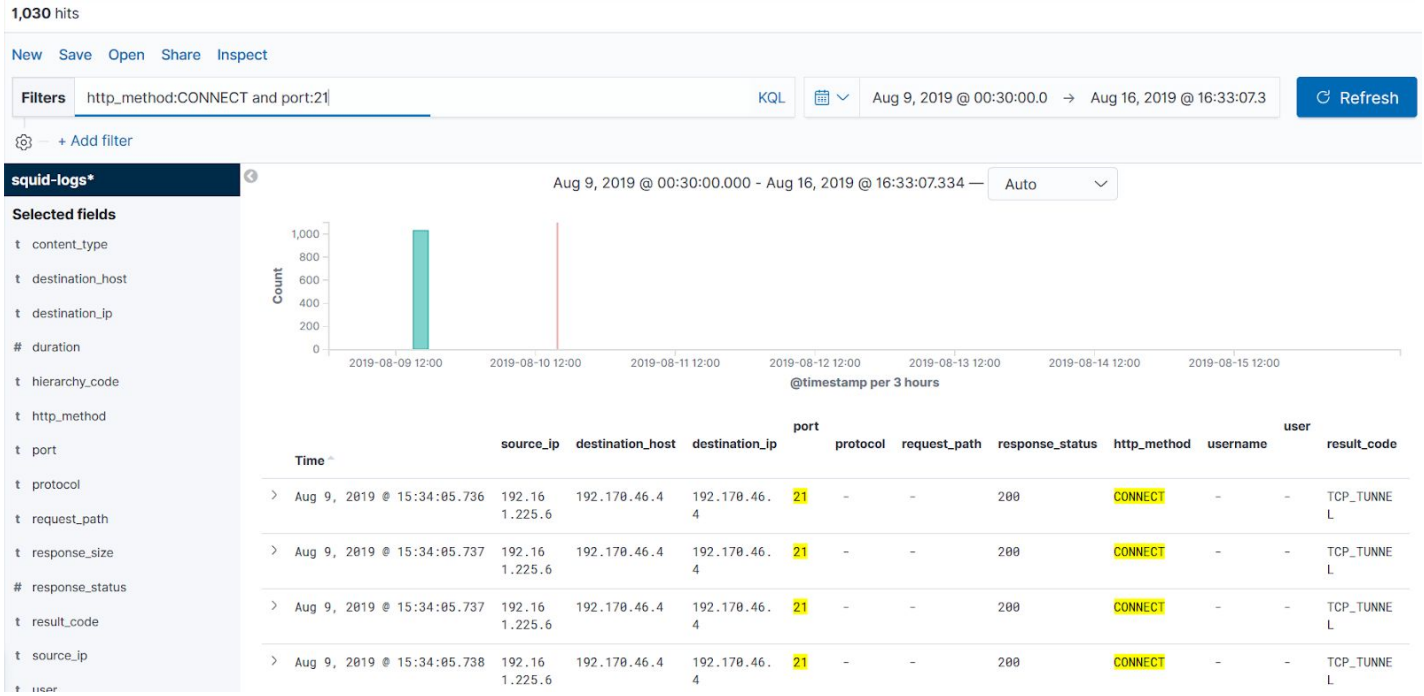
```
Hydra is a tool to guess/crack valid login/password pairs - usage only allowed
for legal purposes. This tool is licensed under AGPL v3.0.
The newest version is always available at http://www.thc.org/thc-hydra
These services were not compiled in: sapr3 oracle.

Use HYDRA_PROXY_HTTP or HYDRA_PROXY - and if needed HYDRA_PROXY_AUTH - environment for a proxy setup.
E.g.:  % export HYDRA_PROXY=socks5://127.0.0.1:9150 (or socks4:// or connect://)
       % export HYDRA_PROXY_HTTP=http://proxy:8080
       % export HYDRA_PROXY_AUTH=user:pass
```

Hydra can be used in conjunction with proxy to perform dictionary attacks.

