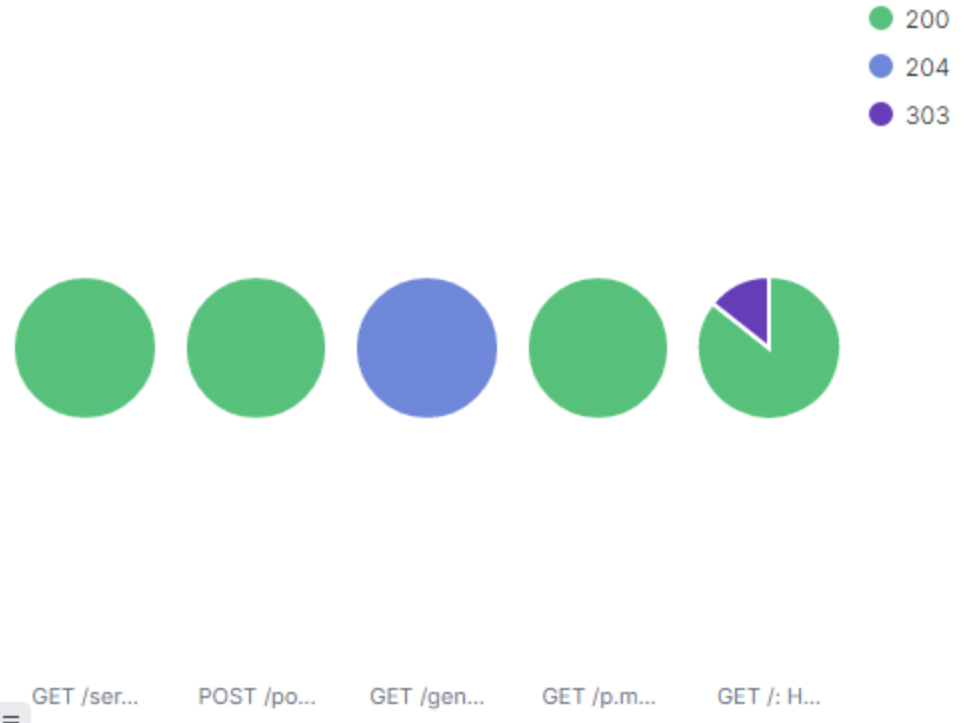
HTTP status codes for the top queries [Packetbeat] ECS



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://127.0.0.1/server-status?auto=	89
http://snnmnkxdhflwgthqismb.com/post.php	14
http://www.gstatic.com/generate_204	7
http://ocsp.godaddy.com	3
http://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js	3

