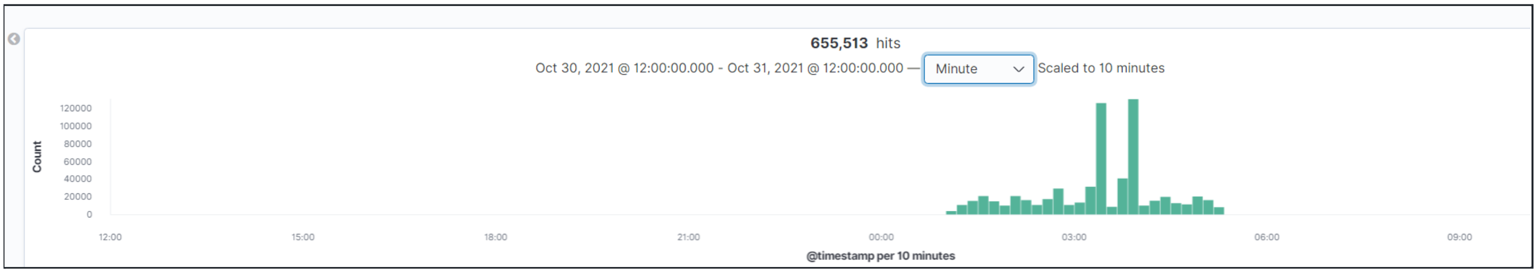
After creating your dashboard and becoming familiar with the search syntax, use these tools to answer the questions below:

1. **Identify the offensive traffic.**

    - **Identify the traffic between your machine and the web machine:**
        - **When did the interaction occur?**

Attacking machine (Kali): 192.168.1.90 Target (Victim) machine: 192.168.1.105

Traffic between the 2 machines occurred on October 31st between the time of 3:30 A.M. and 4:30 A.M.. There was a massive spike in traffic around 3:50 A.M for about 5 minutes. This is probably when the Brute-Force-Attack occurred.



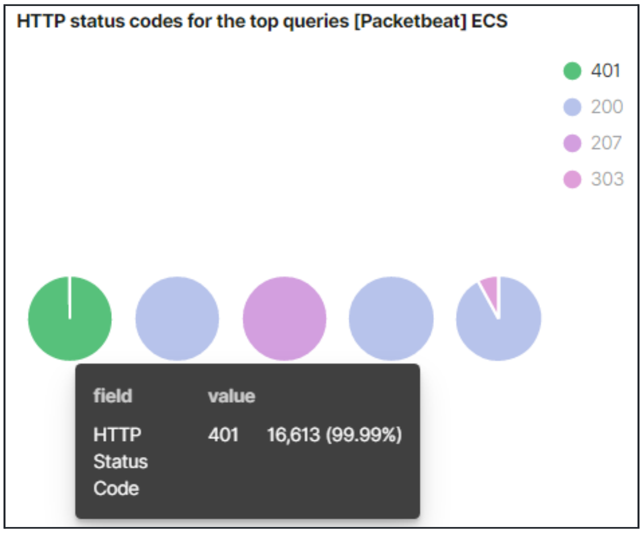
655,513 hits

Oct 30, 2021 @ 12:00:00.000 - Oct 31, 2021 @ 12:00:00.000 — Minute Scaled to 10 minutes

- **What responses did the victim send back?**

The table below shows the top response codes sent back to the Kali machine.

There were over 16,000 401 (unauthorized) reponse codes sent back to the Kali machine, just during the Brute-Force-Attack.



