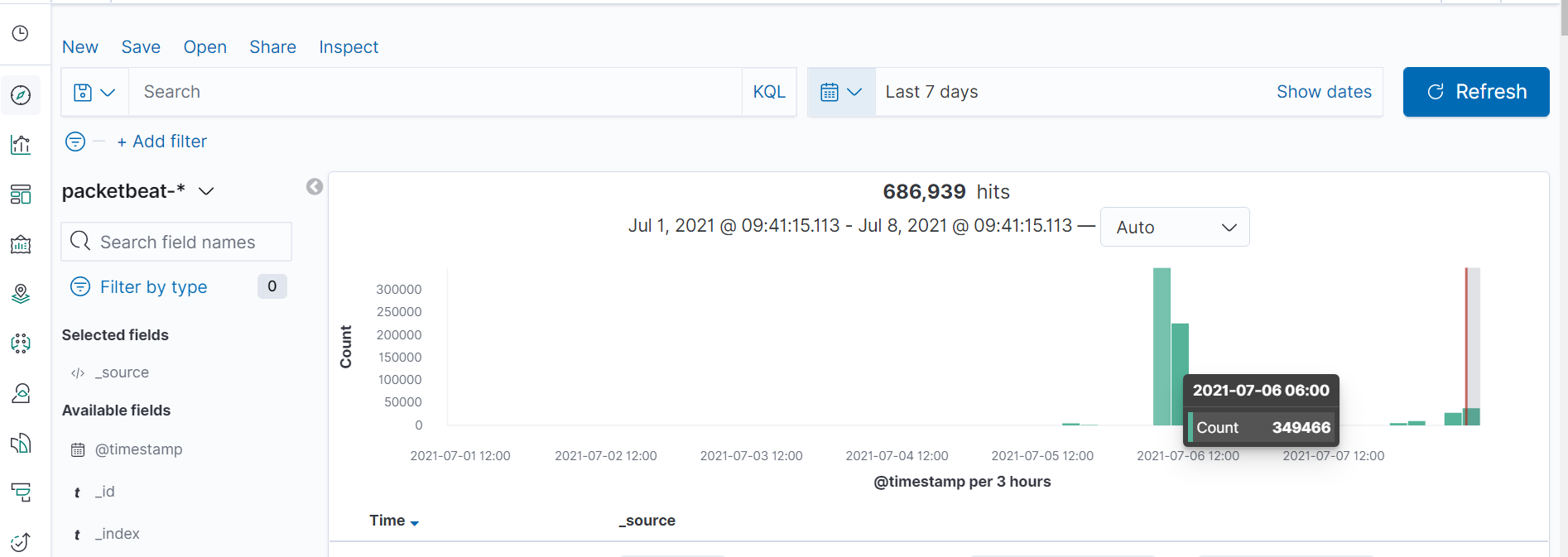
Blue Team Scenario

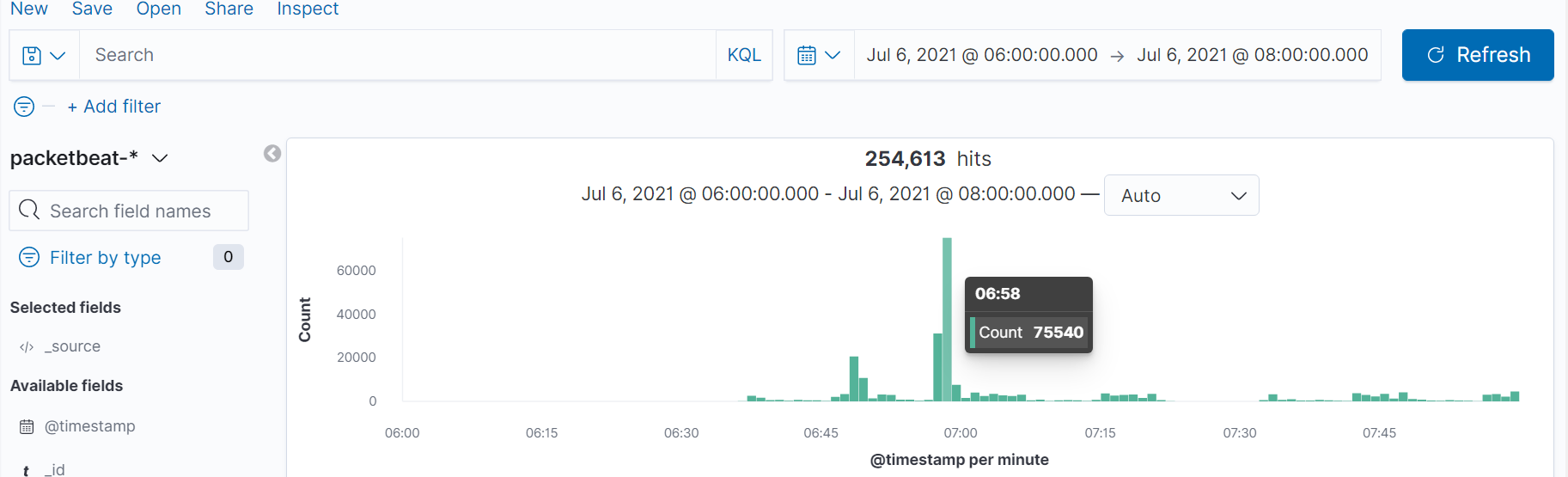
*As a SOC analyst it has been advised that the Red Team were able to conduct a full penetration test of a vulnerable virtual machine in the network. In order to defend against future attacks logs of the attack will be reviewed via Kibana interface. The logs will be used to extract hard data and visualizations for reporting. Log data will be interpreted and mitigation measures for each exploit will be explained. Developing ideas for new alerts that can improve monitoring will also be suggested.*