Export: Raw Formatted

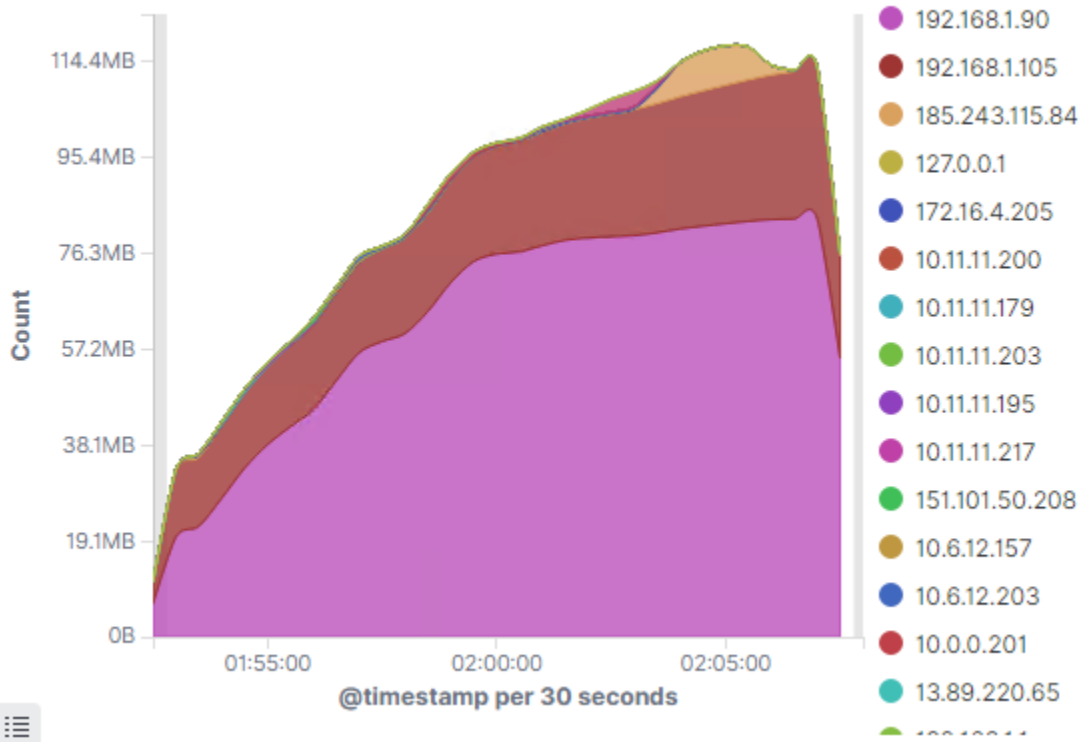
Network Traffic Between Hosts [Packetbeat Flows] ECS

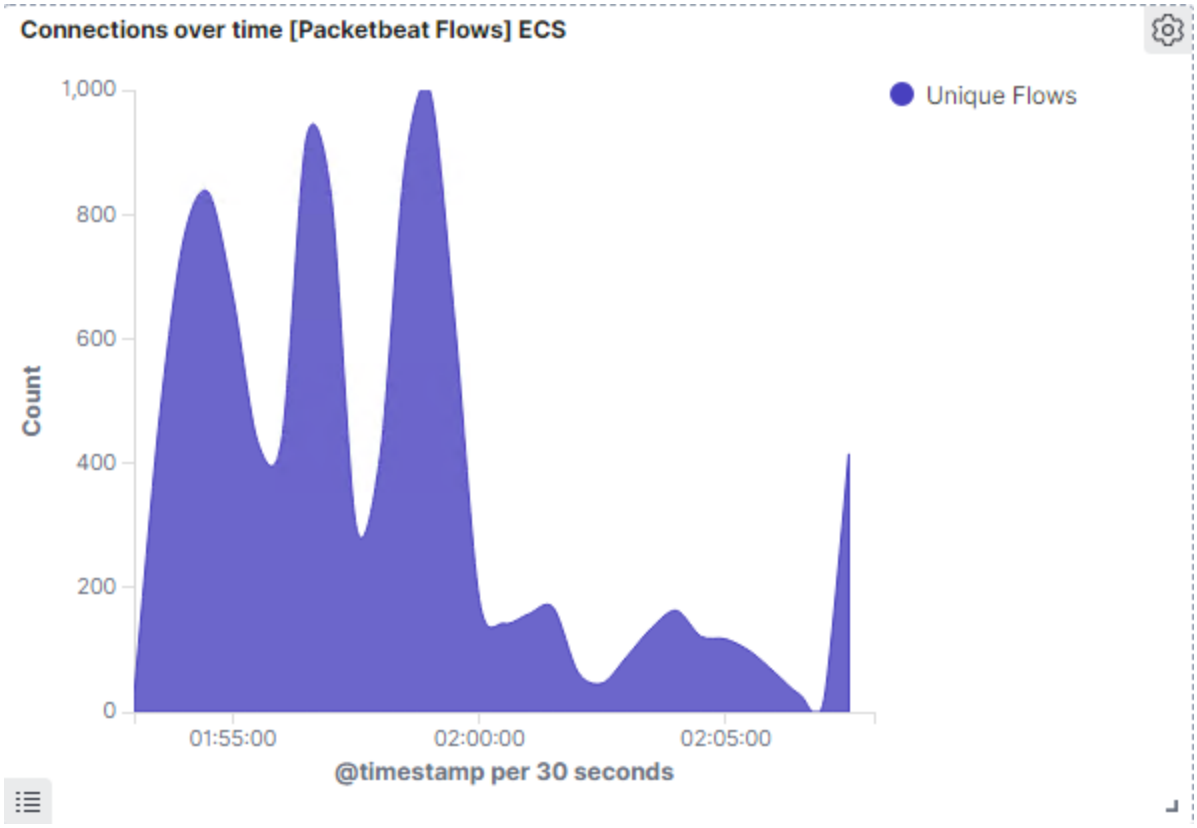
192.168.1.90	35.82.87.100	25.7KB	64.8KB
192.168.1.90	108.138.159.37	24.3KB	245.6KB
192.168.1.90	72.21.91.29	21.3KB	23.5KB
192.168.1.90	35.244.181.201	17.1KB	11.5KB
192.168.1.105	192.168.1.100	642MB	52.3MB
192.168.1.105	91.189.94.4	736B	736B
185.243.115.84	172.16.4.205	35.8MB	102.8MB
166.62.111.64	172.16.4.205	8.2MB	146.8KB
10.0.0.201	64.187.66.143	1.1MB	26.2MB

