APRIL 16, 2025

HOME LAB BUILD

VIRTUAL NETWORK DEFENSE AND MONITORING PROJECT

ANTHONY CAMPBELL
University of Texas at San Antonio

Contents

Introduction

I am an overthinker and have been told I sometimes 'overdo' some things when it comes to trying to think about projects or my write-ups for lab work. I've been telling myself over the last year that I needed to dive into some home projects, and so I hope that my proposals can be refined down to something achievable within my entry level skillset. I have, like many students, a great fear that I will graduate and be met with no sort of opportunities after graduation. I do know that I have to be proactive in gaining experience to deserve those opportunities. I have in the last few months acquired and begun to set things up to enable endeavors such as this, and I look forward to your guidance in this project.

Devices

I have two desktops and a MacBook Pro. The MacBook Pro (2017 Touchbar) I pretty much only casually browse on but that won't be used in this Home lab project.

Home Lab Device

I purchased a refurbished Lenovo ThinkCentre M710q Desktop from Best Buy before the semester started. I connect to it via Windows Remote Desktop Connection tool often. So far, I have used it to run a simple web server for a long running joke among friends, as well as a gaming server for a time or two. I had VirtualBox installed on it before but will be installing VMware Workstation Pro soon.

OS: Windows 10 Pro

Processor: Intel i7-6700T ~2.8GHz

Memory: 32GB (2 x 16GB) (Upgraded from 16GB to 32GB early February)

GPU: Intel HD Graphics 530

Milestones

1: Setup of pfSense firewall

Somewhat straightforward, using pfSense to set up a firewall and prepare network segmentation. I've set up pfSense once or twice within other lab environments, so this will be more experience. I haven't segmented any LANS or VLANS yet so that will be a great experience too. Will likely set up Kali machine to set up pfSense as much as possible without the other VMS created and spun yet.

2: Setup of Snort

I have no experience with Snort, this will be fun. This will include setting up an Ubuntu VM to run Snort. I will also set up some local rules.

3: Win 2019 Server & pfSense Rules revisions

Setting up a Domain Controller for a couple other machines via a Windows 2019 Server (evaluation edition) and setting up some Active Directory Domain Services. Setup of an additional Win 10 VM (likely two) and connecting it to the Active Directory.

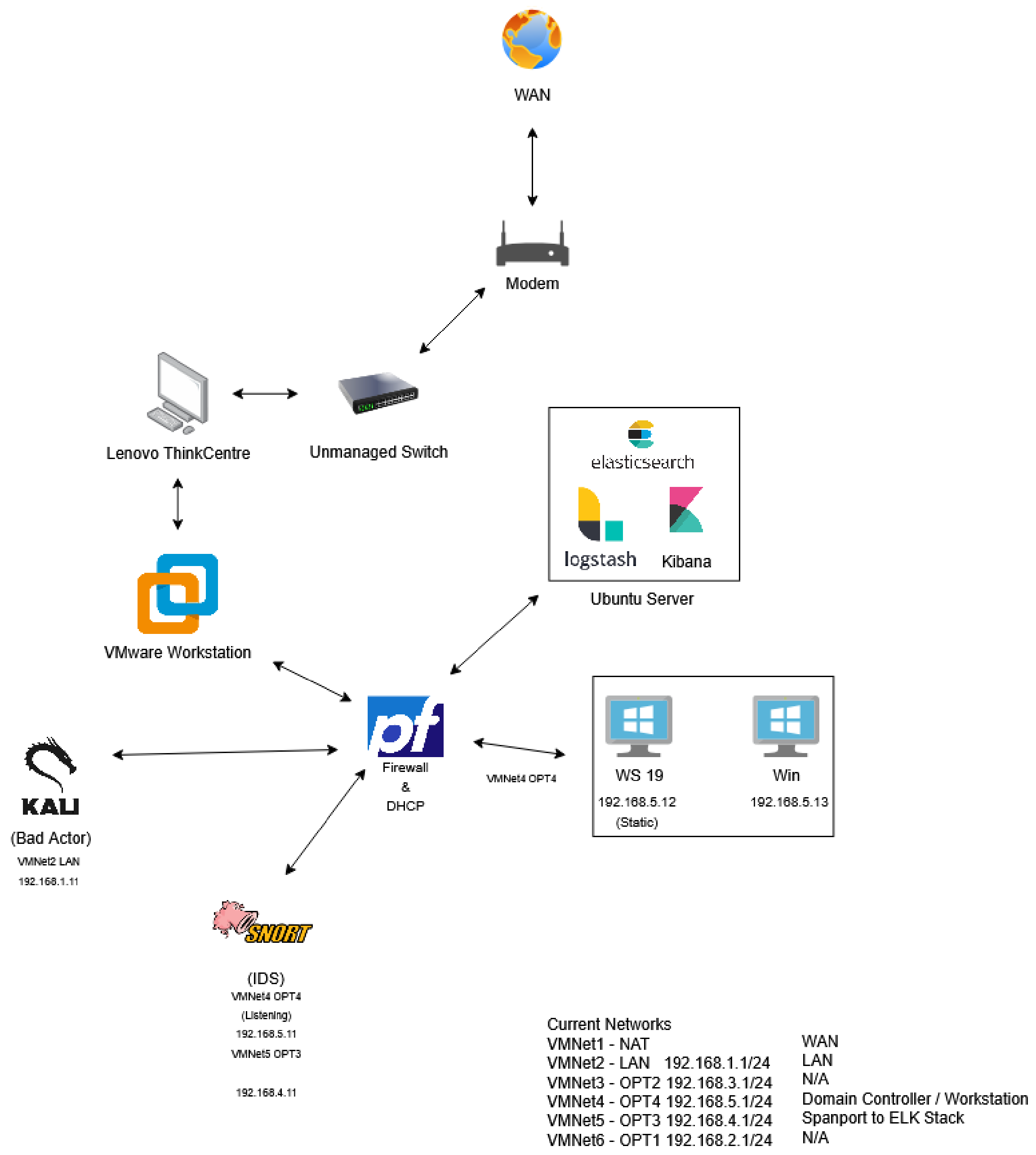
4: Installation of ELK stack on an Ubuntu Server, universal forwarder on windows server

Ubuntu Server setup, ELK installation and setup as well as Universal Forwarder on the Win 2019 server. Configuring logs to be forwarded to Splunk, and then I (hope) that will be essentially it.

5: Further configuration of ELK stack, Bad Actor Probe test [In Progress]

ELK stack configuration is no joke, and on pause with the end of my current semester. When I get back to this project I look forward to troubleshooting and expanding my experience with ELK stack, as well as testing and using it properly beyond installation and some initial installations.

