Project Report on

Design and Deployment of an Enterprise Network

Submitted by

Siddhant Misal	**(250244223030)**
Vipul More	**(250244223056)**
Sejal Nimbalkar	**(250244223038)**
Sarika Patil	**(250244223039)**

Under the guidance of

Mr. Sandeep Walvekar

In partial fulfillment of the award of Post Graduate Diploma in

IT Infrastructure, Systems and Security

(PG-DITISS)

Sunbeam Institute of Information Technology,

Pune (Maharashtra)

PG-DITISS -2025

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included; we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed**.**

Place: Pune

Date:

Siddhant Misal	**Vipul More**	**Sejal Nimbalkar**	**Sarika Patil**
(250244223030)	**(250244223056)**	**(250244223038)**	**(250244223039)**

CERTIFICATE

This is to certify that the project report entitled **“Design and Deployment of an Enterprise Network”**, submitted by **Siddhant Misal** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar	**Mr.Vishal Salunkhe**
Guide	Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **"Design and Deployment of an Enterprise Network"**, submitted by **Vipul More** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar	**Mr.Vishal Salunkhe**
Guide	Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **“Design and Deployment of an Enterprise Network”**, submitted by **Sejal Nimbalkar** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar

Guide

Mr.Vishal Salunkhe

Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

CERTIFICATE

This is to certify that the project report entitled **"Design and Deployment of an Enterprise Network"**, submitted by **Sarika Patil** is the bonafide work completed under our supervision and guidance in partial fulfillment for the award of Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Mr. Sandeep Walvekar
Guide

Mr.Vishal Salunkhe
Course Coordinator

Mr. Nitin Kudale

CEO

Sunbeam Institute of Information Technology

Pune (M.S.) – 411057

APPROVAL CERTIFICATE

This Project II report entitled **“Design and Deployment of an Enterprise Network”** by **Siddhant Misal (250244223030)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ______________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **"Design and Deployment of an Enterprise Network"** by **Vipul More (250244223056)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ______________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **“Design and Deployment of an Enterprise Network”** by **Sejal Nimbalkar (250244223038)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ______________________________

(Signature)

____

(Name)

APPROVAL CERTIFICATE

This Project II report entitled **“Design and Deployment of an Enterprise Network”** by **Sarika Patil (250244223039)** is approved for Post Graduate Diploma in IT Infrastructure, Systems and Security (PG-DITISS) of Sunbeam Institute of Information Technology, Pune (M.S.).

Place: Pune

Date:

Examiner: ____________________

(Signature)

____

(Name)

CONTENTS

ABSTRACT

In the era of digital transformation, enterprises demand secure, reliable, and automated IT infrastructures to support mission-critical operations. This project titled Design and Deployment of an Enterprise Network presents a comprehensive model for building a robust network environment by integrating core infrastructure components and modern DevOps practices.

The enterprise network is designed using pfSense, which acts as a firewall and router to segment and protect network zones. A Web Server hosted in a DMZ (Demilitarized Zone) enables secure access to applications, while a Database Server resides in the internal LAN for data integrity and controlled access.

To facilitate automated software delivery, a CI/CD pipeline using Jenkins is implemented, allowing developers to continuously test and deploy code changes. System reliability is ensured by Nagios, a monitoring solution that tracks system performance and availability. For security, the network employs Snort Intrusion Detection and Prevention System (IDS/IPS) to monitor traffic and alert on potential threats.

The project highlights the complete configuration of each component, implementation of firewall rules, creation of deployment pipelines, and setup of real-time monitoring. This ensures a secure, scalable, and efficient enterprise IT environment suitable for modern organizational needs.

1. INTRODUCTION

Organizations rely heavily on enterprise networks to ensure seamless communication, resource sharing, and business continuity. A well-designed enterprise network serves as the backbone of an organization’s IT infrastructure it connects departments, facilitates application deployment, enables secure communication, supports monitoring, and ensures that business operations run smoothly and securely.

The project Design and Deployment of an Enterprise Network focuses on creating a secure, scalable, and efficient network architecture tailored to the operational needs of a typical enterprise. It covers all aspects of network design including logical segmentation, secure routing, automated application deployment, real-time monitoring, and intrusion prevention.

The core idea behind this project is to integrate various services and technologies into a cohesive network system. It includes the implementation of pfSense as the firewall and router to manage internal and external traffic. The network is logically segmented into WAN (Wide Area Network), LAN (Local Area Network)**,** and DMZ (Demilitarized Zone) to enhance security by isolating internal resources from publicly accessible servers.

A web server is deployed in the DMZ to host organizational applications or websites. Internally, a database server is used to store sensitive data securely.

To facilitate rapid and reliable software development practices, a Jenkins CI/CD pipeline is established. It automates the process of building, testing, and deploying applications, thereby reducing manual errors and accelerating the development lifecycle.

Nagios, a powerful monitoring tool, is employed to oversee the performance and availability of all critical systems and services. It alerts administrators in case of system failures, high resource usage, or other anomalies. To further strengthen the security posture of the network**,** Snort Intrusion Prevention System (IPS) is integrated. It inspects network traffic in real-time to detect and prevent potential threats or attacks.

The goal of this project is not only to design a network but also to demonstrate the deployment and management of enterprise IT services in a secure and automated manner.

The final setup replicates a real-world enterprise IT infrastructure capable of handling daily operations, ensuring security, and allowing for future scalability.

1.1 Applications

- E-Commerce Platforms: Web and database servers support customer-facing applications. CI/CD pipelines allow fast rollouts of new features and security patches. pfSense manages secure traffic routing between zones, Snort blocks malicious requests (like SQL injection), and Nagios prevents business downtime.

- Banking and Financial Services: Security and high availability are key. Jenkins supports frequent secure updates to financial software, pfSense separates internal services from the internet, Snort identifies threats in real time, and Nagios ensures high uptime. Web and database servers handle sensitive financial transactions and user data.

- Startups and Tech Incubators: Startups benefit from a scalable and secure infrastructure where they can rapidly build, test, and deploy applications using Jenkins. pfSense provides basic security with DMZs, Snort guards against data leaks or attacks, and Nagios ensures reliability—all at low cost.

