Jose Salazar
roo821
IS 3523
Lab One: Detection Engineering
October 5, 2024
Professor Burres

This has been one hell of a journey. I am not a complete novice, and I would like to think that I am incredibly skilled at being able to follow directions. I'm not a doctor, but I will admit that there are definitely some opportunity areas with this program. It was a consolation that there may have been an issue with an update that caused all of my issues. My networking skills were certainly called into play and I will admit that this was the most time that I have spent on a lab in my academic career. With that being said, I will say that I sure learned quite a bit about setting these programs up and fine-tuning my networking skills. In my underpaid opinion, I think that this was a blessing in disguise, and I really learned a lot more about PowerShell, cmd, Desktop Docker, VMWare, Elastic, Fleet, and patience. I did have some hiccups outside of getting my setup to show alerts, but as I will document my learning journey, it was a great learning process. I certainly look forward to using all this knowledge to set up a home lab and utilize it to further my career.

Section One. After going through the installation of Desktop Docker and everything leading up to the lab, I realized that I did not know what was going on. I did a lot of reading and did my best to understand what was being asked of me and why. I tried to learn from my mistakes during the installation of Desktop Docker and the containers to make sure that they weren't repeated during the lab. Some mistakes can't be avoided during the learning process, and I made my first mistake during the creation of the .env file. I knew from my early Windows days that you could rename a file and change the extension. What I did not know was that you could create a file that was just an extension. My first roadblock was when I opted to use Windows instead of Linux to change the extension. I had a file called env.env which caused many an issue in getting the Fleet Server to work. After

recalling an issue with the token having the << >> signs being left in, during the setup of Elastic, I figured out that the env.env file, as I so aptly named it, needed to be titled just .env. After spending 30 mintues troubleshooting, figuring out the issue, and correcting it, I was able to move to the next step. Although, this apprehension that I learned would soon come back to haunt me. I will tell you more about this later.

After learning my lesson with setting up Elastic, I was able to recall needing to leave out the << >> and able to enter the token for Fleet without a hitch. I was a bit flustered, and a bit confused at first about getting the Fleet Server connected, but I think I found a solution. I went into ipconfig /all and got the IP address that I needed, but when I went to look at my hosts file at C:\Windows\System32\drivers\etc\hosts, I got a totally conflicting IP address that reflected the UTSA IP where I had initially set up the server. I did forget to mention that I had started the lab at school but continued at home. This may have been part of my downfall for the latter part of this lab. I didn't think much of it, as the IP address that I did put into the Fleet Server gave me a green light and let me go ahead, so I didn't put the IP address from the hosts file in when I ran "docker compose up -d" from the fleet folder. "docker compose up" aggregates the output of each container. With the addition of '-d' or detach, it starts the containers in the background and leaves them running. I was able to configure it ok, as requested in step four of section one. So, I didn't think that there was a problem at this point.

I moved to the installation of VMWare and Windows 10. At the suggestion of Professor Burres, I installed VMWare without incident and moved forward to the download

and installation of Windows 10. I was able to get Windows 10 installed and running without an issue and flew through step 5.

Step 6 and 7 was where I was met with a bit more of a challenge. Well, the tediousness of a lot of typing and trying to learn all the new commands that I encounter. After logging back into Kibana, I navigated over to add a Windows policy. After clicking on the windows tab and pulling up the five lines of commands, I went back to my newly setup Windows 10 VM and opened PowerShell. “cd $env:USERPROFILE\Downloads” is “a command that navigates to the Downloads folder for the user currently logged into a Windows computer” (Microsoft,2024). Now that I am here, I need to tell my Windows VM to communicate with my main computer. I am to type the five lines of commands into PowerShell to accomplish this. First command, “$progresspreference = 'silentlycontinue',” tells the computer to execute “the command, but doesn't display the progress bar” (Microsoft, 2024). Next, comes the big one, “invoke-webrequest -Uri https://artifacts.elastic.co/downloads/ beats/elastic -agent/elastic-agent-8.4.3-windows-x86_64.zip -OutFile elastic-agent-8.4.3-windows-x86_64.zip” What this tells PowerShell to do next is to send a request to this specific website, and to download that specific file “elastic-agent-8.4.3-windows-x86_64.zip.” It then lists what the output file will be named, which in this case is the same thing. The next line is telling PowerShell what it needs to do with that file as soon as it is done downloading it. That line is: “Expand-Archive .\elastic-agent-8.4.3-windows-x86_64.zip -DestinationPath .” Breaking down that command line, it is straight forward. The first part, expands the .zip file(archive), the middle part designates which .zip file needs to be expanded, and the last part designates where the files inside

should be expanded to; This is where it could cause problems. On that last part, if you didn't copy the command correctly, it could get you stuck. The command should read "-DestinationPath ." but some folks left the "." out. With that command, it is being directed to a destination path, but it needs to be told what that path is. The "." specifies that the destination is the current directory. So, I can see why leaving out the period would cause the computer to not unzip the file, if it didn't have anywhere to put it. Finally, comes the final command, but it needs to be modifed from what is in Kibana. "- -insecure" needs to be added to the line. This indicates that the connection will not verify the SSL certificate. The command reads as follows:

"cd elastic-agent-8.4.3-windows-x86_64 .\elastic-agent.exe install --url=https://192.168.1.165:8220 --insecure --enrollment-token=RnZVckZwSUJYaS1qOHVqZWF2OGU6SmZOTzRrR3dRLUtNQzJIQjNSdTAydw=="

To break down this long set of commands, the first command navigates to the directory where the Elastic Agent is stored, setting the current working directory to that location. The second part runs the Elastic Agent executable from the current directory (.\), install instructs the Elastic Agent to install itself, --url=https://192.168.1.165:8220: specifies the URL of the Fleet Server or Elastic Agent endpoint that the agent will connect to, In this case, it is hosted on the IP 192.168.1.165 at port 8220, and we already discussed "--insecure." Finally, we have the big one, the enrollment token, allowing the Elastic Agent to authenticate and register with the Fleet Server for central management and monitoring. Success! It connected and it is showing that it is connected in Elastic, or so I thought.

It was unable to make a full connection. It just said waiting to get data and it never came. I still decided to go forward as if I had been successful. That took me through steps 8 and 9. At step 10, I created the new threshold rule and gave it the threshold guidelines as set in the lab. After creating the new rule, I logged out of the VM and tried to get an alert to come up, but it was still not connecting. I am going to try setting all of this up on my main computer that I am building for a home lab. Unfortunately, it won't be ready in time to complete this lab, but I will revisit this lab at a later date, for my own personal use.

