## **Day 2 Activity File: Incident Analysis with Kibana**

Today, you will use Kibana to analyze logs taken during the Red Team attack. As you analyze, you will use the data to develop ideas for new alerts that can improve your monitoring.

**Important**: Any time you use data in a dashboard to justify an answer, take a screenshot. You'll need these screenshots when you develop your presentation on Day 3 of this project.

⚠ **Heads Up**: To complete today's part of the project, you must complete steps 1-6 from the last class. Finding the flag isn't critical, but you want to get past the point of uploading the reverse shell script.

### **Instructions**

Even though you already know what you did to exploit the target, analyzing the logs is still valuable. It will teach you:

* What your attack looks like from a defender's perspective.
* How stealthy or detectable your tactics are.
* Which kinds of alarms and alerts SOC and IR professionals can set to spot attacks like yours while they occur, rather than after.

#### **Adding Kibana Log Data**

To start viewing logs in Kibana, we will need to import our filebeat, metricbeat and packetbeat data.

Double-click the Google Chrome icon on the Windows host's desktop to launch Kibana. If it doesn't load as the default page, navigate to<http://192.168.1.105:5601>.

This will open 4 tabs automatically, but for now, we only want to use the first tab.

Click on the Explore My Own link to get started.

##### **Adding Appache logs**

Click on Add Log Data 

Click on Apache logs



Scroll to the bottom of the page. Click on Check Data You should see a message highlighted in green: Data successfully received from this module



Return to the Home screen by moving back 2 pages.

##### **Adding System Logs**

Click on Add Log Data

)

Click on System logs



Scroll to the bottom of the page. Click on Check Data You should see a message highlighted in green: Data successfully received from this module



Return to the Home screen by moving back 2 pages.

#### **Adding Apache Metrics**

Click on Add Metric Data



Click on Apache Metrics



Scroll to the bottom of the page. Click on Check Data You should see a message highlighted in green: Data successfully received from this module



Return to the Home screen by moving back 2 pages.

#### **Adding System Metrics**

Click on Add Metric Data



Click on System Metrics



Scroll to the bottom of the page. Click on Check Data You should see a message highlighted in green: Data successfully received from this module



Close Google Chrome and all of it's tabs. Double click on Chrome to re-open it.

#### **Dashboard Creation**

Create a Kibana dashboard using the pre-built visualizations. On the left navigation panel, click on **Dashboards**.

Click on **Create dashboard** in the upper right hand side.



On the new page click on **Add an existing** to add the following existing reports:

* HTTP status codes for the top queries [Packetbeat] ECS
* Top 10 HTTP requests [Packetbeat] ECS
* Network Traffic Between Hosts [Packetbeat Flows] ECS
* Top Hosts Creating Traffic [Packetbeat Flows] ECS
* Connections over time [Packetbeat Flows] ECS
* HTTP error codes [Packetbeat] ECS
* Errors vs successful transactions [Packetbeat] ECS
* HTTP Transactions [Packetbeat] ECS