- Government and Defense Networks: Used to separate public services (hosted in DMZ) from internal, sensitive systems. pfSense supports VPNs and firewall rules, Snort monitors for nation-state cyberattacks, and Nagios ensures services are always operational. Jenkins can automate policy updates and software rollouts.

1.2 Project Plan

Table: Activities Details

Sr. No.	ACTIVITY	WEEK			
		1	2	3	4
1	Project group formation	🟨			
2	Project work to be started in respective labs	🟨			
3	First review with PPT presentation		🟨		
4	Design Use-Case view as per project			🟨	
5	Design Block diagram as per project			🟨	
6	Second review with PPT presentation				🟨
7	Selection			🟨	
8	Final review with PPT presentation			🟨	
9	Implementation coding as per project			🟨	
10	Testing, Troubleshooting with different techniques			🟨	🟨
11	Created Soft copy of project and then final hard copy				🟨

2. LITERATURE SURVEY

Paper 1: - Enterprise Network Design and Security Optimization

Author: Kavita Saini

Description: The project titled Enterprise Network Design and Security Optimization focuses on building a secure, scalable, and efficient network architecture for modern business environments. It emphasizes a layered design (core, distribution, access) with proper segmentation using VLANs and a DMZ to isolate public-facing services like web and mail servers from internal resources. Security is reinforced through firewalls, IDS/IPS systems (like Snort), and access control lists to detect and prevent unauthorized access.

Monitoring tools such as Nagios ensure continuous visibility into network health, while redundancy and failover mechanisms improve availability. Your project aligns with this framework by integrating pfSense for firewall and routing, Jenkins for CI/CD automation, Snort for intrusion detection, and Nagios for monitoring—all of which collectively enhance network performance, security, and manageability.

3.SYSTEM DEVELOPMENT AND DESIGN

3.1 Proposed System

The proposed system for the project “Design and Deployment of an Enterprise Network” is a secure, scalable, and structured network architecture designed to meet enterprise-level performance, reliability, and security needs. It integrates multiple core services and follows a multi-layered security approach through network segmentation and specialized tools.

At the network perimeter, pfSense is deployed as the primary gateway and firewall, managing and filtering all traffic between the WAN (internet), LAN (internal network), and DMZ (public-facing servers). It enforces strict firewall rules, applies Network Address Translation (NAT), and allocates IP addresses via DHCP, ensuring controlled and secure access.

The DMZ zone hosts a web server such as Apache or Nginx to serve enterprise web applications. This server handles both static and dynamic content, with application code updates automatically deployed via Jenkins, which manages the CI/CD pipeline to ensure seamless and error-free releases.

The LAN zone contains a database server, securely isolated from public access, which stores critical enterprise data such as user information and transaction records. Access is strictly controlled via pfSense firewall policies, allowing only authorized systems to communicate with it.

To strengthen security, Suricata IDS/IPS monitors network traffic in real time, detecting and preventing suspicious or malicious activities using signature-based, protocol-based, and anomaly-based detection techniques. Nagios provides continuous monitoring of servers and services, issuing alerts in case of performance degradation, failures, or security incidents.

This architecture ensures high availability, proactive security, efficient automation, and centralized monitoring, making it suitable for robust enterprise IT environments.

3.2 Flow chart

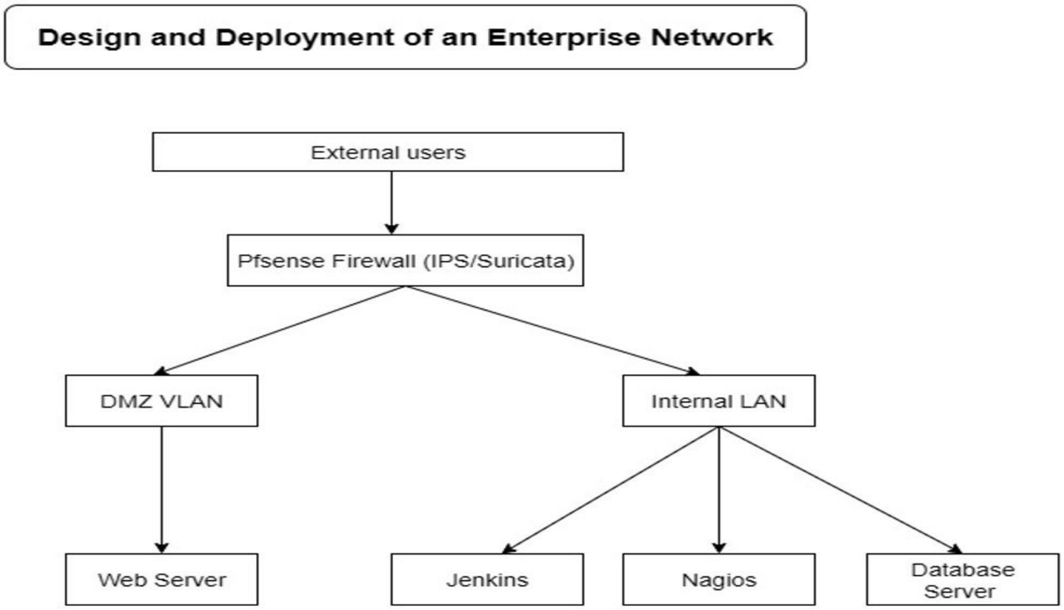

Design and Deployment of an Enterprise Network
External users
Pfsense Firewall (IPS/Suricata)
DMZ VLAN
Internal LAN
Web Server
Jenkins
Nagios
Database Server

Figure: Flowchart

3.3 Technology used

3.3.1 pfSense (Firewall & Router)

Working:

pfSense is an open-source firewall and routing platform deployed as a gateway device to manage and secure traffic between the WAN (internet), LAN (internal network), and DMZ (public-facing servers). Acting as the first line of defense, it inspects and filters all inbound and outbound packets based on predefined firewall rules, blocking unauthorized or suspicious connections.

pfSense also performs Network Address Translation (NAT), allowing multiple internal devices to share a single public IP while hiding their private addresses from external networks. It can function as a DHCP server, automatically assigning IP addresses to devices within the LAN and DMZ, simplifying network management.

