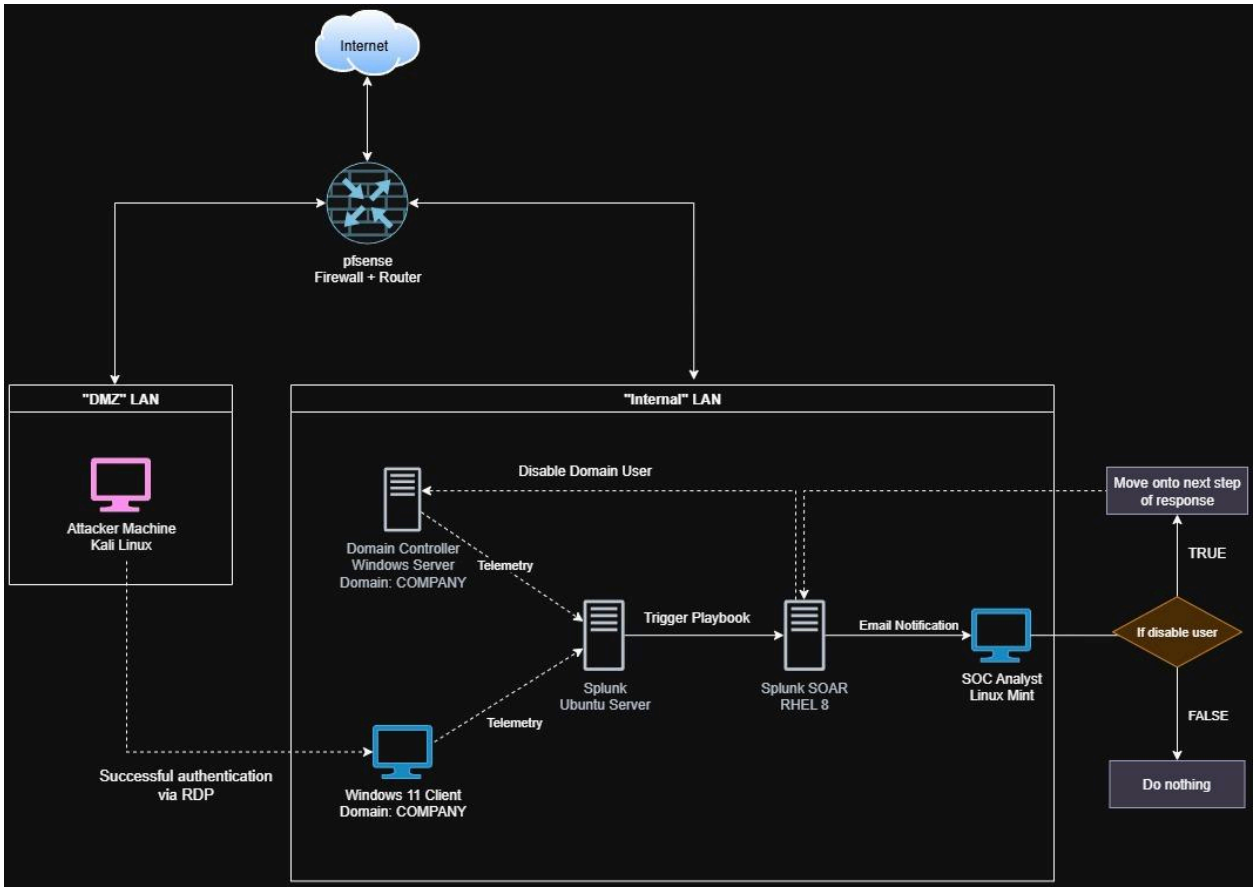
Active Directory + Splunk + SOAR

Diagram:

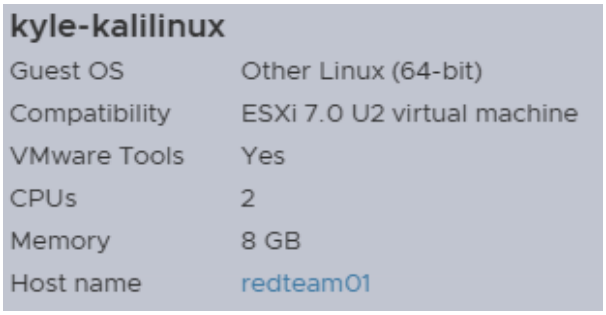


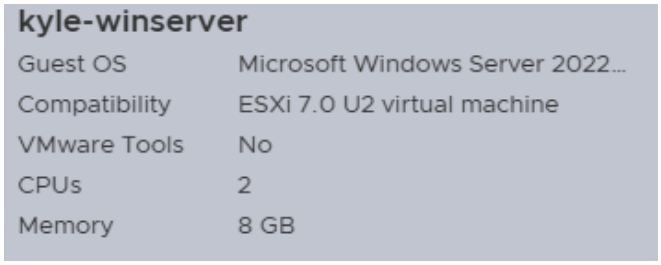
Playbook:

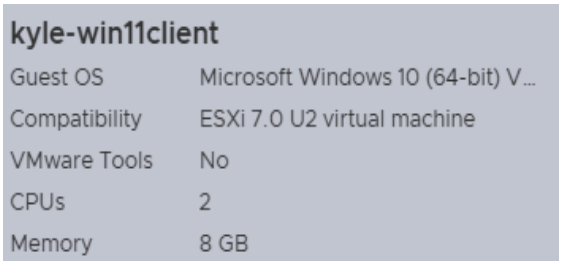
- The SOC Analyst will receive an email notification with information about the alert that was gathered from Splunk
- The Analyst can choose to either say YES, which will then move onto disabling the domain user account in Active Directory, or say NO, which will disregard the alert

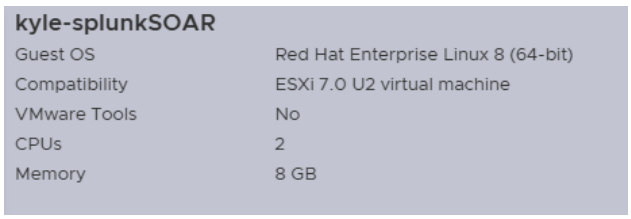
Setup:

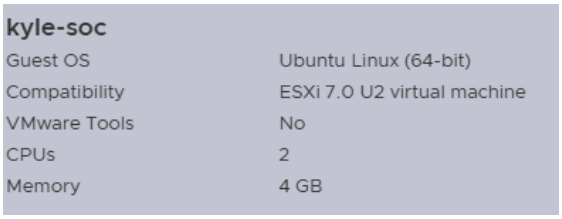
- Splunk Ubuntu Server
- Windows Server 2022 Domain Controller
- Attacker Kali Linux Machine (overkill)
- Windows 11 Client (overkill)
- RHEL 8 Server to host Splunk SOAR [Splunk SOAR Free Trial | Splunk]
- Linux Mint machine to access Splunk Logs. This will be the SOC Analyst computer
- pfSense for FW

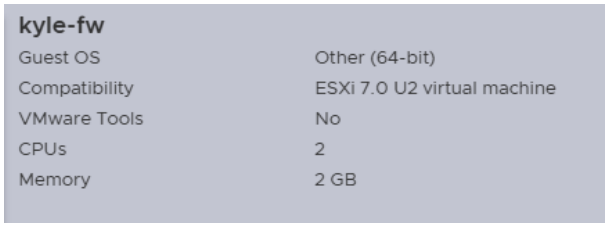












IP Addresses: Setting up all in Private network
192.168.15.1 & 10.10.10.1 - pfSense
192.168.15.8 - Splunk SOAR
192.168.15.9 - splunk-server
192.168.15.10 - AD01 (Windows DC)
192.168.15.11 - Win11Client
192.168.15.12 - SOC Analyst
10.10.10.10 - Kali Linux

Weird issue with Ubuntu setup:
Wasn't showing my IPv4 nor IPv6 address
Fix: Turn the NICs on and Create and set the network config file
sudo ip link set ens192 up
sudo nano /etc/netplan/01-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
   ens192: # External Line
    dhcp4: true

   ens224: # LAN w/ AD01 and Win11Client
    dhcp4: false
    addresses:
      - 192.168.15.9/24 # Static IP for the Splunk Server
    nameservers:
       addresses:
         - 192.168.15.10 # DNS server, which would be the DC since it's able to resolve names
sudo netplan apply

- then try pinging 8.8.8.8
- Now we can do sudo apt-get update && sudo apt-get upgrade -y

<u>Another weird issue</u>:
Splunk-Server and AD01 are unable to ping Win11Client, but win11Client can ping both the Splunk-Server and AD01
Culprit: probably a firewall issue, it's blocking ICMPv4 inbound traffic so outside devices can't reach this one
Type the following in powershell:
netsh advfirewall firewall add rule name="Allow ICMPv4 In" protocol=icmpv4:8,any dir=in action=allow

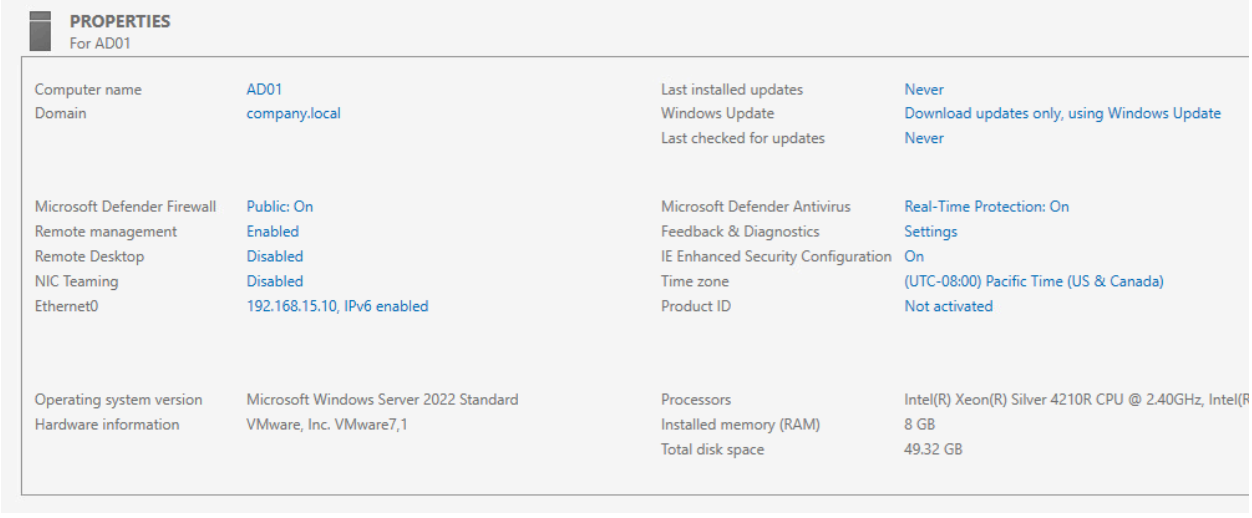
Finally Splunk-Server is able to communicate to Win11Client (192.168.15.11) and AD01 (192.168.15.10), and the other two devices can communicate to each other and the splunk-server now.

```
splunk@splunk-server:~$ ping 192.168.15.11
PING 192.168.15.11 (192.168.15.11) 56(84) bytes of data.
64 bytes from 192.168.15.11: icmp_seq=1 ttl=128 time=0.711 ms
64 bytes from 192.168.15.11: icmp_seq=2 ttl=128 time=0.548 ms
64 bytes from 192.168.15.11: icmp_seq=3 ttl=128 time=0.665 ms
^C
--- 192.168.15.11 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.548/0.641/0.711/0.068 ms
splunk@splunk-server:~$ ping 192.168.15.10
PING 192.168.15.10 (192.168.15.10) 56(84) bytes of data.
64 bytes from 192.168.15.10: icmp_seq=1 ttl=128 time=0.513 ms
64 bytes from 192.168.15.10: icmp_seq=2 ttl=128 time=0.618 ms
64 bytes from 192.168.15.10: icmp_seq=3 ttl=128 time=0.648 ms
64 bytes from 192.168.15.10: icmp_seq=4 ttl=128 time=0.551 ms
^C
--- 192.168.15.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 0.513/0.582/0.648/0.053 ms
splunk@splunk-server:~$
```

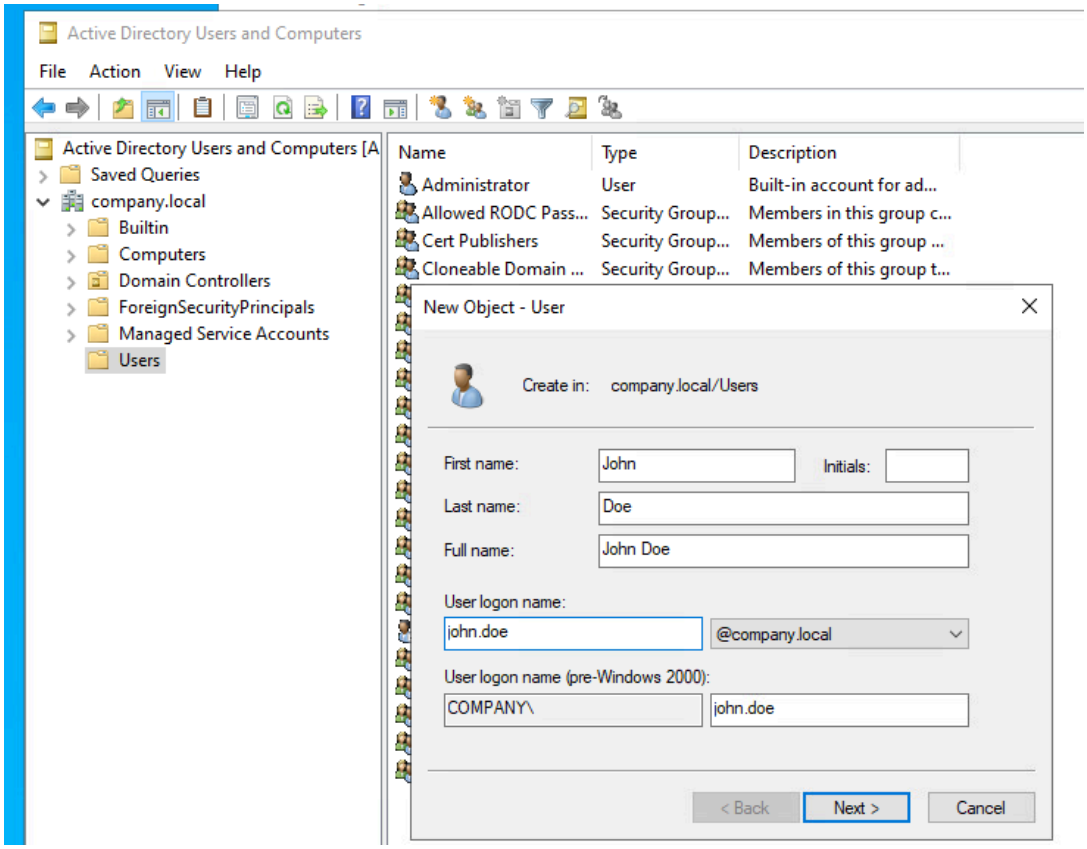
Next step: Create a domain, create an account that is going to be compromised, and join Win11client to the domain so that you can login with the compromised account

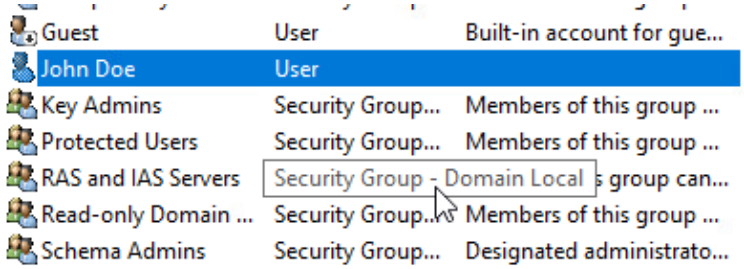
Setup of Windows Server:

- Add “Active Directory Domain Services” role and complete the flagged items
- Created a domain called company.local w/ the NetBIOS of COMPANY

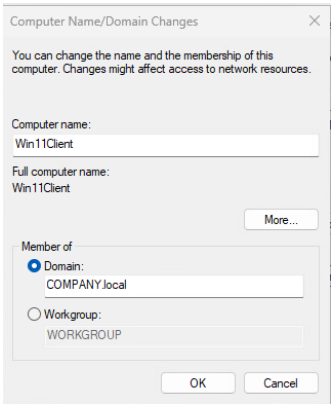


Create a new user “John Doe” in ADUC under Users

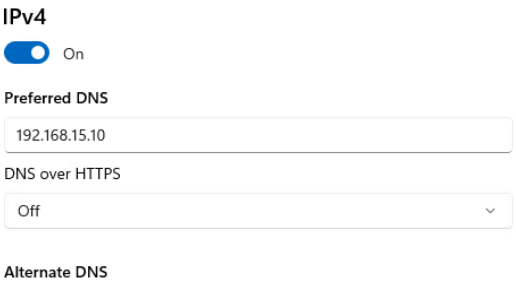




On the Win11Client, go to Advanced System Settings, then join to the COMPANY domain



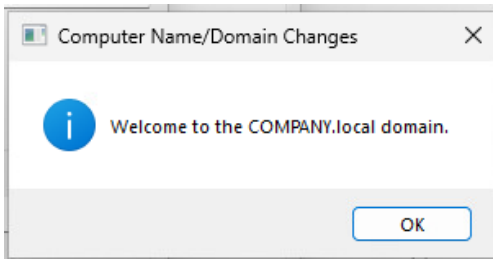
- You will run into an issue when trying to join to a domain and it can’t resolve the name. This is a DNS issue
- the fix for it is to make sure that in your IPv4 Network properties, you set AD01’s IP address as your DNS since AD01 has the DNS role configured



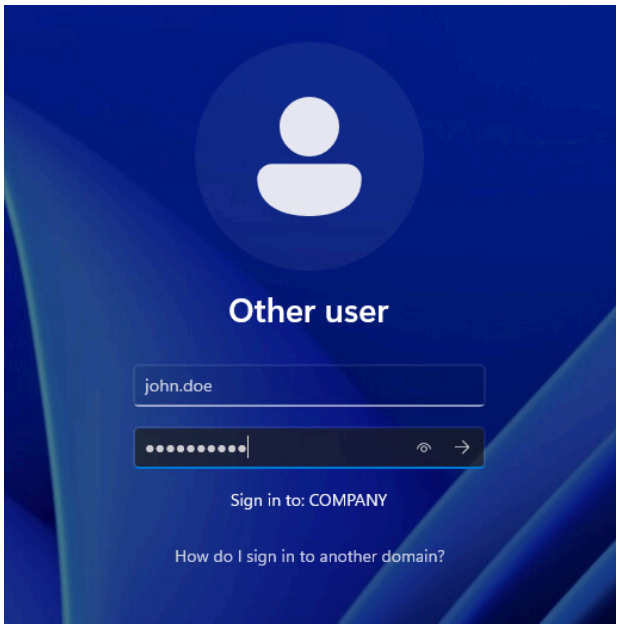
- **Alternate DNS**

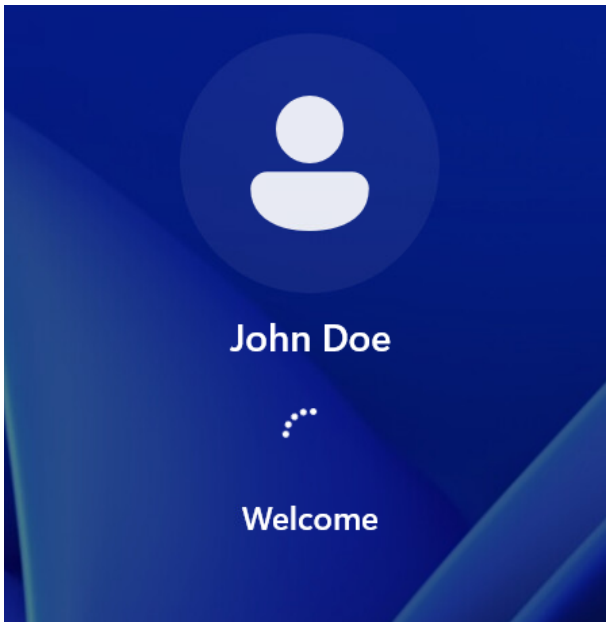
- If you want to be able to use the internet, set the DNS as 8.8.8.8 and as an alternate, set the AD01 DNS server

Win11Client has now been joined to the domain



You should now be able to sign into the computer with the “John Doe” credentials

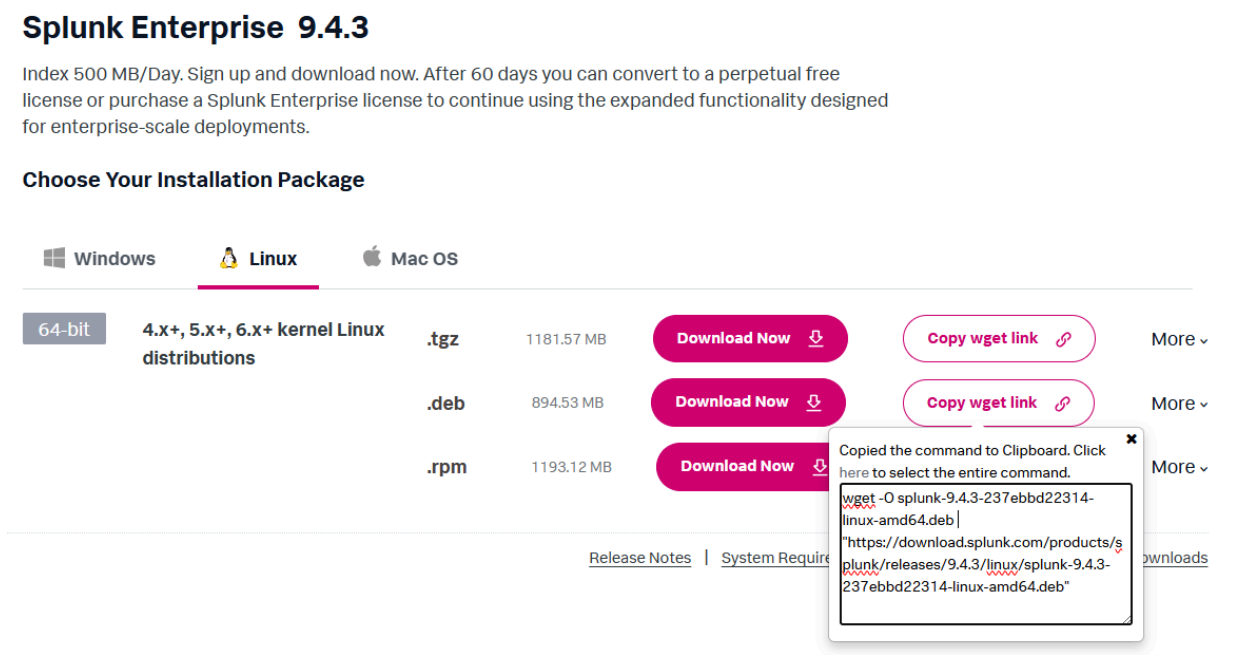




Next step: Install & Configure Splunk, send Telemetry from Windows to Splunk and Create Splunk Alerts

