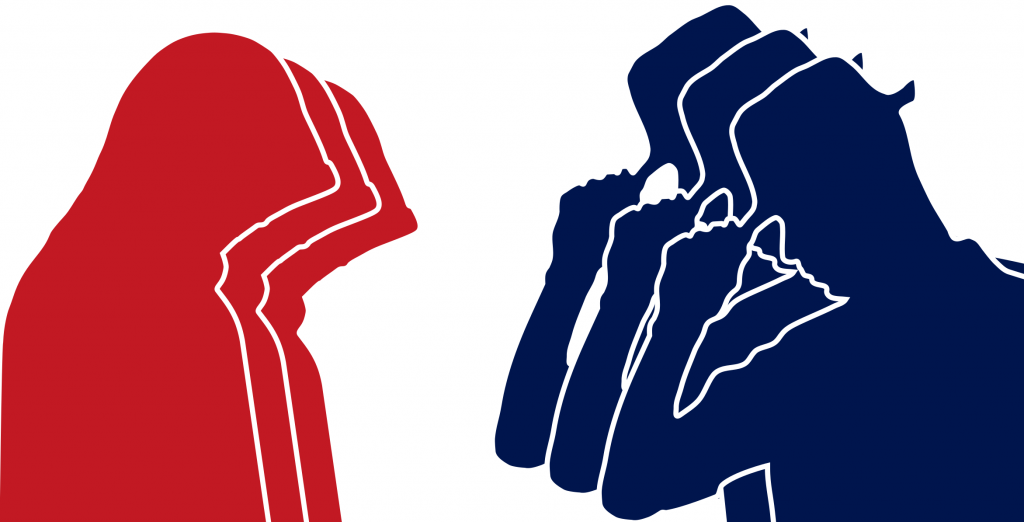
**Project 2 – Red Team vs Blue Team**

**By Symantha Meyers**

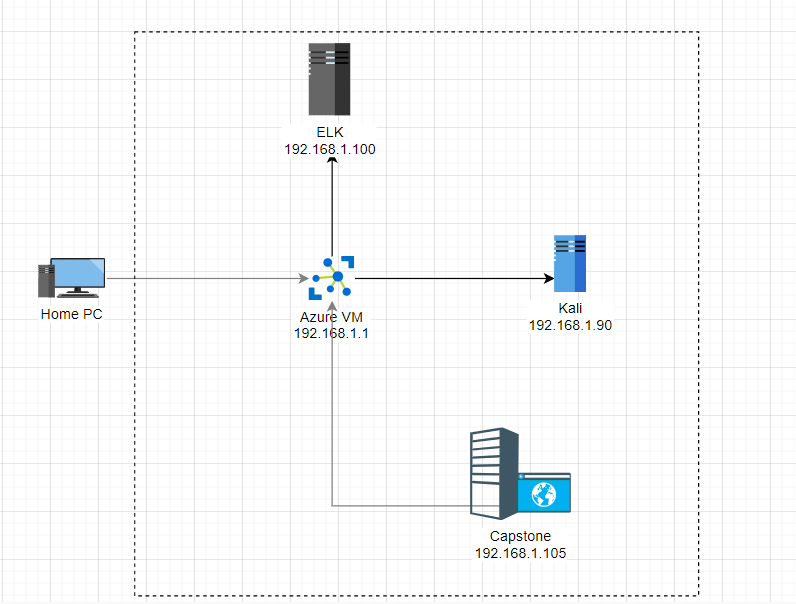
**December 19, 2020**



Capstone Engagement

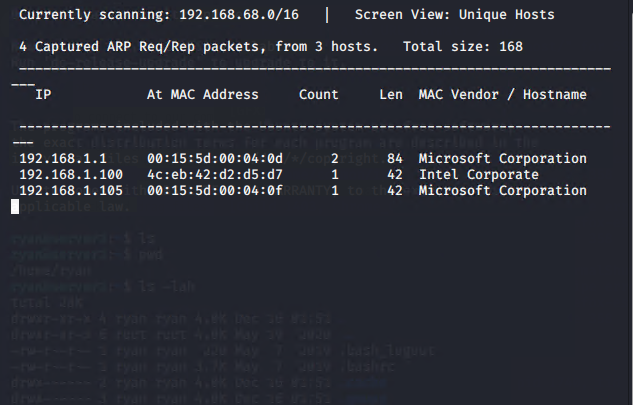
Assessment, Analysis,   
and Hardening of a Vulnerable System

**Network Topology**

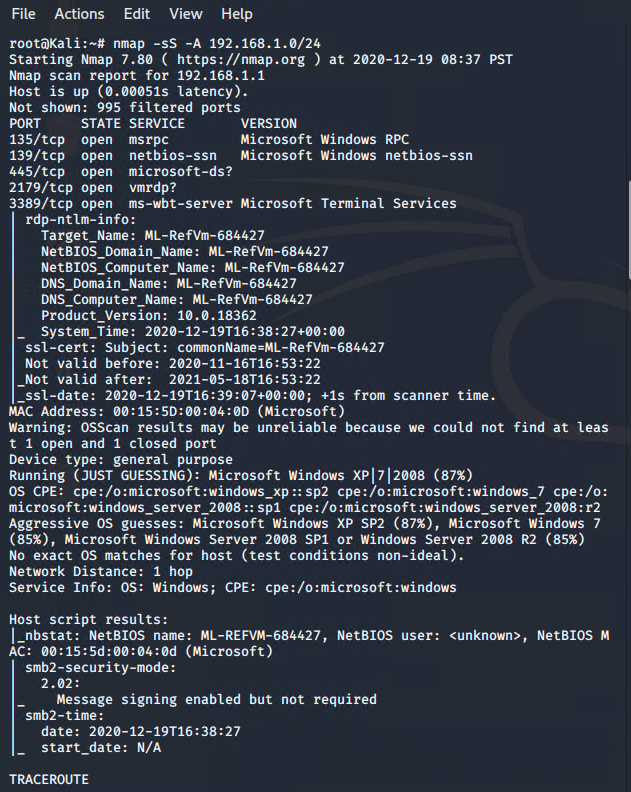
****

**Project 2 – Day 1 – Red Team**

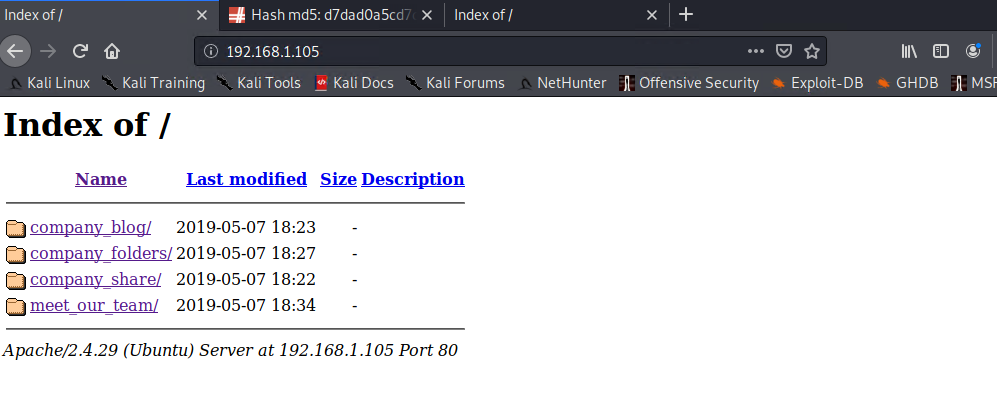
**Ran netdiscover to locate IP addresses on the network range 192.168.1.0/24.**



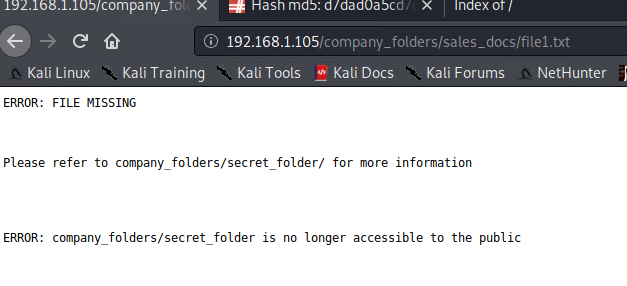
**Ran Nmap -sS -A 192.168.1.0/24 to discover the machines on the network, what ports are open, and what OS they are running.**



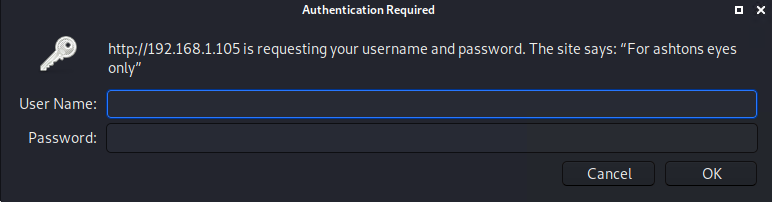
**Opened the Firefox browser and entered the IP addresses found until 192.168.1.105 worked.**



