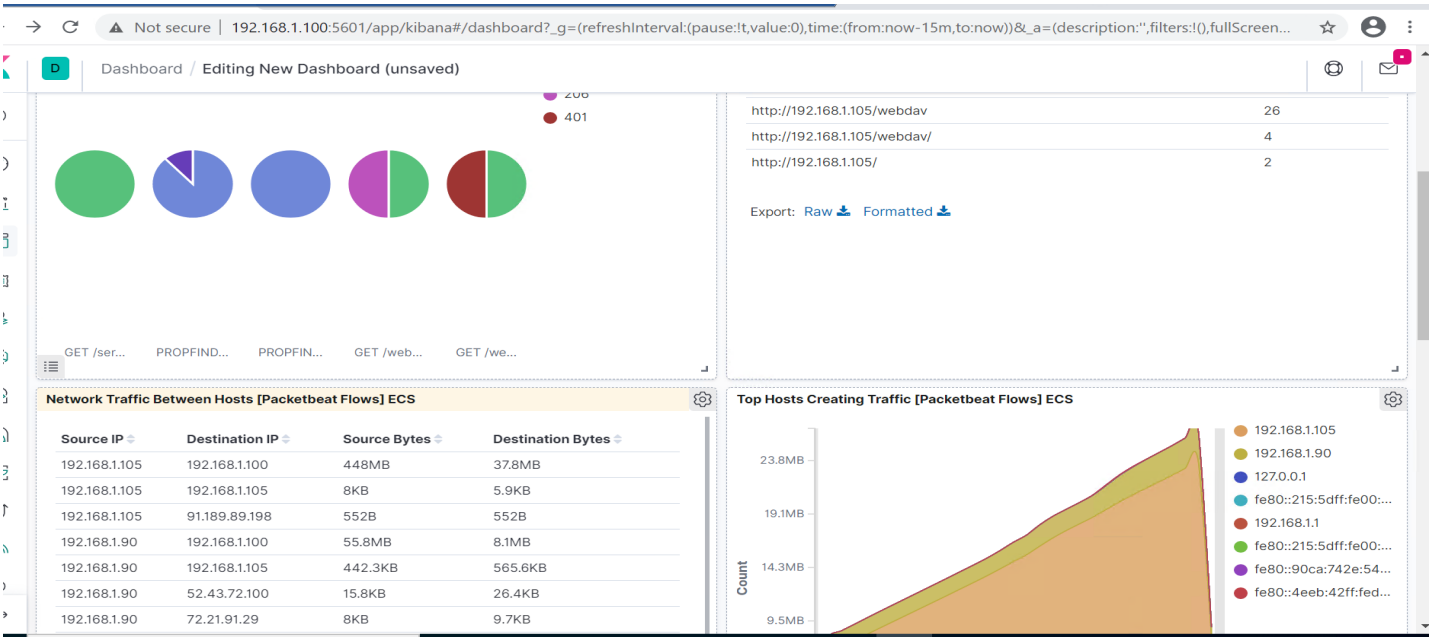
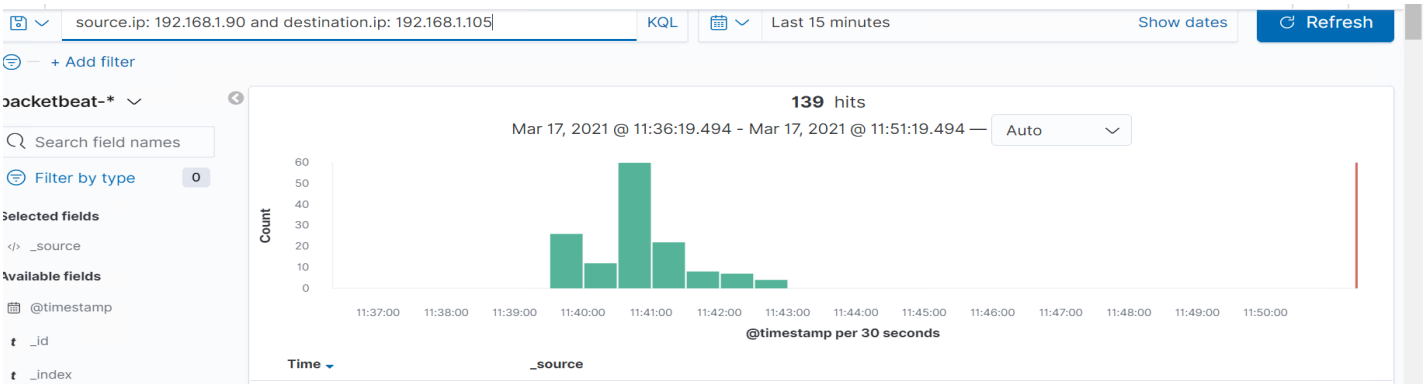