### **Step 1 : Identifying offensive traffic**

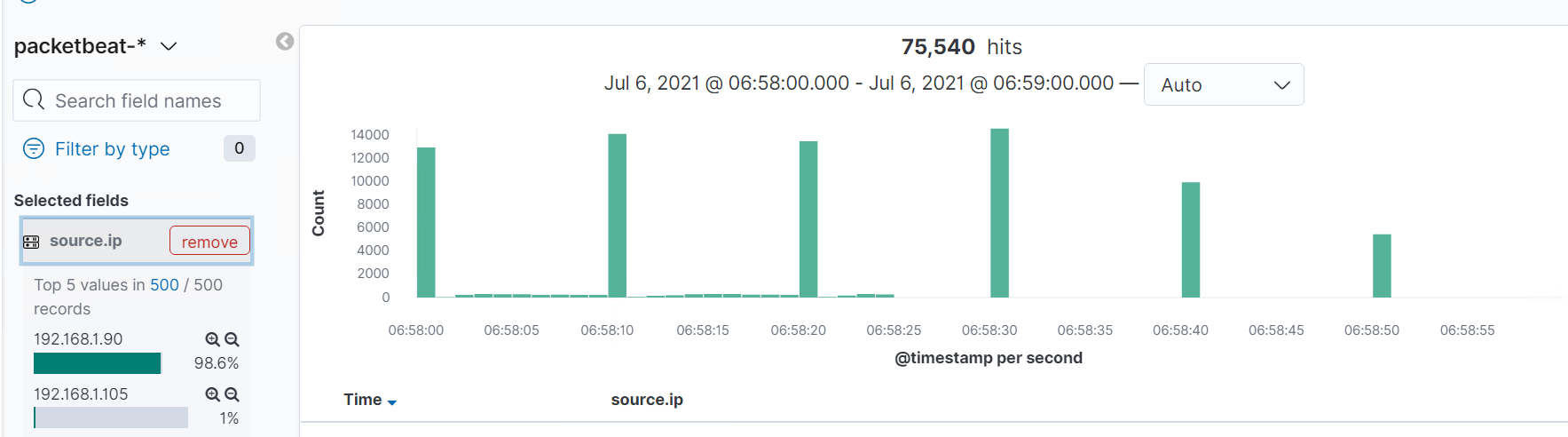
After viewing the last 7 days of traffic there is a clear spike on Tuesday 6th of July at 6AM. To analyze this further the data will be filtered from Tuesday 6th of July at 6AM to 8AM since the traffic seems to have stopped at that point.



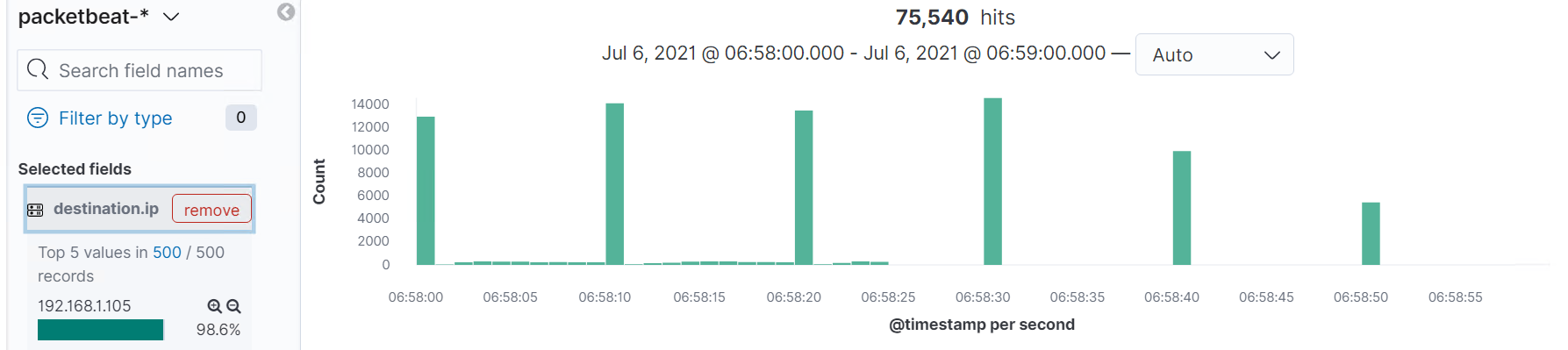
With further investigation the time of attack can be pinpointed to 6:58AM.



In order to find the attacking machine IP address the traffic was analyzed at exactly 6:58AM.Filtering via source IP field also gave inside to which node in the network was the attacking machine.