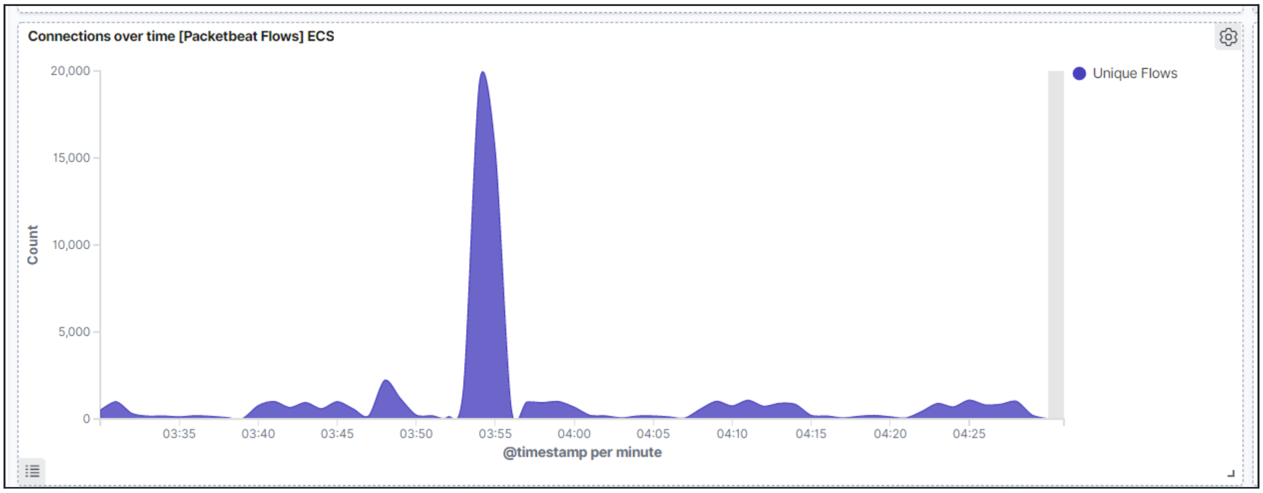
HTTP status codes for the top queries [Packetbeat] ECS

field	value	
HTTP Status Code	401	16,613 (99.99%)

- **What data is concerning from the Blue Team perspective?**

More than 15,000 unauthorized response codes were issued within a matter of minutes. During this time, only 2 200 (authorized) response codes were issued. This means that out of at least 15,000 attempted logins, only 2 were successful.

The graph below shows the massive spike in web traffic that happen within minutes. This graph alone would signal suspicious behavior.

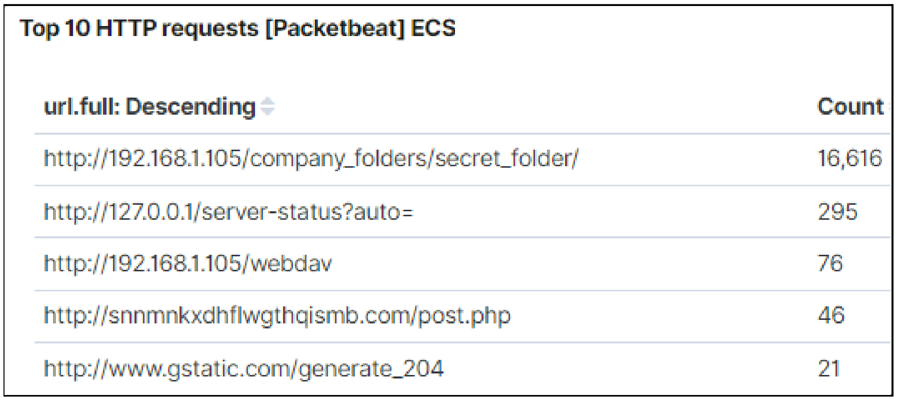


2. **Find the request for the hidden directory.**

- **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)?**

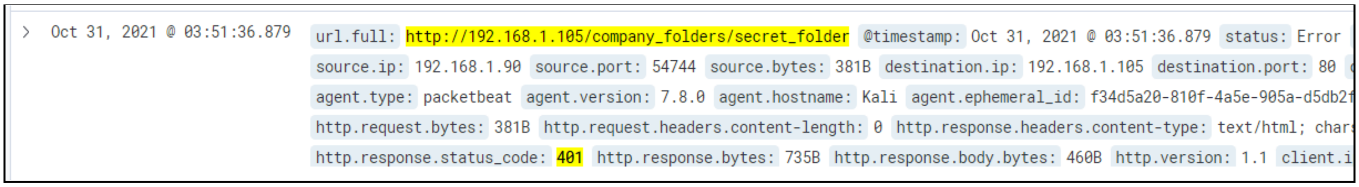
Before the Brute-Force-Attack, I was able to spot 4 http requests made towards the hidden directory. This took place around 3:51 from 192.168.1.90 (Kali).

