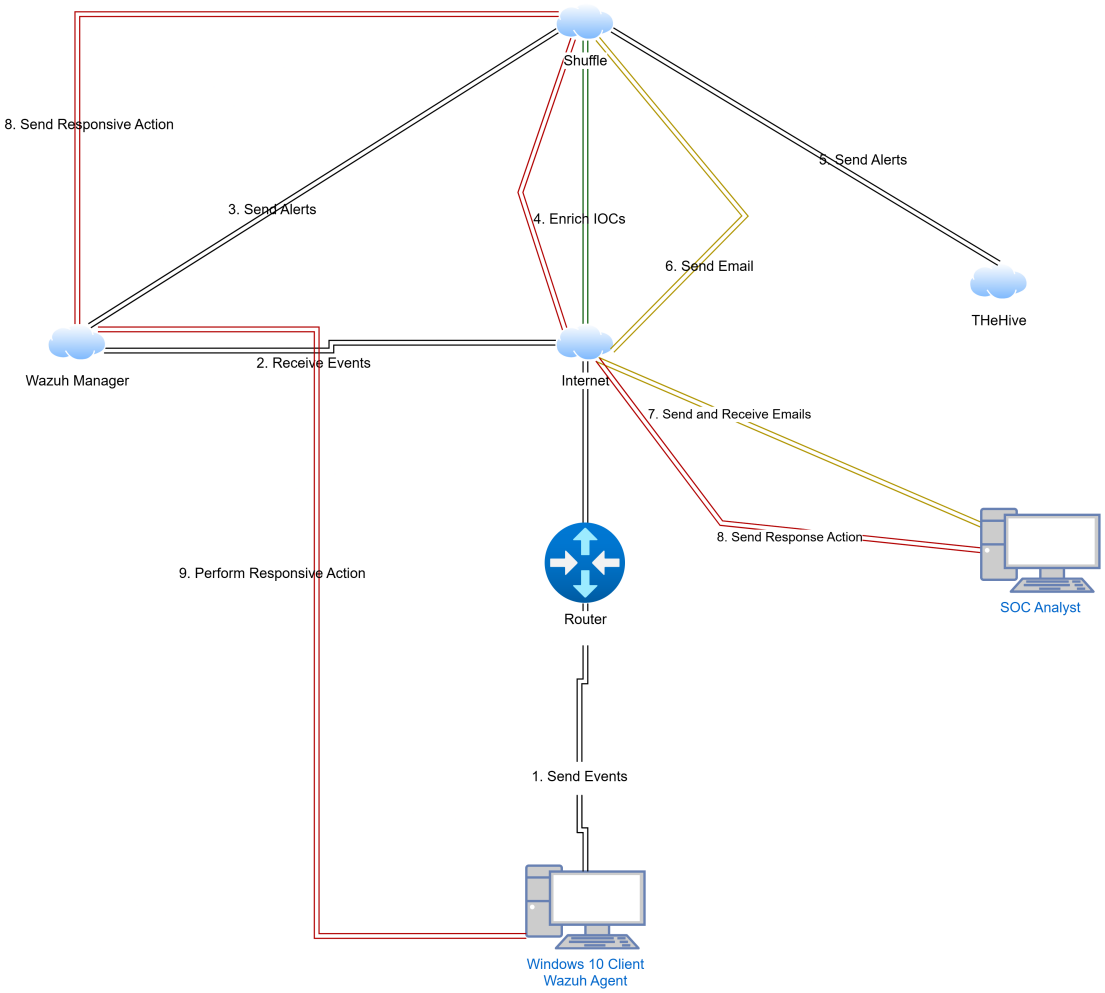
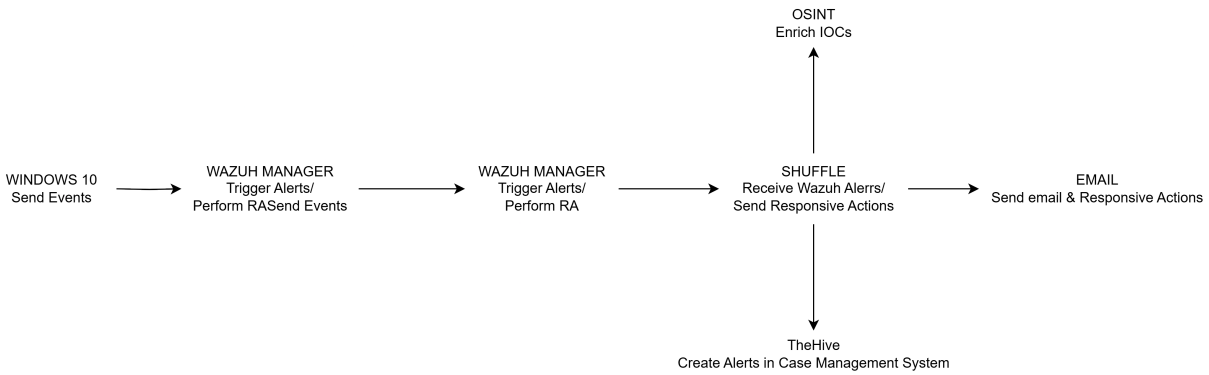
DETECTION ENGINEERING LAB PART 1