By segmenting networks and controlling traffic flow, pfSense prevents direct exposure of internal systems to the internet. Its robust firewall policies, VPN support, and intrusion detection/prevention capabilities help ensure that only legitimate traffic is allowed, maintaining the confidentiality, integrity, and availability of enterprise network resources.

Key Features:

- VLAN and DMZ setup
- NAT and port forwarding
- Stateful firewall with traffic filtering
- VPN support (IPSec, OpenVPN)
- Web-based GUI for configuration

3.3.2 Apache (Web Server)

Working:

In the enterprise network, a web server such as Apache or Nginx is deployed in the Demilitarized Zone (DMZ) to securely host the organization's web application. Positioned between the external network and the internal LAN, the DMZ ensures that the web server is accessible to users over the internet while keeping internal systems protected.

The web server listens for HTTP and HTTPS requests from clients, serving static content like HTML, CSS, and images, as well as dynamic content generated by server-side scripts or applications. It communicates with the backend database server over secure channels to retrieve or store data as required.

Jenkins is integrated into the deployment pipeline to automatically push code updates and new features directly to the web server. This setup enables continuous delivery, reduces manual intervention, and ensures that the latest version of the application is deployed quickly while maintaining security and performance.

Key Features:

- Supports PHP, Python, HTML, and SSL
- Reverse proxy and load balancing (Nginx)

- Modular and extensible architecture
- High performance and scalable
- Logs access and errors for auditing

3.3.3 Mariadb (Database Server)

Working:
The database server is a critical component that stores and manages application data such as user information, transaction logs, configurations, and other essential records. It operates as the backend of the application, responding to SQL queries sent by the web server to retrieve, insert, update, or delete data.

For security, the database server is hosted within the Local Area Network (LAN), isolated from direct internet access to minimize exposure to external threats. This isolation ensures that only authorized internal systems, such as the web server, can communicate with it. Firewall rules are implemented to strictly control access, allowing connections only from trusted IP addresses or specific network segments.

By restricting public access and enforcing authentication, encryption, and secure communication protocols, the database server remains protected from unauthorized access, data breaches, and malicious activity, ensuring the confidentiality, integrity, and availability of application data.

Key Features:

- ACID-compliant data handling
- Strong authentication and access control
- High-speed query processing
- Supports backups and replication
- Cross-platform compatibility

3.3.4 Jenkins (CI/CD Tool)

Working:

Jenkins is an open-source automation server that streamlines the software development lifecycle by automating repetitive tasks. It integrates with version control systems like GitHub to pull the latest source code whenever changes are committed. Once the code is retrieved, Jenkins automatically builds it, compiles necessary components, and runs automated tests to ensure quality and functionality.

If tests pass successfully, Jenkins proceeds to deploy the application to the target environment, such as a web server, eliminating the need for manual deployment steps. This continuous integration and continuous deployment (CI/CD) process reduces human errors, speeds up delivery, and ensures consistent, reliable releases.

Jenkins supports a wide range of plugins, enabling integration with various tools, cloud services, and deployment platforms. By automating building, testing, and deployment, Jenkins improves productivity, fosters collaboration between development and operations teams, and ensures that high-quality software reaches users faster and more efficiently.

Key Features:

- Open-source with a large plugin ecosystem
- Supports Git integration and pipelines
- Automated build, test, and deploy processes
- Notifies build status via email
- Web-based interface for job management

3.3.5 Suricata (IDS/IPS)

Working:

Suricata is an advanced, open-source real-time Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) designed to monitor and secure network environments. It analyzes network traffic in real time to identify suspicious or malicious activities that may indicate cyberattacks, policy violations, or abnormal behavior. Suricata uses a combination of signature-based detection (matching traffic against known threat patterns), protocol-based detection (validating the correct use of network protocols), and anomaly-based detection (identifying deviations from normal traffic patterns).

By inspecting packets deeply (Deep Packet Inspection), Suricata can detect threats such as malware, exploits, port scans, and brute-force attempts. When deployed as an IDS, it generates alerts for security teams to investigate. When configured as an IPS, it can actively block or drop malicious traffic, preventing threats from reaching their targets.

Suricata supports high-performance, multi-threaded processing, enabling it to handle large network loads efficiently. It can log detailed network events, export metadata, and integrate with tools like Kibana, ElasticSearch, and SIEM systems for advanced threat analysis. Additionally, Suricata can extract files from network streams for further examination. With its versatile detection methods and scalability, Suricata plays a crucial role in enhancing network visibility, detecting attacks, and safeguarding critical infrastructure from evolving cyber threats.

.

Key Features:

- Multi-threaded performance (faster than Snort)
- IDS/IPS support (can detect *and* block)
- Supports protocol identification**,** TLS**,** HTTP**,** DNS**,** FTP, etc.
- Works with modern rule sets (ET, Suricata-Community)
- Generates logs in JSON, compatible with SIEM tools like ELK stack
- Real time detection

3.3.6 Nagios (Monitoring Tool)

Working:
Nagios is an open-source monitoring solution used to track the availability, performance, and overall health of IT infrastructure. It monitors a wide range of components, including servers (web, database, Jenkins, application servers) and services (HTTP, SSH, MySQL, SMTP, etc.). Nagios relies on plugins, which are small scripts or executables that collect real-time data about system status and service performance. These plugins run periodic checks, measuring parameters like uptime, CPU and memory usage, disk space, network connectivity, and service response time.

When a check runs, Nagios compares the collected data against predefined thresholds. If a service is down, unreachable, or performing poorly, it changes the status from “OK” to “Warning” or “Critical” based on severity. This triggers alerts, which can be sent via email, SMS, or integrated messaging tools, enabling system administrators to respond quickly.

