1. Identify the offensive traffic.

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

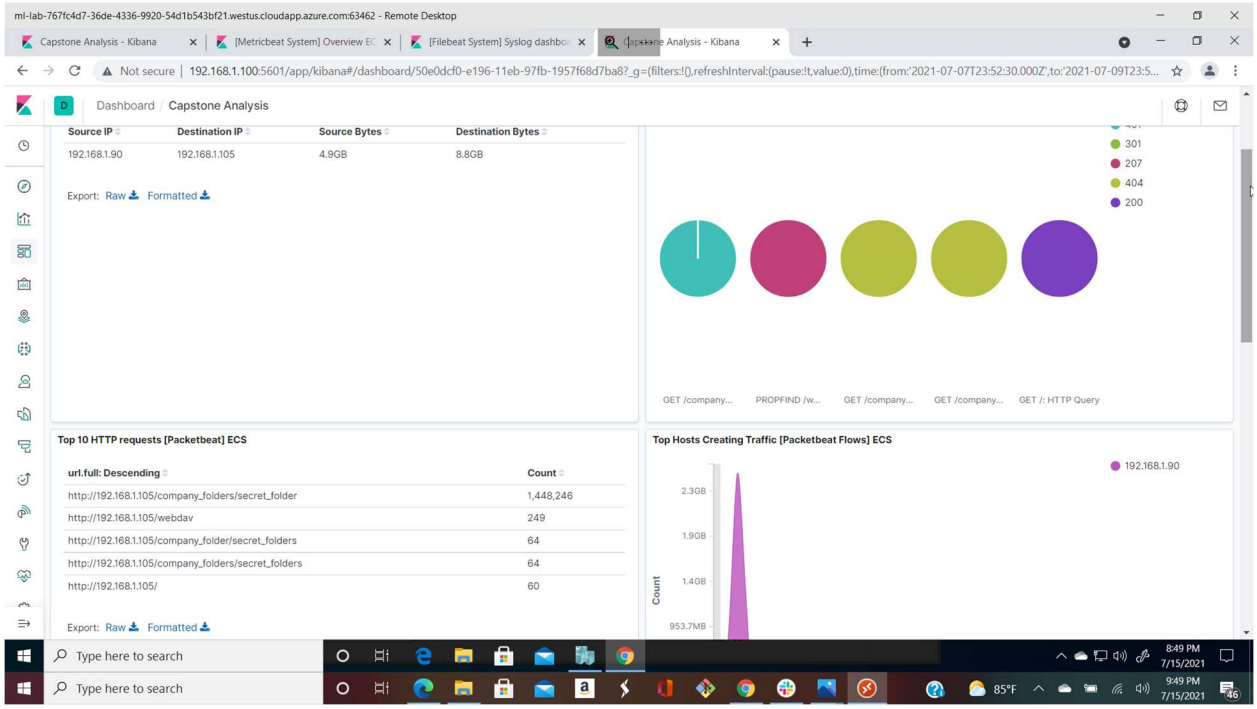
          Between July 7th,2021 and july 8th,2021

        - What responses did the victim send back?