NETWORK DIAGRAM





INSTALLING SERVICES & APPLICATIONS

Wazuh

For this we are going to use the OVA file to make things easier

Windows 11

Get the official eval ISO file from the microsoft website .

In this lab I will be setting just the client alongside wazuh for some monitoring. Building step by step before getting to the final goal as on the network diagram.

Objective

This lab will focus on setting up **Wazuh** to detect, analyze, and respond to security threats in a **Windows 11 environment**. The goal is to create custom detection rules, analyze logs, and simulate attacks to evaluate detection efficiency. Serves as place where I put into practice what I've learnt from platforms like LetsDefend and HTB.

Prerequisites

- Windows 11 (fresh installation or VM)
- Wazuh OVA file (already loaded into your vm)
- Sysmon (for detailed event logging)
- A testing framework (will be using Atomic Red Team and custom scripts for simulating attacks)

But what is Detection Engineering ?

Detection engineering is all about developing a systematic, repeatable process to build a suite of detection rules for your SIEM, enabling you to gain real-time visibility into potential malicious activity.

What Is Detection Engineering?

It’s the practice of creating, deploying, and tuning threat detection rules that identify specific patterns or behaviors indicating potential malicious activity. These rules run in your SIEM, helping you catch threats as they occur. The goal here is to provide actionable, real-time alerts by monitoring critical log sources—even if those logs initially have no prebuilt detections—and then continuously refining the rules to minimize noise and false positives.

