National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Atharva Dhumal
Student ID: x23404388

School of Computing
National College of Ireland

Supervisor: Dr. Mosab Hamdan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Atharva Dhumal
Student ID:	X23404388
Programme:	MSc Cybersecurity **Year:** 2024/25
Module:	Practicum Part 2
Lecturer:	Dr. Mosab Hamdan
Submission Due Date:	11/08/2025
Project Title:	Design and Evaluation of Zero Trust Strategies for Ransomware Containment: A Focus on Strict Verification and Monitoring
Word Count:	1394 **Page Count:** 11

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Atharva Dhumal

Date: 10/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Atharva Dhumal
Student ID: x23404388

1 Introduction

Zero Trust Architecture (ZTA) is a cybersecurity paradigm that eliminates implicit trust within network boundaries and replaces it with a never-ending verification of all resource access, whether that is by users or by devices. According to NIST, Zero Trust can be defined as a direction in which there is no explicit trust directly placed on assets or user accounts based only their physical geographical location or network location. ZTA in practice focuses on strict authentications, excessive access right with least-privilege, and continuous observation of system behaviours. Ransomware defence is one such crucial application as ransomware usually leaves numerous copies of the file encrypted with the deletion of the original, a comprehensive file and process monitoring should be able to pick up on such behaviour pattern. File integrity monitoring in Wazuh, as an example, can highlight such an excessive creation/deletion of files that defines the characteristics of a ransomware attack. The configuration manual describes an experimental Zero Trust testbed based on VirtualBox, Ubuntu 22.04 LTS (victim, Linux distribution), Kali Linux (attacker, Linux distribution) and a set of security tools (Wazuh, auditd, UFW, OpenSSH, etc) to implement monitoring, tight access control, and ransomware containment. The aim is to bring out a step by step guidable that can be documented academically on such a set-up.

2 Configurations

Component	Purpose in Setup	Configuration Steps
Oracle VirtualBox 7.1.12	Hypervisor to generate and operate isolated virtual machines (VMs).	1. Available at the official site of Oracle. 2. Installer Win - use a .exe Installer. 3. Add user to vboxusers group for USB/VM access.
Ubuntu 22.04 LTS (Victim VM)	Wazuh Manager, auditd and UFW is deployed in the following monitored system.	1. Establish VM in VirtualBox (2–4 GB RAM, 2 CPUs, 20+ GB storage). 2. Install OS, by attaching Ubuntu ISO. 3. Update packages.
Kali Linux (Attacker VM)	Kali Linux allows creating an environment of an attacker to test ransomware.	1. Establish VM in VirtualBox (2 Gb RAM 2 CPUs). 2. Install OS, by attaching Kali ISO. 3. Configure Host-only/Internal network to connect to Ubuntu VM.
Static IP Configuration	High risk of IP conflicting amongst VMs. Static IP Configuration Provides consistent network addresses between VMs.	Set each VM with fixed IP on network setting (Host-only/Internal network).
Wazuh Manager + Dashboard	Wazuh Manager + Dashboard SIEM/XDR platform to track audit logs and find anomalies.	1. Add Wazuh repo & GPG key. 2. Install manager & dashboard. 3. Enable & start services.
auditd	The kernel-level logging of	1. Install auditd.

	commands, accesses to files, system calls.	2. Install additional Audit Policies: propel Audit Policies: command exec, sudoers file, and sensitive dirs. 3. Load rules.
Wazuh–auditd Integration	Wazuh-auditd Integration Enables Wazuh to be able to parse audit logs in real time in order to provide alerts.	1. Go to the editing /var/ossec/etc/ossec.conf with audit log. 2. Restart Wazuh agent/manager.
UFW (Uncomplicated Firewall)	UFW (Uncomplicated Firewall) Performs micro-segmentation and limits access to networks.	1. Allow outgoing, set default deny of incoming. 2. Only Kali should be permitted to SSH. 3. Enable firewall.
OpenSSH Server	OpenSSH Server Allows remote secure management across VMs.	1. Install the OpenSSH client and server. 2. Enabling SSH service.
Fake Ransomware (Python)	Fake Ransomware (Python) mimics the ransomware encryption processes used to test detection.	1. Write the script to change/encrypt files. 2. Install in Kali to Ubuntu using SCP / SSH. 3. Perform alert operation and supervise.
htop	This monitors system performance as it is under attack.	Install and use htop when testing attacks.

3 Implementation

Implementation process is sequential and structured to establish, configure and test the Zero Trust Architecture (ZTA) laboratory environment. This gives a replicable, understandable, way to check every step with the visual and technical evidence. Image captures and screenshots are needed to record configuration state at every point and are useful in the later assessment.