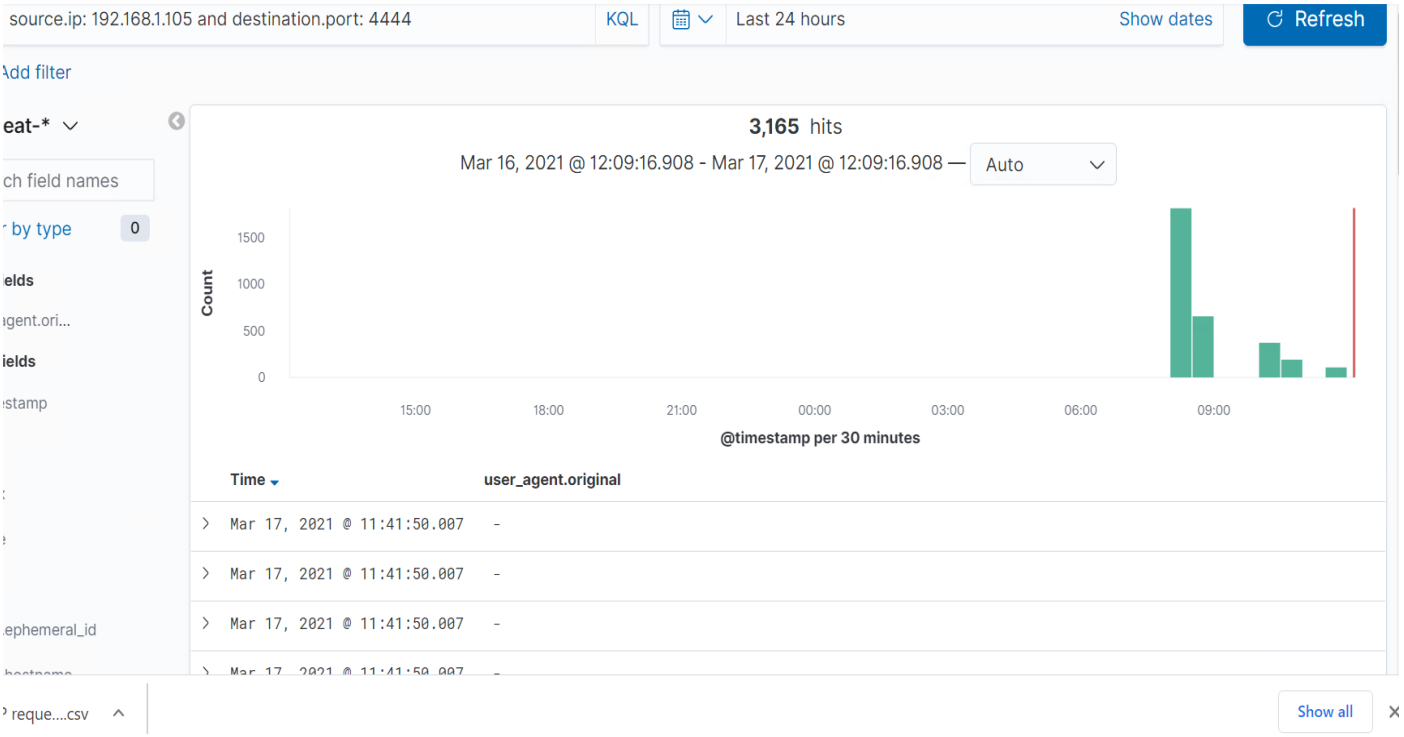
1) Identify the offensive traffic





