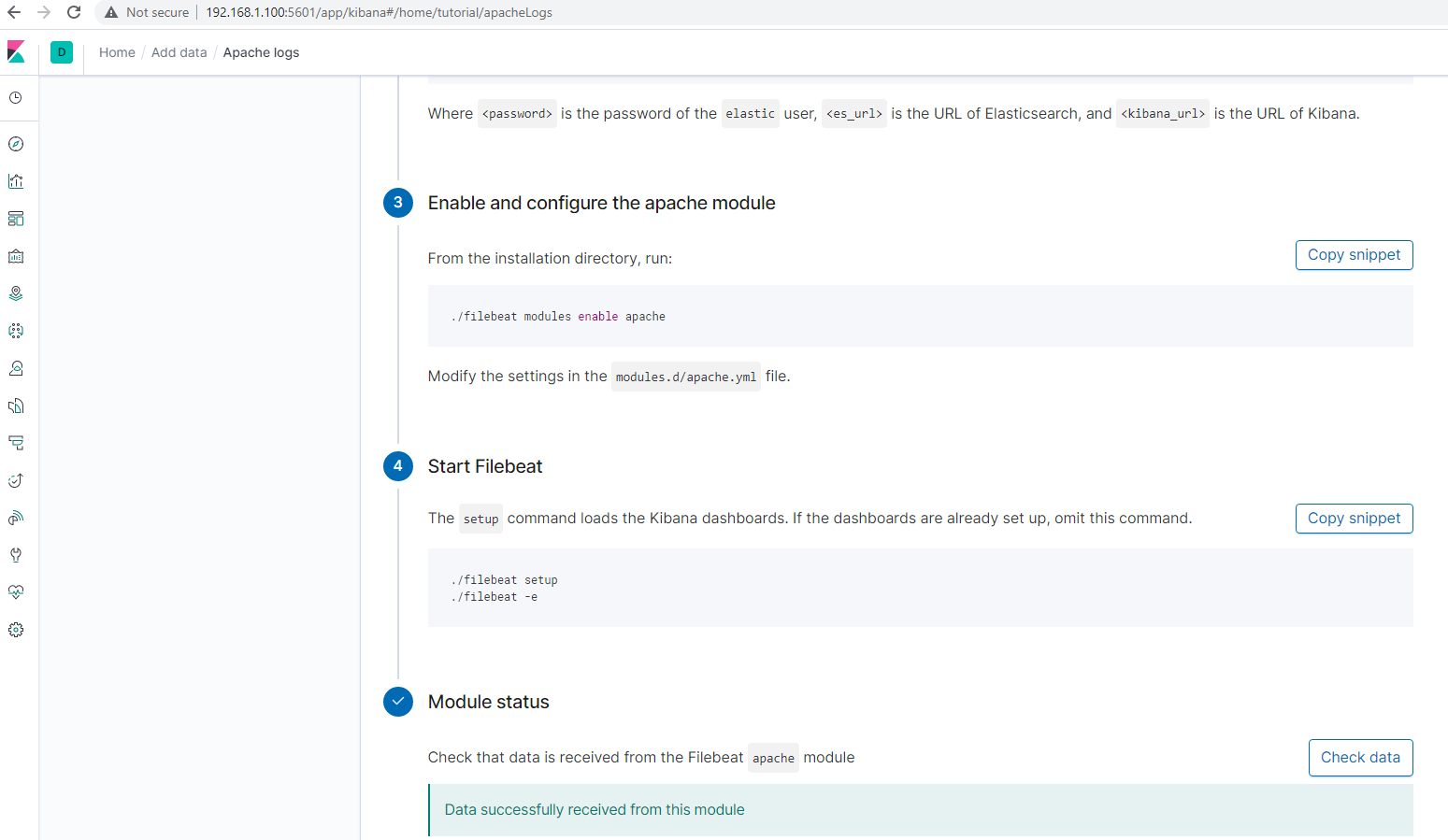
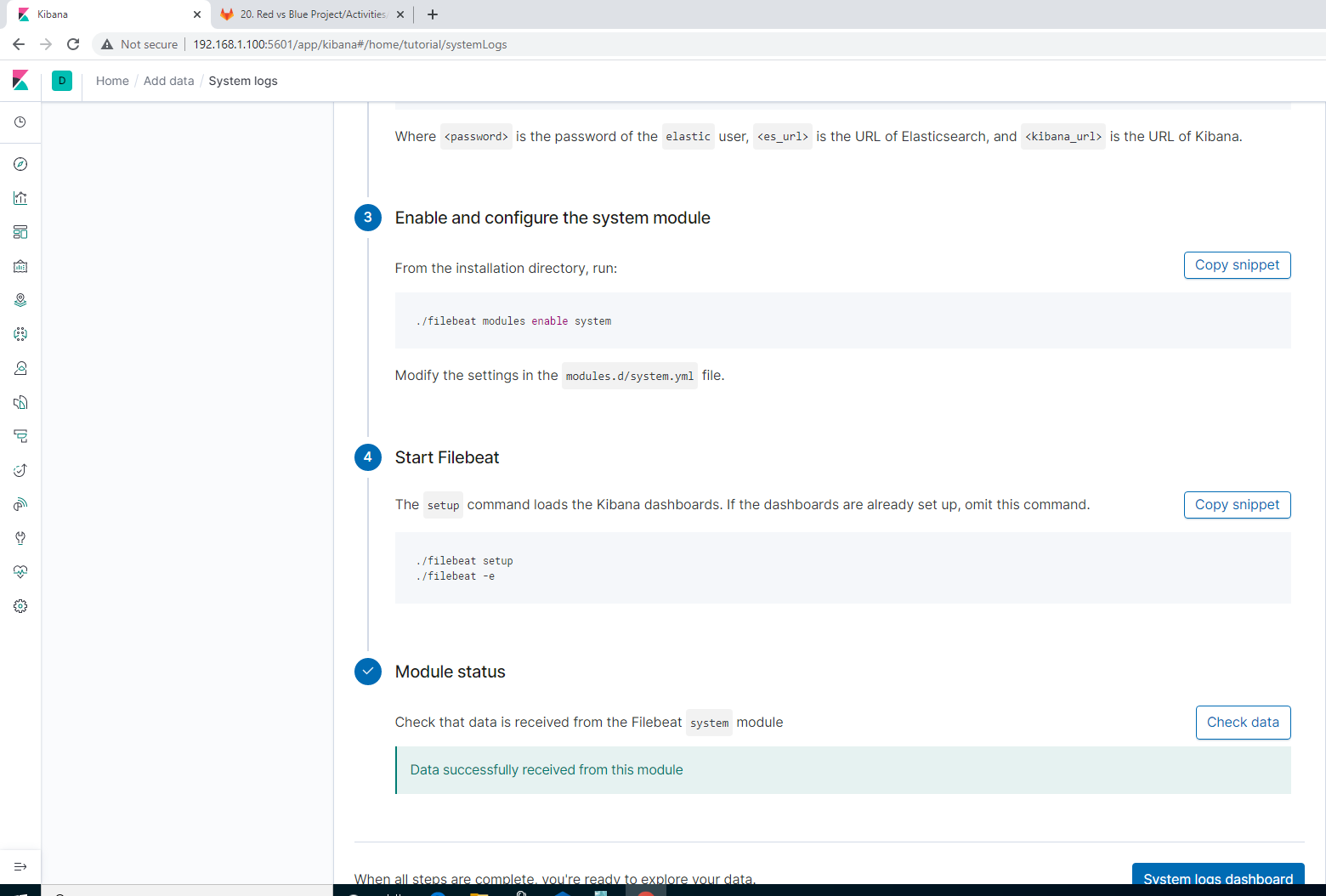
Day 2 Activity File: Incident Analysis with Kibana

