**Homelab Setup:**

**Initial Setup:**

1. Installed Virtualbox
2. Installed Windows Server 2022 (Domain Controller), Windows 10, Kali Linux, pfSense (Router/Firewall), Security Onion (Logging), VDIs.
3. Add pfSense, Windows Server 2022, Windows 10, Kali Linux, SecurityOnion, Ubuntu Server ISOs to Virtualbox
   1. Allocated 3GB of RAM per VM
      1. 12GB for SecurityOnion
      2. 4GB for Splunk
   2. Allocated 2 Processors per VM
   3. Allocated 30GB of Storage per VM for Windows Server/Windows 10/Kali Linux
      1. Automatically set to ‘dynamically allocated’ to change based on needs
   4. Allocated 10GB of Storage for pfSense
      1. Automatically set to ‘dynamically allocated’ to change based on needs
   5. Allocated 100GB of Storage for Security Onion
      1. Automatically set to ‘dynamically allocated’ to change based on needs
   6. Allocated 50GB of Storage for Splunk
      1. Automatically set to ‘dynamically allocated’ to change based on needs
4. Change boot order for each VM to Hard Disk first then Optical Drive (After initial boot)
   1. This is to help with boot speeds following initial configurations
5. Select Proper ISO files for Each VM for Optical Drives.

**Setup pfSense:**

1. In Virtualbox Manager
   1. Set Network Adapters:
      1. Adapter 1: NAT for WAN
      2. Adapter 2: Internal Network for Windows Gateway/LAN
      3. Adapter 3: ‘External’ Attacker Network for Kali Linux Gateway/LAN
      4. Adapter 4: Network for Management of SecurityOnion/Splunk
      5. Adapter 5: Network for SPAN Port
         1. Virtualbox GUI doesn’t support for NIC 5-8 configuration, must be through command line
            1. cd "C:\Program Files\Oracle\VirtualBox"
            2. VBoxManage list vms

Select name for pfSense, mine is “PfSense”

* + - * 1. VBoxManage modifyvm "PfSense" --nic5 intnet

Adds internal network type for nic5

* + - * 1. VBoxManage modifyvm "PfSense" --intnet 5 “Span-Network”

Sets Internal Network name

* + - * 1. VBoxManage modifyvm "PfSense" --nicpromisc5 allow-all

Allows for Promiscuous mode

Allows all traffic on segment not just traffic directed to pfSense

1. Boot pfSense
2. Following Install Procedure:
   1. Install pfSense
   2. Keep Default Keymapping
   3. Partition Disk via ‘Auto UEFI’
      1. Faster boot than ‘Auto BIOS’
3. Setup em1 IPV4 Gateway Address and DHCP Pool for Windows/Victim Network:
   1. Set Interface(s) IP Address
   2. Select LAN Interface
      1. **Gateway Address/Netmask: 192.168.50.1/24**
   3. Decline WAN IPV4 upstream gateway address
      1. Already using Virtualbox’s NAT as WAN IPV4 upstream gateway address
   4. Decline IPV6 setup (Not needed, 4 hosts total)
   5. **Setup DHCP Server Pool Range**
      1. **192.168.50.10 - 192.168.50.100**
   6. Allow HTTP as webConfigurator Protocol
      1. Allows for HTTP web Configuration of pfSense for various Services
4. Setup em2 IPV4 Gateway Address and DHCP Pool for Kali Linux/Attacker Network:
   1. Set Interface(s) IP Address
   2. Select LAN Interface
      1. **Gateway Address/Netmask: 192.168.100.1/24**
   3. Decline WAN IPV4 upstream gateway address
      1. Already using Virtualbox’s NAT as WAN IPV4 upstream gateway address
   4. Decline IPV6 setup (Not needed, 4 hosts total)
   5. **Setup DHCP Server Pool Range**
      1. **192.168.100.10 - 192.168.100.100**
5. Change LAN1 (Victim) DHCP Settings:
   1. The following will allow additional PCs to join the domain seamlessly
      1. Have DC1 (192.168.50.101) as Primary DNS Server
      2. Change Domain Name to homelab.local
6. Setup em3 IPV4 Gateway Address and DHCP Pool for Management Network:
   1. Set Interface(s) IP Address
   2. Select LAN Interface
      1. **Gateway Address/Netmask: 192.168.150.1/24**
   3. Decline DHCP Setup
      1. Will utilize static mapping
7. Setup em3 IPV4 Gateway Address and DHCP Pool for SpanPort (Mirror LAN1):
   1. Don’t enter an IP address
      1. Will bridge the network to LAN1 later during Web Configuration