**I clicked around the folders and files until I saw:**

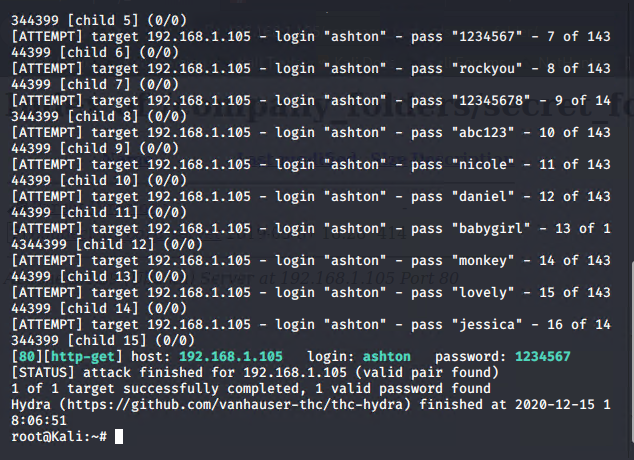


**which referenced a hidden folder. When I tried to access the hidden folder, it asked for a password and mentioned that it was for “ashton’s eyes only”.**

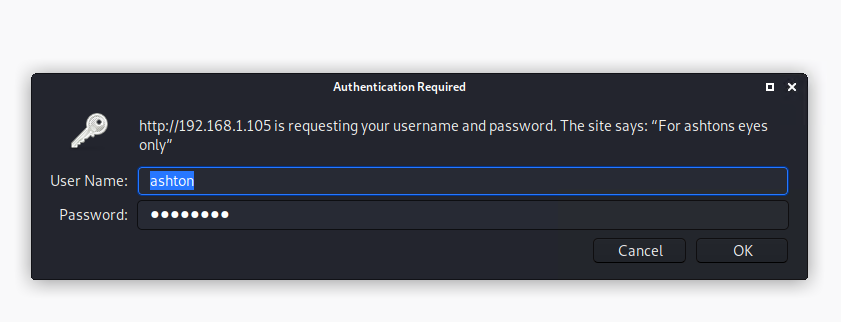
****

**In terminal, ran gunzip on rockyou.txt.gz in /usr/share/wordlists.**

**In terminal, ran hydra -l ashton -P /usr/share/wordlists/rockyou.txt -f -vV http-get://192.168.1.105/company\_folders/secret\_folder to get the password for ashton’s account**



**Logged into the secret share as ashton**



**Got the hash for ryan’s password from the text within the secret file**

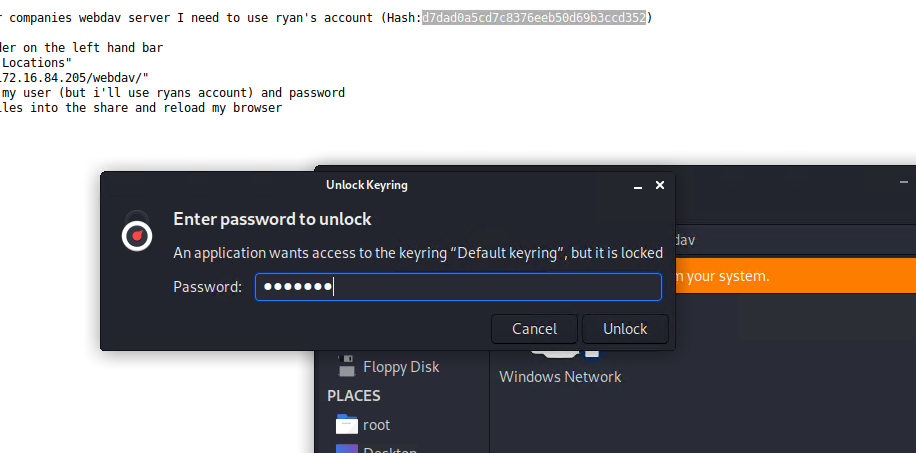


**Cracked the password hash with the CrackStation website tool**

****

**Per the info in the hidden shared file, using the file explorer, went to dav://192.168.1.105/webdav.**

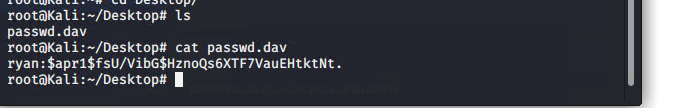
**At the prompt, I entered the password I had previously found.**



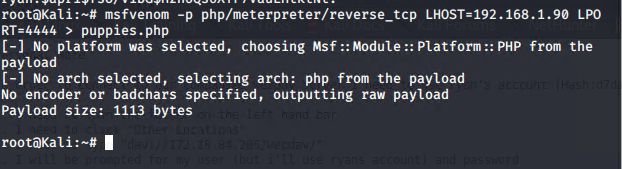


**Copied the file to my desktop**

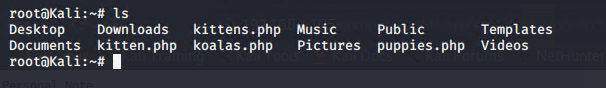
**Then cat’d the passwd.dav file to find the MD5 password hash for ryan.  
As we already know ryan’s password, I didn’t need to do anything with this data.**



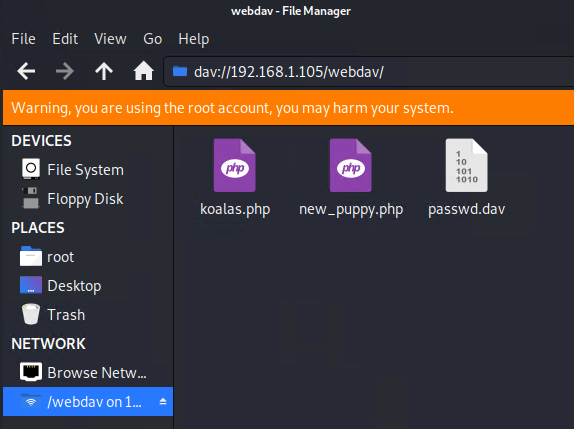
**Using msfvenom, created a malware payload.**

****

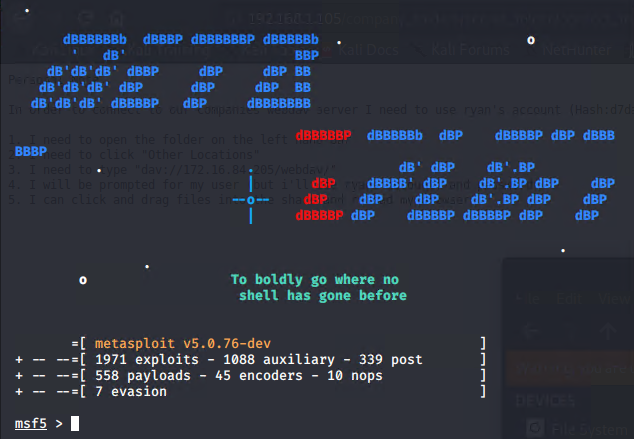
**Ran ls to double-check that the file was created.**



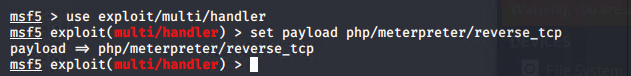
**Went back to file manager and copied the new payload to the webdav server.**



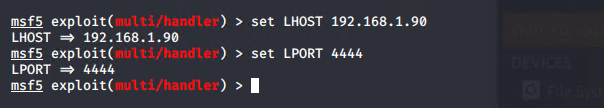
**Opened a meterpreter console.**



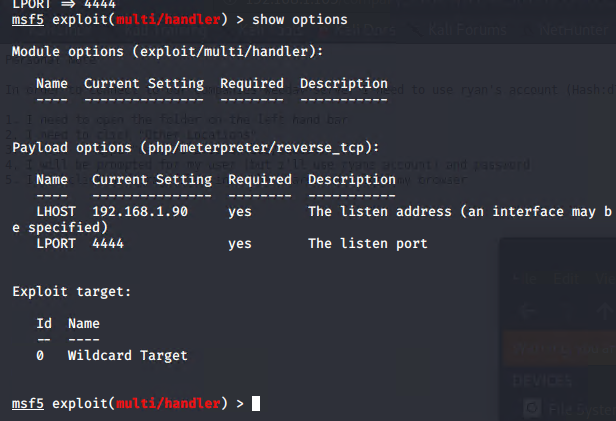
**Loaded the exploit and set the payload.**



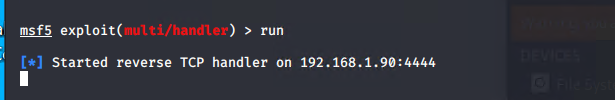
**Entered “show options” and entered the LHOST and LPORT info for the Kali server.**