- We will be using Splunk Universal Forwarder on anything sending telemetry to Splunk server
    - Why we need it:
      You need UF when you're **collecting logs from remote systems**
      Event logs, syslogs, application logs
      You want lightweight, secure, and efficient forwarding
      UF uses minimal CPU/memory and it's secure (encrypted via SSL)
      Source machine doesn't allow Splunk Enterprise installation

To download Splunk Server: Go to Splunk Enterprise Free Trial | Splunk and copy the wget link for .deb (since Ubuntu is a debian based linux distro)
Also download the Download Universal Forwarder for Remote Data Collection | Splunk to download the Universal Forwarder to forward (collect and send machine data) to the Splunk server



Run the wget command. Then run the following:

```
splunk@splunk-server:~$ sudo dpkg -i splunk-9.4.3-237ebbd22314-linux-amd64.deb
[sudo] password for splunk:
Selecting previously unselected package splunk.
(Reading database ... 124908 files and directories currently installed.)
Preparing to unpack splunk-9.4.3-237ebbd22314-linux-amd64.deb ...
No need to run the pre-install check
Unpacking splunk (9.4.3) ...
Setting up splunk (9.4.3) ...
complete
splunk@splunk-server:~$
```

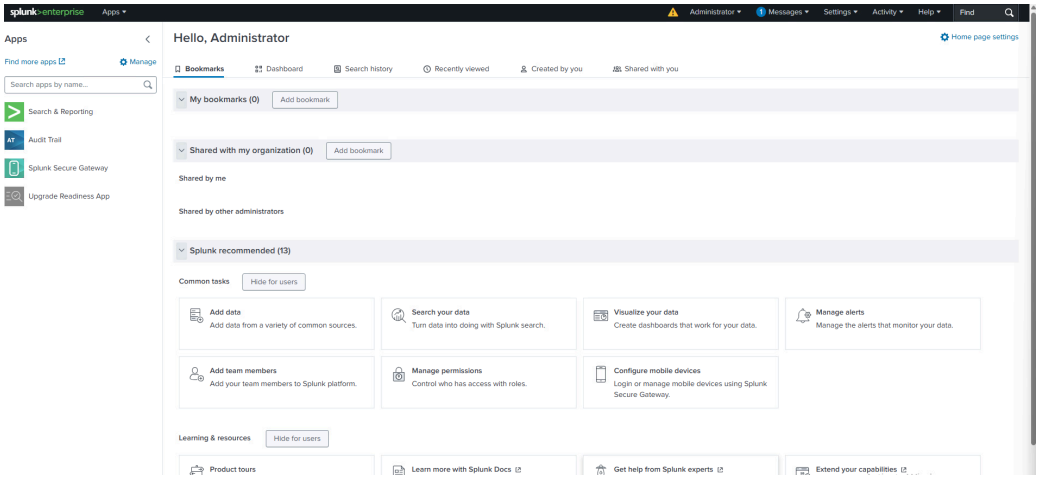
Change directory to /opt/splunk/bin and start splunk by doing ./splunk start:

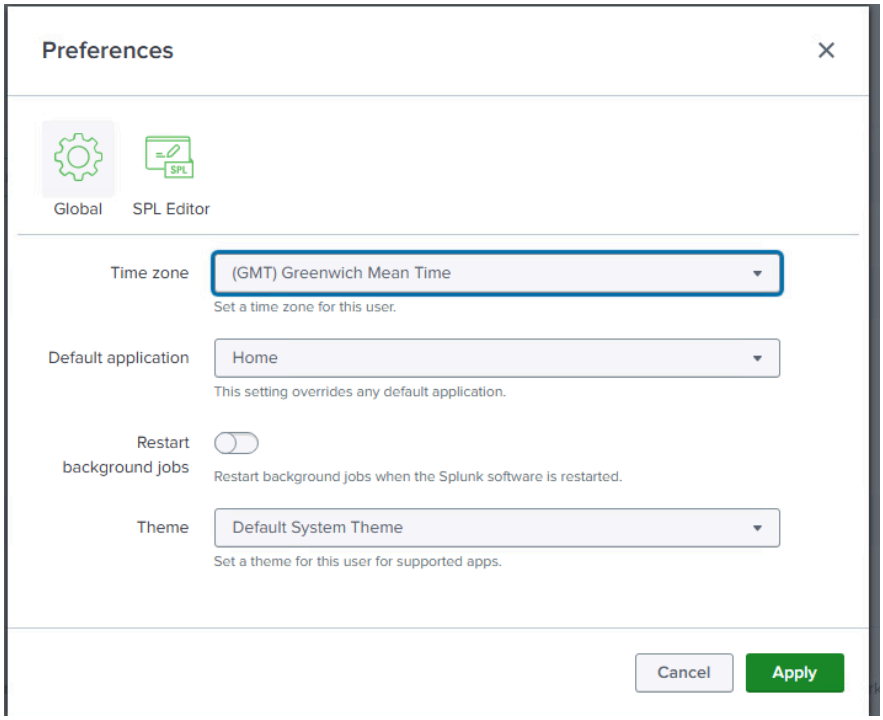
```
splunk@splunk-server:~$ cd /opt/splunk
splunk@splunk-server:/opt/splunk$ ls
bin             etc  include  license-eula.txt  openssl  quarantined_files  share                                                  swidtag
copyright.txt   ftr  lib      LICENSE.txt       opt      README-splunk.txt  splunk-9.4.3-237ebbd22314-linux-amd64-manifest
splunk@splunk-server:/opt/splunk$ cd bin
splunk@splunk-server:/opt/splunk/bin$ ls
2to3-3.7                  fill_summary_index.py   mongod-4.4                    prichunkpng  rapidDiag                        splunkd
2to3-3.9                  genAuditKeys.py         mongod-5.0                    priforgepng  recover-metadata                 splunkmon
bloom                     genRootCA.sh            mongod-6.0                    prigreypng   rest_handler.py                  splunk-optimize
bottle.py                 genSignedServerCert.py  mongod-7.0                    pripalpng    rsync                            splunk-optimize-lex
btool                     genSignedServerCert.sh  mongodump                     pripamtopng  rsync-ssl                        splunk-preinstall
btprobe                   genWebCert.py           mongod_upgrade                pripnglsch   runScript.py                     splunk-tlsd
bzip2                     genWebCert.sh           mongorestore                  pripngtopam  S3benchmark                      supervisor-simulator
classify                  idle3                   noah_self_storage_archiver.py priweavepng  safe_restart_cluster_master.py   tarit.py
ColdStorageArchiver_GCP.py idle3.7                node                          pydoc3       scripts                          tocsv.py
ColdStorageArchiver.py    idle3.9                 openssl                       pydoc3.7     scrubber.py                      tsidxprobe
coldToFrozenExample.py    importtool              parse_xml_buckets.py          pydoc3.9     searchtest                       tsidxprobe_plo
compsup                   installit.py            pcre2-config                  python       setSplunkEnv                     tsidx_scan.py
copyright.txt             jars                    pcregextest                   python3      shc_upgrade_template.py          untarit.py
dbmanipulator.py          jsmin                   pid_check.sh                  python3.7    signtool                         walklex
etcd                      locktest                pip                           python3.7m   slim                             wheel
etcdctl                   locktool                pip3                          python3.9    spl2-orchestrator
etcdutl                   mongod                  pip3.7                        pyvenv       spl-lang-server-sockets
exporttool                mongod-4.2              pip3.9                        pyvenv-3.7   splunk
splunk@splunk-server:/opt/splunk/bin$ ./splunk start
```

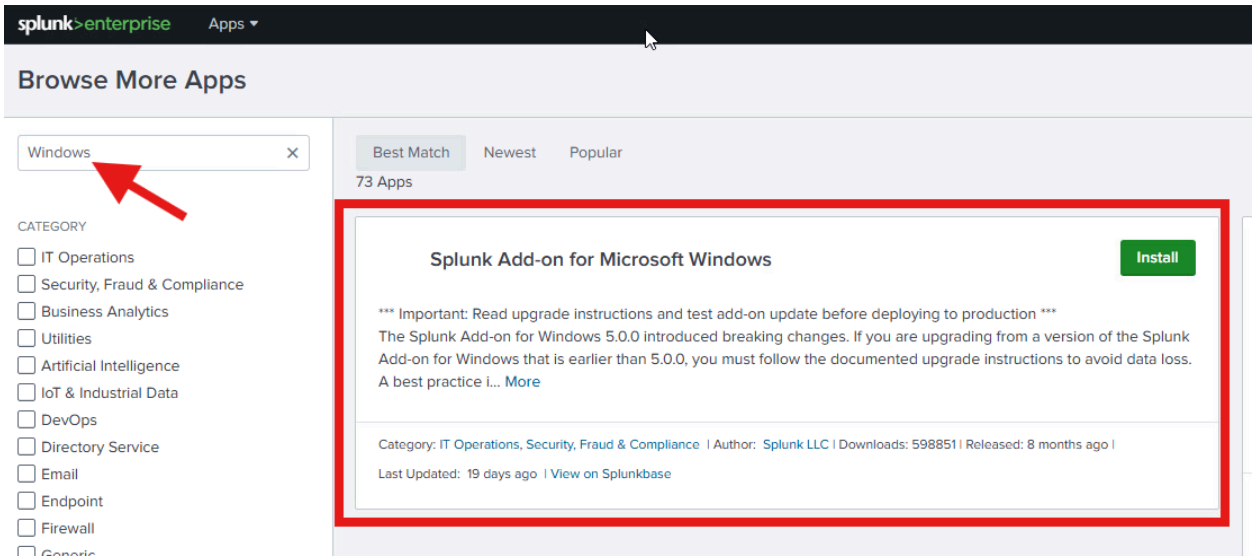
Agree with the terms and assign a username and password:

- Now I am able to access the private IP 192.168.15.9:8000 in my web browser (within the LAN and see the splunk interface)

Splunk Server configuration:







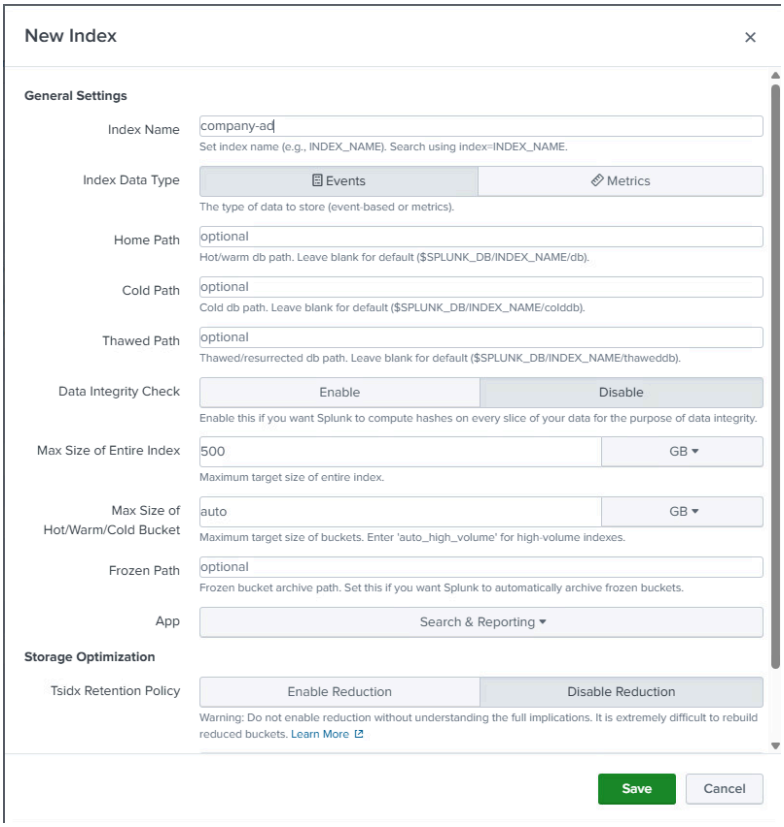
Why do we need this add-on?

- **<u>It's pretty much necessary</u>** since it'll parse raw Windows Event logs and provide us with field extractions and tagging for event logs like
    - Security, System, Application, Setup, Forwarded Events
    - (basically makes Windows logs readable to Splunk and easier to manage)
- The add-on normalizes Windows data to something standard to Splunk and gives you useful fields off the rip
- If installed on Universal Forwarders, it can configure
    - WinEventLog collection
    - Windows performance counters
    - Registry monitoring

- WMI data collection

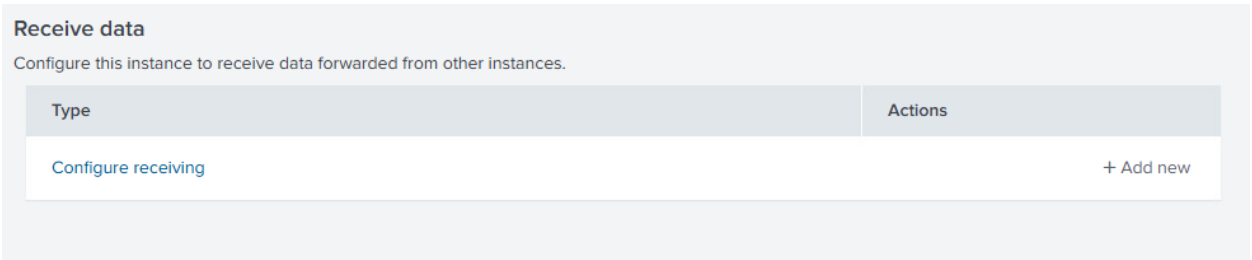
Create a new index:

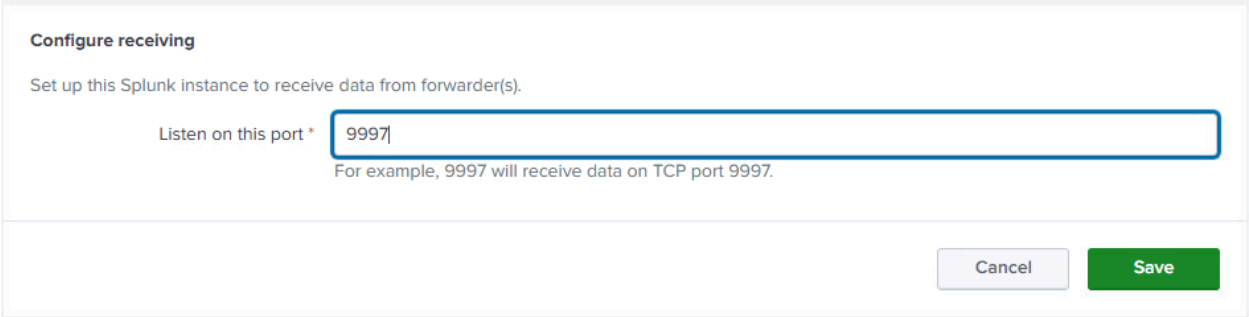
- Why?
  - An index is a data repository where events are stored and is the first layer of separation for managing, searching, and securing data
  - Helps separate data within an org and improve security via access controls, apply data retention policies for individual indexes, and improve search performance, and aid in troubleshooting
- How?
  - Settings -> Indexes -> New Index



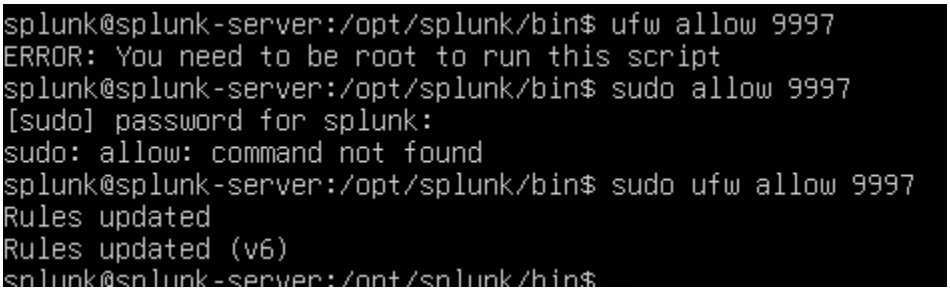
Configure Forwarding and receiving

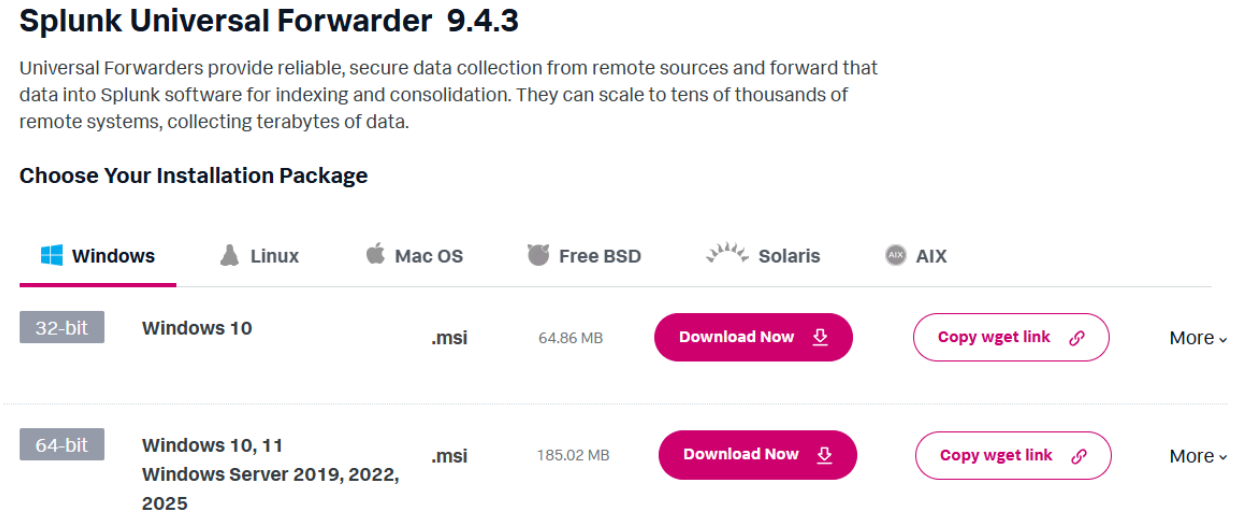
- You will need to configure the splunk server to receive data (basically identify the port that it listens from)
- Settings -> Forwarding and receiving -> Receive data (Configure receiving + Add new)



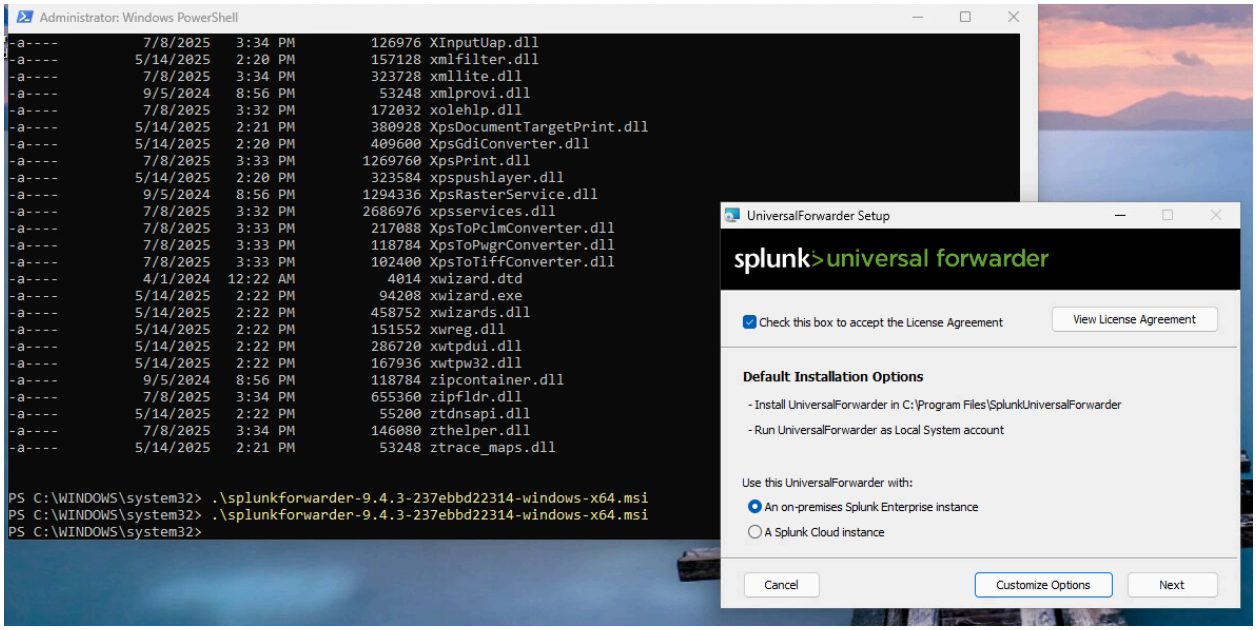


- This is done on the Splunk side, **however** if the Splunk-server (ubuntu) itself has the firewall turned off, you'll need to turn it on to allow for traffic to come through that port
    - sudo ufw allow 9997