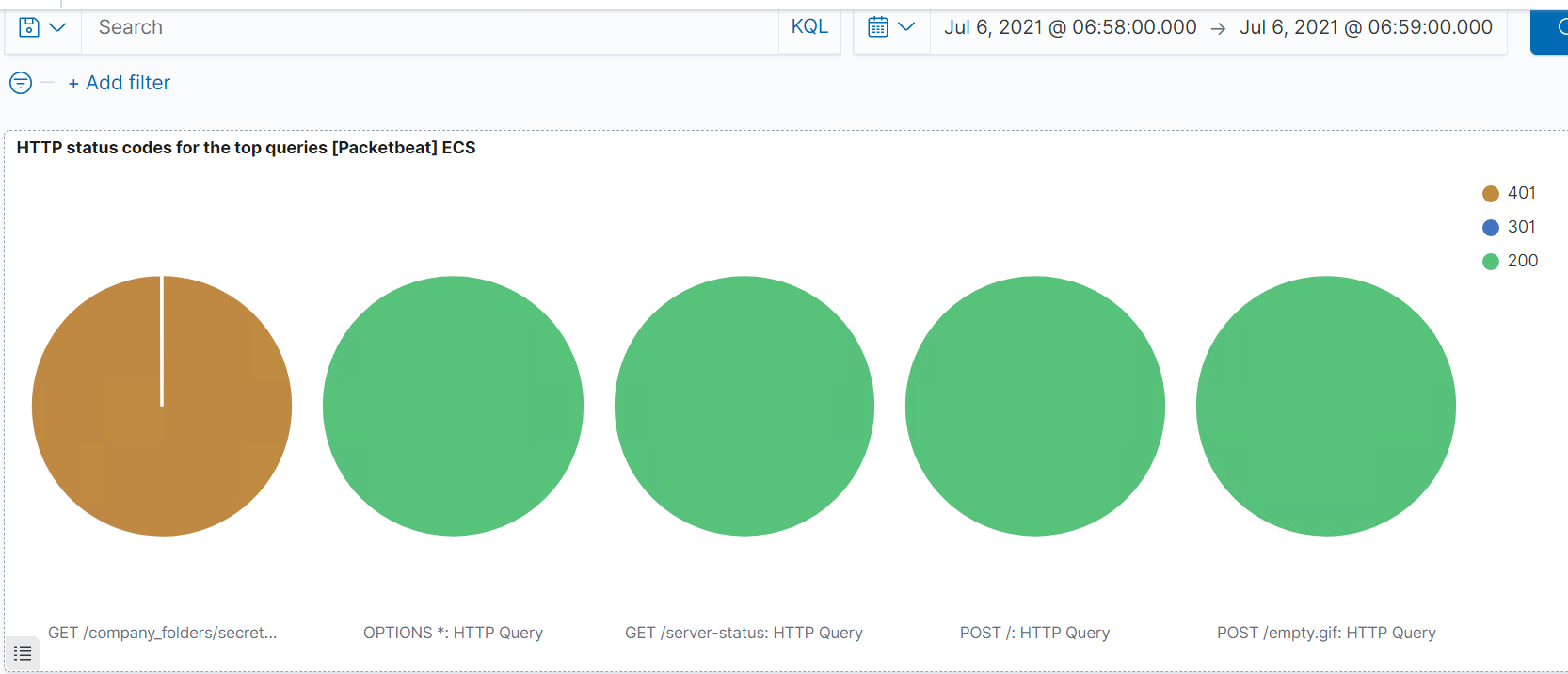


The screenshot above captures count by timestamp per second and portrays the IP address 192.168.1.90 to send roughly 12,000 counts every 10 seconds. It can be safely assumed that this is the IP of the attacking machine.   
To find the victim machine IP the source IP field will be swapped with destination IP.

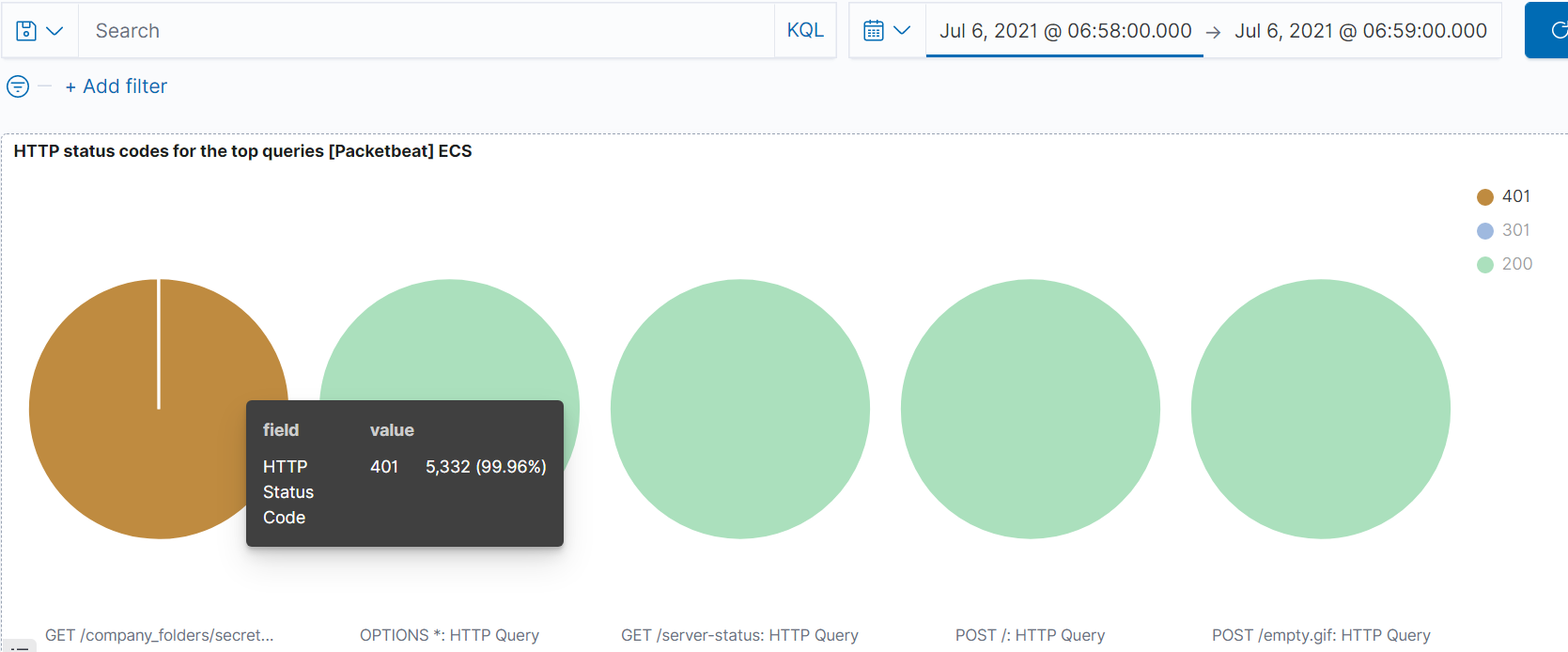


The timestamp of IP 192.168.1.105 matches with the attacking machine so therefore it is concluded that this is when the attack happened.