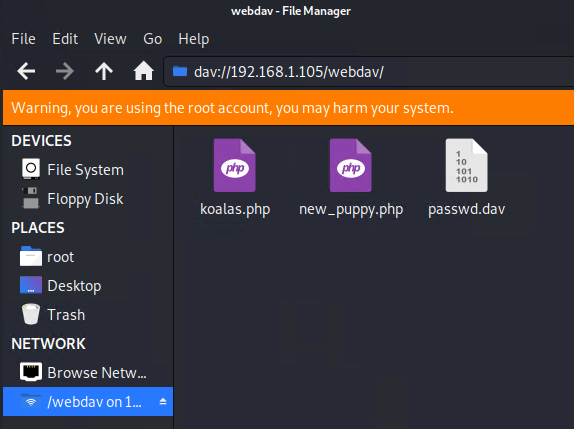
**Ran “show options” again to verify.**

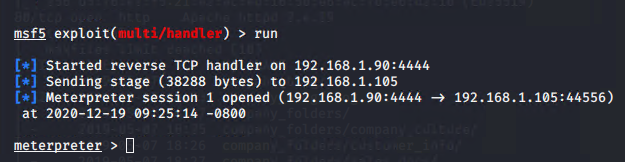


**Ran the exploit to start “listening” for the target computer.**

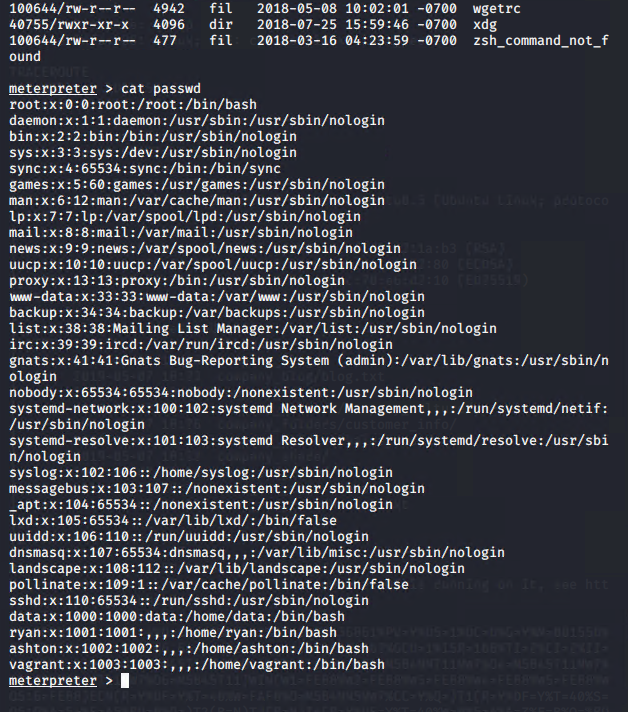


**Back on the web server, ran the exploit and a meterpreter prompt opened in Kali.**

****

****

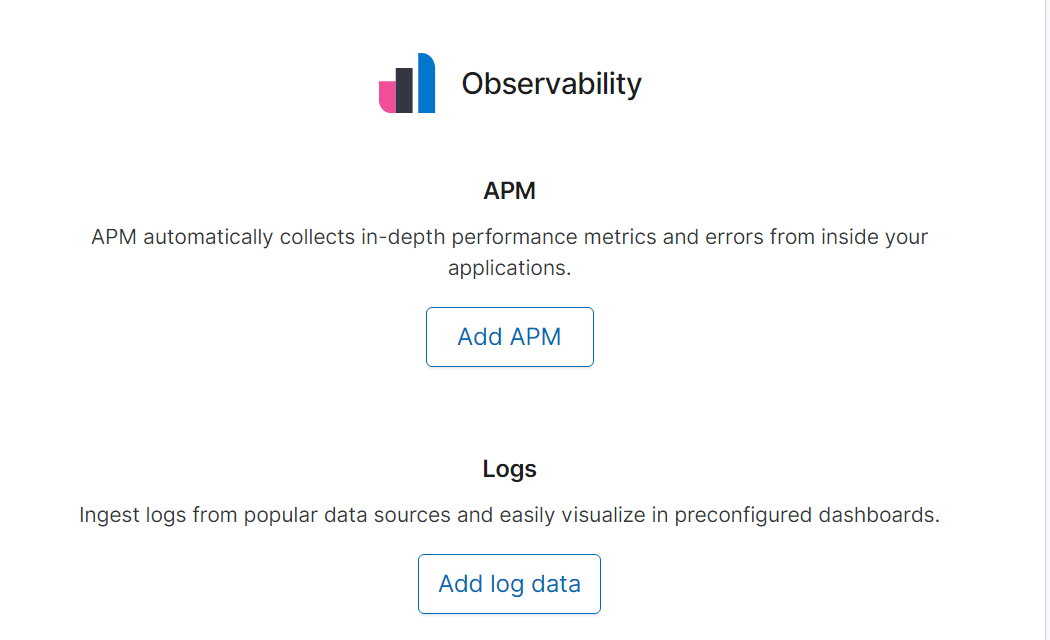
**Tested connection and rights by running a few commands.**

****

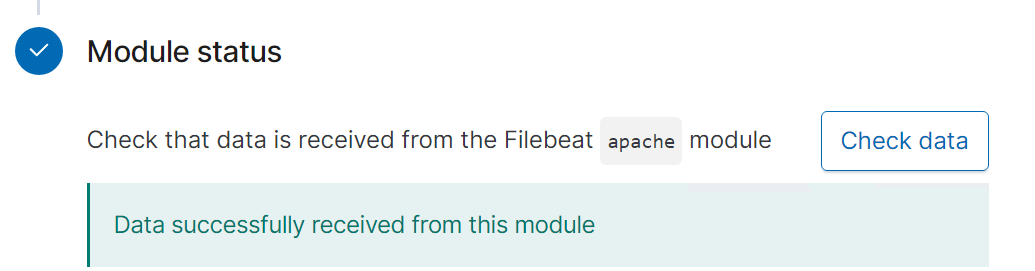
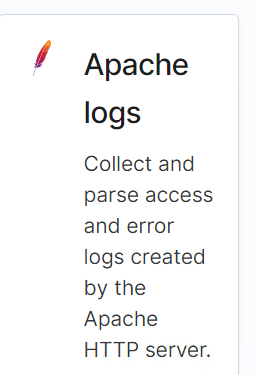
**Project 2 – Day 2 – Blue Team**

**Step 1 – Setup – Adding Log & Metric Data:**

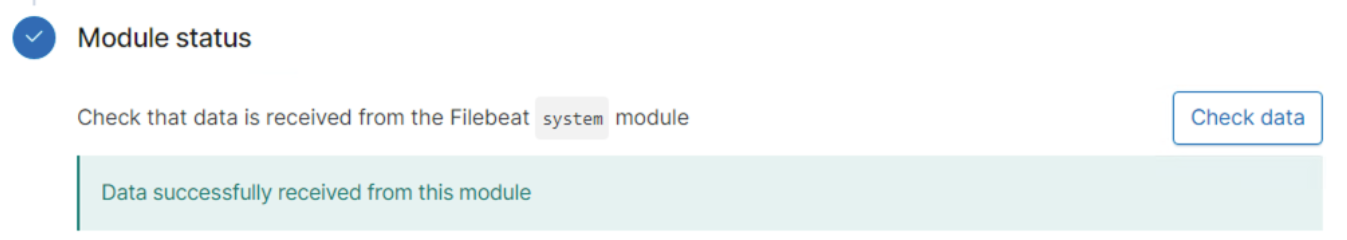
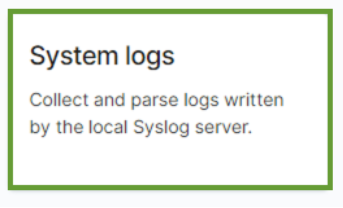
**Added log data to Kibana**

****

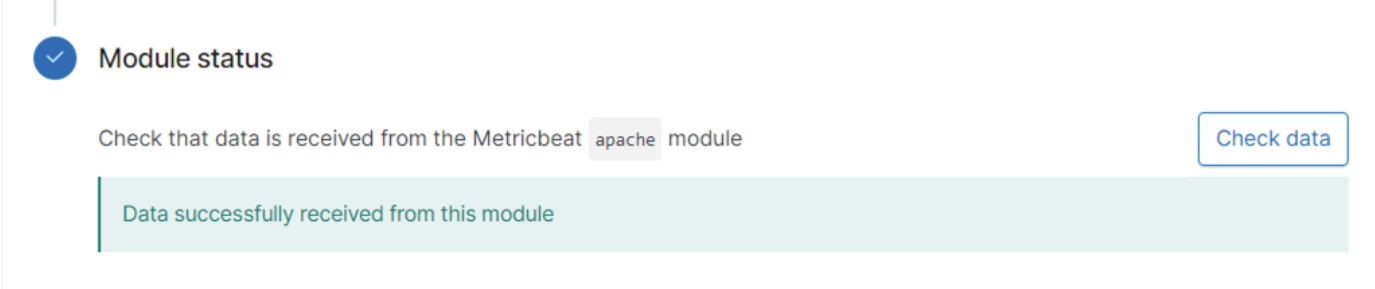
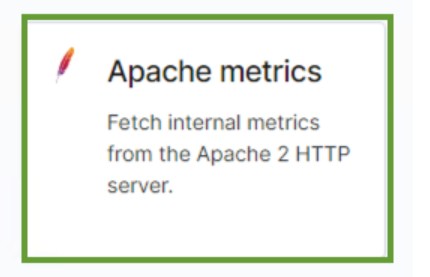
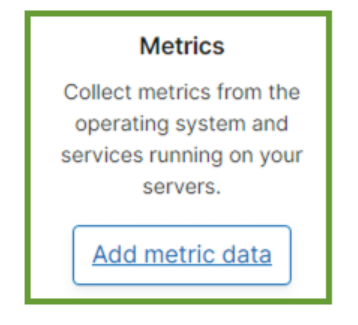
**Added Apache Logs and verified the log’s addition**