Key Phases of the Detection Engineering Process

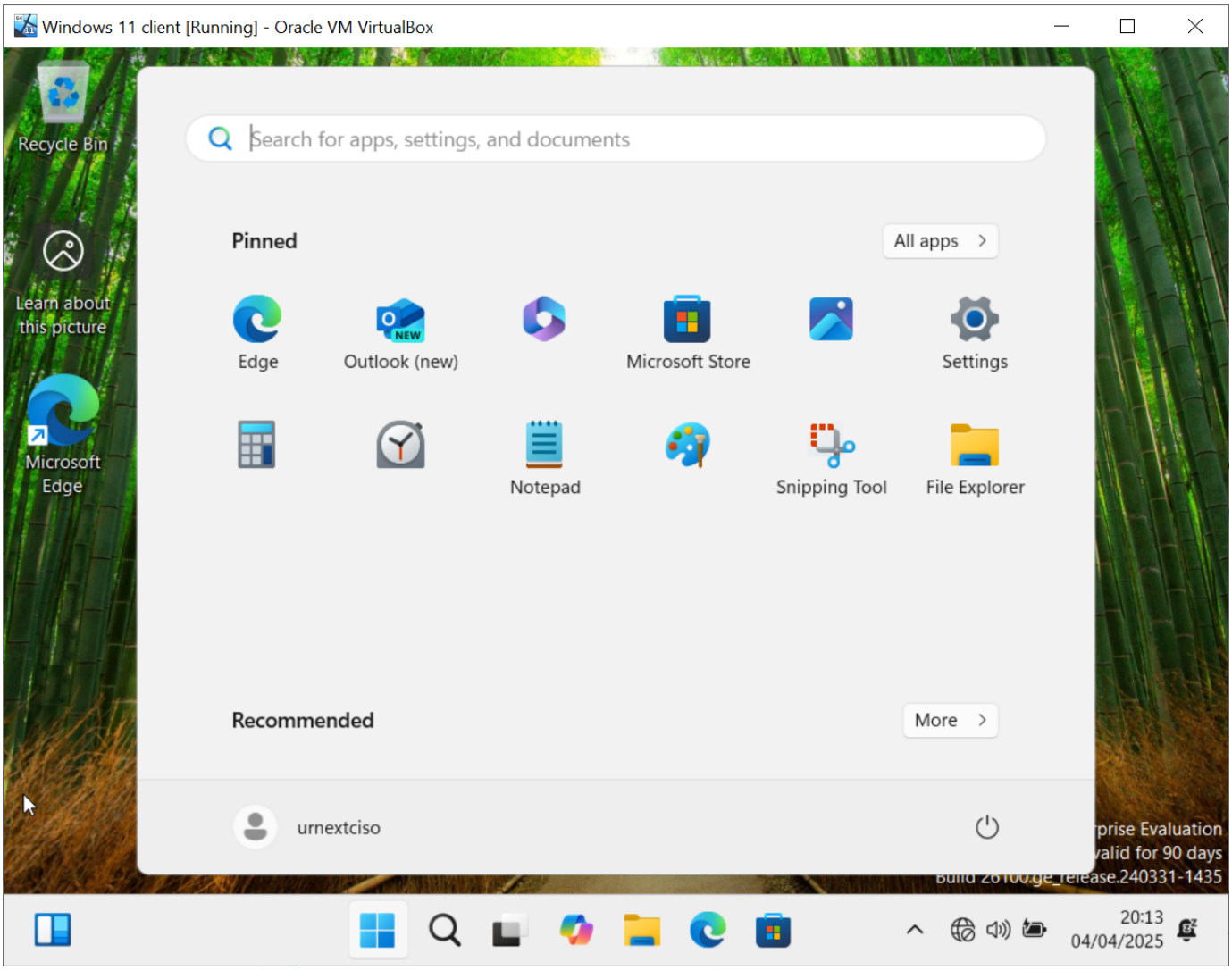
1. **Research Phase:**
   - **Understand the Log Source:**
     Dive into the documentation and ask stakeholders (admins, IT, and end-users) about the system's regular activities and potential attack surfaces.
   - **Key Questions to Ask:**
     - What events are critical to business operations?
     - What legitimate actions could also be abused by an attacker?
     - What sensitive data or configurations need monitoring?
2. **Detection Brainstorming:**
   - **Idea Generation:**
     Write down all potential detections you can think of.
   - **Focus on Dual-Nature Activities:**
     Consider actions that are normal but could be malicious in the wrong context (e.g., administrative commands, configuration changes, or data exfiltration behaviors).
3. **Diving into the Logs:**
   - **Baseline Establishment:**
     Ingest at least a week's worth of logs to identify what "normal" looks like.
   - **Count By Queries:**
     Use aggregation (like counting event occurrences) to pinpoint anomalies or rare events that might indicate a threat.
4. **Detection Organization:**
   - **Grouping Logic:**
     Organize your detections by event types, actions, or similar threat behaviors to reduce redundancy and simplify management.
   - **Consider Exceptions:**
     Identify trusted sources or known benign behaviors that might otherwise trigger alerts and plan for exceptions.
5. **Detection Creation:**
   - **Rule Writing:**
     Write clear and concise rules that capture the behavior you want to monitor.
   - **Types of Detections:**
     - **True/False:** A simple presence check for a key-value pair.
     - **Threshold:** Alerts trigger when a certain volume of events is reached within a set time frame.
     - **Scheduled:** Regular scans based on fixed intervals.
6. **Validation and Testing:**
   - **Simulate Scenarios:**
     Validate your rules by testing against sample logs that represent both "success" (should trigger) and "failure" (should not trigger) cases.
   - **Edge Cases:**
     Test boundary conditions to ensure accuracy.
7. **Iteration:**