**Setup Windows Server 2022:**

1. In Virtualbox Manager
   1. Change Network Adapter 2 to Internal Network for LAN1
2. Boot/Setup Windows Server 2022
   1. Select Standard Desktop Experience
   2. Allow Default Storage Allocation
   3. Setup Administrator Account
3. Rename PC via About Your PC to DC1
4. Reboot
5. Install Virtualbox Guest Additions
   1. Enhances performance/usability
6. Change IPV4 Ethernet Adapter to use loopback (127.0.0.1) as primary DNS server
   1. Will still use pfSense as a backup if it can’t resolve locally

**Setup Active Directory on Windows Server 2022:**

1. Install Active Directory Domain Services via Server Manager on Windows Server 2022
   1. Promote Windows Server 2022 as Domain Controller
      1. Create new forest
         1. Domain Name: (homelab.local)
      2. Forest Functional Level: 2016
      3. Install DNS on Windows Server 2022
      4. Set as Global Catalog (CC) Server
      5. Leave ‘Additional Options’ and ‘Paths’ as Default
   2. Complete Domain Controller Installation
2. Turn off Windows Firewall/Defender \*
   1. Allows for an environment full of vulnerabilities to exploit

**Setup Certificate Services on Windows Server 2022:**

1. Install Active Directory Certificate Services on Windows Server 2022
   1. Add only Certification Authority as a Role Service
      1. Other Roles over not needed for the purpose of this lab
2. Active Directory Certificate Services Configuration:
   1. Select Enterprise CA
      1. Simplifies management of Certificates
   2. Select Root CA
      1. Since DC will be atop the PKI hierarchy
   3. Create new private key
      1. One doesn’t already exist
   4. Select SHA256
      1. Secure and compatible with most system
   5. Keep default Common/Distinguished Name:
      1. Common: homelab-DC1-CA
      2. Distinguished: CN=homelab-DC1-CA, DC=homelab, DC=local
   6. Change validity to 99 years
      1. Prevents issue if the lab is still in use years down the line
   7. Keep default Certificate database/log location:
      1. C:\Windows\system32\CertLog

**pfSense Web Configuration via Edge on Windows Server 2022:**

1. Launch Edge
2. Access Gateways (192.168.50.1) pfSense Configuration Portal
3. Login via default pfSense credentials
4. Complete pfSense Setup
   1. Keep default hostname: pfSense
   2. Domain: lab.local
   3. Primary DNS Server: 1.1.1.1 (Cloudflare DNS)
   4. Secondary DNS Server: 8.8.8.8 (Google's DNS)\
   5. Uncheck DNS Override
      1. Prevents WAN DHCP from overriding configuration
   6. Keep default Time Server configuration
   7. Keep default WAN configuration (Already using VirtualBox NAT as WAN)
   8. Uncheck ‘Block RFC1918 Private Networks’ and ‘Block bogon networks’
      1. Since all interfaces within the lab use private addresses these will need to be unchecked for upstream communication and to allow VMs to test the firewall
   9. Keep existing LAN IP Address configuration 192.168.50.1/24
   10. Change admin password
5. Default Allow Any Protocol from LAN to Any Destination rule for IPV4/6 and Anti-Lockout Rule for LAN is established
6. Copy ALLOW rules for OPT1(Attacker)/OPT2(Management) in the Web Config Portal
7. Change Names for various interfaces:
   1. em1= VICTIM
   2. Em2 = KALIATTACKER
   3. Em3 = MANAGEMENTSECONIONSPLUNK
   4. Em4 = SPANPORT
8. Bridge SPANPORT to VICTIM:
   1. Interfaces > Assignments > Bridges:
      1. Add
         1. Select VICTIM
         2. Advanced > Scroll to SPANPORT > Select SPANPORT > Save

**Add Static DHCP Mapping for DC1 (Windows Server 2022):**

1. Access Gateways (192.168.50.1) pfSense Configuration Portal
2. Access DHCP Server via Services Dropdown Menu
3. Add Static DHCP Mapping
   1. IP Address: **192.168.50.101**
   2. Hostname: DC1
4. Keep other configurations default