STEP 1: Set Up the Virtual Environment

1.1 Install VirtualBox (or VMware) [1]

- Install **VirtualBox** here https://www.virtualbox.org/
- Download ISOs:
  - Ubuntu 22.04 LTS [2]
  - Kali Linux (latest) [3]

1.2 Create 2 VMs

- VM1: Ubuntu (Victim)
- VM2: Kali (Attacker)

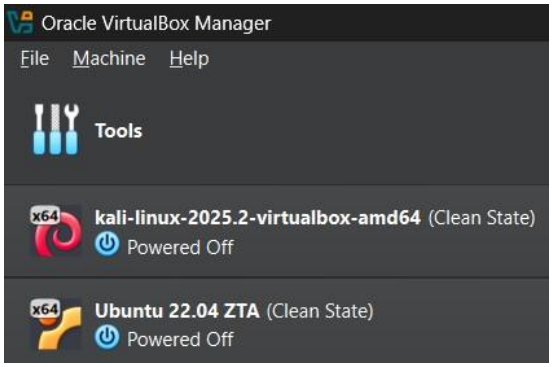

Figure 1. VirtualBox with both VM installed

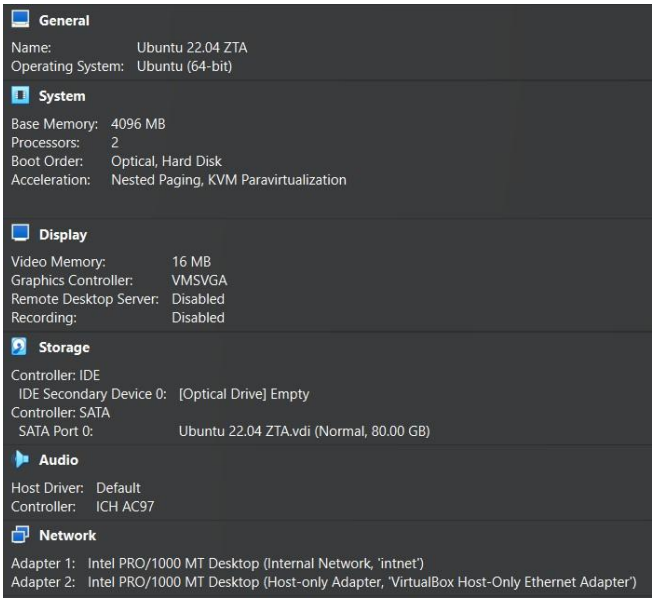

Figure 2. VM settings for Ubuntu

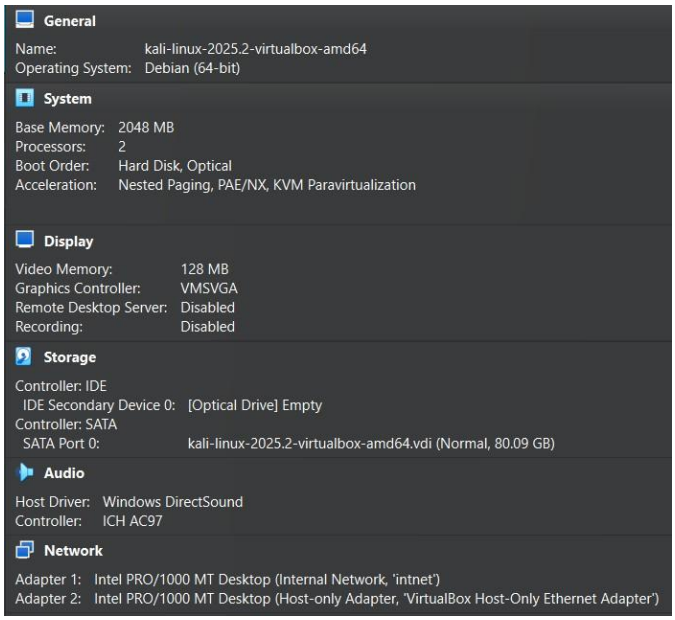

Figure 3. VM settings for Kali

- Set **Internal Network** on both VMs so they are isolated but can talk to each other:
  - Go to Settings > Network > Adapter 1 > Attached to: Internal Network
  - Name: ZTA-Net

1.3 Assign Static IPs

On VM1:

sudo nano /etc/netplan/01-netcfg.yaml

```
network:
  version: 2
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.56.101/24]
```

Apply:

sudo netplan apply

```
rookie1@rookie1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:99:3f:5b brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 86349sec preferred_lft 86349sec
    inet6 fd17:625c:f037:2:a014:2cb6:53ea:e526/64 scope global temporary dynamic
       valid_lft 86392sec preferred_lft 14392sec
    inet6 fd17:625c:f037:2:3eb7:7048:3da5:928e/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86392sec preferred_lft 14392sec
    inet6 fe80::861e:7d22:1854:26fb/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:9a:64:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/24 brd 192.168.56.255 scope global noprefixroute enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9a:6436/64 scope link
       valid_lft forever preferred_lft forever
rookie1@rookie1:~$
```