Section Two. Moving forward, for the second section of the lab, I researched many different possible rules to use and went down a rabbit hole. I learned many new things that are outside the scope of this paper, but since I can't really test this out, I plan on being a bit more ambitious than a forgotten password.

First, I created a rule in Elastic, using "Custom Query" as the rule type. I intend to detect PowerShell encoded commands in Windows event logs. I will use "process.name: "powershell.exe" AND process.args: "*-EncodedCommand*." This query looks for the execution of PowerShell (process.name: "powershell.exe) and checks if the -EncodedCommand argument was used (process.args: "*-EncodedCommand*"). I set the name to PowerShell EncodedCommand Attack. I set the severity to "high" as I can't think of a reason why it would be lower and gave it a risk score of 75. I did note in the description that this detects the use of PowerShell encoded commands, which is often used by attackers to hide malicious commands. I set this to run every 5 minutes.

Since I couldn't get Elastic to work with my Windows VM, I will just have to describe what I would do to simulate an attack. The commands used for this attack are:

```
$command = 'Write-Host "This is a test attack"'
$bytes = [System.Text.Encoding]::Unicode.GetBytes($command)
$encodedCommand = [Convert]::ToBase64String($bytes)
powershell.exe -EncodedCommand $encodedCommand
```

This is essentially a small program, with the variables being set. $command is the main variable that holds the command that will be encoded. $bytes converts the $command string into a byte array using Unicode. $encodedCommand encodes the byte array into a Base64 string. Finally, PowerShell is run with the -EncodedCommand parameter, which allows you to execute the Base64-encoded string as a PowerShell command.

Of course, for me, this is all theory, but I am excited to try this out once we get access to the VPN. I did include some screenshots of the second half of this lab to illustrate what I did for creating a rule and the problem I was having with Elastic. I wish I could pinpoint to one problem, but as time went on, it was like dominoes and one thing stopped working after another. I truly look forward to being able to use what I learned during this lab. The complex problems that came up through all aspects of this lab was a great learning experience. I can say that I now know more about networking, containers, and VMs after this.

