

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 Apache logs

Click on Add Log Data

 **Observability**

APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Add APM

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

Metrics


Collect metrics from the operating system and services running on your servers.


Add metric data


Click on Apache logs

Add Data to Kibana

All **Logs** Metrics SIEM Sample data

 **ActiveMQ logs**
Collect ActiveMQ logs with Filebeat.

 **Apache logs**
Collect and parse access and error logs created by the Apache HTTP server.

 **AWS Cloudwatch logs**
Collect Cloudwatch logs with Functionbeat.

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

 **Module status**

Check that data is received from the Filebeat `apache` module

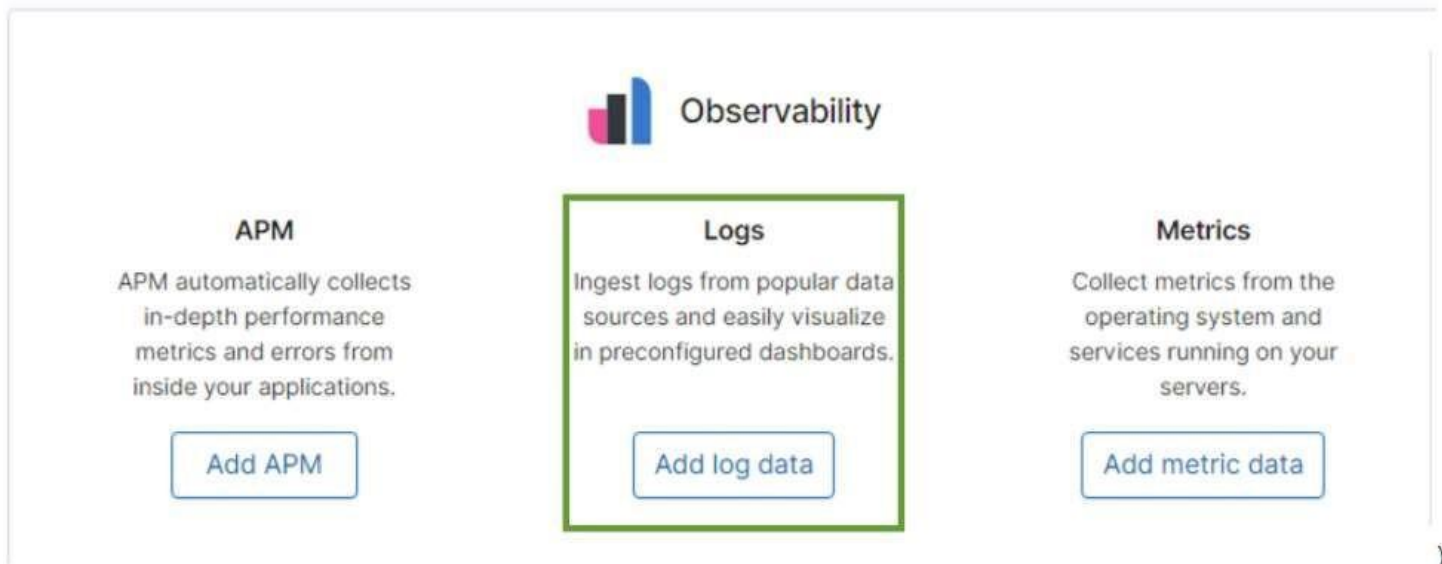
Check data

Data successfully received from this module

Return to the Home screen by moving back 2 pages.

Adding System Logs

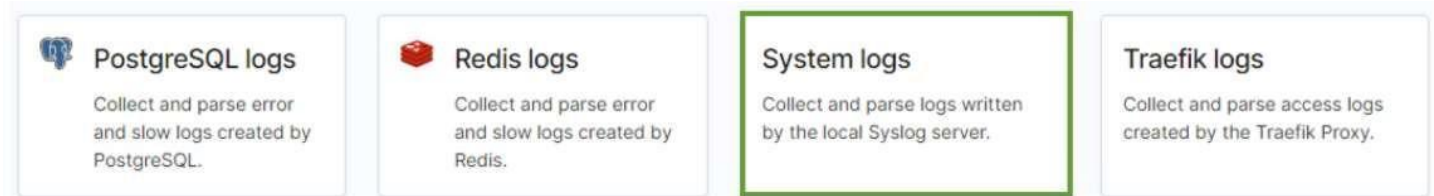
Click on Add Log Data



The screenshot shows the 'Observability' dashboard with three main sections: APM, Logs, and Metrics. The 'Logs' section is highlighted with a green border. Each section has a description and an 'Add' button.