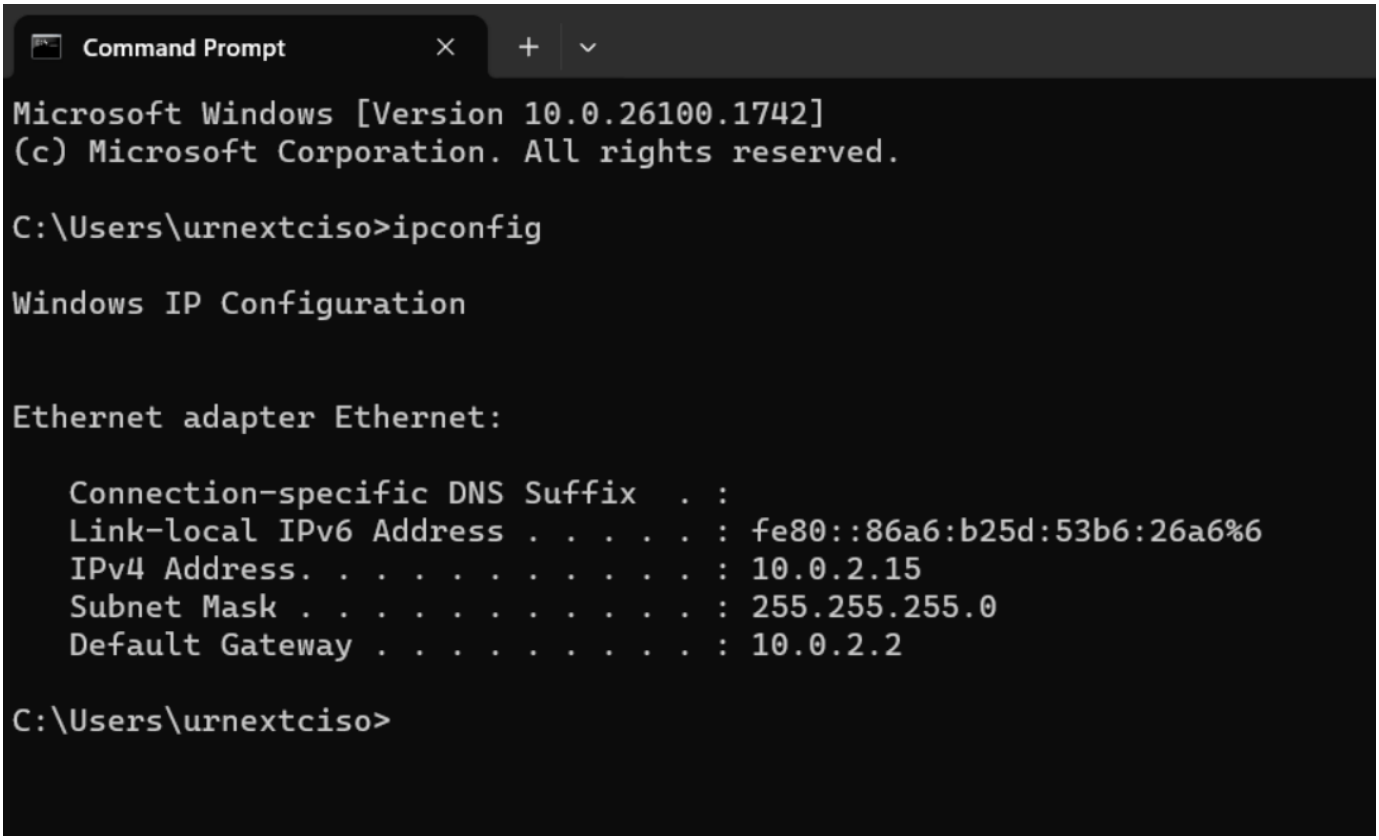
- **Continuous Improvement:**
  Monitor the alerts over time, adjust thresholds, and refine exceptions based on feedback and evolving threat landscapes.
- **Review and Update:**
  Regularly revisit your detection suite to incorporate new threat intelligence and remove noise.

