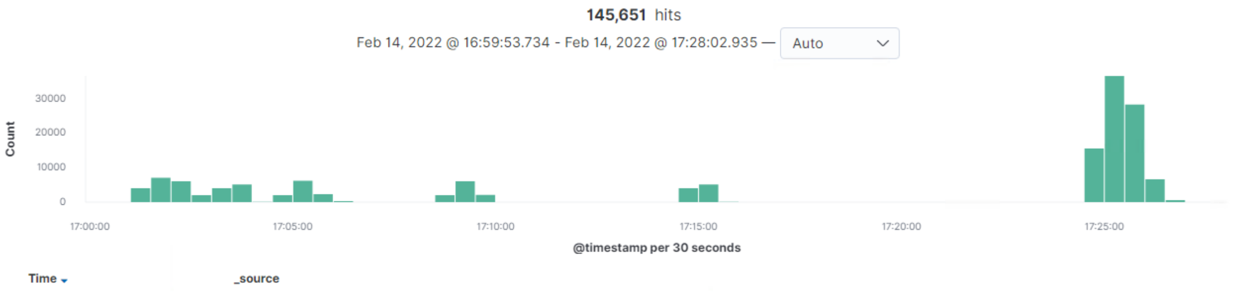
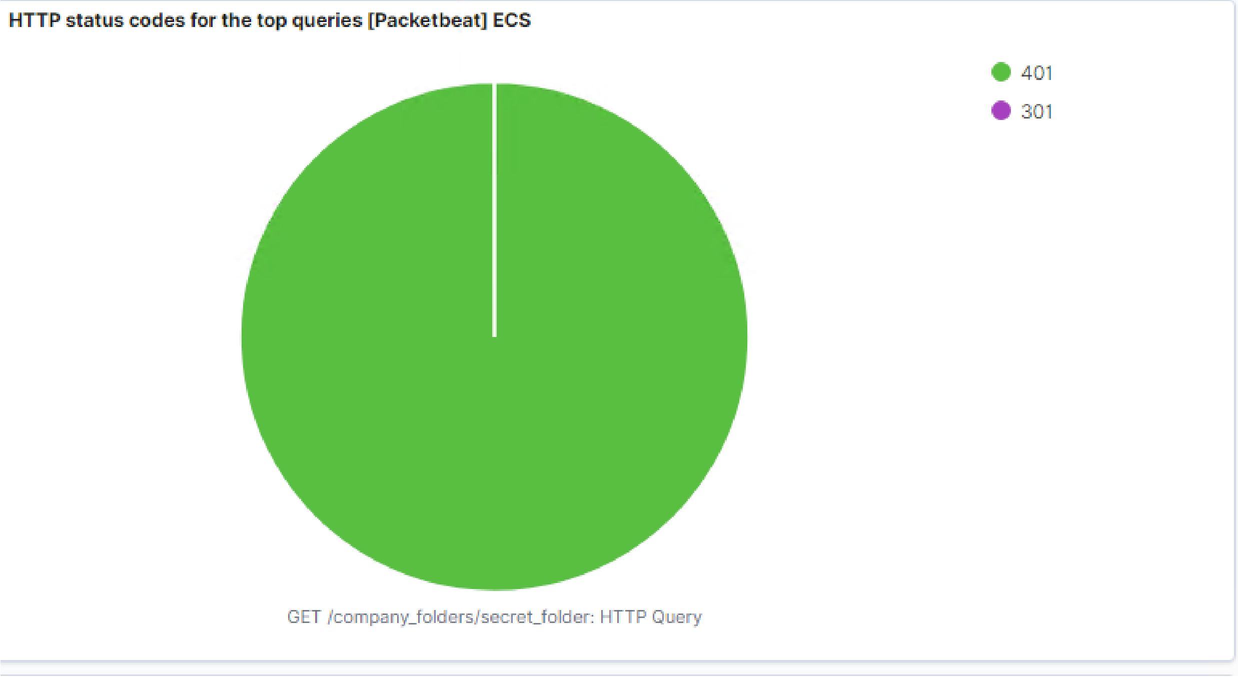
1. Identify the offensive traffic.
   - The traffic occurred around 17:00 in the windows machines time and occurred for around 2 hours with spikes
   - The responses returned were error codes 401, and 301 in which most were 401.