APM	Logs	Metrics
APM automatically collects in-depth performance metrics and errors from inside your applications.	Ingest logs from popular data sources and easily visualize in preconfigured dashboards.	Collect metrics from the operating system and services running on your servers.
Add APM	Add log data	Add metric data

Click on System logs



This screen shows four log source options: PostgreSQL logs, Redis logs, System logs (highlighted with a green border), and Traefik logs. Each option includes a description of the log type.

PostgreSQL logs	Redis logs	System logs	Traefik logs
Collect and parse error and slow logs created by PostgreSQL.	Collect and parse error and slow logs created by Redis.	Collect and parse logs written by the local Syslog server.	Collect and parse access logs created by the Traefik Proxy.

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



The 'Module status' section shows a check for data received from the Filebeat 'system' module. A green message bar indicates success.

✓ **Module status**

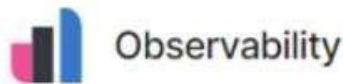
Check that data is received from the Filebeat `system` module [Check data](#)

Data successfully received from this module

Return to the Home screen by moving back 2 pages.

Adding Apache Metrics

Click on Add Metric Data



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

[Add APM](#)

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

[Add log data](#)

Metrics

Collect metrics from the operating system and services running on your servers.

[Add metric data](#)

Click on Apache Metrics

Add Data to Kibana

All Logs **Metrics** SIEM Sample data



ActiveMQ metrics

Fetch monitoring metrics from ActiveMQ instances.



Aerospike metrics

Fetch internal metrics from the Aerospike server.



Apache metrics

Fetch internal metrics from the Apache 2 HTTP server.

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



Module status

Check that data is received from the Metricbeat `apache` module

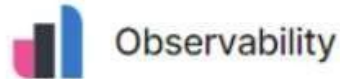
[Check data](#)

Data successfully received from this module

Return to the Home screen by moving back 2 pages.

Adding System Metrics

Click on Add Metric Data



APM

APM automatically collects in-depth performance metrics and errors from inside your applications.

Add APM

Logs

Ingest logs from popular data sources and easily visualize in preconfigured dashboards.

Add log data

Metrics

Collect metrics from the operating system and services running on your servers.

Add metric data

Click on System Metrics

STAN metrics
Fetch monitoring metrics from the STAN server.

System metrics
Collect CPU, memory, network, and disk statistics from the host.

Traefik metrics
Fetch monitoring metrics from Traefik.

Uptime Monitors
Monitor services for their availability

uWSGI metrics
Fetch internal metrics from the uWSGI server.

vSphere metrics
Fetch internal metrics from vSphere.

Windows metrics
Fetch internal metrics from Windows.

Zookeeper metrics
Fetch internal metrics from a Zookeeper server.

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

Module status

Check that data is received from the Metricbeat `system` module

Check data

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.

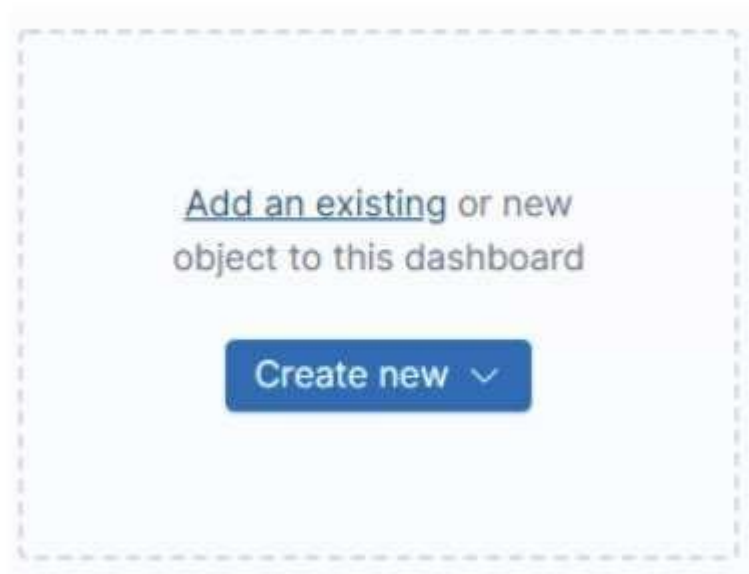
Click on Create dashboard in the upper right hand side.