A few minutes later, I was able to spot more than 16,000 requests made to that directory within 3 minutes. All these requests came from the 192.168.1.90 ip address.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder/	16,616
http://127.0.0.1/server-status?auto=	295
http://192.168.1.105/webdav	76
http://snnmnkxdhflwgthqismb.com/post.php	46
http://www.gstatic.com/generate_204	21



```
>  Oct 31, 2021 @ 03:51:36.879   url.full: http://192.168.1.105/company_folders/secret_folder @timestamp: Oct 31, 2021 @ 03:51:36.879 status: Error
                                 source.ip: 192.168.1.90 source.port: 54744 source.bytes: 381B destination.ip: 192.168.1.105 destination.port: 80 
                                 agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: f34d5a20-810f-4a5e-905a-d5db2f
                                 http.request.bytes: 381B http.request.headers.content-length: 0 http.response.headers.content-type: text/html; char
                                 http.response.status_code: 401 http.response.bytes: 735B http.response.body.bytes: 460B http.version: 1.1 client.i
```

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

The webpage/folder requested: http://192.168.1.105/company_folders/secret_folder/

The hidden directory contained a link: “connect_to_corp_server”. The link contained instructions on how to connect to the company’s webdav server.

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

  You can also set an alert to notify us when an unknown ip address requests access to the hidden directory.

  To alert us of a potential Brute-Force Attack, you can set an alarm that would notify us when a specific number of failed login attempts occur within a 5 minute timespan.

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

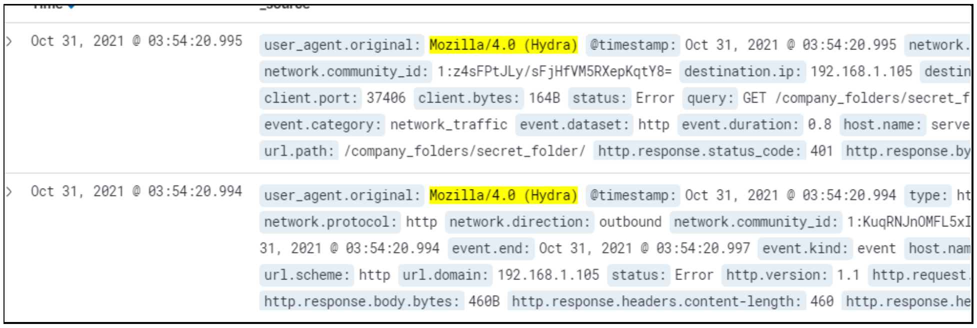
  Only allow ip addresses from known (employee) ip addresses to access the hidden directory.

  The directories/webpages open to the public that revealed information about how to get to the hidden directory. As a company, we need to make sure no confidential information can be seen by anyone on the internet. It would be best to keep this file and/or directory off this server.

3. **Identify the brute force attack.**

- **After identifying the hidden directory, you used Hydra to brute-force the target server. Answer the following questions:**
  - **Can you identify packets specifically from Hydra?**

    Yes. “Hydra” can be found in the user_agent.original field.