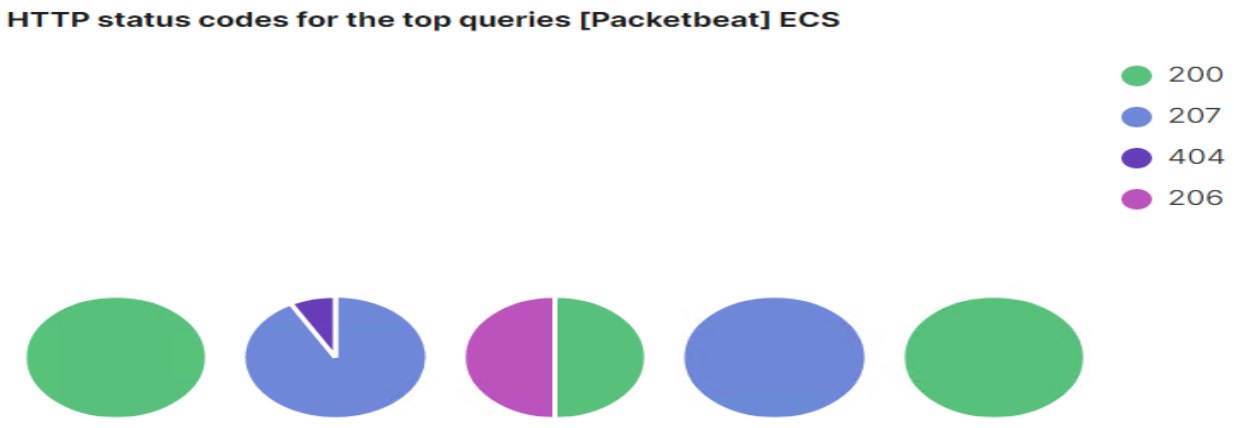


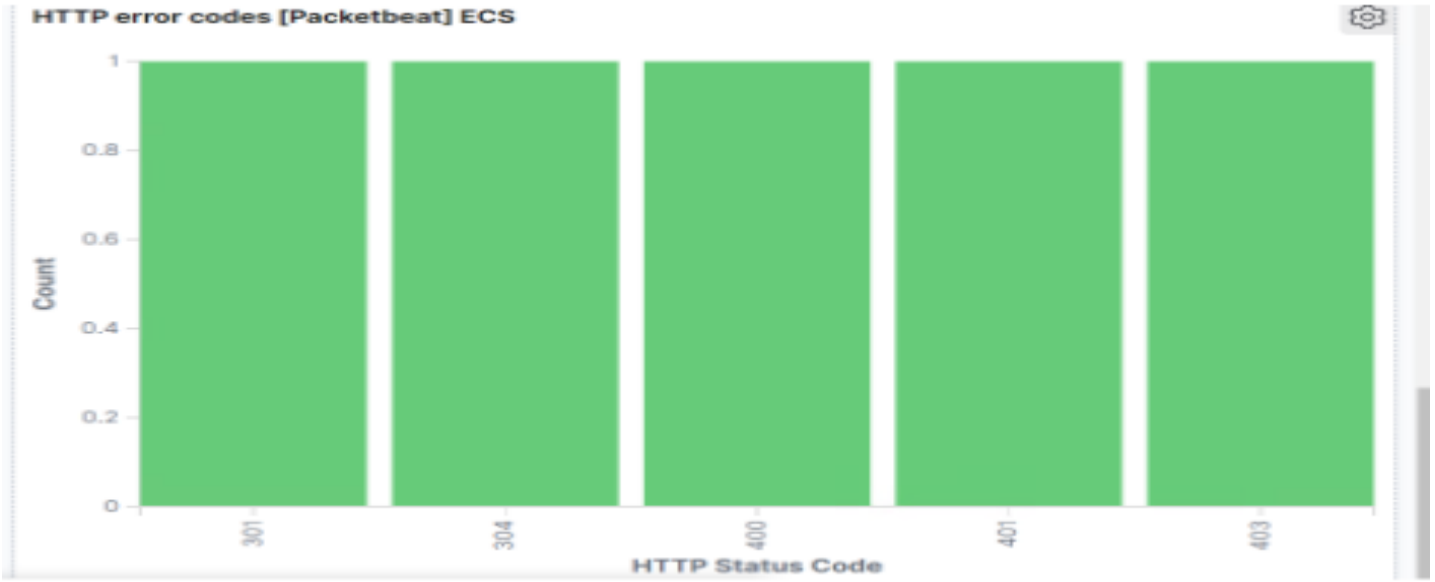
2) Identify the traffic between your machine and the web machine:

When did the interaction occur?

