Capstone Project: Red Team - Blue Team

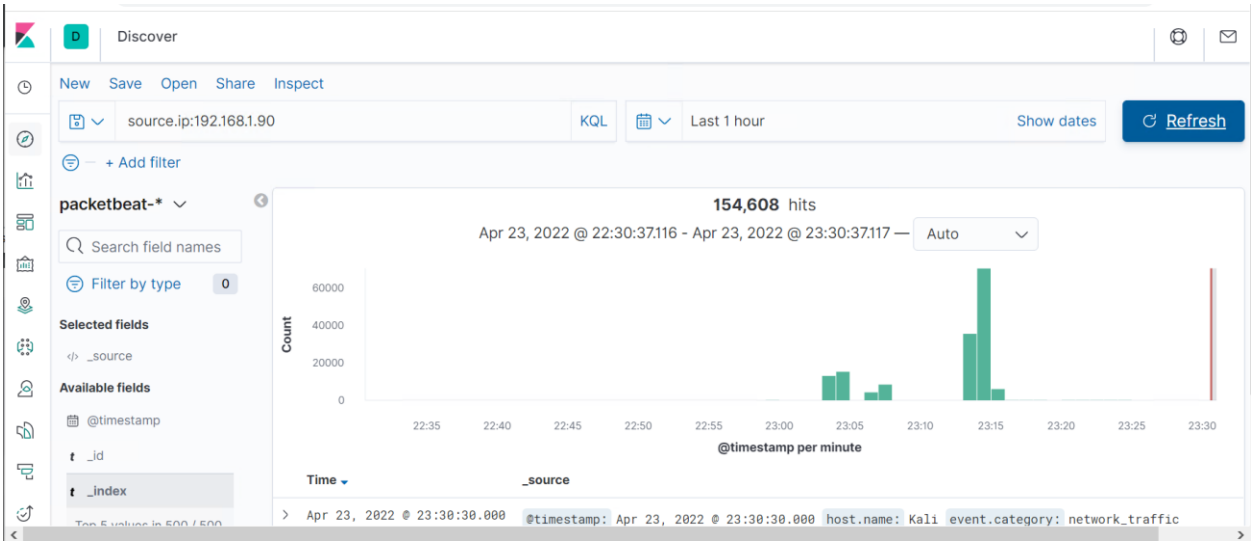
Day 2: Solved

Skye Falzett

2022.04.25

1. **Identify the offensive traffic.**
    - **Identify the traffic between your machine and the web machine:**

Searching for the traffic with the attack Kali Linux Machine (IP: 192.168.1.90):



- **When did the interaction occur?**

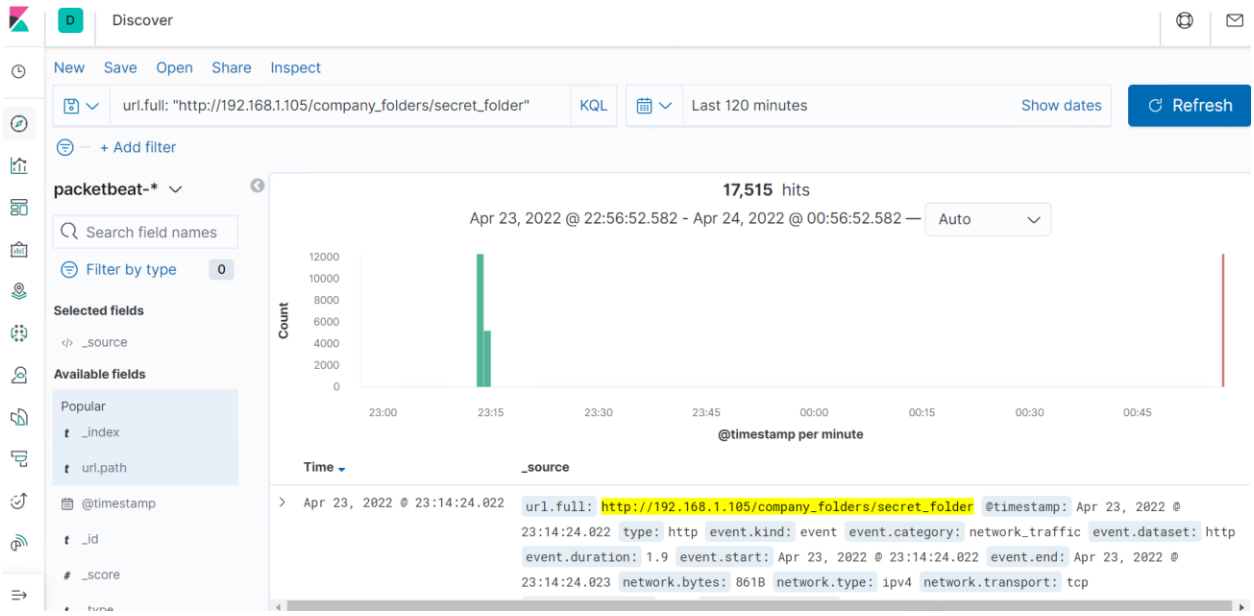
The interaction began at approximately 23:04 UTC on April 23, 2022, and continued until the Kali machine stopped its meterpreter session over about 3 hours.