Adding Kibana Log Data

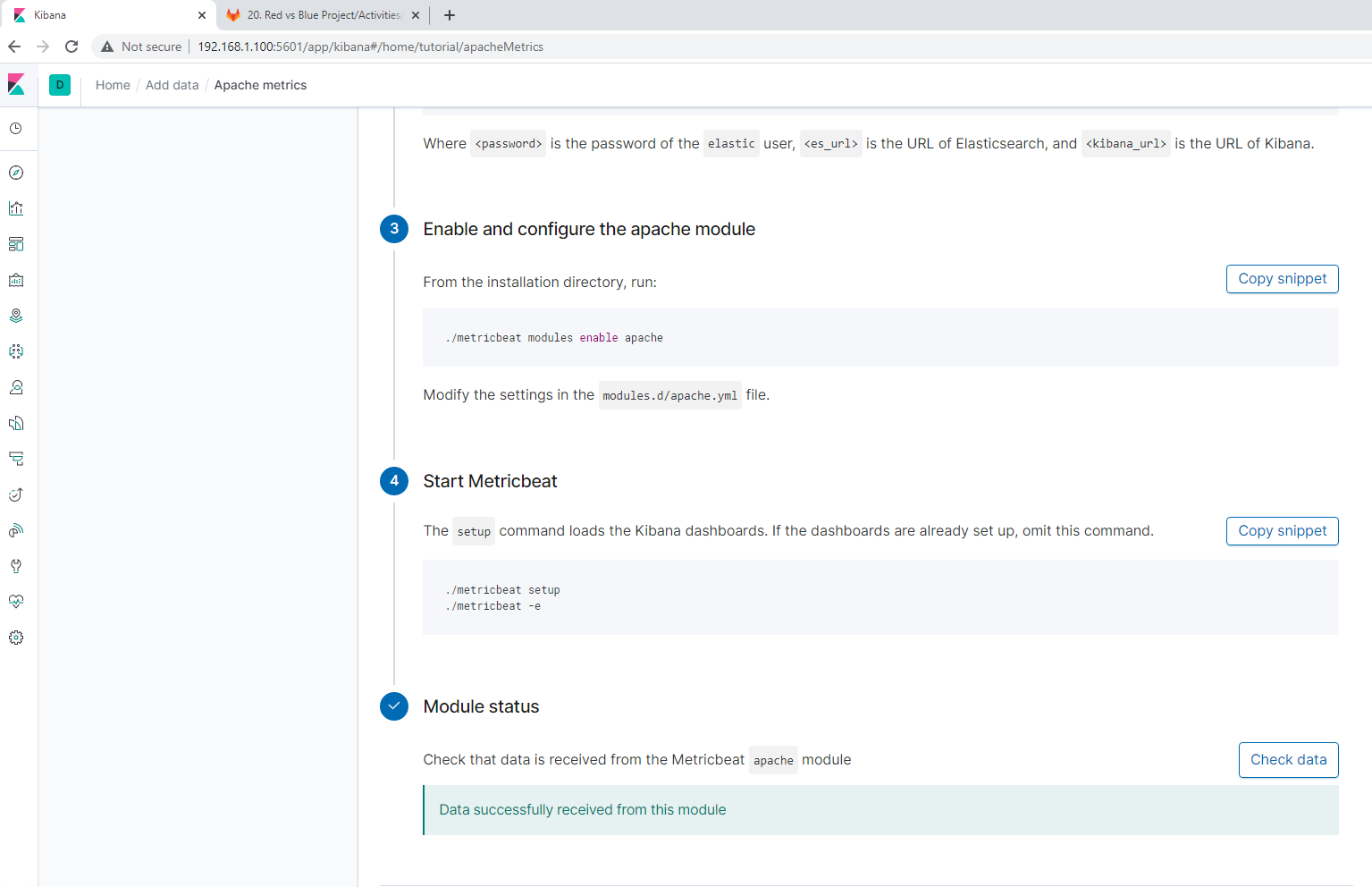
Apache Logs:



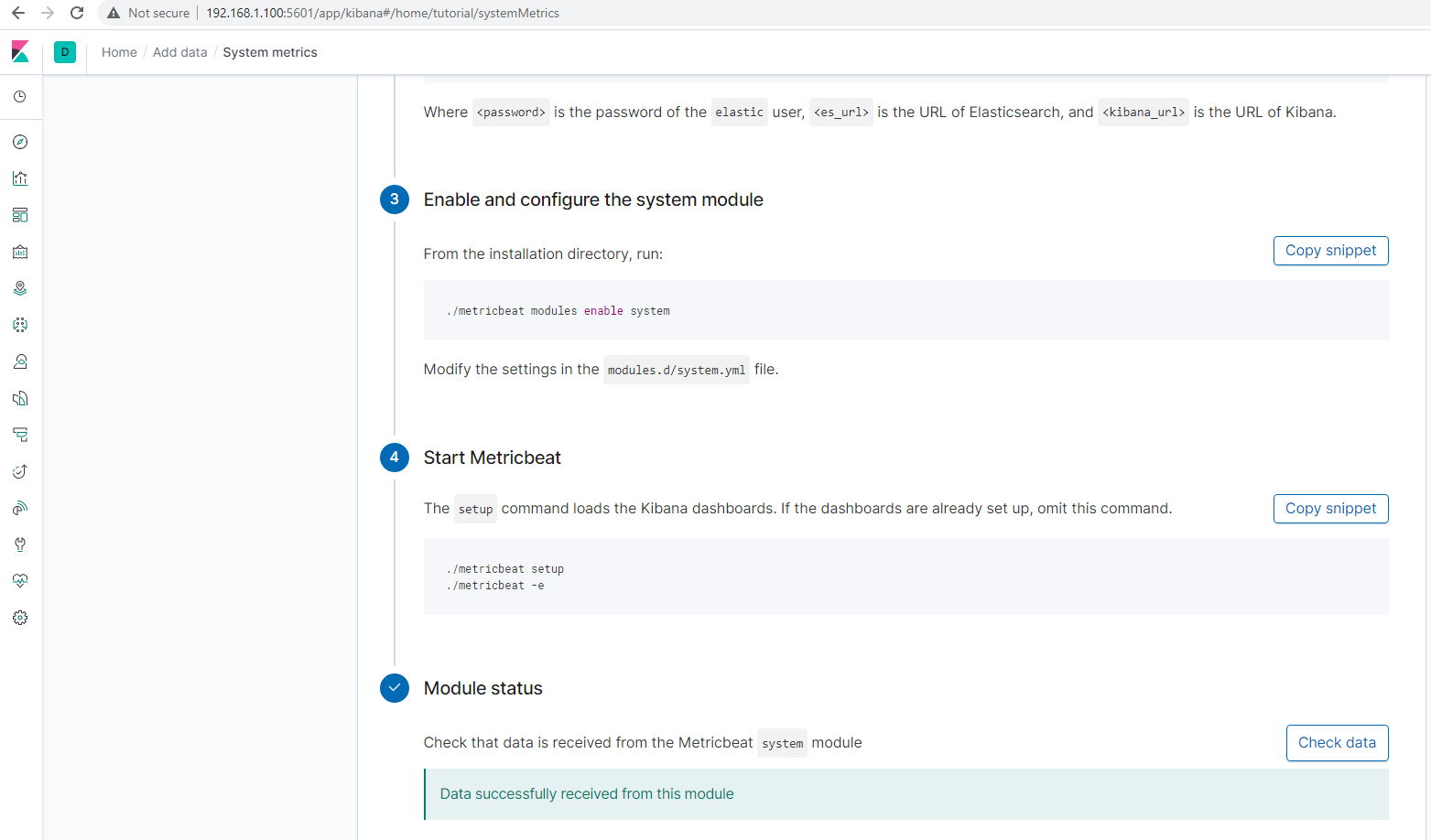
System Logs:



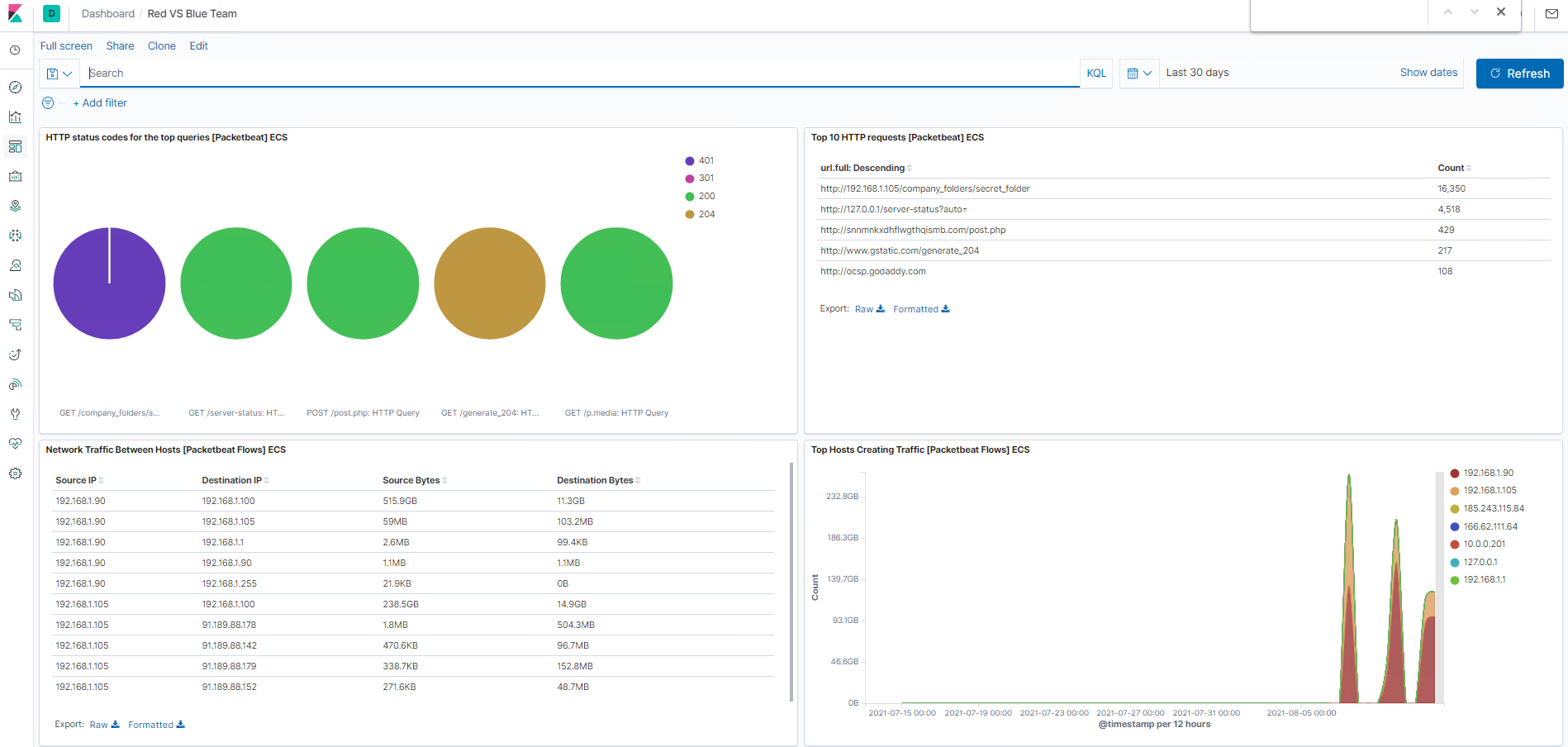
Apache Metrics:

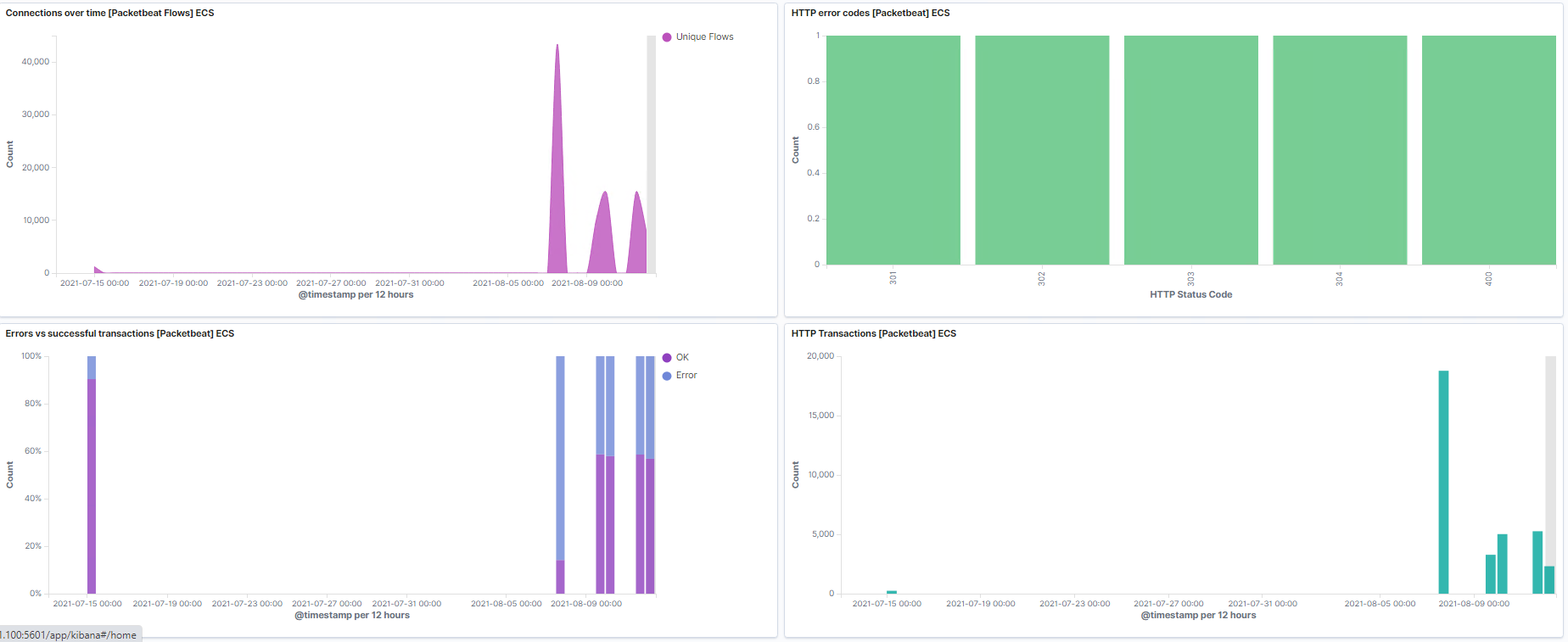


System Metrics:



DashBoards:





Source IP:

My Reference Timestamp for source.ip : 192.168.1.90



Destination Ip:192.165.1.105

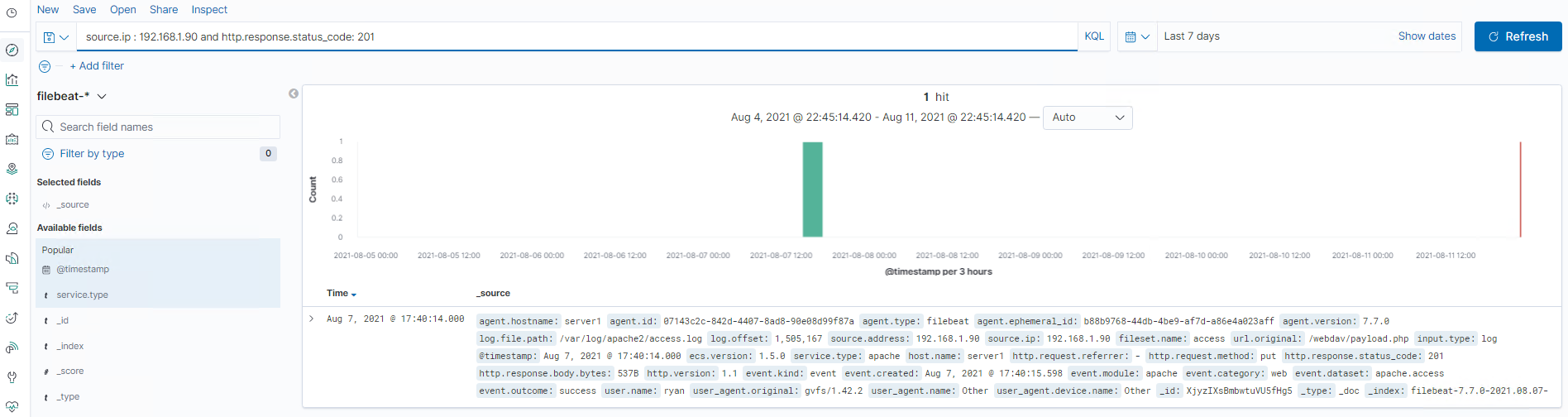


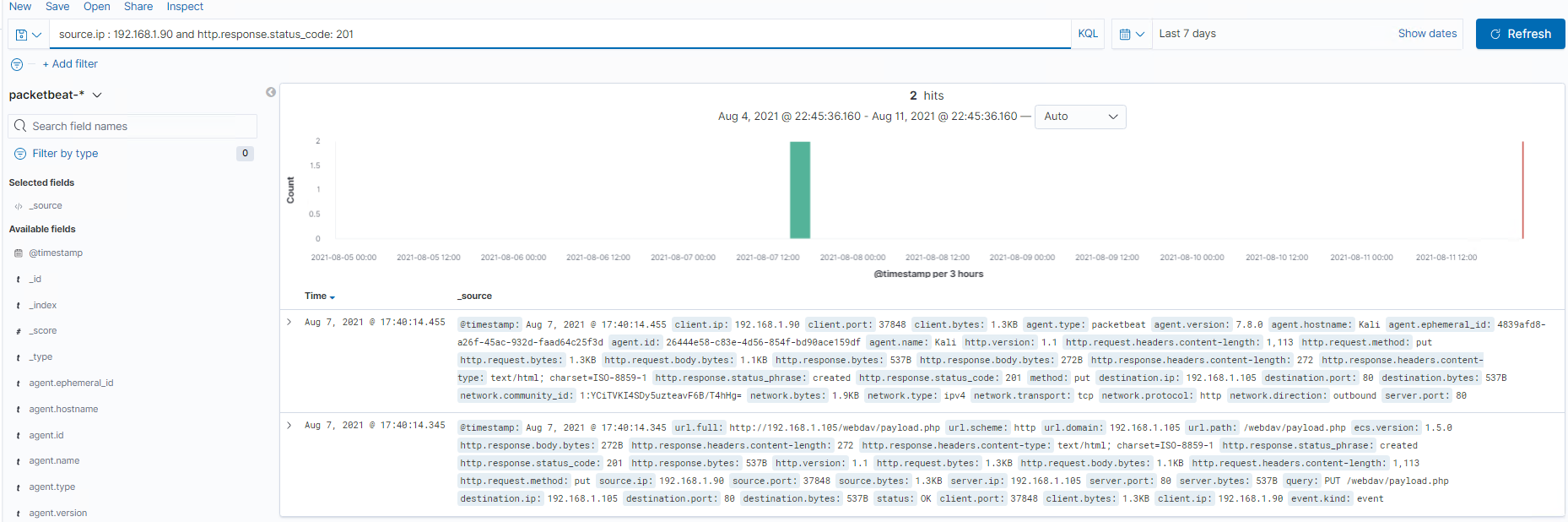
Identifying the offensive traffic.

Identifying the traffic between Red Team machine and the web machine:

Discover Search:

source.ip : 192.168.1.90 and http.response.status\_code: 201





What responses did the victim send back?