Figure 4. Static Ip assigned to Ubuntu

Do the same on VM2, with IP 192.168.56.102

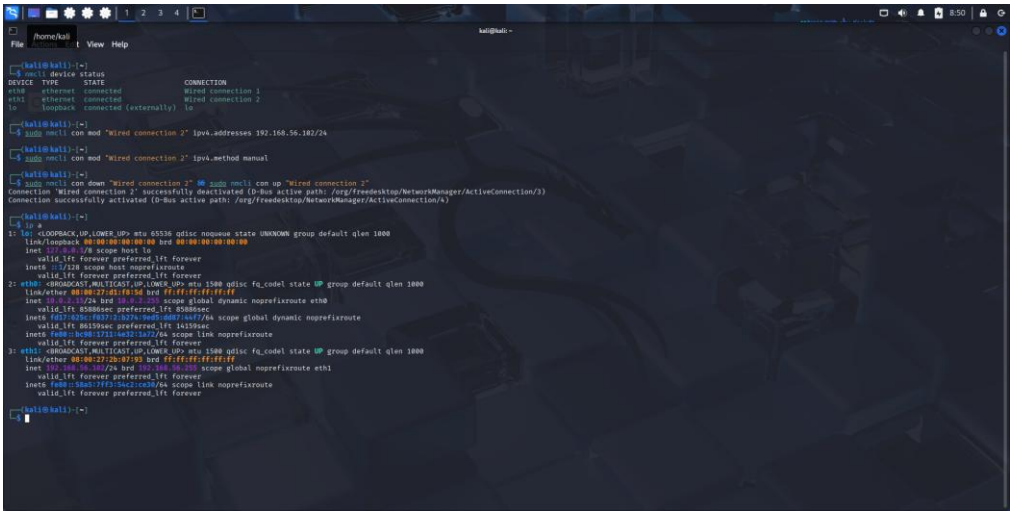

Figure 5. Static Ip assigned to Kali

STEP 2: Set Up Micro-Segmentation

2.1 On Ubuntu (VM1):
Install firewall: [6]
sudo apt update
sudo apt install ufw

2.2 Deny all incoming traffic:
sudo ufw default deny incoming
sudo ufw allow from 192.168.56.102 to any port 22 (Allow only attacker access via SSH)
sudo ufw allow out
sudo ufw enable

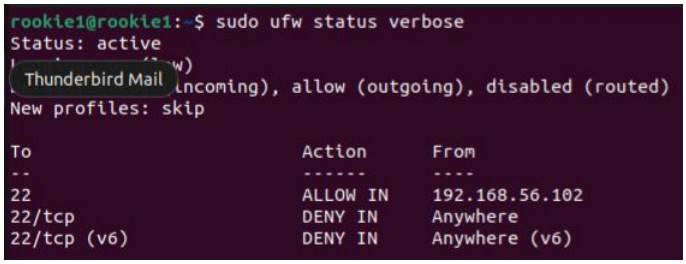

Figure 6. ufw status verbose

STEP 3: Set Up Strict Verification
3.1 Create New User with Limited Privileges:
sudo adduser testuser
sudo usermod -L testuser # Lock the password
sudo usermod -s /usr/sbin/nologin testuser # Disable shell access
3.2 Optional: Limit file access
sudo chmod 700 /home/testuser

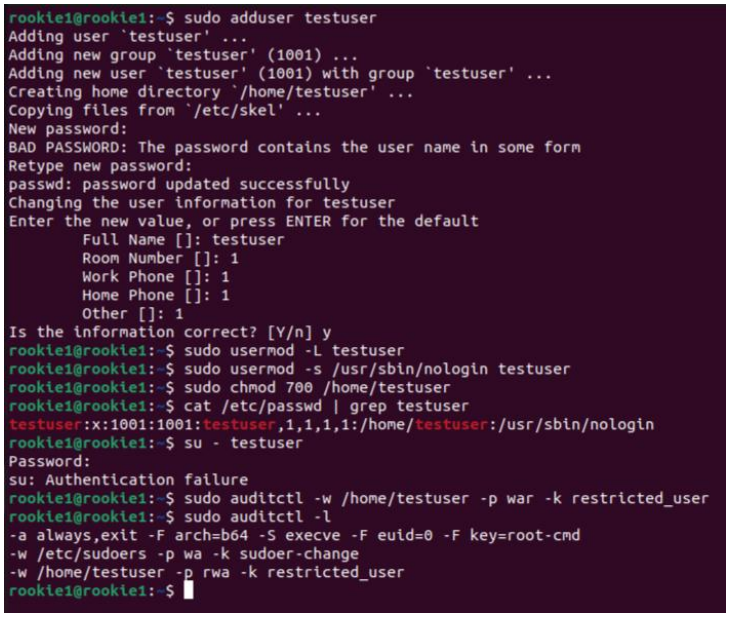

Figure 7. Testuser added

STEP 4: Install and Configure Wazuh [4]
4.1 On VM1 (Ubuntu - Victim):