Rough Diagram of Network



Current Networks

VMNet1 - NAT	WAN
VMNet2 - LAN 192.168.1.1/24	LAN
VMNet3 - OPT2 192.168.3.1/24	N/A
VMNet4 - OPT4 192.168.5.1/24	Domain Controller / Workstation
VMNet5 - OPT3 192.168.4.1/24	Spanport to ELK Stack
VMNet6 - OPT1 192.168.2.1/24	N/A

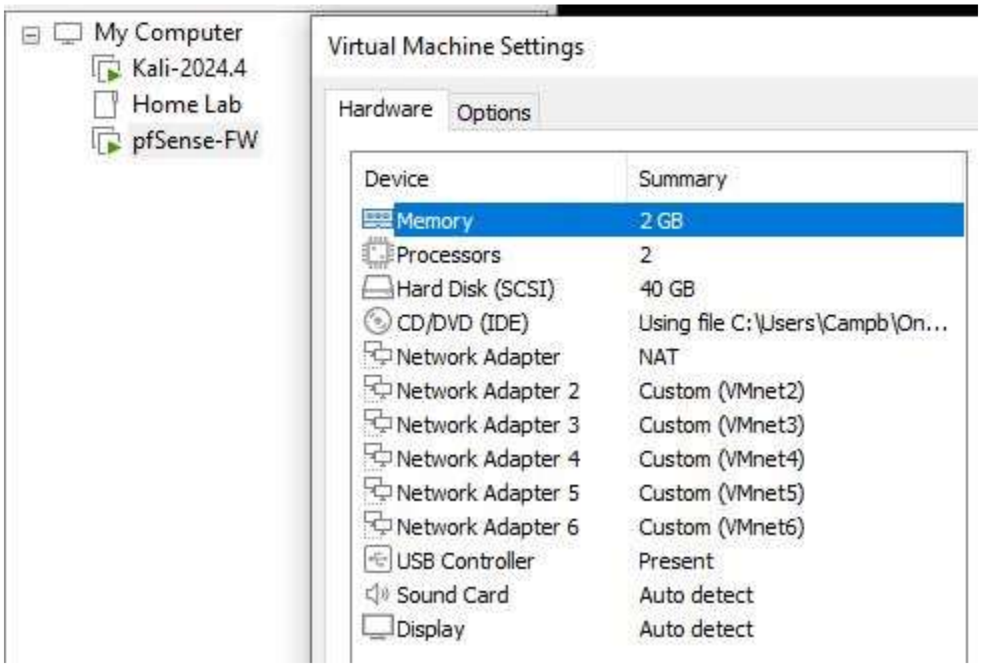
Figure 1: Updated with Milestone 3.

Milestone 1 Report

pfSense Initial Configuration

This initial setup was easy and straightforward after getting over an initial provisioning bump. Though my prior experience with pfSense involved installing it onto an Ubuntu server VM I debated setting it up via the Netgate installer as that would be the go-to for a bare metal install (though via USB).

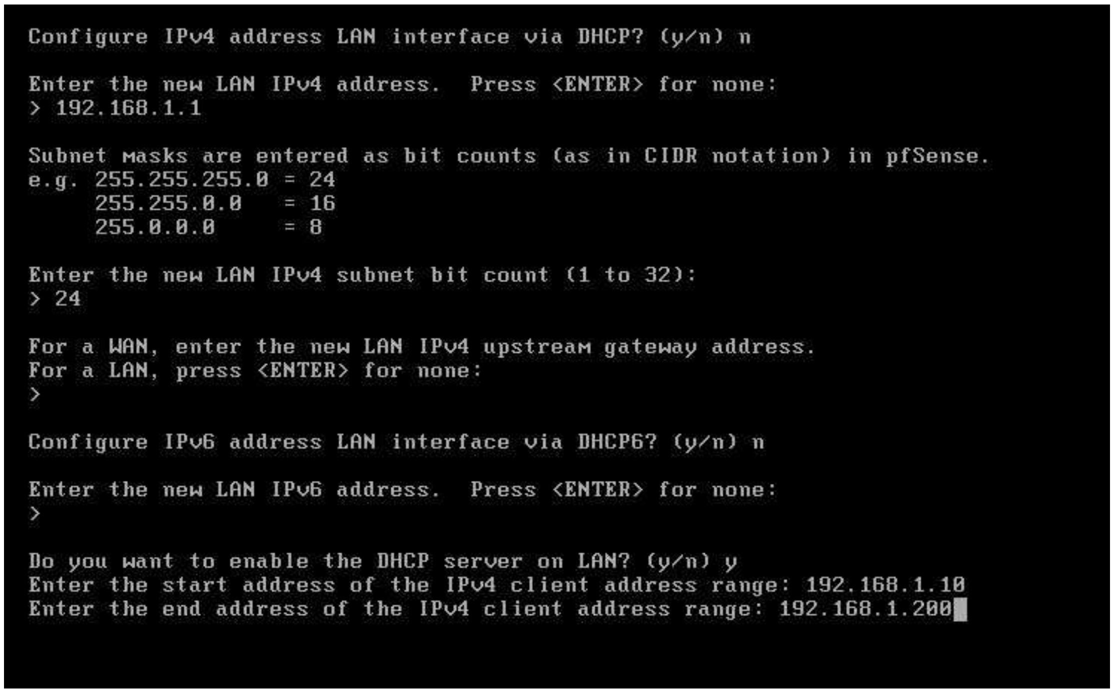
The first VM I tried to set up using the Netgate Installer method booted incorrectly, and it ended up being because I was operating on the “20 GB free for pfSense” logic. When I scrapped it and allotted 40 GB of storage for it, it booted perfectly fine, and I was able to begin configuration of pfSense. I went into the hardware settings and added custom VMnet segments to set up soon after. I know there is an actual LAN segment option, but from some reading most people seemed to recommend using the VMnets.



Device	Summary
Memory	2 GB
Processors	2
Hard Disk (SCSI)	40 GB
CD/DVD (IDE)	Using file C:\Users\Campb\On...
Network Adapter	NAT
Network Adapter 2	Custom (VMnet2)
Network Adapter 3	Custom (VMnet3)
Network Adapter 4	Custom (VMnet4)
Network Adapter 5	Custom (VMnet5)
Network Adapter 6	Custom (VMnet6)
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

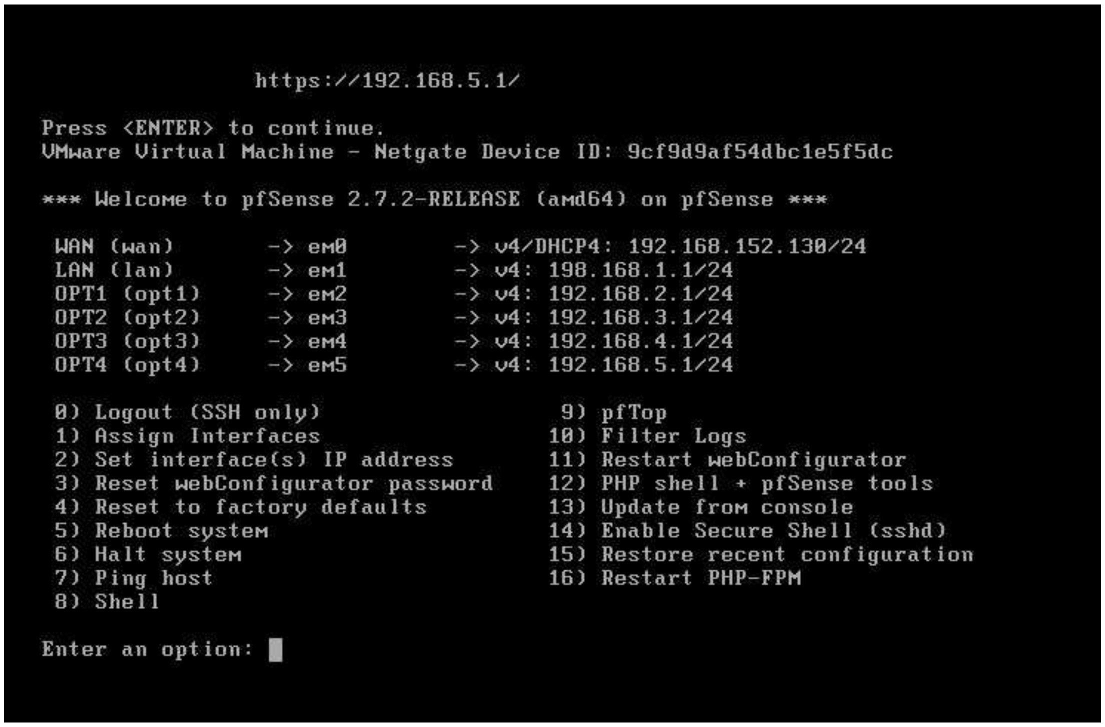
Figure 2: Segmentation has begun.