**Example for adding the first report:**

****

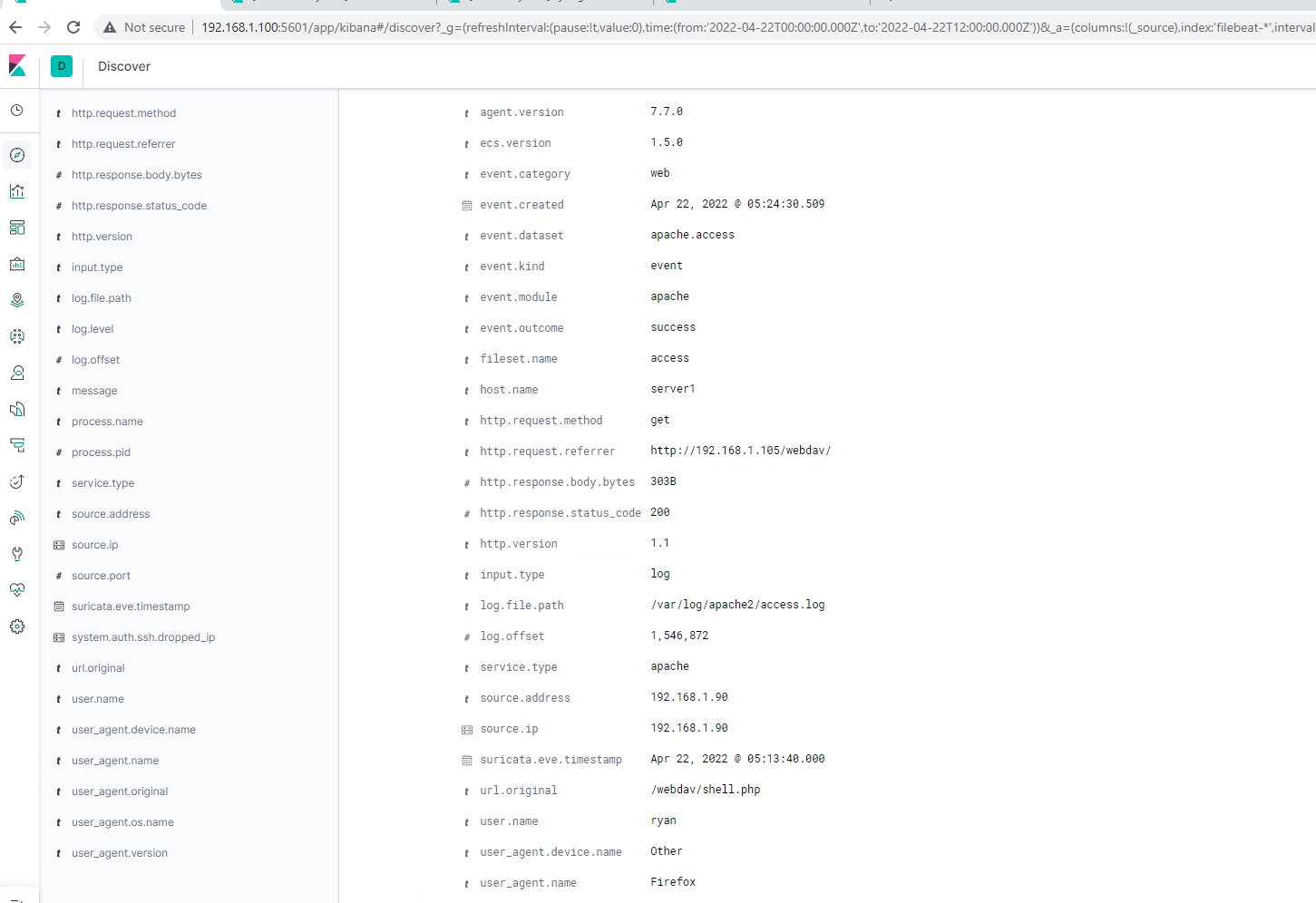
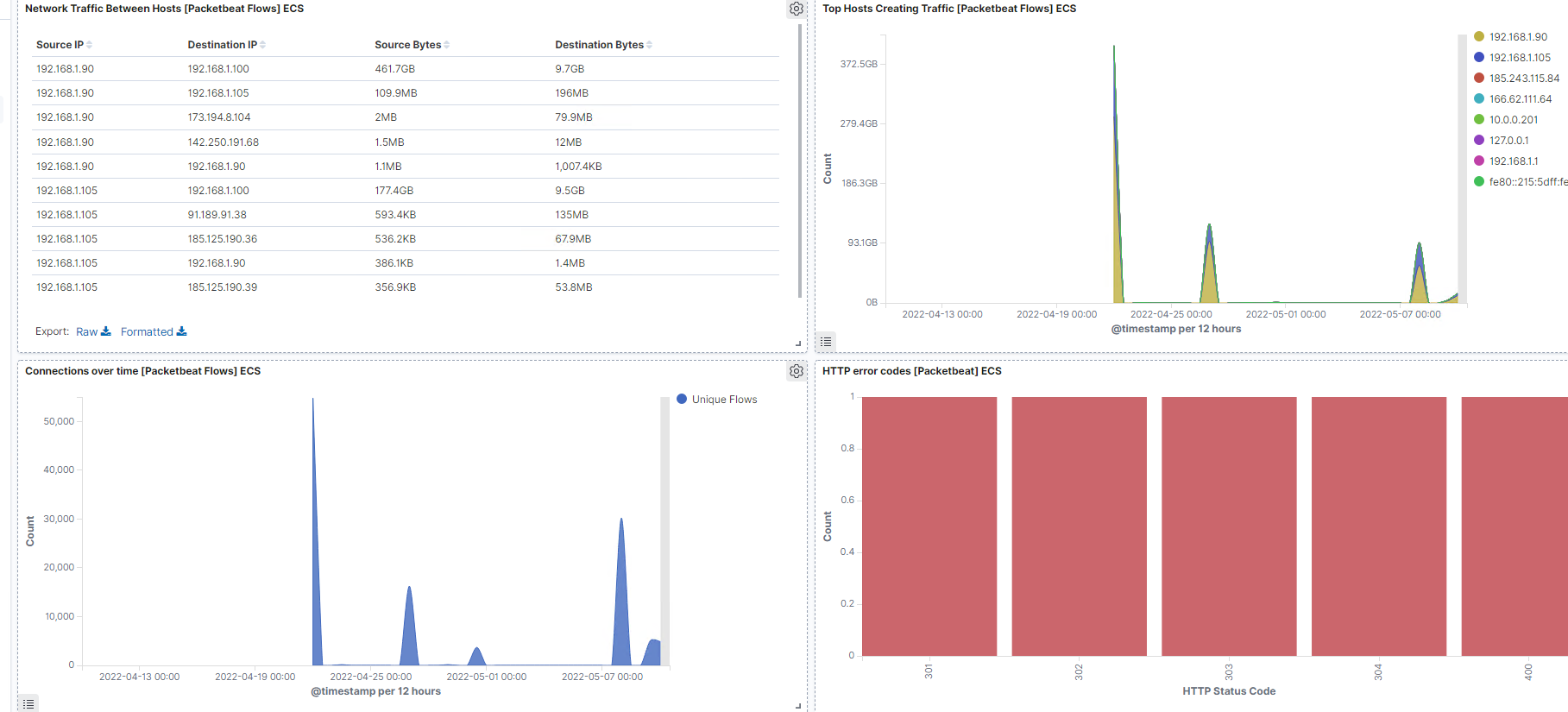