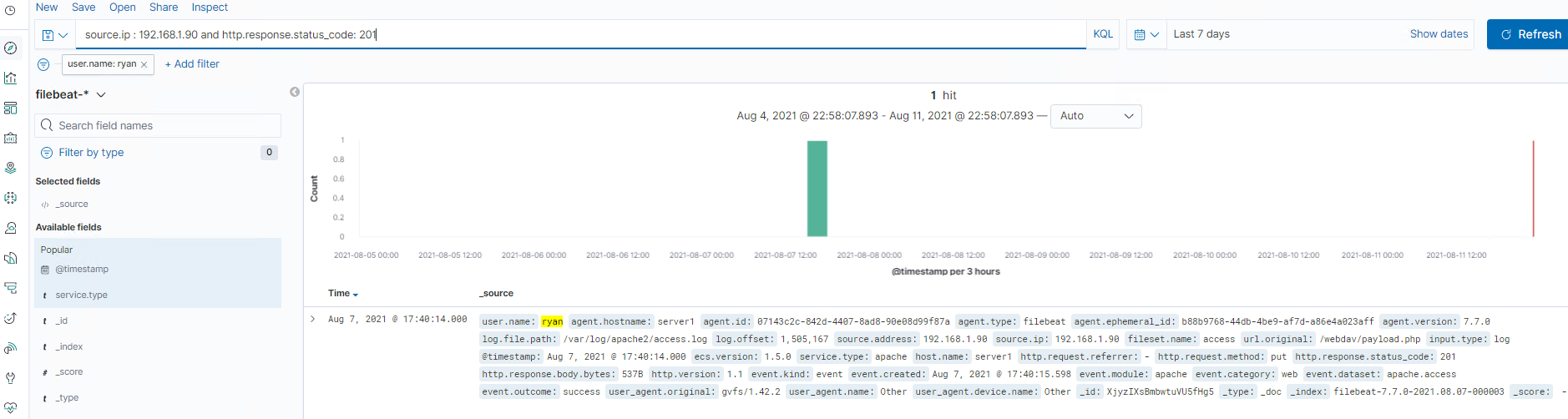
* 201 response which is a successful in both Filebeat & Packetbeat
* What data is concerning from the Blue Team perspective?

Pentesting(Kali Machine) which was used for uploading payload.php request.Cadaver is used to access WebDav

Source IP 192.168.1.90

User.name:ryan(weakened account as other has also worked on it previously)

Apache access- Filebeat201 confirmed as file was on the server-Filebeat



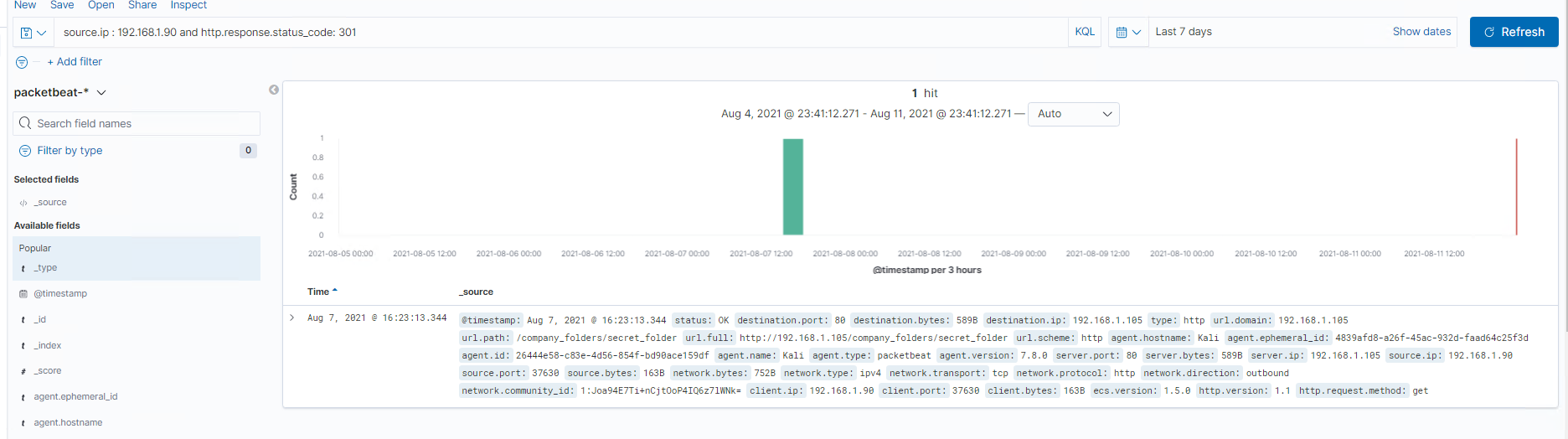
Finding the request for the hidden directory.

Discover Search:

In your attack, you found a secret folder. Let's look at that interaction between these two machines.

* How many requests were made to this directory? At what time and from which IP address(es)?

1 hits were found depending on filebeat . Get requests has happened at 16:23 from IP 192.168.1.90.

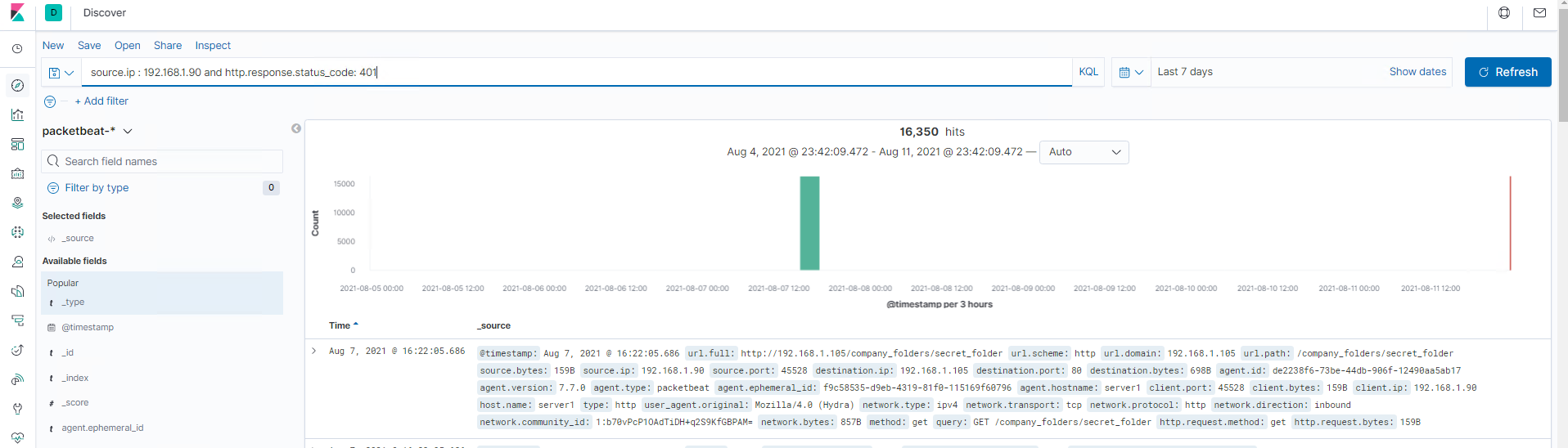


* Which files were requested? What information did they contain?