****

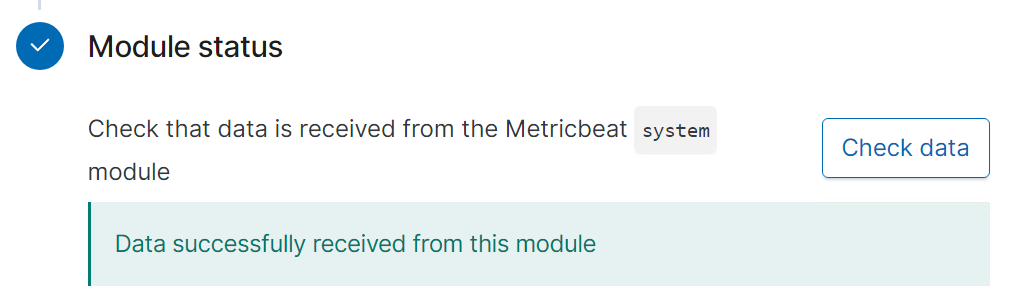
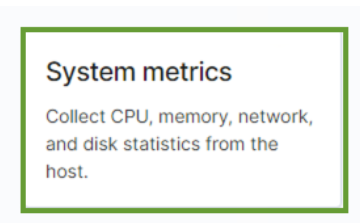
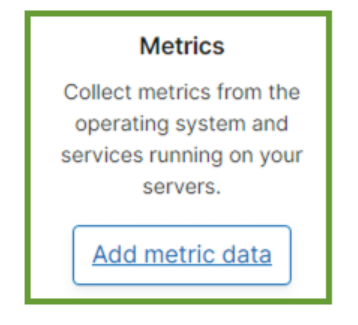
**Uploaded log data for system logs and verified successful upload**

****

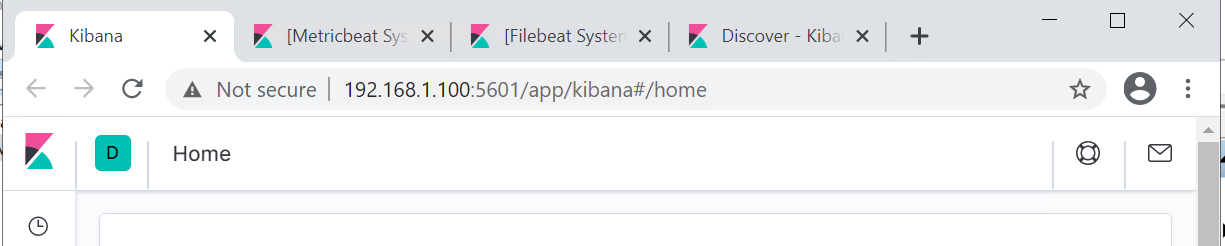
**Uploaded Apache Metrics and verified successful upload**

****

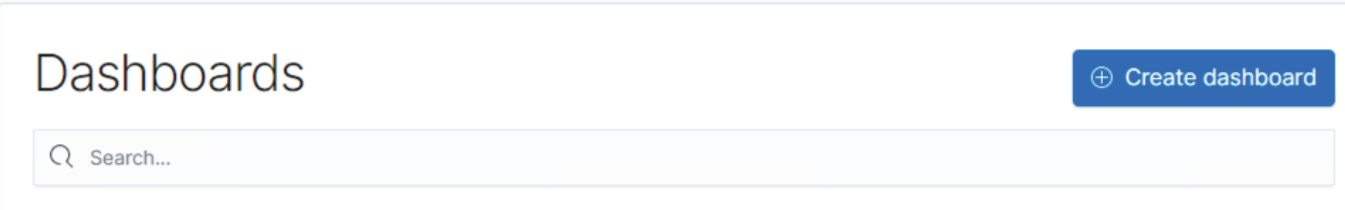
**Added system metrics and verified upload success**

****

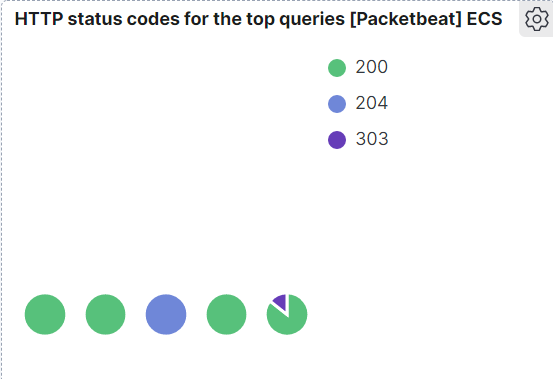
**Reopened Chrome now with four Kibana tabs, for each log or metric**

****

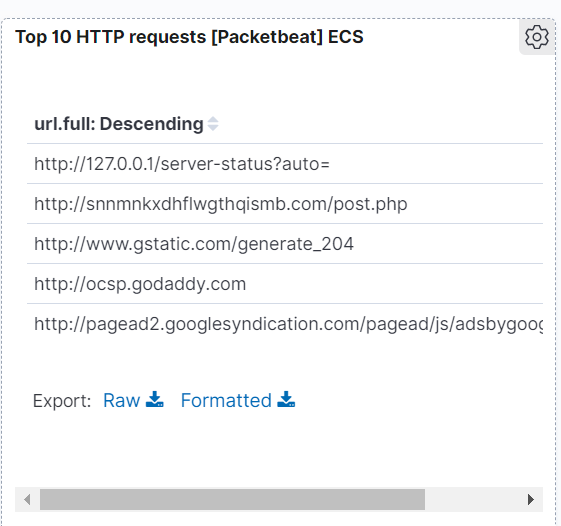
**Step 2 – Set up - Creating the dashboard:**

****

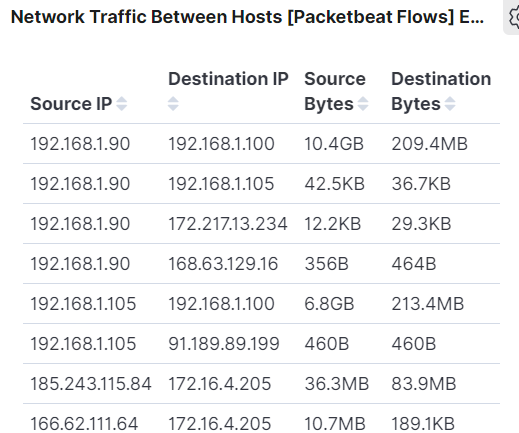
**Added “HTTP status codes for the top queries” panel**

****

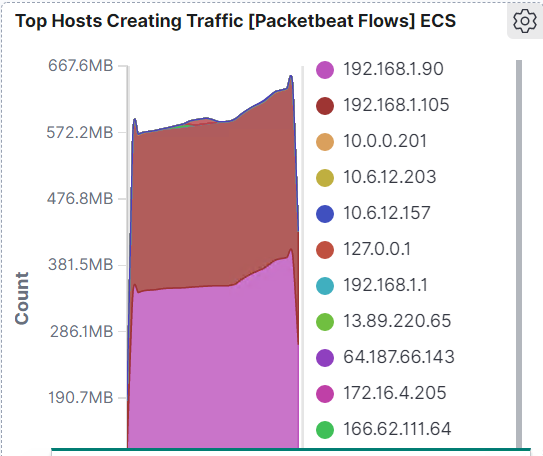
**Added “Top HTTP Requests” panel**

****

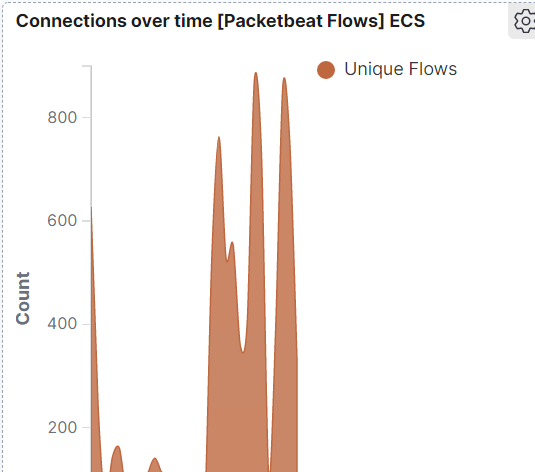
**Added “Network traffic between hosts” panel**

****

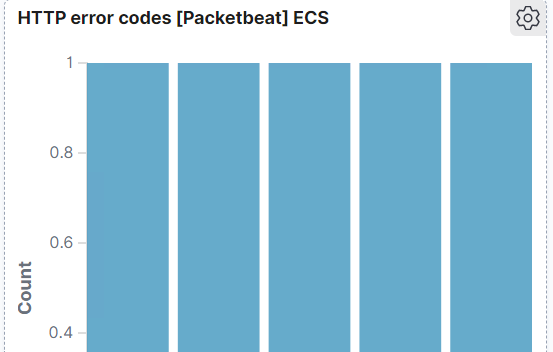
**Added “Top Hosts Creating Traffic” panel**