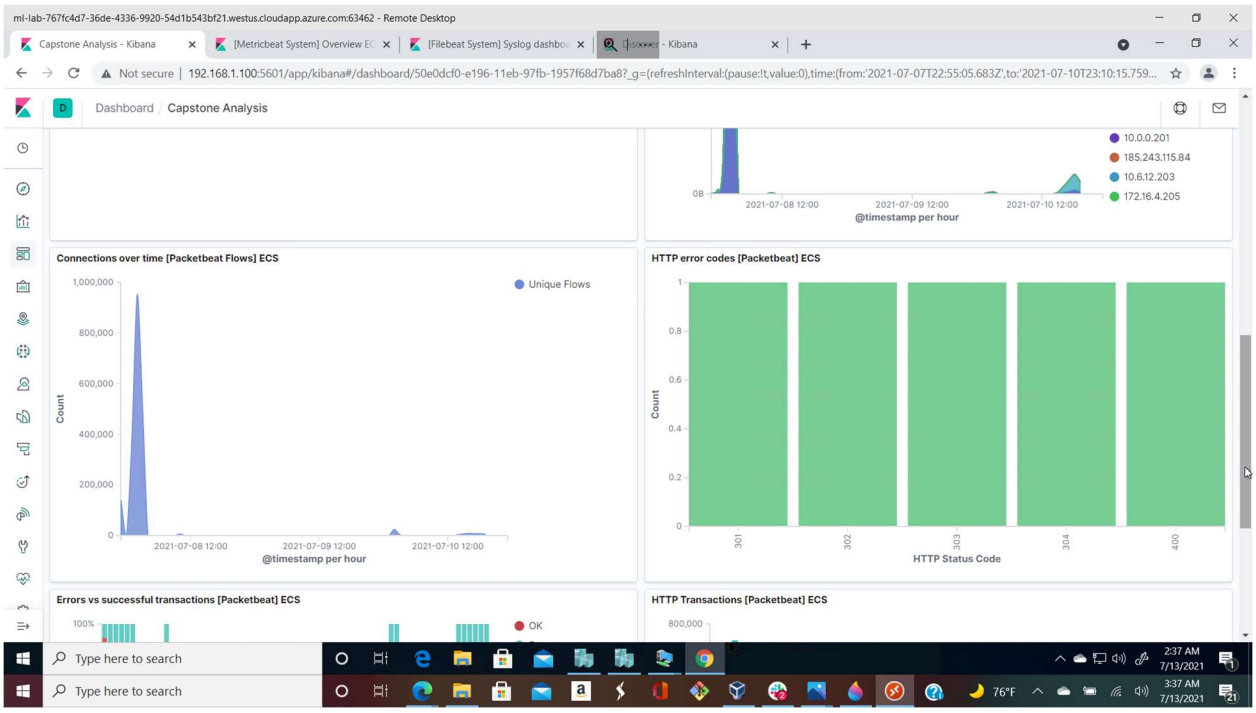
          200,207,301,303,401

        - What data is concerning from the Blue Team perspective?

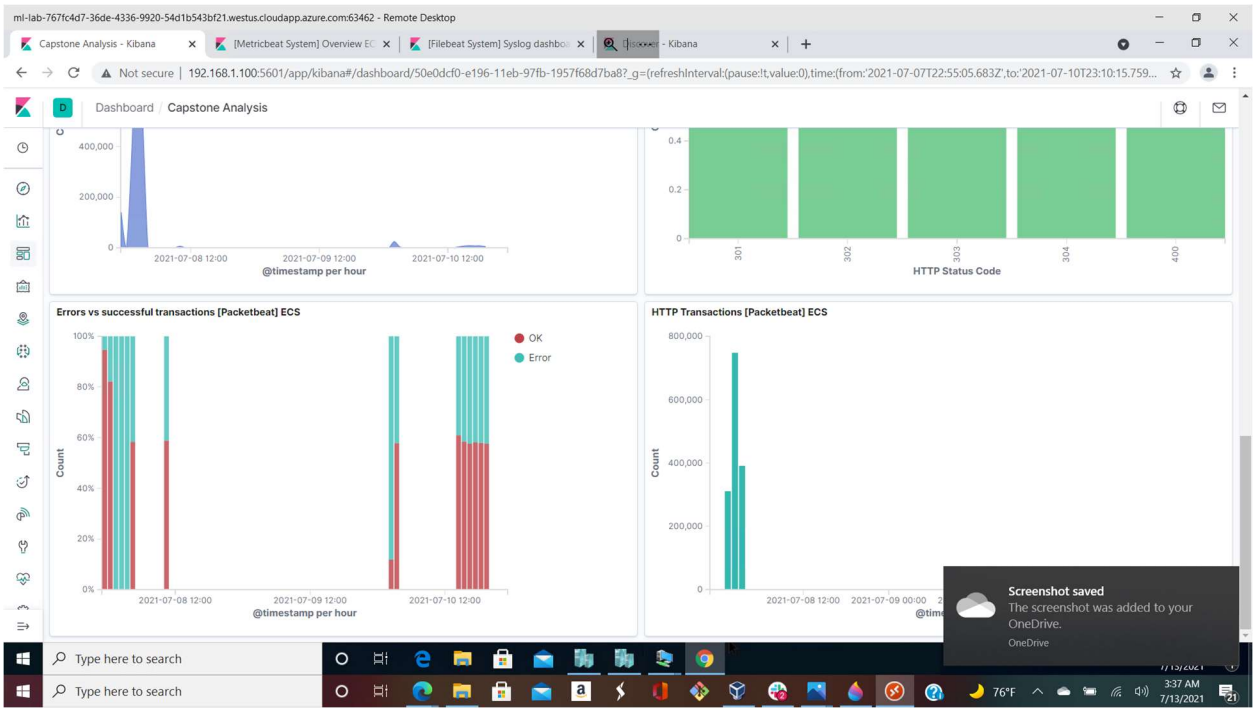
          Data with response_status_code 401,301