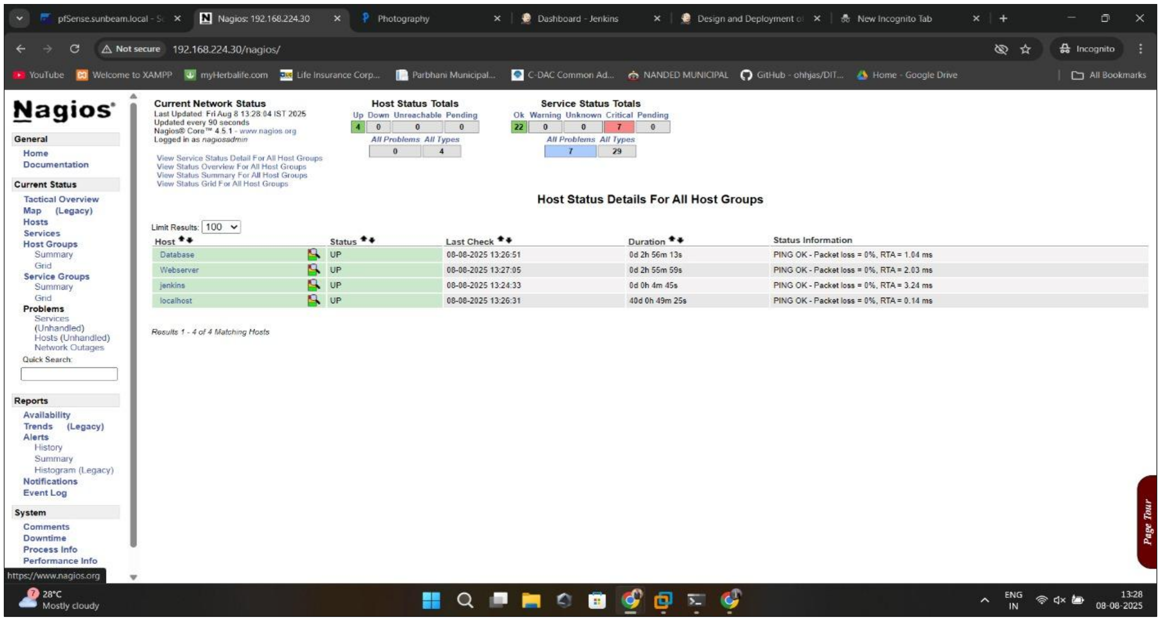
Nagios also maintains detailed logs and provides web-based dashboards for visualizing performance trends and identifying recurring problems. This proactive monitoring helps organizations prevent outages, optimize resource usage, and ensure critical applications remain available to users. By automating health checks and providing early warnings, Nagios plays a vital role in maintaining reliable, secure, and high-performing IT systems across diverse environments.

Key Features:

- Host and service monitoring
- Alerting via email/SMS
- Custom plugin support
- Web-based dashboard
- Historical reporting and graphs

4. Project Output

4.1 Nagios



Current Network Status
Last Updated: Fri Aug 8 13:28:04 IST 2025
Updated every 90 seconds
Nagios® Core™ 4.5.1 - www.nagios.org
Logged in as *nagiosadmin*

Host Status Totals

Up	Down	Unreachable	Pending
4	0	0	0

All Problems	All Types
0	4

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
22	0	0	7	0

All Problems	All Types
7	29

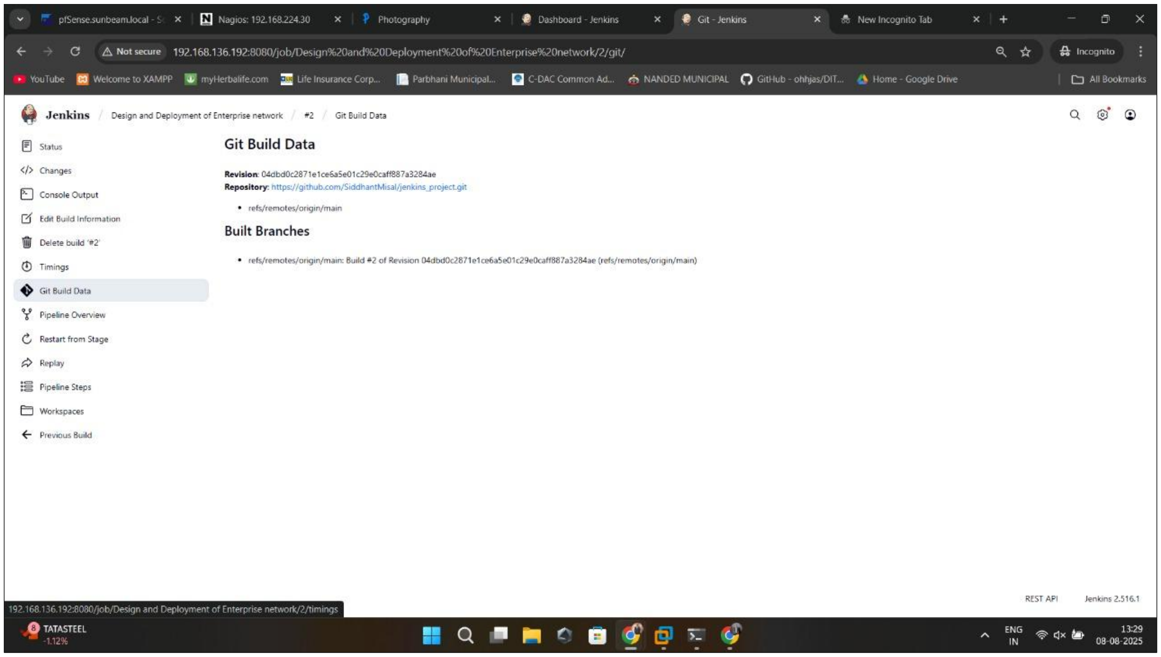
Host Status Details For All Host Groups

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
Database	UP	08-08-2025 13:26:51	0d 2h 56m 13s	PING OK - Packet loss = 0%, RTA = 1.04 ms
Webserver	UP	08-08-2025 13:27:05	0d 2h 55m 59s	PING OK - Packet loss = 0%, RTA = 2.03 ms
jenkins	UP	08-08-2025 13:24:33	0d 0h 4m 45s	PING OK - Packet loss = 0%, RTA = 3.24 ms
localhost	UP	08-08-2025 13:26:31	40d 0h 49m 25s	PING OK - Packet loss = 0%, RTA = 0.14 ms