Install Wazuh agent:
curl -s https://packages.wazuh.com/key/GPG-KEY-WAZUH | gpg --dearmor -o /usr/share/keyrings/wazuh.gpg
echo "deb [signed-by=/usr/share/keyrings/wazuh.gpg] https://packages.wazuh.com/4.x/apt/ stable main" | sudo tee /etc/apt/sources.list.d/wazuh.list
sudo apt update
sudo apt install wazuh-manager -y

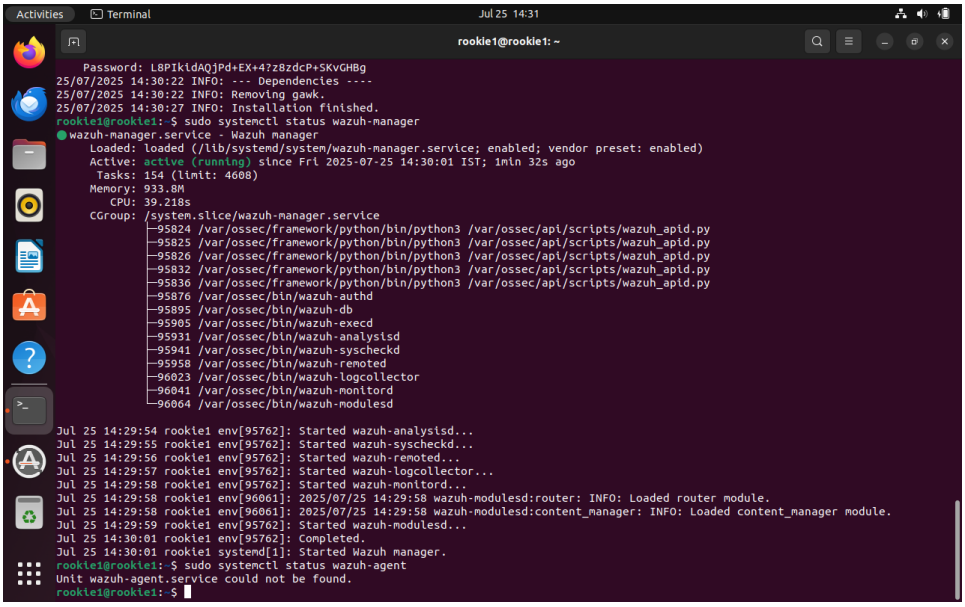

Figure 8. Wazuh status

4.2 Start Wazuh:

sudo systemctl enable wazuh-manager
sudo systemctl start wazuh-manager

```
rookie1@rookie1:~$ sudo tail -f /var/ossec/lllogs/ossec.log
tail: cannot open '/var/ossec/lllogs/ossec.log' for reading: No such file or directory
tail: no files remaining
rookie1@rookie1:~$ sudo tail -f /var/ossec/logs/ossec.log
2025/07/25 14:29:58 sca: INFO: Starting evaluation of policy: '/var/ossec/ruleset/sca/cis_ubuntu22-04.yml'
2025/07/25 14:29:59 indexer-connector: WARNING: IndexerConnector initialization failed for index 'wazuh-states-vulnerabilities-rooki
e1', retrying until the connection is successful.
2025/07/25 14:29:59 wazuh-modulesd:syscollector: INFO: Evaluation finished.
2025/07/25 14:30:00 wazuh-modulesd:vulnerability-scanner: INFO: Vulnerability scanner module started.
2025/07/25 14:30:04 wazuh-syscheckd: INFO: (6009): File integrity monitoring scan ended.
2025/07/25 14:30:04 wazuh-syscheckd: INFO: FIM sync module started.
2025/07/25 14:30:09 sca: INFO: Evaluation finished for policy '/var/ossec/ruleset/sca/cis_ubuntu22-04.yml'
2025/07/25 14:30:09 sca: INFO: Security Configuration Assessment scan finished. Duration: 11 seconds.
2025/07/25 14:30:52 rootcheck: INFO: Ending rootcheck scan.
2025/07/25 14:31:01 indexer-connector: INFO: IndexerConnector initialized successfully for index: wazuh-states-vulnerabilities-rooki
e1.
```