- For some reason, Splunk UF is considered a "malicious download" to I will be installing using the wget link on the endpoints
    - wget -O splunkforwarder-9.4.3-237ebbd22314-windows-x64.msi "https://download.splunk.com/products/universalforwarder/releases/9.4.3/windows/splunkforwarder-9.4.3-237ebbd22314-windows-x64.msi"

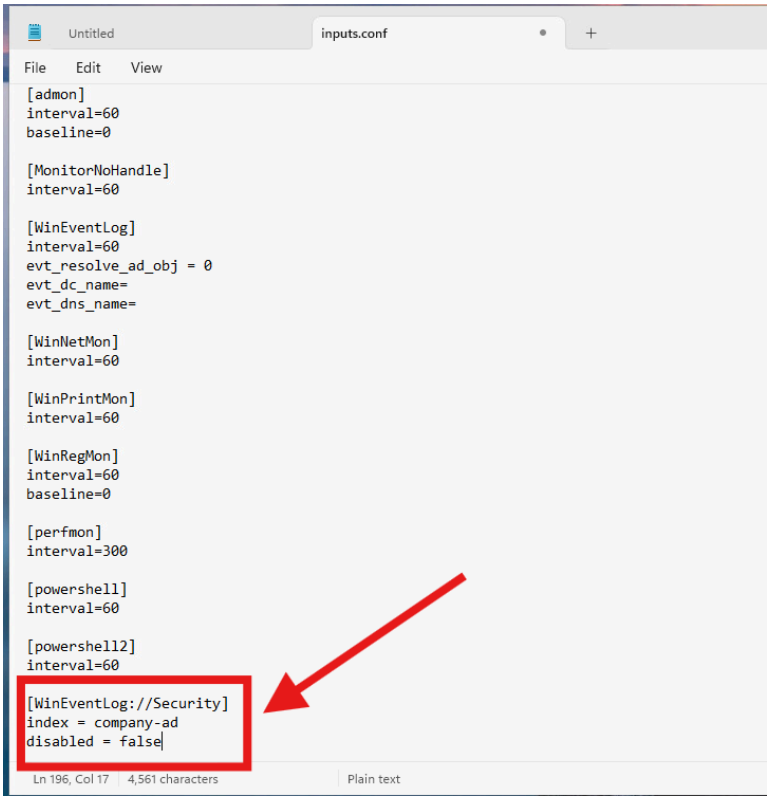


No Deployment Server so no need to fill out. But we DO have a Receiving Indexer, which is our Splunk server (it receives the info and indexes em)

- 192.168.15.9:9997

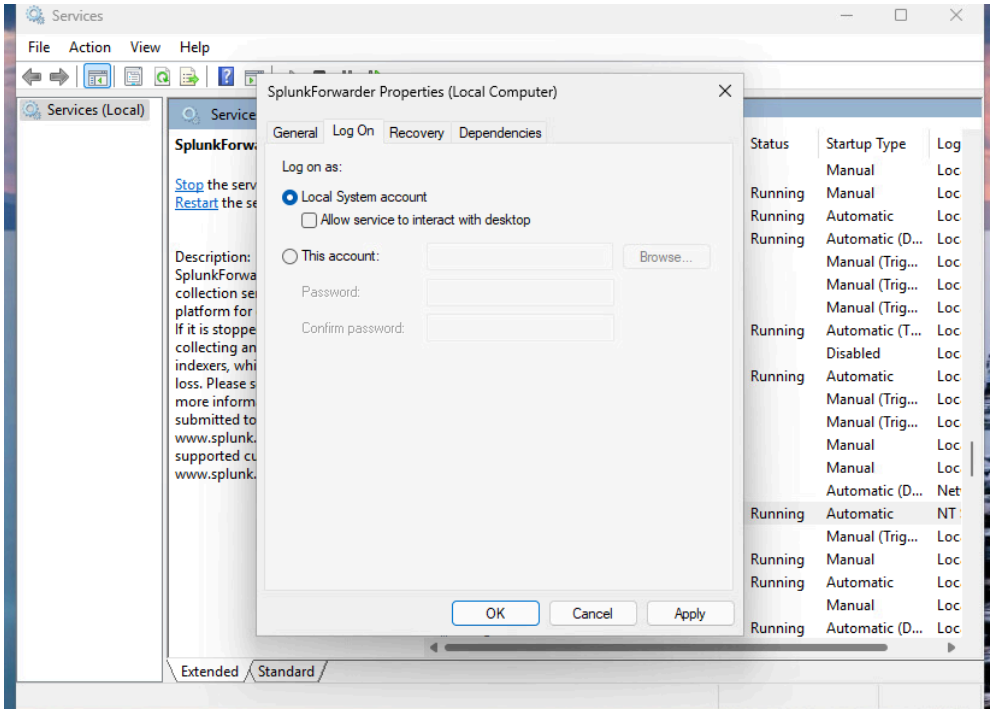
Once we finish downloading Universal Forwarder, we'll need to edit the **inputs.conf** file in the path: **C:\Program Files\SplunkUniversalForwarder\etc\system\local**.

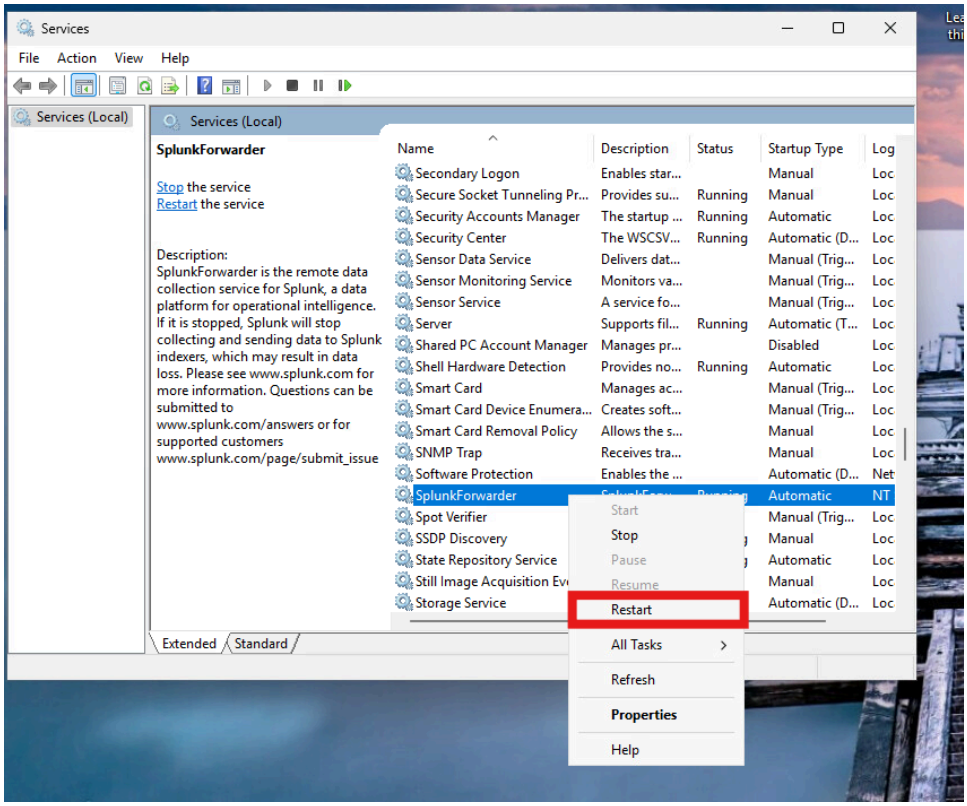
- It may not be there by default, so you can go to C:\Program Files\SplunkUniversalForwarder\etc\system\default and copy the inputs.conf file from there and paste it into local (**You must do it as an administrator**)
- **<u>Why do we need inputs.conf?</u>**
    - This file tells the Universal Forwarder what data to collect (e.g. Windows Event Logs, file paths, perf counters)
        - Where it's coming from
        - How often to poll or monitor
        - Which index and sourcetype to tag the data with
            - In this case we want the one that we created earlier which is: index=company-ad
- After pasting inputs.conf into local, open notepad as **Administrator** and open the inputs.conf file (Change the extension search to All files to find inputs.conf)
    - You will need to do this since you won't have sufficient permissions with a standard account if you want to edit the conf file



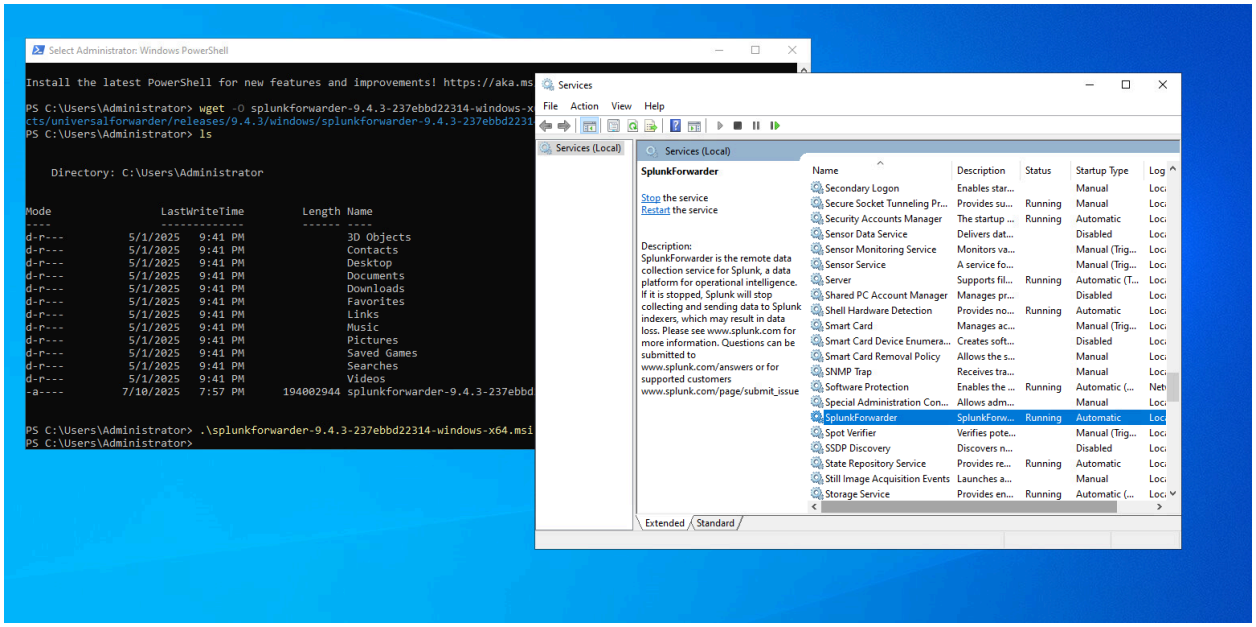
Now that the input.conf file is configured to send Windows Security Events to the index company-ad, we can now restart the Splunk UF service:

- First, change the SplunkForwarder properties to log on as the Local System Account:
  - This is because local accounts have full access to Event logs, Registry
  - And **<u>doesn't require domain credentials to run the service</u>**

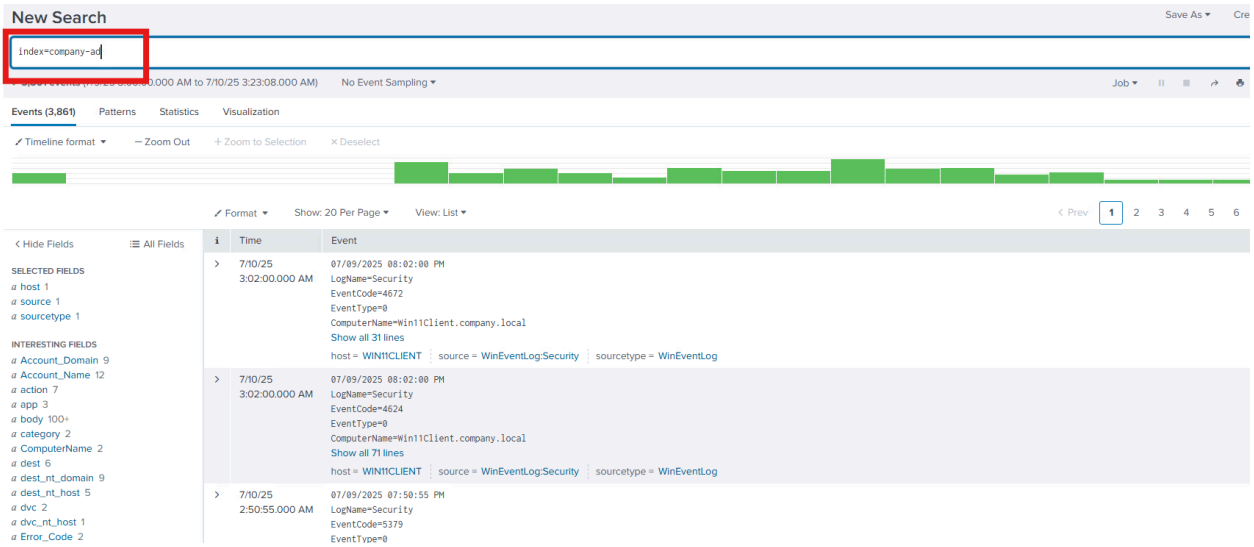




Repeat steps for the Domain Controller.

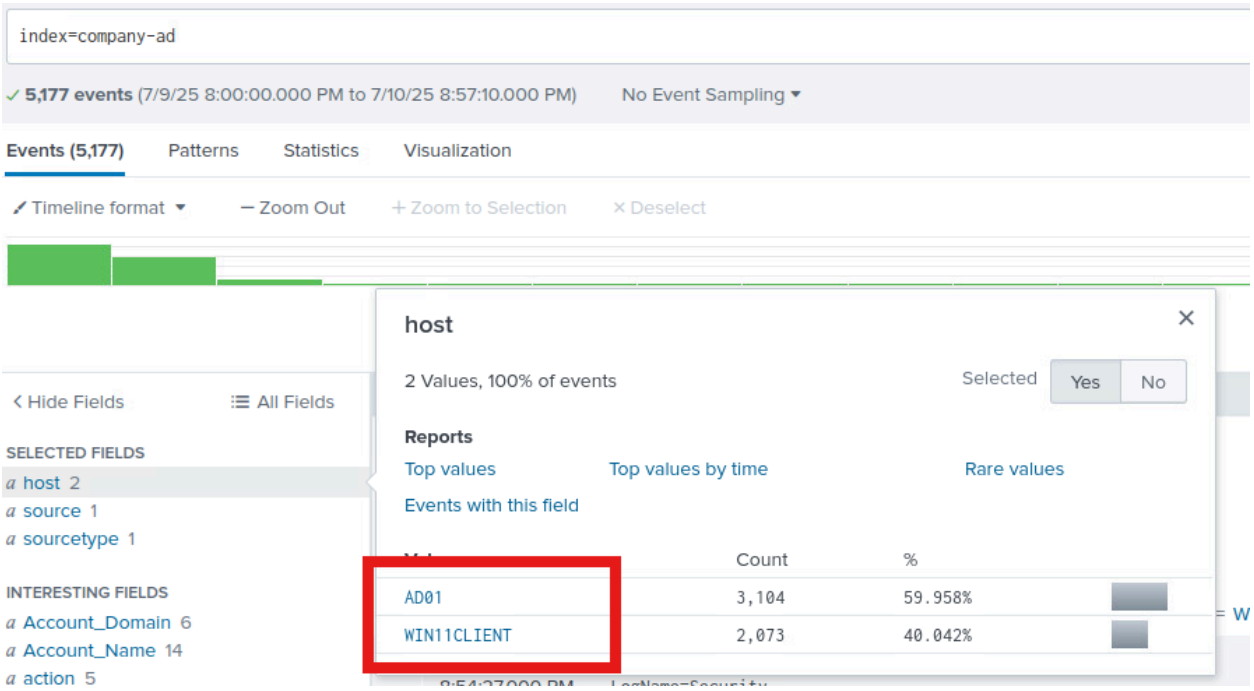


Now that the Splunk Receiving Indexer is set up (Splunk-server) and the Universal Forwarder is set up on the Windows machines, we can see telemetry being sent over to the Splunk server.

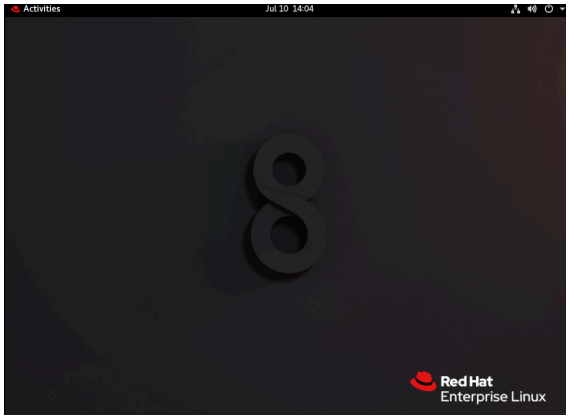


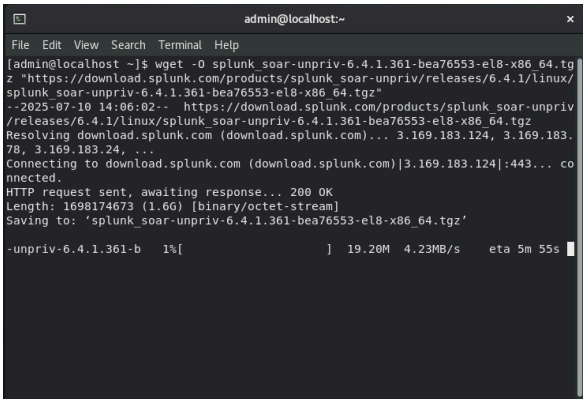
You can see that logs are being forwarded from both devices (AD01 & Win11Client) since there are logs from 2 hosts in Splunk:

- If not, make sure that the service is running and outputs.conf has the right configurations. Also make sure that it is on the same LAN as the Splunk server



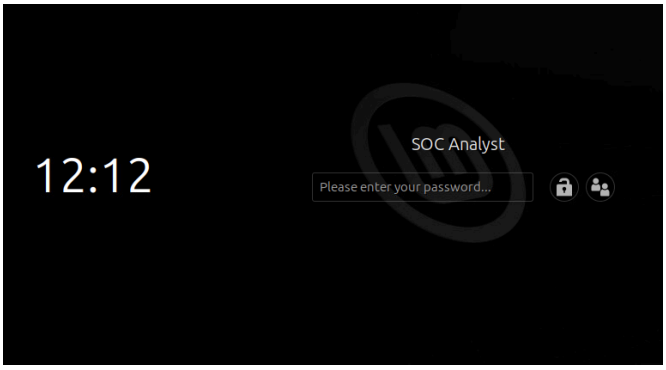
RHEL 8 environment:

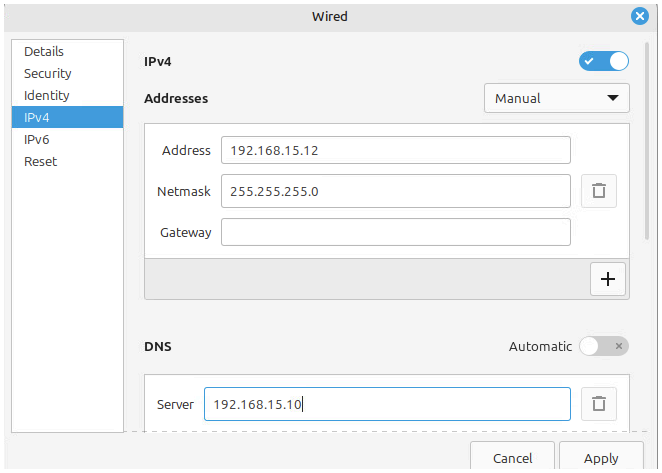




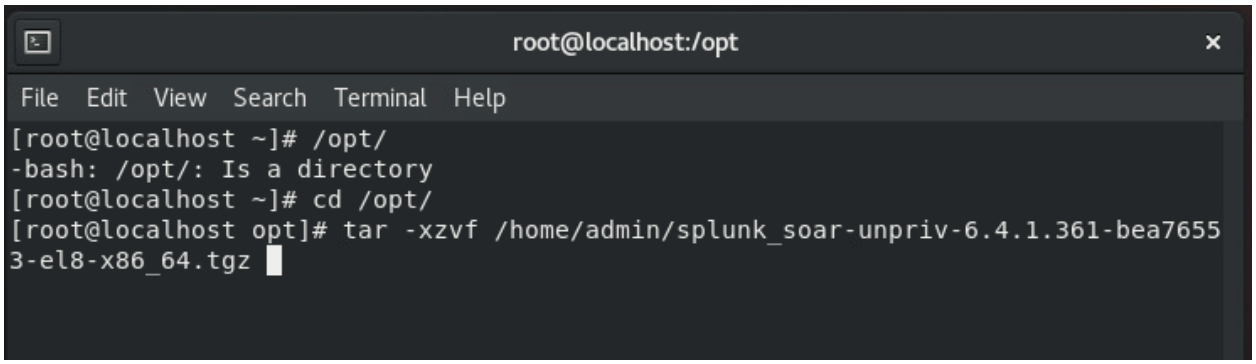
- You will need to do a yum update and yum upgrade **AND you will need to register the system with your RedHat account** https://www.redhat.com/

Mint Environment:

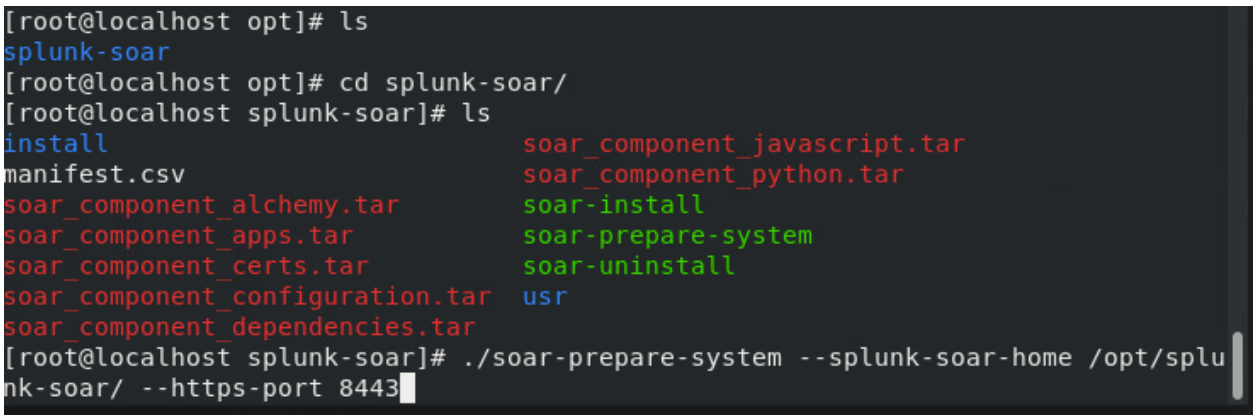




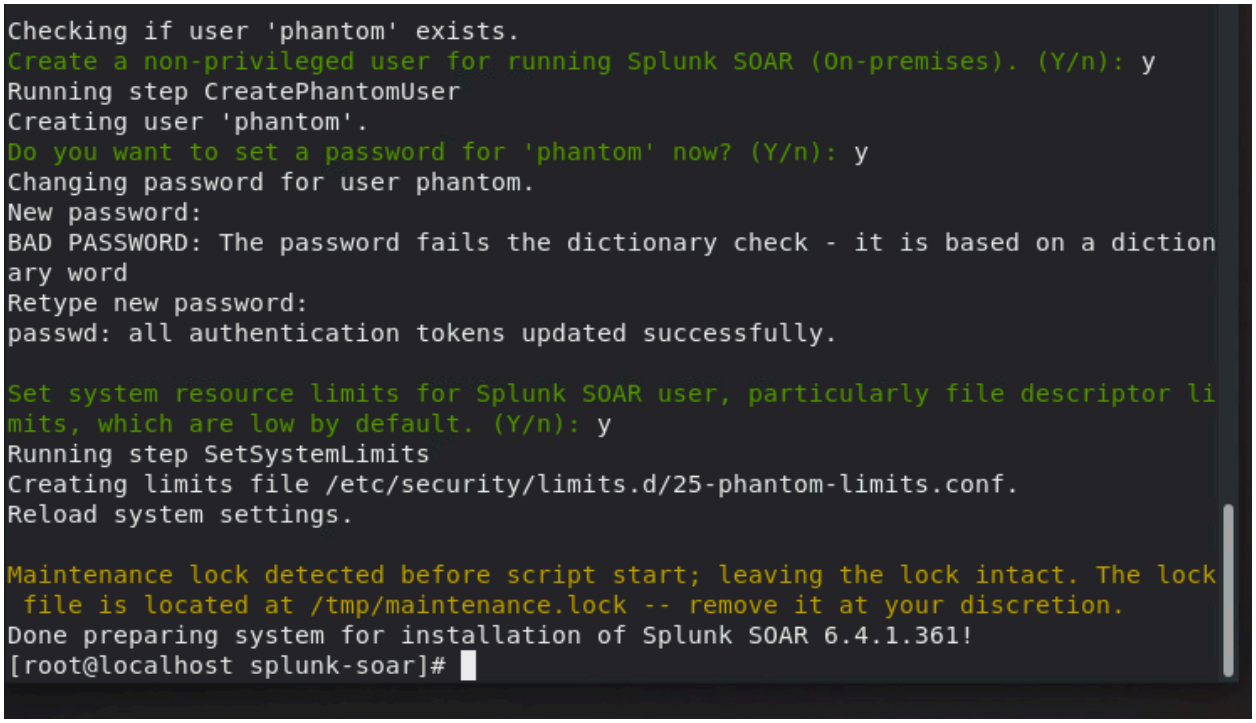
SOAR Setup: [Tutorial: Zero to Splunk SOAR (in UNDER 10 Minutes)]
can do sudo -i to stay in root



Prepare the system:



- This runs the system preparation script, sets the installation path of where SOAR will live on RHEL, and optionally setting the HTTPS port used to connect to SOAR



- As part of the preparation script, it will also create the user "**phantom**" . This is important.

Now that the initial preparation for Splunk SOAR is setup, you will need to change the ownership of splunk-soar to the **phantom** user that was created.

Why?

- **phantom** service (Splunk SOAR) runs under the **phantom** user
- If the install directory is owned by root, the SOAR won't be able to write logs, configurations, or data
  - You'll also get permissions denied errors during runtime
- Changing ownership to **phantom** helps the SOAR service start and run as expected and have proper file permissions

```
Done preparing system for installation of Splunk SOAR 6.4.1.361!
[root@localhost splunk-soar]# cd /opt
[root@localhost opt]# ls
splunk-soar
[root@localhost opt]# chown phantom: splunk-soar/
[root@localhost opt]# ls -lah
total 4.0K
drwx------.  3    1337    1337   25 Jun  6 17:17 .
dr-xr-xr-x. 17 root    root     224 Jul 10 13:51 ..
drwxr-xr-x.  5 phantom phantom 4.0K Jul 10 14:25 splunk-soar
[root@localhost opt]#
```

Now cd into splunk-soar and then switch user:

Switch user (su) from root to phantom by doing:

su phantom

- You won't be able to traverse through the directory as phantom within /opt/ so you should add these permissions to the /opt directory
  - chmod +x /opt

```
[root@localhost ~]# cd /opt
[root@localhost opt]# chmod +x /opt
[root@localhost opt]# ls -lah
total 4.0K
drwx--x--x.  3    1337    1337   25 Jun  6 17:17 .
dr-xr-xr-x. 17 root    root     224 Jul 10 13:51 ..
drwxr-xr-x.  5 phantom phantom 4.0K Jul 10 14:25 splunk-soar
[root@localhost opt]# su phantom
[phantom@localhost opt]$ cd splunk-soar/
[phantom@localhost splunk-soar]$ ls
install                          soar_component_javascript.tar
manifest.csv                     soar_component_python.tar
soar_component_alchemy.tar       soar-install
soar_component_apps.tar          soar-prepare-system
soar_component_certs.tar         soar-uninstall
soar_component_configuration.tar usr
soar_component_dependencies.tar  var
```

Now you can run the soar-install

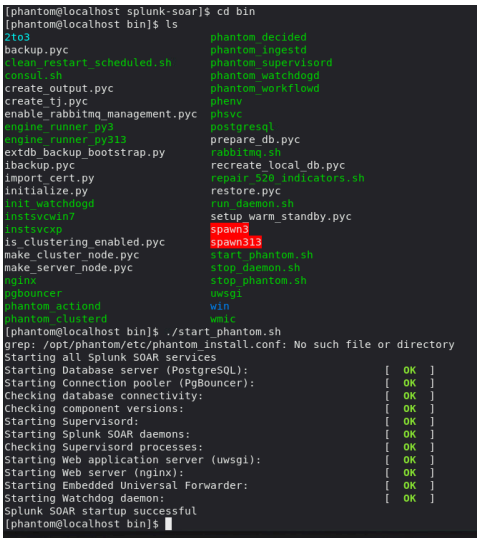
- ./soar-install –splunk-soar-home /opt/splunk-soar/ –https-port 8443

Now that the installation for SOAR is complete, you can switch over to the LAN and change the static IP for both the SOAR machine and the SOC machine

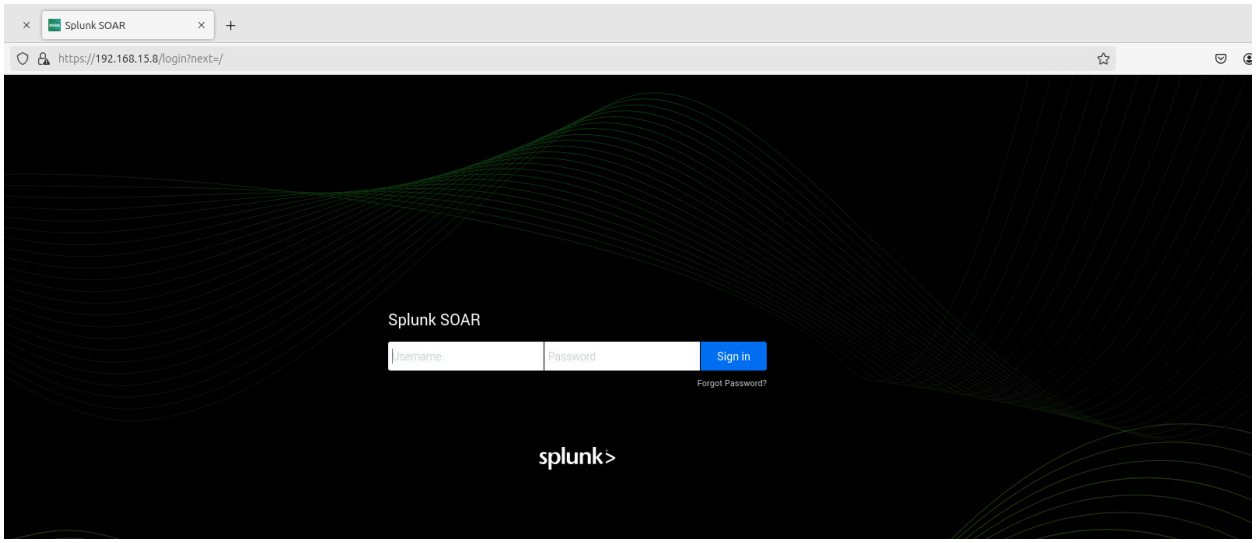
IP Addresses:

- Splunk SOAR: 192.168.15.8, Linux Mint: 192.168.15.12
- Make sure to set the DNS to the AD01
- Also make sure to change VMWare network adapter settings to “Lab-only” so that all devices are within the same lab and they are able to ping each other

Change directories into /opt/splunk-soar/bin and start up Splunk SOAR:

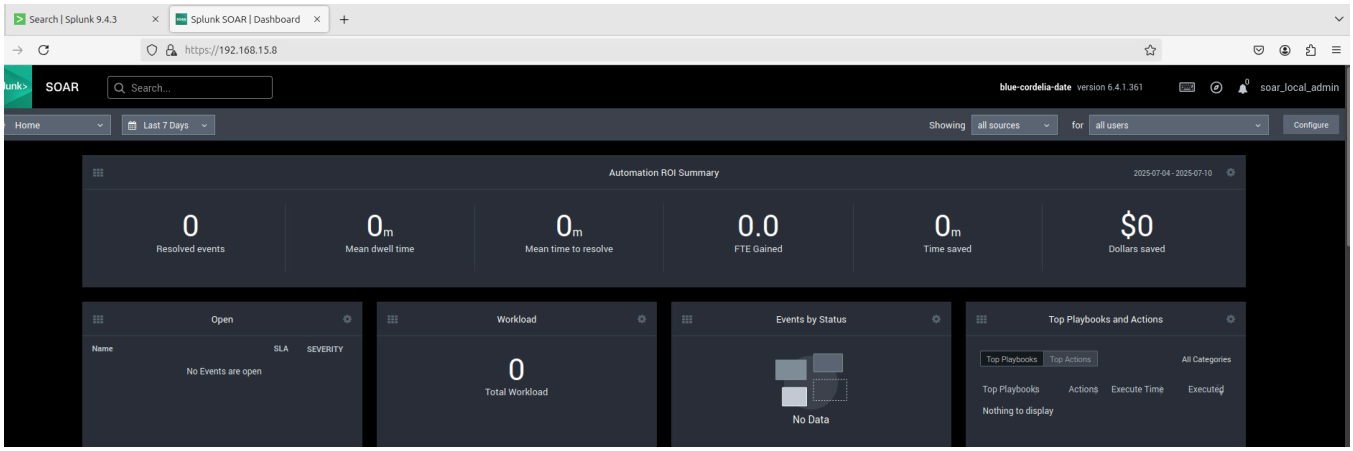


Now in the SOC Analyst computer, you should be able to see the Splunk SOAR platform on the IP of the RHEL server (192.168.15.8):



Default login: soar_local_admin

Dashboard page:



We will now set up the attacking method and circle back to Splunk and make sure that the attack is properly logged within Splunk.

Attacker Setup

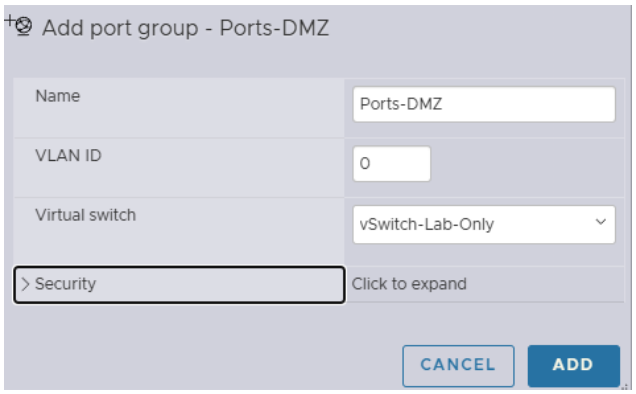
For generating telemetry:

We want to have the Kali VM to be in a different VLAN/subnet than the “company” network (e.g. 10.10.10.0/24) and have an RDP connection from Linux to Windows.

- This will generate telemetry with the source IP 10.10.10.10, simulating that of an external attacker
- The use of private IP's simulates that of a corporate environment since attackers tend to sit on the DMZ systems (since firewalls/routers typically have NAT and translate public IP to private IP within the subnet)

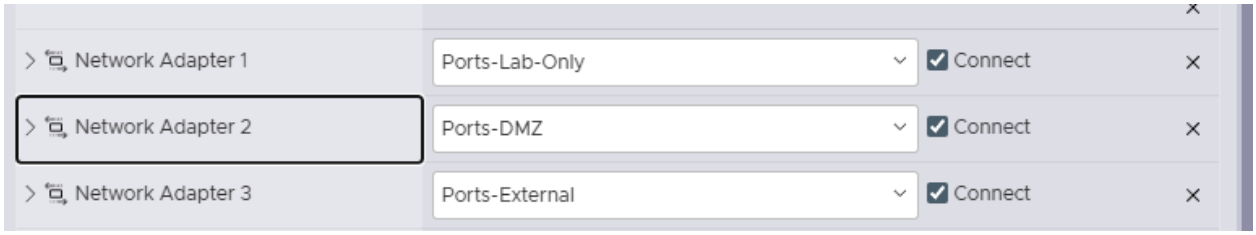
To do this, I setup pfSense to route traffic between them but restrict/allow specific ports (in this case RDP);

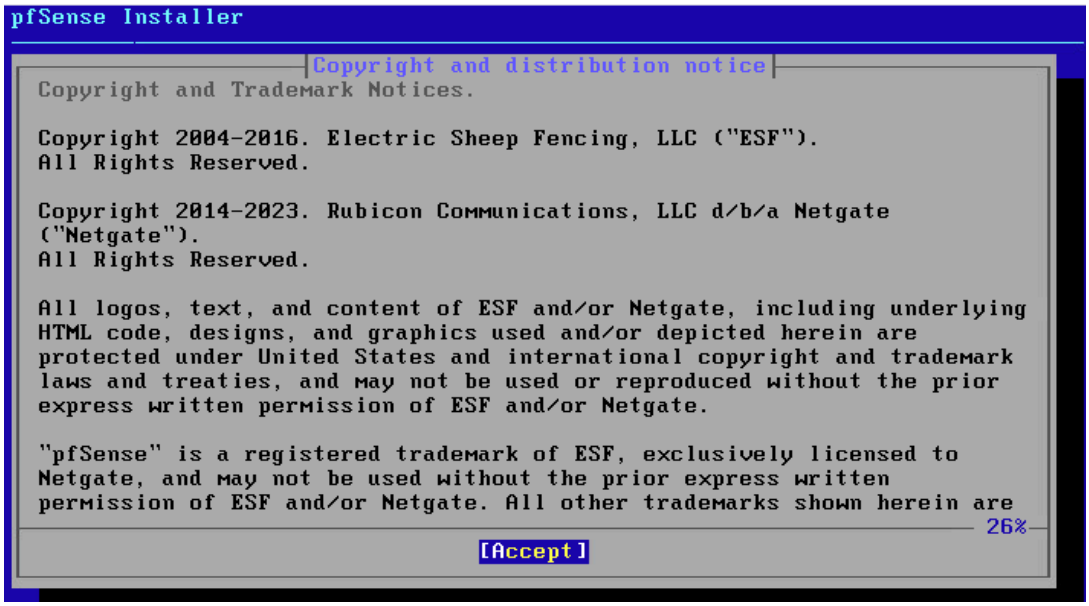
- **This emulates an environment with weak firewall configurations / poorly segmented enterprise network that is allowing attackers to easily RDP.**
    - FW should restrict east-west and north-south traffic across VLANs
- Have a NIC in both subnets and route between them; allow RDP (TCP 3389) inbound
  In VMWare, you can add a port group



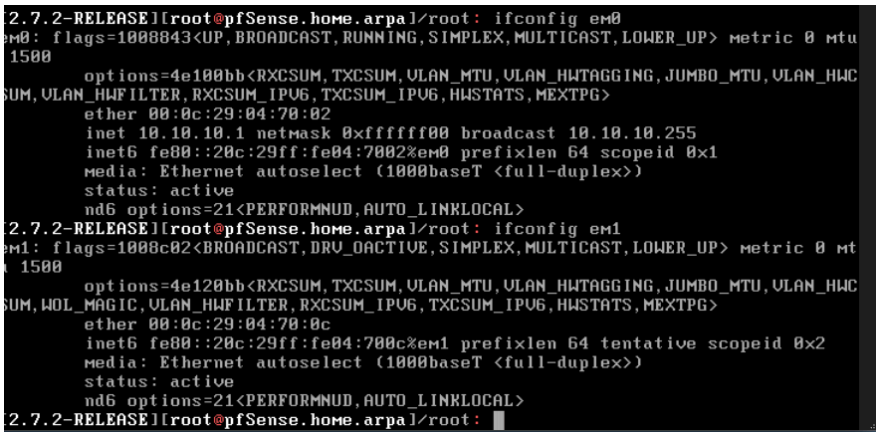
pfSense setup:

- We want both subnets to route through the firewall to gain access to the internet
- We also want the firewall to control traffic between the Lab-Only and DMZ

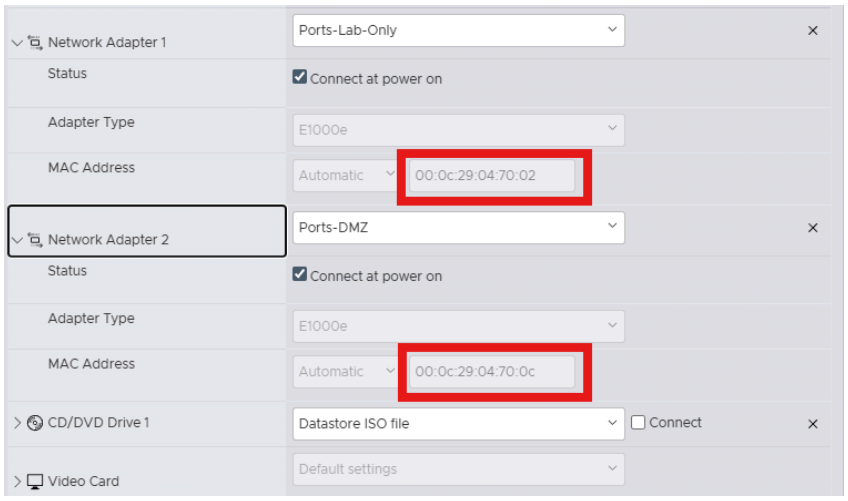




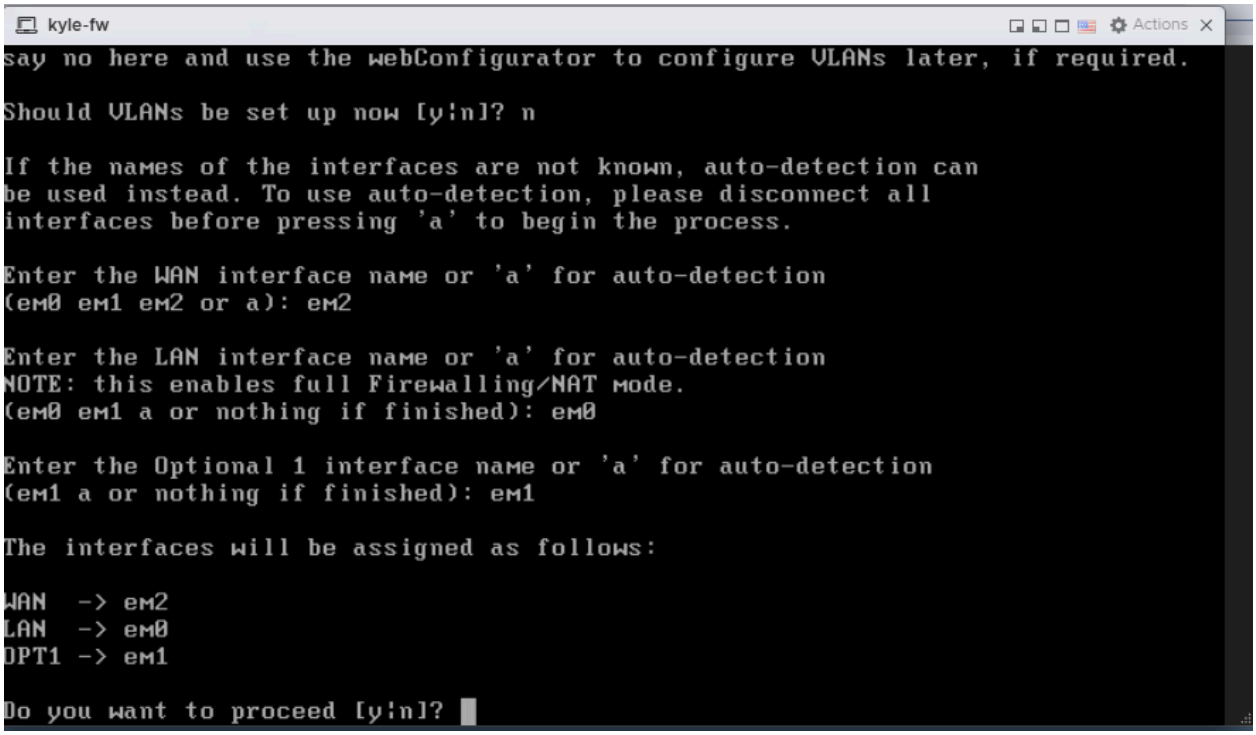
There should be two adapters; em0 and em1. Hit “Shell” (Option 8) and do an ifconfig to see the MAC address:



Then go to VMWare and look at the MAC Addresses:



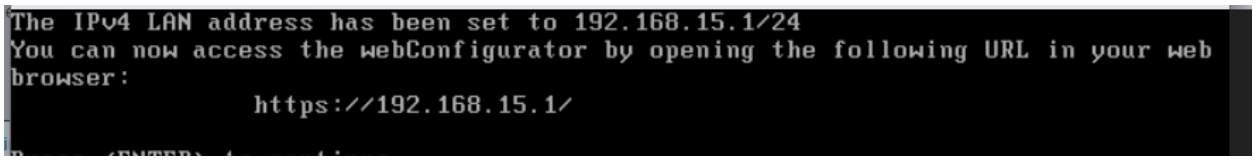
Type exit to get back to the main menu, then Assign Interfaces to each NIC (Option 1). First set the external line to be the WAN:



From there, set the interface IP addresses and set it as static so that each subnet has a gateway (x.x.x.1) (Option 2): **You don't need to configure VLANs within pfsense because you already separated them within VMWare.**

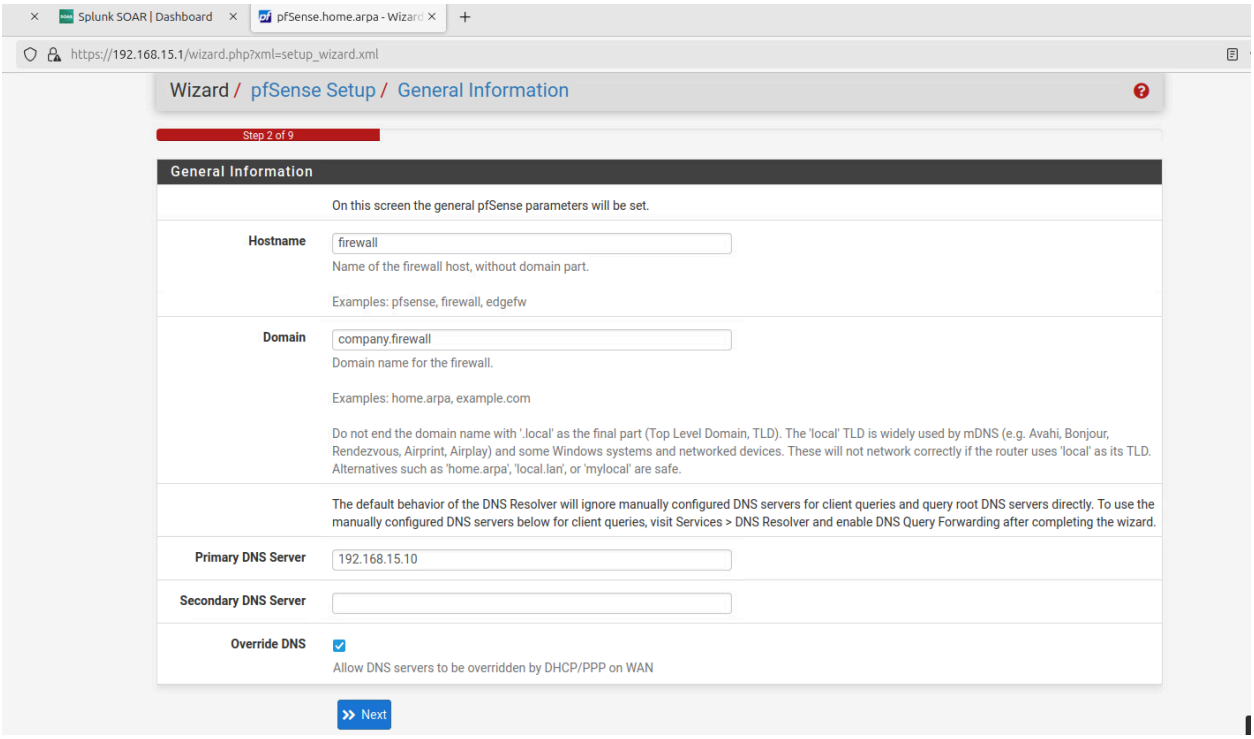
- **You only need to configure VLANs if everything is within the same physical switch and isn't managed by anything else other than pfsense.**

- This is for the Lab-only VLAN/Port group:

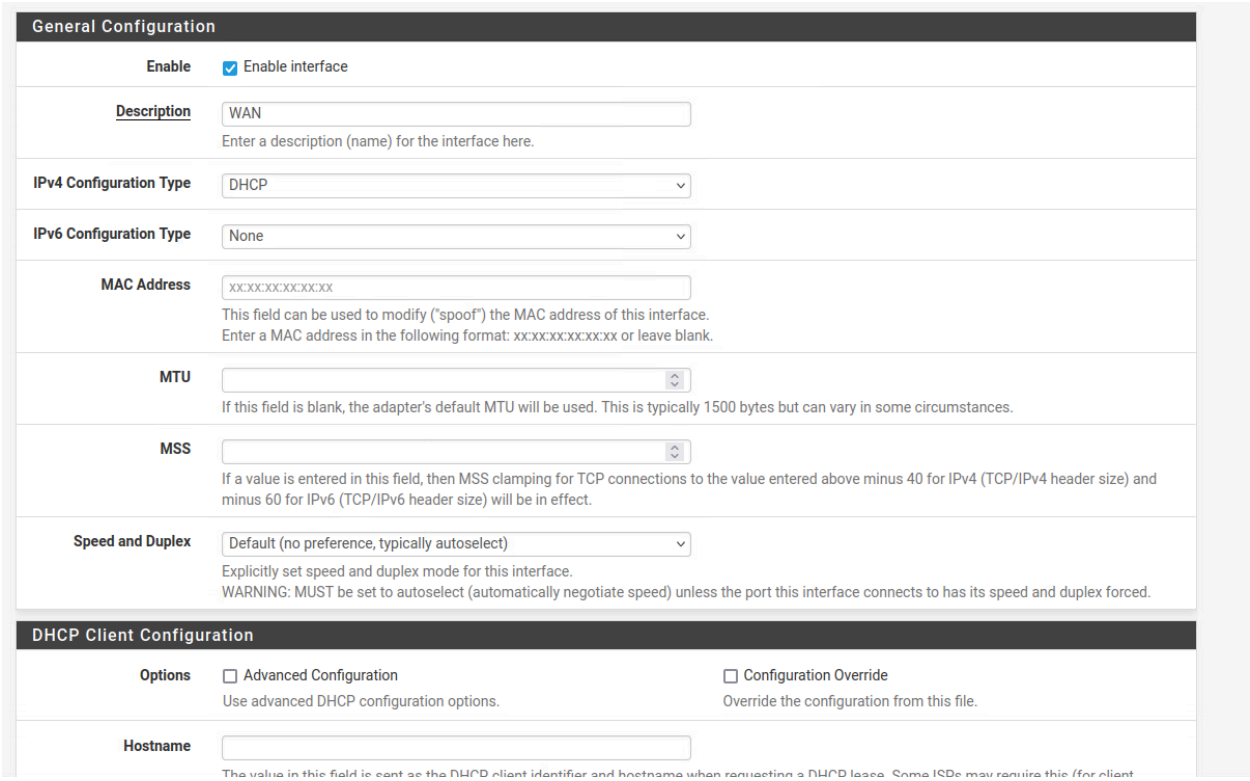


- This is the full setup for the DMZ Port group:

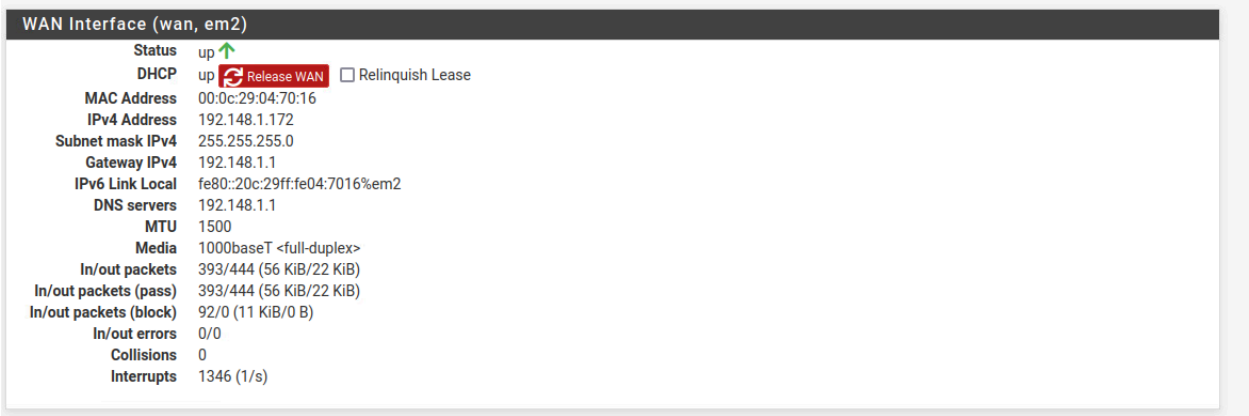
Now we test to see if we can access the pfSense web portal on a device in the main VLAN:



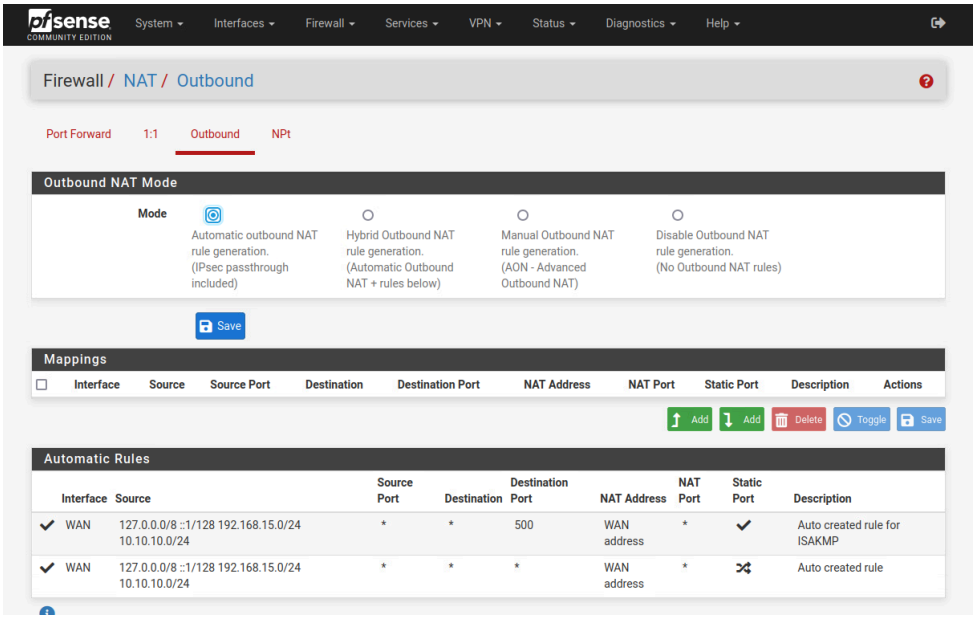
To setup internet (WAN) for LAN devices: Set the gateways for the devices to the firewall IP (192.168.15.1) and set DHCP in pfsense on the WAN Interface:



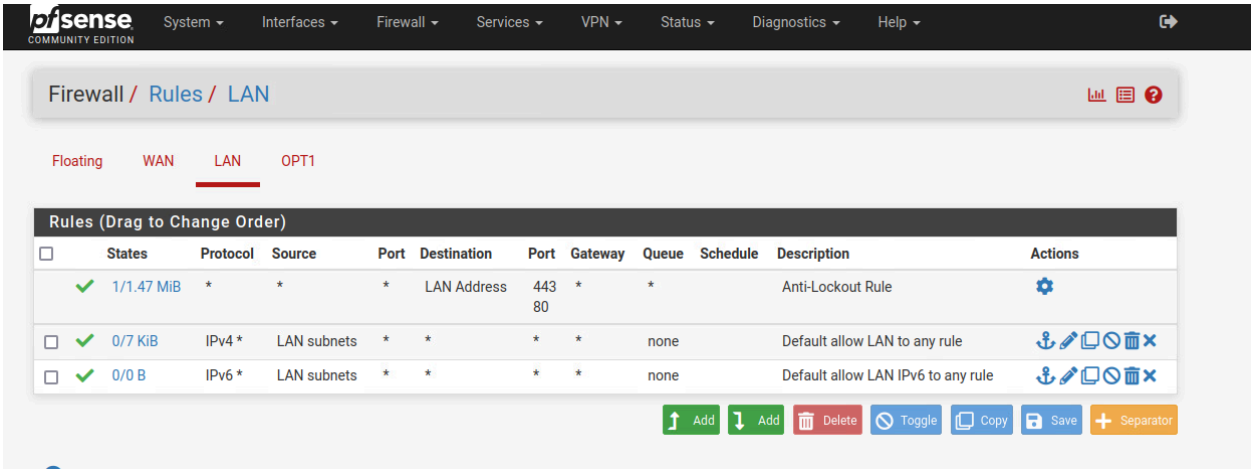
Check the status of the interface (make sure that it is up):



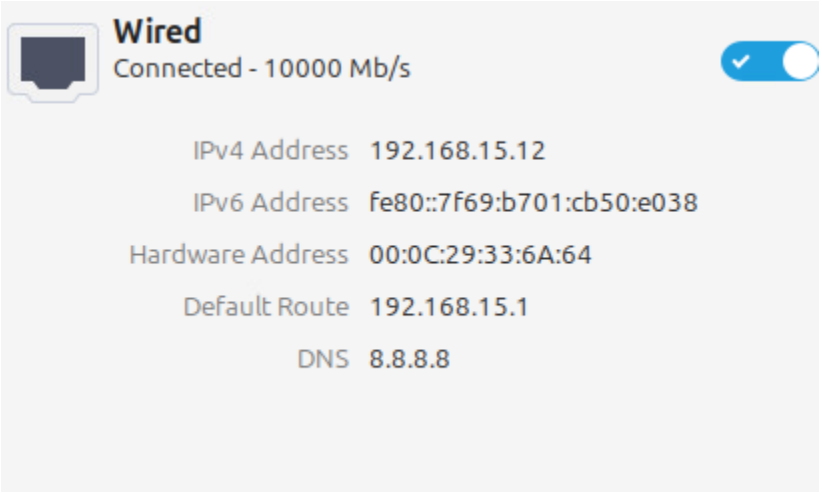
Then check Outbound NAT, since that is what will turn private IP's in public IP's:



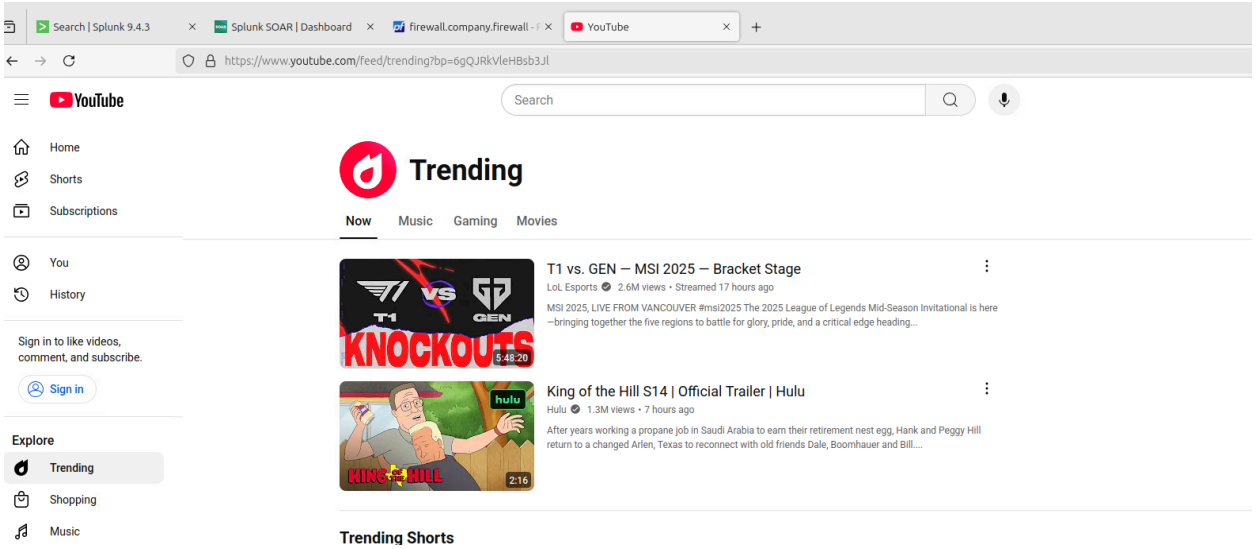
Then, make sure you have a firewall rule that lets you route traffic outbound:



Test on a machine within the LAN (this is the SOC computer):

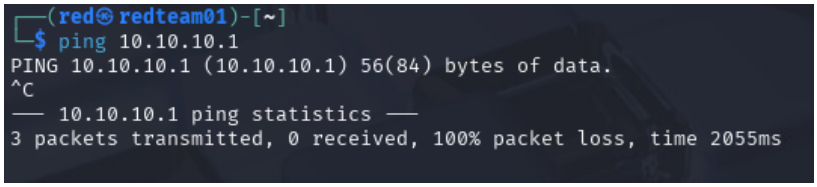


- Make sure that you change the Default Route (Gateway) to the Firewall: 192.168.15.1
- Also make sure that you change the DNS to an external DNS server like 8.8.8.8 so that you can resolve a name like youtube.com to an IP address

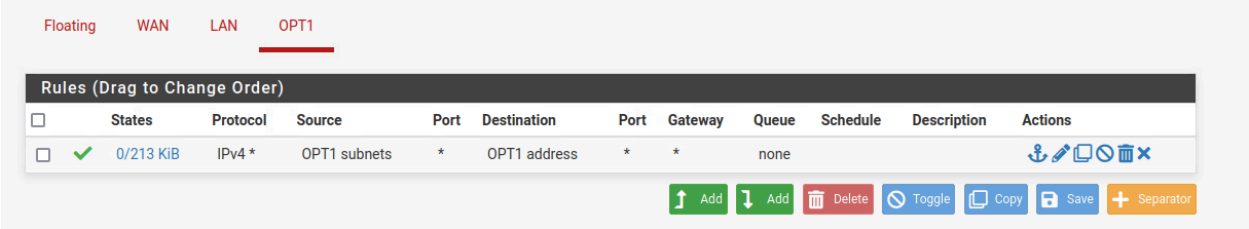


(The only reason why I added this was because the console pfsense made me force one of the LAN's to be a WAN, so I just decided to add an external internet to cover the WAN, and make OPTIONAL1 (OPT1) the DMZ LAN)

As of right now, we are not able to ping 10.10.10.1 (firewall) from the Kali machine in the subnet 10.10.10.0/24. Why?



Fix: It may be a firewall rule thing within pfsense, you need to allow inbound traffic

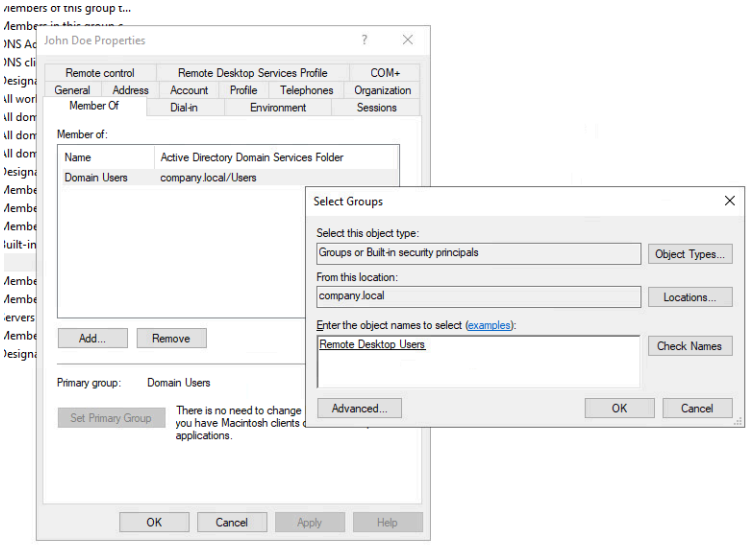


- You must allow ANY protocol (I put TCP, which still causes the machine to not be able to ping since ping uses protocol ICMP) from the OPT1 subnet to be able to communicate with the OPT1 address (10.10.10.1)
- **IMPORTANT TO NOTE:**
  - **FW RULES ONLY APPLY TO TRAFFIC *ENTERING* ON THAT SPECIFIC INTERFACE (e.g. OPT1, LAN)**
  - **So if you want two interfaces to interact with each other, you need to specify an inbound rule on one interface, and an outbound rule on the other interface**
    - e.g. Allow inbound traffic from src:OPT1 to dest:LAN within the LAN interface
      - You will also have to Allow **outbound** traffic from src:OPT1 subnets to dest:ANYTHING within the OPT1 interface
    - If you want to be able to ping across LANs, you need to make sure that all of the devices have the firewall IP address set as the default gateway.