    a) Http response codes



b) Spike in Connection over time



c) Increase in Error versus successful logons

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)?1,448,246 hits, 03:31.30.235 WAT , 192.1681.90
        - Which files were requested? What information did they contain?

          connect_to_corp_server folder that contain information about the CEO ryan's account login details

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

          An alert to alarm when there is an access to the company_folders/secret_folder directory from unauthorized IPs

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

          We can use account lockout after 3 failed attempts with the use of a second factor authentication login (multifactor method) If there is a need to include such a folder. If not folders containing sensitive data should be removed totally from the webserver.

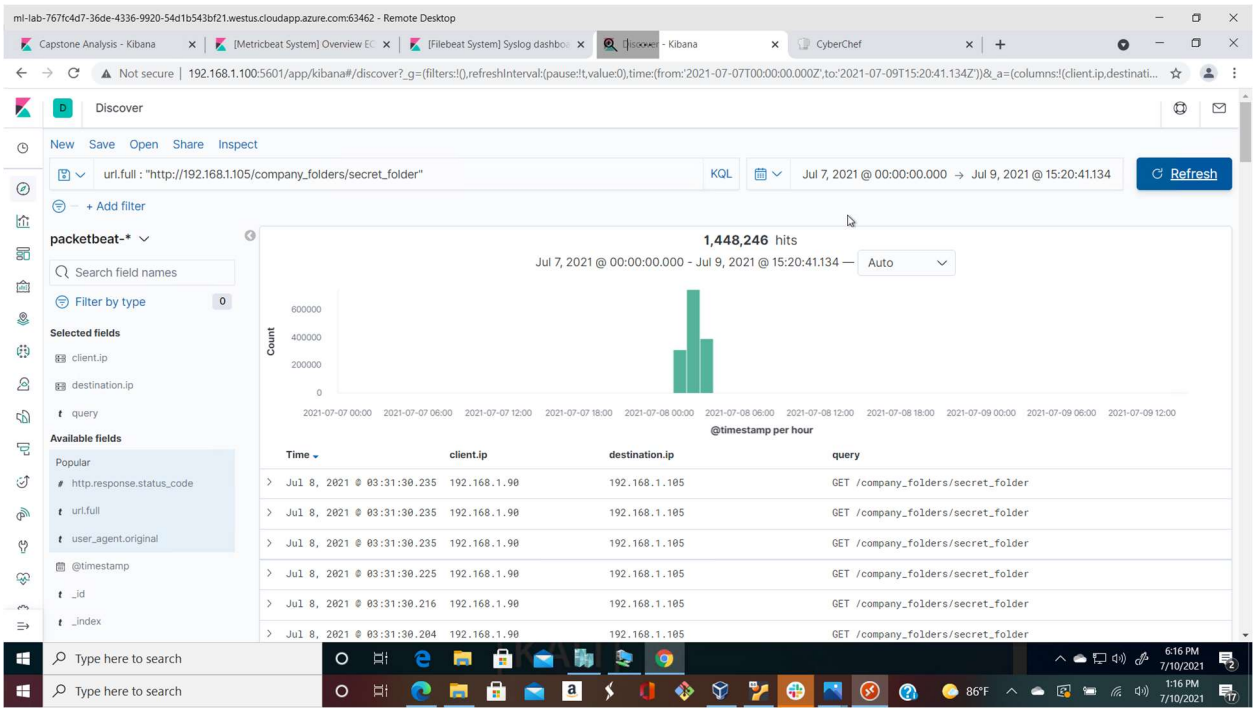
3. Identify the brute force attack.