****

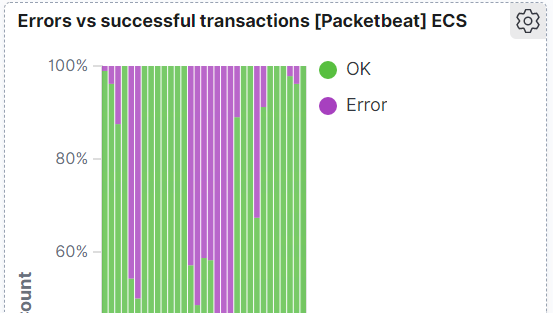
**Added “Connections over time” panel**

****

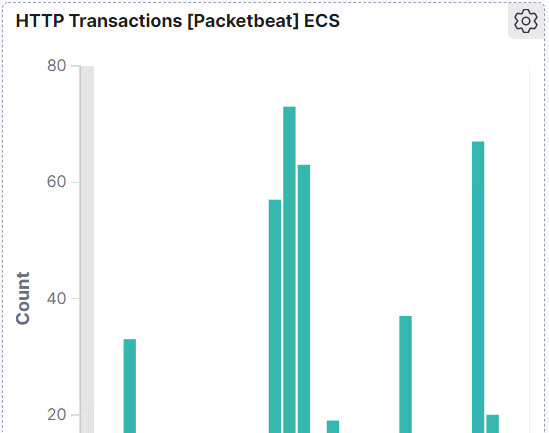
**Added “HTTP error codes” panel**

****

**Added “Errors vs Successful transactions” panel**

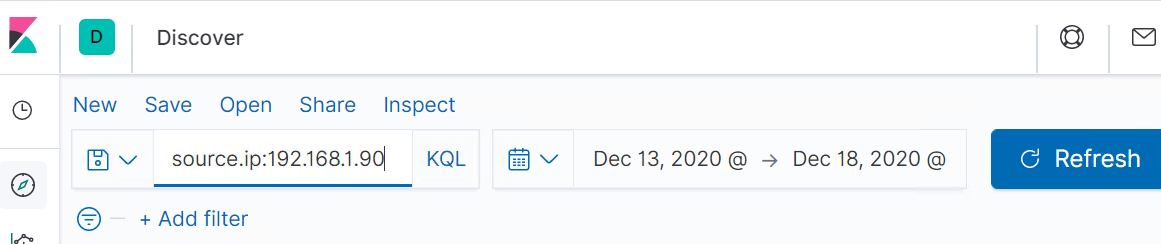
****

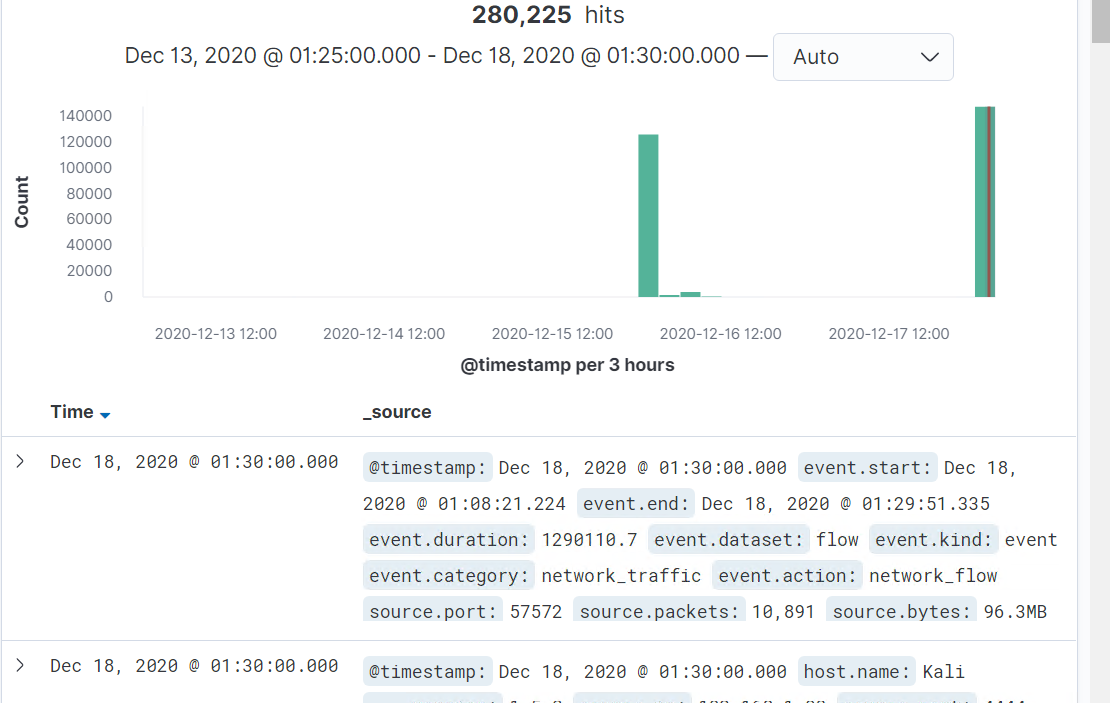
**Added “HTTP transactions” panel**

****

**Step 3 – Interpretation, Discovery, and Mitigation**

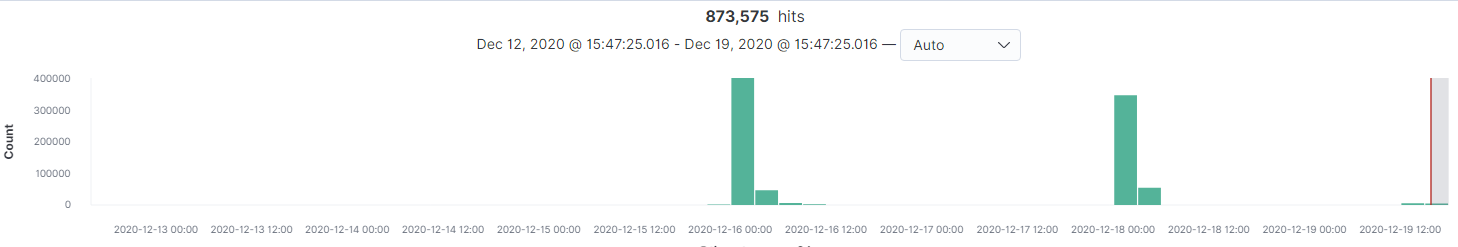
**Looking at the data in Packetbeat, I was able to discern the date and times of the attacks.**