The interaction occurred on May 17th 2021.

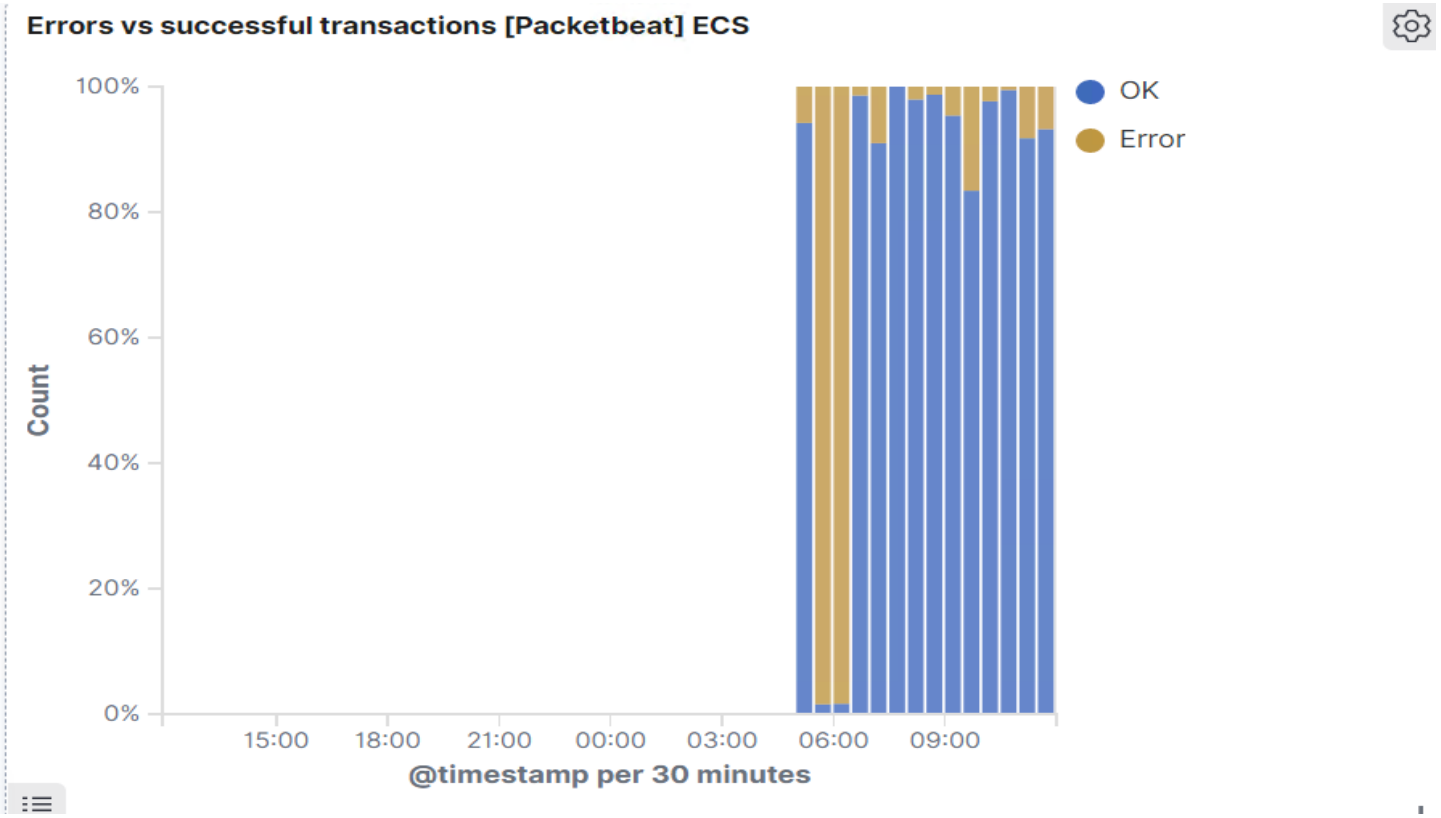
What responses did the victim send back?