**145,651** hits

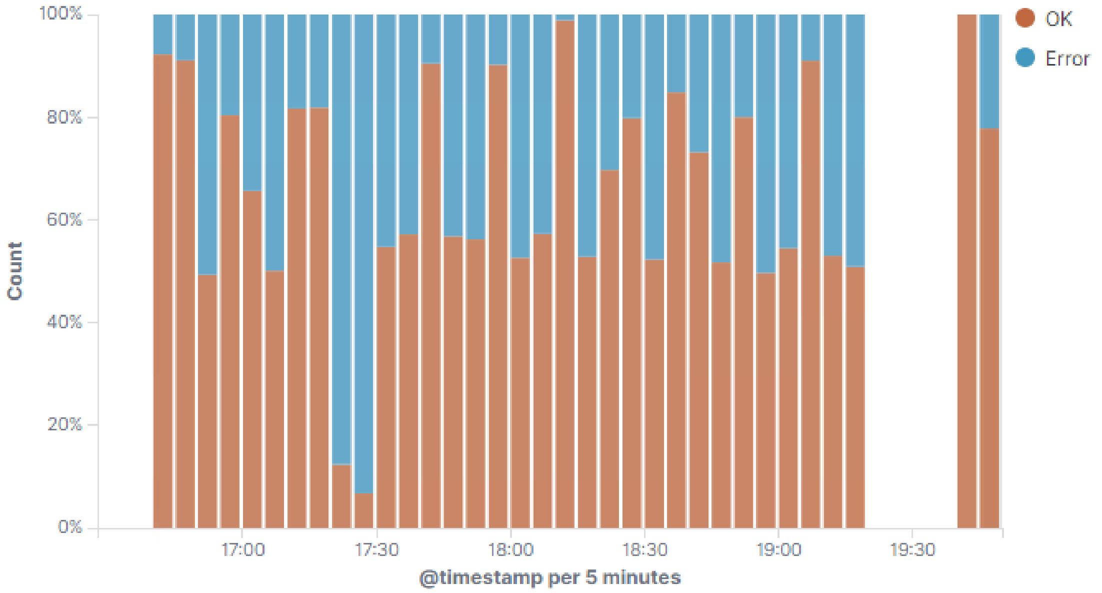
Feb 14, 2022 @ 16:59:53.734 - Feb 14, 2022 @ 17:28:02.935 — Auto



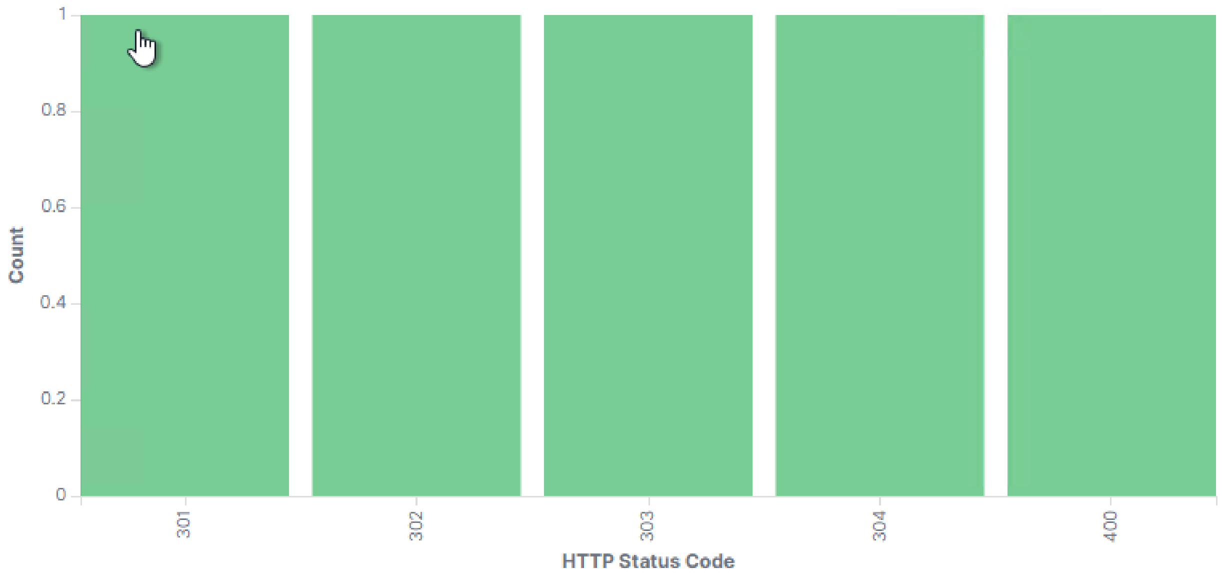
HTTP status codes for the top queries [Packetbeat] ECS