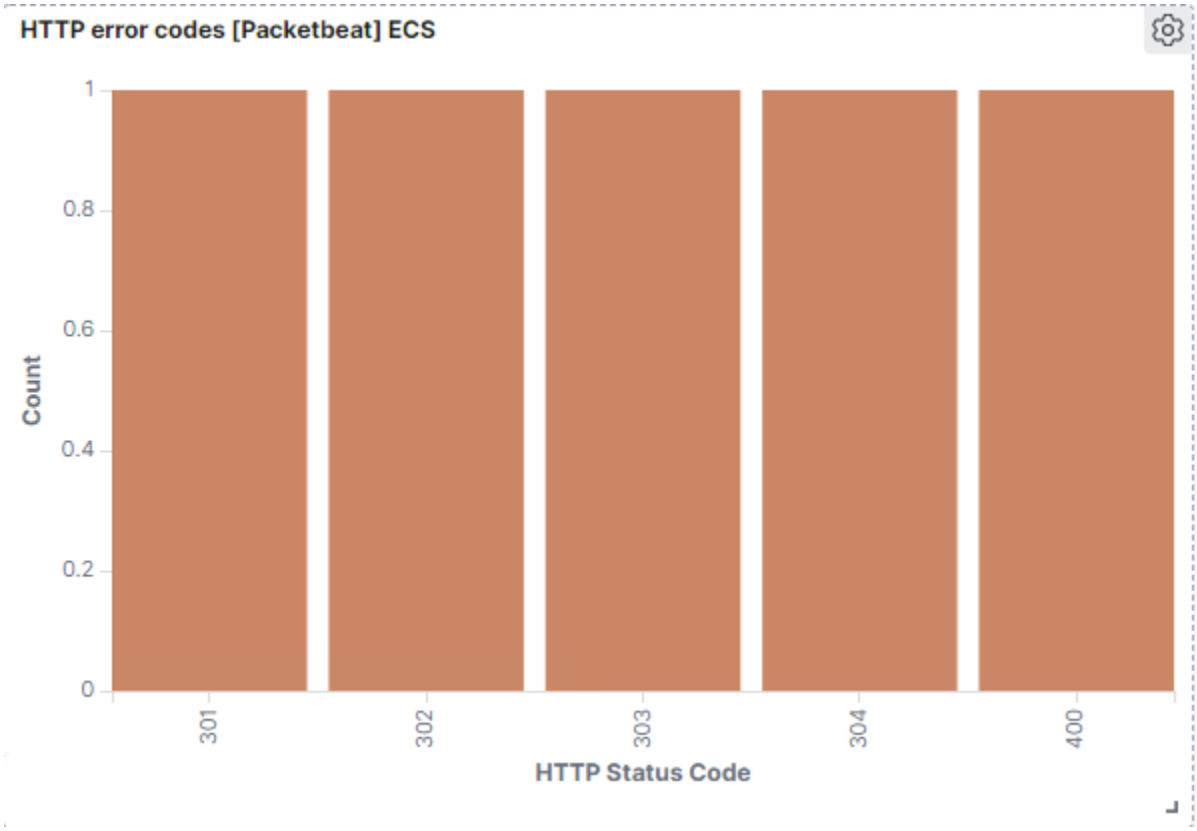
Export: Raw Formatted

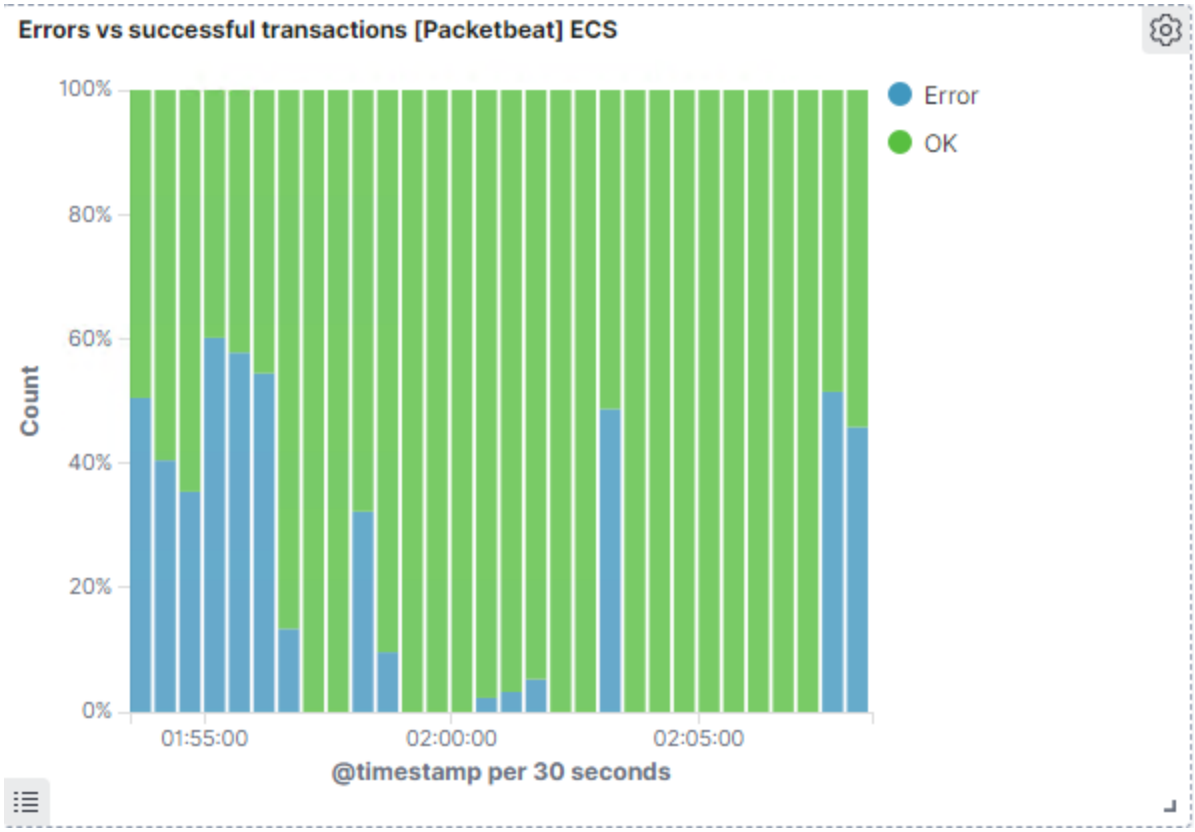
1 2 »