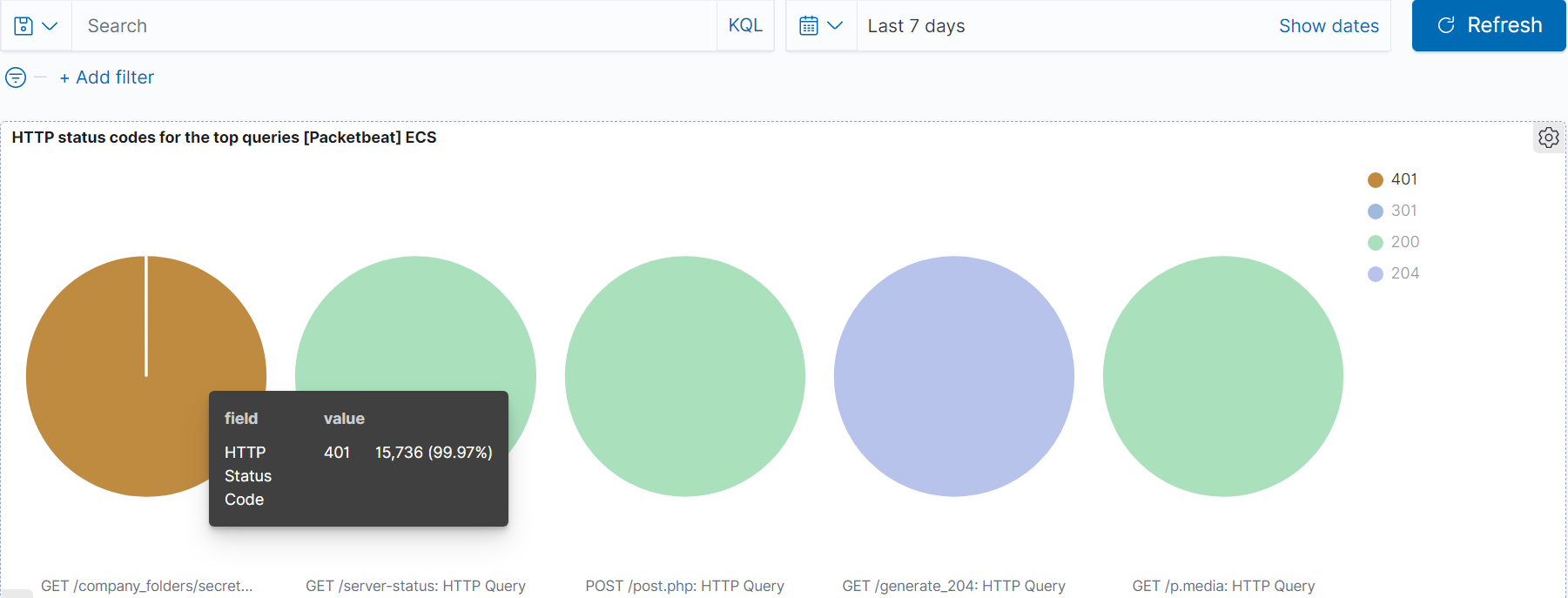
From this attack it can be seen on the dashboard below that the main responses given to the attacking machine by the victim was HTTP 200 & 401.



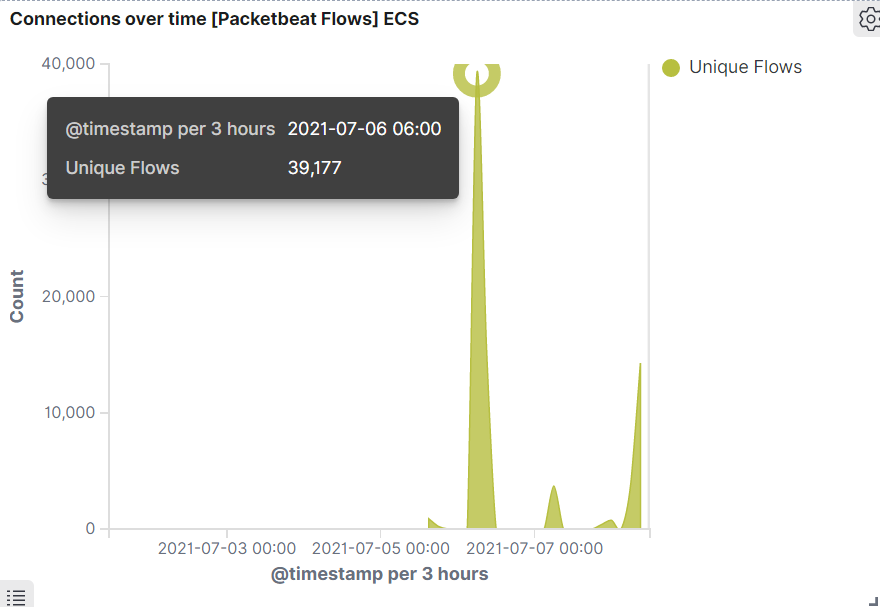
There is concern as a SOC analyst for the blue team as there were over 5000 http 401 unauthorized requests in one minute as shown in the screenshots below.



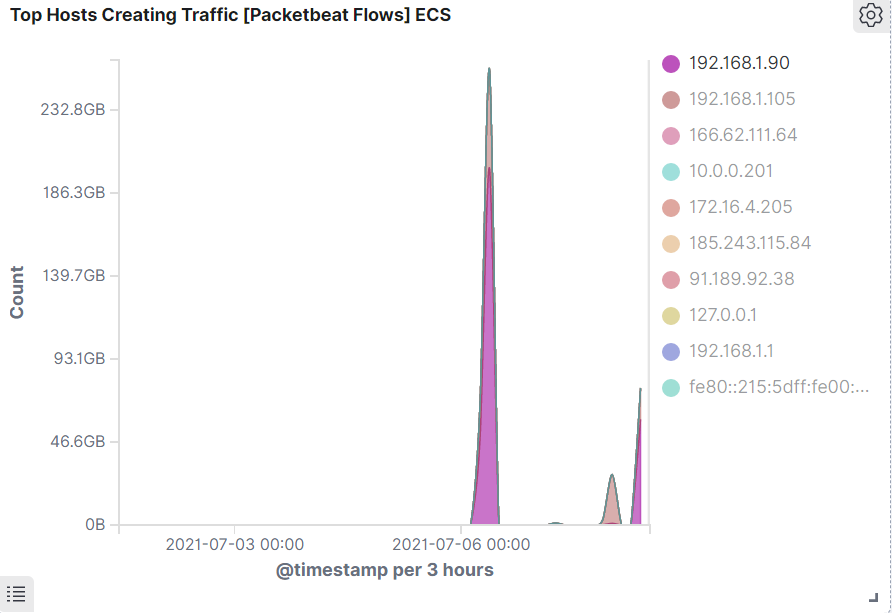
In the last week there were 15,736 http 401 errors in total. Therefore having over 5000 in the span of one minute is a concern for the blue team that should be further investigated.



Another concern for the blue team to look into is the spike of unique flow that can be seen in the Connections over time [Packetbeat Flows] ECS chart. This chart indicates that there were a total of 39,177 packets that were sent at 6AM Tuesday 6th July 2021.