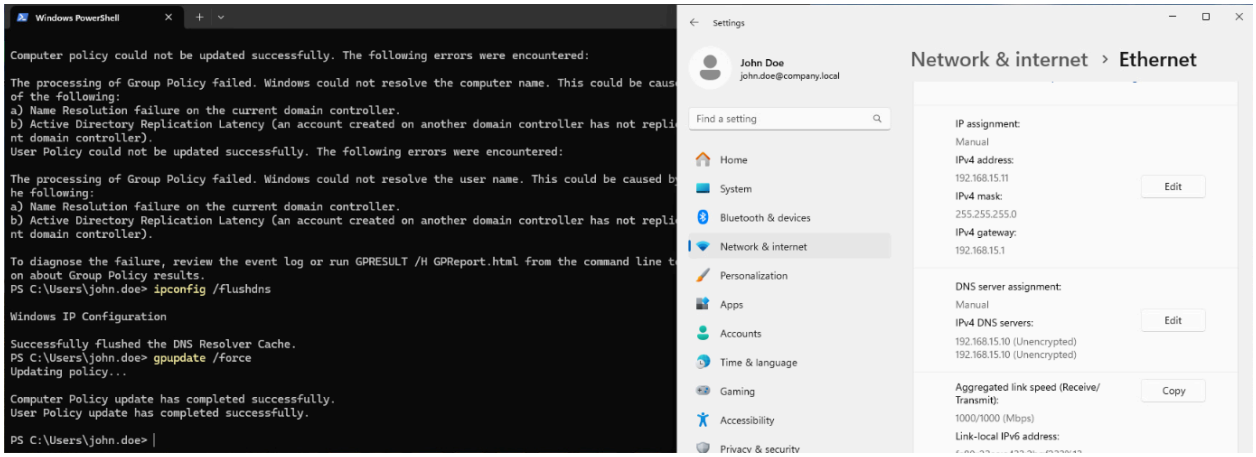
```
┌──(red㉿redteam01)-[~]
└─$ ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=0.502 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.571 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.329 ms
^C
── 10.10.10.1 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.329/0.467/0.571/0.101 ms
```

- Now it works

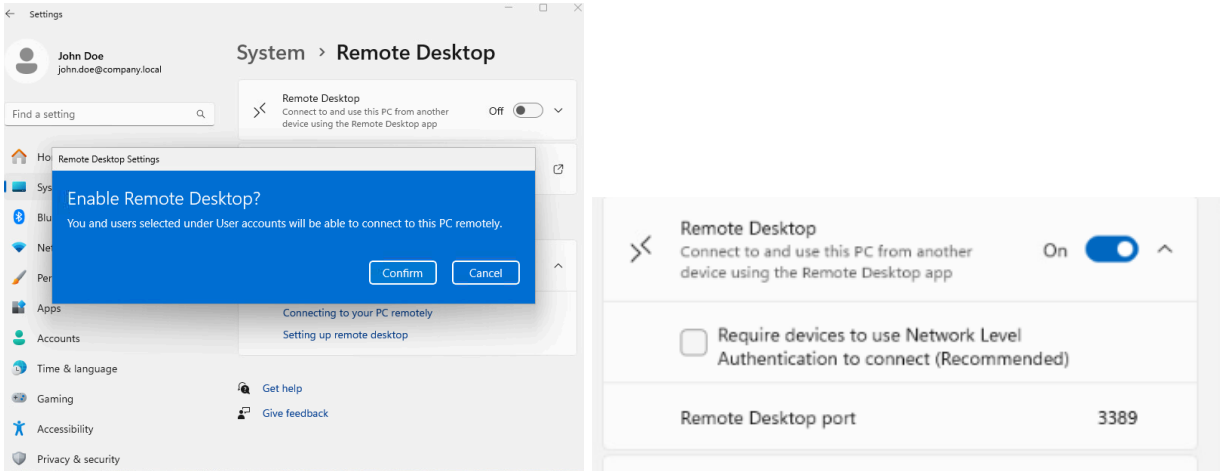
Troubleshooting RDP:



Make sure to do a gpupdate /force using the AD01 DNS (192.168.15.10) so that you don't get a Name Resolution Failure:



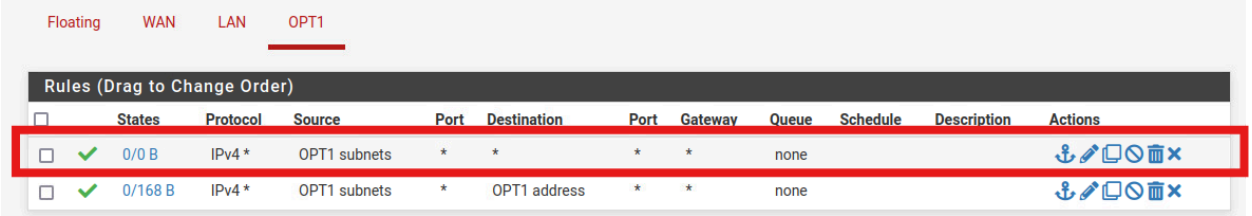
Do this on the client side:

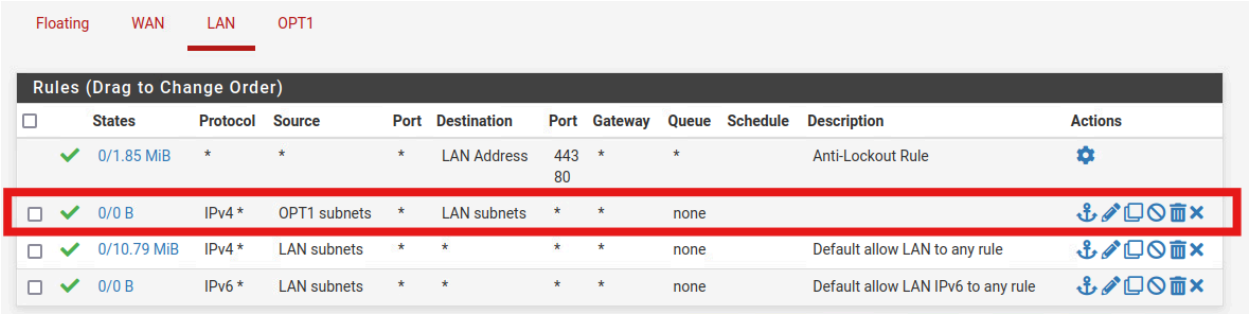


Make sure NLA is turned off; this allows for a less safe connection to happen which is good for the Kali attacker
Ignore that; just trying to generate some telemetry so we will just do RDP group policy locally:

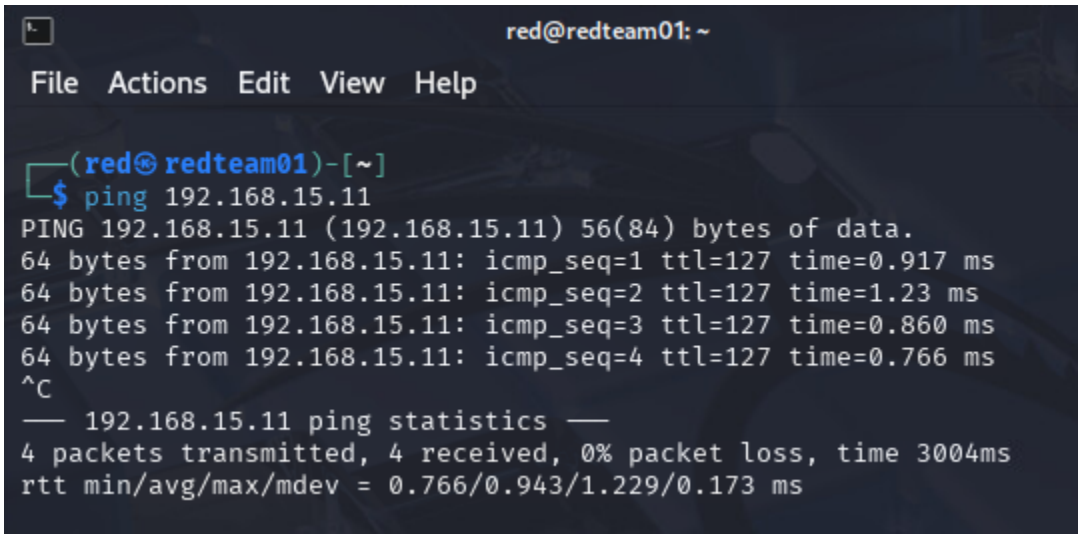
- net localgroup "Remote Desktop Users" "company\john.doe" /add
- net localgroup "Remote Desktop Users"
    - this is to check that john.doe is in
- secpol.msc
    - Local Policies > User Rights Assignment > Allow log on through Remote Desktop Services
    - Make sure “Remote Desktop Users” is listed
- gpupdate /force
- Restart-Service TermService -Force

HERE IS THE WEAK VULNERABILITY SCENARIO:
Let's say there's are devices in the DMZ subnet and you need them to connect to devices via RDP on the LAN subnet, so you decide to create the following **firewall rule**:



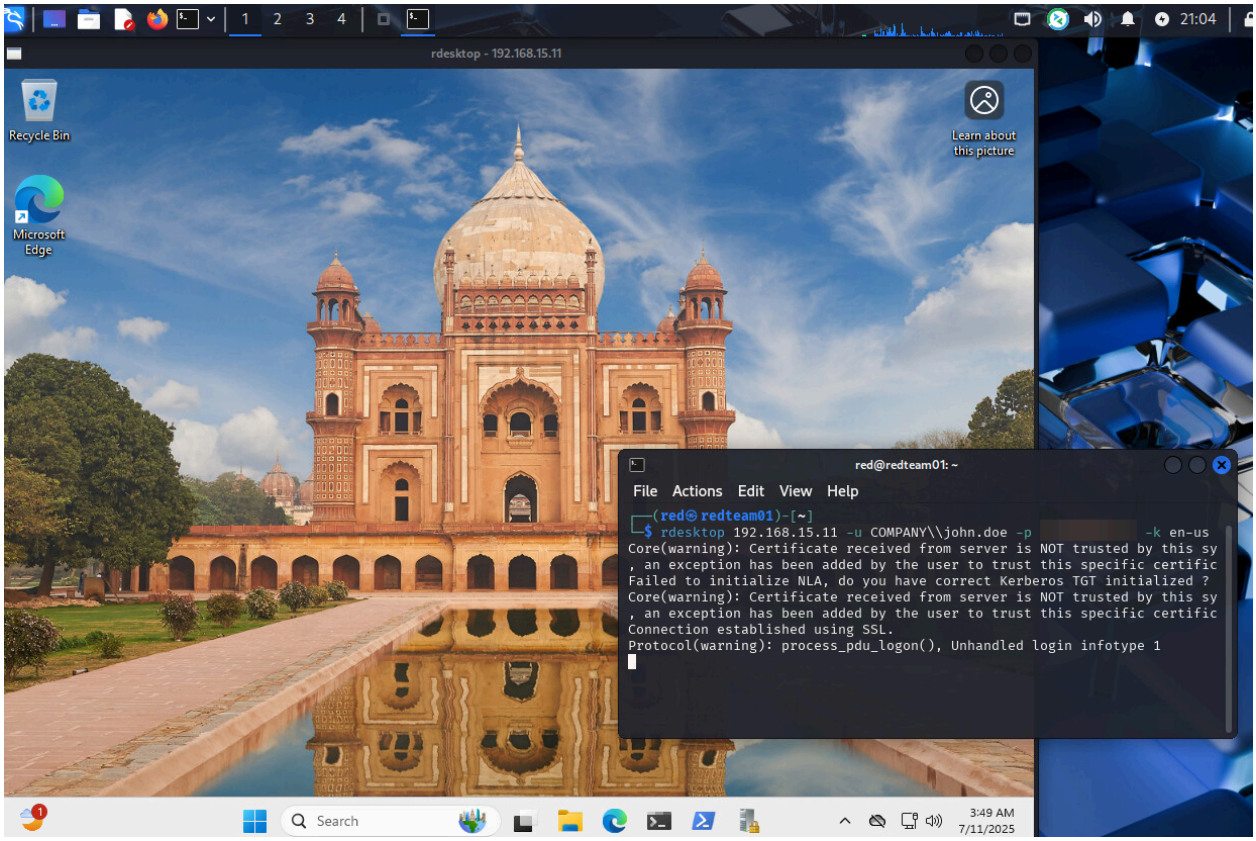


There's now an issue:
An attacker is in the DMZ and is able to ping the a user's computer within the COMPANY VLAN:



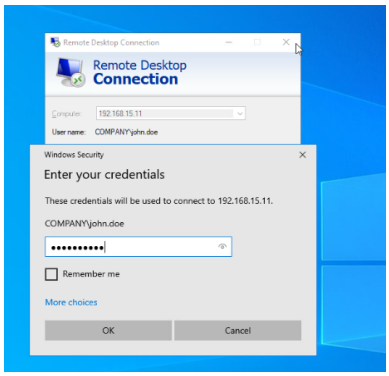
The attacker already knows the a domain user's username and password, and can now RDP into Win11Client using **rdesktop:**
(it wasn't working because I didn't have RDP turned on for both the user and the computer)

Now we can finally RDP from the Kali VM:



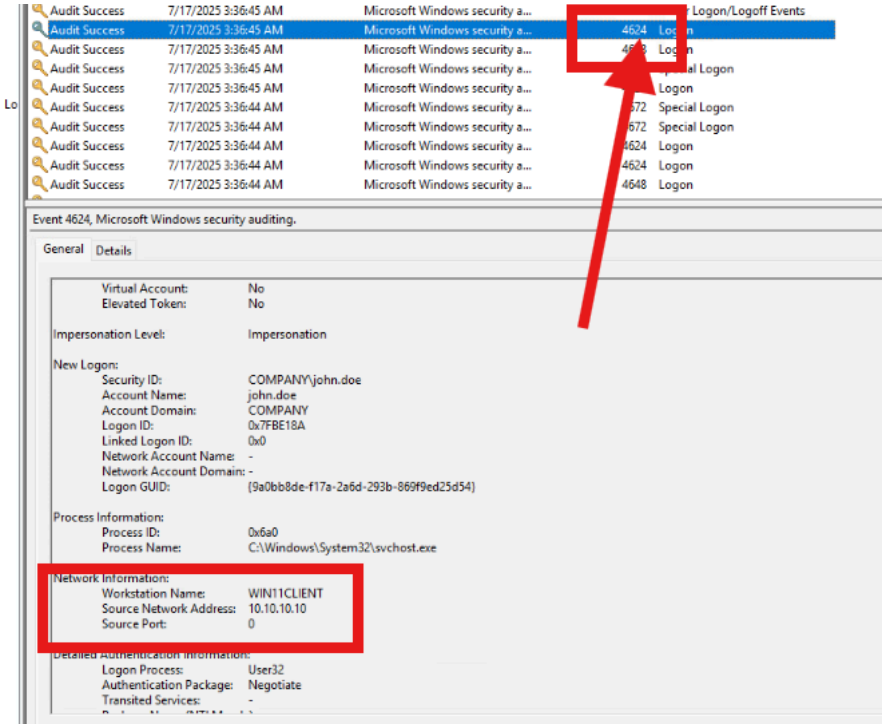
- You have to put single quotes around the double quotes if you have any in your password
    - ‘“example”’

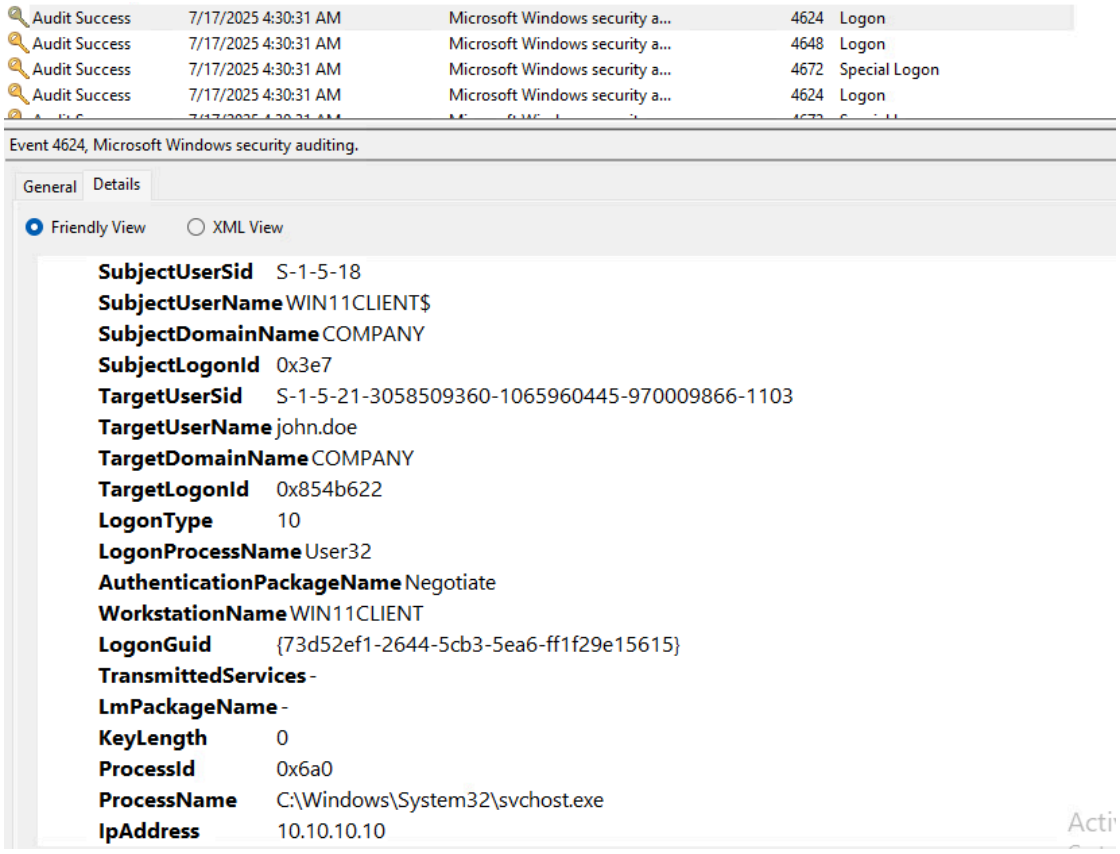
This should generate some telemetry on Splunk, but it doesn’t. Troubleshooting…:



Also Disabled Legacy (Basic) Audit Policy:

- gpedit.msc
- Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options
- Enable: Audit: Force audit policy subcategory settings (Windows Vista or later) to override audit policy category settings
- then gpupdate /force

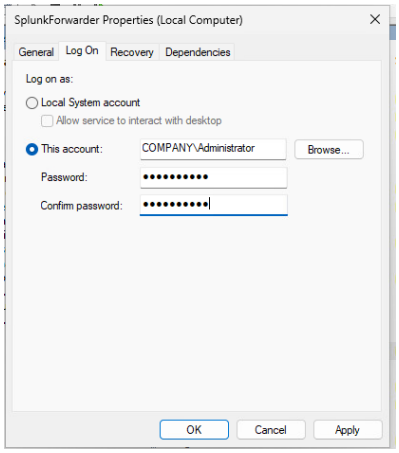


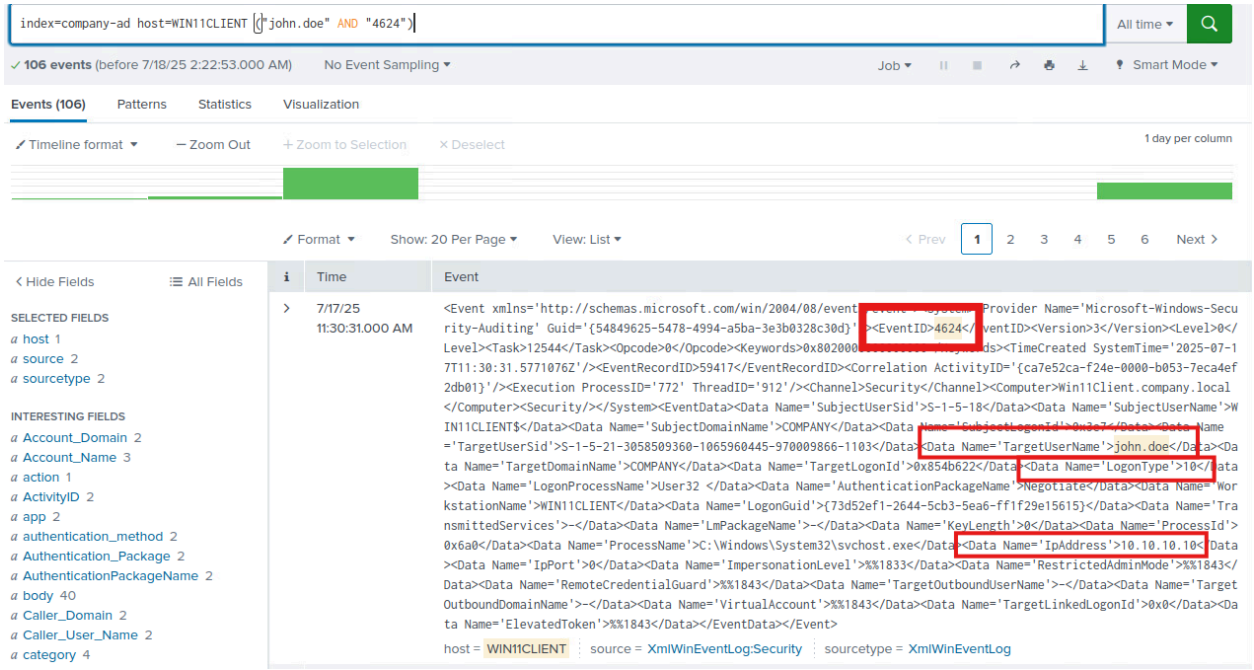


It has the Logon Type: 10

It is still not forwarding to Splunk. I think it's lacking permissions to even see it

- **You can create a splunk specific service account with specific permissions**
  - we will just run the service as the administrator account





We finally got it to forward to Splunk!