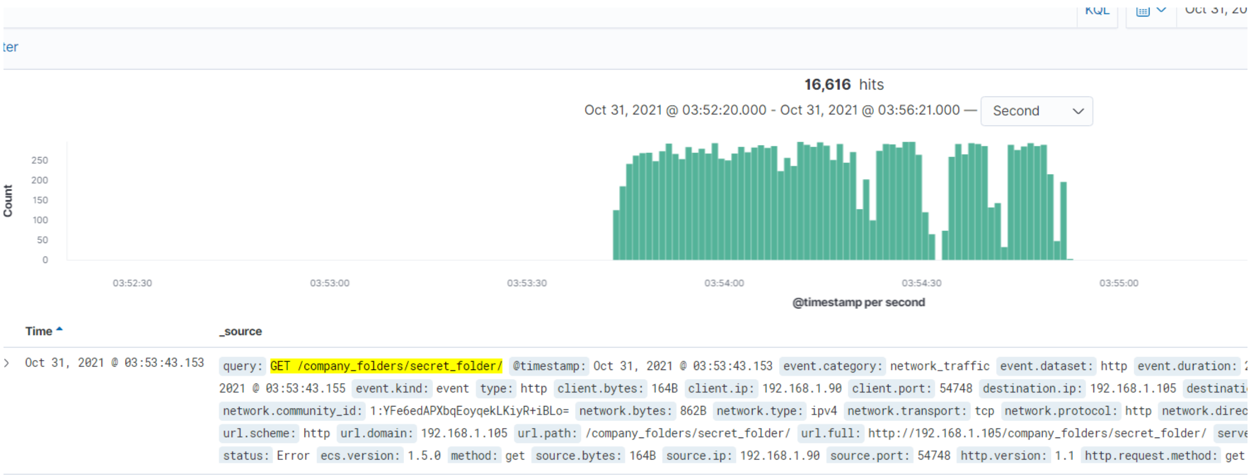
```
Time                         _source
> Oct 31, 2021 @ 03:54:20.995  user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 31, 2021 @ 03:54:20.995 network.
                               network.community_id: 1:z4sFPtJLy/sFjHfVM5RXepKqtY8= destination.ip: 192.168.1.105 destin
                               client.port: 37406 client.bytes: 164B status: Error query: GET /company_folders/secret_f
                               event.category: network_traffic event.dataset: http event.duration: 0.8 host.name: serve
                               url.path: /company_folders/secret_folder/ http.response.status_code: 401 http.response.by

> Oct 31, 2021 @ 03:54:20.994  user_agent.original: Mozilla/4.0 (Hydra) @timestamp: Oct 31, 2021 @ 03:54:20.994 type: ht
                               network.protocol: http network.direction: outbound network.community_id: 1:KuqRNJnOMFL5xI
                               31, 2021 @ 03:54:20.994 event.end: Oct 31, 2021 @ 03:54:20.997 event.kind: event host.nam
                               url.scheme: http url.domain: 192.168.1.105 status: Error http.version: 1.1 http.request.
                               http.response.body.bytes: 460B http.response.headers.content-length: 460 http.response.he
```

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

    Between 3:53 A.M. and 3:55 A.M., a total of 16,616 http get requests were made from 192.168.1.90 (Kali) to 192.168.1.105 (Victim).

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

  A total of 16,614 requests were made before the correct password was found.



```
> Oct 31, 2021 @ 03:53:43.153  query: GET /company_folders/secret_folder/ @timestamp: Oct 31, 2021 @ 03:53:43.153 event.category: network_traffic event.dataset: http event.duration:
                               2021 @ 03:53:43.155 event.kind: event type: http client.bytes: 164B client.ip: 192.168.1.90 client.port: 54748 destination.ip: 192.168.1.105 destinati
                               network.community_id: 1:YFe6edAPXbqEoyqekLKiyR+iBLo= network.bytes: 862B network.type: ipv4 network.transport: tcp network.protocol: http network.direc
                               url.scheme: http url.domain: 192.168.1.105 url.path: /company_folders/secret_folder/ url.full: http://192.168.1.105/company_folders/secret_folder/ serve
                               status: Error ecs.version: 1.5.0 method: get source.bytes: 164B source.ip: 192.168.1.90 source.port: 54748 http.version: 1.1 http.request.method: get
```

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

  To alert us of a potential Brute-Force Attack, you can set an alarm that would notify us when a specific number of 401 error codes are generated within a 5 minute timespan. For example, more than 50 failed login attempts within a 5 minute timespan would alert us.

  We can also set an alarm to notify us when “Hydra” is located in the user_agent.original field.

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

  Set a maximum of 5 failed login attempts before the user’s account is locked. The account can be locked out for a specific duration of time, or the user would be required to contact the company’s IT department to reset password and unlock the account.

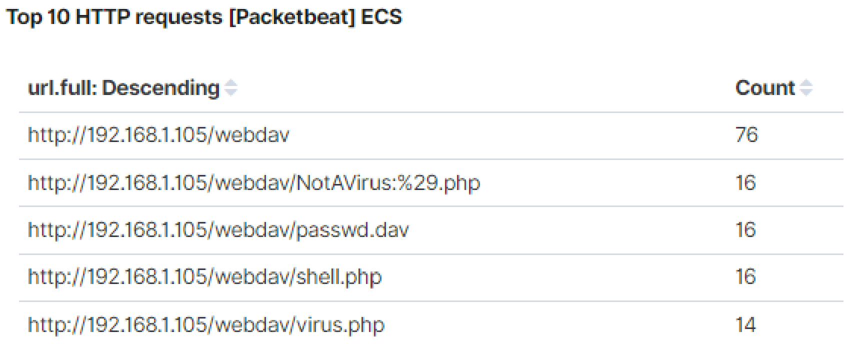
  You can also set a limit of http requests allowed from one ip address within a specific time span. For example: One ip address can not send more than 50 https requests to the hidden directory within a 5 minute time span.