    - 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, using a search term source.ip:192.168.1.90 and destination.ip:192.168.1.105 and url.full:http://192.168.1.105/company_folders/secret_folder and user_agent.original:"mozilla/4.0 (hydra) and http.response.status_code;401

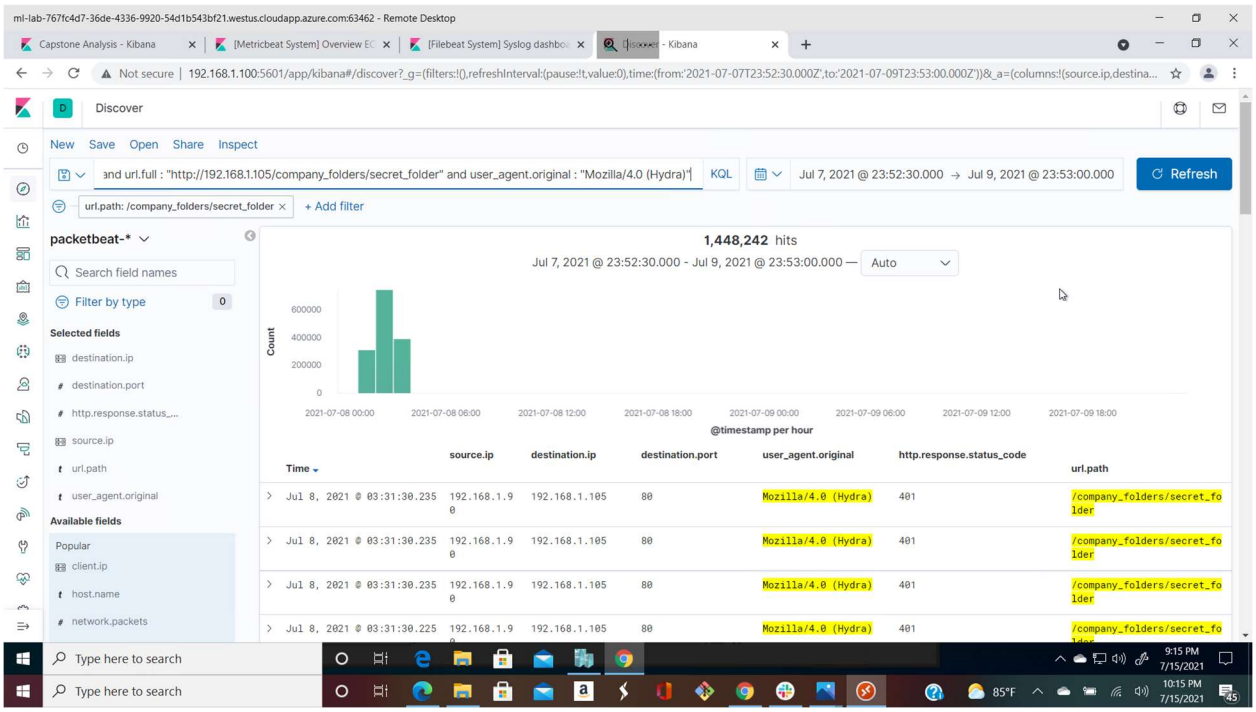
        - How many requests were made in the brute-force attack?

          Over 1,448,246m hits inclusive of bruteforce attempt to discover other passwords for Ryan and Hannah



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

  1,448,242m counts inclusive of bruteforce attempt to discover other passwords for Ryan and Hannah



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

  We could set an alert if 401 Unauthorized is returned from any server over a certain threshold that would weed out forgotten passwords. Start with 10 in one hour and refine from there.