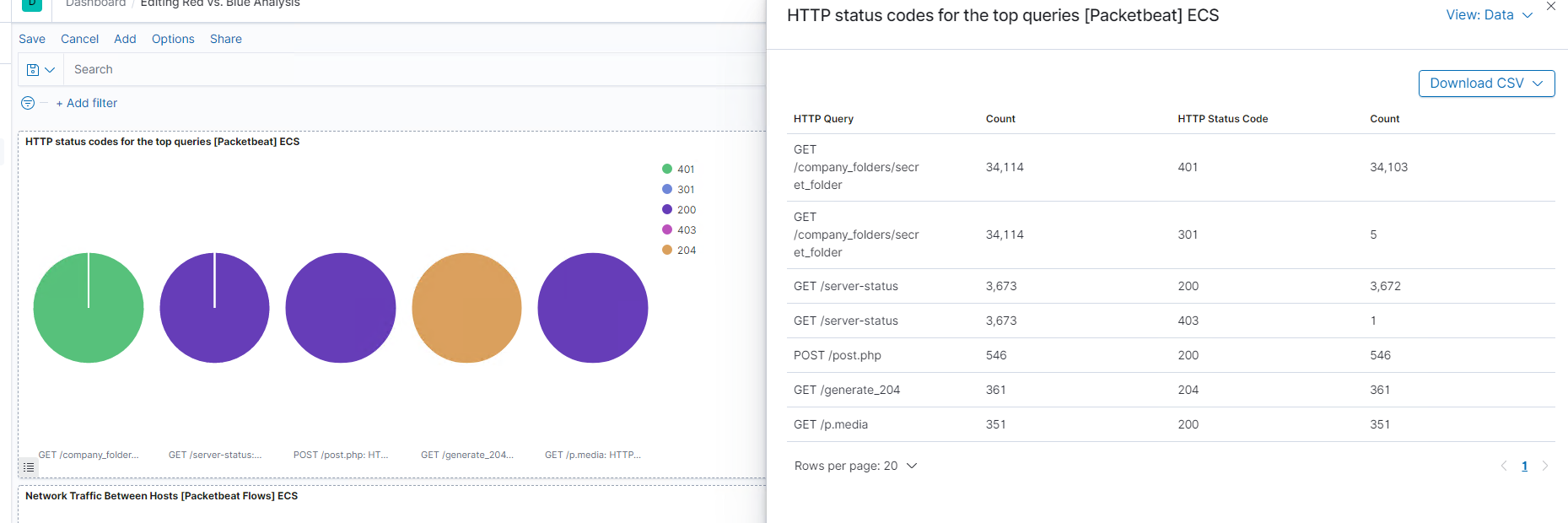
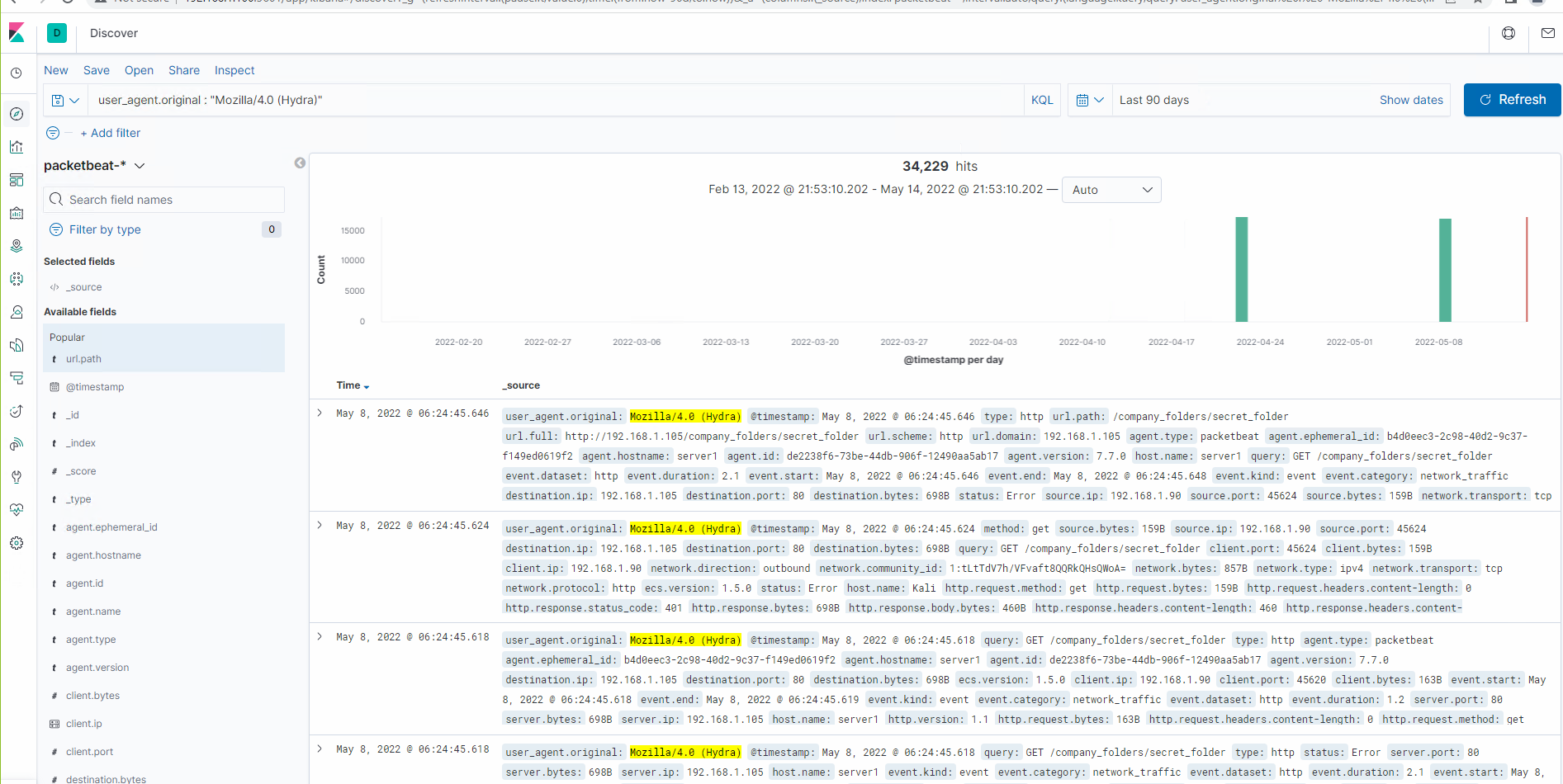