Figure 9. Wazuh log

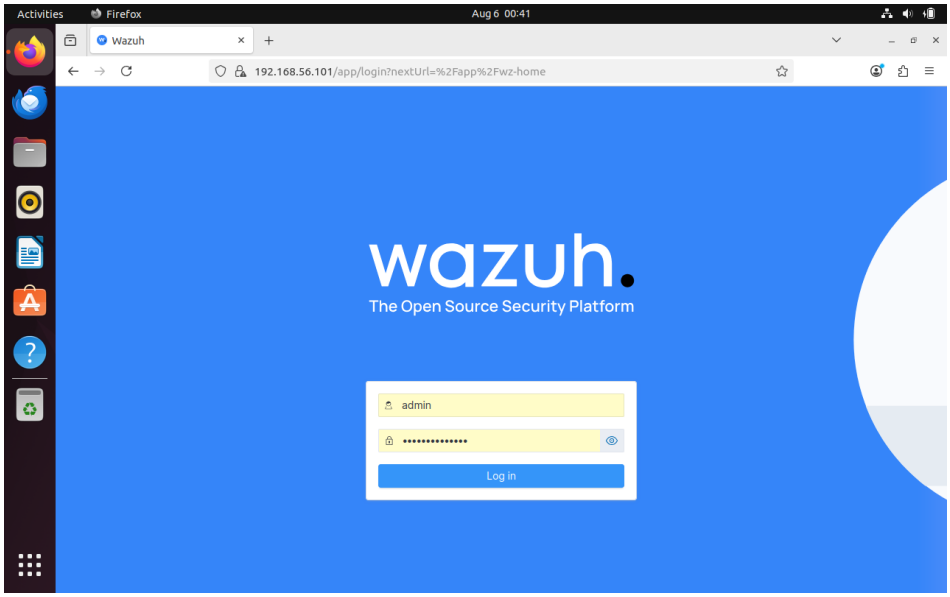

Figure 10. Wazuh login page

STEP 5: Simulate Ransomware Behaviour

5.1 On Kali (VM2):

Install OpenSSH if you want to connect: [7]

sudo apt install openssh-client

ssh username@192.168.56.101

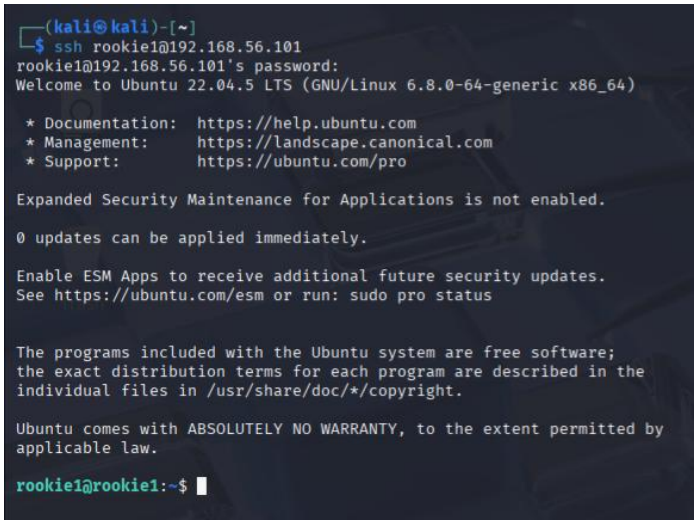

Figure 11. SSH connection to victim machine

5.2 Create Fake Ransomware Script on VM1:

echo "Top secret" > secret1.txt

echo "Don't share this" > secret2.txt

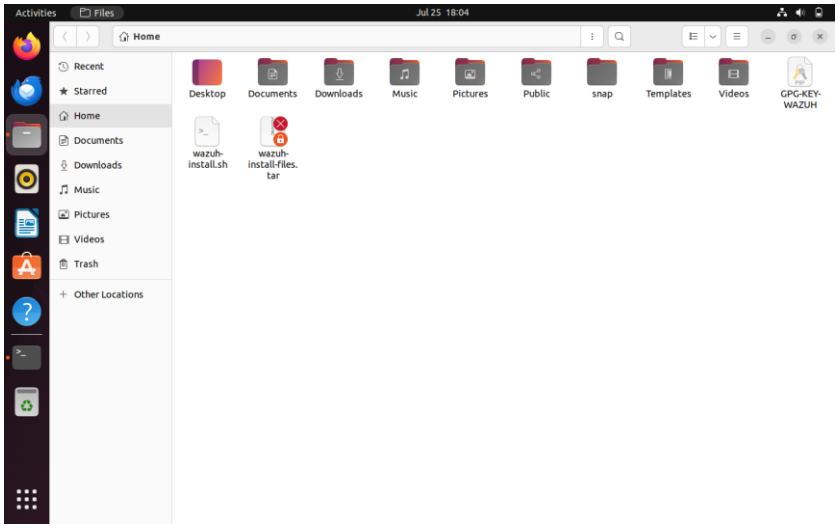

Figure 12. Before creation of files (victim files)

nano fake_ransom.py
Paste:

```
import os, glob

for file in glob.glob("*.txt"):
    os.rename(file, file + ".locked")
```