Appendices

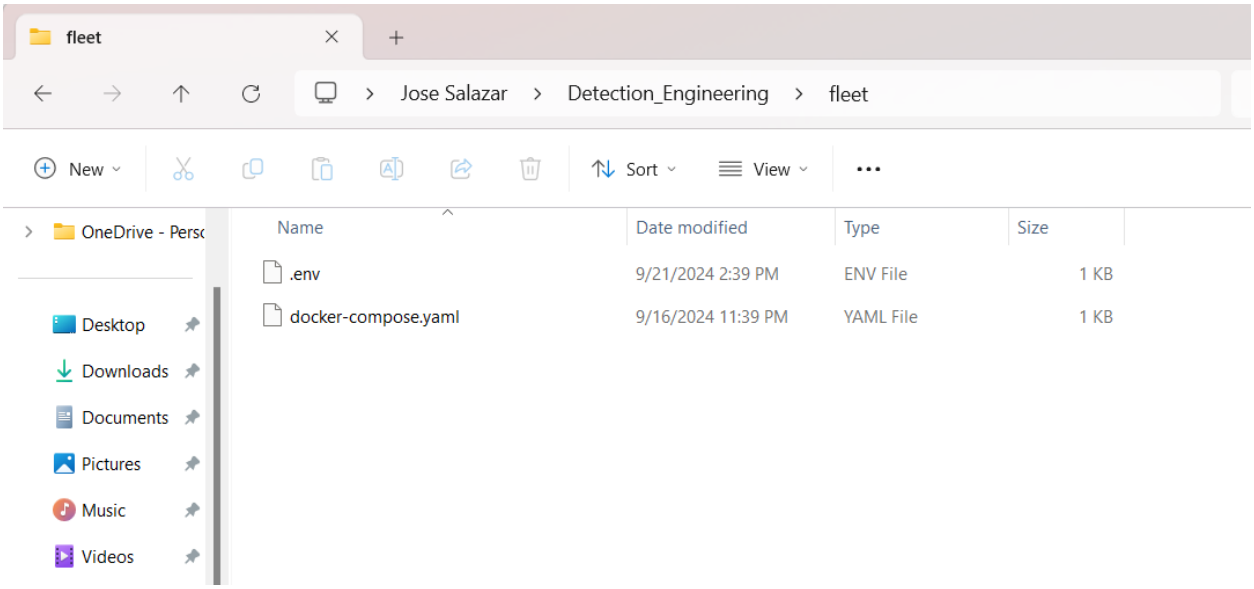

Figure 1. Screenshot of Step 1

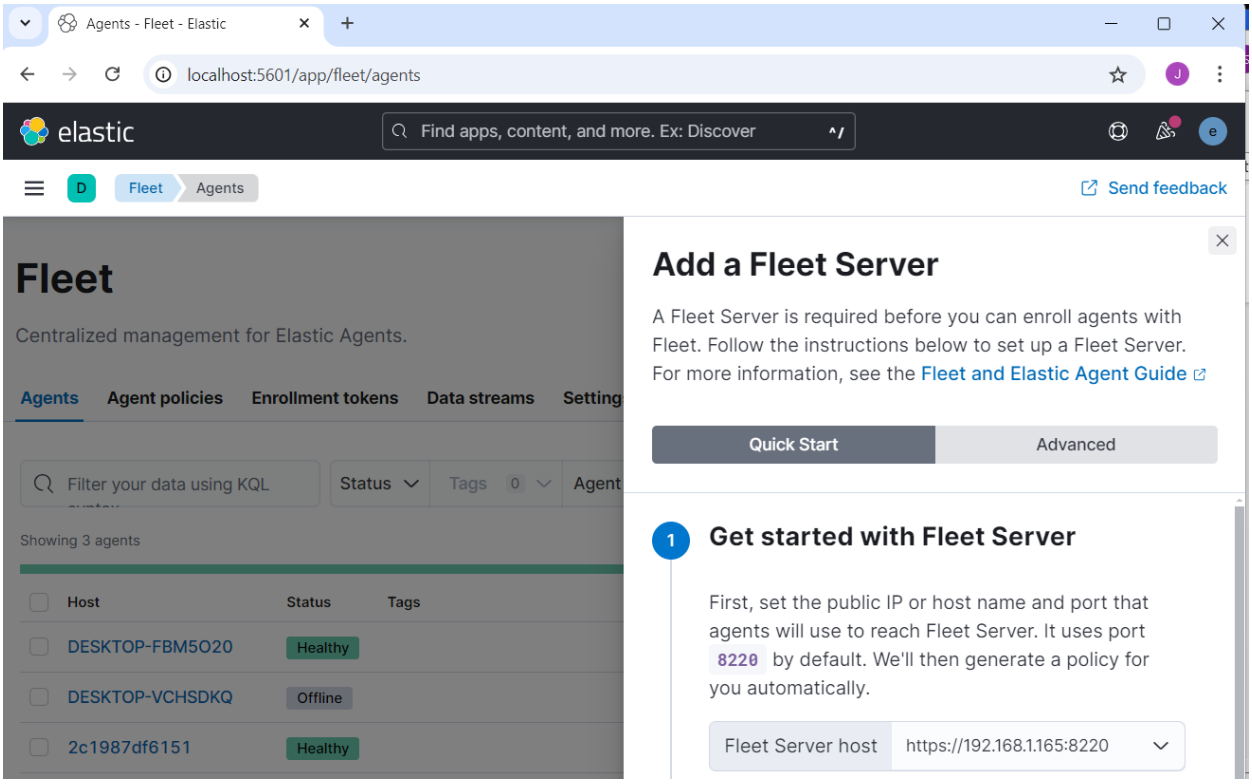

Figure 2. Screenshot of Step 2 – Fleet Server setup

Figure 3. Fleet Token in .env file

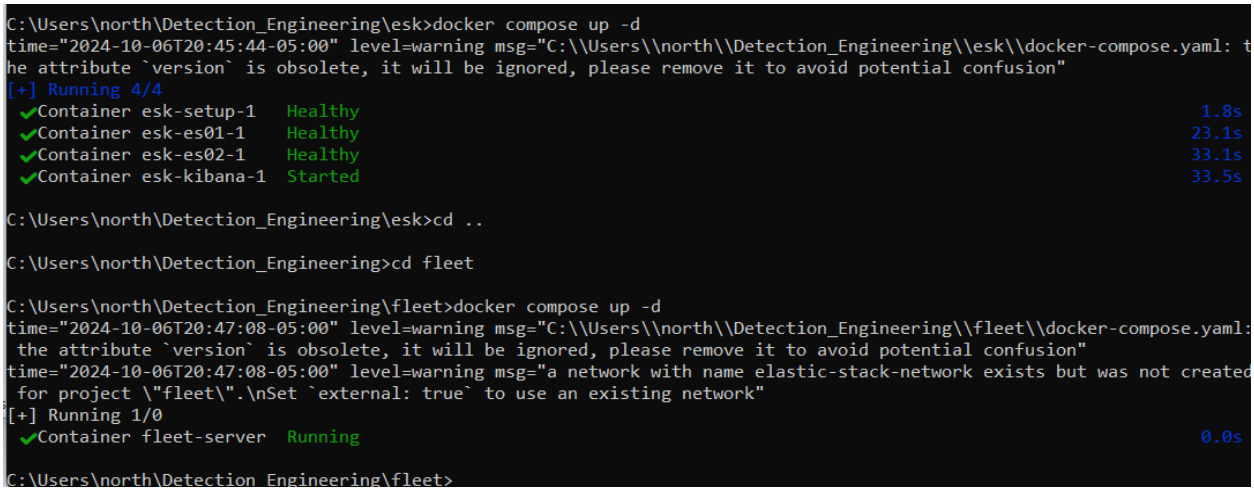