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

The victim machine sent back various HTTP responses and GET requests.

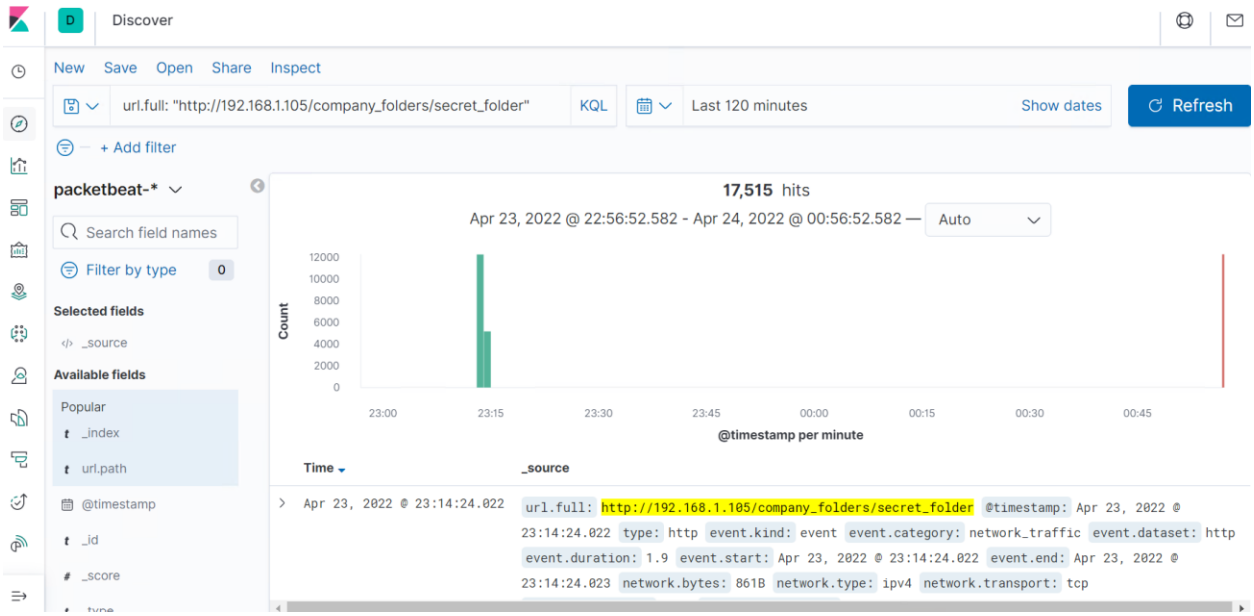
- **What data is concerning from the Blue Team perspective?**

A large amount of GET requests from the Kali machine at IP: 192.168.1.90 throughout a short period of time is concerning. We can also see the target of these GET requests was the company's "secret_folder."