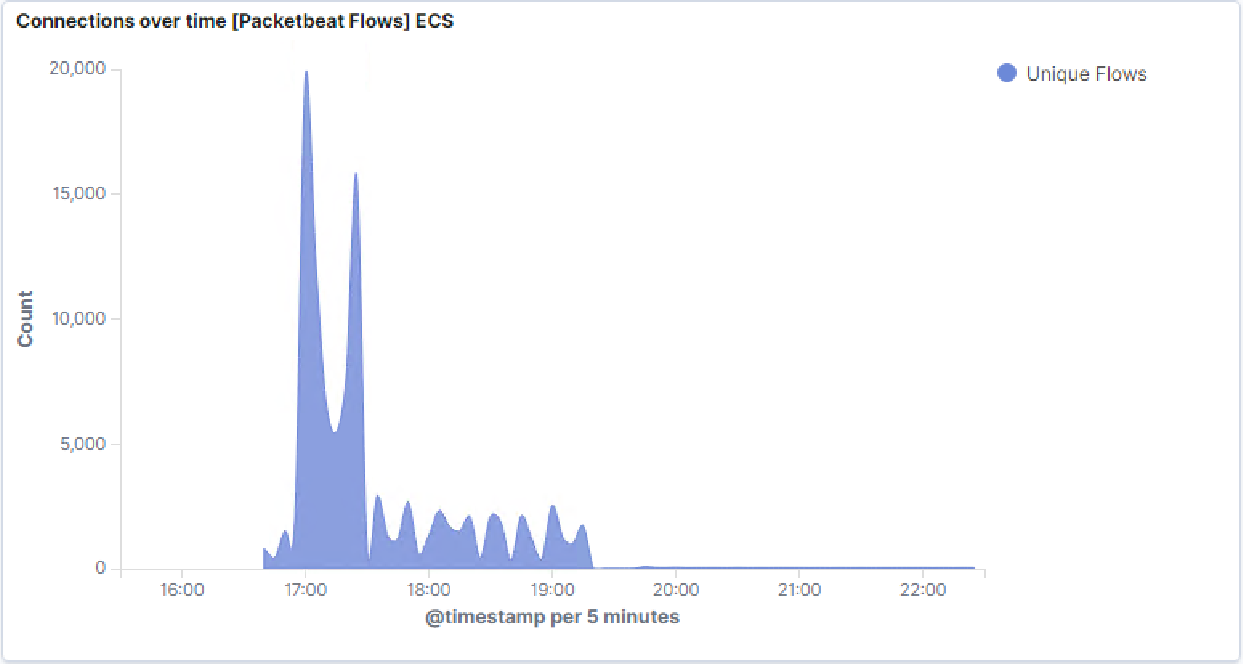
2. What data is concerning from the Blue Team perspective?
   - The data that is concerning are the connections over time as well as the error vs successful transactions occurring. This could mean that a brute force attack occurred and is creating such a large number of requests.