4. **Find the WebDav connection.**

   ○ **Use your dashboard to answer the following questions:**

   - **How many requests were made to this directory?**

     During the time the web traffic spiked, I was able to generate a graph that shows the requests made to the WebDav directory:



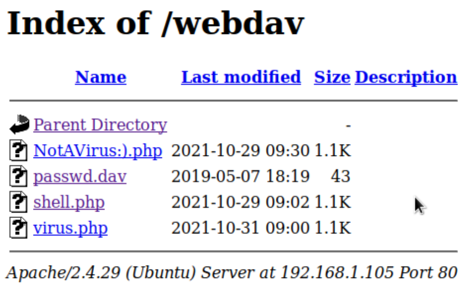
Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/webdav	76
http://192.168.1.105/webdav/NotAVirus:%29.php	16
http://192.168.1.105/webdav/passwd.dav	16
http://192.168.1.105/webdav/shell.php	16
http://192.168.1.105/webdav/virus.php	14

- **Which file(s) were requested?**

  The WebDav directory was requested for access by a user. At that time, there was 1 filed (passwd.dav) located in the directory.

  side note.. I uploaded a few different files to the directory during the attack (Day 1) just to mess around a bit.

  Index of /webdav

Name	Last modified	Size	Description
Parent Directory		-	
NotAVirus:).php	2021-10-29 09:30	1.1K	
passwd.dav	2019-05-07 18:19	43	
shell.php	2021-10-29 09:02	1.1K	
virus.php	2021-10-31 09:00	1.1K	

  Apache/2.4.29 (Ubuntu) Server at 192.168.1.105 Port 80

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

  Similar to the alarm designed for question #2, I would recommend an alarm to notify us when an unknown ip address requests/gains access to the WebDav connection. We would need to keep a record of all employee ip addresses in order to create this alarm. Any ip address not on record that is requesting access would send us a notification.

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

  1) Limit this sort of access on a company's webpage.

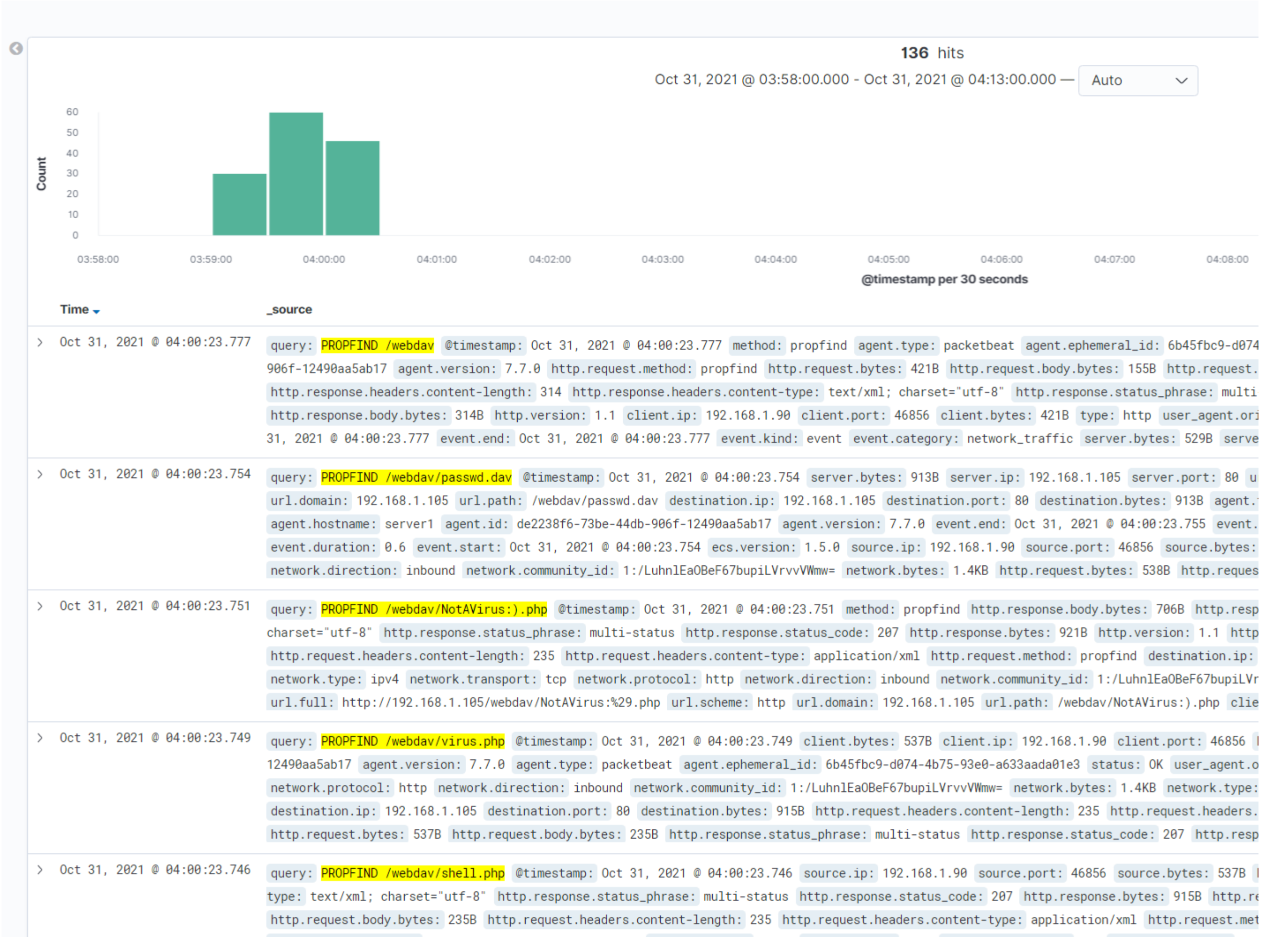
  Again, there seemed to be confidential information (instructions on how to gain access and upload to WebDav server) that was available on the server. If a hacker is somehow able to gain access to this directory, they would be able to do more damage by uploading malicious files/scripts to the server.