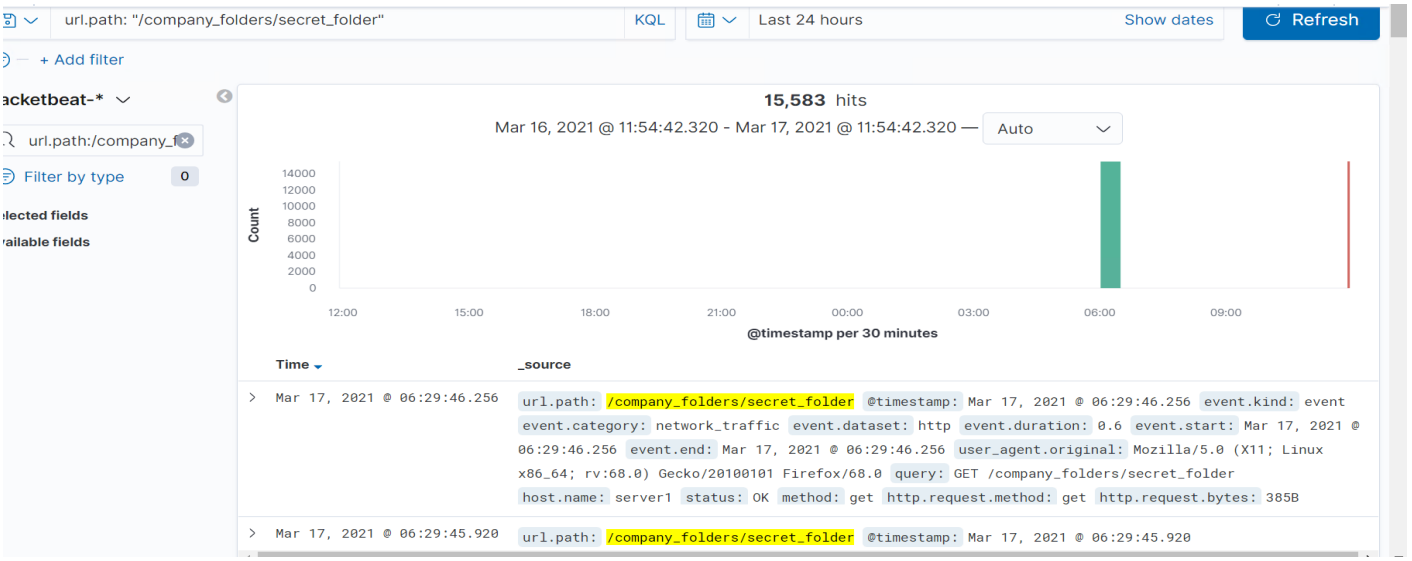
We can see that some of the top responses were 200, 207, 404, 206 and we also see the Http errors codes which were 301,304,400,401,403.

What data is concerning from the Blue Team perspective?



From the blue team perspective, we can see that the successful transactions number was much higher than the error transactions. This should not be the case as an attacking machine should not be able to make successful transactions in the victim's machine.

Find the request for the hidden directory.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,583
http://127.0.0.1/server-status?auto=	2,346
http://192.168.1.105/webdav/shell.php	1,235
http://192.168.1.105/webdav	591
http://192.168.1.105/webdav/passwd.dav	94

Export: Raw Formatted

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)?

They were requested 15,583 times, at around 6am from the ip address 192.168.1.90.

Which files were requested? What information did they contain?

Files such as the secret_folder and files related to the webdav server were requested. These files contained confidential instructions on how to access the company's webdav server.

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

I would set an alert that would notify me if there are any get requests to get access to this secret file.

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

I would suggest deleting the directories and files that contain sensitive information because even if the passwords in the files are hashed, they are easily cracked by using websites such as crackstation.net. This information should be stored

1. Identify the brute force attack.

@timestamp	Mar 17, 2021 @ 06:29:46.256
_id	5ALgPngB97Jw3-P3QXBe
_index	packetbeat-7.7.0-2021.03.17-000002
_score	-
_type	_doc
agent.ephemeral_id	02b39bf0-2a47-4c80-8634-82c8528d6f6a
agent.hostname	server1
agent.id	de2238f6-73be-44db-906f-12490aa5ab17
agent.type	packetbeat
agent.version	7.7.0
client.bytes	385B
client.ip	192.168.1.90
client.port	52818
destination.bytes	626B
destination.ip	192.168.1.105
destination.port	80
network.protocol	http
network.transport	tcp
network.type	ipv4
query	GET /company_folders/secret_folder
server.bytes	626B
server.ip	192.168.1.105
server.port	80
source.bytes	385B
source.ip	192.168.1.90
source.port	52818
status	OK
type	http
url.domain	192.168.1.105
url.full	http://192.168.1.105/company_folders/secret_folder
url.path	/company_folders/secret_folder

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, the packets which are specifically from hydra are under the user_agent.original section.