What to Look For in Detection Engineering

- **Clarity & Simplicity:**
  Rules should be easy to understand and maintain. They must clearly differentiate between benign and suspicious activity.
- **Contextual Accuracy:**
  Consider the environment and typical user behavior. Focus on the context—what's normal for one environment might be suspicious in another.
- **Balance Between Coverage and Noise:**
  Strive for broad coverage of potential threats while reducing false positives. It's better to start with more alerts and refine than to miss critical events.
- **Proven, Repeatable Process:**
  A mature detection engineer uses a repeatable process (as outlined above) that not only builds detections effectively but also evolves them as the threat landscape changes.
- **Use of Existing Resources:**
  Look at open-source detection libraries (e.g., from Panther, Splunk, or community GitHub repositories) to get inspiration or validate your logic.

So now we have our windows workstation installed . If you have problems with the sign in check out this blog to solve it .:


Windows 11 client [Running] - Oracle VM VirtualBox
Recycle Bin
Learn about this picture
Microsoft Edge
Search for apps, settings, and documents
Pinned
All apps
Edge
Outlook (new)
Microsoft Store
Settings
Notepad
Snipping Tool
File Explorer
Recommended
More
urnextciso
prise Evaluation
valid for 90 days
Build 26100.ge_release.240331-1435
20:13
04/04/2025

