Final Project Report

Virtual Company Lab: Core Server Infrastructure,
Monitoring and Intrusion Detection

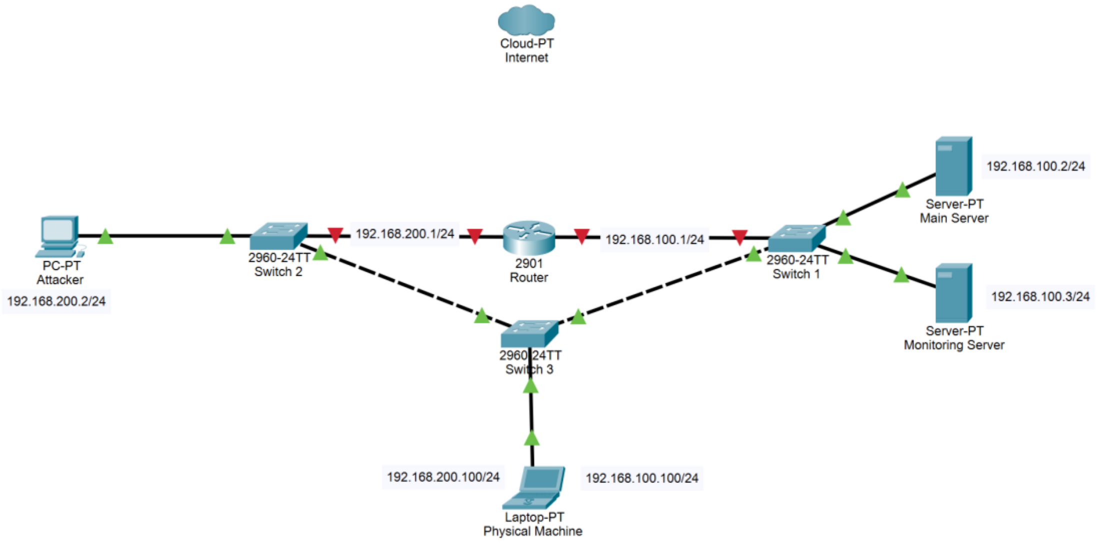
Ava Moshfegh
Student ID 1400012268003
University of Guilan
Supervisor Dr. Mohammad Salehi
10 September 2025

Introduction

This project presents the design and implementation of a compact virtual laboratory that models a small company network. The laboratory was deployed on VMware and consists of four virtual machines. The objective was to practice system administration, automation and security monitoring by building a realistic but manageable environment. Automation was implemented with Ansible to ensure repeatable and consistent provisioning across the local machines. The automation covers installation and configuration of services so that manual changes on each machine are not required. The project also served as an opportunity to apply knowledge gained from LPIC level one and level two self-study and to deepen skills in network security and observability.

Topology

The lab topology uses four virtual machines that simulate a small corporate network. The environment includes three Ubuntu servers on the internal network and one Kali Linux machine on the external network. The internal machines include a core server that hosts infrastructure services, a monitoring server that collects logs and visualizes metrics, and a router that connects the internal network to the internet and enforces basic access rules using iptables. The external machine acts as an attacker and generates test traffic.



Core Server and Deployed Services

The core server functions as the primary infrastructure node for the virtual company. It hosts a set of foundational services that support identity management, name resolution, web access and network security telemetry. Services deployed on the core server include the following.

BIND for Domain Name Service, ISC DHCP for dynamic address assignment:

These services provide internal name resolution and automatic assignment of IP addresses to hosts. Centralizing name and address management simplifies service discovery and reduces administrative effort.

Nginx:

A lightweight web server is used to host test pages and to provide a target for web based probes and monitoring checks. The web server demonstrates how application services appear in monitoring dashboards and how they are affected by attack traffic.

OpenLDAP as the central directory service:

A centralized directory stores user and group records and serves as the authoritative authentication source for the internal network. Centralized identity allows accounts to be managed in one place and used by multiple machines and services.

SSSD on client hosts to integrate with OpenLDAP:

The internal hosts use a system service that connects to the directory and provides seamless login and local credential caching. This architecture reduces repeated manual user management on each host while improving login performance.

Suricata:

A host based network intrusion detection system inspects network traffic for suspicious patterns and generates structured alerts. These alerts are exported as logs for aggregation and correlation on the monitoring server.

Log Forwarding with rsyslog and Filebeat:

System logs and security logs are forwarded to the monitoring server using standard logging agents. Log forwarding centralizes event data and enables historical investigation and dashboarding.