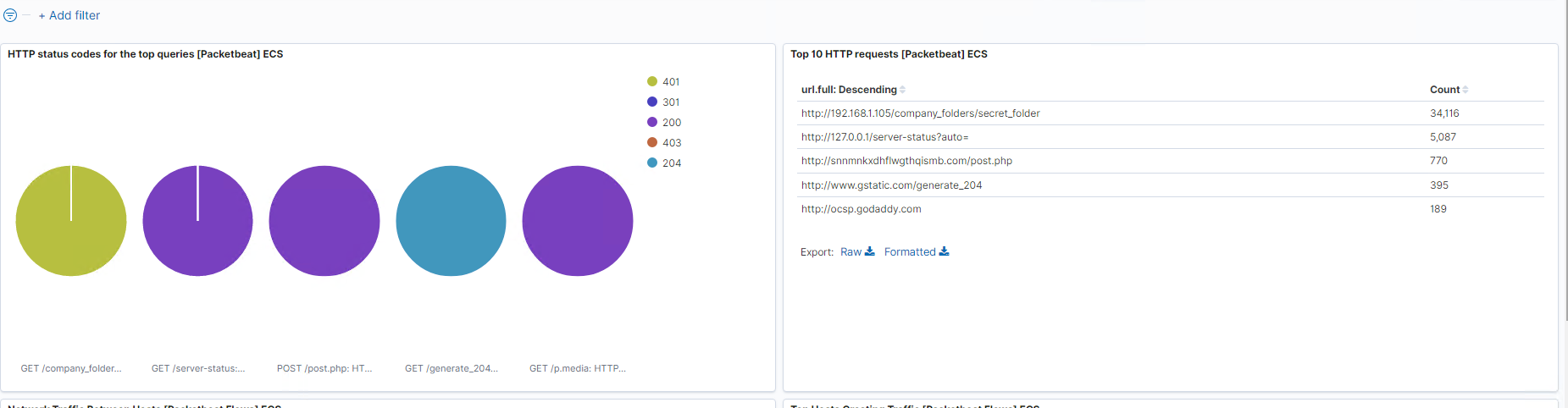