The Hydra can utilize the HTTP connect method to create a tunnel.

Apply filter for HTTP CONNECT method and port 21 and check the time difference between consecutive requests. If the time difference is very less between consecutive requests, it can be concluded that a possible attack is going on.

Filter: http_method:CONNECT and port:21



Around 33 requests were sent in under 1 second on port 21.

The IP address of the source machine is: 192.161.225.6

Q9. Upon detecting malicious activity, the squid proxy was configured with authentication. However, an attacker was able to perform a dictionary attack and found out the password of one of the users. Find out the username whose password was compromised.

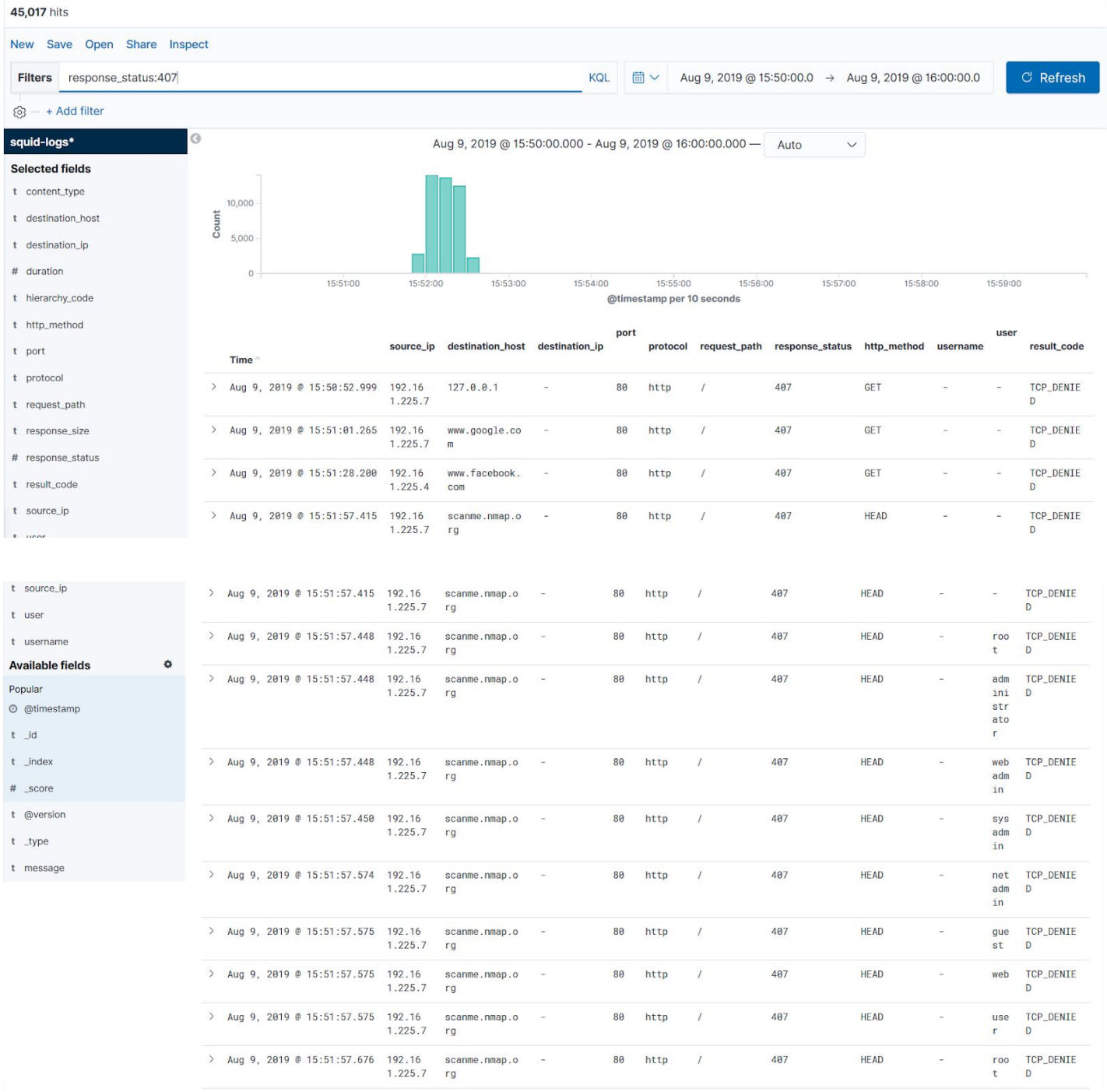
Answer: admin

Solution:

The squid proxy will respond with 407 (Proxy Authentication Required) status code incase the credentials are not provided or are incorrect. The Result code will be TCP_DENIED

Apply a filter for 407 response status code and check the time difference between consecutive requests. (The result code can also be applied as a filter)

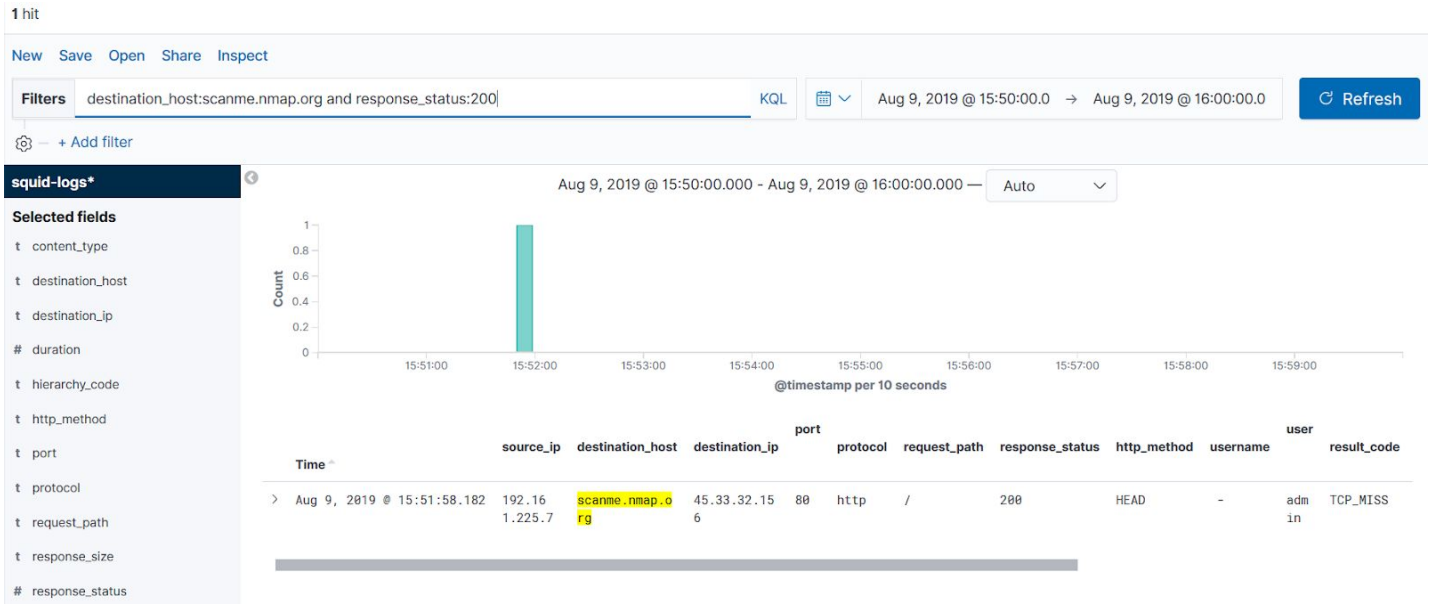
Filter: response_status:407



More than 100 requests have been sent under 1 second with various username, an attempt was being made to access scanme.nmap.org through the proxy.

To identify if the attempt was successful, apply a filter for the destination host scanme.nmap.org and look for 200 response status code

Filter: destination_host:scanme.nmap.org and response_status:200



Since the request was forwarded to scanme.nmap.org, it can be concluded that the attacker was able to discover the correct credentials of the squid proxy server.