Figure 4. Screenshot of Step 3 – Docker being run

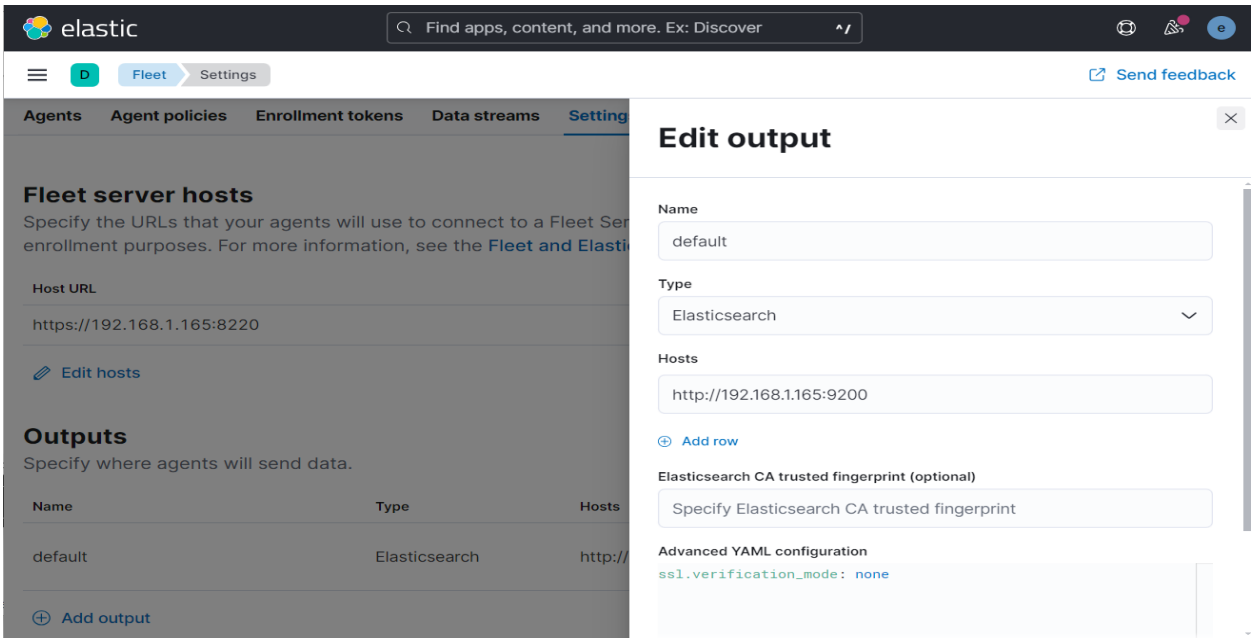

Figure 5. Screenshot of Step 4 – Settings in Fleet server

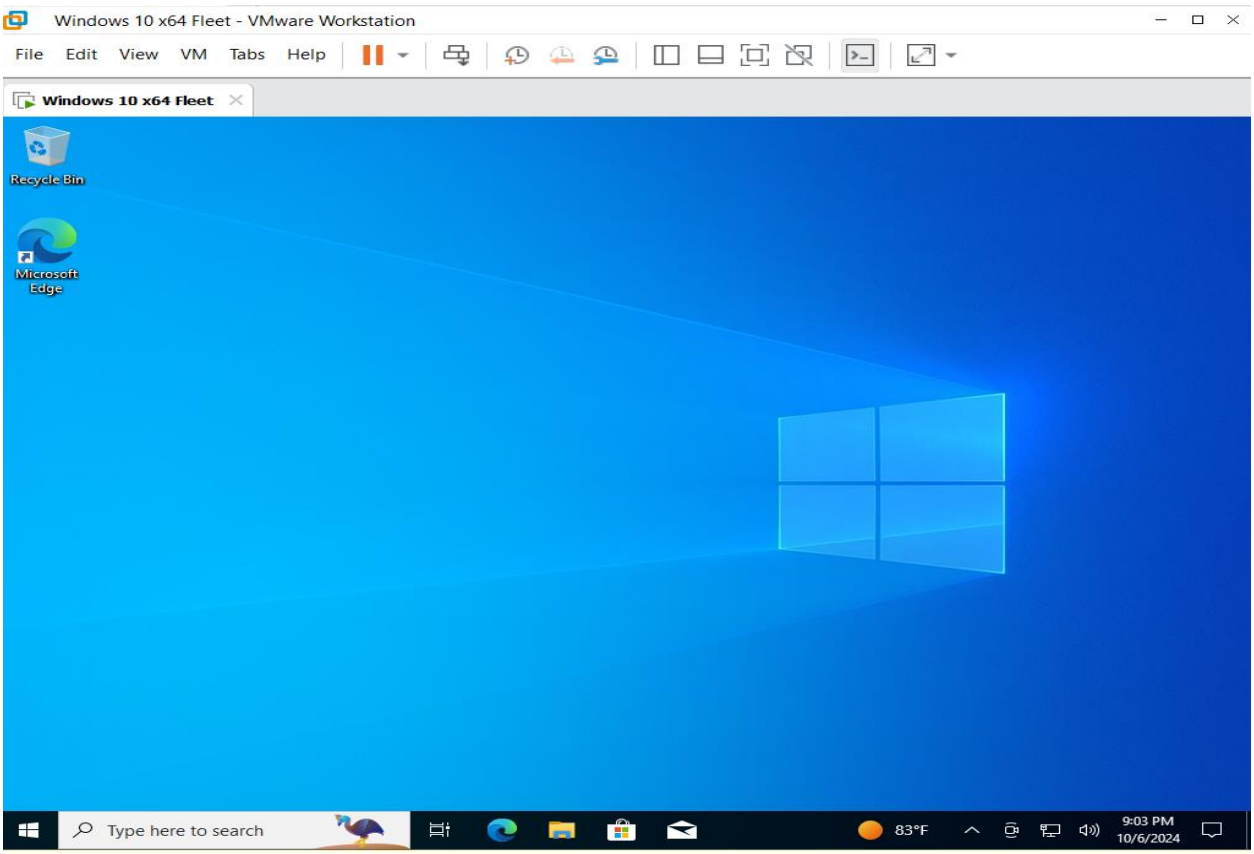

Figure 6. Screenshot of Step 5 - Windows 10 installed on VMWare