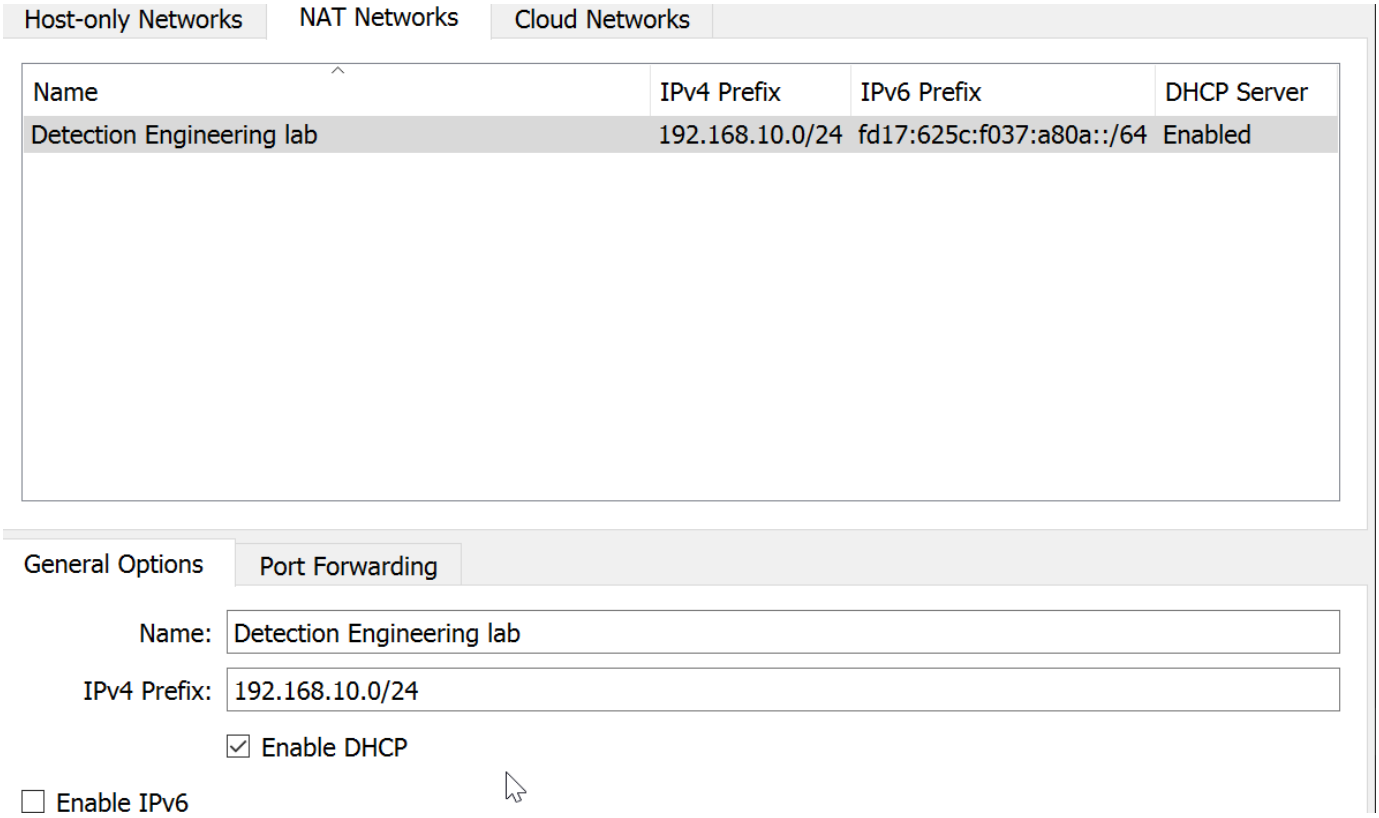

Command Prompt
Microsoft Windows [Version 10.0.26100.1742]
(c) Microsoft Corporation. All rights reserved.
C:\Users\urnextciso>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::86a6:b25d:53b6:26a6%6
IPv4 Address. . . . . : 10.0.2.15
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.2.2
C:\Users\urnextciso>

Since we are going by default Vbox installations with no firewall installed to segment the network both the PC and the Wazuh manager will be on the same network. So knowing our network in which we are working in , we can then access the wazuh server and set the ip to static .

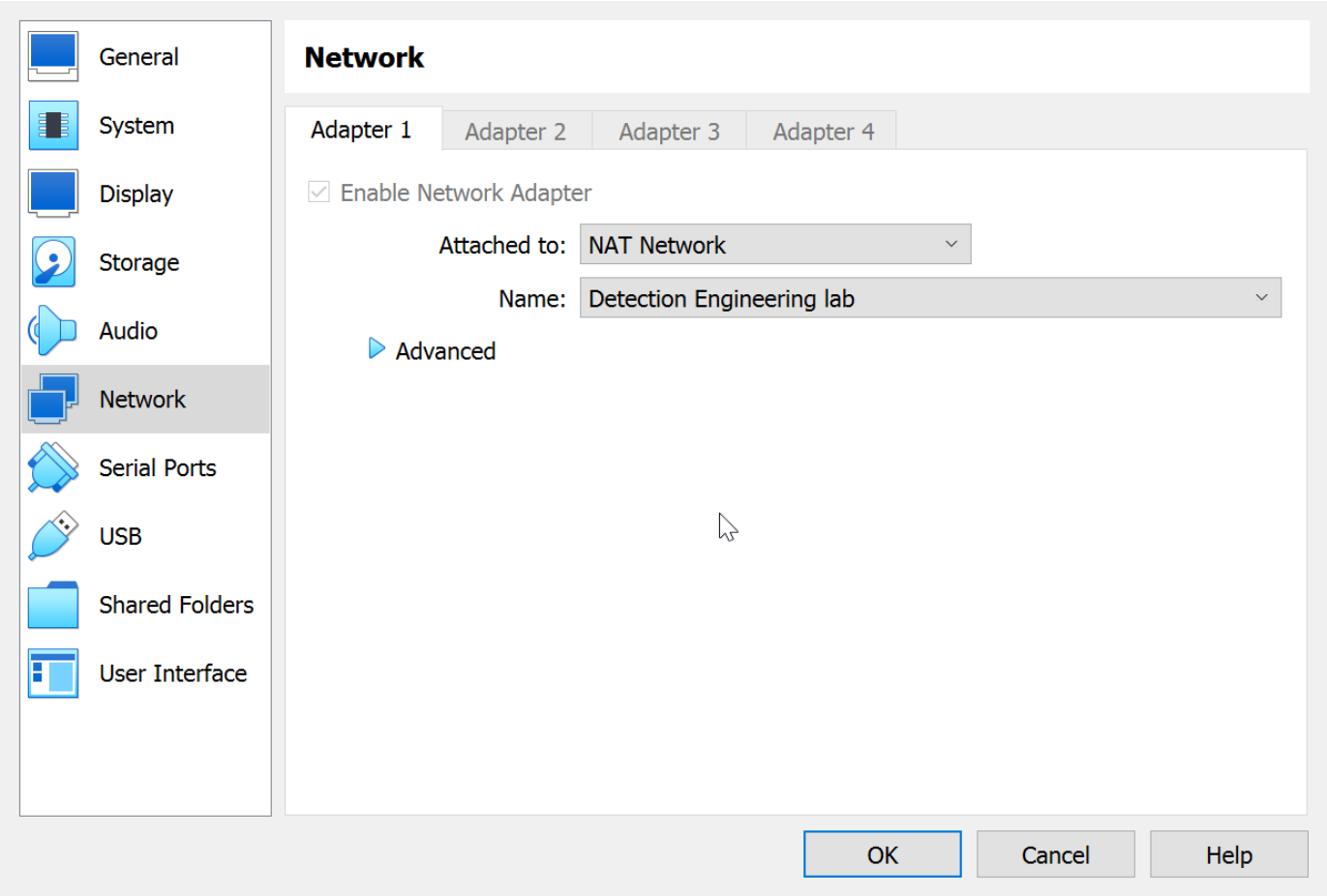
To access the server , first boot it up and use the default logins :