I then went through and assigned interfaces and then set interface IP addresses as well. With this basic configuration complete, I should soon be able to use the web GUI on another VM to further set it up as I get the other VMs set up with snort, ELK stack, etc… [1] [2]



```
Configure IPv4 address LAN interface via DHCP? (y/n) n

Enter the new LAN IPv4 address.  Press <ENTER> for none:
> 192.168.1.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new LAN IPv4 subnet bit count (1 to 32):
> 24

For a WAN, enter the new LAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
>

Configure IPv6 address LAN interface via DHCP6? (y/n) n

Enter the new LAN IPv6 address.  Press <ENTER> for none:
>

Do you want to enable the DHCP server on LAN? (y/n) y
Enter the start address of the IPv4 client address range: 192.168.1.10
Enter the end address of the IPv4 client address range: 192.168.1.200
```

Figure 3: Configuring LAN interface settings.



```
                https://192.168.5.1/

 Press <ENTER> to continue.
 VMware Virtual Machine - Netgate Device ID: 9cf9d9af54dbc1e5f5dc

 *** Welcome to pfSense 2.7.2-RELEASE (amd64) on pfSense ***

  WAN (wan)       -> em0        -> v4/DHCP4: 192.168.152.130/24
  LAN (lan)       -> em1        -> v4: 198.168.1.1/24
  OPT1 (opt1)     -> em2        -> v4: 192.168.2.1/24
  OPT2 (opt2)     -> em3        -> v4: 192.168.3.1/24
  OPT3 (opt3)     -> em4        -> v4: 192.168.4.1/24
  OPT4 (opt4)     -> em5        -> v4: 192.168.5.1/24

  0) Logout (SSH only)                  9) pfTop
  1) Assign Interfaces                 10) Filter Logs
  2) Set interface(s) IP address       11) Restart webConfigurator
  3) Reset webConfigurator password    12) PHP shell + pfSense tools
  4) Reset to factory defaults         13) Update from console
  5) Reboot system                     14) Enable Secure Shell (sshd)
  6) Halt system                       15) Restore recent configuration
  7) Ping host                         16) Restart PHP-FPM
  8) Shell

 Enter an option:
```

Figure 4: Completed basic configuration.

Kali Machine

I also got the Kali machine up and running, updated. And ready to help further set pfSense up when the time comes. I do believe that there is some additional tweaking that I need to investigate and may even have to redo this part but that is part of the experience and "fun." When I tried to connect to the pfSense web GUI it timed out. I changed Kali's NIC to the same LAN pfSense NIC, but no connection can be reached quite yet.

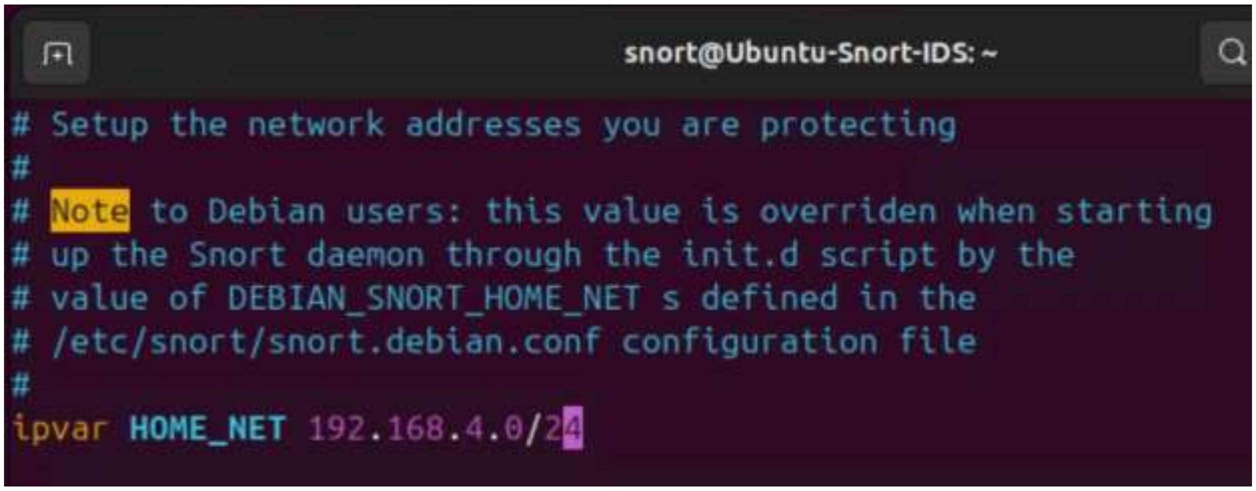
Learning Experience

I've gained a lot of experience in installation of pfSense, and this experience would certainly help installing pfSense bare metal as opposed to my prior experience of installing it via a ubuntu server. I hadn't configured virtual NICs or pfSense opts for a wider network before either – my prior pfSense experience was just configuring it for the device it was installed on. This is honestly still an ongoing learning experience, and I am sure it will continue to be so even after I fully configure this and troubleshoot out other issues.

Milestone 2 Report

Snort Initial Configuration

I had decided to install Snort of an Ubuntu VM so I got that provisioned with 8GB of memory, 50GB of NVME space, and 4 processors. I also added VMnet4 and VMnet5 to the machine, for the connections to the Domain Controlled machines on one and the ELK stack on the other. After running all update && upgrades I then installed Vim and Snort. The installation was straight forward, and after I found the config adjusted the settings with Vim. I set the HOME_NET variable to the IP range that the Domain Controlled machines will be on.

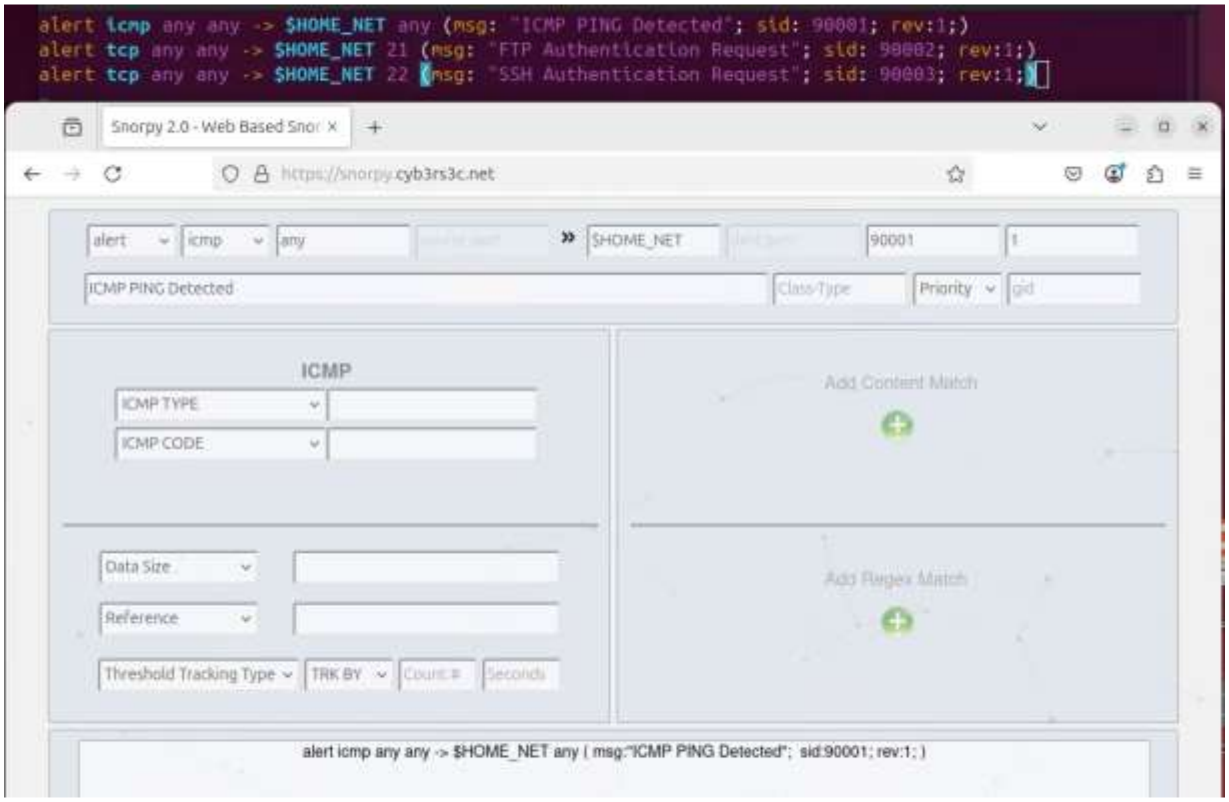


```
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overriden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET s defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.4.0/24
```