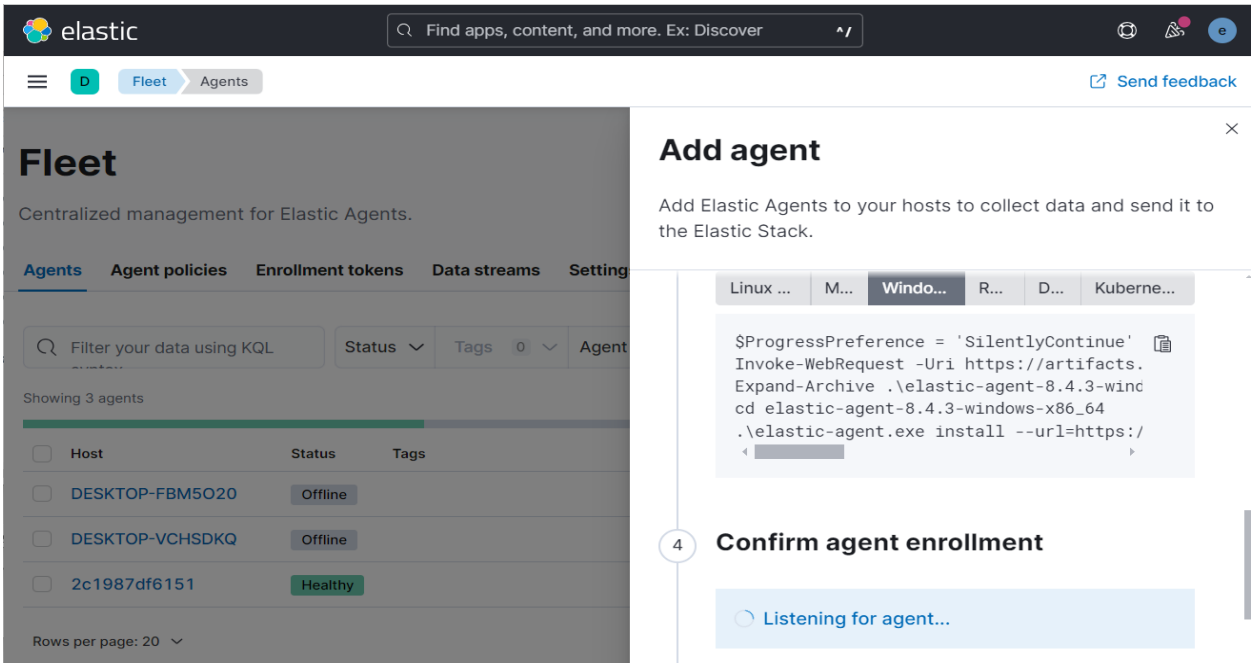

Figure 7. Screenshot of Step 6 -Add Windows Policy

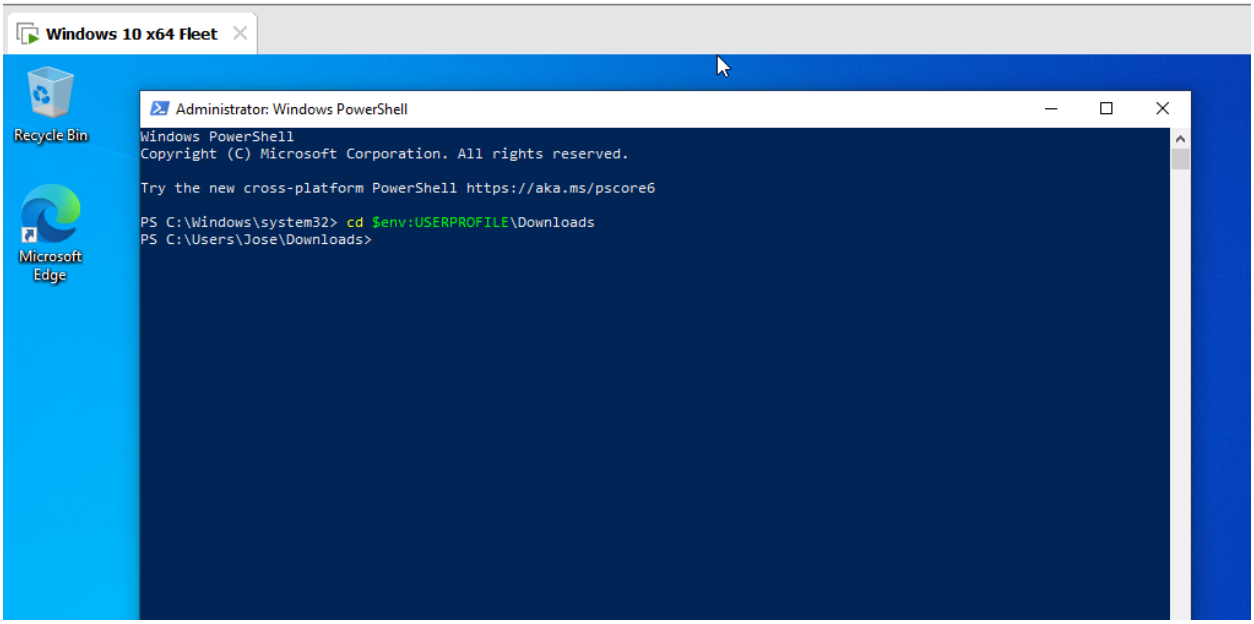

Figure 8. Screenshot of Step 7 - PowerShell open as administrator - Agent Enrollment