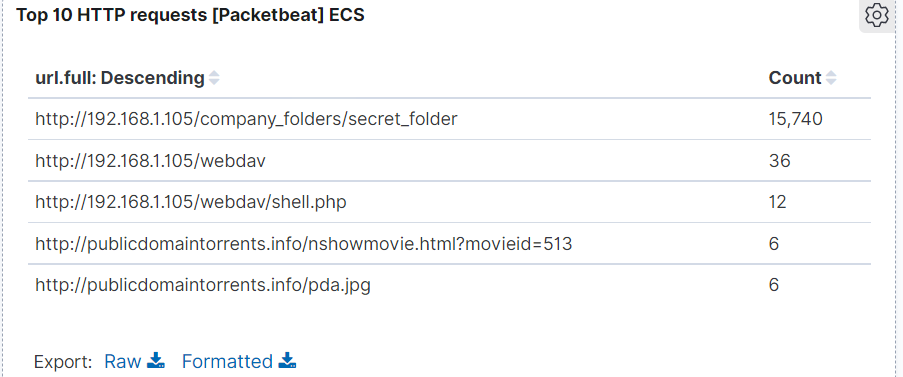
The screenshot below that comes from the chart also indicates that all the traffic has come from the attacking machine under IP 192.168.1.90. This may be an indication of a port scan.



### **Step 2 : Hidden Directory Request**

In the Top 10 HTTP requests [Packetbeat] ECS panel it can be seen that there is a high amount of requests being sent to the *192.168.1.105/company\_folder/secret\_folder* directory.

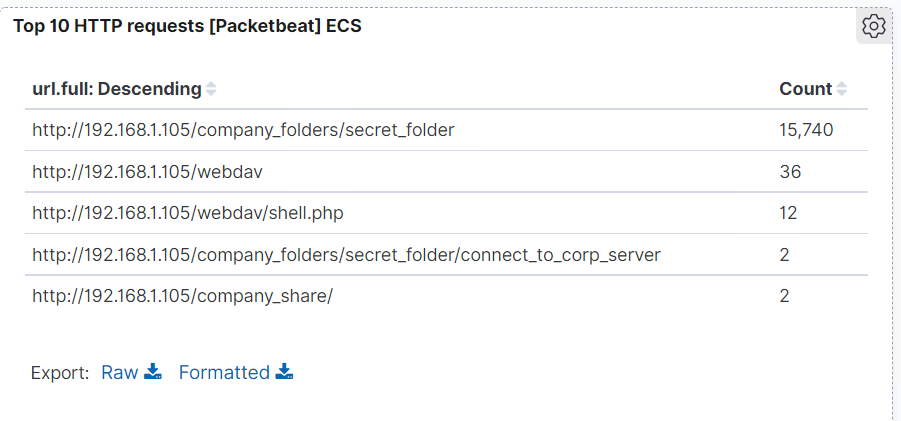
A total of 15,740 requests were made to this directory.



As there are multiple http requests sent in any general environment a filter will be used to isolate the victim IP domain



After this filter has been applied this is the results given



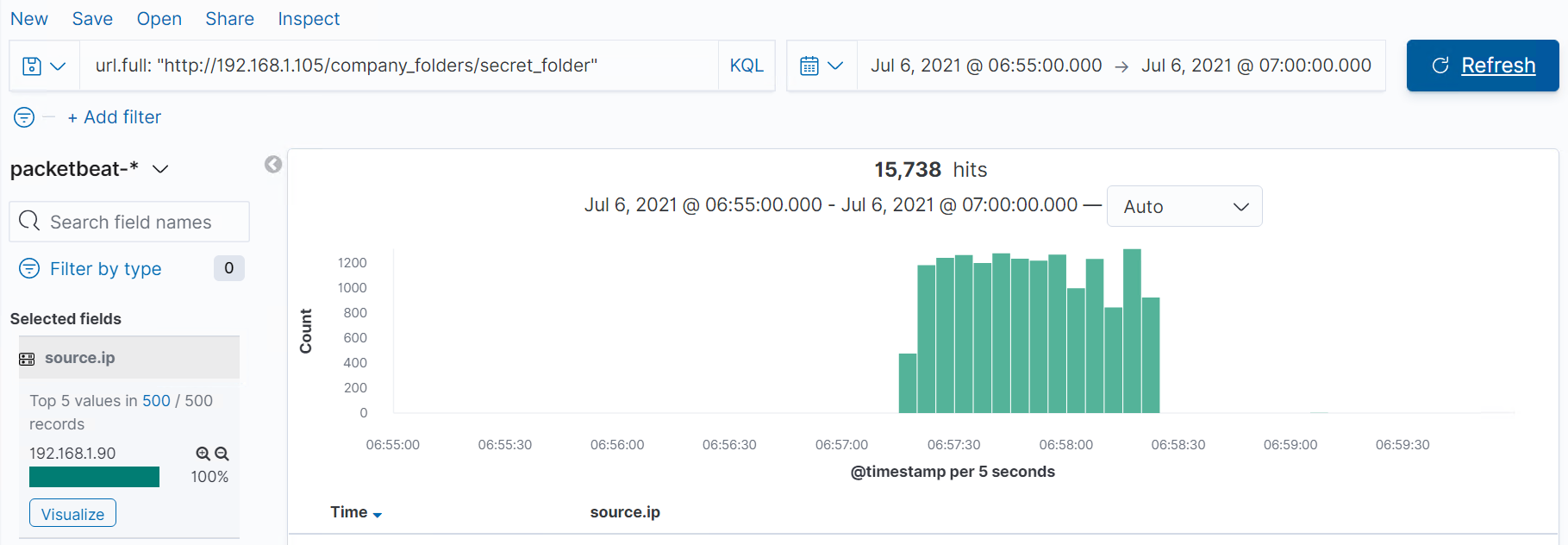
In the secret\_folder it can be seen that the connect\_to\_corp\_server has been accessed twice. This file contains information about connecting to the company server via the Hypertext Transfer Protocol Webdav. Ultimately allowing users to perform remote web content authoring operations.