Run:
python3 fake_ransom.py

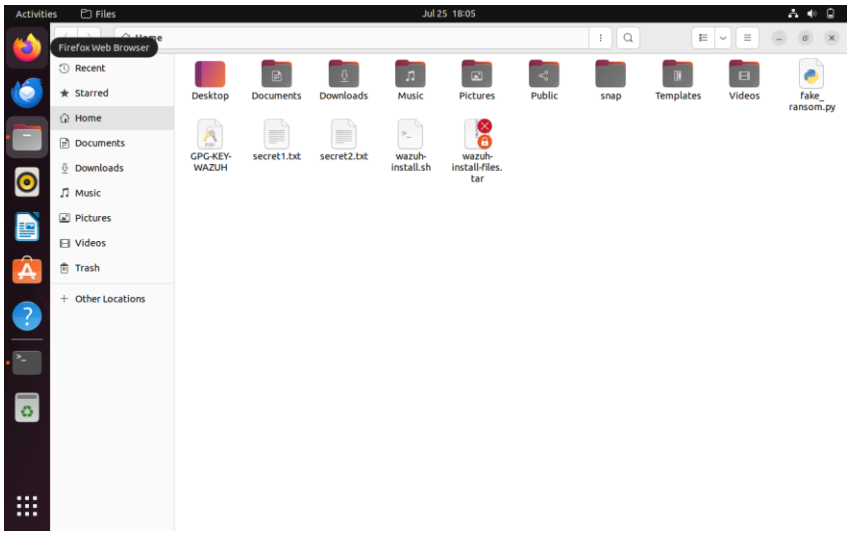

Figure 13. Fake ransomware script added

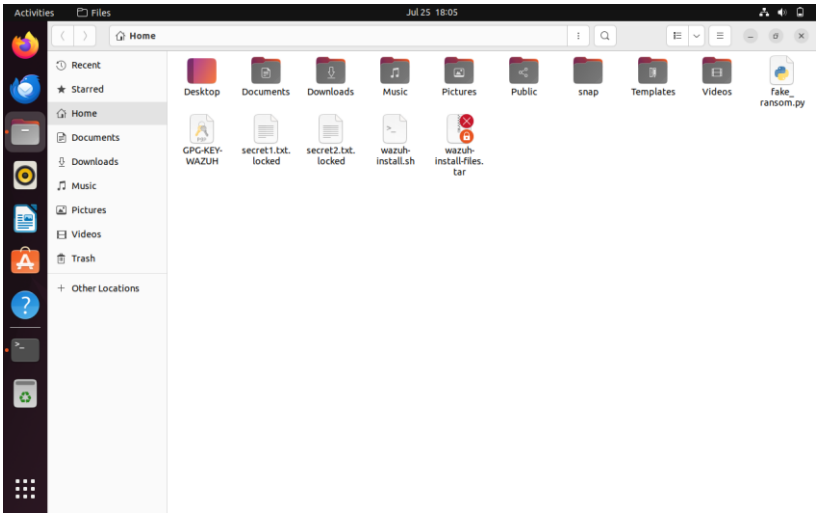

Figure 14. Locked files due to ransomware script

STEP 6: Enable Audit Monitoring [5]

6.1 On VM1:

sudo apt install auditd
sudo auditctl -w /home/ -p war -k ransomware_test
Run ransomware again, then check:
sudo ausearch -k ransomware_test