The attacker was able to discover the password for user “admin”.

Q10. Find the approximate duration of the SSH session initiated by the attacker machine through the password protected squid proxy. Provide the duration in seconds.

Answer: 28

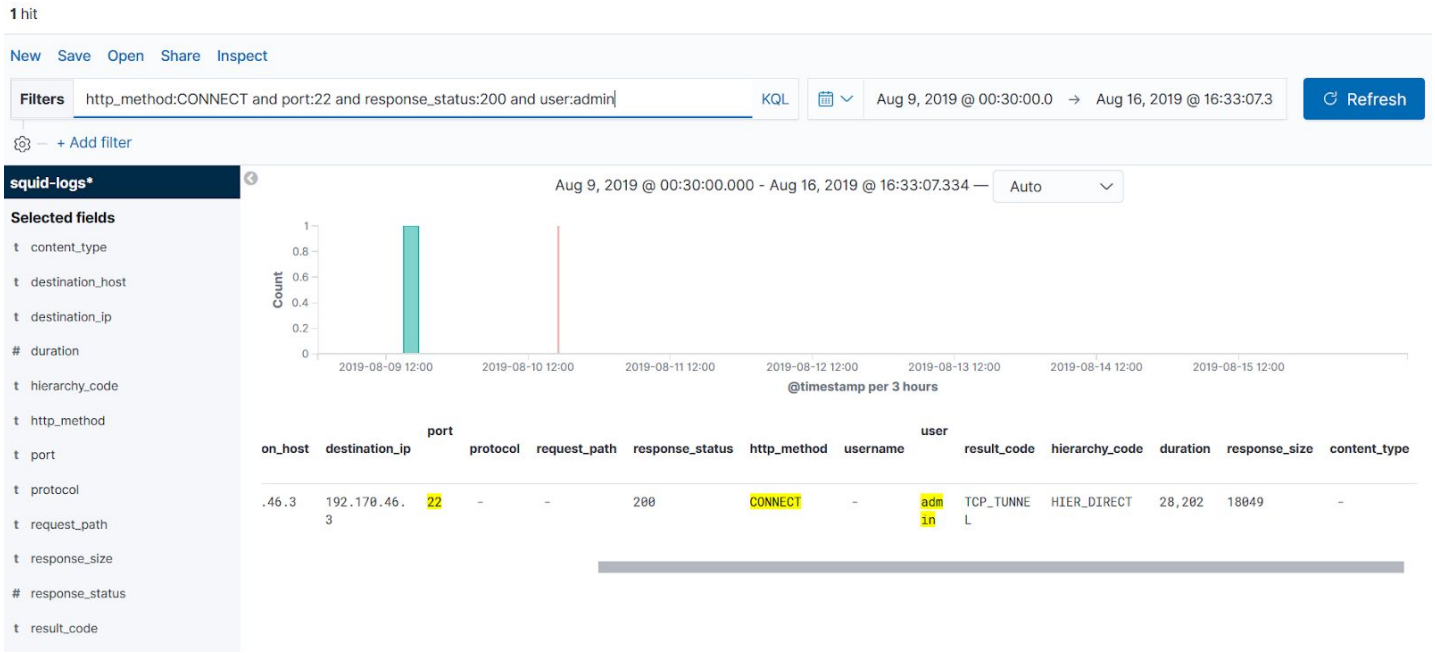
Solution:

An SSH session requires a HTTP tunnel to be created with the help of HTTP CONNECT method. Since the SSH session was successful, the response status code will be 200. The user field will contain a username which is required to authenticate with the squid proxy server.

The squid user was “admin”

Apply a filter for HTTP CONNECT method and check for requests which received 200 response status code.

Filter: http_method:CONNECT and port:22 and response_status:200 and user:admin



The duration for which the squid proxy was busy was 28202 milliseconds.

The SSH session would have lasted for around 28 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)