Figure 5: I only want to monitor this one network for now.

In the same config file, we learn where the local.rules file is as well, so I open that next with Vim to add some new rules. I decided to add what I feel would be typical alerts, some for pinging, FTP connection, and SSH connections made, and found a cool tool to help me craft the first rule. When I did some ifconfigs I realized that the DHCP server from pfSense was off for the OPT segments, so I then went and enabled that real quick before restarting machines to be assigned IP addresses.

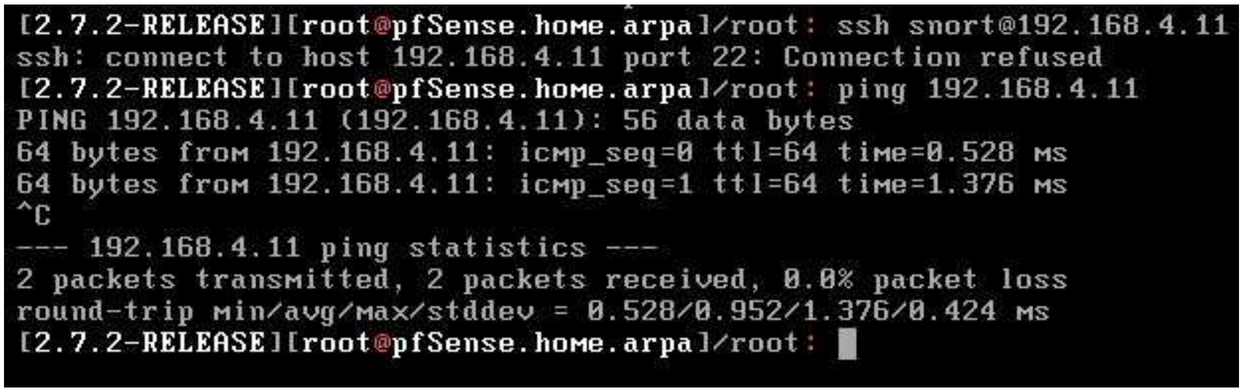
With that done I then used a shell on the pfSense machine to make an SSH connection to the ubuntu machine that runs Snort, as well as some pings to the machine. Snort alerts pop off as expected. There may be more adjustments as I go on, but I am happy to see that it is functioning as expected. [3] [4]



```
alert icmp any any -> $HOME_NET any (msg: "ICMP PING Detected"; sid: 90001; rev:1;)
alert tcp any any -> $HOME_NET 21 (msg: "FTP Authentication Request"; sid: 90002; rev:1;)
alert tcp any any -> $HOME_NET 22 (msg: "SSH Authentication Request"; sid: 90003; rev:1;)
```

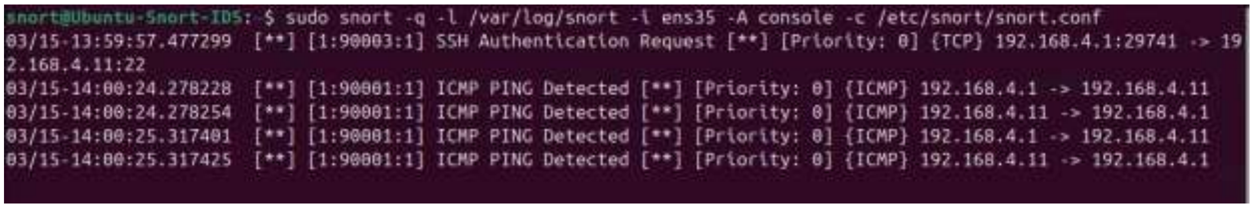
```
alert icmp any any -> $HOME_NET any ( msg:"ICMP PING Detected"; sid:90001; rev:1; )
```

Figure 6: Rules being written.



```
[2.7.2-RELEASE][root@pfSense.home.arpa]/root: ssh snort@192.168.4.11
ssh: connect to host 192.168.4.11 port 22: Connection refused
[2.7.2-RELEASE][root@pfSense.home.arpa]/root: ping 192.168.4.11
PING 192.168.4.11 (192.168.4.11): 56 data bytes
64 bytes from 192.168.4.11: icmp_seq=0 ttl=64 time=0.528 ms
64 bytes from 192.168.4.11: icmp_seq=1 ttl=64 time=1.376 ms
^C
--- 192.168.4.11 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.528/0.952/1.376/0.424 ms
[2.7.2-RELEASE][root@pfSense.home.arpa]/root:
```

Figure 7: Here we are testing.



```
snort@Ubuntu-Snort-IDS:~$ sudo snort -q -l /var/log/snort -i ens35 -A console -c /etc/snort/snort.conf
03/15-13:59:57.477299  [**] [1:90003:1] SSH Authentication Request [**] [Priority: 0] {TCP} 192.168.4.1:29741 -> 19
2.168.4.11:22
03/15-14:00:24.278228  [**] [1:90001:1] ICMP PING Detected [**] [Priority: 0] {ICMP} 192.168.4.1 -> 192.168.4.11
03/15-14:00:24.278254  [**] [1:90001:1] ICMP PING Detected [**] [Priority: 0] {ICMP} 192.168.4.11 -> 192.168.4.1
03/15-14:00:25.317401  [**] [1:90001:1] ICMP PING Detected [**] [Priority: 0] {ICMP} 192.168.4.1 -> 192.168.4.11
03/15-14:00:25.317425  [**] [1:90001:1] ICMP PING Detected [**] [Priority: 0] {ICMP} 192.168.4.11 -> 192.168.4.1
```

Figure 8: And our alerts.

Learning Experience

I learned a lot of good basics of Snort and rule writing for it, the biggest takeaway from this though is that it is just the beginning. Sure, my few rules and the other default rules are at working and not flagging a whole lot – but my network it is currently monitoring is a whole two machines with no other IoT, machines, and other devices on the network sending various communications, broadcasts, etc. the bigger obstacle will be once I get the Domain Control server and other machines setup, any false alerts coming up and rewriting the rules and adjusting settings so those occur less or not at all.

I also see that the configuration for this is vastly different than what something like Security Onion would have been, as Security Onion has a neat little UI as opposed to using Vim, Nano, or some other text editor to edit a text file. Learning with Snort, I think will allow me to take that experience to other IDS/IPS applications and appreciate when that have a UI and otherwise not be so intimidated if it's a text edit config.