Errors vs successful transactions [Packetbeat] ECS

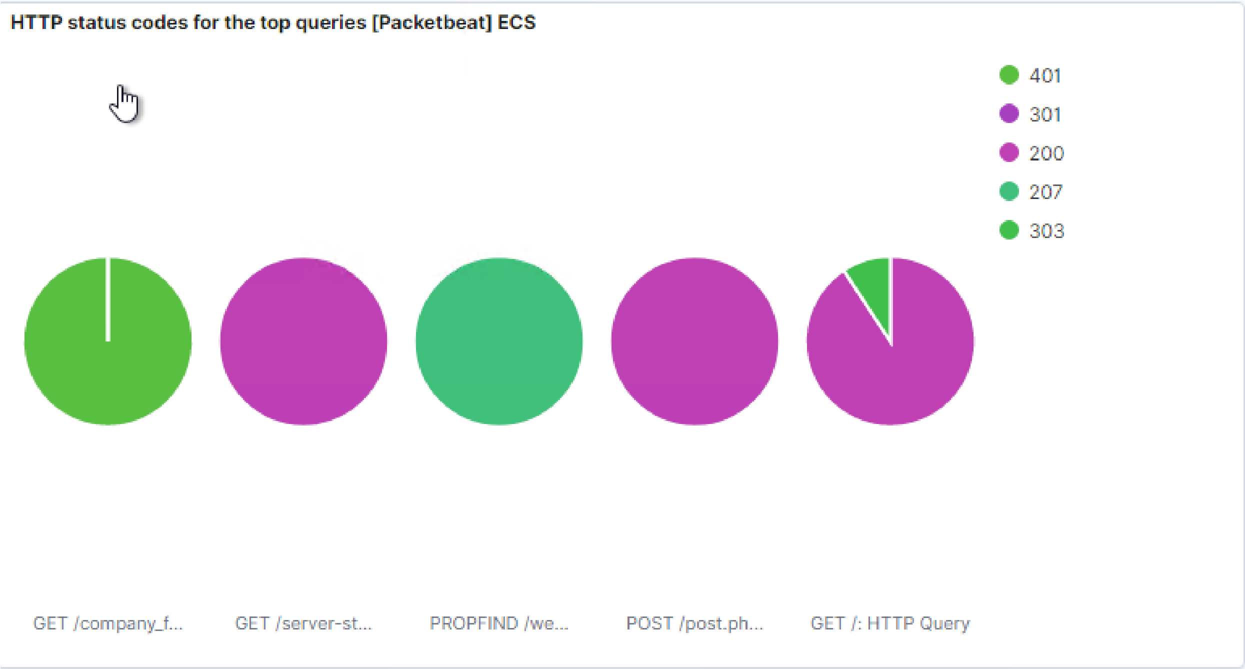


HTTP error codes [Packetbeat] ECS

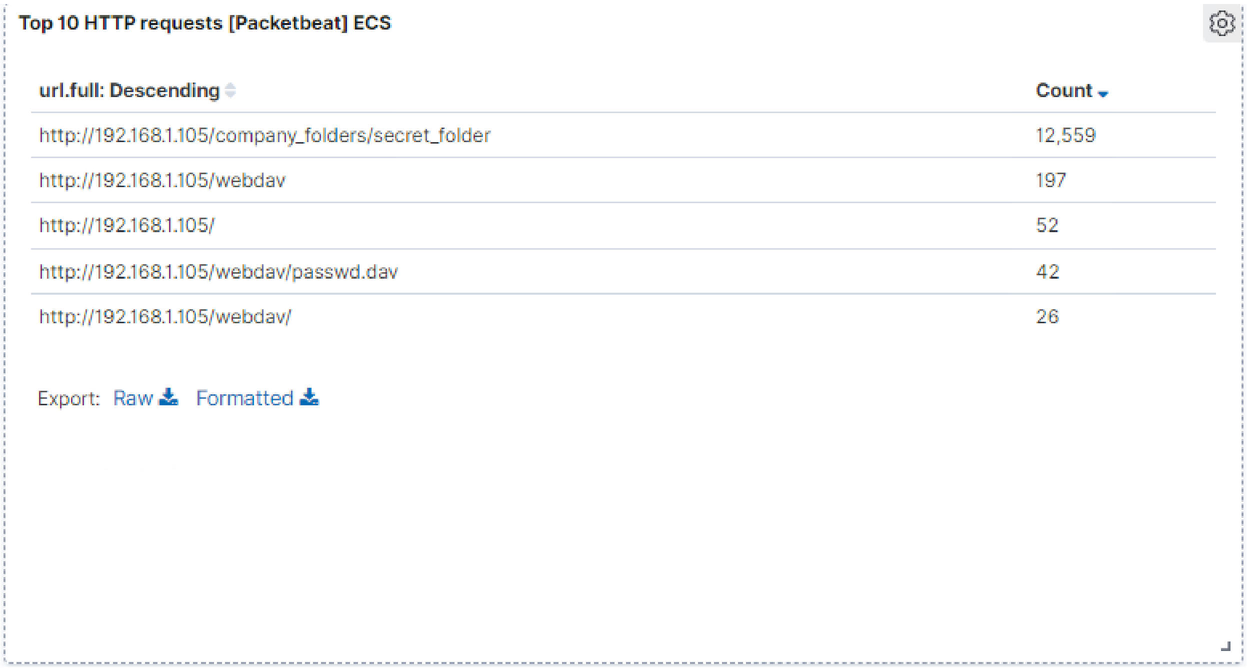




3. In your attack, you found a secret folder. Let's look at that interaction between these two machines.
   - How many requests were made to this directory? At what time and from which IP address(es)?
     - We can see in the screenshots that over 12,00 requests were made to the secret folder file, you can see the attack occurred during the 3 hours we are looking into, and most traffic came from the 192.168.1.90 ip.



HTTP status codes for the top queries [Packetbeat] ECS



Top 10 HTTP requests [Packetbeat] ECS

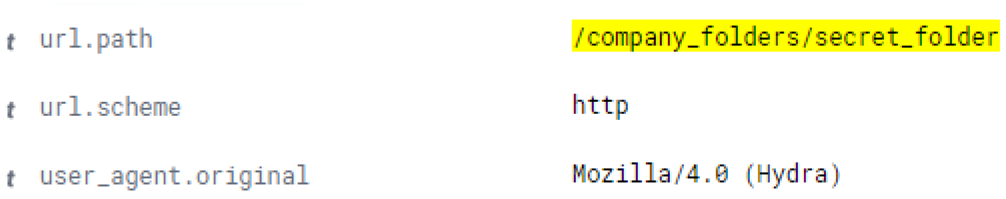
| url.full: Descending | Count |
|---|---|
| http://192.168.1.105/company_folders/secret_folder | 12,559 |
| http://192.168.1.105/webdav | 197 |
| http://192.168.1.105/ | 52 |
| http://192.168.1.105/webdav/passwd.dav | 42 |
| http://192.168.1.105/webdav/ | 26 |

Export: Raw Formatted

- Which files were requested? What information did they contain?
  - As show in the above screenshot the accessed primarily the webdav, secret_folder and passwd.ed areas.
- What kind of alarm would you set to detect this behavior in the future?
  - Setting an alert if any of these files are accessed such as the /secret_folder could detect any suspicious activity.
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.

