TENSHI GUARD

Endpoint Security Monitoring System

Submitted By: CYSE2

Members: Ogheneochuko Victor Adjene

: Samrit Shrestha

: Greeshma Melattu Joyson

: Sarath Saseendran Pillai

: Vikas Kumar

Loyalist College, North York, Toronto

Cyber Security

Table of Contents

1. Abstract

With the growing dependence on digital systems, small businesses are becoming more vulnerable to cyber threats, malware, and unauthorized access. While enterprise-grade tools like CrowdStrike, Symantec, and Microsoft Defender provide strong protection, they are often expensive and complex, making them impractical for small organizations, clinics, or schools.

This project introduces an Endpoint Security Monitoring System (ESMS) based on Wazuh**,** designed to deliver affordable, user-friendly, and real-time security monitoring for small-scale operations. The Wazuh Manager**,** deployed on a Kali Linux virtual machine, acts as the central monitoring server. Wazuh agents installed on client systems continuously track system activity, network traffic, and potential threats. All alerts and logs are sent to the manager, allowing the service provider to detect and respond to incidents quickly.

Each client receives a unique user ID and password to access a personalized dashboard showing live alerts and system status. The dashboards are customized for different sectors—such as clinics, educational institutions, and restaurants—making security insights easy to understand, even for non-technical users.

By combining open-source technology with simple, real-time monitoring, ESMS offers a cost-effective alternative to complex enterprise solutions, helping small businesses protect their digital environments efficiently and independently.

2. Introduction

An Endpoint Security Monitoring System (ESMS) is designed to protect and monitor devices such as computers, servers, mobile phones, and IoT systems. Since these devices are common targets for cyberattacks, continuous monitoring is essential for keeping an organization's network secure. ESMS works by tracking system activity in real time and detecting suspicious behavior before it causes harm.

Endpoint security acts as the first layer of defense. It provides visibility and control over all connected devices, helping organizations spot unusual activity, prevent malware or ransomware attacks, and safeguard sensitive data. It also helps meet regulatory requirements such as GDPR, HIPAA, and PCI DSS, which are important for maintaining compliance and trust.

The system collects data through agents installed on endpoint devices. These agents gather information about user actions, system processes, and network activity, which is then analyzed using rules, behavioral analytics, and AI techniques. When a threat is detected, alerts are sent to security teams or SIEM platforms, and some systems can automatically isolate compromised devices or block malicious actions.

Modern endpoint monitoring goes beyond traditional antivirus tools, which only detect known threats. Advanced solutions like EDR and XDR use behavior-based detection and AI to identify both known and unknown attacks. This provides stronger visibility, better scalability, and faster automated response, making ESMS an essential part of today's cybersecurity strategies.