Milestone 3 Report

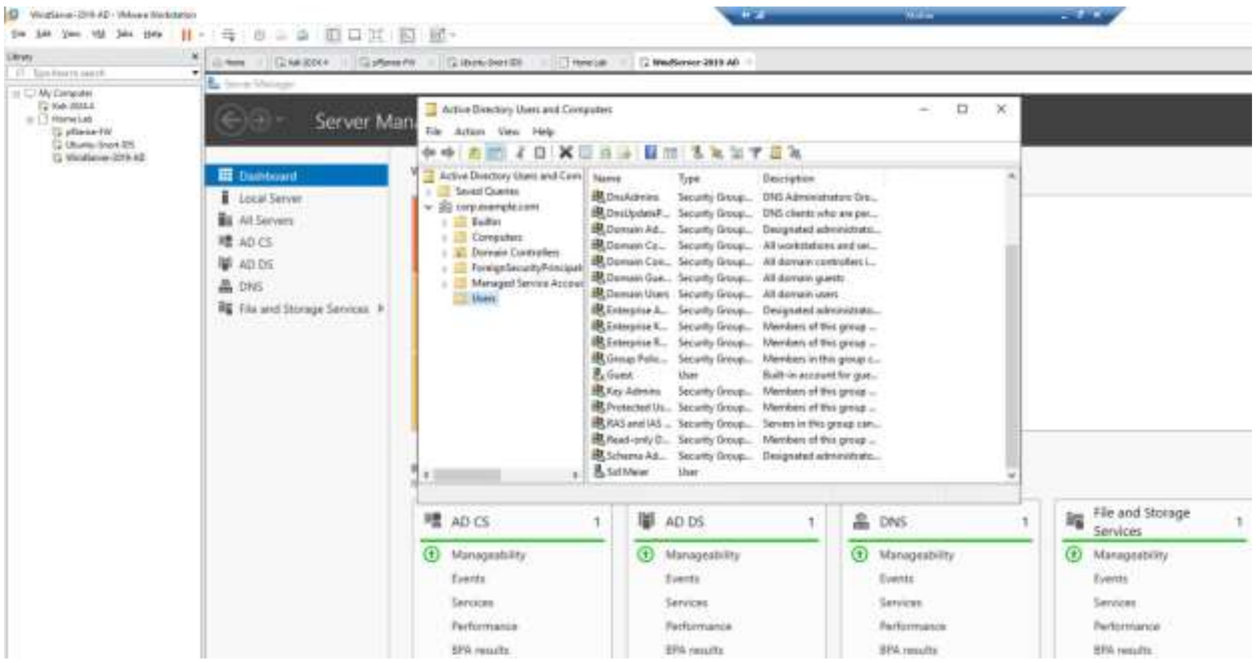
Windows Server 2019

I installed Windows Server with little to no issue, I do find I have to point the VM's optical drive back to the ISO and after one weird boot error pushed the memory to 4GB, up from 2GB, which resolved it. I also set the Network Interface to VNET 4. I changed the name of the machine, in settings, to "ADServer" and then start poking around through the Server Manager Dashboard – I had elected to install with the Desktop GUI.

The next prompts walk me through it well, I took a moment to read over each option before adding the features of "Active Directory Domain Services." I started the installation, and closed the wizard as it mentions it will give me a notification once it is complete. Once complete I was prompted to promote the server to a Domain Controller in the notifications, which is what I set out to do here.

Knowing this is new I select "Add a new forest" and continue going allow and reading with the steps, I opted to name it "corp.example.com" – this AD and project aren't going to be put to full practicality, obviously, but good to try and follow what seems to be a standard. I left the other options as their default, changing no directories, etc., and installed the configuration.

Rebooting in a see the new admin account and login. It takes a moment for the dashboard to refresh, but I see our new AD settings. I finish up by setting up a user account, "Sid Meier." Lastly, I set the IP address to be static, at *192.168.5.12*. It seems like when I set up pfSense's OPTs that I mixed up a couple of the VMnet ordering, so IP ranges and VMnet numbering don't match down sequentially but that is okay, I will be auditing and updating the network diagram soon enough.



Name	Type	Description
DnsAdmins	Security Group...	DNS Administrators Gro...
DnsUpdateP...	Security Group...	DNS clients who are per...
Domain Ad...	Security Group...	Designated administrato...
Domain Co...	Security Group...	All workstations and ser...
Domain Con...	Security Group...	All domain controllers i...
Domain Gue...	Security Group...	All domain guests
Domain Users	Security Group...	All domain users
Enterprise A...	Security Group...	Designated administrato...
Enterprise K...	Security Group...	Members of this group ...
Enterprise R...	Security Group...	Members of this group ...
Group Polic...	Security Group...	Members in this group c...
Guest	User	Built-in account for gue...
Key Admins	Security Group...	Members of this group ...
Protected Us...	Security Group...	Members of this group ...
RAS and IAS ...	Security Group...	Servers in this group can...
Read-only D...	Security Group...	Members of this group ...
Schema Ad...	Security Group...	Designated administrato...
Sid Meier	User	

Figure 9: Sid Meier is not quite ready to continue his good work, but soon.

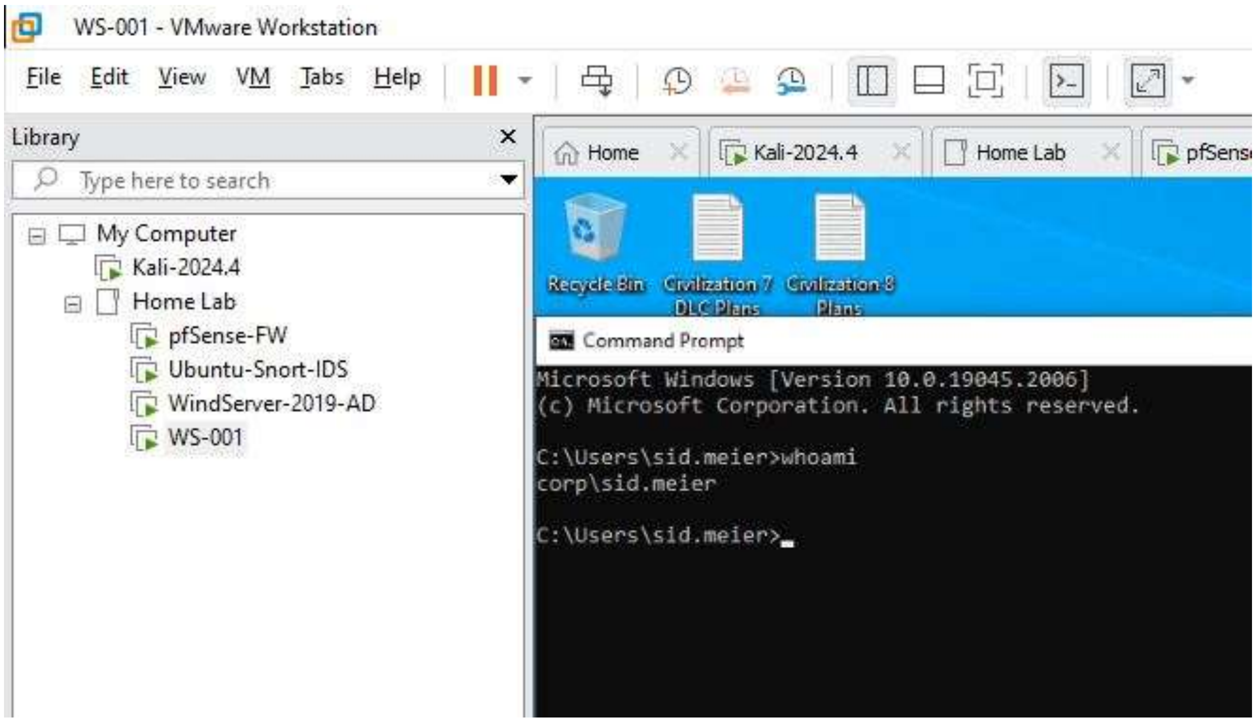
Windows Machine

I originally had planned on setting up two windows 10 work machines but think until I get the ELK stack up and see where we are standing with available hardware that we will just work with one "worker terminal" for now. I navigated to the system setting and found I was unable to connect to it. Since the 2019 Server was set up as DNS as well and is the focal point for this machine, I use *ipconfig /all* and see that the DNS server is pointed at the regular gateway IP, 192.168.5.1 instead of the 2019 server, 192.168.5.12. So, I go to the network settings and adjust. I flush the DNS, register DNS, and try again and am successful. [5]