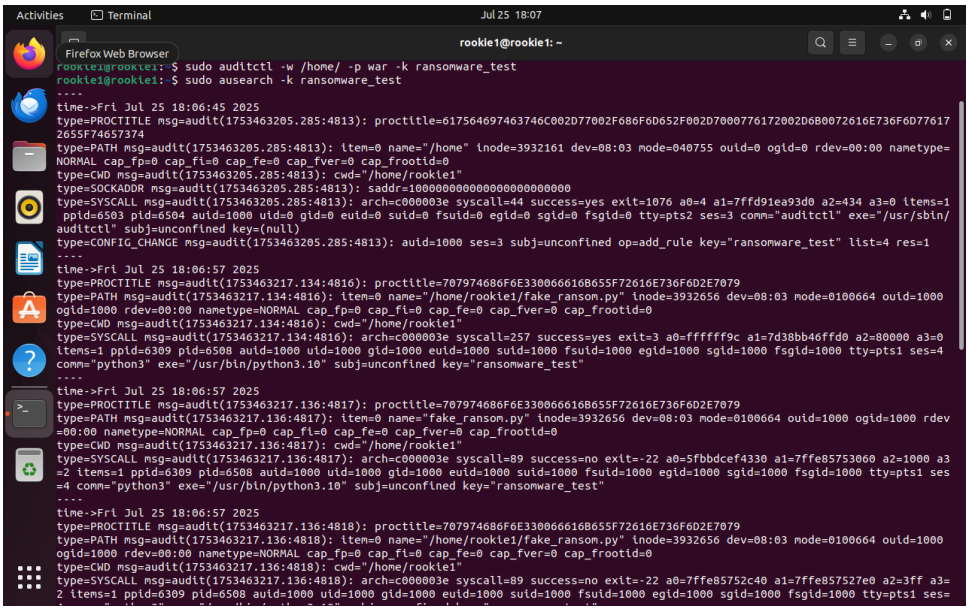

Figure 15. ausearch logs after the fake ransomware python script was run

STEP 7: Collect Evaluation Metrics

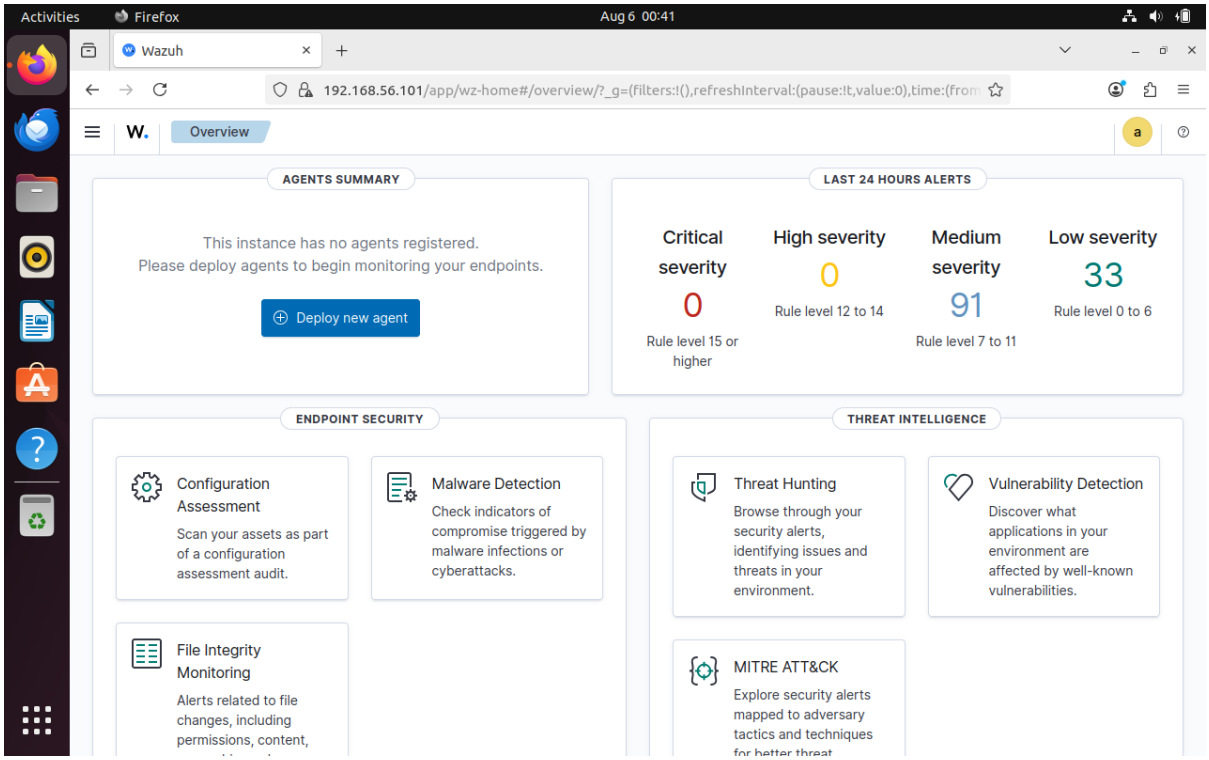

Figure 16. Wazuh dashboard referring to threat severity

The **commands listed in below table** (also shown in the image) should be run inside the **Victim VM (Ubuntu)** — because that is where:

- The **ransomware simulation** happens
- The **Wazuh agent** or auditd is logging the events
- The **system metrics** like CPU/RAM usage and file encryption activity occur

Metric	Run on	Detailed Instruction
Propagation Success	**Victim VM (Ubuntu)**	After running the fake ransomware script:ls *.locked | wc -l
Detection Latency	**Victim VM (Ubuntu)**	Run:sudo ausearch -k ransomware_testNote timestamps of first file change
Containment Time	**Victim VM (Ubuntu)**	Manually note: 1. When alert is triggered2. When ufw rules are applied
CPU/RAM Usage	**Victim VM (Ubuntu)**	Install and run:sudo apt install htop -y && htop [8]
False Positives	**Victim VM (Ubuntu)**	Monitor alerts:sudo tail -f /var/ossec/logs/ossec.logLook for alerts unrelated to attack

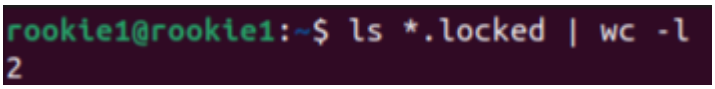

Figure 17. Number of locked files (Propagation success)

Workflow:

1. Log into **Victim VM**

2. Run fake ransomware:
   python3 fake_ransom.py
3. Check how many .locked files:
   ls *.locked | wc -l
4. View alert time:
   ausearch -k ransomware_test
5. Start htop to capture CPU usage:
   htop
6. Monitor Wazuh logs:
   tail -f /var/ossec/logs/ossec.log

STEP 8: Document Your Findings

- Architecture diagram
- Description of each step
- Screenshots
- Metric table

Ethical note: simulation only, no real malware.