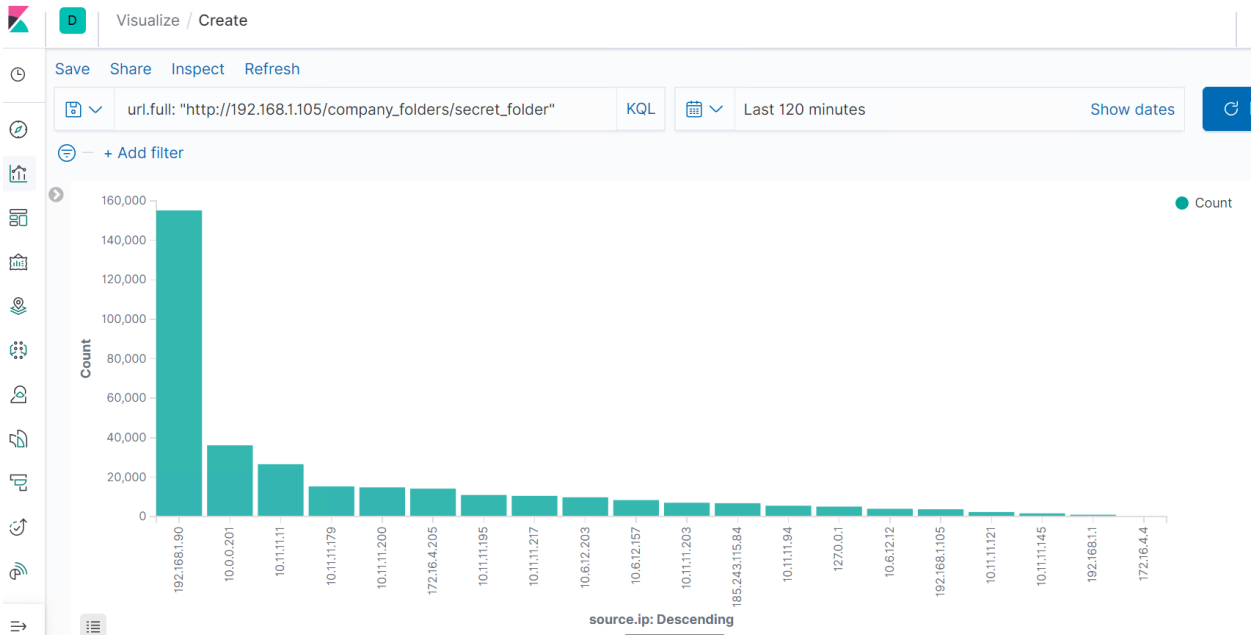


2. **Find the request for the hidden directory.**
    - **In your attack, you found a secret folder. Let's look at that interaction between these two machines.**
        - **How many requests were made to this directory? At what time and from which IP address(es)?**

There were 17,515 hits to the “secret_folder” directory.



This included activity from these IP addresses:



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

The file requested was inside the company's "secret_folder." Inside there were instructions on how to access the WebDAV directory.

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

An alarm can be set for any activity hits located in the "secret_folder." However, if there is regular traffic in the secret folder for other users, we can just set an alert for when there is an unusually high amount of traffic in the folder. We would need a baseline to decide on "unusually high" traffic.