user: wazuh-user
password: wazuh

Moving forward to make things easier, we can now create a NAT network which our lab will pull addresses from , before setting the Static IPV4 address of our wazuh server. It looks something like this :



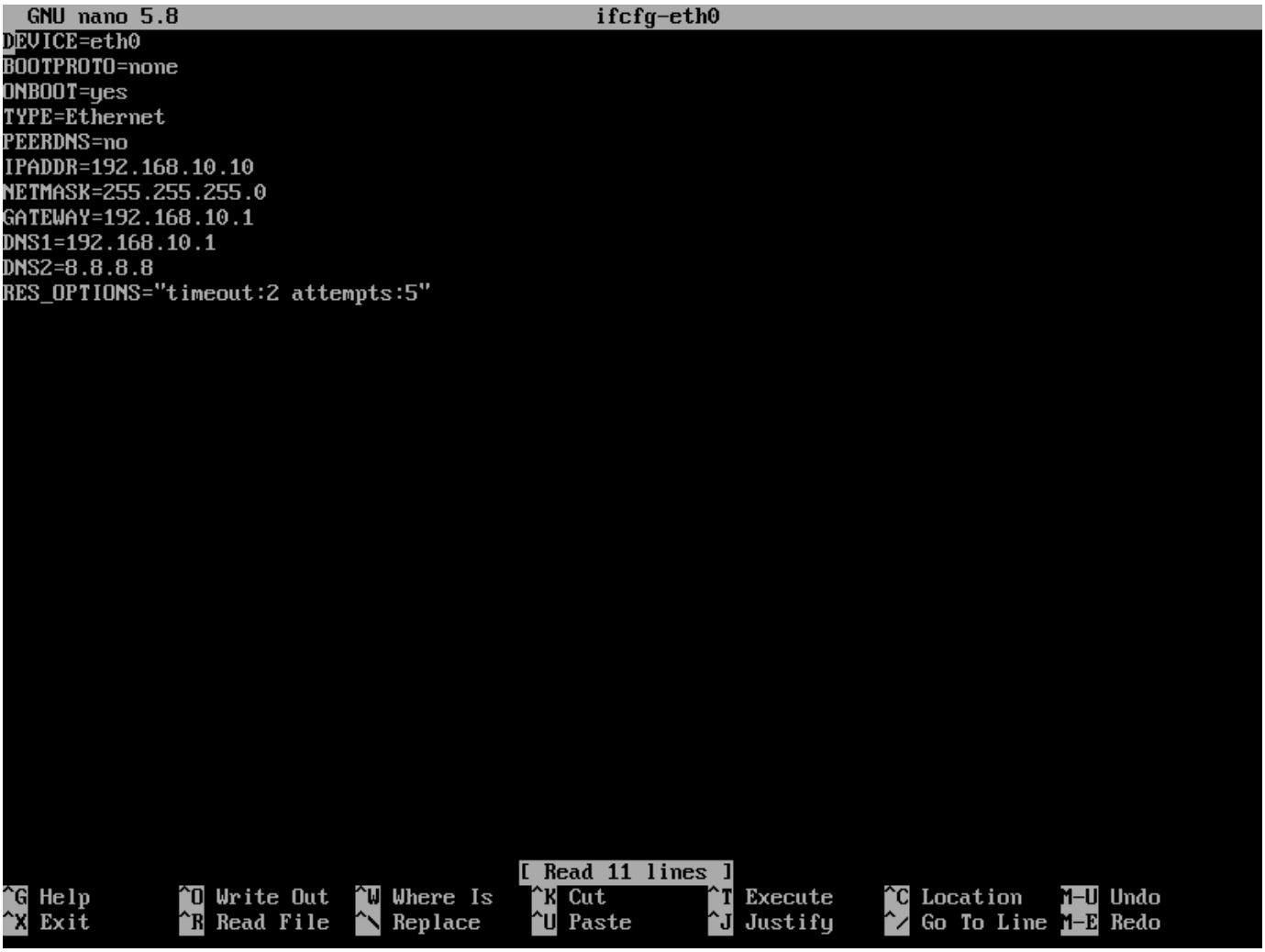
Set the Network settings of the Various hosts to NAT NETWORKS , like so :