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:



Add panels



HTTP status



Sort

Types

4



Http Status over time [Filebeat AWS]



HTTP Status Codes [Metricbeat CouchDB] ECS

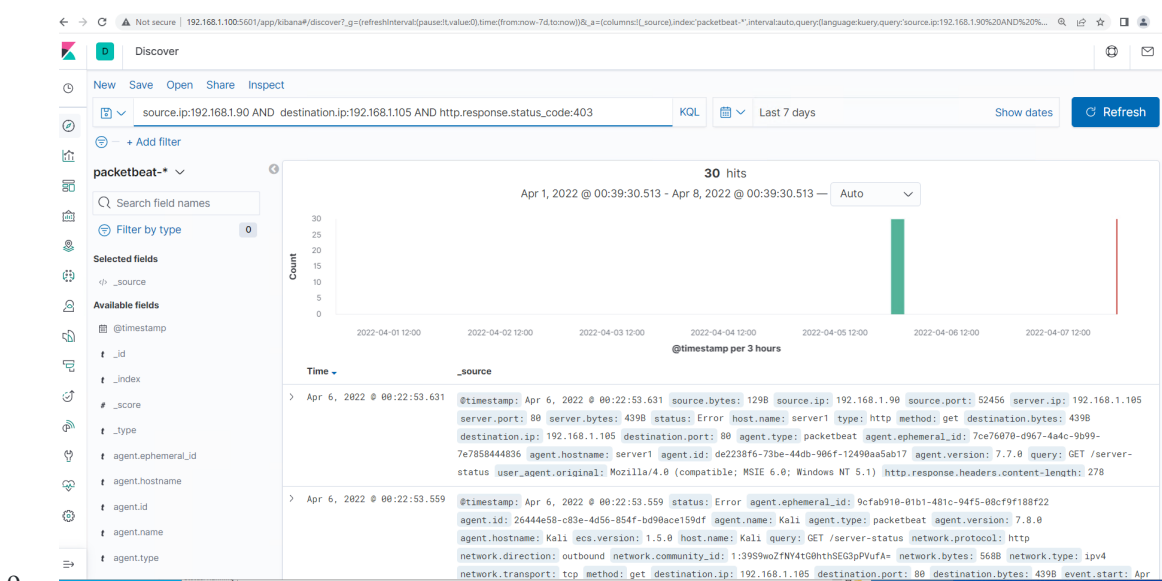


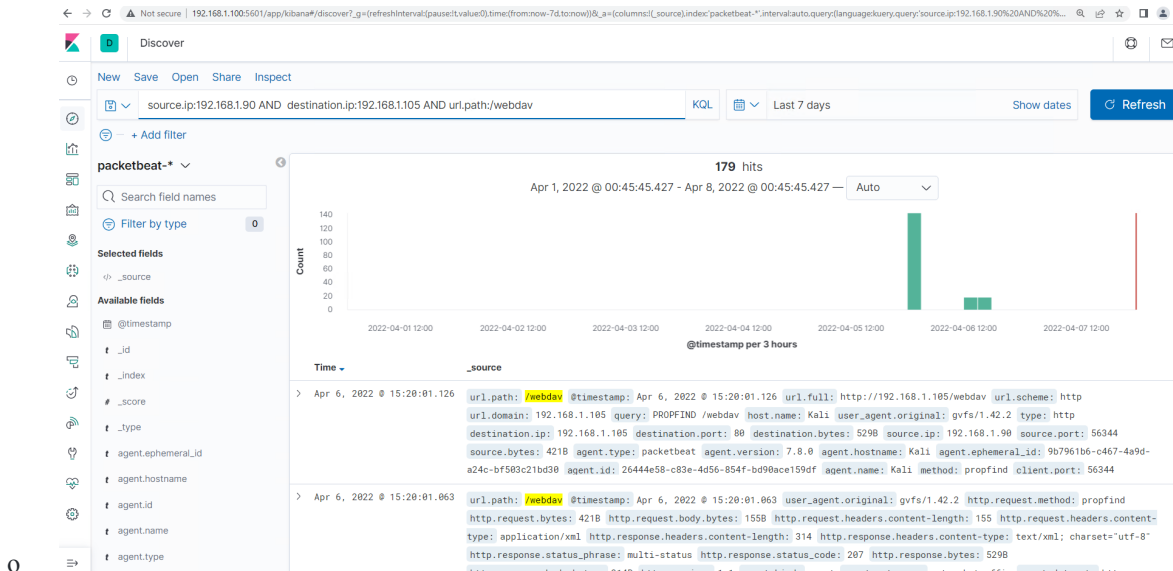
HTTP status codes for the top queries [Packetbeat] ECS

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:
 - o url o status_code o 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.