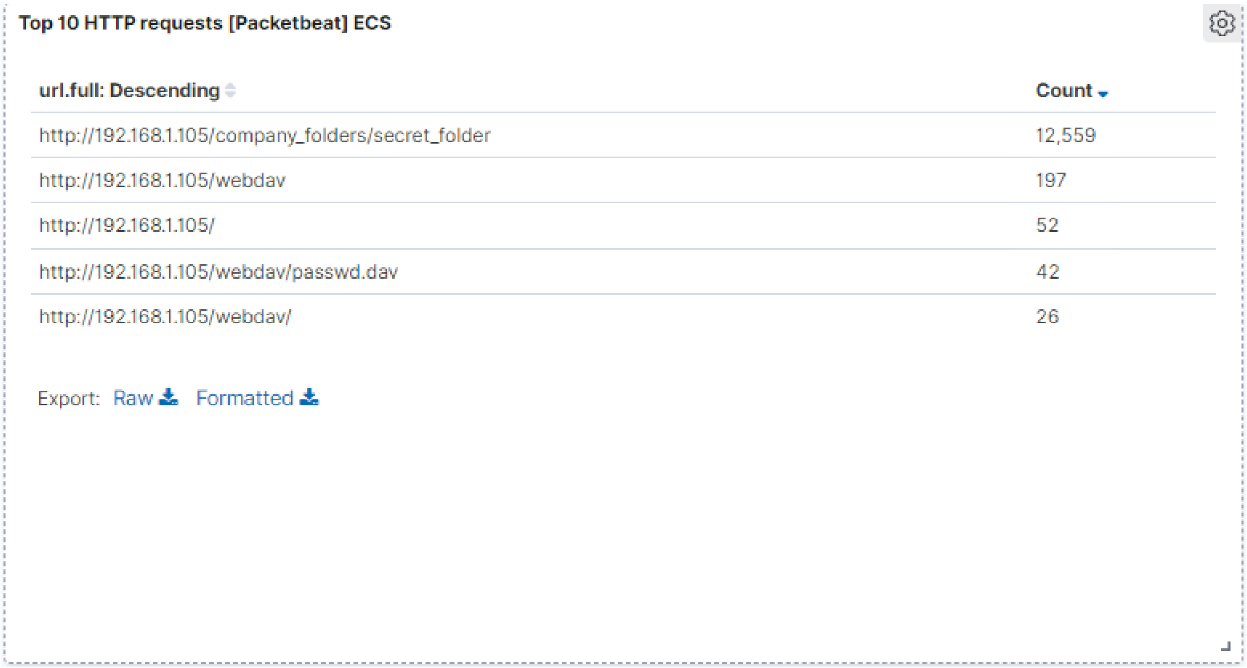
        - You could require secondary authorization, more complex passwords, more complex usernames, you would want to educate whoever was running it as data exposure to sensitive information is rampant.

    - 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, setting filters to search for the source.ip of 192.168.1.90, destination.ip of 192.168.1.105 ,user_agent.original: Mozilla/4.0 hydra and the url.path for /company_folders/secret_folder.



| | | |
|---|---|---|
| t | url.path | /company_folders/secret_folder |
| t | url.scheme | http |
| t | user_agent.original | Mozilla/4.0 (Hydra) |

    - How many requests were made in the brute-force attack?
        - You can see in the screenshots about 12,559 requests went up against the secret folder file.