Then assign a static IP to the wazuh server by editing the config file located at /etc/sysconfig/network-scripts/ifcfg-eth0 , from this :

```
[wazuh-user@wazuh-server network-scripts]$ cat ifcfg-eth0
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Ethernet
USERCTL=yes
PEERDNS=yes
DHCPV6C=yes
DHCPV6C_OPTIONS=-nw
PERSISTENT_DHCLIENT=yes
RES_OPTIONS="timeout:2 attempts:5"
[wazuh-user@wazuh-server network-scripts]$
```

To this :



Shortly after starting the VM, the Wazuh dashboard can be accessed from the web interface by using the following credentials:

> URL: https://<wazuh_server_ip>
> user: admin
> password: admin

And that will be all for this first part.

When designing this I was thinking of how i'd love to learn and what might be both excited/boring (sometimes).

What's Next ?

To be able to see what's going on during the different phases of our attacks, we will install the Wazuh agent and then pick up with the following scenarios :

1. Malware Simulation & Active Response

- **Objective:** Detect the creation or modification of a file that mimics malware behavior (e.g., an EICAR test file or a benign “malicious” file) in a critical directory (such as Downloads or Desktop).

- **Key Components:** File Integrity Monitoring (FIM), VirusTotal integration, and an Active Response script to automatically remove the file.
- **Technical Focus:** Configure Wazuh to monitor specific directories and trigger custom rules when a suspicious file is detected, then execute an active response to remediate.

2. Suspicious PowerShell Activity Monitoring

- **Objective:** Identify and alert on potentially malicious PowerShell commands that might be used to obfuscate actions (e.g., Base64-encoded commands or unusual command-line parameters).
- **Key Components:** Wazuh's log collection from the PowerShell event channel, custom rules for detecting encoded commands or anomalous usage.
- **Technical Focus:** Enable PowerShell Script Block Logging and configure the Wazuh agent to forward PowerShell events. Write custom rules that look for specific indicators (e.g., unusual flags or Base64 strings).

3. Lateral Movement & Network Anomaly Detection

- **Objective:** Simulate lateral movement techniques (such as remote management commands, abnormal network connections, or scheduled task manipulations) and detect them.
- **Key Components:** Sysmon for detailed process and network logging, Wazuh's event correlation, and custom rules to identify atypical network behavior or process executions.
- **Technical Focus:** Configure Sysmon with an enhanced ruleset, forward its logs via the Wazuh agent, and create correlation rules that flag unusual remote command executions or network traffic patterns.

4. Persistence Mechanism & Registry Monitoring

- **Objective:** Detect unauthorized changes to registry keys or startup folders that adversaries might use to establish persistence.
- **Key Components:** Windows Registry monitoring and FIM targeting critical areas like the "Run" and "RunOnce" registry keys or startup directories.
- **Technical Focus:** Ensure the Wazuh agent is configured to monitor key registry hives and the startup folder. Develop custom rules to alert on unexpected modifications.