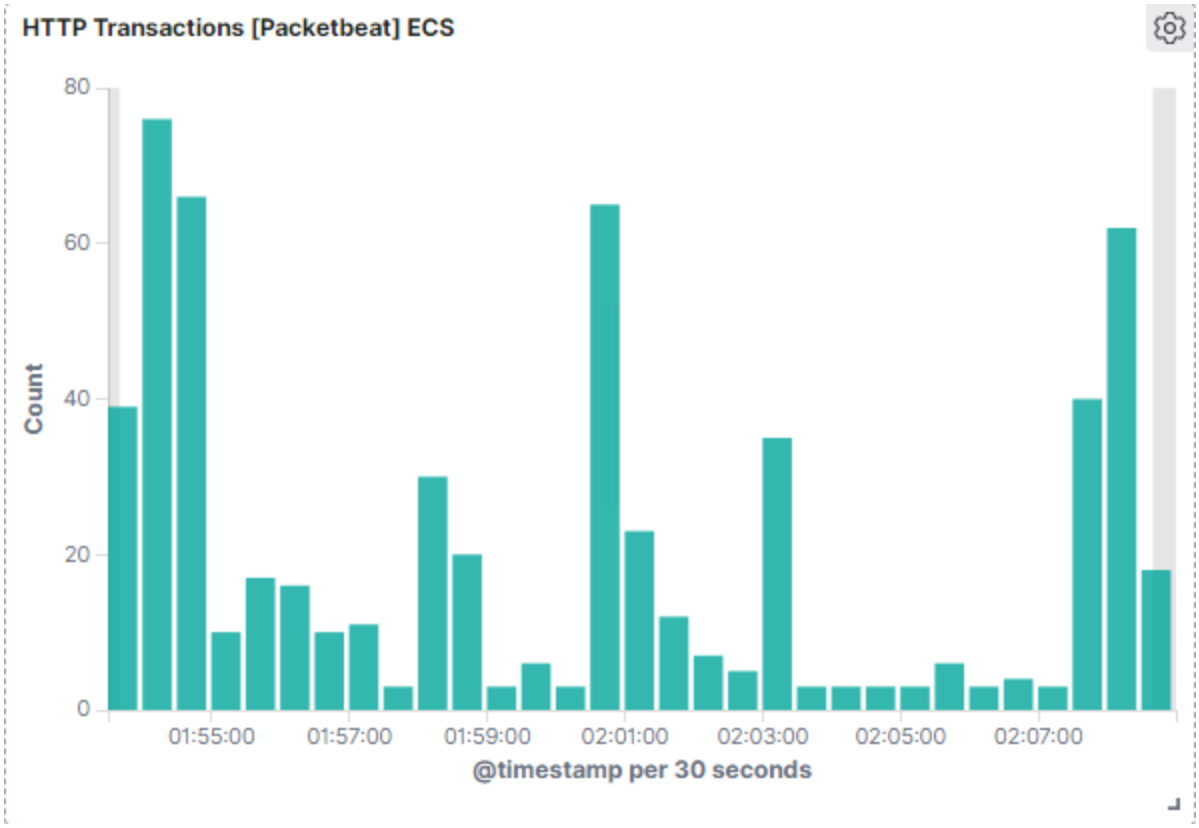
Top Hosts Creating Traffic [Packetbeat Flows] ECS




Connections over time [Packetbeat Flows] ECS
Unique Flows
1,000
800
600
400
200
0
Count
01:55:00
02:00:00
02:05:00
@timestamp per 30 seconds


HTTP error codes [Packetbeat] ECS
1
0.8
0.6
0.4
0.2
0
Count
301
302
303
304
400
HTTP Status Code


Errors vs successful transactions [Packetbeat] ECS
100%
80%
60%
40%
20%
0%
Count
01:55:00
02:00:00
02:05:00
@timestamp per 30 seconds
Error
OK