**Add Users to Domain:**

1. Access Active Directory Users and Computers
2. Click homelab.local Dropdown Menu
3. Select Users
4. Right-Click and Select New User
5. Add ‘Bill’ and ‘Fred’ to the homelab.local Domain
   1. Used ‘Bill’ and ‘Fred’ as Logon to keep it simple

**Setup Windows 10 Pro (PC1):**

1. In Virtualbox Manager
   1. Change Network Adapter 2 to Internal Network for LAN1
2. Boot/Setup Windows 10
   1. Select Windows 10 Pro
   2. Allow Default Storage Allocation
   3. Setup Offline Admin Account
3. Rename PC via About Your PC to PC1
4. Reboot
5. Install Virtualbox Guest Additions
   1. Enhances performance/usability
6. Change IPV4 Ethernet Adapter to use DC1s IP (192.168.50.101) as primary DNS server
   1. Will still use pfSense as a backup if it can’t resolve locally

**Join PC1 to homelab.local Domain: (Step is only necessary if PC1 didn’t automatically join domain through previous pfSense configuration)**

1. Open ‘About Your PC’
2. Select Advanced System Settings
3. Select Change
4. Select Domain and enter homelab.local
5. Enter Admin Credentials
6. Reboot PC1

**Setup Kali Linux (PC2):**

1. In Virtualbox Manager
   1. Change Network Adapter 2 to Internal Network for LAN2
2. Boot/Setup Kali Linux
   1. Select Graphical Install
   2. Select Default Language/Location
   3. Use eth0 (NAT/WAN) interface for installation
   4. Enter domain (homelab.local) (Doesn’t actually add to AD Domain)
   5. Setup Non-Admin Account:
      1. Full Name: John
      2. Username: john (Must be lowercase)
   6. Select TimeZone
   7. Select Virtual Disk and default Partition Options
   8. Keep default Desktop Environment Install and Tools
   9. Install GRUB boot loader
3. Update/Upgrade Packages
4. Reboot

**Set Static IP Mapping for Security Onion via Edge on DC1: \***

1. Access pfSense Portal
2. Status
3. DHCP Lease
4. Select Add Static Mapping Option
   1. **Security Onion IPV4 Static Address: 192.168.50.102**
5. NOTE: If you use the DHCP pool and then try to change to a static address later in either pfSense or Security Onion Server the Docker running Security Onion SOC won’t change the IP Address since its locked in after the initial setup

**Setting up SecurityOnion (seconion):**

1. In Virtualbox Manager
   1. Change Network Adapter 2 to Internal Network for LAN3
      1. Management Interface for Web Interface
   2. Change Network Adapter 3 to Internal Network for SPAN-Network (Promiscuous Mode)
      1. This NIC is going to be the Monitor Interface for all of LAN1
2. Boot/Setup Security Onion
   1. Setup Administrative Account
      1. User/Password
   2. Allow Setup script to run
   3. Reboot
   4. Login
   5. Continue with SecurityOnion setup
   6. Standard Installation
   7. Evaluation Installation Mode (Quick, all-in-one testing)
   8. Set Hostname (seconion)
   9. Select emp0s8 as the Management Interface
   10. **Select Static IP** 
       1. **IP Address: 192.168.150.10/24**
   11. Enter Gateway
       1. Gateway: 192.168.150.1
   12. DNS Server Config:
       1. Add DC1 (192.168.50.101) as Primary (Internal Inquires)
       2. pfSense Management Interface (192.168.150.1) as Secondary DNS Server for non-domain inquires
   13. DNS Search Domain
       1. Same as AD Domain: homelab.local
          1. Allows seconion to resolve hosts with typing full FQDN
             1. Ex. dc1 to dc1.homelab.local
   14. Select Standard Setup
       1. Allows for internet access
   15. Direct Internet Access
   16. Pre-Install Checkups for Packages
   17. Select emp0s9 as the Monitor Interface
       1. Will utilize the Span-Network NIC setup earlier
   18. Automatic Patch Schedule
   19. Change Home Network to 192.168.50.0/24
       1. Allows Security Onion to know which is the internal (trusted) network and external (untrusted) network
   20. Keep all optional services
       1. Especially Wazuh (Will use for HIDS on DC1/PC1)
   21. Keep default Docker IP range
       1. Default is 172.16.0.0/16 (No LANs conflict)
   22. Setup email (Doesn’t have to be a real Email) and Password to access the Web Interface
   23. Select IP for web interface access
   24. Select ‘No’ for NTP Servers
       1. pfSense will be sync’d to which is already running ntp.
   25. Select ‘Yes’ for so-allow (Allows other devices to access Services on SecurityOnion)
       1. Enter DC1 address
3. Reboot