All service installation and configuration steps were automated with Ansible playbooks. The Ansible automation ensures that the environment is consistent and that the deployment process can be repeated or extended with minimal manual work.

Monitoring Server and Attack Simulation

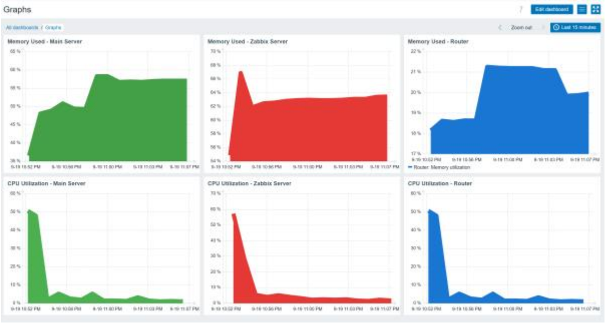
The monitoring server aggregates operational and security signals from the internal infrastructure and presents them through dashboards. The monitoring stack collects metrics and logs, raises alerts, and stores historical data for analysis.

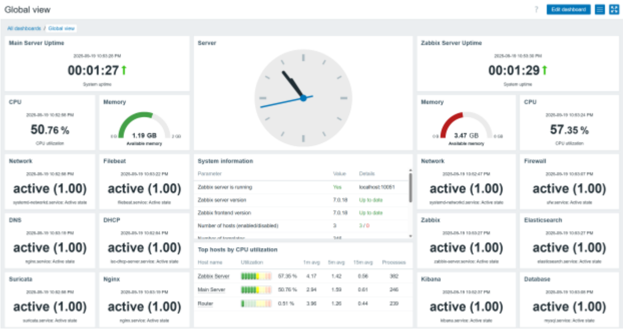
Zabbix Monitoring Platform:
A mature monitoring system was deployed to track host availability, service health and selected log events. The system provides visual dashboards and a rules engine that triggers notifications when preconfigured thresholds or event patterns are observed.

Log Pipeline:
System logs and intrusion detection alerts are forwarded to the monitoring server using specific log agents. rsyslog collects local syslog messages from each host and forwards them to the monitoring server. Suricata runs on the core server and produces structured IDS alerts in JSON format. Filebeat runs as the lightweight shipper that reads Suricata alert files and other log files and transmits them to the monitoring environment for indexing and correlation. The monitoring server receives and stores these inputs so they can be searched, correlated and displayed. Centralized logging with rsyslog, Suricata and Filebeat enables timely detection of anomalies and supports thorough post-incident review.

Attack Scenarios:
The external Kali Linux machine was used to simulate attacker activity against the core server. The primary tools and targets used in the tests were nmap for network reconnaissance and service enumeration, brute force and credential testing against the SSH service, and simple application probing against the Nginx web server. Suricata monitored network traffic and generated alerts for reconnaissance and suspicious packets. System-level events such as successful and failed SSH logins were recorded by the host syslog and forwarded by rsyslog to the monitoring server. These combined sources provided both network-level and host-level evidence of attack activity.

Integration and Observability:
Alerts generated by Suricata and log-based events forwarded by Filebeat are ingested on the monitoring server and visualized in Zabbix. Suricata feeds produce IDS alerts that Filebeat ships into the monitoring pipeline. Zabbix consumes metric and event data, correlates IDS alerts with host logs, and presents them on dashboards and alert channels. This integration provides end-to-end observability from raw network events and host logs to centralized visualization and actionable alerts. Screenshots of the Zabbix dashboards and sample Suricata logs illustrate real detection examples.


Graphs
Edit dashboard
All dashboards / Graphs
Zoom out
Last 15 minutes
Memory Used - Main Server
Memory Used - Zabbix Server
Memory Used - Router
Router: Memory utilization
CPU Utilization - Main Server
CPU Utilization - Zabbix Server
CPU Utilization - Router

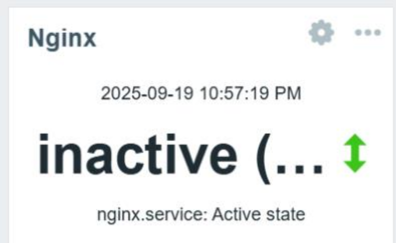


System information

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Zabbix server version	7.0.18	Up to date
Zabbix frontend version	7.0.18	Up to date
Number of hosts (enabled/disabled)	3	3 / 0