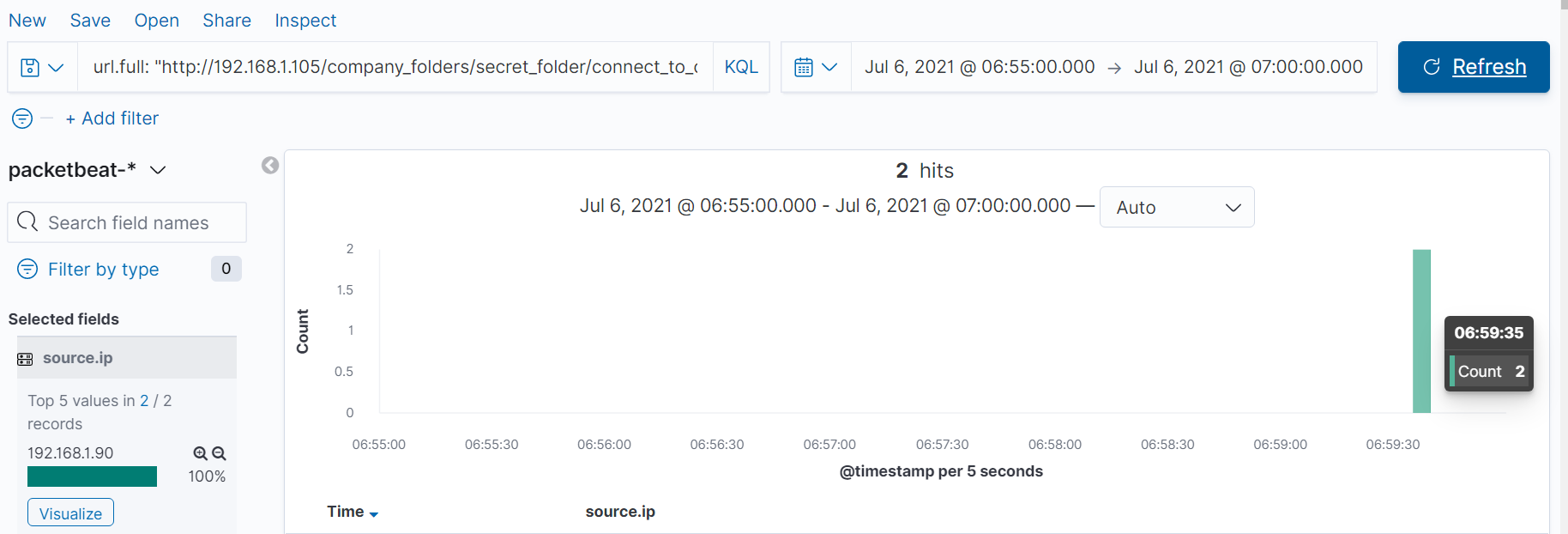
In order to find the exact moment the attack occurred there were two filters used.

url.full: “<http://192.168.1.105/company_folders/secret_folder>”

url.full: “<http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server>”

These filters only search for packets that were connecting to company\_folders/secret\_folder & company\_folders/secret\_folder/connect\_to\_corp\_server files.



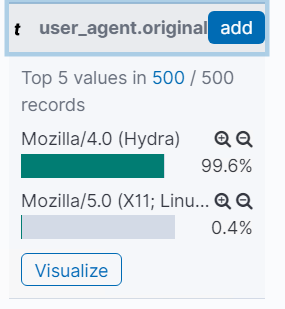


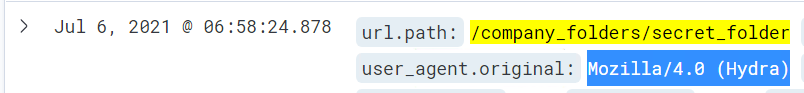
From screenshots above it is shown that an attack was happening from 6:57 AM to 6:59 AM. Within the 2 minute span there was a successful connection to the connect\_to\_corp\_server file at exactly 6:59 AM. The source IP field on the left identifies the attacking machine to be IP 192.168.1.90.

### **Step 3 : Uncovering Brute Force Attack**

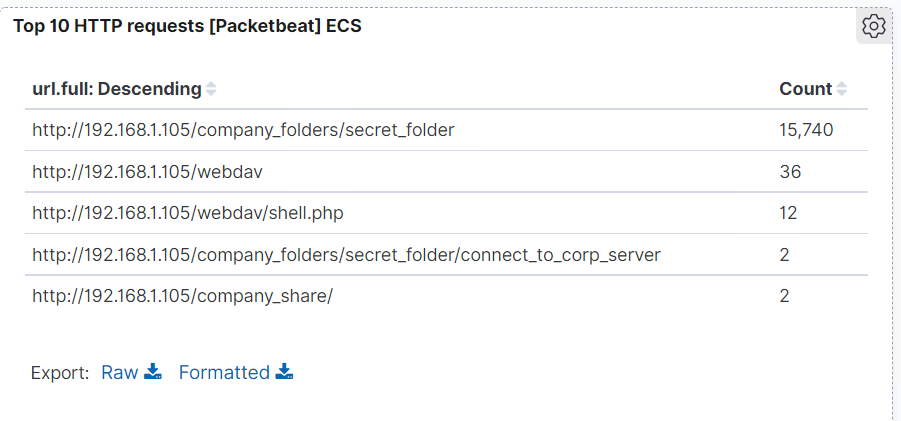
As per screenshots already provided previously it is known that there were 15,740 requests made to the company\_folders/secret\_folder directory and only 2 requests successfully made to the connect\_to\_corp\_server file. Therefore it took the attacker 15,738 password bash attempts before the password was cracked.

After consulting the user\_agent.original field it can be seen that a parallelized network login cracker named Hydra was used to brute force the attack.