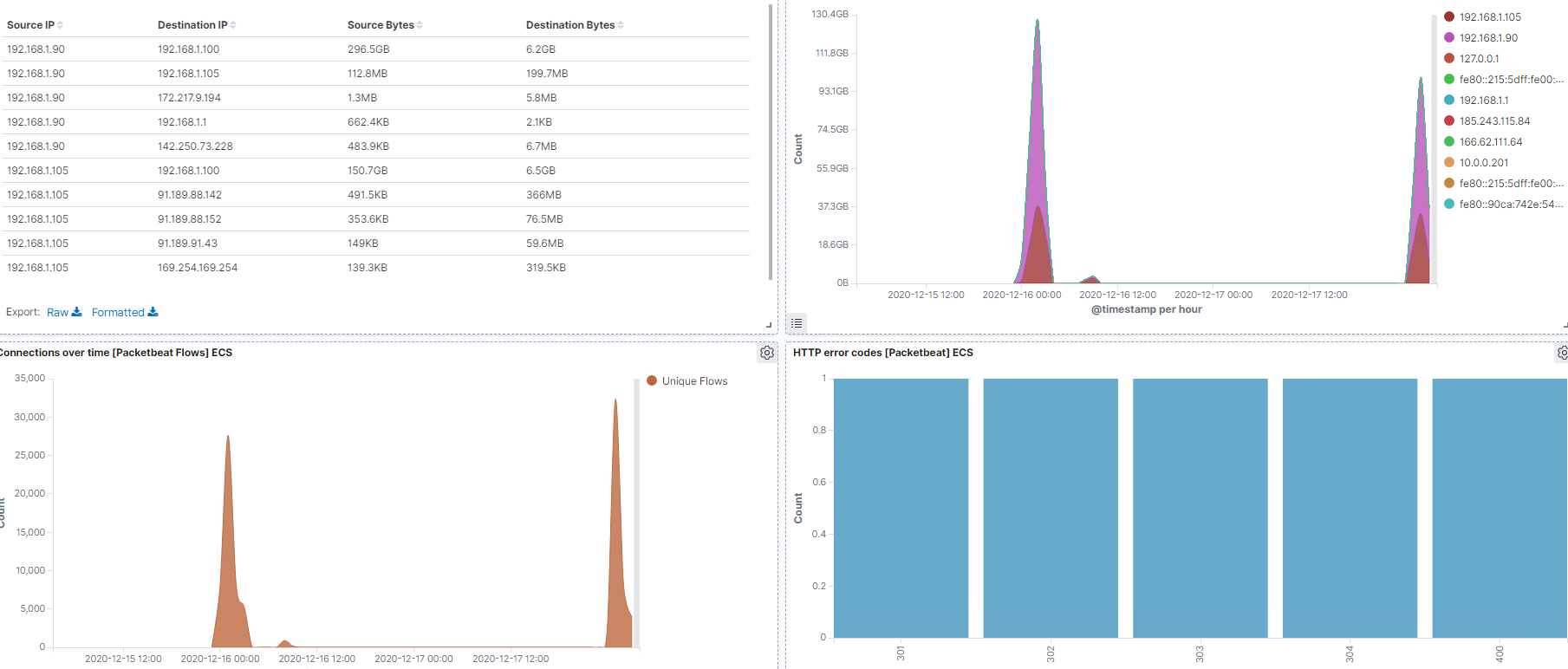
****

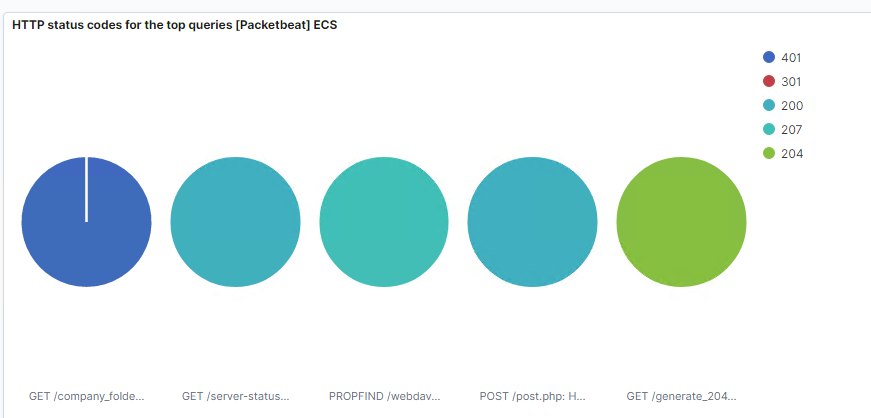
****

**There were two attacks, one on the 15th-16th of December and one on the 17th-18th of December.**

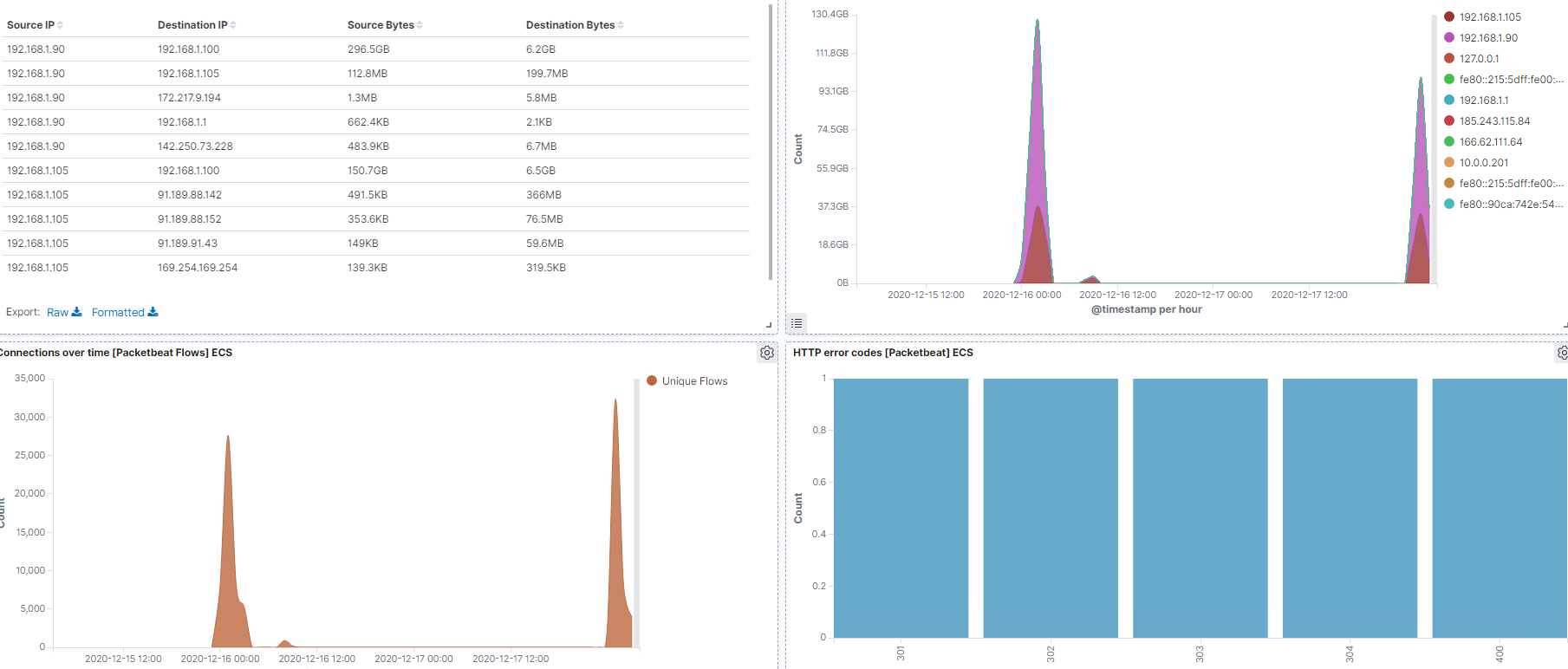
****

****

****

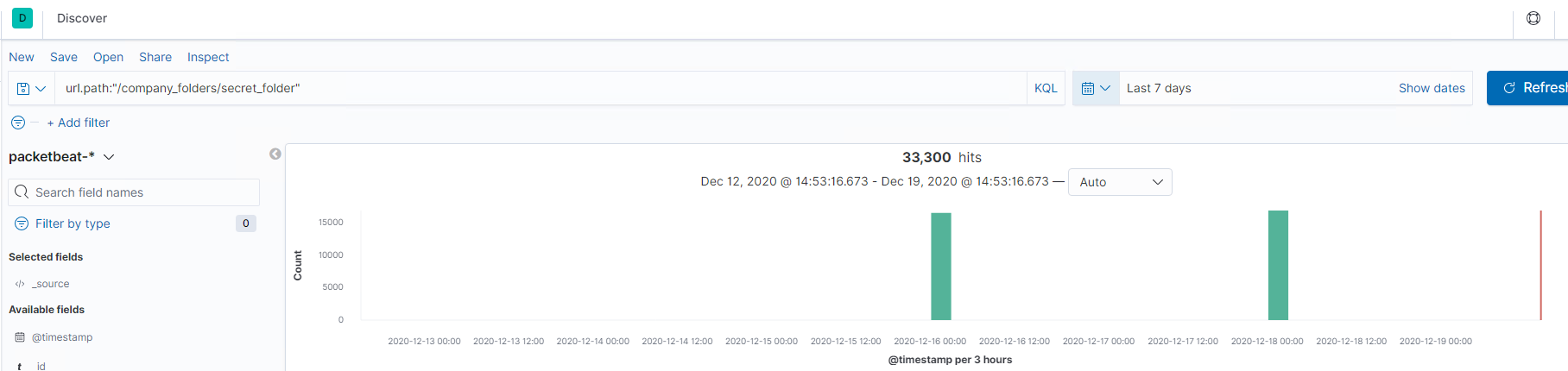
**The victim computer sent back the following responses:   
**

**The data that concerns the blue team is the amount of traffic over a short period of time, per the below screenshot where you can see the spikes in traffic.**