Active Directory Org Unit, and Test Login

Back on the 2019 Server, I created a new Organizational Unit called "Office Admin" to make it easier to apply group policies, and "as the business grows and we add more staff" they are then able to be easily managed and added to the org unit. I move WS-001 to the new unit and prepare to login to the domain for the first time.

Back on our WS-001, I select other users and see the CORP domain. If I saw something such as IT, or some other subdomain I would realize at this point (hopefully) that I am at the wrong computer. I go on and enter Sid Meier's credentials and go through the initial "Let's get started" Windows 10 prompt before opening Command Prompt and perform a *whoami* command.



```
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sid.meier>whoami
corp\sid.meier

C:\Users\sid.meier>_
```

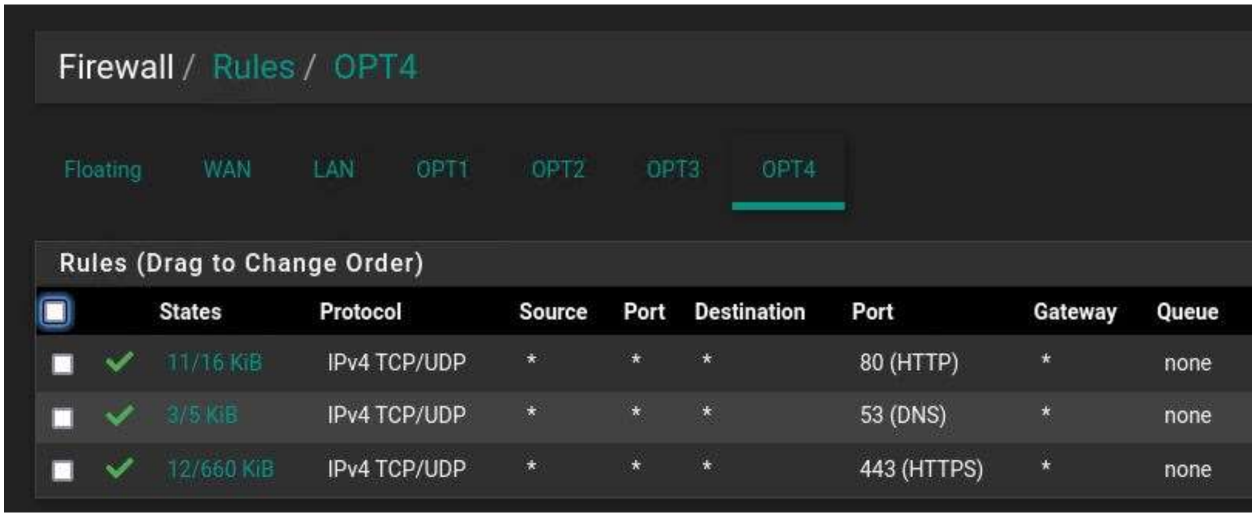
Figure 10: Now he is ready to work.

pfSense Configuration Check and Network Audit

At this point I want to check back in with my pfSense server to see if I need to change any other settings. I essentially did some reconfigurations sooner than I thought in Milestone 2 Report (See Snort Initial Configuration paragraph 2) with DHCP. Knowing all the devices so far do not and have not had WAN access so far, I decided to look at the DNS settings on pfSense. I go to general settings and set the DNS servers to Cloudflare's DNS server, then Google's.

I use the interface settings and some *ipconfig / ifconfig* commands to audit and notate the network so far (see this reflected in Rough Diagram of Network).

With our Workstation now setup and Sid needing some access to internet I look at the LAN rules for comparison, and then OPT4 rules, which by default has none. I must allow the network to reach out to the internet, so I add some rules to do so, and may be tweaked some more later.



		States	Protocol	Source	Port	Destination	Port	Gateway	Queue
	✓	11/16 KiB	IPv4 TCP/UDP	*	*	*	80 (HTTP)	*	none
	✓	3/5 KiB	IPv4 TCP/UDP	*	*	*	53 (DNS)	*	none
	✓	12/660 KiB	IPv4 TCP/UDP	*	*	*	443 (HTTPS)	*	none

Figure 11: The internet (is now allowed).

Learning Experience

I got a good handle on setting up and managing a Windows Server environment, going through the process of configuring Active Directory and organizing users. I mostly let myself stumble through the setup, figuring things out as I went with minimal outside help—only looking up details on the ipconfig command from Microsoft when needed. Along the way, I had to work through a few issues, which gave me a better grasp of troubleshooting and adjusting network settings. Working with pfSense also helped me understand firewall rules and how to structure network access. Overall, this was a solid hands-on experience that gave me a better feel for server administration and how all these pieces come together.