Results 1 - 4 of 4 Matching Hosts

4.2 Jenkins



Git Build Data

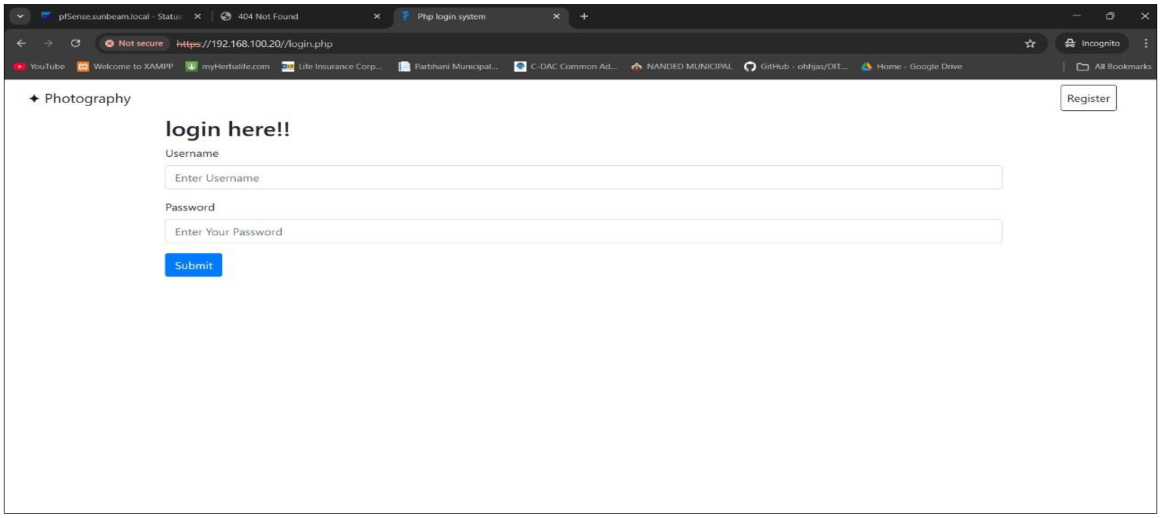
Revision: 04dbd0c2871e1ce6a5e01c29e0caff887a3284ae
Repository: https://github.com/SiddhantMisal/jenkins_project.git

- refs/remotes/origin/main

Built Branches

- refs/remotes/origin/main: Build #2 of Revision 04dbd0c2871e1ce6a5e01c29e0caff887a3284ae (refs/remotes/origin/main)

4.3 Web Application


pfSense.sunbeam.local - Status
404 Not Found
Php login system
Not secure
https://192.168.100.20//login.php
Incognito

✦ Photography

Register

login here!!

Username

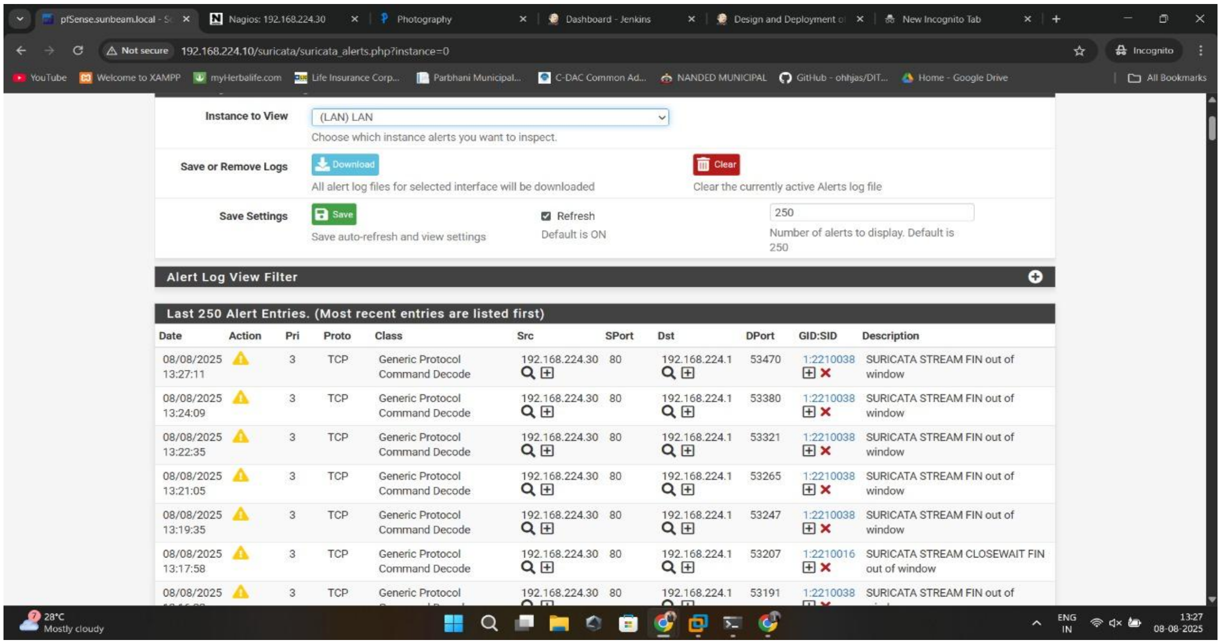
Enter Username

Password

Enter Your Password

Submit

4.4 LAN traffic



Instance to View: (LAN) LAN

Choose which instance alerts you want to inspect.