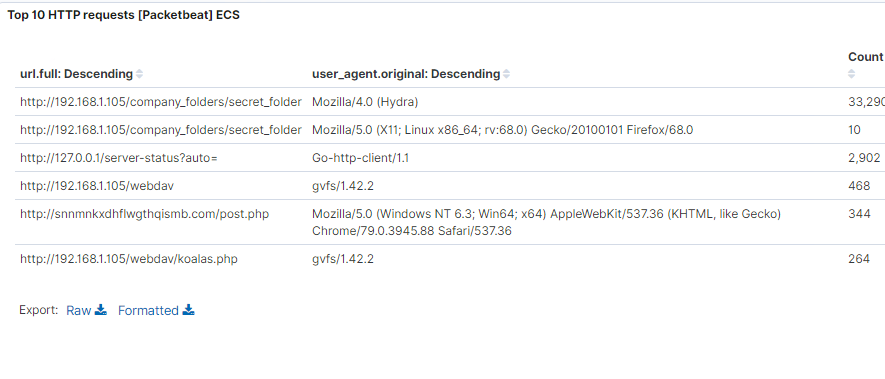
****

**ACCESSING THE SECRET SHARE**

**Looked for the data showing when I was looking for and finding the secret share.**

****

**These files were the top files requested:**

****

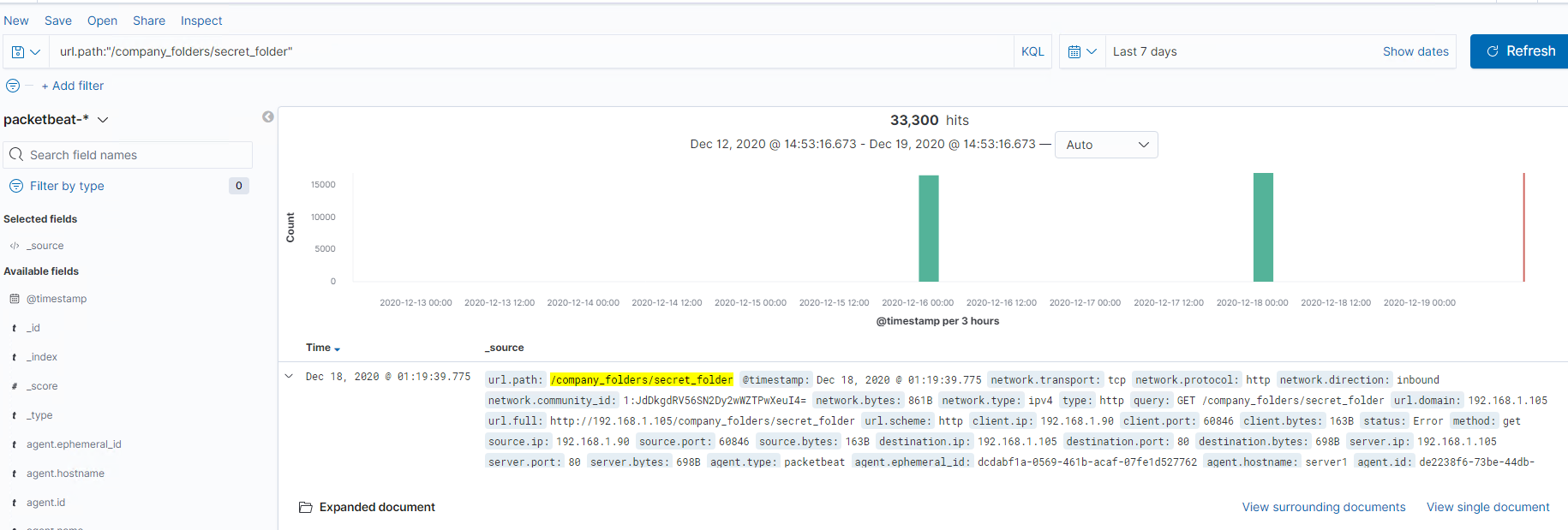
**Possible mitigation strategies:**

**We could set up an alert for the future if any of these files are requested or if they are requested by any user other than authorized users + the specific machines that those users use and/or their IP addresses.**

**One way to harden the server would be to not have the password hash in the instructional text file… or better yet not have an instructional txt file at all.**

**BRUTE FORCING THE PASSWORD FOR THE SECRET SHARE**

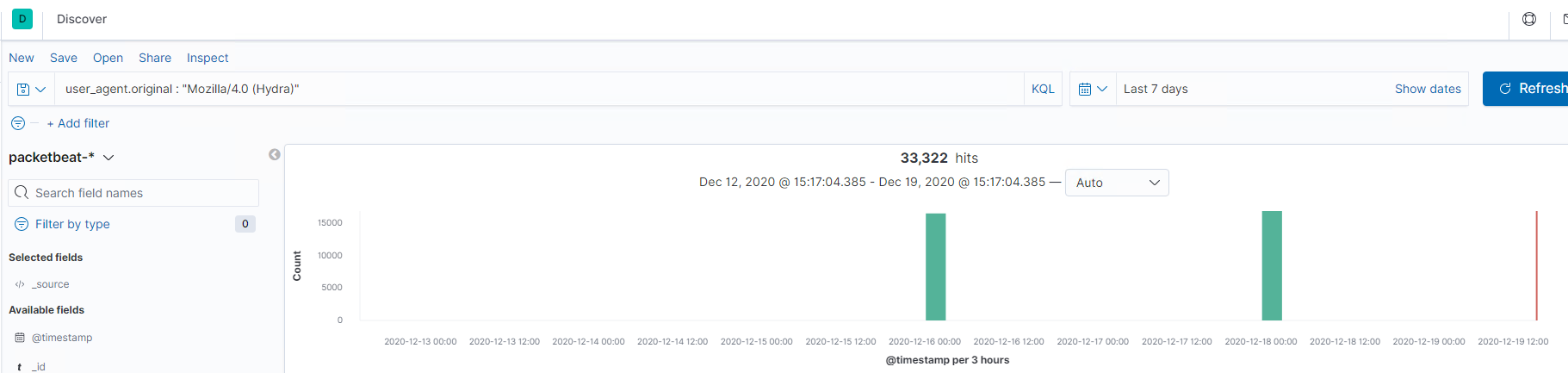
**In the following search, you can see the data requested by Hydra when brute forcing the password for the secret share.**

****

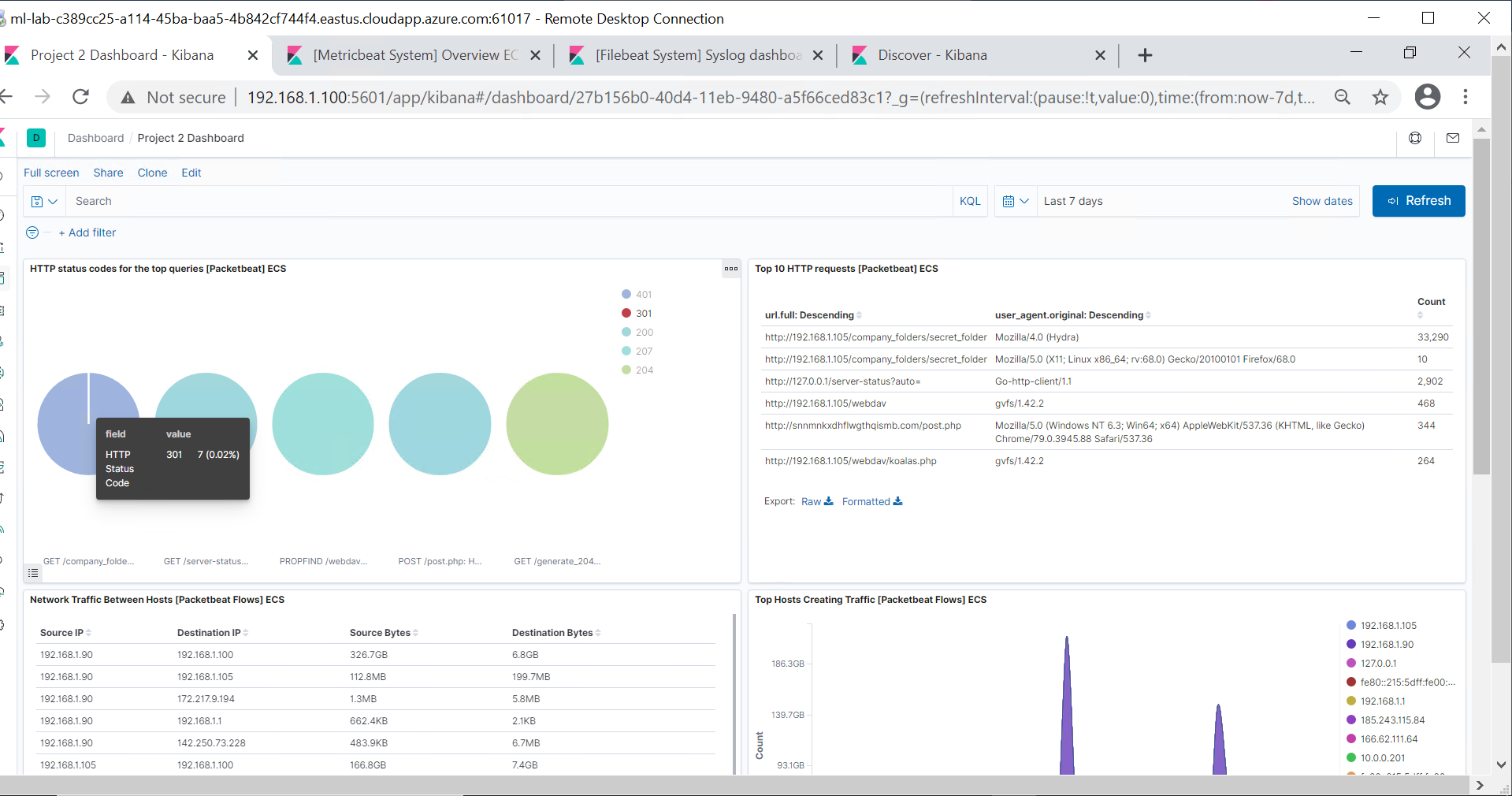
**The data specifically requested by Hydra can be seen in the following screenshot.**