3. Literature Review

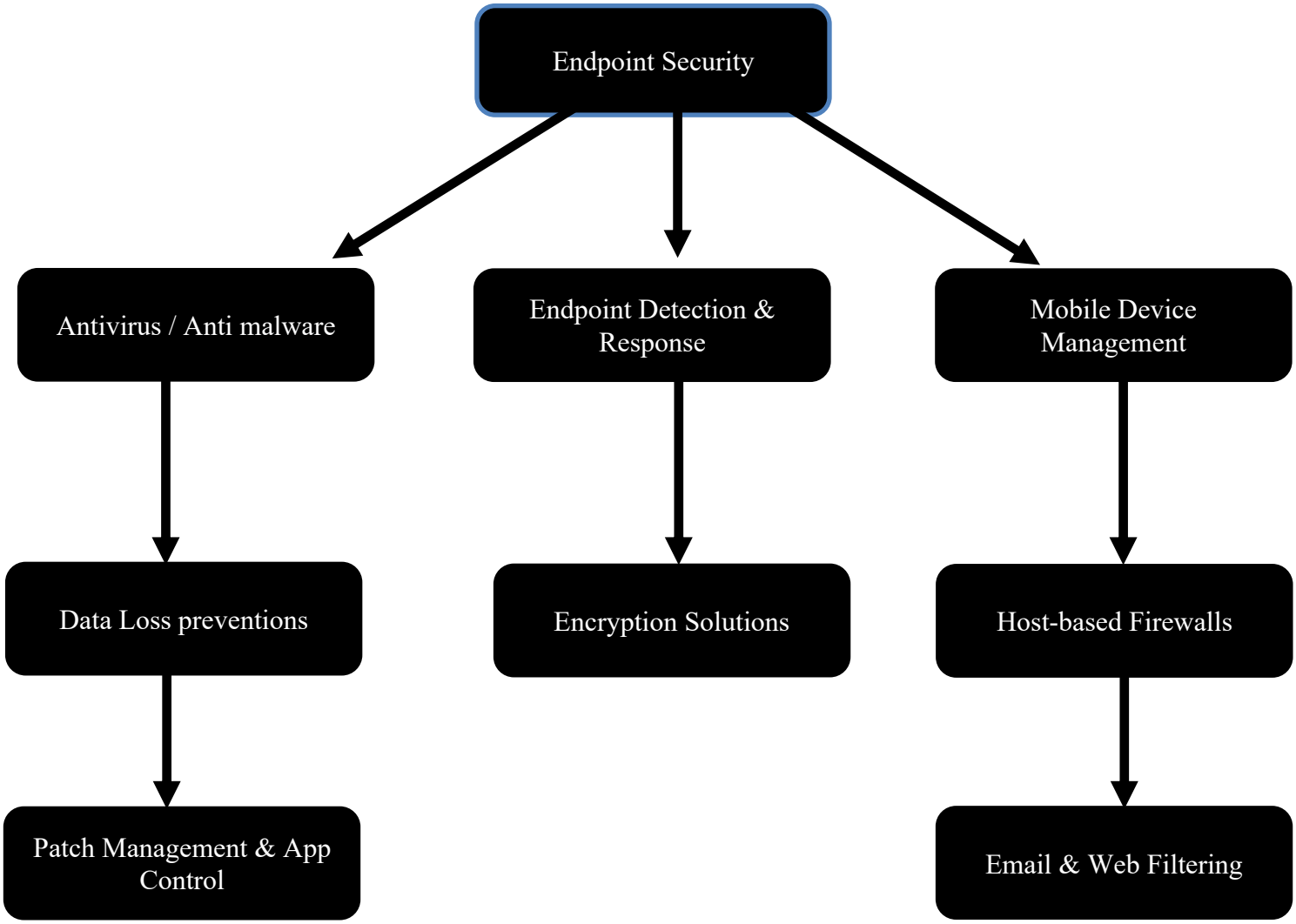
a. What is Endpoint Security?

Endpoint security refers to the practice of securing end-user devices such as laptops, desktops, smartphones, servers, and Internet of Things (IoT) devices that connect to a network. Each device that communicates with a network represents an endpoint, and these devices are often the primary target for attackers seeking to gain unauthorized access, steal sensitive data, or disrupt operations.

Traditionally, endpoint protection meant using simple antivirus software to detect and remove malware. However, with the evolution of cyberattacks, modern endpoint security has expanded to include Endpoint Detection and Response (EDR), Data Loss Prevention (DLP), firewalls, encryption, and Mobile Device Management (MDM) systems (Symantec, 2021).

In simple terms, endpoint security provides a defensive shield for organizations, ensuring that no single device can become the weak point through which cybercriminals compromise the entire network.

b. Types of Endpoint Security Monitoring system (Flow Chart)



(Flow chart 1: Types of endpoint security monitoring system)

Modern endpoint security systems include a variety of tools and technologies:

Tool	What It Does (Simple)	Examples / Notes
Antivirus / Anti-malware	Detects and removes viruses, ransomware, and other malware.	Norton, McAfee, Bitdefender. Works best on known threats.
Endpoint Detection & Response (EDR)	Monitors continuously and detects suspicious behavior. Can isolate infected devices.	CrowdStrike, SentinelOne. Great for new/unknown attacks.
MDM / UEM	Manages and secures mobile devices and laptops. Enforces device policies.	Intune, Workspace ONE.
Data Loss Prevention (DLP)	Stops sensitive data from leaving the device accidentally or intentionally.	Protects email, USB, cloud uploads.
Host-based Firewall	Filters and monitors network traffic on the device itself. Blocks unauthorized access.	Built into OS or added tools.
Encryption Solutions	Encrypts data so it can't be read if the device is lost or stolen.	BitLocker, VeraCrypt.
Email & Web Filtering	Blocks phishing, spam, and harmful websites before reaching users.	Often part of security suites.
Patch Management / App Control	Keeps software updated and blocks unapproved apps.	Reduces vulnerabilities.

c. Components of an Endpoint Security Monitoring System

An Endpoint Security Monitoring System is composed of several key components that work together to protect, detect, and respond to threats across all connected devices within a network.

The Endpoint Agent is the core component installed on user devices such as computers, servers, or mobile devices. It continuously collects data about system activities, network connections, file integrity, and application behavior. The Central Management Console acts as the command center where security administrators can monitor all endpoints, deploy policies, and view alerts in real time.

The Threat Detection Engine analyzes collected data using predefined rules, behavioral analysis, and sometimes machine learning to identify suspicious activities. The Data Collection and Log Analysis Module aggregates and correlates logs from endpoints, firewalls, and other network devices to identify patterns of compromise.