**Setting up Ubuntu Server (for Splunk):**

1. In Virtualbox Manager
   1. Change Network Adapter 1 to Internal Network for LAN3
2. Boot/Setup Ubuntu Server (Recommended by Splunk):
   1. Select Language (English)
   2. Accept default layout/variant options for Keyboard (English US)
   3. Select default Ubuntu Server
   4. Select enpos3 (NIC for LAN3)
      1. Select IPV4 Manual
         1. **Subnet: 192.168.150.0/24**
         2. **Address: 192.168.150.20**
         3. **Gateway: 192.168.150.1**
         4. **Name Servers: 192.168.50.101, 192.168.150.1**
            1. **DC1 as primary, pfSense LAN3 interface as secondary**
         5. **Search Domains: homelab.local (AD Domain)**
   5. Skip Proxy Address configuration
      1. Using pfSense to access the internet
   6. Keep default archive mirror configuration address
      1. Address used for packages/updates.
   7. Keep default storage configuration (Setup earlier: 50GB)
   8. **Profile Configuration:**
      1. **Select Name**
      2. **Server Name: ‘splunk’**
      3. **Username**
      4. **Password**
   9. Skip Ubuntu Pro
      1. Can be enabled later
   10. Install OpenSSH Server
   11. Skip Featured Server Snaps
       1. None are needed for this lab
3. Reboot
4. Install GUI
   1. First install tasksel (Task Selector)
      1. Allows simple installs of predefined packages (i.e., Ubuntu Desktop)
         1. ‘Sudo apt install tasksel’
   2. Install Ubuntu Desktop
      1. Sudo apt install ubuntu-desktop
5. Reboot server
   1. Sudo reboot
6. Login
   1. Should now see GUI Desktop

**Splunk Setup:**

1. Install Free Splunk Enterprise from [Splunk.com](http://splunk.com)
   1. Install as a tgz
2. Open Terminal and navigate to Downloads directory
   1. ‘Cd Downloads/’
3. Extract the contents of the file
   1. ‘Tar xvzf (filename)’
4. Change to Splunk Directory
   1. ‘Cd splunk’
5. Change to Splunk Bin directory
   1. ‘Cd bin’
6. Start Splunk Instance
   1. ‘./splunk start’
7. Use a spacebar to move through the General Terms.
8. Accept terms
9. Create Admin account
10. Access Splunk Console via http://splunk:8000
11. While in the /Downloads/splunk/bin directory, perform the following command:
    1. ‘Sudo ./splunk enable boot-start’
       1. Allows splunk to run on boot to prevent having to manually start splunk

**Add Splunk to so-allow list on SecOnion:**

1. Enter:
   1. Sudo so-allow
   2. Select ‘A’
   3. Enter 192.168.150.20

**Adding Splunk Universal Forwarders to Endpoints (DC1, PC1):**

Universal Forwarders are lightweight Splunk agents that collect/forward logs to a Splunk indexer

1. In Splunk Enterprise:
   1. Settings > Forwarding and receiving
   2. Configure receiving
   3. New Receiving Port
   4. Add the default listening port (9997)
2. Setup Splunk Index (in Splunk Enterprise):
   1. Settings > Indexes
   2. New Index
      1. Name: winevent\_log
      2. Leave other settings default besides:
         1. Tsidx Retention Policy: Enable
            1. 2 Day

Prevents large storage usage

1. On DC1
   1. Access [Splunk.com](http://splunk.com) > Navigate to Splunk Universal Forwarder download:
      1. Install for Windows Server 2022
   2. Run the .msi
   3. Setup Admin account
   4. Enter Deployment Server info:

Used to config Universal Forwarder

* + 1. Host/IP: 192.168.150.20
    2. Port: 8089 (Default Port)
  1. Enter Receiving Port info:
     1. Host/IP: 192.168.150.20
     2. Port: 9997(Default Port, same port previously configured in Splunk Enterprise)

1. On PC1:
   1. Access [Splunk.com](http://splunk.com) > Navigate to Splunk Universal Forwarder download:
      1. Install for Windows Server 2022
   2. Run the .msi
   3. Setup Admin account
   4. Enter Deployment Server info:

Used to config Universal Forwarder

* + 1. Host/IP: 192.168.150.20
    2. Port: 8089 (Default Port)
  1. Enter Receiving Port info:
     1. Host/IP: 192.168.150.20
     2. Port: 9997(Default Port, same port previously configured in Splunk Enterprise)

**Creating Forwarder Server and adding Endpoint Agents (Universal Forwarders):**

1. Add Splunk Forwarder Server (In Splunk Enterprise):
   1. Settings > Add Data
   2. Select Windows DC1 and PC1
   3. Name Windows Forwarder
   4. Select Source:
      1. Local Event Logs
         1. If this doesn’t show up:
            1. Settings > Add Data > Forwarder > Existing > Select Server Class.
      2. Select Application, Security, and System
         1. ForwardedEvents - Events forwarded from another Win Machine
         2. Setup - OS Setup/Installation Logs
            1. Both above Logs aren’t security critical/setup
   5. Select Index: winevent\_log

**Wazuh Setup (Security Onion):**

1. Enter ‘sudo so-allow’
2. Select ‘w’ for Wazuh Agent
   1. Opens TCP port 1514 for Agents to forward to
3. Enter ‘so-allow’
4. Select ‘r’ for Wazuh registration service
   1. Opens TCP port 1515 to allow Wazuh agent registration
5. Add Endpoints as Agents:
   1. Enter ‘sudo so-wazuh-agent-manage’
      1. Allows management of Wazuh Agents
   2. Add DC1 as an Agent:L
      1. Enter ‘a’ to Add Agent:
         1. Provide Name: DC1
         2. Enter IP: 192.168.50.101
   3. Add PC1 as an Agent:L
      1. Enter ‘a’ to Add Agent:
         1. Provide Name: PC1
         2. Enter IP: 192.168.50.11
6. Pull Authentication Key for Endpoints:
   1. Enter ‘e’
      1. Select Host via ID (i.e., ‘002’ for DC1)
         1. Do the same for PC1
   2. Note: Copy/paste the corresponding key into the notepad of the endpoint. They will be used for Agent registration/authentication.

**Register/Authenticate Wazuh Agents (On Endpoints):**

1. Access [Wazuh.com](http://wazuh.com)
   1. Locate the correct version supported by the security onion version
      1. Note: Older versions of Security Onion don’t support Wazuh 4.x, confirm with Security Onion version documentation
   2. Install for Windows
      1. Follow the prompts, before finishing check the box for running the configuration
2. Register/Authenticate DC1:
   1. Enter the IP Address of the Manager (Security Onion’s IP): 192.168.150.10
   2. Enter Authentication key for the endpoint
3. Repeat the above for PC1
4. Note: If the status shows stopped perform the following:
   1. Run Command Prompt as Admin:
      1. Enter ‘net start wazuh’
      2. Select Refresh on Wazuh Agent Manager Application
         1. It should show running now
   2. Alternatively, open Task Manager > Services:
      1. Locate ‘OssecSvc’
      2. Right-Click > Start

**Setup Splunk HEC:**

1. Create new Ingestion:
   1. In Splunk Enterprise:
      1. Settings > New Data > Monitor > HTTP Event Collector:
      2. Name the Collector
      3. Next
      4. Select Create a new Index (seconion\_hec)
      5. Review and finish
2. Make sure SSL is on for HEC
   1. Settings > Data Inputs > HEC > Global Settings > Check SSL box

**Setup Filebeat on Security Onion:**

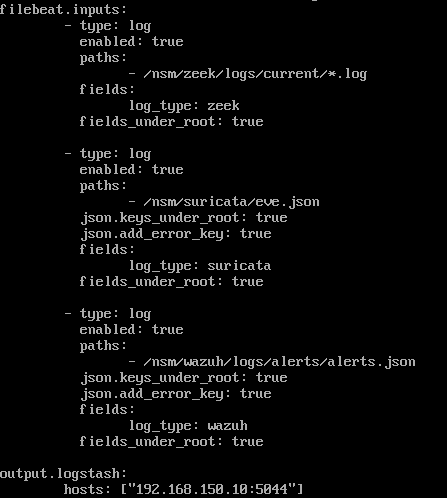
1. Configure Filebeat to Forward logs via Syslog to Splunk:

Filebeat: Tool used to forward specific logs to a destination, like splunk. (Not typically installed)