Apache server is used to process all file requests. User.name:ashton

* What kind of alarm would you set to detect this behavior in the future?

Since it was too many 16,350 hits on http.response.status.code :401 , we can base the alarm off with 401 response codes and if there are too many unauthorized attempts.

* 
* Identify at least one way to harden the vulnerable machine that would mitigate this attack

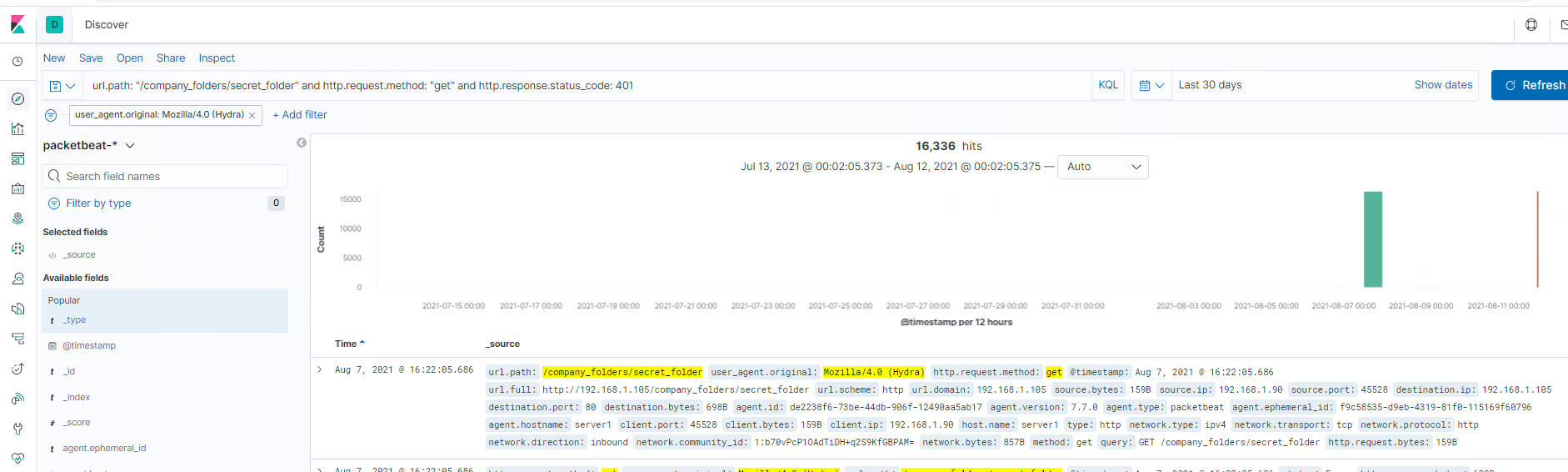
Account should be locked out after 10 subsequent attempts and also please add 2 more factor authentication which allows no access to access and Ips.

Identify the brute force attack.

* After identifying the hidden directory, you used Hydra to brute-force the target server. Answer the following questions:

Discover Search

url.path: "/company\_folders/secret\_folder" and http.request.method: "get" and http.response.status\_code: 401



* + Can you identify packets specifically from Hydra?

user\_agent.original: Mozilla/4.0 (Hydra) is the packets from Hydra

* + How many requests were made in the brute-force attack?

16,336

* + How many requests had the attacker made before discovering the correct password in this one?

16,335 or 16,338 (May be 3 lesser than successful attempts)

* + What kind of alarm would you set to detect this behavior in the future and at what threshold(s)?

We will keep an alarm with bad login attempts more than 100 and also with user.agent.original:Hydra or HTTP Response code of 401