****

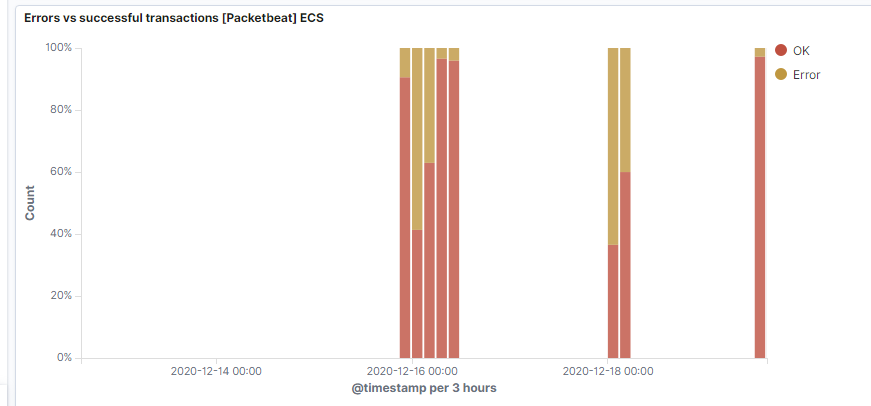
**Per the below screenshot, there were 33,322 hits with the user\_agent.original: Mozilla/4.0 (Hydra).**

****

**But you can see that only 7 hits were successful.**

****

**You can also see the information in the following screenshot of errors vs successful transactions:**

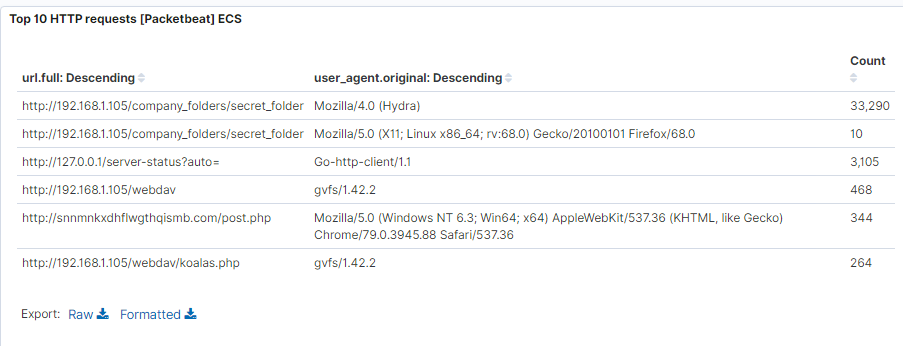
****

**Possible Mitigation Strategies:**

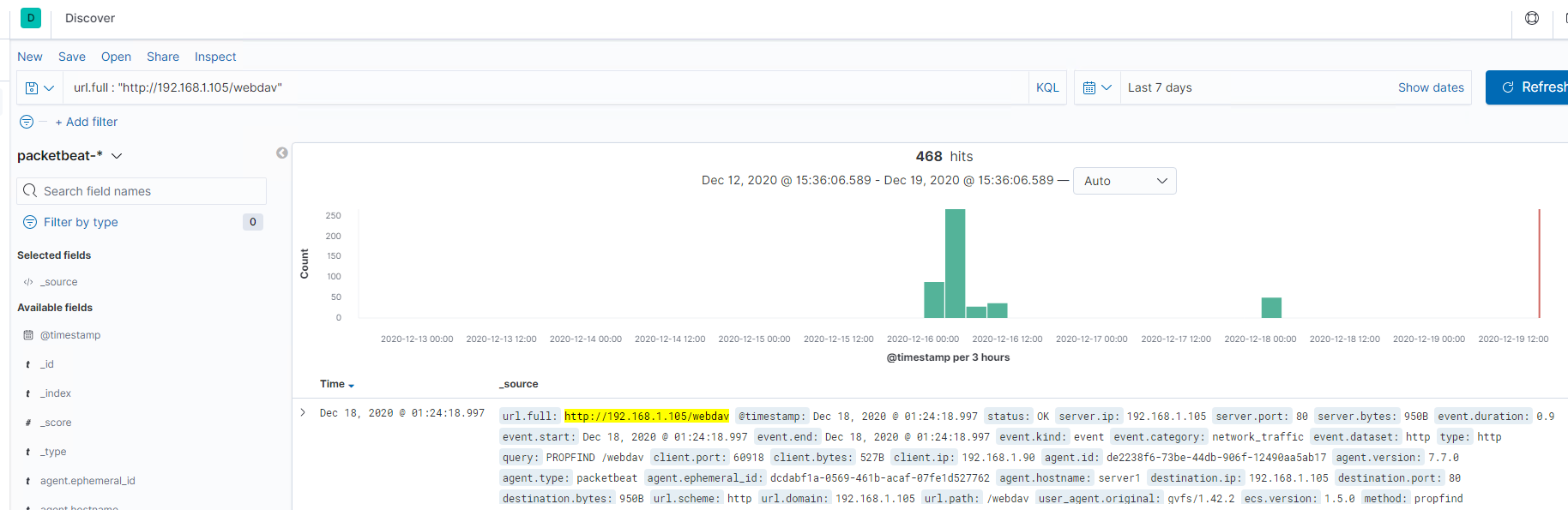
**The prospective alerts that could be set might be to look for any instances of the user\_agent.original of Mozilla/4.0 (Hydra) and to limit the number of Error 401 responses to 20 in one hour and adjust as needed.**

**WEBDAV SERVER**

**The number of requests made to the WebDav directory was 468**

****

**In the above screenshot, you can see that the koalas.php file was requested 264 times and per the below screenshot, you can see that the webdav folder was requested 468 times.**

****

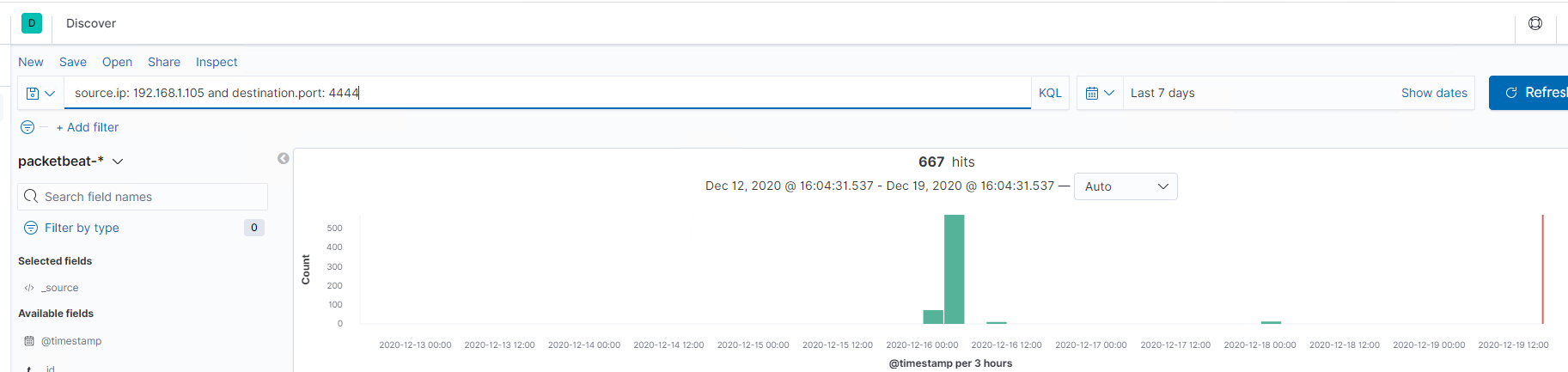
**Possible mitigation strategies:**

**You could set up an alert that will go off if any machine connects to the webdav folder that isn’t supposed to have access.**

**One way to harden the server would be to set up a firewall rule that restricts access by any machine that is not a machine used by the actual correct user.**

**METERPRETER ATTACK**

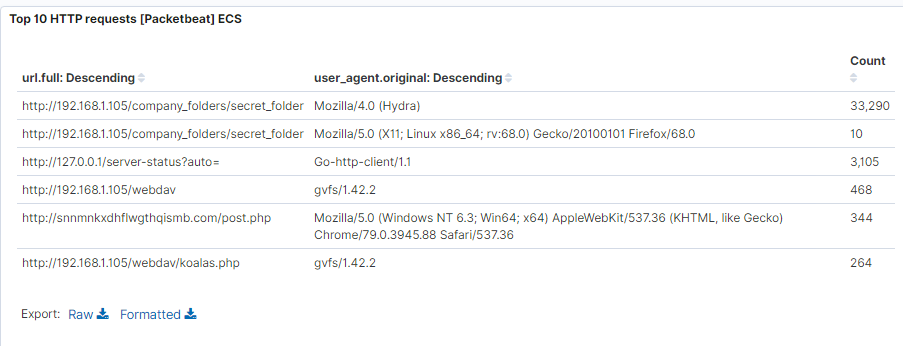
**Ran a search for the IP of the attacker and source port 4444 to view the meterpreter attacks**

****

**There were 667 hits**

**Possible mitigation strategies:**

**As meterpreter’s default port is 4444, we can set an alert for any traffic from port 4444 and/or for any .php files being uploaded.**

****