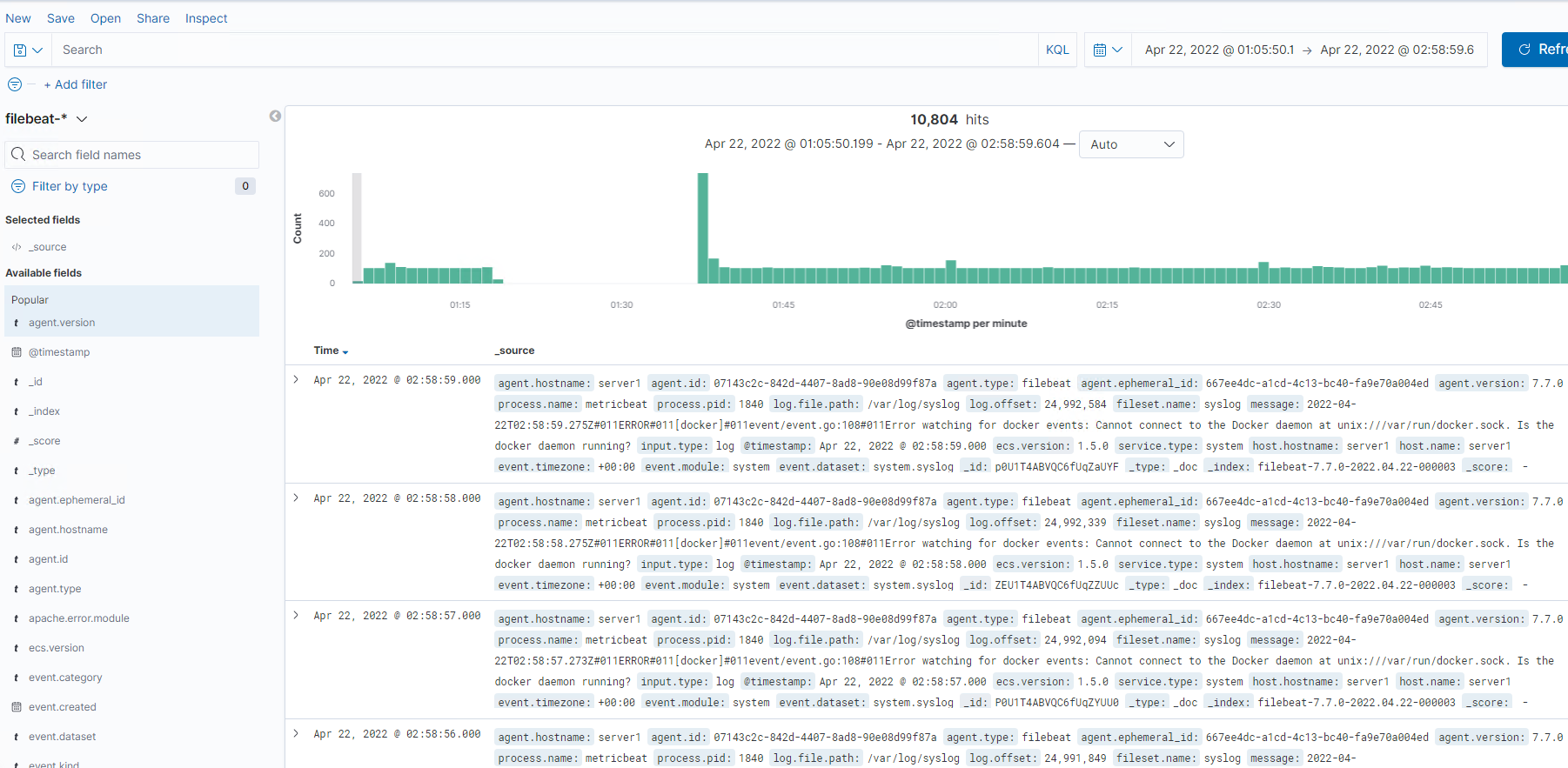
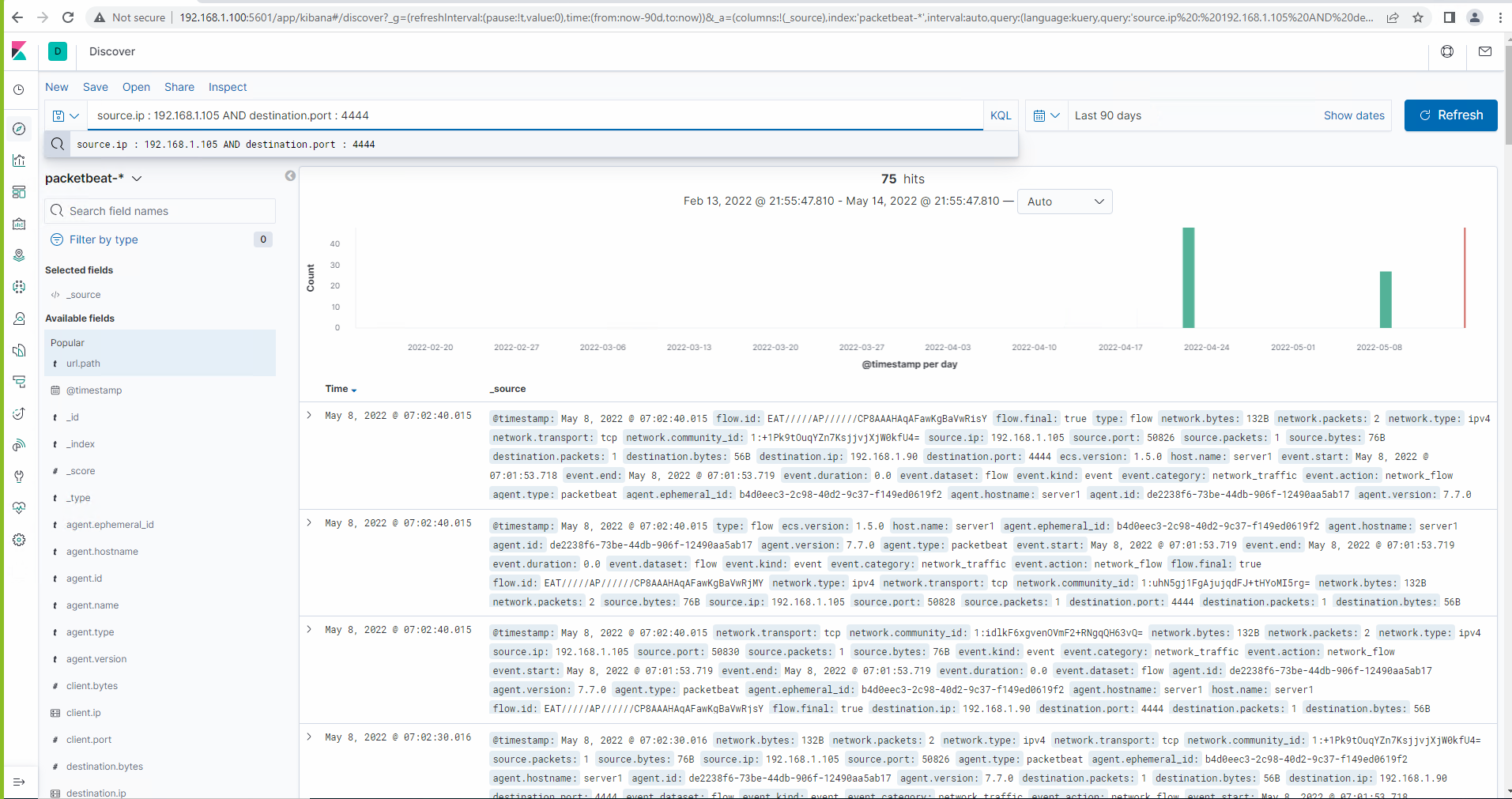
****