Save or Remove Logs: Download — All alert log files for selected interface will be downloaded; Clear — Clear the currently active Alerts log file

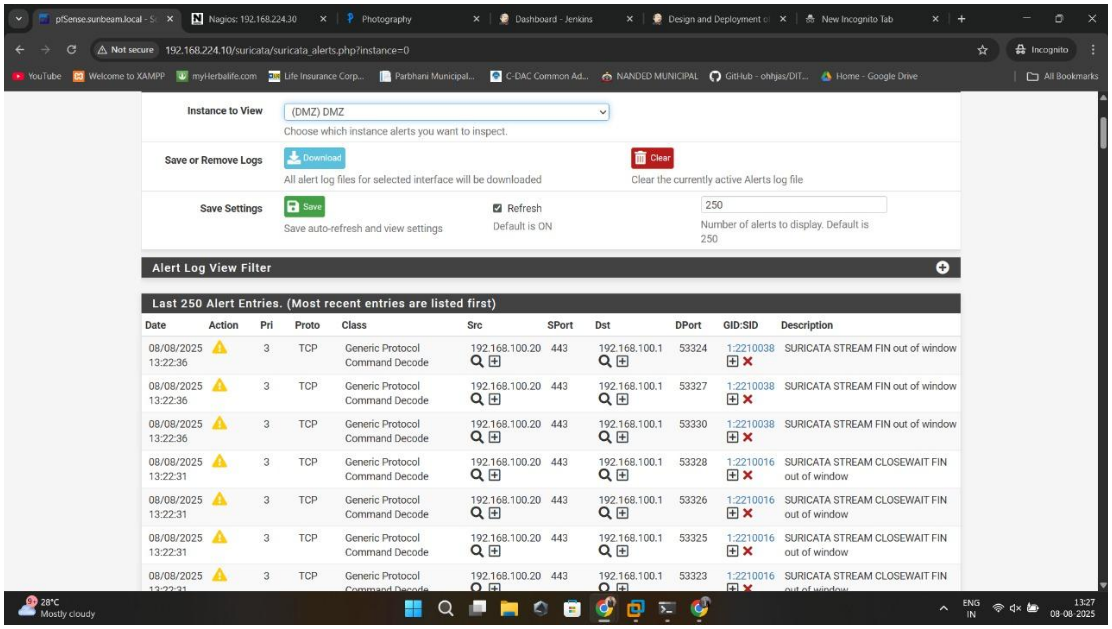
Save Settings: Save — Save auto-refresh and view settings; ☑ Refresh — Default is ON; 250 — Number of alerts to display. Default is 250

Alert Log View Filter

Last 250 Alert Entries. (Most recent entries are listed first)

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
08/08/2025 13:27:11	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53470	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:24:09	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53380	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:22:35	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53321	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:21:05	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53265	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:19:35	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53247	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:17:58	⚠	3	TCP	Generic Protocol Command Decode	192.168.224.30	80	192.168.224.1	53207	1:2210016	SURICATA STREAM CLOSEWAIT FIN out of window
08/08/2025	⚠	3	TCP	Generic Protocol	192.168.224.30	80	192.168.224.1	53191	1:2210038	SURICATA STREAM FIN out of

4.5 DMZ



Instance to View: (DMZ) DMZ
Choose which instance alerts you want to inspect.

Save or Remove Logs: Download
All alert log files for selected interface will be downloaded
Clear
Clear the currently active Alerts log file

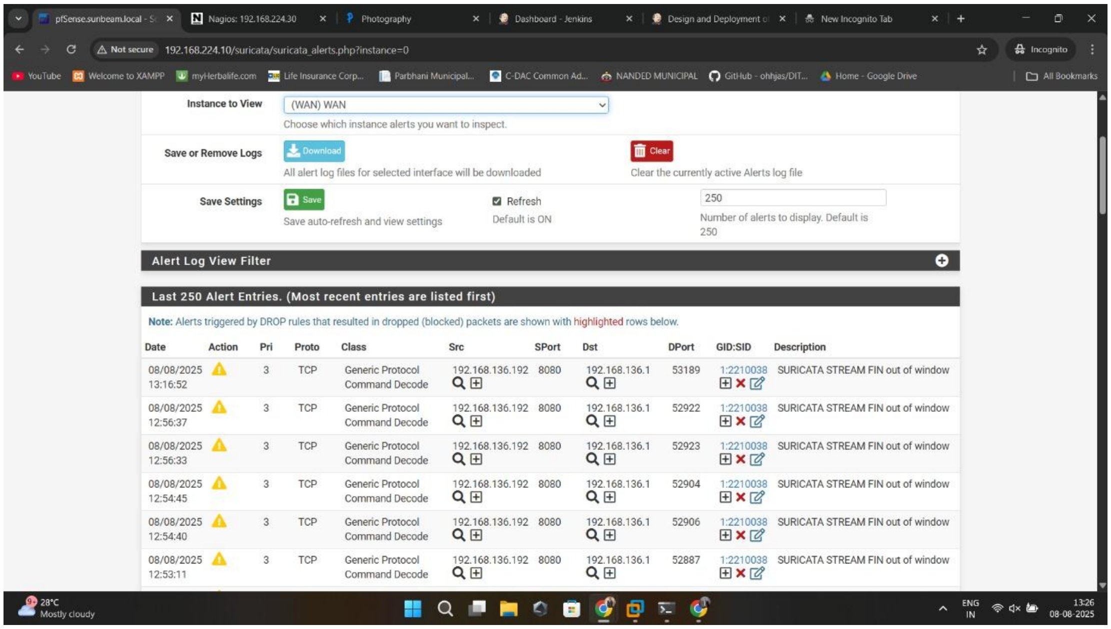
Save Settings: Save
Save auto-refresh and view settings
☑ Refresh
Default is ON
250
Number of alerts to display. Default is 250

Alert Log View Filter

Last 250 Alert Entries. (Most recent entries are listed first)

Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
08/08/2025 13:22:36	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53324	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:22:36	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53327	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:22:36	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53330	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 13:22:31	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53328	1:2210016	SURICATA STREAM CLOSEWAIT FIN out of window
08/08/2025 13:22:31	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53326	1:2210016	SURICATA STREAM CLOSEWAIT FIN out of window
08/08/2025 13:22:31	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53325	1:2210016	SURICATA STREAM CLOSEWAIT FIN out of window
08/08/2025 13:22:31	⚠	3	TCP	Generic Protocol Command Decode	192.168.100.20	443	192.168.100.1	53323	1:2210016	SURICATA STREAM CLOSEWAIT FIN out of window

4.6 WAN



Instance to View: (WAN) WAN
Choose which instance alerts you want to inspect.