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

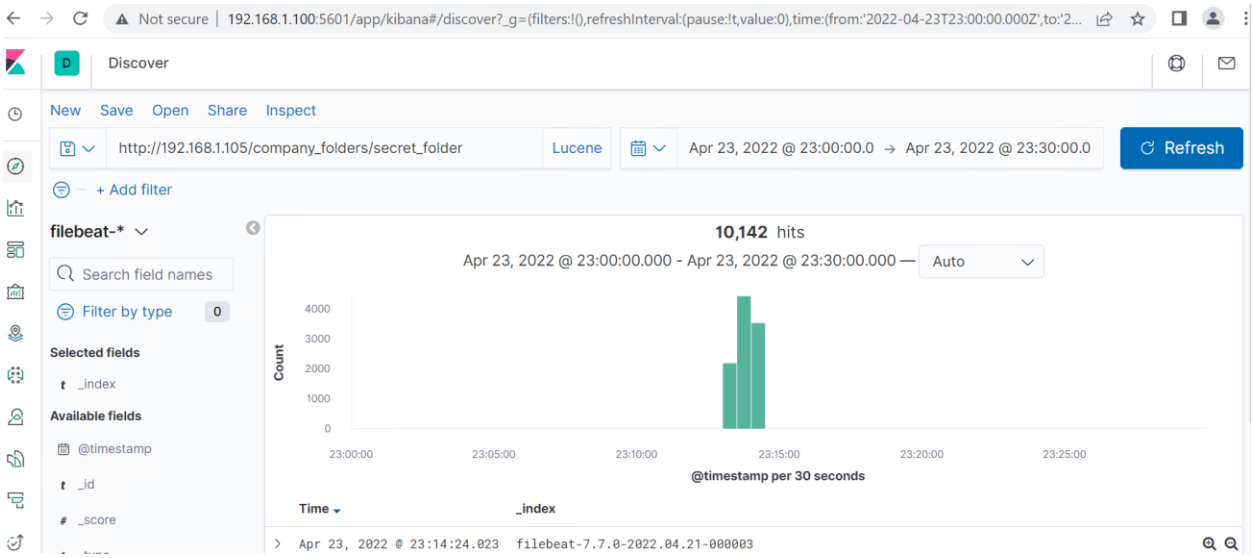
The webserver can be hardened by only allowing traffic from authorized IP addresses.

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: we can set the "user_agent.original" to "Mozilla/4.0 (Hydra)" and it shows the Hydra attack hits.

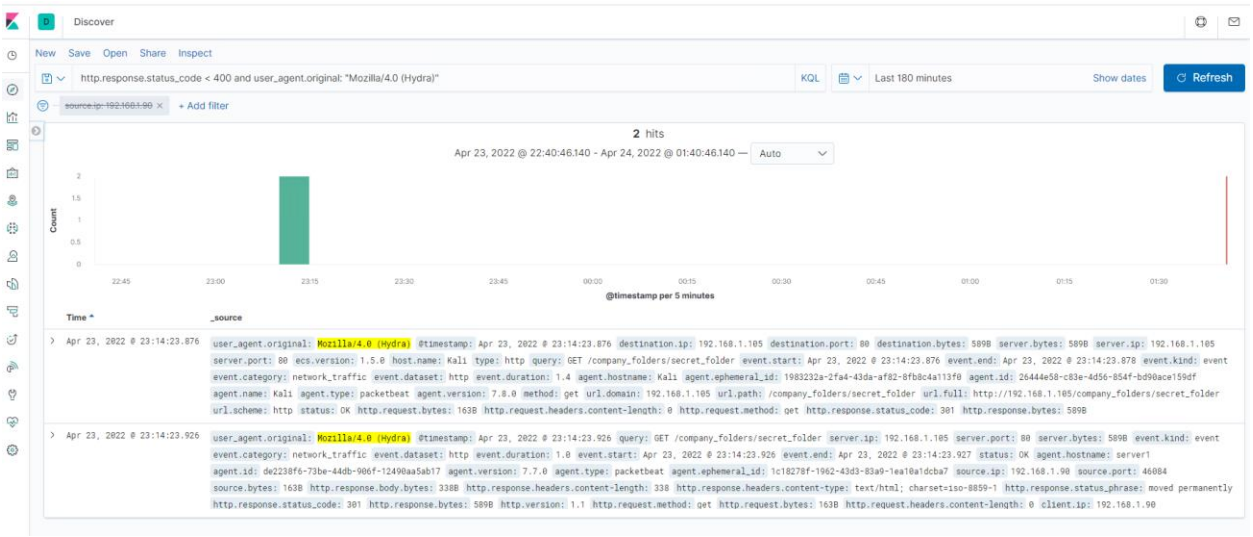
- **How many requests were made in the brute-force attack?**

There were 10,142 hits to this directory, http://192.168.1.105/company_folders/secret_folder, between 23:13 and 23:15 UTC (during the brute force attack).