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

There were 15,583 requests made during the brute-force attack.

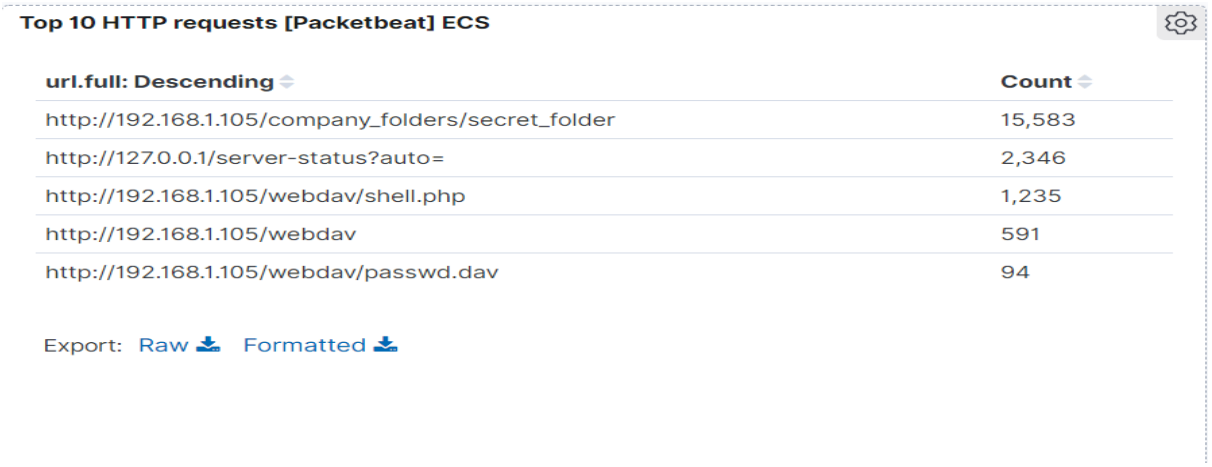
What kind of alarm would you set to detect this behavior in the future and at what threshold(s)?

In the future, I would create an alarm that lets me know if Hydra or any other well-known bruteforce tools are ever included in the user_agent.original section because I know that Hydra is a tool used to make a bruteforce entry. I would set this threshold at 1 per day because you cannot take chances with such tools.

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

One way to harden the vulnerable machine is to limit the amounts of times a person tries to enter and gets the 400-series authorization codes. They could forget and try to guess their password about 5 times and if they are still getting the 400 codes, they would be locked out for some time and would not be able to access the vulnerable server.

Find the WebDav connection.



Top 10 HTTP requests [Packetbeat] ECS

url.full: Descending	Count
http://192.168.1.105/company_folders/secret_folder	15,583
http://127.0.0.1/server-status?auto=	2,346
http://192.168.1.105/webdav/shell.php	1,235
http://192.168.1.105/webdav	591
http://192.168.1.105/webdav/passwd.dav	94

Export: Raw Formatted

Use your dashboard to answer the following questions:

How many requests were made to this directory?

There were about 591 requests made into this directory.

Which file(s) were requested?

The shell.php and the passwd.dav files were requested.

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

We can set an alarm to alert us anytime these files are requested by unrecognized ip addresses.

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

We can whitelist the allowed Ip's on the servers firewall so that only they have access to these files.

Can you identify traffic from the meterpreter session?

Yes, I can identify traffic from the meterpreter session as I can see the shell.php file in the top 10 http requests. I know that a shell is created when meterpreter is in session so I can correlate the shell.php file request to the meterpreter session.

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

I would set an alarm that would alert me if any sort of file is uploaded onto the vulnerable machine from an unknown source. I would also ensure that an alert is created when any php file is uploaded to the server.

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

I would again make rules that allow only known and trusted ip's to upload any files to the web server.