  2) Allow only known (employee) ip addresses to access the server.

5. **Identify the reverse shell and meterpreter traffic.**

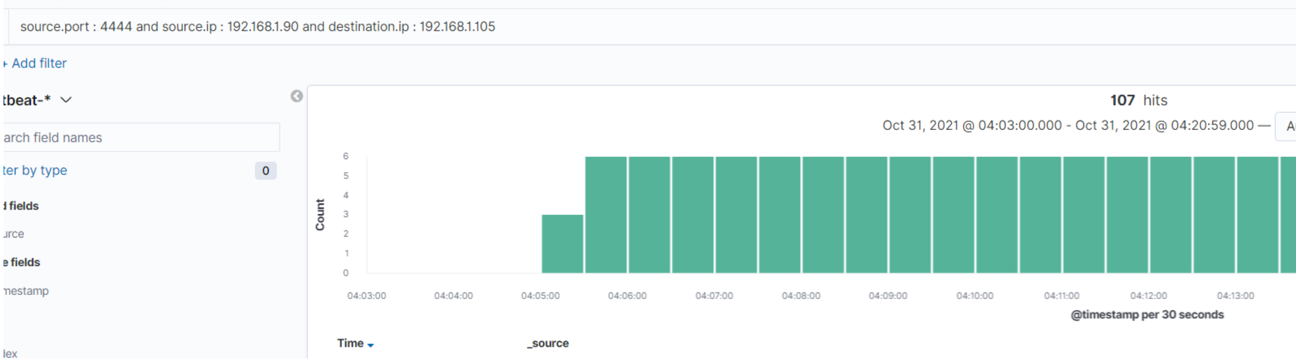
- **To finish off the attack, you uploaded a PHP reverse shell and started a meterpreter shell session. Answer the following questions:**
  - **Can you identify traffic from the meterpreter session?**