- We can see:
  - Event ID: 4624
  - Target User Name: john.doe
  - Logon Type: 10
  - IpAddress: 10.10.10.10 (the attacker's IP address)
- I tinkered around with so many settings that I forgot which one allowed this to be fixed. Here is the input.conf file from Win11Client's forwarder:

```
[WinEventLog://Security]
index = company-ad
disabled = false
renderXml = true
checkpointInterval = 5
start_from = oldest
```
i found these extra things online
Now I can clean up with XML output, generate this alert as a cron job in Splunk for me to test my automation response.

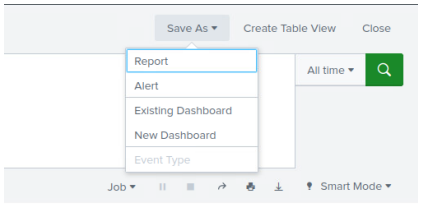
Cleaning up Splunk query
We can clean up the Splunk query using Regular Expressions (though I do not know how to):

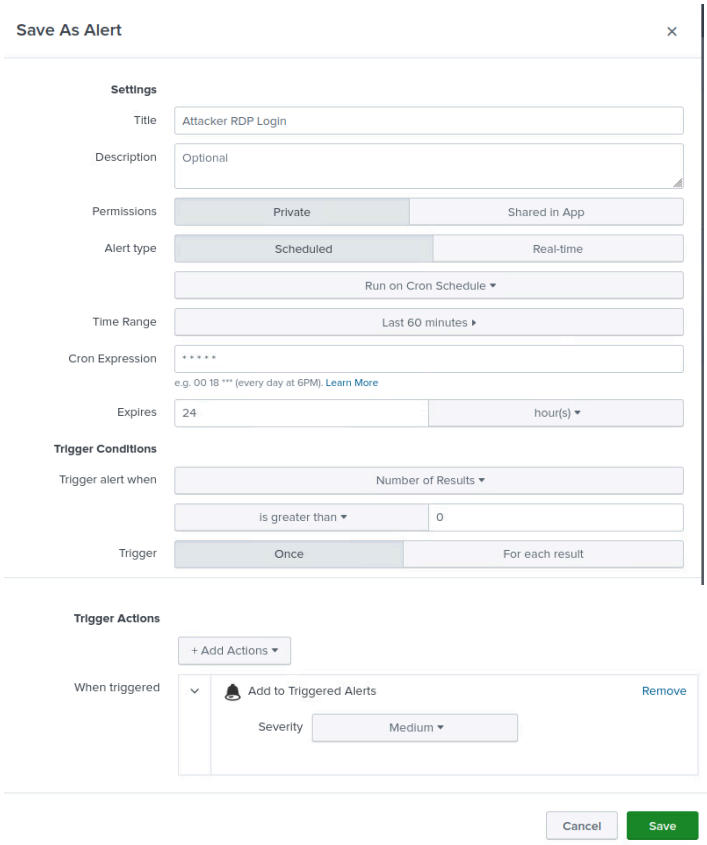
index=company-ad host=WIN11CLIENT EventCode=4624
| rex “<Data Name=\”TargetUserName\”>(?<TargetUserName>[^<]+)</Data>”
| rex “<Data Name=\”IpAddress\”>(?<IpAddress>[^<]+)</Data>”
| rex “<Data Name=\”LogonType\”>(?<LogonType>[^<]+)</Data>”
| search LogonType=10
| table _time, EventCode, TargetUserName, IpAddress, LogonType
| head 1

- head 1 will grab the top most / most recent event

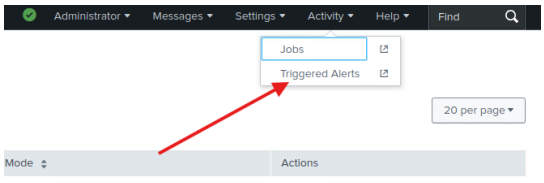
Line by line:
- rex “<Data Name=\”TargetUserName\”>(?<TargetUserName>[^<]+)</Data>”
  - rex: Splunk command that applies a regex to each event
  - Data Name=\”TargetUserName\”: Matches this literal part of the XML: “Data Name=TargetUserName”
    - <: Matches a literal opening angle bracket
    - Data Name= Matches the literal text *Data Name*= in the XML
    - \” : Escaped quote character; just like powershell
    - TargetUserName: Actual field name we want to target
    - \” : Closing quote
    - >: Matches the end of the opening tag
  - (?<TargetUserName>[^<]+)
    - (and): Defines the capture group (the thing we want to extract)
    - ?<TargetUserName>: This names the group “TargetUserName”;**THIS BECOMES YOUR SPLUNK FIELD**
    - [^<]+
      - [] : Defines a character set
      - ^: Inside a character set, it means “NOT”
      - <: We want to stop when we hit this (because it’s the next XML tag), basically the **delimiter**
      - [^<]: Match **any character except <**
      - +: Match one or more of those characters
  - </Data> : Matches the literal closing tag that ends the XML value

Save as alert:

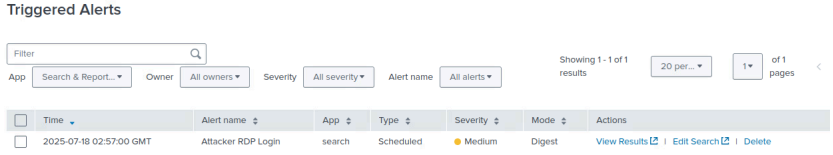




Run a cron job, scheduled to run every minute.
Can view it in:



Successfully triggered an Alert!!



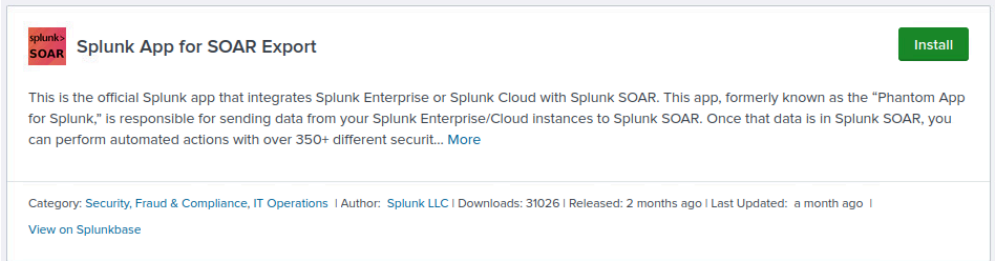
SPLUNK SOAR AUTOMATION

Plan for automation:

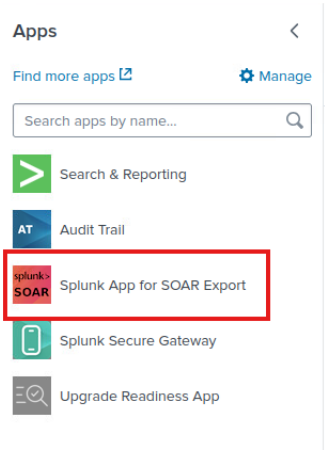
- Once an alert is triggered in Splunk, it should send an email with the Computer name/host, Username used to log in (TargetUserName), IpAddress (IP address of the remote computer), Event ID, Event code
- Once the email is sent, it will ask:

  "Would you like to disable the account: *TargetUserName***?"**
- in which the SOC analyst can pick Yes or No in SOAR
- This will then get sent back to the SOAR playbook and then disable the Active Directory account for that user.

Apps -> Find more apps -> **Splunk App for SOAR Export**

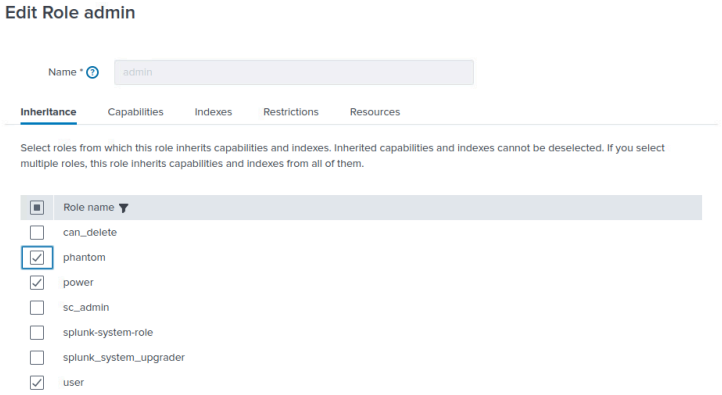


- Need this one because it **sends alerts and results** from **Splunk Enterprise TO Splunk SOAR**

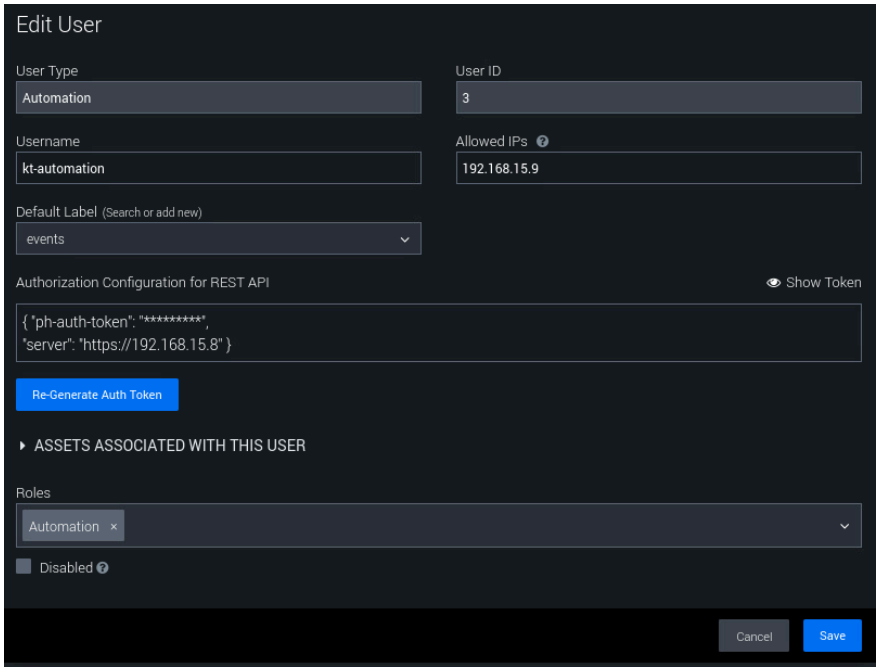


Follow the https://docs.splunk.com/Documentation/SOARExport/latest/UserGuide/ConfigureSOARserver **t**o connect SOAR App for SOAR Export to Splunk SOAR with the appropriate users. Just listing a few steps here:

Setting inheritance for **admin** to include **phantom** capabilities:

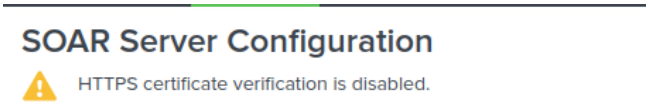


Creating a new automation user in Splunk SOAR w/ the IP address of the Splunk instance, and grabbing the REST API key:

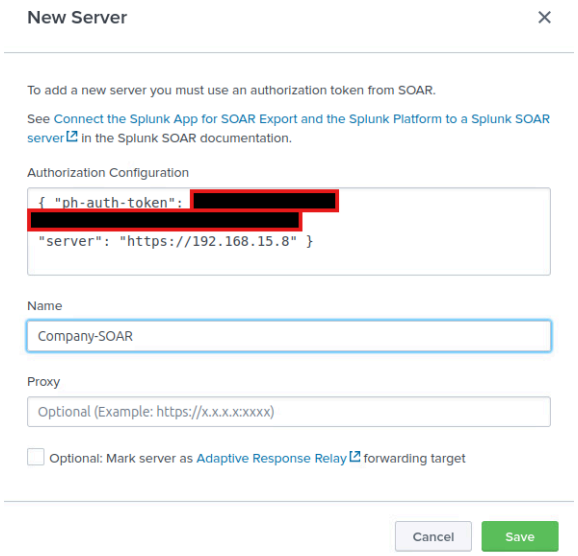


Kept getting an error about IP address not being correct on the SOAR side.
[Intro to Splunk SOAR Phantom | Sending Splunk Alerts To SOAR Phantom]
Followed this video; **TLDR: go to**

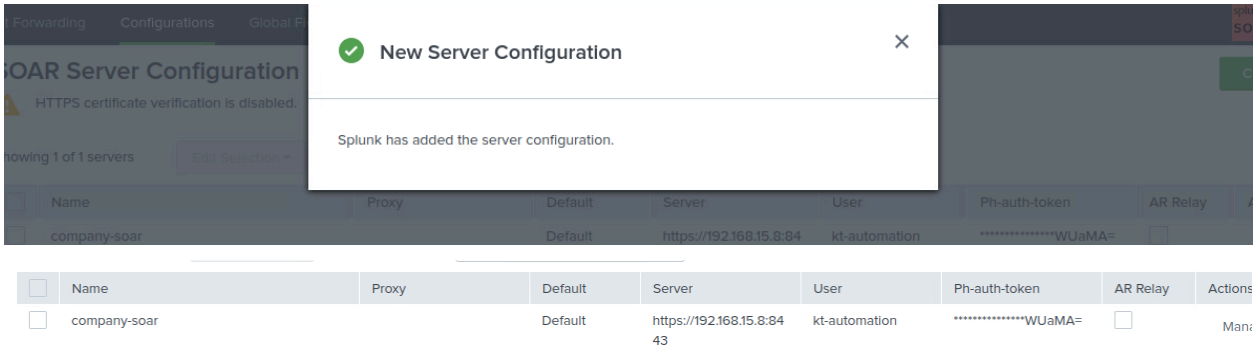
- cd /opt/splunk/etc/apps/phantom/local/phantom.conf AND
- cd /opt/splunk/etc/apps/splunk_app_soar/local/soar.conf
    - Type in the following:
      [verify_certs]
      value = false
- After restarting Splunk server, it should result in this:



In Splunk App for SOAR Export, create a new server:

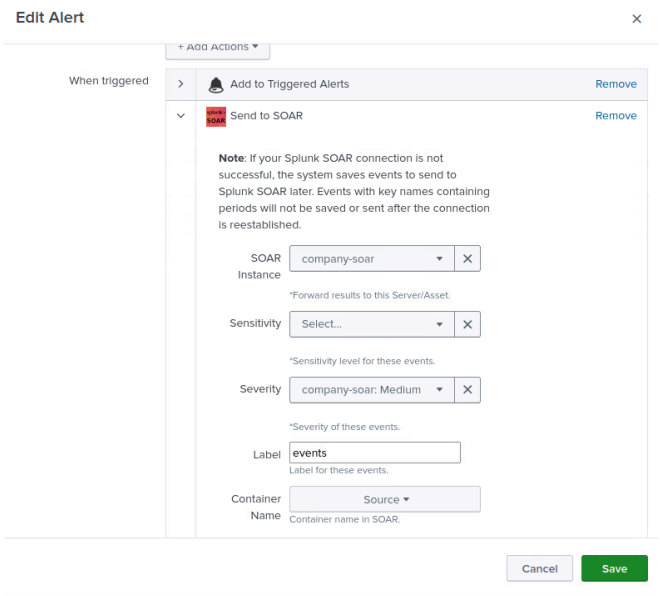


Now you can add the thing:

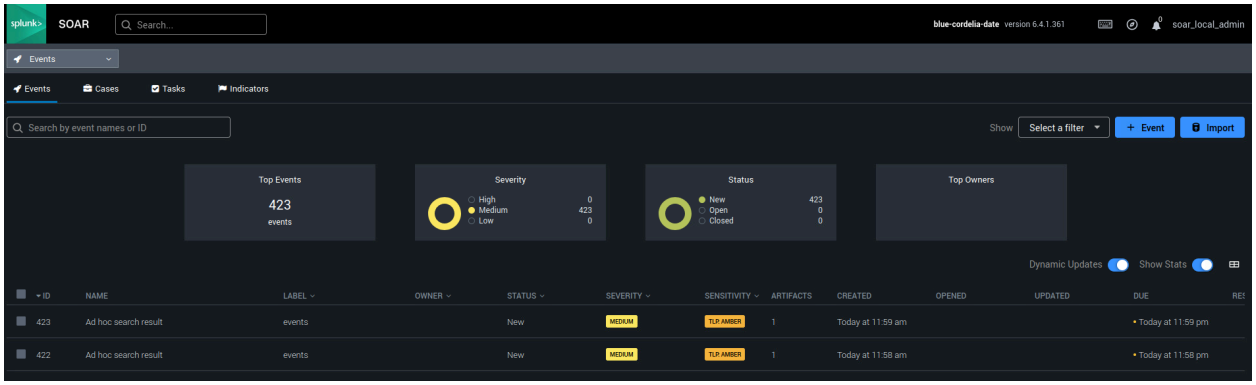


Name	Proxy	Default	Server	User	Ph-auth-token	AR Relay	Actions
company-soar		Default	https://192.168.15.8:8443	kt-automation	••••••••••••••••WUaMA=		Mana

I'm setting up the alert to send to SOAR:



This is what shows up in my SOAR: (I left it running overnight)

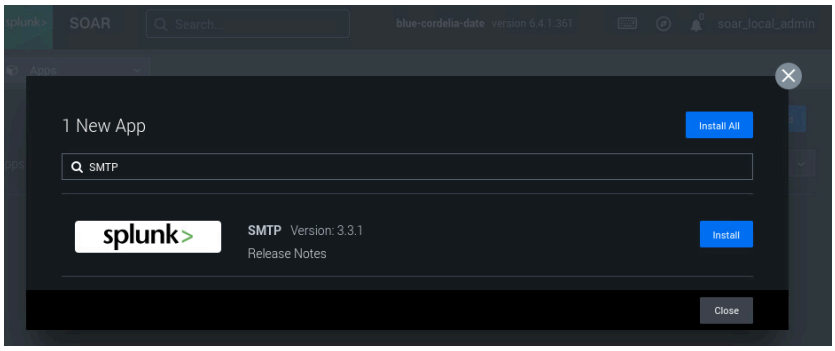


Now to create a playbook:

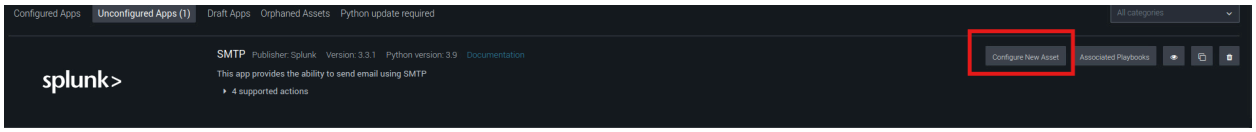
We want it to:

- Receive an alert
- Have user asking submit a YES/NO response on SOAR
- Disable the specified user account in AD based on that response

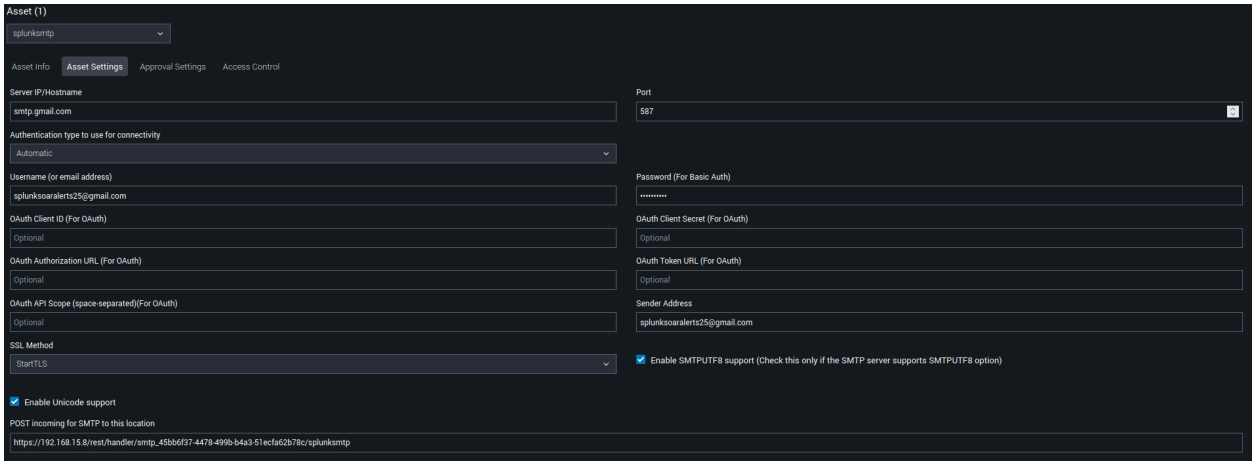
First: We need to set up an email connector (SMTP) **Make sure DNS on RHEL 8 is set to 8.8.8.8 or something that can reach external sites**



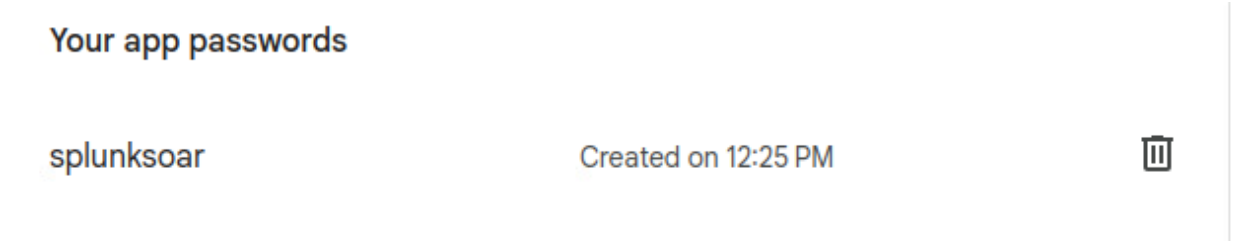
Go to Apps -> New App -> Search up “SMTP” and install it



Now you can configure the asset and include all of the things it needs



Use google voice for 2 Factor and set up a new app password using https://myaccount.google.com/apppasswords
and use the key/password that is created in your SMTP Asset settings where it says “Password (For Basic Auth)



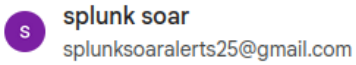
Create a gmail account and use Gmail's SMTP server to send the email

When using the Gmail SMTP server:

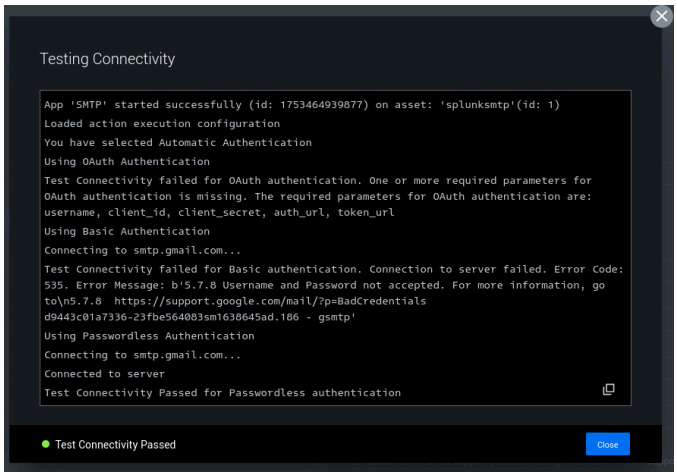
- The sending limit is 2,000 messages per day. Learn more about email sending limits.
- Spam filters might reject or filter suspicious messages.
- The fully qualified domain name of the SMTP service is smtp.gmail.com.
- Configuration options include:
  - Port 25, 465, or 587
  - SSL and TLS protocols
  - Dynamic IP addresses

- Using port 587 and hostname smtp.gmail.com

Splunk SOAR Alert email address:



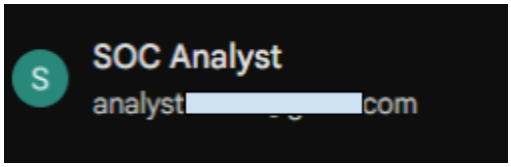
Test the connectivity to the SMTP server:



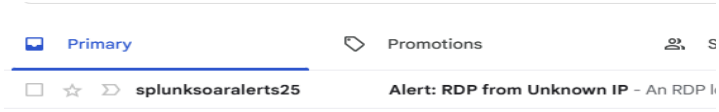
Now we need to build the playbook

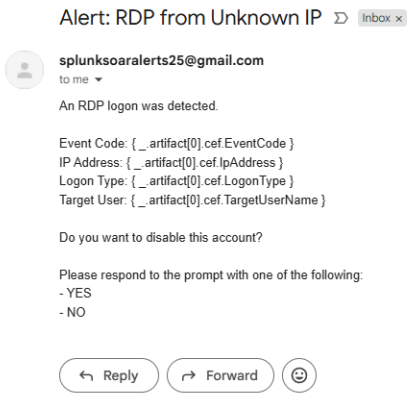
- Extracting username from alert
- Send an email to analyst
- Wait for YES/NO response
  - If YES, it disables the account using AD

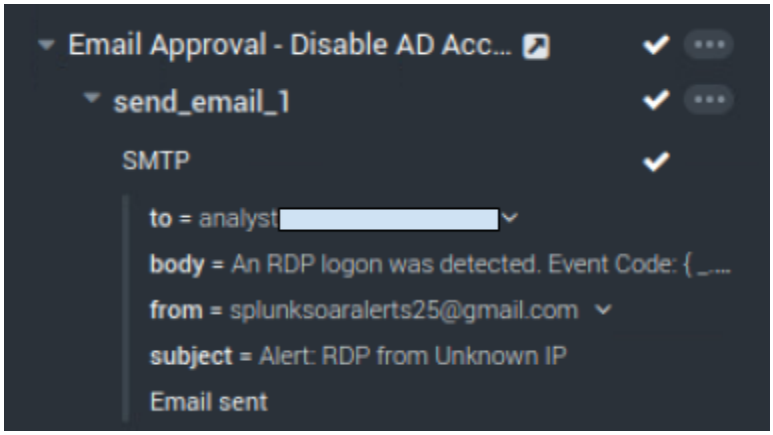
SOC Analyst email address:



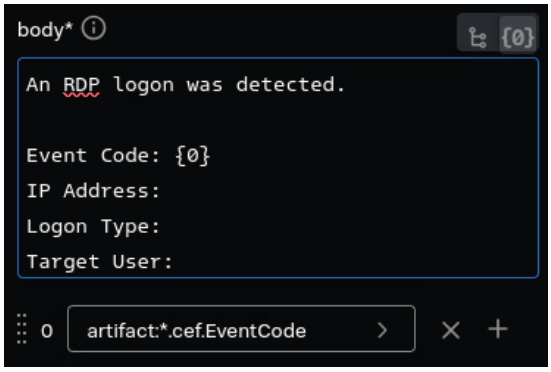
Email has finally been received on the analyst-side:

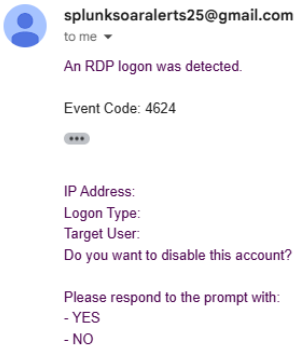






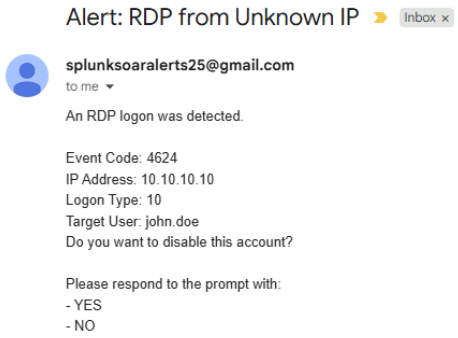
Now to fix the actual contents of the email.



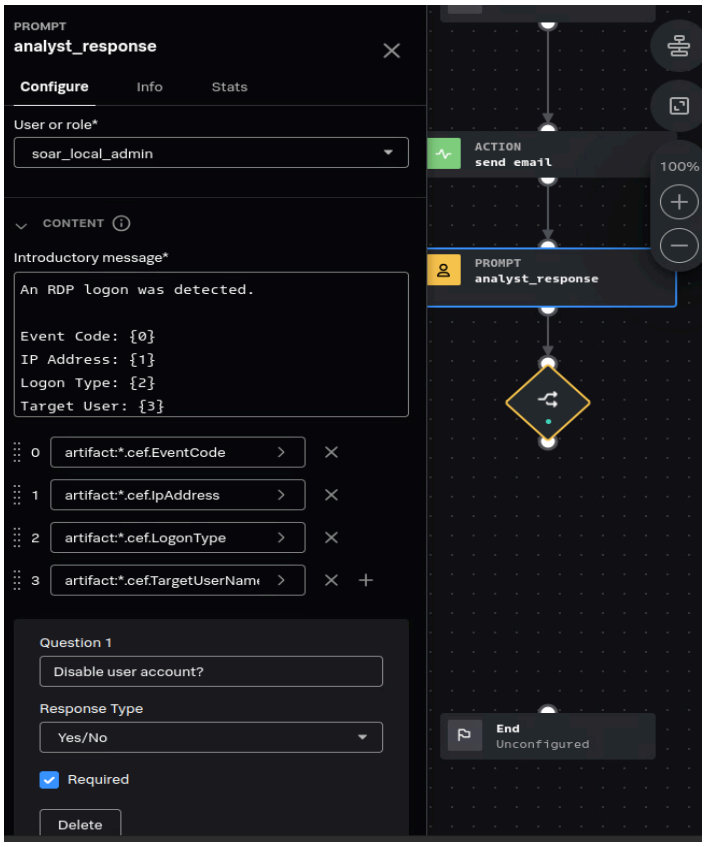


What works: **artifact:*.cef.EventCode**

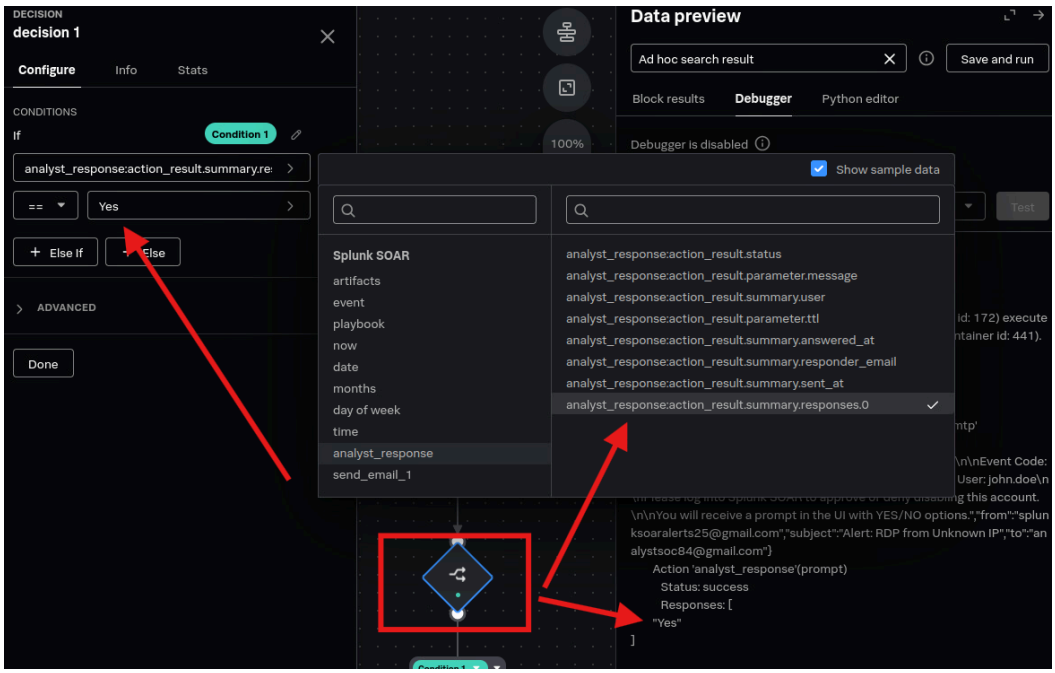
- You can just type in the name of the variable, even if it doesn't exist within the list of .cef



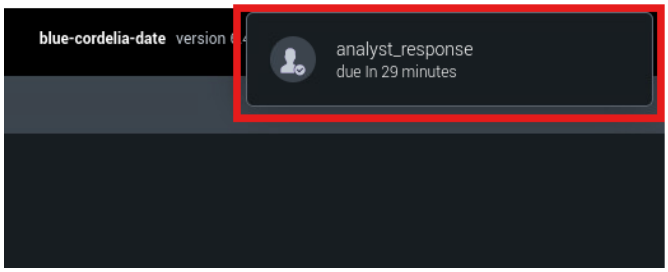
Now we need to create a prompt for the analyst to respond to (Yes/No) and then push the output into a Decision Block:

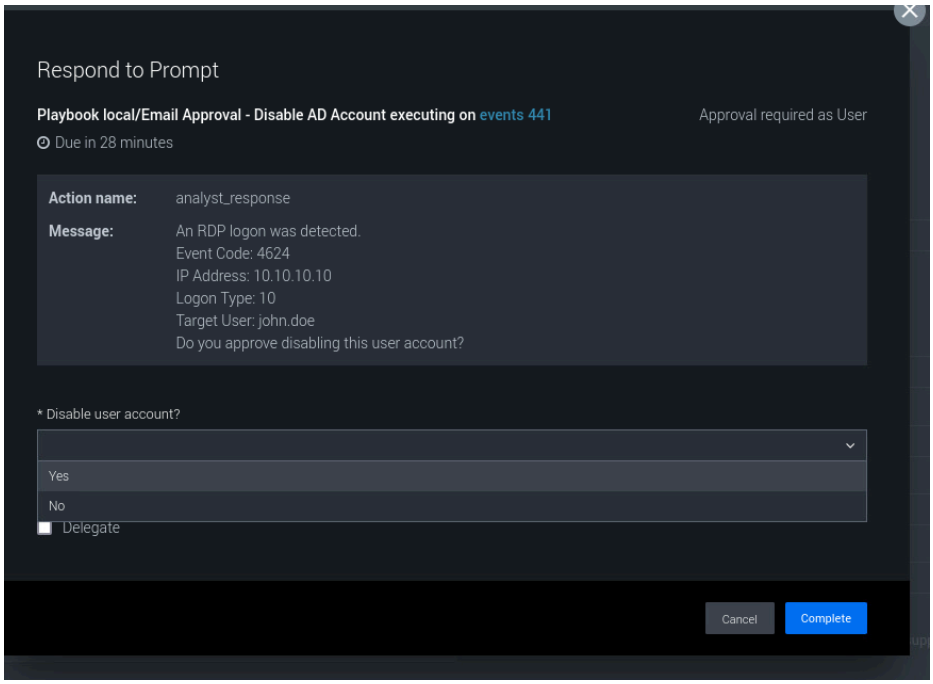


Decision Block: we choose analyst_response:action_result.summary.responses.0 because it is the first index in which the response is saved in. (look at the logs)



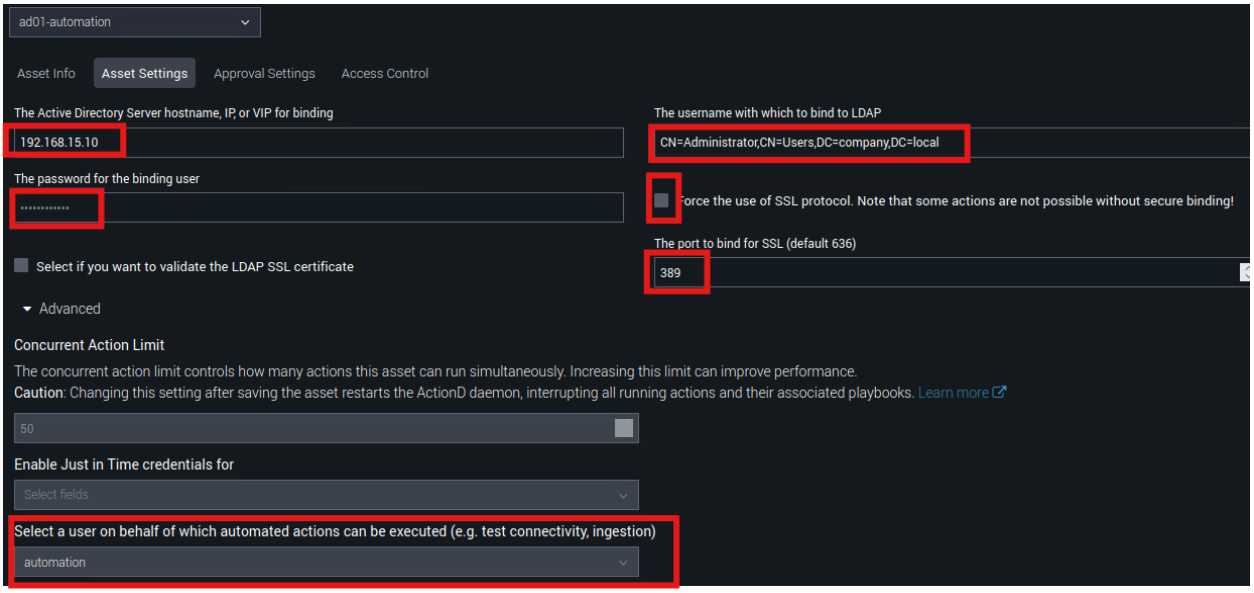
When running, this will show up in Splunk SOAR:



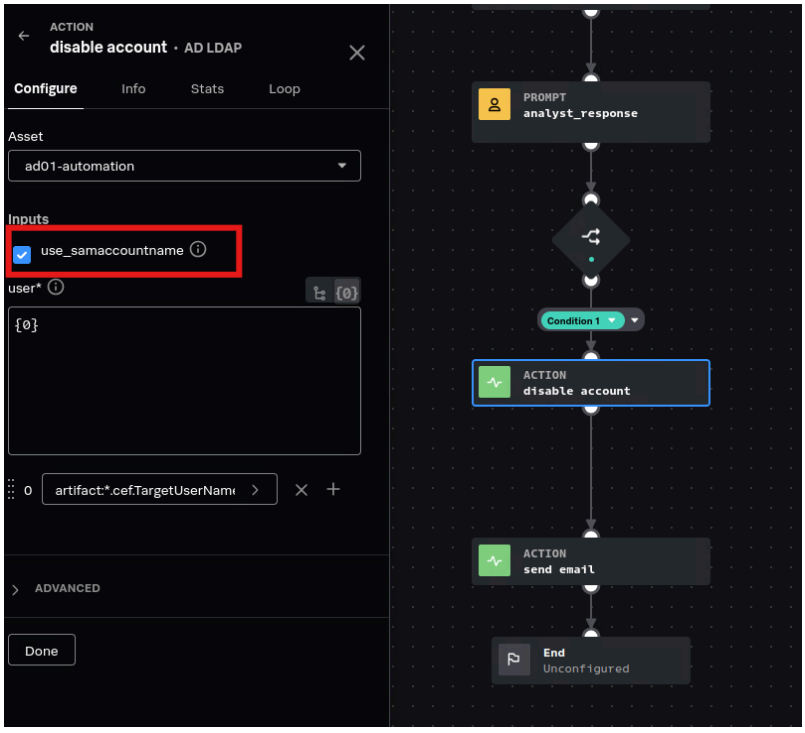


Now we want to take the TargetUserName and disable it in AD
Get the AD LDAP App in SOAR

Connect to AD01 using Administrator (ideally you'd have a separate account for this that handles modifying of users but since this is a homelab environment, we'll just use Administrator since it has all the permissions required)

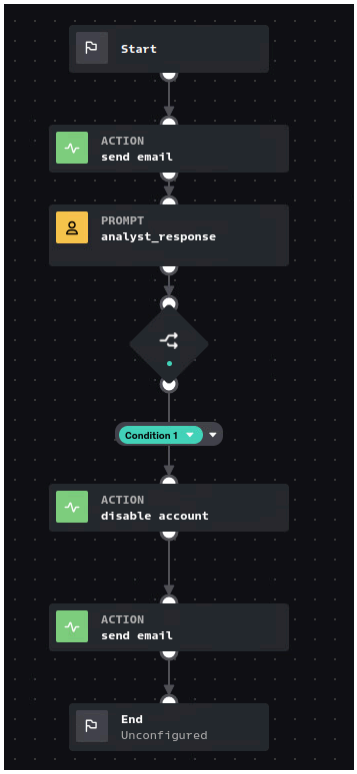


Now we add a new AD Action Block where we "disable account" using **artifact:*.cef.TargetUserName:**



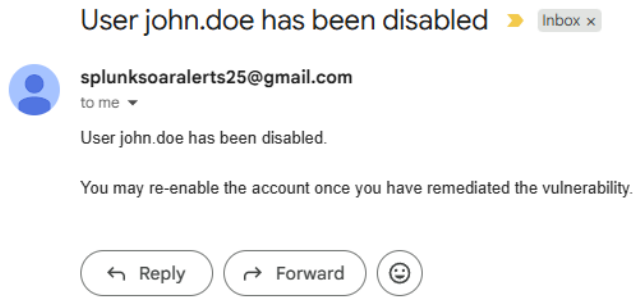
You can either use the sAMAccountName (john.doe) or the more production-ready way would be to use the DistinguishedName (CN=John Doe,CN=Users,DC=company,DC=local)

- You can get this buy putting in a "get user" block and pass it into the disable account block

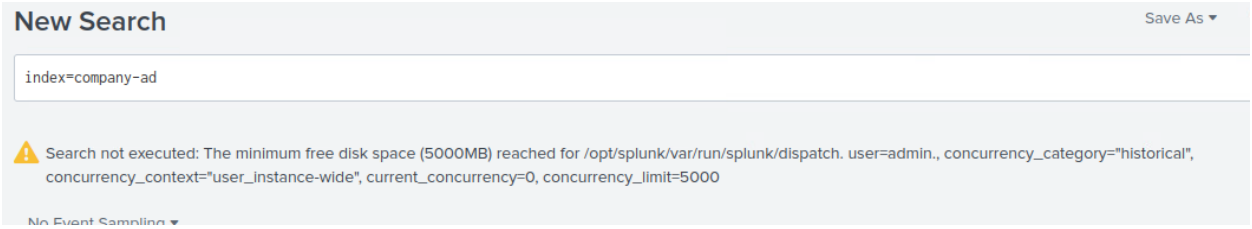


Now when we look in AD01, we should see that account John Doe has been disabled

An email should be sent to the user like this:

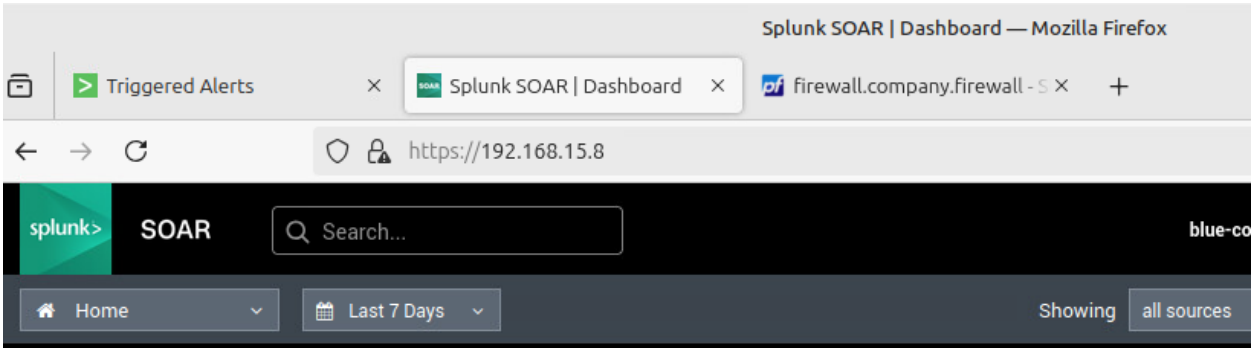


Misc. Troubleshooting:



Getting this error when trying to query in Splunk

- Go to settings -> Server Settings -> General Settings -> Change the “Pause indexing if free disk space (in MB) falls below” to something low
- Restart splunk: /opt/splunk/bin sudo ./splunk restart



I’m viewing the Splunk logs, Splunk SOAR and pfsense within my SOC Analyst Mint VM