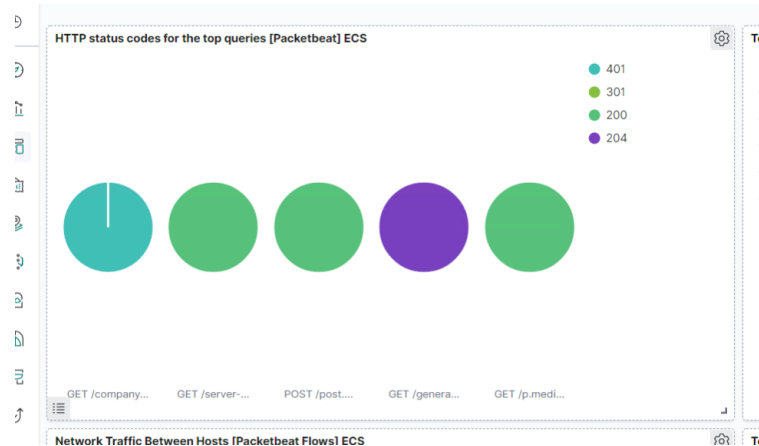
- Identify the traffic between your machine and the web machine:

When did the interaction occur?

- April 6, 2022 between 00:00 & 03:00

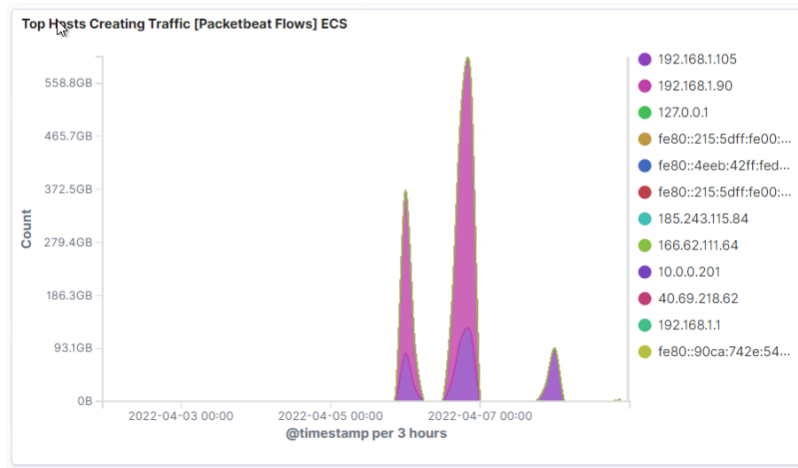
What responses did the victim send back?

- 401, 403, 201, 200, 204



What data is concerning from the Blue Team perspective?

- The access allowed on 200, the forbidden 403 and 401, the massive visible rise in traffic that we notice, especially from one source



o

2. Find the request for the hidden directory.

- o 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)?

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,434
http://127.0.0.1/server-status?auto=	6,749
http://snnmnkxdhflwgtqismb.com/post.php	735
http://www.gstatic.com/generate_204	371
http://ocsp.godaddy.com	162

Export: Raw Formatted

Which files were requested? What information did they contain?

```
http.response.headers.content-type text/html; charset=iso-8859-1
```

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

Alarm for attempted connections within an hour

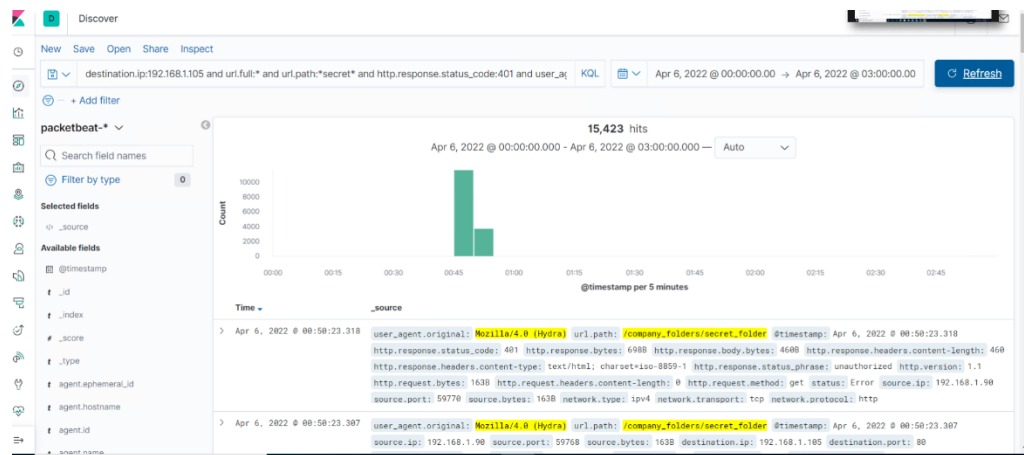
Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Complex password, filename obscuration with care, alarm when accessed outside of regular hours