  Settingan alert for a threshold for error 401 greater than 10 is a good fit to detect forgotten passwords on the server

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

  After the limit of 10 for 401 Unauthorized codes have been returned from a server, that server can automatically drop traffic from the offending IP address for a period of 1 hour. We could also display a lockout message and lock the page from login for a temporary period of time from that user.

  We can use multifactor authentication

  Use of CAPTCHA to avoid inhuman activities

4. Find the WebDav connection.

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

       249

     - Which file(s) were requested?

       We can see the passwd.dav file was requested as well as a file named shell.php

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

       We can create an alert anytime this directory is accessed by a machine *other* than the machine that should have access.

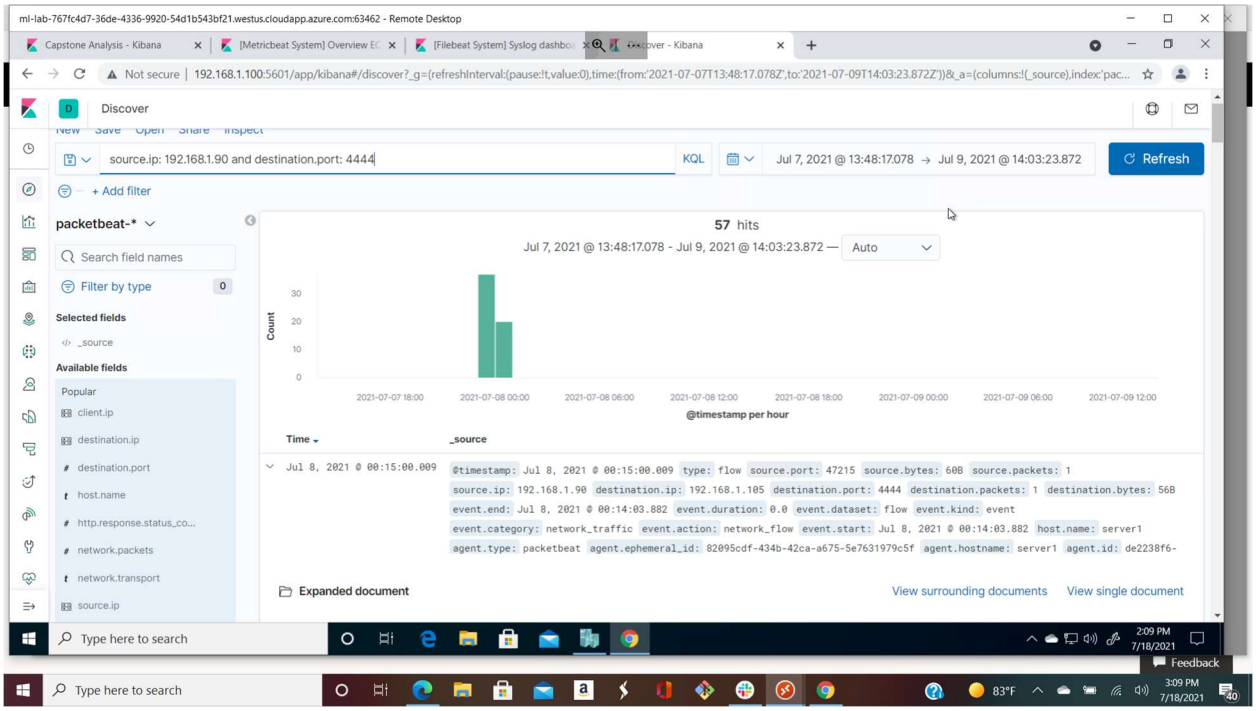
     - Identify at least one way to harden the vulnerable machine that would mitigate this attack.

       Connections to this shared folder should not be accessible from the web interface.

Connections to this shared folder could be restricted by machine with a firewall rule.

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, we can search with source.ip:192.168.1.90 and destination.port:4444 because most of the time attackers tend to use the default port 4444 for their attacks and tend to forget to change ports number.



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

          We can set an alert for any traffic moving over port 4444.

          We can set an alert for any .php file that is uploaded to a server.

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

          Removing the ability to upload files to this directory over the web interface would take care of this issue.

Installing anti-malware systems