Save or Remove Logs: Download
All alert log files for selected interface will be downloaded
Clear
Clear the currently active Alerts log file

Save Settings: Save
Save auto-refresh and view settings
☑ Refresh
Default is ON
250
Number of alerts to display. Default is 250

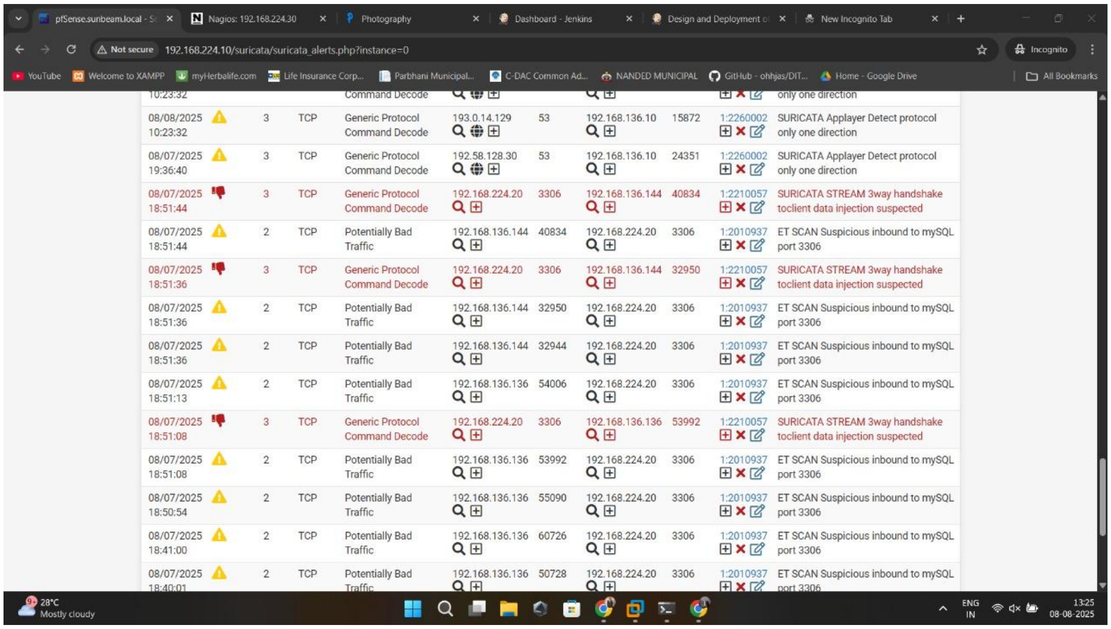
Alert Log View Filter

Last 250 Alert Entries. (Most recent entries are listed first)

Note: Alerts triggered by DROP rules that resulted in dropped (blocked) packets are shown with highlighted rows below.

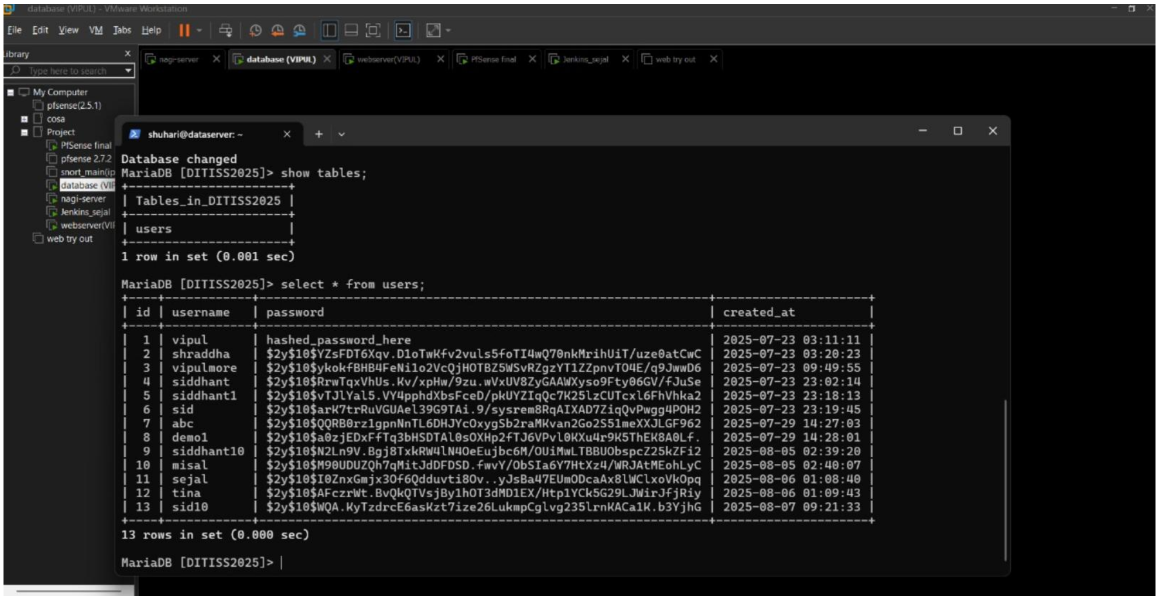
Date	Action	Pri	Proto	Class	Src	SPort	Dst	DPort	GID:SID	Description
08/08/2025 13:16:52	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	53189	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 12:56:37	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	52922	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 12:56:33	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	52923	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 12:54:45	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	52904	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 12:54:40	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	52906	1:2210038	SURICATA STREAM FIN out of window
08/08/2025 12:53:11	⚠	3	TCP	Generic Protocol Command Decode	192.168.136.192	8080	192.168.136.1	52887	1:2210038	SURICATA STREAM FIN out of window