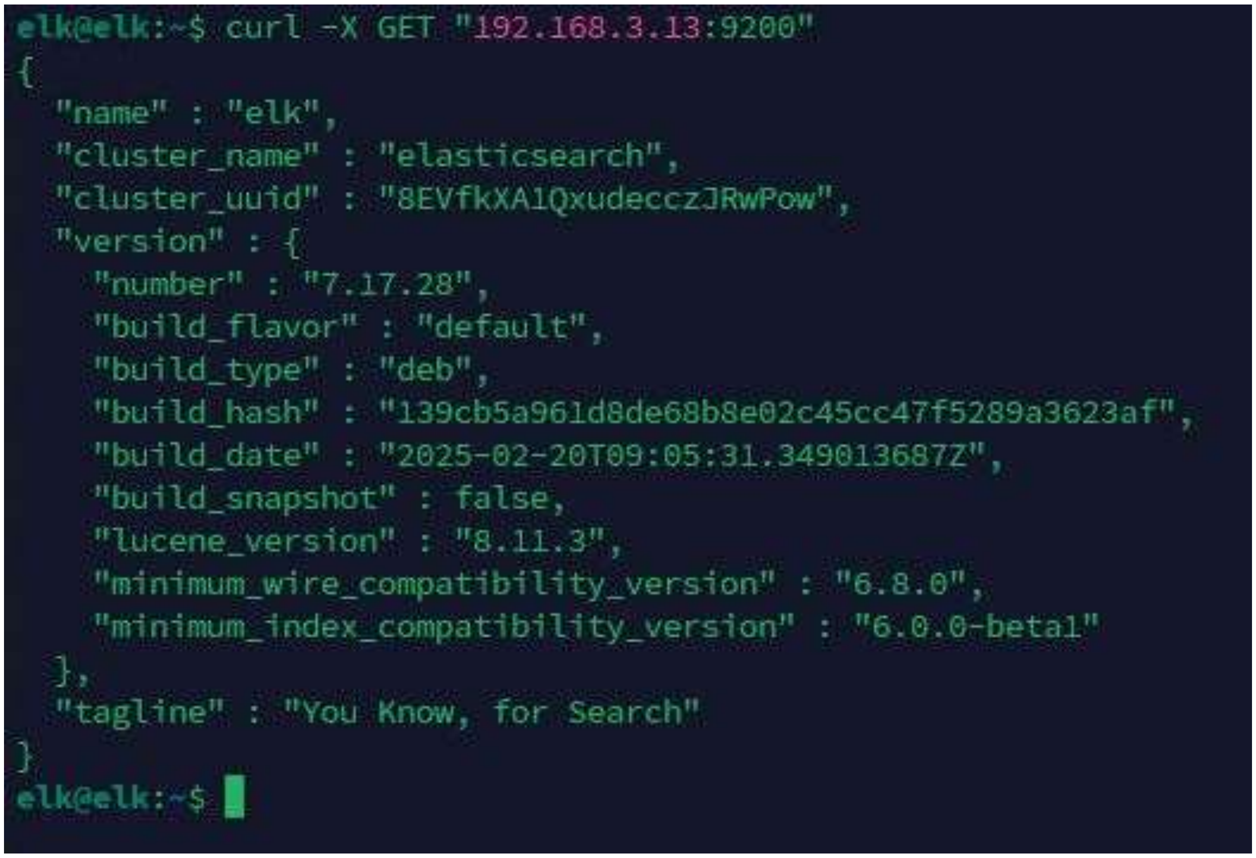
Milestone 4 Report

Prep and Elasticsearch

As I begin, I boot the machine up and then elect to ssh into it from my host machine using Termius, so that I may easily copy and paste commands, as well as just have a better all around experience. I set up an Ubuntu Server to install my ELK stack onto, though I will acknowledge from some reading that in larger deployments it would be better to have multiple instances and nodes for an installation. [6]

I take a moment to install Java, as it is a dependency, and NGINX to act as a web and proxy server to access the Kibana dashboard. I *wget* the apt-key for Elasticsearch and install apt-transport-https so I can fetch packages over HTTPS for Elastic's repo, which I add to the system's repository and the update the list of available system packages. After installing Elasticsearch I use vim to configure it.

I rename the cluster name to "home-lab" but otherwise leave most settings the default. I set the network.host to the server's IP address; 192.168.3.13, then move onto the Discovery section to add a line to help with our small deployment. I started Elasticsearch and enabled it to automatically start when the system boots up then run a *curl* command to test the configuration.

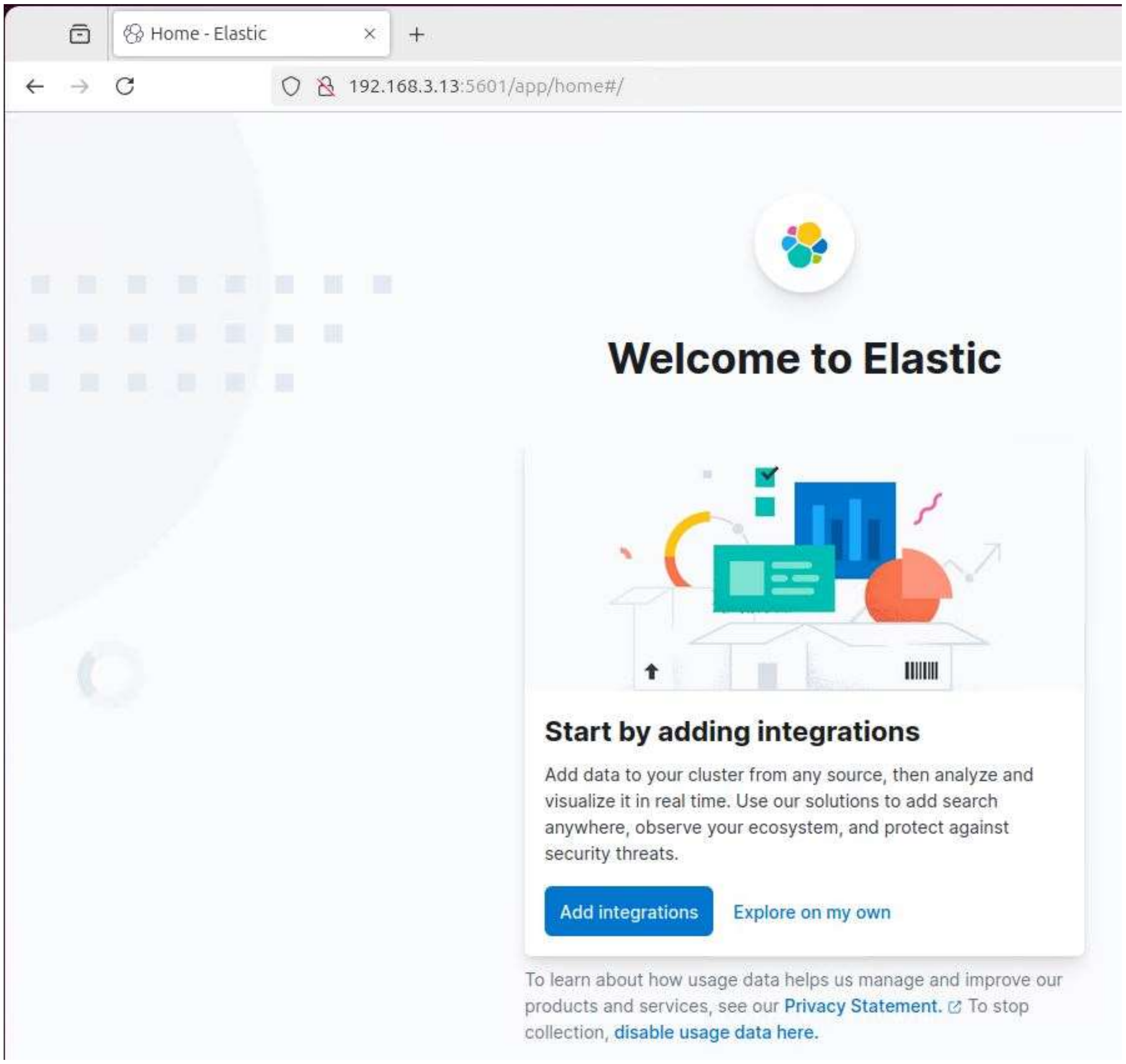


```
elk@elk:~$ curl -X GET "192.168.3.13:9200"
{
  "name" : "elk",
  "cluster_name" : "elasticsearch",
  "cluster_uuid" : "8EVfkXA1QxudecczJRwPow",
  "version" : {
    "number" : "7.17.28",
    "build_flavor" : "default",
    "build_type" : "deb",
    "build_hash" : "139cb5a961d8de68b8e02c45cc47f5289a3623af",
    "build_date" : "2025-02-20T09:05:31.349013687Z",
    "build_snapshot" : false,
    "lucene_version" : "8.11.3",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
elk@elk:~$ 
```

Figure 12: We've got Elasticsearch configured.

Kibana

I installed Kibana, configured the *kibana.yml* to our local device and to communicate with Elasticsearch. I went to try and access the Kibana web interface via the bad actor Kali machine that found an access point but was unable to which reminds me that we need to configure pfSense to allow traffic on that port. The 192.168.3.1/24 range is the OPT2 interface, so I set a firewall to allow traffic from LAN interface to access OPT2 subnets and port 5601, and vice versa on both interfaces but was still unable to connect. I have a Ubuntu machine we will play pretend is on the subnet range as the ELK stack that IT uses to access it and will troubleshoot some more later. Otherwise, to me, it would absolutely make sense that the interface would just but setup to be accessed by the subnet it is on, an IT/Security subnet of machines (example).