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

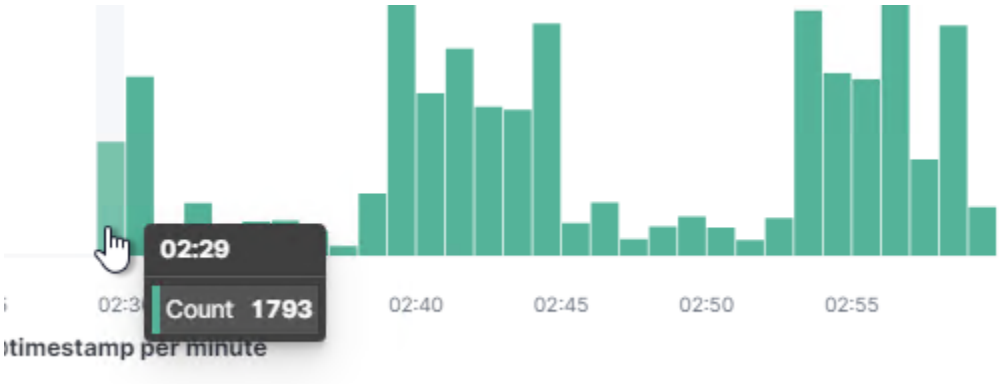
- On the Discover page, locate the search field.
- Start typing source and notice the suggestions that come up.
- Search for the source.ip of your attacking machine.
- Use AND and NOT to further filter you search and look for communications between your attacking machine and the victim machine.
- Other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

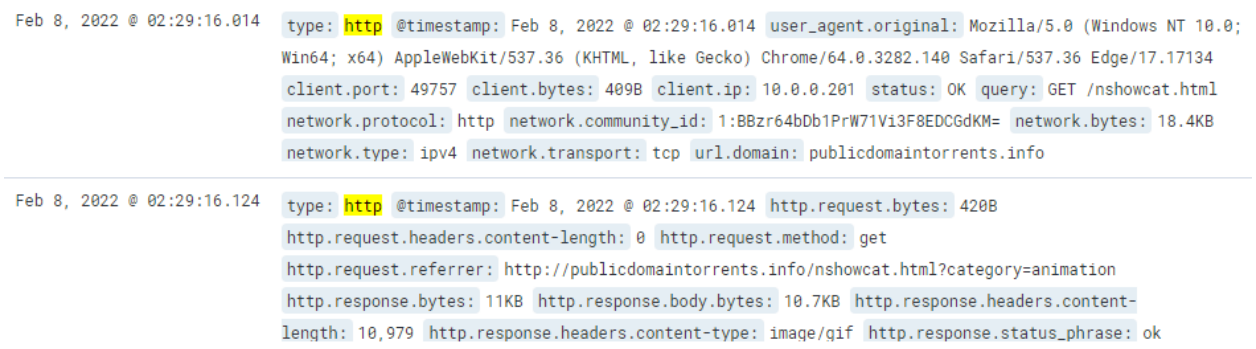
- When did the interaction occur?



The first significant activity spike occurred on **Feb 8, 2022 at 02:29:00**

- What responses did the victim send back?