### **Step 4 : Find the WebDAV Connection**

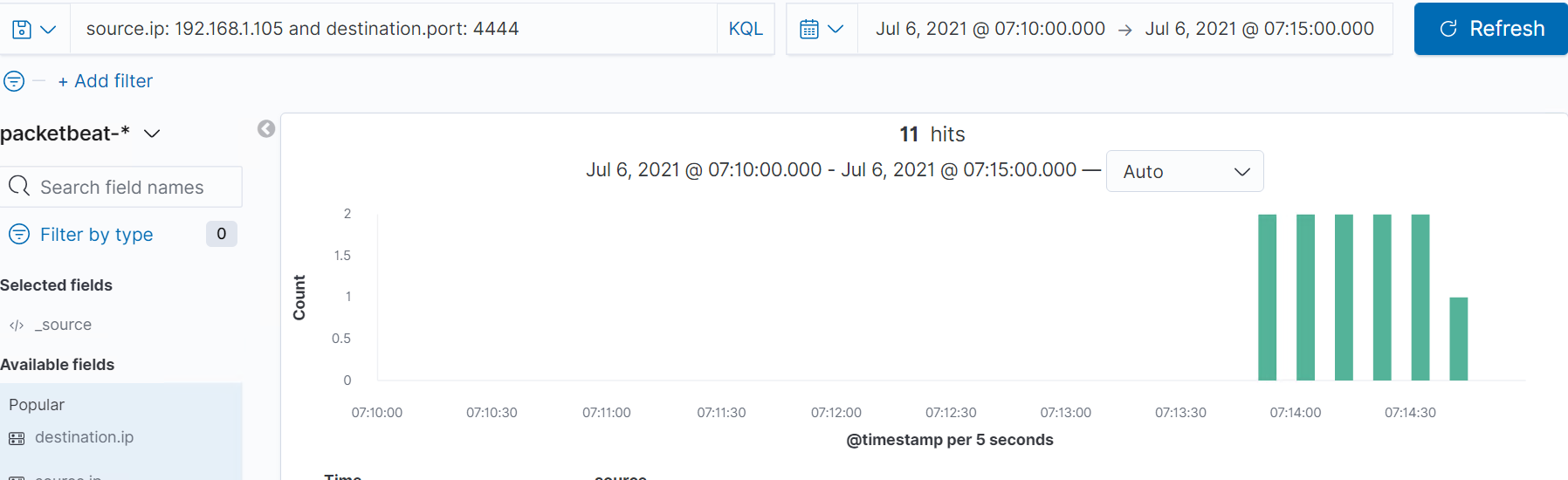


This screenshot portrays that the webdav directory was also accessed by the attacker. A total of 36 requests were made to this directory and a shell.php file was also requested. As advised by the Red Team Report once a shell was placed onto the server the attacker was able to gain root access.

### **Step 5 : Meterpreter Session**

As a default the meterpreter runs on port 4444. If the attacker didn’t change the port then it is possible to view the meterpreter traffic using the target machine as the source IP. The KQL command to search this was:

**source.ip: 192.168.1.105 and destination.port: 4444**



From the screenshot provided above a meterpreter session had occurred at 7:14 AM.

### 

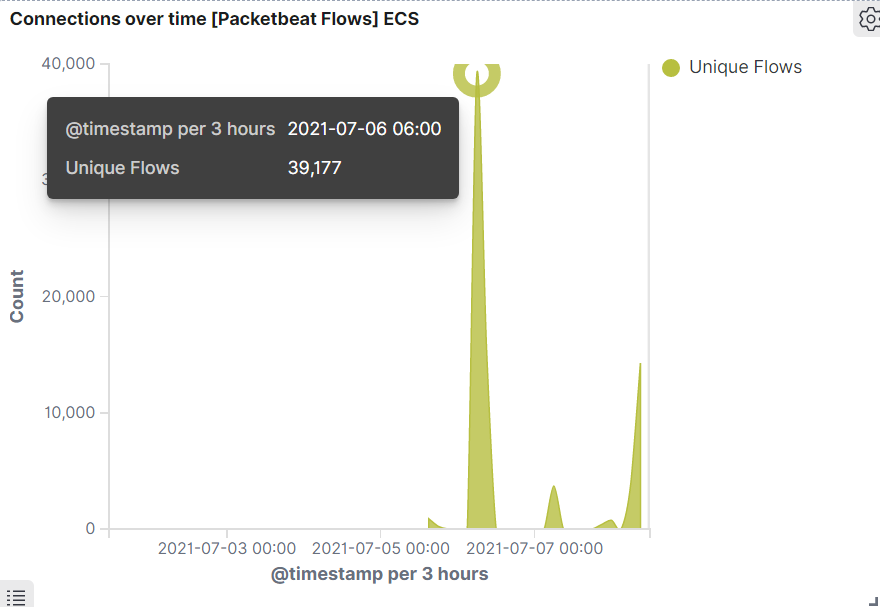
### **Mitigation Strategies & Alarms**

There are multiple mitigation strategies and alarms that can be created in order to prevent this attack from reoccurring.

**Blocking Port Scan:**

As a general rule of thumb the first action of an attack is to start by identifying host status by sending ICMP packets using ping scan. This will ping every node in the network to identify any live activity. Hence a snort alarm can be created to alert for this activity.

In theory port scans trigger a large amount of requests to different ports / IP addresses within a short timespan. These can be detected counting the number of requested ports for each source IP address. As seen with the connections over time chart provided there is a clear spike in unique flow which indicates a port scan.



Since the average flow is nearly zero the threshold will start at 100 within the hour.