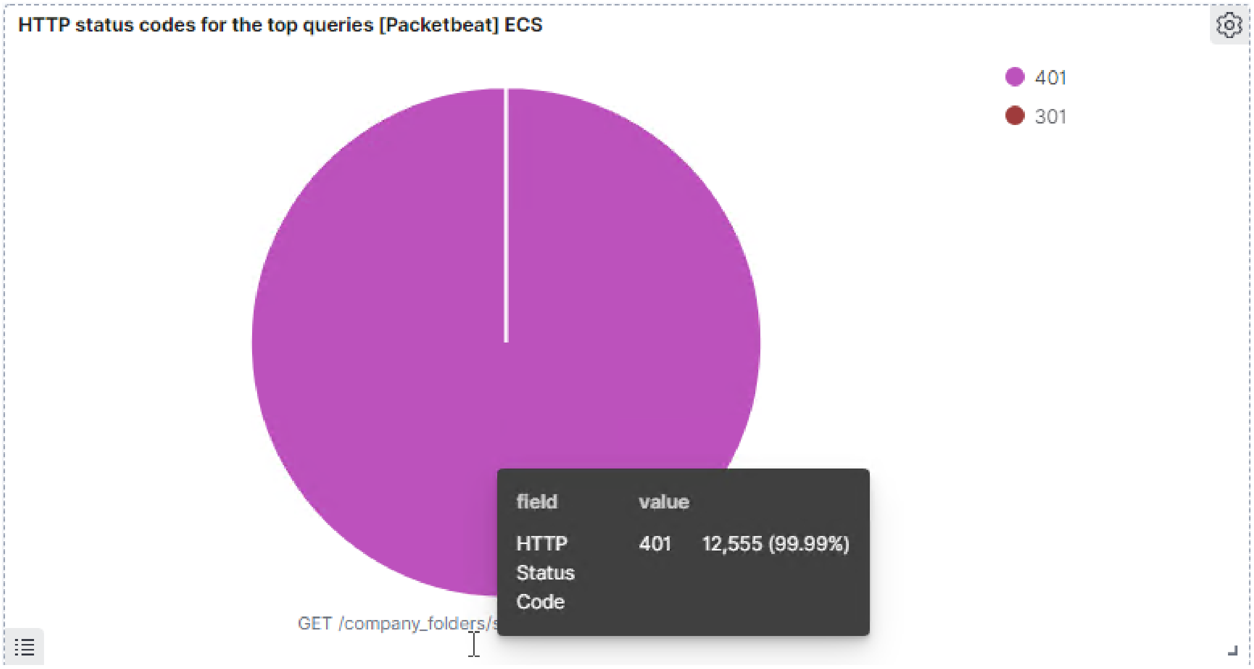


**Top 10 HTTP requests [Packetbeat] ECS**

| url.full: Descending | Count |
|---|---|
| http://192.168.1.105/company_folders/secret_folder | 12,559 |
| http://192.168.1.105/webdav | 197 |
| http://192.168.1.105/ | 52 |
| http://192.168.1.105/webdav/passwd.dav | 42 |
| http://192.168.1.105/webdav/ | 26 |

Export: Raw Formatted

- How many requests had the attacker made before discovering the correct password in this one?
    - The brute force attack ran for a couple minutes and after 12,555 attempts it succeeded.