4.7 Blocked traffic



10:23:32				Command Decode						only one direction
08/08/2025 10:23:32	⚠	3	TCP	Generic Protocol Command Decode	193.0.14.129	53	192.168.136.10	15872	1:2260002	SURICATA Applayer Detect protocol only one direction
08/07/2025 19:36:40	⚠	3	TCP	Generic Protocol Command Decode	192.58.128.30	53	192.168.136.10	24351	1:2260002	SURICATA Applayer Detect protocol only one direction
08/07/2025 18:51:44	👎	3	TCP	Generic Protocol Command Decode	192.168.224.20	3306	192.168.136.144	40834	1:2210057	SURICATA STREAM 3way handshake toclient data injection suspected
08/07/2025 18:51:44	⚠	2	TCP	Potentially Bad Traffic	192.168.136.144	40834	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:51:36	👎	3	TCP	Generic Protocol Command Decode	192.168.224.20	3306	192.168.136.144	32950	1:2210057	SURICATA STREAM 3way handshake toclient data injection suspected
08/07/2025 18:51:36	⚠	2	TCP	Potentially Bad Traffic	192.168.136.144	32950	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:51:36	⚠	2	TCP	Potentially Bad Traffic	192.168.136.144	32944	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:51:13	⚠	2	TCP	Potentially Bad Traffic	192.168.136.136	54006	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:51:08	👎	3	TCP	Generic Protocol Command Decode	192.168.224.20	3306	192.168.136.136	53992	1:2210057	SURICATA STREAM 3way handshake toclient data injection suspected
08/07/2025 18:51:08	⚠	2	TCP	Potentially Bad Traffic	192.168.136.136	53992	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:50:54	⚠	2	TCP	Potentially Bad Traffic	192.168.136.136	55090	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:41:00	⚠	2	TCP	Potentially Bad Traffic	192.168.136.136	60726	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306
08/07/2025 18:40:01	⚠	2	TCP	Potentially Bad Traffic	192.168.136.136	50728	192.168.224.20	3306	1:2010937	ET SCAN Suspicious inbound to mySQL port 3306

4.8 Database Server



```
Database changed
MariaDB [DITISS2025]> show tables;
+----------------------+
| Tables_in_DITISS2025 |
+----------------------+
| users                |
+----------------------+
1 row in set (0.001 sec)

MariaDB [DITISS2025]> select * from users;
+----+------------+--------------------------------------------------------------+---------------------+
| id | username   | password                                                     | created_at          |
+----+------------+--------------------------------------------------------------+---------------------+
|  1 | vipul      | hashed_password_here                                         | 2025-07-23 03:11:11 |
|  2 | shraddha   | $2y$10$YZsFDT6Xqv.D1oTwKfv2vuls5foTI4wQ70nkMrihUiT/uze0atCwC | 2025-07-23 03:20:23 |
|  3 | vipulmore  | $2y$10$ykokfBHB4FeNi1o2VcQjHOTBZ5WSvRZgzYT1ZZpnvTO4E/q9JwwD6 | 2025-07-23 09:49:55 |
|  4 | siddhant   | $2y$10$RrwTqxVhUs.Kv/xpHw/9zu.wVxUV8ZyGAAWXyso9Fty06GV/fJuSe | 2025-07-23 23:02:14 |
|  5 | siddhant1  | $2y$10$vTJlYal5.VY4pphdXbsFceD/pkUYZIqQc7K25lzCUTcxl6FhVhka2 | 2025-07-23 23:18:13 |
|  6 | sid        | $2y$10$arK7trRuVGUAel39G9TAi.9/sysrem8RqAIXAD7ZiqQvPwgg4POH2 | 2025-07-23 23:19:45 |
|  7 | abc        | $2y$10$QQRB0rz1gpnNnTL6DHJYcOxygSb2raMKvan2Go2S51meXXJLGF962 | 2025-07-29 14:27:03 |
|  8 | demo1      | $2y$10$a0zjEDxFfTq3bHSDTAl0sOXHp2fTJ6VPvl0KXu4r9K5ThEK8A0Lf. | 2025-07-29 14:28:01 |
|  9 | siddhant10 | $2y$10$N2Ln9V.Bgj8TxkRW4lN4OeEujbc6M/OUiMwLTBBUObspcZ25kZFi2 | 2025-08-05 02:39:20 |
| 10 | misal      | $2y$10$M90UDUZQh7qMitJdDFDSD.fwvY/ObSIa6Y7HtXz4/WRJAtMEohLyC | 2025-08-05 02:40:07 |
| 11 | sejal      | $2y$10$I0ZnxGmjx3Of6Qdduvti8Ov..yJsBa47EUmODcaAx8lWClxoVkOpq | 2025-08-06 01:08:40 |
| 12 | tina       | $2y$10$AFczrWt.BvQkQTVsjBy1hOT3dMD1EX/Htp1YCk5G29LJWirJfjRiy | 2025-08-06 01:09:43 |
| 13 | sid10      | $2y$10$WQA.KyTzdrcE6asKzt7ize26LukmpCglvg235lrnKACa1K.b3YjhG | 2025-08-07 09:21:33 |
+----+------------+--------------------------------------------------------------+---------------------+
13 rows in set (0.000 sec)

MariaDB [DITISS2025]> |
```

5. CONCLUSION

5.1 Conclusion

The project successfully demonstrates the design and implementation of a secure, scalable, and automated enterprise network infrastructure using open-source technologies. By integrating key components such as pfSense for firewall and routing, Jenkins for CI/CD automation, Nagios for monitoring, Suricata for intrusion detection, and Apache and Mariadb/PostgreSQL for application hosting, the network ensures high availability, efficient performance, and strong security. The segmentation of services using DMZ and VLANs minimizes the attack surface, while automated deployment and monitoring significantly reduce administrative overhead and response time.

Overall, the system reflects a real-world enterprise environment that balances performance, automation, and cybersecurity.

5.2 Future Scope

Docker is majorly considered as a best solution for service availability. It can be attached to implement continuous integration and continuous development i.e., CICD. It can be used in development when the software is getting develop this tool can used there to continuous security evaluation of software so that developers can program it more securely. As we have developed this tool for small scale. In future It can be used for large scale.

REFERENCES

Paper 1: - Research paper1--Enterprise Network Design and Security Optimization

Author: Kavita Saini