Top hosts by CPU utilization

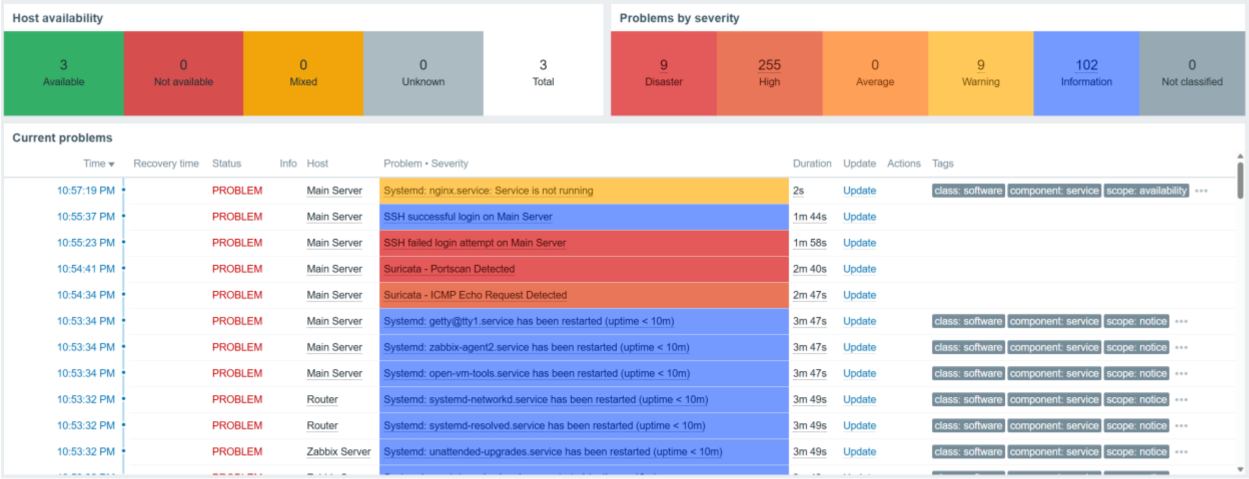
Host name	Utilization	1m avg	5m avg	15m avg	Processes
Zabbix Server	57.35 %	4.17	1.42	0.56	382
Main Server	50.76 %	2.94	1.59	0.61	246
Router	0.51 %	3.96	1.26	0.44	239


Nginx

2025-09-19 10:57:19 PM

inactive (…

nginx.service: Active state


Host availability

3	0	0	0	3
Available	Not available	Mixed	Unknown	Total

Problems by severity

9	255	0	9	102	0
Disaster	High	Average	Warning	Information	Not classified

Current problems

Time ▾	Recovery time	Status	Info	Host	Problem • Severity	Duration	Update	Actions	Tags
10:57:19 PM		PROBLEM		Main Server	Systemd: nginx.service: Service is not running	2s	Update		class: software component: service scope: availability •••
10:55:37 PM		PROBLEM		Main Server	SSH successful login on Main Server	1m 44s	Update		
10:55:23 PM		PROBLEM		Main Server	SSH failed login attempt on Main Server	1m 58s	Update		
10:54:41 PM		PROBLEM		Main Server	Suricata - Portscan Detected	2m 40s	Update		
10:54:34 PM		PROBLEM		Main Server	Suricata - ICMP Echo Request Detected	2m 47s	Update		
10:53:34 PM		PROBLEM		Main Server	Systemd: getty@tty1.service has been restarted (uptime < 10m)	3m 47s	Update		class: software component: service scope: notice •••
10:53:34 PM		PROBLEM		Main Server	Systemd: zabbix-agent2.service has been restarted (uptime < 10m)	3m 47s	Update		class: software component: service scope: notice •••
10:53:34 PM		PROBLEM		Main Server	Systemd: open-vm-tools.service has been restarted (uptime < 10m)	3m 47s	Update		class: software component: service scope: notice •••
10:53:32 PM		PROBLEM		Router	Systemd: systemd-networkd.service has been restarted (uptime < 10m)	3m 49s	Update		class: software component: service scope: notice •••
10:53:32 PM		PROBLEM		Router	Systemd: systemd-resolved.service has been restarted (uptime < 10m)	3m 49s	Update		class: software component: service scope: notice •••
10:53:32 PM		PROBLEM		Zabbix Server	Systemd: unattended-upgrades.service has been restarted (uptime < 10m)	3m 49s	Update		class: software component: service scope: notice •••