* 1. In Security Onion:
     1. Install Filebeat:
        1. Enter ‘curl -L -O https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-9.0.0-x86\_64.rpm’
           1. Downloads official filebeat package
        2. Enter ‘sudo rpm -i filebeat-9.0.0-x86\_64.rpm’
           1. Installs Filebeat via rpm package
        3. Enter ‘sudo systemctl start filebeat’
           1. Starts filebeat
        4. Enter ‘sudo systemctl enable filebeat’
           1. Enables Filebeat to start on boot
        5. Enter ‘sudo systemctl status filebeat’
           1. Look for ‘Running’ to confirm its on
  2. Next, change the //etc/filebeat/filebeat.yml (Config File) to forward Zeek, Suricata and Wazuh logs.
     1. Zeek: Network security monitoring tool that analyzes network traffic in real-time
        1. Collects: Network session logs (HTTP, DNS, SSL, FTP, etc…)
           1. Network connection metadata
           2. Protocol-specific events and suspicious behaviors
           3. Detects anomalies, scans, and traffic patterns
     2. Suricata: Network intrusion detection and prevention system that inspects network packets for signatures and anomalies
        1. Collects: Alerts triggered by known signatures
           1. Network flow data/metadata
           2. Protocol anomalies, suspicious packet payloads
     3. Wazuh: Host based intrusion detection system that monitors endpoints for security events
        1. Collects: Host logs
           1. Security alerts
           2. Configuration changes
           3. OSSEC-based rules for threat detection

Rules maintained by OSSEC/Wazuh Devs and local administrators.

* 1. Copy the below script in the screenshot below into //etc/filebeat/filebeat.yml and //etc/logstash/conf.d/(filename) respectively
  2. Restart filebeat:
     + 1. Enter ‘sudo systemctl restart filebeat’



Note: Whitespace is important in YAML, recreate exactly as seen.

Explanation:

1. Type: Log
   1. Tells filebeat the type is a Log (Any file that contains logs, including JSON)
2. Enabled: True
   1. Actives input and collection
3. Paths:
   1. Specifies the path
4. Fields:

Log\_type: (Example)

* + 1. Creates a field titled log\_type, allows Splunk to parse for that field type and find data collected from that specific tool

1. fields \_under\_root: true
   1. Puts custom field (Log\_type) to top level instead of being nested under fields:
      1. Example:
         1. Instead of field: { "fields": { "log\_type": "wazuh" } }
         2. It looks like: { "log\_type": "wazuh" }
2. Json.keys\_under\_root: true:
   1. Does the same as fields \_under\_root: true, just with json instead of fields
3. Json.add\_error\_key: true:
   1. Adds key if there is an issue with the logs, can help with log error troubleshooting
4. Codec: json\_lines:
   1. Sends each log event as a properly delimited JSON line
      1. Makes the logs human readable and easily parsable
5. Output.logstash

Used logstash (Data processing pipeline tool used to collect, process, and forward logs/events from sources to a destination)

* 1. Host: [“192.168.150.10:5044”]
     1. Sending to Security Onion port 5044 (Port Logstash listens on)
        1. Note: Using logstash because it supports a splunk\_hec plugin

**Setup Logstash for Security Onion:**

1. Add Elastic repo (Holds Logstash and other tools)
   1. Add the following into a shell script
      1. #!/bin/bash
      2. sudo rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch
      3. cat <<EOF | sudo tee /etc/yum.repos.d/elastic.repo
      4. [elastic-8.x]
      5. name=Elastic repository for 8.x packages
      6. baseurl=https://artifacts.elastic.co/packages/8.x/yum
      7. gpgcheck=1
      8. gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch
      9. enabled=1
      10. autorefresh=1
      11. type=rpm-md
      12. EOF
      13. sudo yum clean all
   2. Run the script
2. Install Logstash
   1. “sudo yum install logstash”
3. Start and enable Logstash on boot:
   1. “sudo systemctl start logstash”
   2. “sudo systemctl enable logstash”
4. Next create a docker container with a custom docker image of Logstash with the splunk\_hec plugin added
   1. “docker pull docker.elastic.co/logstash/logstash:8.18.2”
      1. Pulls the official image
   2. Create a Dockerfile in /root/logstash-docker/ named Dockerfile
      1. Will be used to build customer docker image with splunk\_hec
         1. “FROM docker.elastic.co/logstash/logstash:8.18.2”
         2. “RUN logstash-plugin install logstash-output-splunk\_hec”
   3. Build custom Logstash container image
      1. “Docker build -t custom-logstash:latest .”
   4. Create a ‘Config’ Folder in /root/logstash-docker/
   5. Create a file in the Config folder titled logstash.yml
      1. “Xpack.monitoring.enabled: false”
         1. Disabled Licensereader/checker
            1. If not disabled it will try and check the license in elasticsearch which hasn’t been setup.
   6. Create a ‘logstash.conf’ file in /root/logstash-docker/pipeline/
      1. See image below for script
   7. Run the following command as one: (Only run after logstash.conf is created)
      1. “docker run -d --name logstash --restart unless-stopped -v /root/logstash-docker/pipeline:/usr/share/logstash/pipeline -v /root/logstash-docker/config/logstash.yml:/usr/share/logstash/config/logstash.yml -p 5044:5044 custom-logstash:latest logstash -f /usr/share/logstash/pipeline/logstash.conf
         1. Command creates a docker container titled ‘logstash’ and boots on start unless stopped. Creates a logstash.conf file in /usr/share/logstash/pipeline within the container to use the conf and uses the logstash.yml file to disable lisensereader/checker (only works in .yml). Uses port 5044 and the custom-logstash image we created earlier. Final command just forces the use of the logstash.conf file.
5. Check Splunk HEC (seconion\_hec)
   1. Should be receiving logs now.