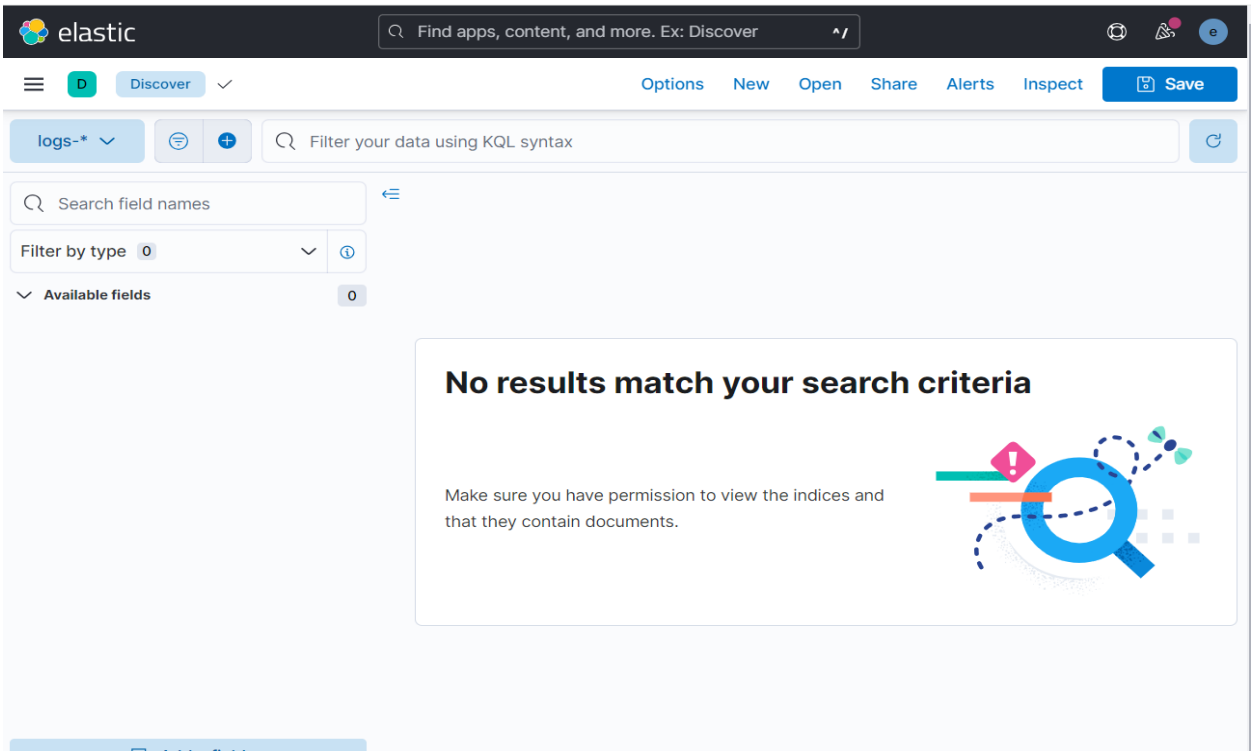

Figure 9. Screenshot of Step 8 - No Logs

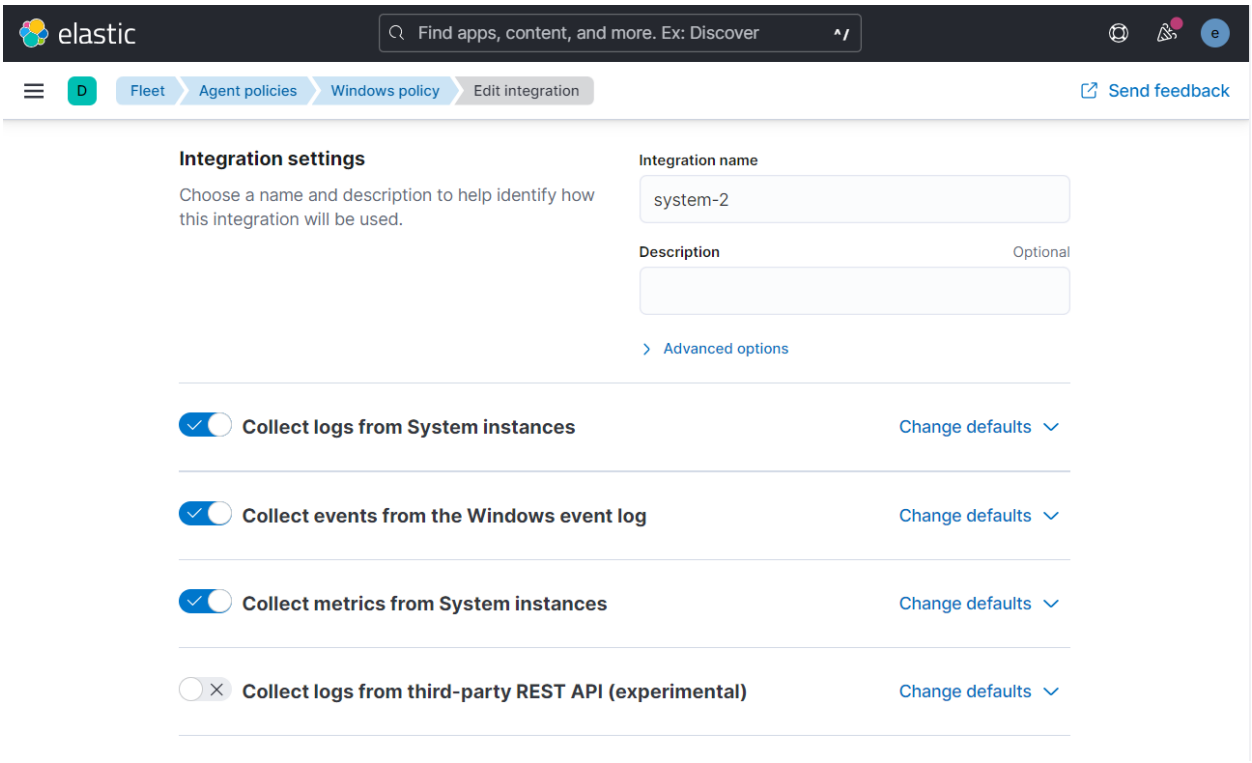

Figure 10. Screenshot of Step 9 - Windows Logs Checked

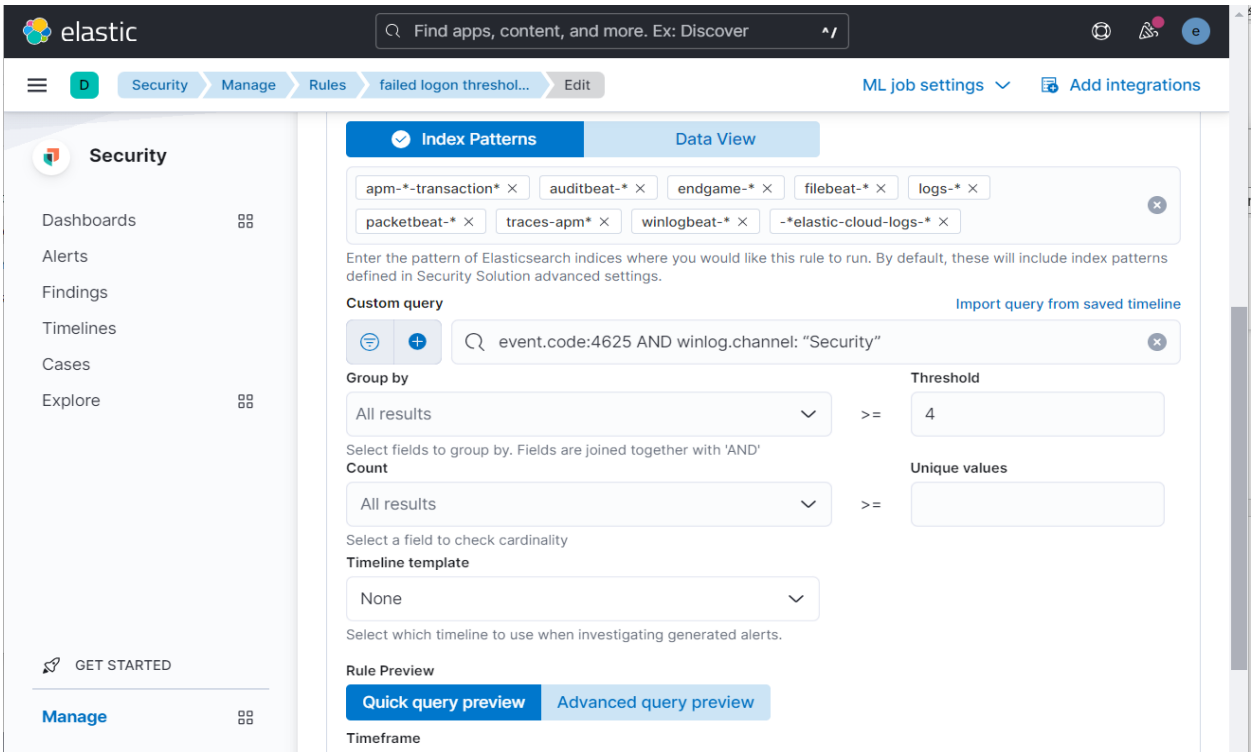

Figure 11. Screenshot of Step 10 - Rule creation

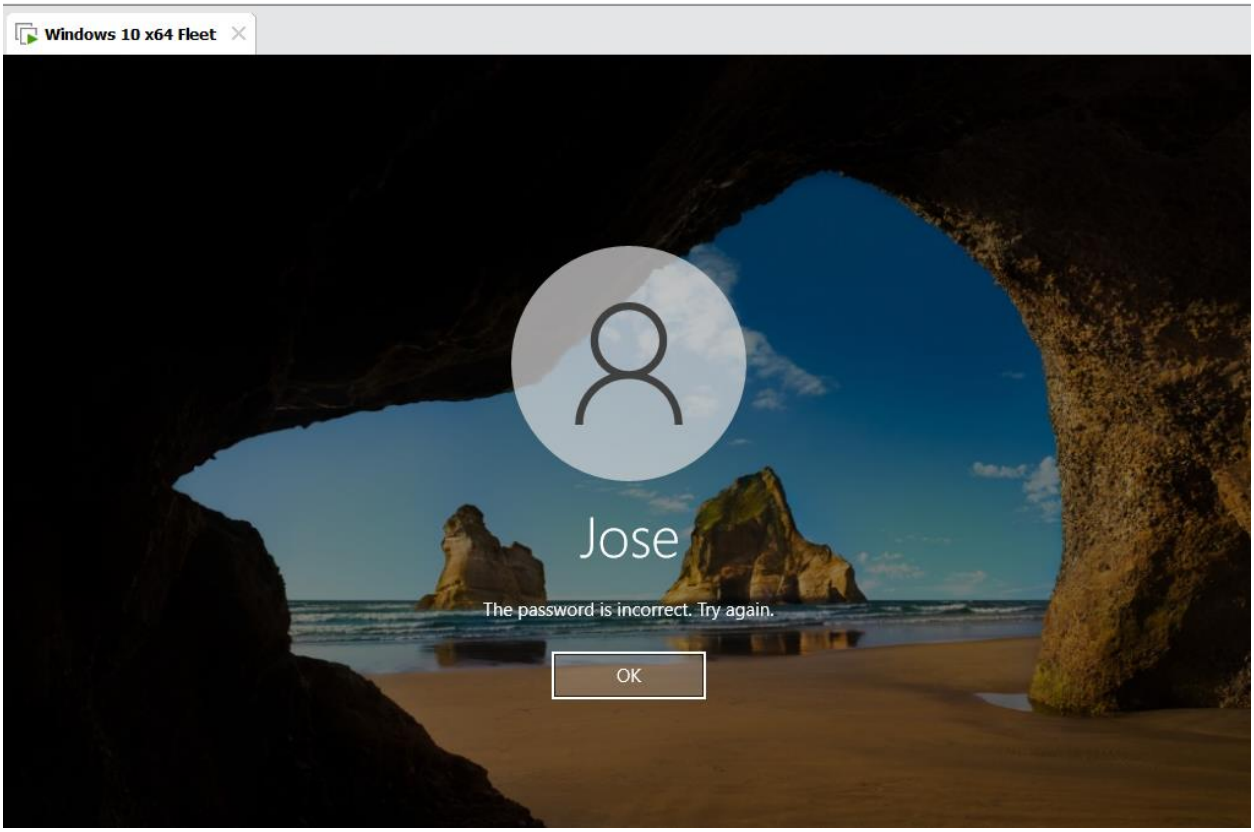

Figure 12. Screenshot of Step 11 - Wrong Password

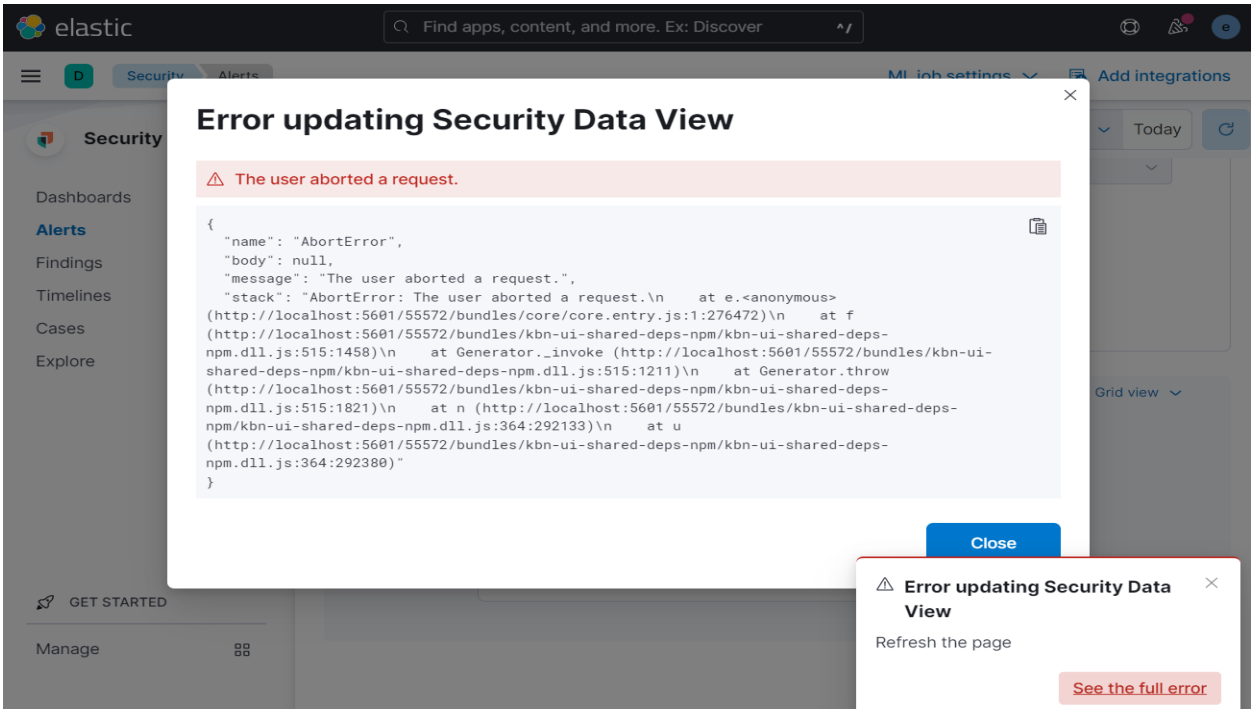

Figure 13. Screenshot of Step 12 - Unable to retrieve security alerts