CLEAR PLAN TO RUN NEW EXPERIMENTS IN THE FUTURE:

Use existing **Algorithm (Phases 1–6)** as the backbone, below I list each add-on with concrete steps so you can plug them into the same workflow.

Exp A: Establish a clean baseline (No ZTA vs ZTA)

Goal: Causal improvement deltas (Δ latency, Δ files encrypted, Δ CPU).

Steps (one testbed):

1. **Baseline (No-ZTA):** Disable auditd rules, stop Wazuh agent, set UFW to allow all. Snapshot.
2. Run the same Python simulator; record: detection latency (expected "none"), files encrypted, CPU/RAM.
3. **ZTA condition:** Restore snapshot, enable auditd + Wazuh + UFW (current config).
4. Repeat runs, collect metrics. Target **N≥30** per condition; report mean, SD, 95% CI, and Δ between conditions. (Attach one master table.)

Exp B: Separate the Wazuh manager (resilience)

Goal: Reflect realistic SOC placement.

Steps (add one VM):

1. Add a **Wazuh Manager VM**; leave **Agent on Ubuntu endpoint** only. Update agent to point to new manager IP.
2. Rerun Exp A's ZTA condition, confirm alerts flow and measure same metrics. (Note in methods that only topology changed.)

Exp C: Lateral movement containment

Goal: Validate micro-segmentation beyond a single host.

Steps (add one VM):

1. Add a third Linux VM ("Neighbor").
2. UFW on Endpoint: default-deny, allow only necessary management flows; **explicitly block SMB 445/139 and RDP 3389**, log deny.
3. From Attacker, attempt scans and connections (e.g., nmap, smbclient, ssh) to Neighbor, verify blocks in UFW logs and record any successful sessions.
4. Report success rate of blocks (%), with example log lines.

Exp D: Identity flavored hardening (minimal)

Goal: Keep it lightweight but closer to ZTA's "verify explicitly."

Steps (same VMs):

1. Keep the **non-interactive user** (/usr/sbin/nologin) and restrictive permissions, add simple **PAM** lockouts (e.g., faillock) and sudo policy to require TTY + reason codes.
2. Attempt credential abuse (password guess, su, sudo policy violations); log and report alert mapping.

Exp E: Statistical hygiene
Goal: Make results publishable.
Steps:

1. Pre-declare metrics: detection latency (s), files encrypted (%), CPU/RAM (%), alert FP/TP counts under a defined benign workload.
2. Use **N≥30** per condition; compute mean, SD, 95% CI; if distributions look normal, two-sample t-tests ZTA vs No-ZTA; otherwise non-parametric (Mann-Whitney).
3. Update Abstract and Table 2 from the master results table only.

C) One-page "master table" you'll produce after new runs
(Template fill with numbers from Exp A/B/C/E)

Metric	**No-ZTA (mean±SD, N)**	**ZTA-same-host (mean±SD, N)**	**ZTA-manager-off-host (mean±SD, N)**
Detection latency (s)	_	_	_
Files encrypted (%)	_	_	_
CPU during attack (%)	_	_	_
FP rate (benign)	_	_	_
Lateral-move blocks (%)	_	_	_

References

[1] Oracle (2023). *Oracle VM VirtualBox*. [online] VirtualBox. Available at: https://www.virtualbox.org/.

[2] Ubuntu 22.04.2 LTS *(Jammy Jellyfish)*. [online] Available at: https://www.releases.ubuntu.com/22.04/.

[3] Kali Linux. (n.d.). *Get Kali*. [online] Available at: https://www.kali.org/get-kali/#kali-platforms.

[4] Wazuh (2018). *Wazuh*. [online] Wazuh. Available at: https://wazuh.com/.

[5] GeeksforGeeks (2022). *Auditd Tool for Security Auditing on Linux Server*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/linux-unix/auditd-tool-for-security-auditing-on-linux-server/.

[6] Ubuntu (2023). *UFW - Community Help Wiki*. [online] help.ubuntu.com. Available at: https://help.ubuntu.com/community/UFW.

[7] Terpollari, O. (n.d.). *How to Install and Configure OpenSSH Server In Linux*. [online] www.tecmint.com. Available at: https://www.tecmint.com/install-openssh-server-in-linux/.

[8] GeeksforGeeks (2019). *htop command in Linux with examples*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/linux-unix/htop-command-in-linux-with-examples/.