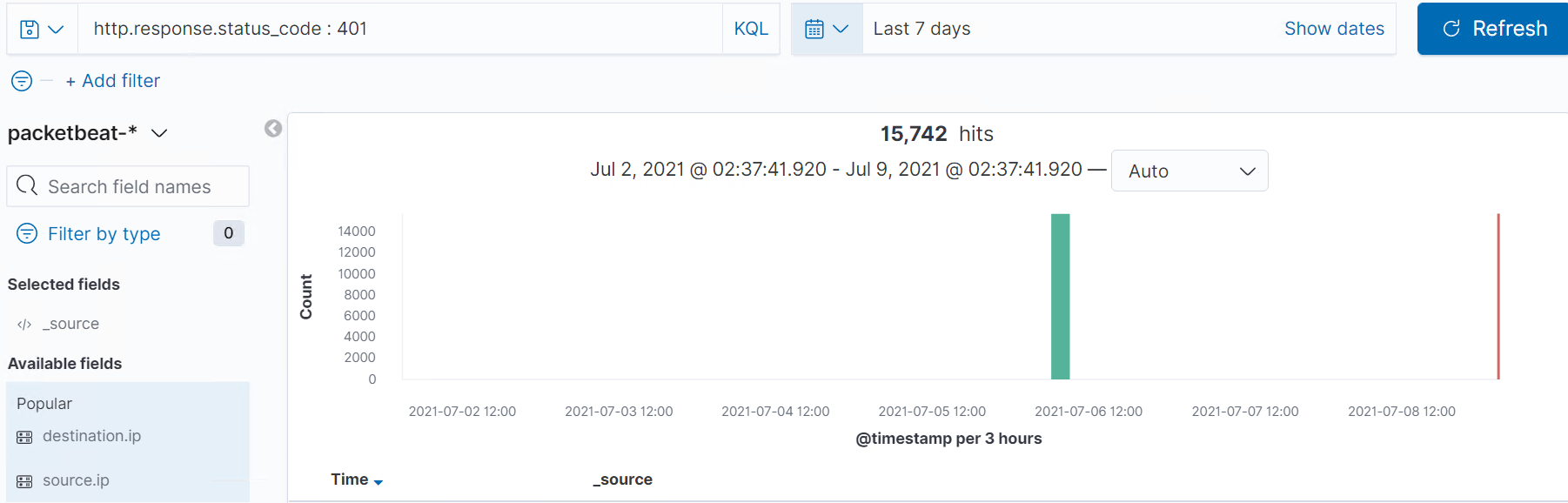
|  |  |
| --- | --- |
| **Alarm** | **System Hardening** |
| **-Create an alert that will capture the incoming traffic coming on the host (192.168.1.105) network for ICMP protocol.**  **-The snort alert will look like this below**  *alert icmp any any -> 192.168.1.105 any (msg: "NMAP ping sweep Scan)*  **-Once the unique flow threshold of 100 has been triggered send an alarm** | **-Once there is an unauthorized machine in the network that is conducting a ping sweep, lock it out of the system.**  **-Once the unique flow threshold of 100 has been triggered lock out the source IP from the system** |

**Hidden Directory Request:**

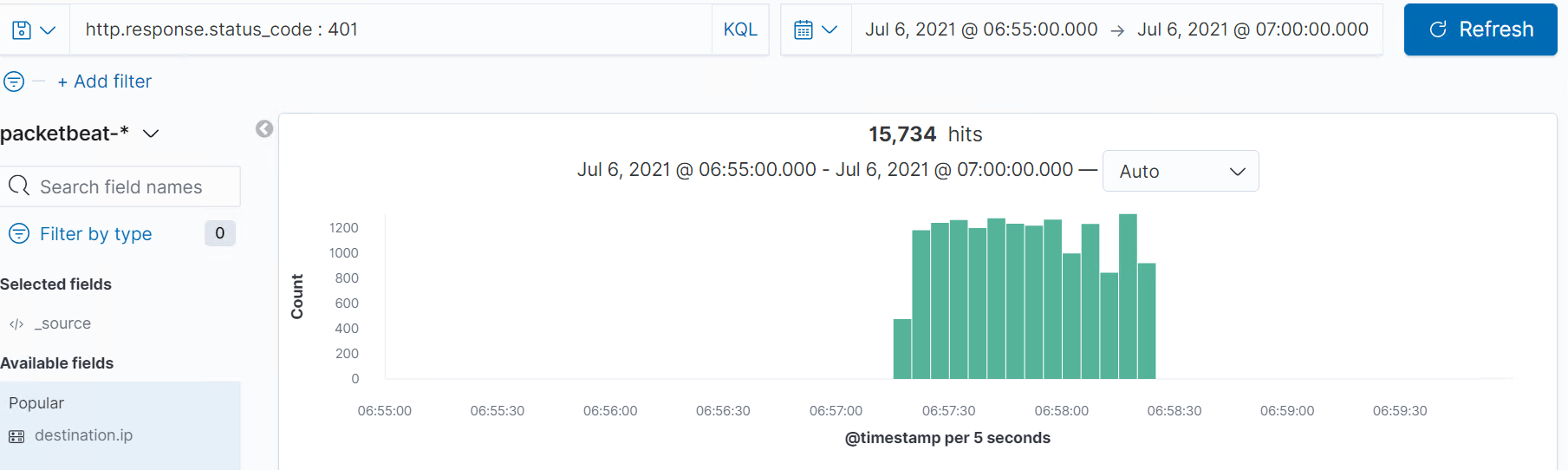
|  |  |
| --- | --- |
| **Alarm** | **System Hardening** |
| **-Create an alert for any machine that attempts to access**  [company\_folders/secret\_folder/connect\_to\_corp\_server](http://192.168.1.105/company_folders/secret_folder/connect_to_corp_server) | **-Remove file & directory from server**  **-Linux command :**  *rm -r company\_folders/secret\_folder* |

**Preventing Brute Force:**

As the Brute force creates a high amount of 401 errors, in order to evaluate a threshold the average of 401 http errors was inspected throughout the week



As this is a controlled environment there were no other 401 Unauthorized codes the baseline threshold will start at 10 an hour.



|  |  |
| --- | --- |
| **Alarm** | **System Hardening** |
| **-Create an alert if user\_agent.original value includes Hydra**  **-Create an alert if there are more than 10 HTTP 401 Unauthorized errors from any server** | **-Deny access if user\_agent.original value includes Hydra**  **-If a server returns more than 10 HTTP 401 errors then lock it out for an hour.** |

**Detecting WebDAV Connection:**

|  |  |
| --- | --- |
| **Alarm** | **System Hardening** |
| **-Create an alert if there is another machine accessing the WebDAV folder other than the original intended client** | **-Restrict access to the WebDAB folder via firewall**  **-Make the WebDAV folder inaccessible from a web interface**  **Command to close the port from a linux server :**  *sudo ufw deny 80* |

**Identifying Reverse Shell Uploads:**

|  |  |
| --- | --- |
| **Alarm** | **System Hardening** |
| **-Create an alert for any traffic coming from port 444 since this is the default meterpreter port**  **-Set an alert for any .php file uploaded to the server** | **-Restrict users from uploading files from a web interface** |

### 