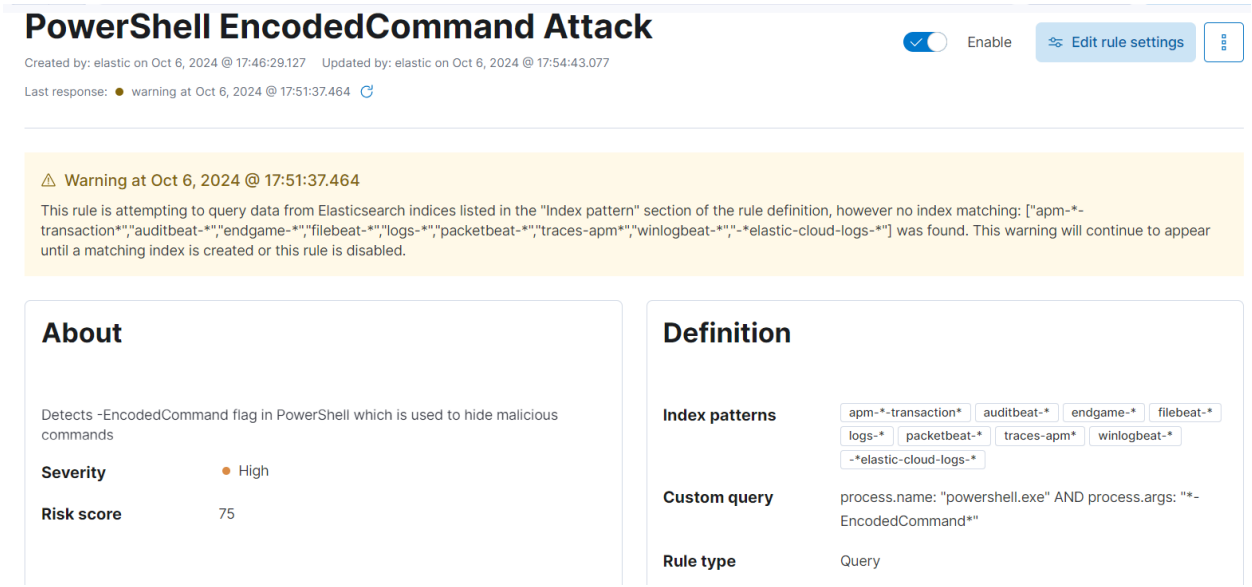

Figure 14. Screenshot of Rule Created for Section 2

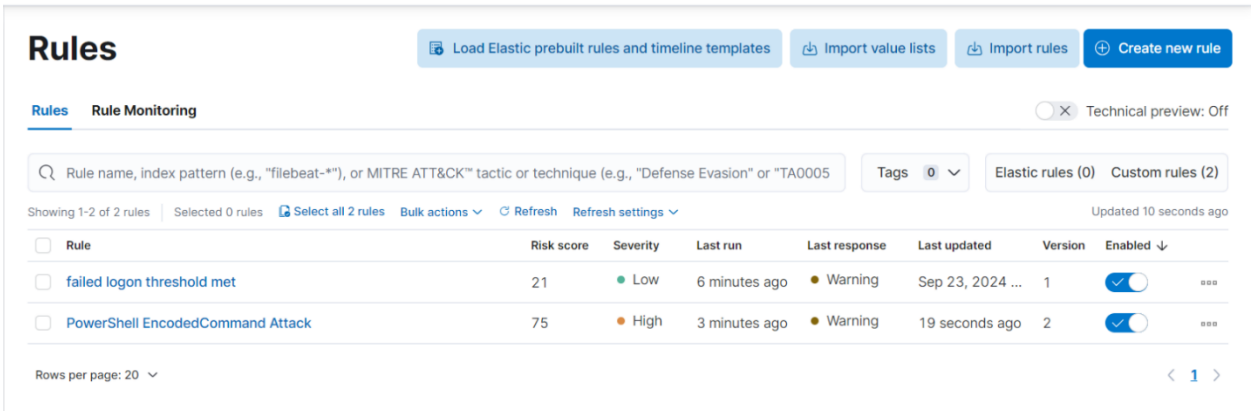

Figure 15. Screenshot of all rules created

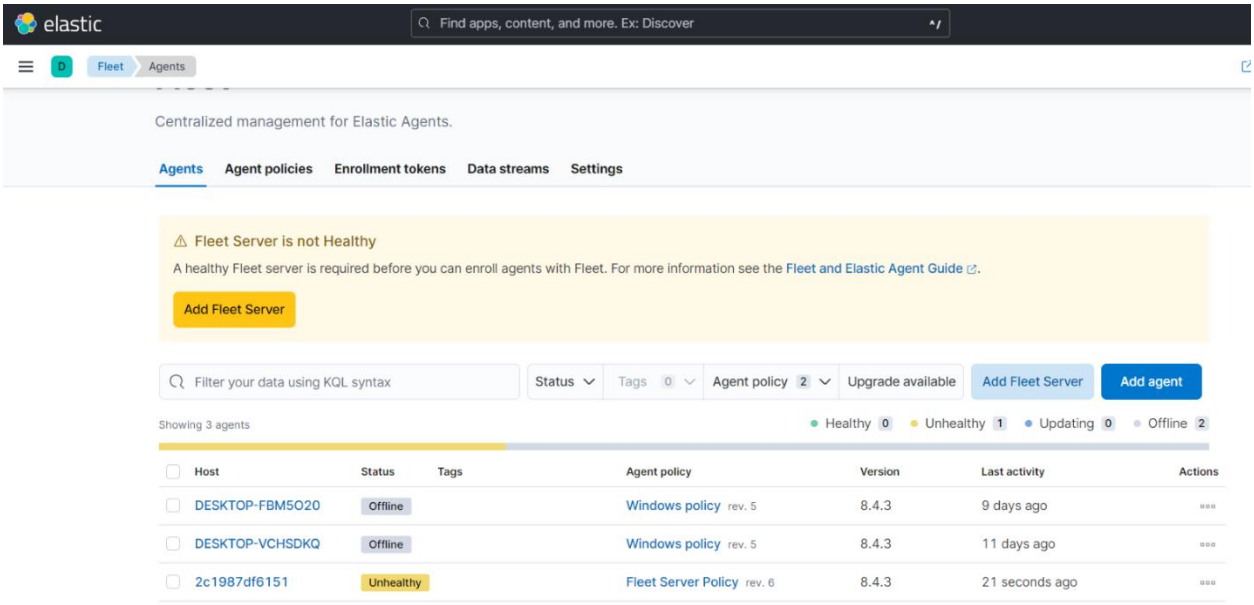

Figure 16. Screenshot of Fleet dashboard – Not working

Bibliography

Elastic. (2024). Fleet and Elastic Agent installation layout. Elastic. https://www.elastic.co/guide/en/fleet/current/installation-layout.html

Microsoft. (2024). about_Preference_Variables. Microsoft Learn. Retrieved October 2, 2024, from https://learn.microsoft.com/en-us/powershell/module/microsoft.powershell.core /about/about_preference_variables?view=powershell-7.4

Microsoft Support. (2024). Download files from the web. Microsoft. https://support.microsoft.com/en-us/windows/download-files-from-the-web-abb92c09-af3a-bd99-d279-a89848b54b0b

MITRE ATT&CK. (2024). *Command and Scripting Interpreter: PowerShell*. Retrieved October 2, 2024, from https://attack.mitre.org/techniques/T1059/001/