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

The attack machine made approximately 10,140 hits until a successful password attempt. We can see this as I could search the results for when there weren't error codes in the results:



We see that there were two non-error coded hits during this time. This is when hydra successfully got the password. (I got approximately 10,140 by subtracting the two successful hits from the total of 10,142 hits.)

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

An alarm set for unusually high hits on the directory would be helpful. Any hits above twice to three times the average baseline for that directory should be sufficient to see this kind of activity.

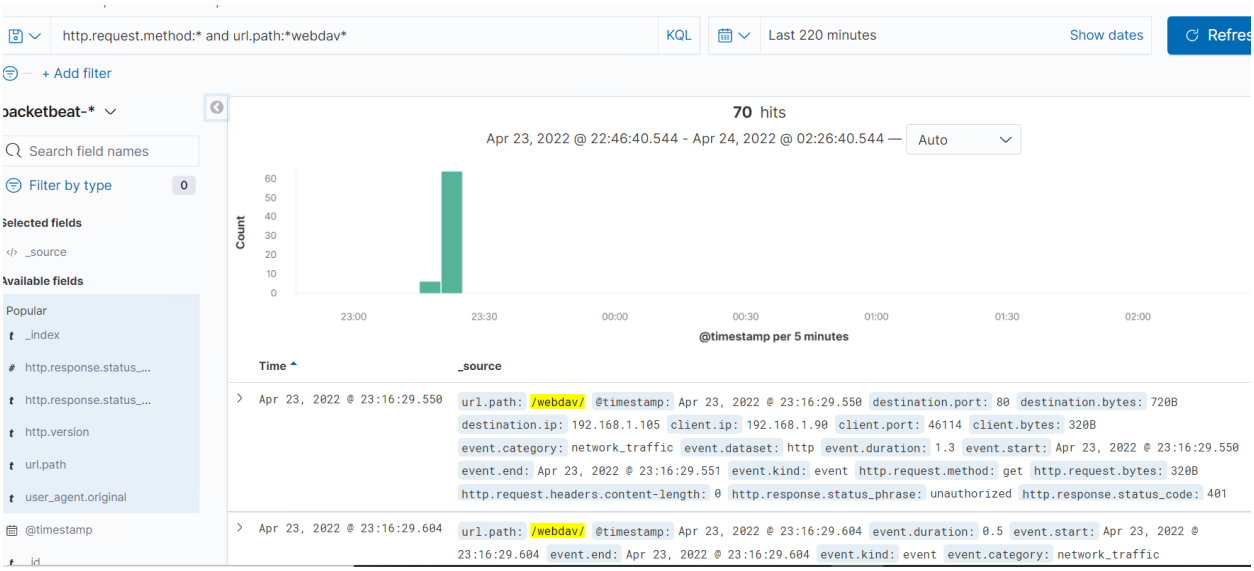
An alarm set for consecutive failed login denials would help identify this kind of brute force attempt. For example, let's say we harden the directory by only allowing ten login attempts before an IP is locked out from attempting login. After that, if the attacker is savvy enough to switch to a different IP to attack from, we can also set the alarm for a threshold for when multiple consecutive users have been locked out due to failed login attempts. If there are more than, say, five different users that have attempted and failed logins to the point that they have been locked out from login, it should trigger an alert.