    It was hard to find traffic that identified with a meterpreter session. I did not see anything that specifically mentioned meterpreter, metasploit, (ect..) inside the packets. However, alot of traffic coming from the suspicious ip address ran over port 4444. Port 4444 is the default port used to establish a meterpreter session. You can search for traffic coming from that port. Around 4:00 A.M., I was able to find web traffic coming from the WebDav server and the Kali machine.



```
> Oct 31, 2021 @ 04:00:23.777  query: PROPFIND /webdav @timestamp: Oct 31, 2021 @ 04:00:23.777 method: propfind agent.type: packetbeat agent.ephemeral_id: 6b45fbc9-d074
                               906f-12490aa5ab17 agent.version: 7.7.0 http.request.method: propfind http.request.bytes: 421B http.request.body.bytes: 155B http.request.
                               http.response.headers.content-length: 314 http.response.headers.content-type: text/xml; charset="utf-8" http.response.status_phrase: multi
                               http.response.body.bytes: 314B http.version: 1.1 client.ip: 192.168.1.90 client.port: 46856 client.bytes: 421B type: http user_agent.ori
                               31, 2021 @ 04:00:23.777 event.end: Oct 31, 2021 @ 04:00:23.777 event.kind: event event.category: network_traffic server.bytes: 529B serve

> Oct 31, 2021 @ 04:00:23.754  query: PROPFIND /webdav/passwd.dav @timestamp: Oct 31, 2021 @ 04:00:23.754 server.bytes: 913B server.ip: 192.168.1.105 server.port: 80 u
                               url.domain: 192.168.1.105 url.path: /webdav/passwd.dav destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 913B agent.
                               agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 event.end: Oct 31, 2021 @ 04:00:23.755 event.
                               event.duration: 0.6 event.start: Oct 31, 2021 @ 04:00:23.754 ecs.version: 1.5.0 source.ip: 192.168.1.90 source.port: 46856 source.bytes:
                               network.direction: inbound network.community_id: 1:/LuhnlEa0BeF67bupiLVrvvVWmw= network.bytes: 1.4KB http.request.bytes: 538B http.reques

> Oct 31, 2021 @ 04:00:23.751  query: PROPFIND /webdav/NotAVirus:).php @timestamp: Oct 31, 2021 @ 04:00:23.751 method: propfind http.response.body.bytes: 706B http.resp
                               charset="utf-8" http.response.status_phrase: multi-status http.response.status_code: 207 http.response.bytes: 921B http.version: 1.1 http
                               http.request.headers.content-length: 235 http.request.headers.content-type: application/xml http.request.method: propfind destination.ip:
                               network.type: ipv4 network.transport: tcp network.protocol: http network.direction: inbound network.community_id: 1:/LuhnlEa0BeF67bupiLVr
                               url.full: http://192.168.1.105/webdav/NotAVirus:%29.php url.scheme: http url.domain: 192.168.1.105 url.path: /webdav/NotAVirus:).php clie

> Oct 31, 2021 @ 04:00:23.749  query: PROPFIND /webdav/virus.php @timestamp: Oct 31, 2021 @ 04:00:23.749 client.bytes: 537B client.ip: 192.168.1.90 client.port: 46856
                               12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: 6b45fbc9-d074-4b75-93e0-a633aada01e3 status: OK user_agent.o
                               network.protocol: http network.direction: inbound network.community_id: 1:/LuhnlEa0BeF67bupiLVrvvVWmw= network.bytes: 1.4KB network.type:
                               destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 915B http.request.headers.content-length: 235 http.request.headers.
                               http.request.bytes: 537B http.request.body.bytes: 235B http.response.status_phrase: multi-status http.response.status_code: 207 http.resp

> Oct 31, 2021 @ 04:00:23.746  query: PROPFIND /webdav/shell.php @timestamp: Oct 31, 2021 @ 04:00:23.746 source.ip: 192.168.1.90 source.port: 46856 source.bytes: 537B
                               type: text/xml; charset="utf-8" http.response.status_phrase: multi-status http.response.status_code: 207 http.response.bytes: 915B http.re
                               http.request.body.bytes: 235B http.request.headers.content-length: 235 http.request.headers.content-type: application/xml http.request.met
```

I took a screenshot (below) of all the meterpreter session activity.



```
source.port : 4444 and source.ip : 192.168.1.90 and destination.ip : 192.168.1.105
```

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

During the attack (Day 1) I uploaded a few different files to the WebDav server just to mess around.

The goal is to eliminate the possibility of an unauthorized user getting this far and accessing the WebDav server. However, if an cybercriminal made it this far:

I would create an alarm to alert me whenever traffic involves port 4444.

An alarm can also be created to notify us whenever a new php file is uploaded to the server.

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

  If the proposed mitigation methods above did not prevent a cybercriminal gaining access to the WebDav server - Set a rule that only allows authorized users from a specific ip address to upload files to the server, or entirely removing the ability to upload files over the web interface.