1. Input { beats { port => 5044}}
   1. Sets logstash to listen on port 5044 for data from Filebeat
2. Filter:
   1. Json { source/target}:
      1. Filters the input from Filebeat into json format and changes the name of the message
   2. Mutate { rename}:
      1. Renames parsed\_json to message again
3. Output:
   1. Stdout { codec => rubydebug }
      1. Used to debug issues with Logstash
         1. Logstash used JRuby
            1. JRuby has compatibility issues with the Security Onion iso / Linux library version I have, thus the need to use a docker instance to isolate the library.
   2. Splunk\_hec:
      1. Sends processed data to Splunk HEC
   3. Hec\_host:
      1. Specifies host
   4. Port:
      1. Default port for HEC is 8088
   5. Hec\_token:
      1. Token for the HEC configured
         1. Found in Settings > Data Inputs > HEC > Show Token
   6. Index:
      1. Sending to configured index
   7. Sourcetype => “\_json”
      1. Specifies to send in json (Which is default behavior for HEC anyway)

**Cyberattack/Pentesting Labs:**

**Utilizing the MITRE Att&CK Framework:**

**Reconnaissance:**

1. Scan for hosts
   1. nmap -sn 192.168.100.0/24
      1. Reveals 3 hosts.
         1. 192.168.50.1 (pfSense.local.lan)
         2. 192.168.50.11 (PC1)
         3. 192.168.50.101 (DC1)
2. Scan for open Ports:
   1. nmap -sS -Pn -p- -T4 192.168.50.0/24 -oN full-port-scan.txt
      1. pfSense:
         1. 53 (DNS)
         2. 80 (HTTP)
      2. PC1:
         1. 135 (msrpc)
         2. 139 (netbios-ssn)
         3. 445 (microsoft-ds)
         4. 3389 (ms-wbt-server)
         5. 5040 (Windows Deployment Services Server)
         6. 7680 (pando-pub)
         7. 49671 (Dynamic Port)
      3. DC1:
         1. 53 (DNS)
         2. 88 (kerberos-sec)
         3. 135 (msrpc)
         4. 139 (netbios-ssn)
         5. 389 (ldap)
         6. 445 (microsoft-ds)
         7. 464 (kpassw5)
         8. 593 (http-rpc-epmap)
         9. 3268 (globalcatLDAP)
         10. 5357 (wsdapi)
         11. 5985 (wsman)
         12. 9389 (Active Directory Web Services/ADWS)
         13. 49669, 49673, 49674, 49686, 49690, 61248 (Dynamic Ports)
   2. Scan Common Exploitable Ports for Versions:
      1. nmap -sV -p22,80,135,139,445,3389,8000 192.168.50.X -oN version-scan.txt
         1. pfSense:
            1. 80 (HTTP) - Version (nginx)
         2. PC1:
            1. 135 (msrpc) - Version (Microsoft Windows RPC)
            2. 139 (netbios-ssn) - Version (Microsoft Windows netbios-ssn)
            3. 445 (microsoft-ds?) - Version (N/A)
            4. 3389 (ms-wbt-server) - Version (Microsoft Terminal Services)
         3. DC1:
            1. 135 (msrpc) - Version (Microsoft Windows RPC)
            2. 139 (netbios-ssn) - Version (Microsoft Windows netbios-ssn)
            3. 445 (microsoft-ds?) - Version (N/A)