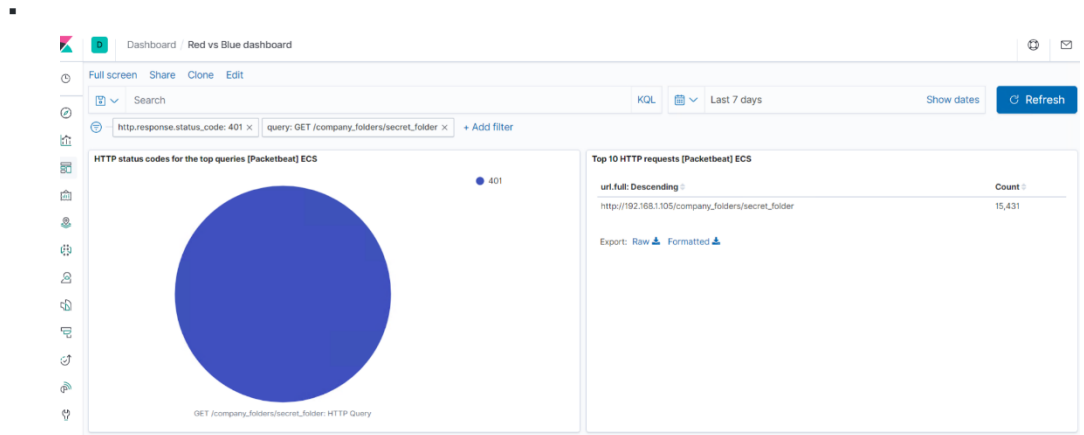
3. Identify the brute force attack.

- o 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?



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



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

15,423

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

Set an alarm for 401 error code going above the set threshold & to send an alert if so

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

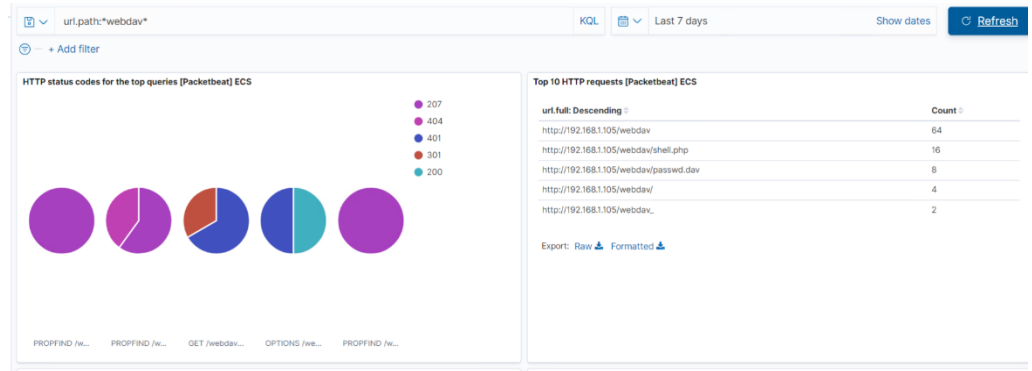
Create a difficult password

4. Find the WebDav connection.

- o Use your dashboard to answer the following questions:

How many requests were made to this directory?