HTTP responses were sent back

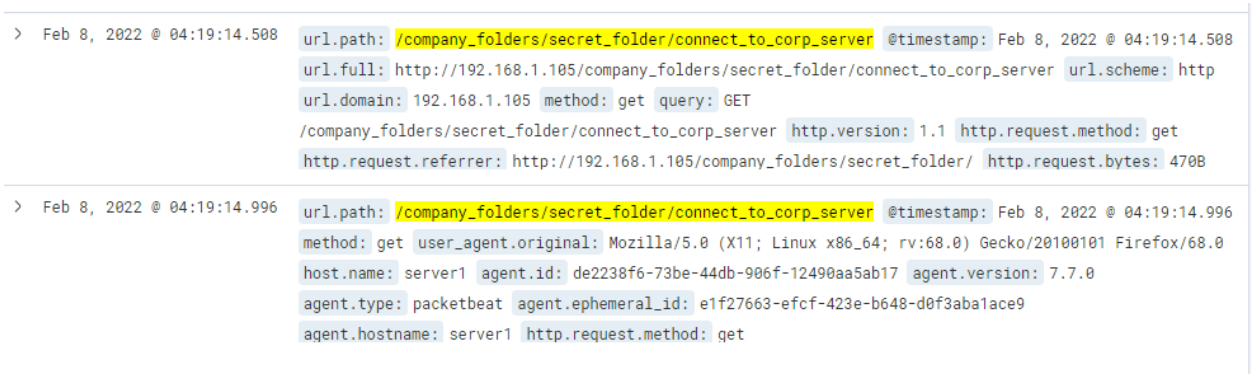


- What data is concerning from the Blue Team perspective?

High quantities of http requests, more than normal, all stemming from agent.name of Kali

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)?



2 requests and both from **192.168.1.90 at 04:19:14**

url.path : "/company_folders/secret_folder/connect_to_corp_server"

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

Information contained was the username and password hash of the CEO as well as instructions on how to upload files to the company's server.

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

The type of alarm I would set would be for non-whitelisted company IP addresses that are accessing this folder.

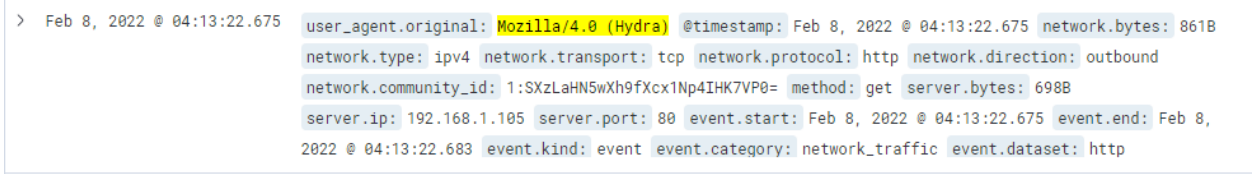
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Whitelisting

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?

source.ip : 192.168.1.90 AND user_agent.original :"Mozilla/4.0 (Hydra)" AND http.response.status_code : 401



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

16,658 hits from HYDRA overall

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

16,656 hits from HYDRA were failures

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

You would set an alarm detects multiple requests within milliseconds and minimum threshold of 5 or more requests within that millisecond

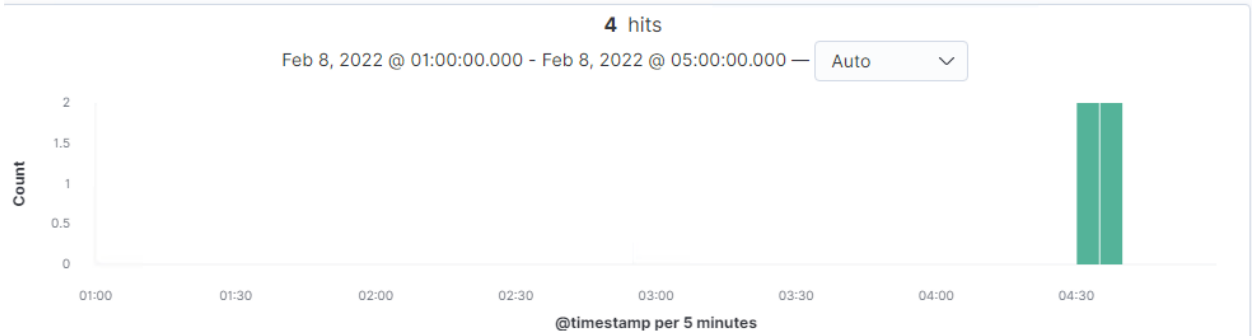
- Identify at least one way to harden the vulnerable machine that would mitigate this attack.