**Installing a Vulnerability Scanner (Nessus):**

1. On Splunk (Ubunu Server):
   1. Go to Tenables Nessus download page and install for the specific distribution/architecture in use
   2. Run the Curl command shown for Linux
   3. Unpack via dpkg -i (filename)
      1. -i Installs the file
   4. Start the service: “systemctl start nessusd”
   5. Enable to run on boot “systemctl enable nessusd”

**Nessus Configuration:**

1. Access Nessus via **https://192.168.150.20:8834**
2. Register with Nessus Essentials
3. Register via email
4. Create Admin Account
5. Download Plugins
   1. May 5 - 10 Mins
6. Eventually a “Welcome to Nessus” window will appear prompts for targets
   1. Enter 192.168.50.0/24 to discover all potential targets within the victim subnet
      1. Select DC1/PC1
7. Nessus will then run a Basic Network Scan

**DC1 Email Service Setup (hMailServer):**

DNS Manager > Add DNS A/MX Records in DNS Manager > Right Click homelab.local > Add A record, Name=mail, IP address=DC1 (192.168.50.101), check Create associated PTR record (Used for reverse DNS lookup, acts as anti-spam mechanism) > Add MX Record > Right click homelab.local > Select MX Record > Leave “Host or child domain blank” > Set FQDN to mail.homelab.local or use browse to select A record for mail, leave priority default (10).

**Verify MX Record via NSLookup:**

Nslookup via CMD > “set type=MX” > “homelab.local”. Should show mail exchanger and IP.

**Install (**[**https://www.hmailserver.com/download**](https://www.hmailserver.com/download)**)**

> Next > Accept Terms > Next (4x) > Set admin password > Install > Prompts for Microsoft .NET Framework 2.0 > If it can’t install, manually install .NET Framework 2.0 via Server Manager: Server Manager > Add Roles and Features > Features > .NET Framework 3.5 (Included .NET 2.0 and 3.0) > Launch hMailServer Admin > Connect with hMailServer admin password > Add Domain > Enter mail.homelab.local > Right-Click Accounts > Add AD Account > Select Bill (Should autopopulate settings.

**Add and Allow SMTP(25) and IMAP(143) to DC1’s Incoming Firewall Rules**

> Firewall > Rules > Add > Specify SMTP and port 25 > Add > Specify IMAP (143).

* Can test using the Powershell command “Test-NetConnection -ComputerName 192.168.50.101 [25/143]” if Status comes back as “True” then it works.

**Install Thunderbird (Easy to setup for IMAP/SMTP)**

* Outlook is very finicky to setup because it forces Autodiscovery via cloud regardless of the setup type being used.

Next > Standard Install > Next > Install > Manual Setup > Enter: Name: Bill, Email:Bill@mail.homelab.local>Password:(Domain Password) > AutoDetect > Continue > Continue > Success

**Add Domains to PfSense:**

DC1 > PfSense WebConfig > Status > DNS Resolver > Add both domains and IPs to Host overrider > Save Config.

**Install Postfix:**

“sudo apt install postfix mailutils -y”

Select Internet Site

Provide Host Name for the Mail Server: “postfix.lab.local”

Add the following to the /etc/postfix/[main.cf](http://main.cf) file:  
“

myhostname = postfix.lab.local

mydomain = lab.local

myorigin = $mydomain

inet\_interfaces = all

inet\_protocols = ipv4

mydestination = $myhostname, localhost.$mydomain, localhost

“

Restart Postfix and Enable (To run on Boot):

“sudo systemctl restart postfix”

“sudo systemctl enable postfix”

Send a Test Email:  
“mail -s "Lab Test" [Bill@mail.homelab.local](mailto:Bill@mail.homelab.local)”

Check ThunderBird on the Windows PC and confirm its received

**Setup GoPhish:**

“gophish” in Kali terminal should launch a webpage

Login with default creds provided via the terminal (Password may vary based on version)

Set new Password

**Setup Sending Profiles for Test Email:**

Select “Sending Profiles”

Select New Profile

Enter Postfix Test

SMTP From (What recipient sees): [john@mc.com](mailto:john@mc.com)

Host: 127.0.0.1

Click Send Test Email

Provide [Bill@mail.homelab.local](mailto:Bill@mail.homelab.local)

Confirm Bill’s mailbox received the email