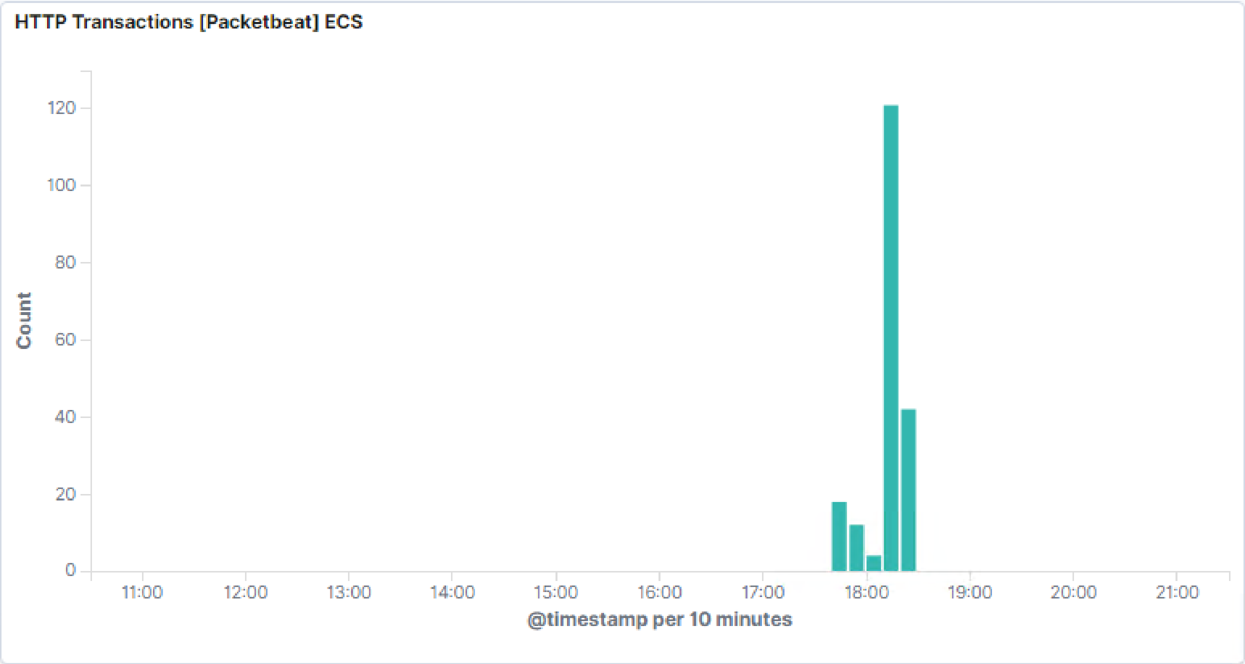


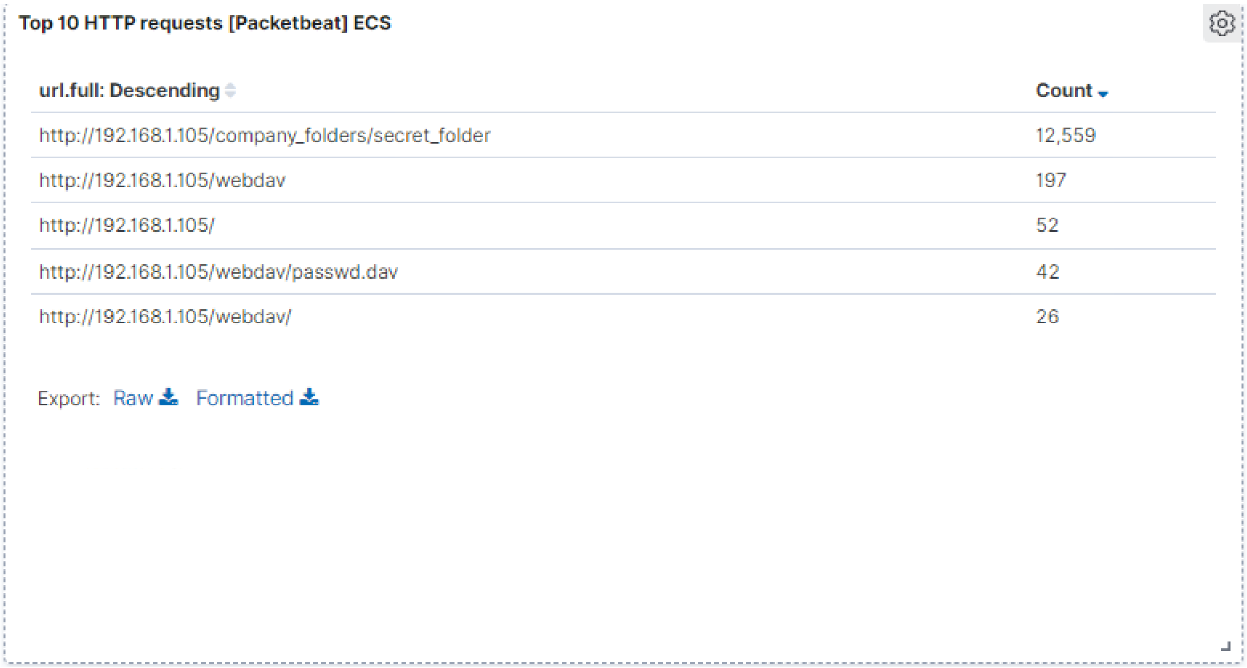
| field | value | |
|---|---|---|
| HTTP Status Code | 401 | 12,555 (99.99%) |

- What kind of alarm would you set to detect this behavior in the future and at what threshold(s)?
    - I believe this is a huge attack so we could have about 30 or so failed logins results in an alert being sent out to the appropriate sources. You could also make an alert if the hydra original agent appears in the packets.
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.
    - You could set a limited number of logins before a lockout occurs and have someone investigate, if suspicious have them blacklist the ip.

.