• 64



Which file(s) were requested?
shell.php, passwd.dav

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

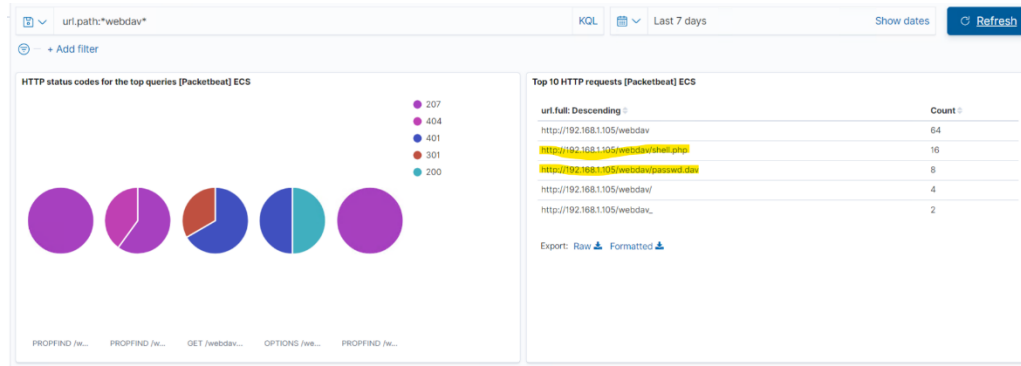
Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Complex passwords, patch machine again latest vulnerabilities

5. Identify the reverse shell and meterpreter traffic.

- o To finish off the attack, you uploaded a PHP reverse shell and started a meterpreter shell session. Answer the following questions:

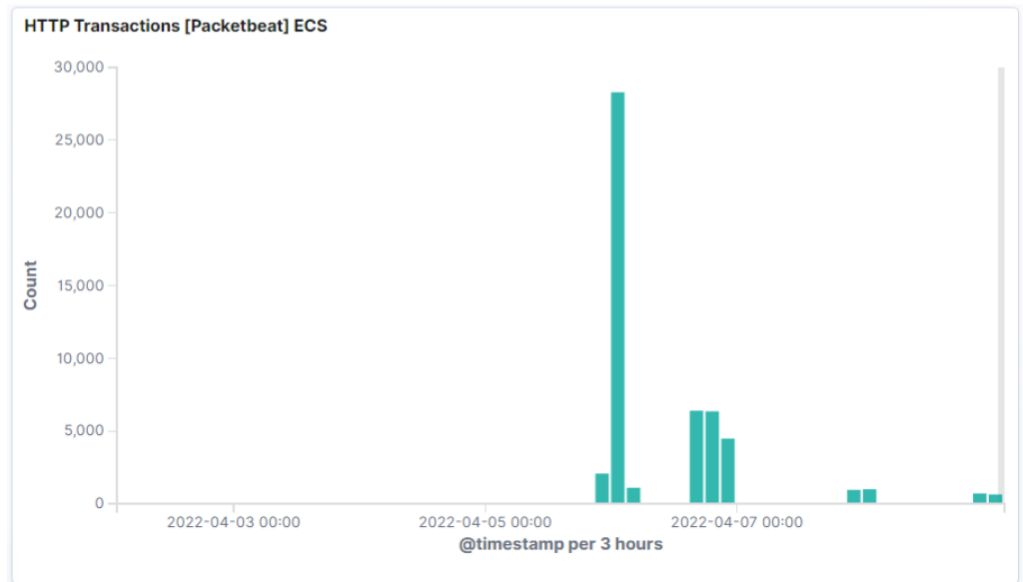
Can you identify traffic from the meterpreter session?



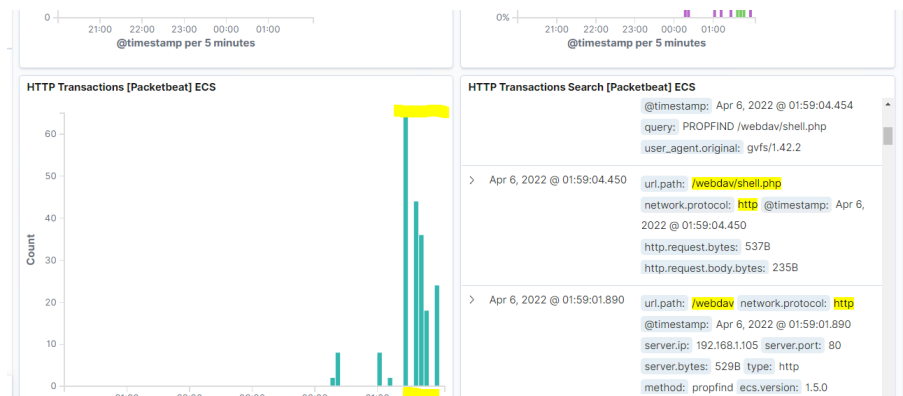
http://192.168.1.105/webdav/shell.php

http://192.168.1.105/webdav/passwd.dav

- Started by examining the php file access



O



O

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

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