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

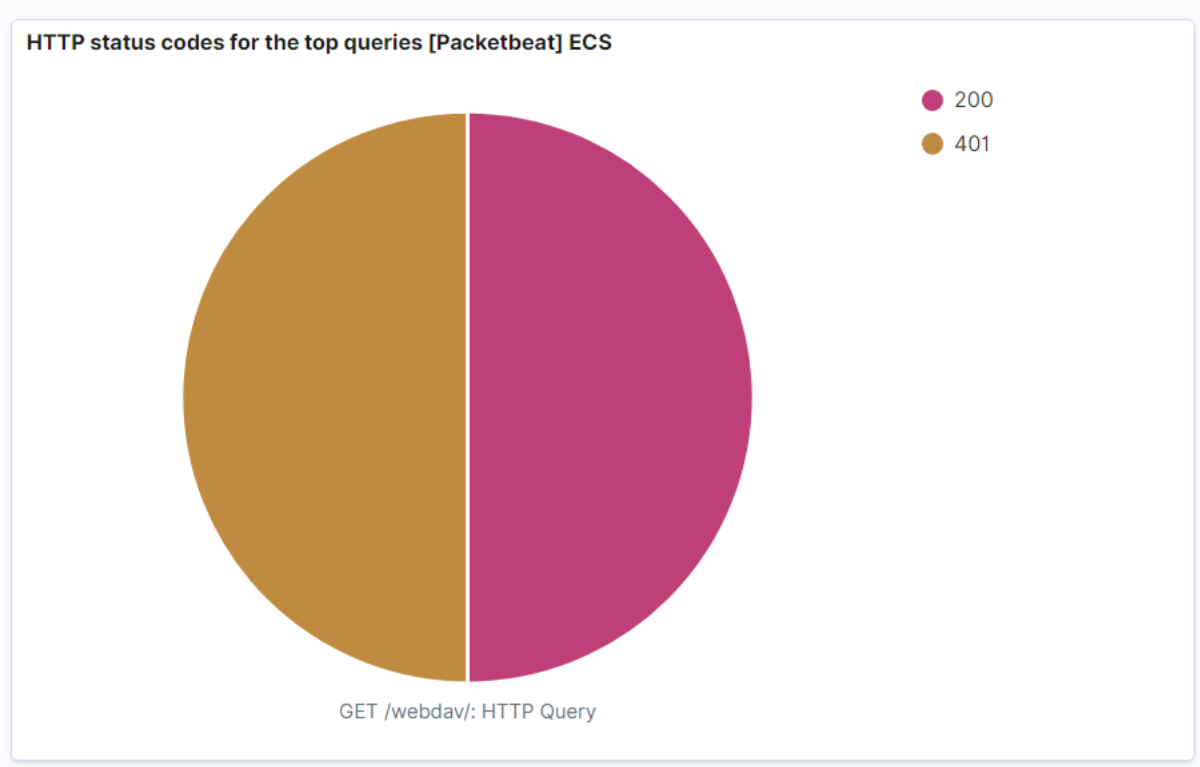
A straightforward way to harden the webserver would be to prevent login attempts after so many unsuccessful tries. If the login attempts were limited to 10 tries (or any tries under the 10,000 plus tries it took to crack the password), the brute force attack would have been stopped before access.

4. **Find the WebDav connection.**
    - **Use your dashboard to answer the following questions:**
        - **How many requests were made to this directory?**

Search results indicate 70 hits to the WebDav directory:

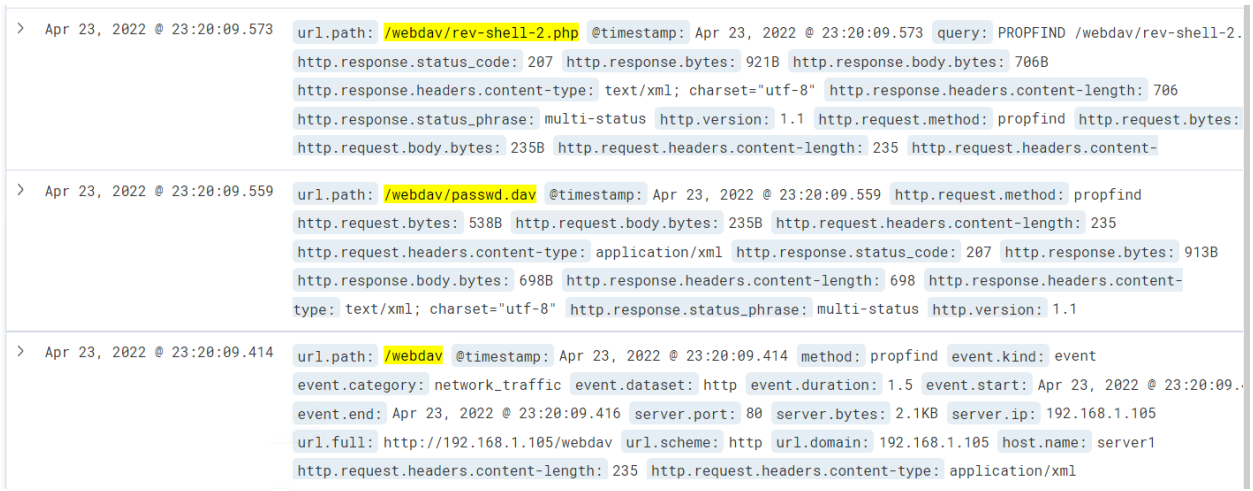


These requests included these GET requests with these HTTP responses:



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

The WebDAV directory was requested along with /WebDAV/password.dav. Here is a snippet of the types of hits:



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

Like with the “secret_folder” access, we can set an alarm that triggers when there is a higher level of hits than the average baseline.

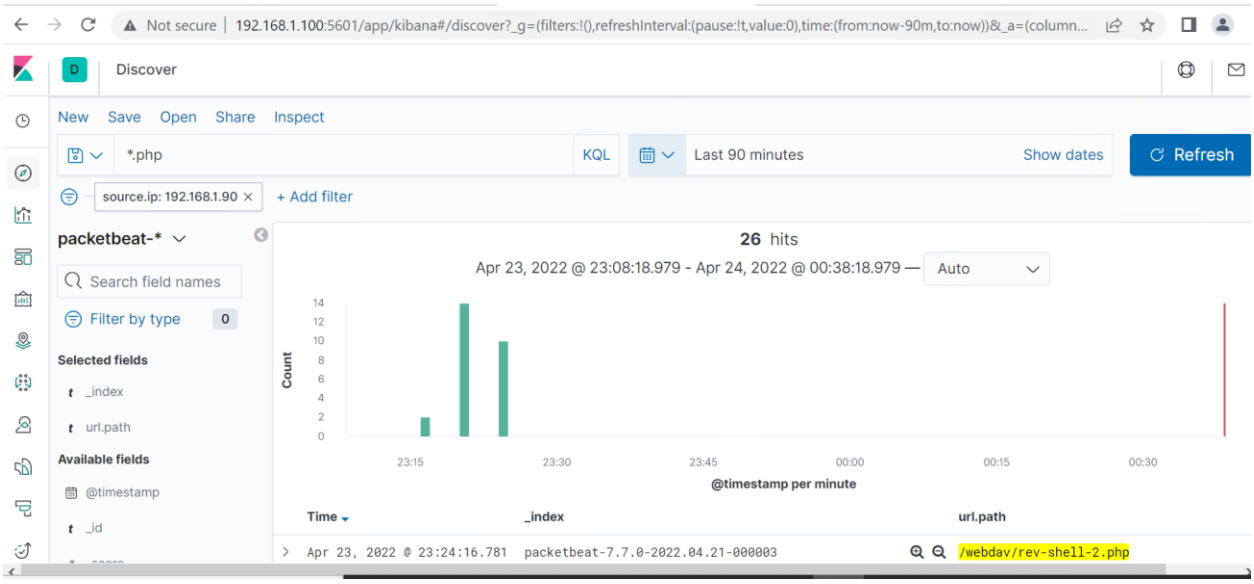
There can also be an alarm that triggers when any IP other than predetermined “safe” IPs tries to access the /webdav/ directory.

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

Limit access to this WebDAV directory to only specific, intracompany IP addresses.

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?**

Yes: searching *.php gives these results:



Additionally, then searching url.path: /webdav/rev-shell-2.php gives these results:

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

Different auditing software programs can mainly target identifying activity that looks like a reverse-shell attack. This would be the most efficient way to identify one.

Short of that, using the kibana interface, alerts can be set when traffic is higher than the baseline average inside the WebDAV directory.

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

First, instructions on specifically accessing and loading anything to the /webdav/ directory should not be written out on the webserver. Additionally, access to this file and directory can be limited not only by user name and password but also by location to only specified IPs.