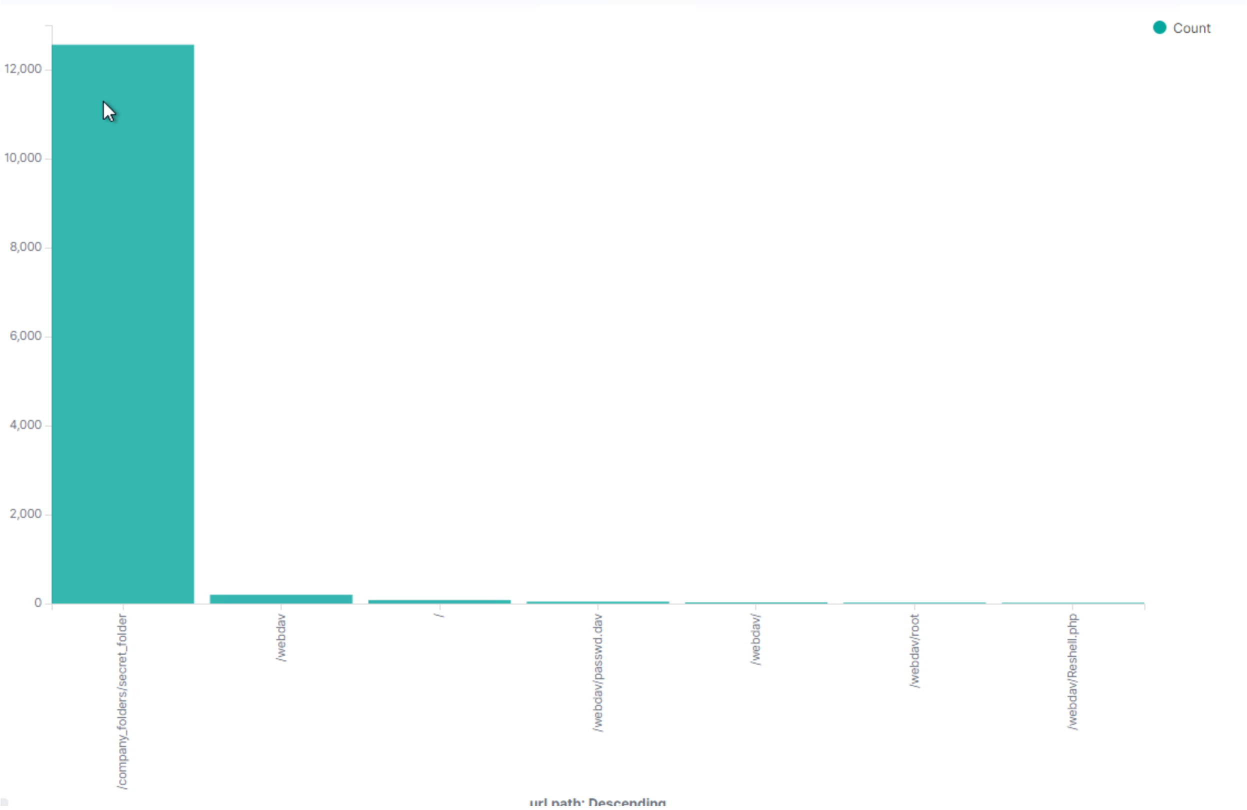
- Use your dashboard to answer the following questions:
- How many requests were made to this directory?
    - The screenshot shows that about 127 requests were made



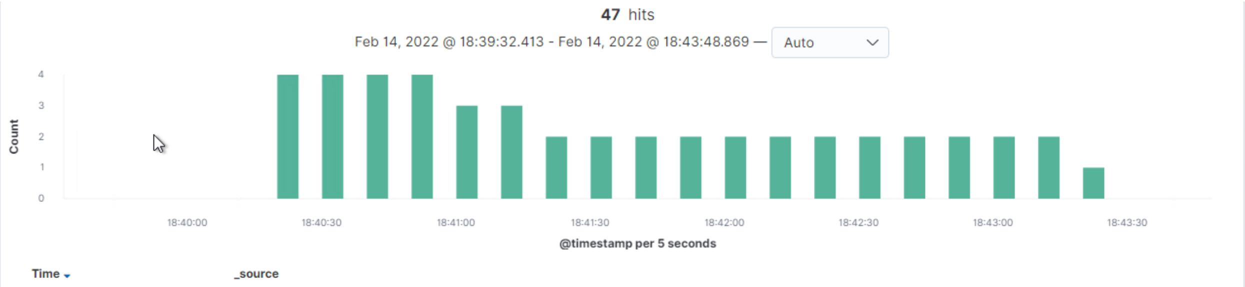

Top 10 HTTP requests [Packetbeat] ECS

| url.full: Descending | Count |
|---|---|
| http://192.168.1.105/company_folders/secret_folder | 12,559 |
| http://192.168.1.105/webdav | 197 |
| http://192.168.1.105/ | 52 |
| http://192.168.1.105/webdav/passwd.dav | 42 |
| http://192.168.1.105/webdav/ | 26 |

- Which file(s) were requested?
    - The screenshot above and below shows the files requested with password.dav being requested 42 times and reshell.php being requested 17 times



- What kind of alarm would you set to detect such access in the future?
    - Much like the other a simple alert when a machine, user or Ip address not on a whitelist or secured location accesses these documents.
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.
    - This webdav folder should not be able accessible from remote sources or at least have more secured passwords, usernames and multi factor authentication in place.
- 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?
    - Searching for the source port of the victim machine on 192.168.1.105 and port 4444 I was able to see when they communicated which was the reverse shell session starting and there were 47 requests.



47 hits

Feb 14, 2022 @ 18:39:32.413 - Feb 14, 2022 @ 18:43:48.869 — Auto

- What kinds of alarms would you set to detect this behavior in the future?
  - You can monitor the port itself on 4444 to see if any traffic is moving along, it and send an alert out if there is unauthorized access. An alert tailored for the file type php could also be made for whenever that file type is uploaded.
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.
  - To prevent this from occurring you would remove the ability to upload files over the webdev interface as well as add additional alerts and monitoring to files being uploaded. You could also not allow any filetypes such as .php from being uploaded.