Timeout after multiple of the same requests

4. Find the WebDav connection.

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

4 requests were made to this directory



Time	_source
Feb 8, 2022 @ 04:34:50.091	url.path: /webdav/ @timestamp: Feb 8, 2022 @ 04:34:50.091 method: get event.dataset: http event.duration: 0.8 event.start: Feb 8, 2022 @ 04:34:50.091 event.end: Feb 8, 2022 @ 04:34:50.091 event.kind: event event.category: network_traffic status: Error host.name: Kali agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: 66669a31-be8f-4ba8-bf3b-87d58546ae0a agent.id: 26444e58-c83e-4d56-854f-bd90ace159df agent.name: Kali agent.type: packetbeat query: GET /webdav/
Feb 8, 2022 @ 04:34:50.652	url.path: /webdav/ @timestamp: Feb 8, 2022 @ 04:34:50.652 type: http client.ip: 192.168.1.90 client.port: 47874 client.bytes: 320B host.name: server1 agent.hostname: server1 agent.id: de2238f6-73be-44db-906f-12490aa5ab17 agent.version: 7.7.0 agent.type: packetbeat agent.ephemeral_id: e1f27663-efcf-423e-b648-d0f3aba1ace9 method: get url.full: http://192.168.1.105/webdav/ url.scheme: http url.domain: 192.168.1.105 destination.port: 80 destination.bytes: 720B destination.ip: 192.168.1.105

- Which file(s) were requested?

Passwd.dav

Feb 8, 2022 @ 04:36:49.720	url.path: /webdav/passwd.dav @timestamp: Feb 8, 2022 @ 04:36:49.720 user_agent.original: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0 url.scheme: http url.domain: 192.168.1.105 url.full: http://192.168.1.105/webdav/passwd.dav status: OK client.ip: 192.168.1.90 client.port: 47880 client.bytes: 408B method: get source.bytes: 408B source.ip: 192.168.1.90 source.port: 47880 http.request.method: get http.request.referrer: http://192.168.1.105/webdav/
Feb 8, 2022 @ 04:36:50.291	url.path: /webdav/passwd.dav @timestamp: Feb 8, 2022 @ 04:36:50.291 user_agent.original: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0 event.start: Feb 8, 2022 @ 04:36:50.291 event.end: Feb 8, 2022 @ 04:36:50.292 event.kind: event event.category: network_traffic event.dataset: http event.duration: 0.9 type: http source.ip: 192.168.1.90 source.port: 47880 source.bytes: 408B client.ip: 192.168.1.90 client.port: 47880 client.bytes: 408B host.name: server1

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

An alarm that would trigger whenever non-whitelisted ip's accessing webdav, even set an alert for when anyone accesses webdav

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

Filter documents on public facing infrastructure to inhibit access to internal sensitive information. Further protecting the CEO's login information.

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?

> Feb 8, 2022 @ 05:17:48.624 url.path: /webdav/payloadshell.php @timestamp: Feb 8, 2022 @ 05:17:48.624 user_agent.original: gvfs/1.42.2 ecs.version: 1.5.0 host.name: Kali agent.name: Kali agent.type: packetbeat agent.version: 7.8.0 agent.hostname: Kali agent.ephemeral_id: 66669a31-be8f-4ba8-bf3b-87d58546ae0a agent.id: 26444e58-c83e-4d56-854f-bd90ace159df method: propfind destination.ip: 192.168.1.105 destination.port: 80 destination.bytes: 491B type: http

> Feb 8, 2022 @ 05:17:48.627 url.path: /webdav/payloadshell.php @timestamp: Feb 8, 2022 @ 05:17:48.627 source.port: 47908 source.bytes: 429B source.ip: 192.168.1.90 server.ip: 192.168.1.105 server.port: 80 server.bytes: 491B user_agent.original: gvfs/1.42.2 query: PROPFIND /webdav/payloadshell.php url.full: http://192.168.1.105/webdav/payloadshell.php url.scheme: http url.domain: 192.168.1.105 http.request.body.bytes: 146B http.request.headers.content-length: 146 http.request.headers.content-

url.path: "/webdav/payloadshell.php"

- 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.