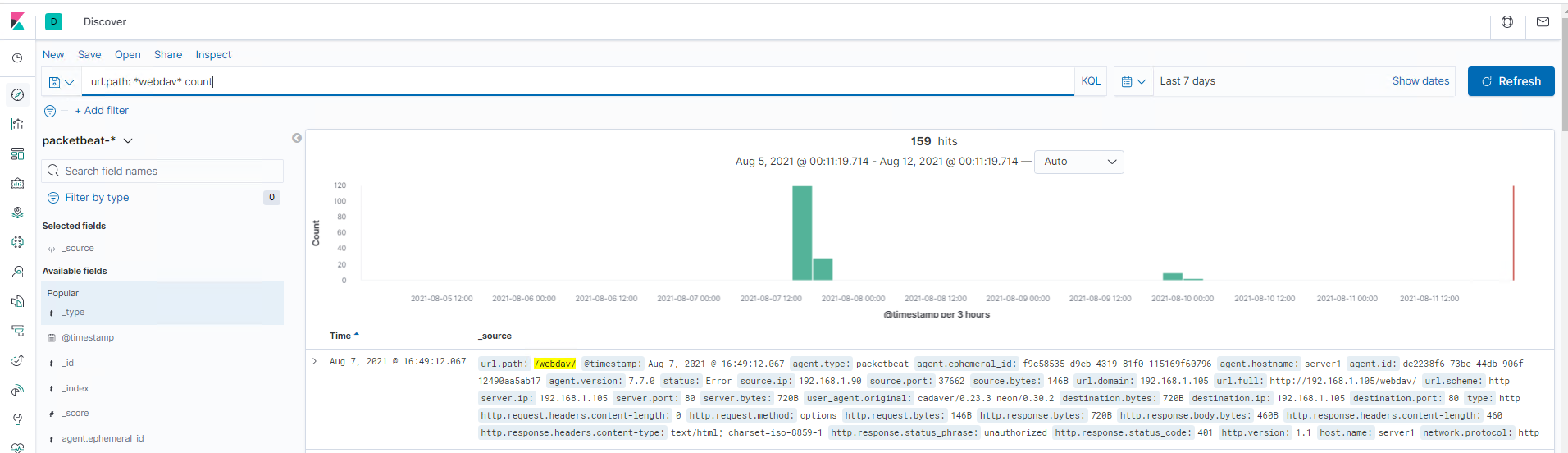
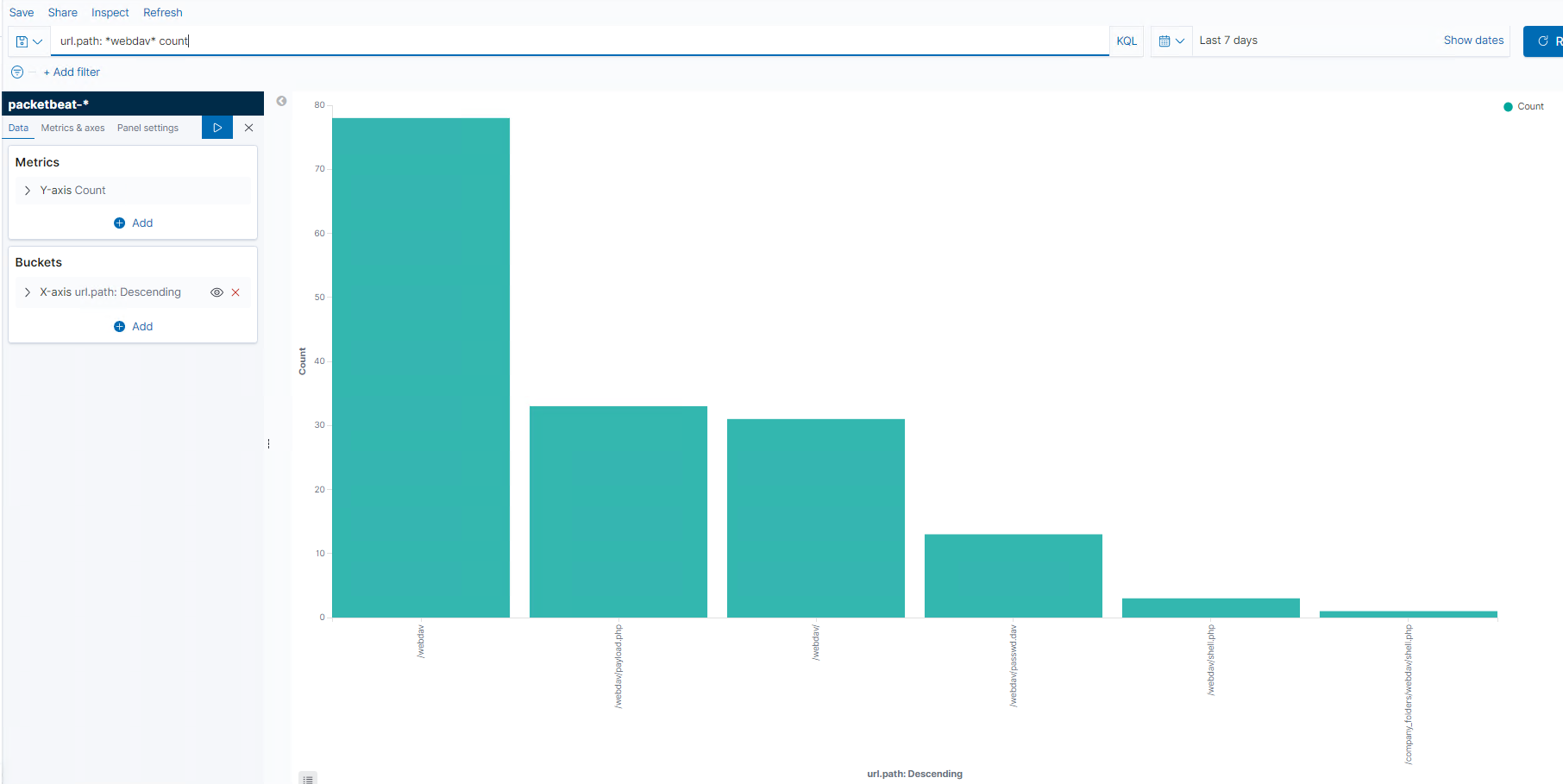
* + Identify at least one way to harden the vulnerable machine that would mitigate this attack.

We need to block with agent which is Hydra(user.agent.original) as a main fire wall rule and also lockout any bad login with over 20 consequent attempts .

* Find the WebDav connection.

Discover:

url.path: \*webdav\* count

* 
* 
  + Use your dashboard to answer the following questions:
    - How many requests were made to this directory?

159 hits(packetbeat)

* + - Which file(s) were requested?

payload.php and passwd.dav (Please see attached in visualize)

* + - What kind of alarm would you set to detect such access in the future?

Would keep Alarm when files which are created with extension like.php

* Identifying the reverse shell and meterpreter traffic.

Discover Search:

destination.ip: 192.168.1.90 and destination.port: 4444



destination.port: 80



To finish off the attack, a PHP reverse shell and a meterpreter shell session was started. Some questions to answer:

Can you identify traffic from the meterpreter session?

Yes we can identify the traffic from meterpreter from 192.1168.1.105 & 192.168.1.90 using the shell.php. and also this session isn’t clear because of encrypts traffic.

What kinds of alarms would you set to detect this behavior in the future?

Set an alarm when a new machine is connected and also new outbound traffic is noted

Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Port 4444 is a default port of meterpreter and we can set a rule to block the traffic from Port 4444