The remaining steps will be a process of self-discovery to be completed without screen shot examples.

Get familiar with running search queries in the Discover screen with Packetbeat. This will be located on your fourth tab in Chrome.

* On the Discover page, locate the search field.
* Start typing source and notice the suggestions that come up.
* Search for the source.ip of your attacking machine.
* Use AND and NOT to further filter you search and look for communications between your attacking machine and the victim machine.
* Other things to look for:
  + url
  + status\_code
  + error\_code

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? **Apr 22, 2022 @ 03:00**
     + What responses did the victim send back? **Status\_code 200-207**
     + What data is concerning from the Blue Team perspective? An influx of requests took place around the same time which is suspicious 
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)?34,105 requests were made. & 03:00 && 192.168.1.90
     + Which files were requested? What information did they contain? Company\_folders/secret\_folder && http://192.168.1.105/company\_folders/secret\_folder
     + What kind of alarm would you set to detect this behavior in the future? I would put a limitation on the number of request allowed per hour and failed login attempts (Account Lockout)
     + Identify at least one way to harden the vulnerable machine that would mitigate this attack. Ensure Strict Access control and implement authentication protection. Strong passwords, account restrictions,
3. Identify the brute force attack.user\_agent.original : "Mozilla/4.0 (Hydra)" 
   * After identifying the hidden directory, you used Hydra to brute-force the target server. Answer the following questions: **Hydra is a tool that sends out HTTP packets in a brute force manner generally to enumerate specific parts of a server**
     + Can you identify packets specifically from Hydra? **A large amount of HTTP request (packets) were noticed **
     + How many requests were made in the brute-force attack? **34,229**
     + How many requests had the attacker made before discovering the correct password in this one? **34,228**
     + What kind of alarm would you set to detect this behavior in the future and at what threshold(s)? I would implement an alert for a brute force attack recognizing many failed login attempts (weak passwords) , etc
     + Identify at least one way to harden the vulnerable machine that would mitigate this attack.
4. Find the WebDav connection. (url.path : “/webdav”  
   url.path: "/webdav"
   * Use your dashboard to answer the following questions:
     + How many requests were made to this directory?
     + Which file(s) were requested?
     + What kind of alarm would you set to detect such access in the future?
     + Identify at least one way to harden the vulnerable machine that would mitigate this attack.
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. I was able to identify traffic coming from source ip 192.168.1.105 and open port 4444 that was set in a meterpreter session on the attack machine.**
     + What kinds of alarms would you set to detect this behavior in the future? Put an alert for **Time** and **Size** aspects. If an influx of packets are being sent/received in short amount of time that’s a flag. Or, if the size of a packet is larger than normal that could raise some flags as well. You also want to make sure to analyzes the relationship between file descriptors and the different logs.
     + Identify at least one way to harden the vulnerable machine that would mitigate this attack. **Block certain ports ( ex. 80, 443, etc)**