Welcome to Elastic

Start by adding integrations

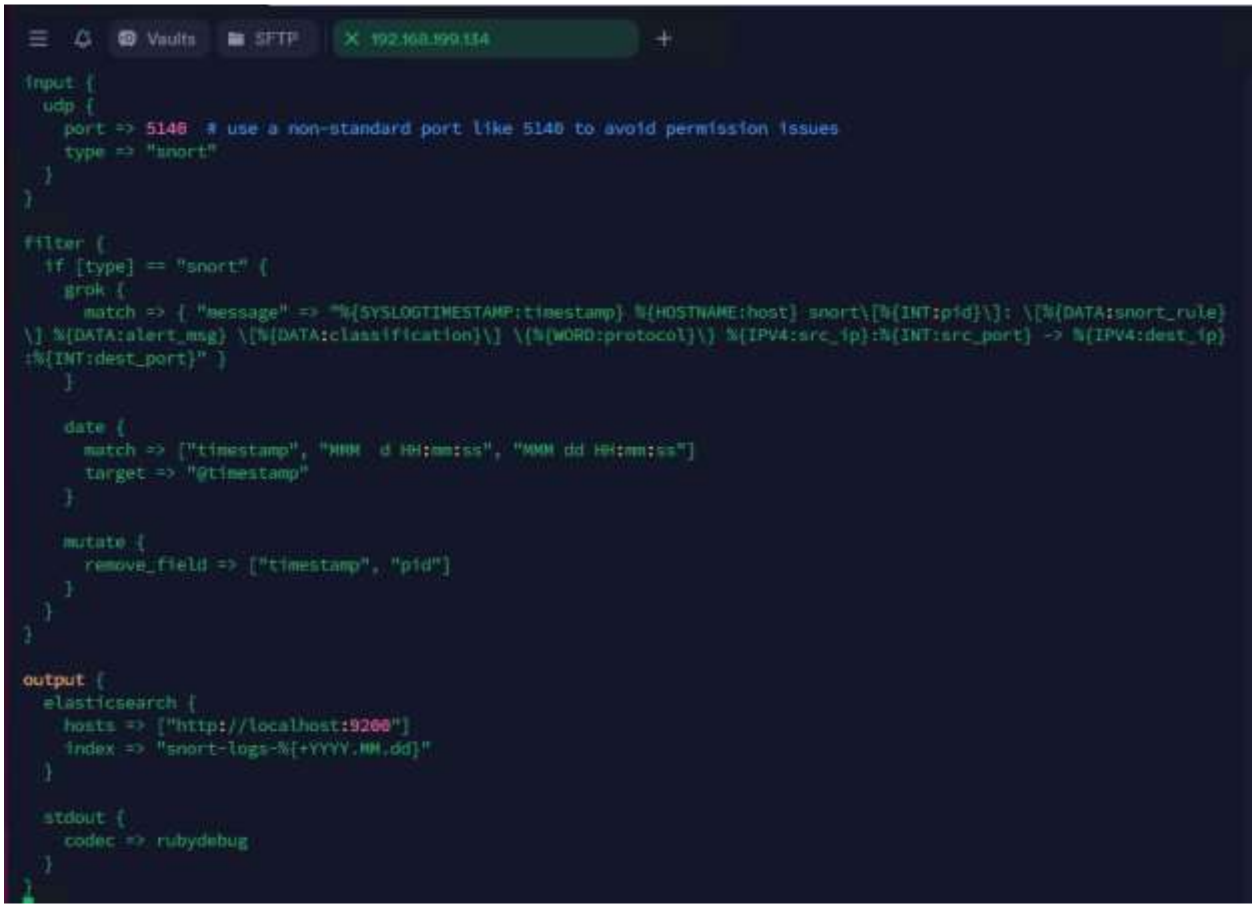
Add data to your cluster from any source, then analyze and visualize it in real time. Use our solutions to add search anywhere, observe your ecosystem, and protect against security threats.

To learn about how usage data helps us manage and improve our products and services, see our Privacy Statement. To stop collection, disable usage data here.

Figure 13: We can access Elastic Web UI locally.

Logstash and Filebeat

I install Logstash, and start and enable at startup. I install Filebeat as well, after making sure Kibana is currently running.



```
input {
  udp {
    port => 5140  # use a non-standard port like 5140 to avoid permission issues
    type => "snort"
  }
}

filter {
  if [type] == "snort" {
    grok {
      match => { "message" => "%{SYSLOGTIMESTAMP:timestamp} %{HOSTNAME:host} snort\[%{INT:pid}\]: \[%{DATA:snort_rule}
\] %{DATA:alert_msg} \[%{DATA:classification}\] \{%{WORD:protocol}\} %{IPV4:src_ip}:%{INT:src_port} -> %{IPV4:dest_ip}
:%{INT:dest_port}" }
    }

    date {
      match => ["timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss"]
      target => "@timestamp"
    }

    mutate {
      remove_field => ["timestamp", "pid"]
    }
  }
}

output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "snort-logs-%{+YYYY.MM.dd}"
  }

  stdout {
    codec => rubydebug
  }
}
```

I also head back to the Snort machine and configure the *output alert_syslog:* to forward to the ELK stack machine.

Configuration to be continued

Going back to the Kibana dashboard, I go to add the integrations. I try to start setting up the pipeline further, but I've got some configuration wrong somewhere. As I have been going over these configurations and looking for additional information off and on. I think I will have to work on this additionally, and keep going with configurations in Milestone 5.

Learning Experience

Configurations galore, it required a lot of trial and error as even these initial setups with small mistakes cause those initial configurations even to fail. I am starting to realize that there is for sure a chance I do not fully finish this during the semester, but I absolutely may end up later scrapping this and rebuilding from scratch with more detailed documentation, and slow steps so that everything is configured with much better documentation on my part.

It absolutely makes sense that there are specialists whose entire position is just building and maintaining systems such as these.

Bibliography

[1] M. Alani, "Installing and configuring PfSense firewall on a virtual machine," n/a, 9 May 2023. [Online]. Available: https://www.youtube.com/watch?v=VxFGymlrOc0. [Accessed 27 February 2025].

[2] InfoSec Pat, "How To Install And Setup pfSense Firewall On VMware Workstation Pro - InfoSec Pat 2024," 20 February 2024. [Online]. Available: https://www.youtube.com/watch?v=Ayr_av2EX_U. [Accessed 27 February 2025].

[3] A. N. Aytemiz, "How to Install and Configure Snort on Ubuntu," LetsDefend, 30 August 2024. [Online]. Available: https://letsdefend.io/blog/how-to-install-and-configure-snort-on-ubuntu. [Accessed 12 March 2025].

[4] C. Davis, "SNORTY," n.a. n.a. n.a.. [Online]. Available: https://snorpy.cyb3rs3c.net/. [Accessed 15 March 2025].

[5] Microsoft, "ipconfig," Microsoft, n.d. n.d. n.d.. [Online]. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/ipconfig. [Accessed 1 April 2025].

[6] "Mukul", "https://www.tetrain.com/blogs/post/105/6-best-practices-for-setting-up-configuring-the-elk-stack.html," Tetra, 18 October 2024. [Online]. Available: https://www.tetrain.com/blogs/post/105/6-best-practices-for-setting-up-configuring-the-elk-stack.html. [Accessed 14 April 2025].

[7] elastic, "Elastic Documentation," Elastic, n.d. n.d. n.d.. [Online]. Available: https://www.elastic.co/guide/index.html. [Accessed 12 April 2025].