The Incident Response Module allows administrators to isolate infected endpoints, remove malicious files, or block network access to contain threats. Lastly, the Reporting and Alerting System provides real-time notifications and detailed analytics dashboards for decision-making and compliance purposes.

Together, these components ensure a continuous cycle of prevention, detection, and response, maintaining the security and integrity of organizational endpoints.

d. Different Modules in an Endpoint Security Monitoring System

An Endpoint Security Monitoring System consists of multiple integrated modules that work together to ensure complete protection of organizational devices.

The Antivirus and Anti-Malware Module detect, quarantines, and removes malicious software. The Firewall Module monitors inbound and outbound traffic to block unauthorized access. The Intrusion Detection and Prevention Module (IDS/IPS) identify suspicious behavior and prevents exploitation attempts.

The Application Control Module restricts unauthorized applications from running on endpoints, while the Device Control Module manages external devices such as USB drives to prevent data leaks. The Patch Management Module ensures that all endpoints are updated with the latest security patches to minimize vulnerabilities. The Data Encryption Module protects sensitive information stored or transmitted through the device.

Finally, the **Monitoring and Reporting Module** provides continuous visibility, generates alerts, and creates security reports to assist administrators in analyzing endpoint activities and incidents effectively.

e. Importance of endpoint security

The need for endpoint security arises because of the growing complexity of IT environments and the sophistication of modern cyberattacks. The key reasons are:

1. **Endpoints are prime attack targets** – A majority of cyberattacks (e.g., ransomware, phishing, spyware) begin at the endpoint level (CrowdStrike, 2023).
2. **Remote and hybrid work** – With employees connecting from home networks and personal devices, the attack surface has expanded dramatically.
3. **Protection of sensitive data** – Endpoints often store or process critical business data such as intellectual property, customer information, and financial records.

4. **Regulatory compliance** – Laws such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry Data Security Standard (PCI-DSS) require organizations to secure endpoints.
5. **Growing attack surface** – With the rise of IoT and Bring Your Own Device (BYOD) policies, organizations must secure thousands of diverse devices.
6. **Insider threats** – Malicious or negligent employees may leak sensitive information or accidentally install malware.

Without endpoint security, a single infected laptop or phone could act as a gateway for attackers to infiltrate the corporate network, spread malware, or steal confidential data.

f. How is an Endpoint Security Monitoring system deployed or provisioned?

An Endpoint Security Monitoring System is deployed in three main layers:

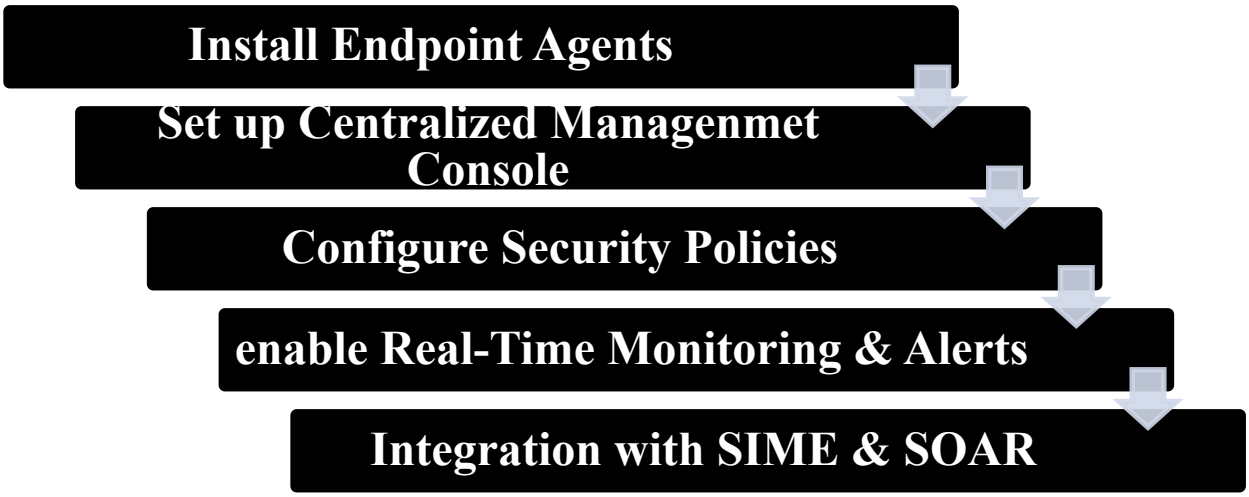
1. **Planning & Preparation Phase**

Before deployment, organizations need to plan:

(Flow chat 2: Steps in Planning and preparation phase of endpoint security monitoring system deployment)

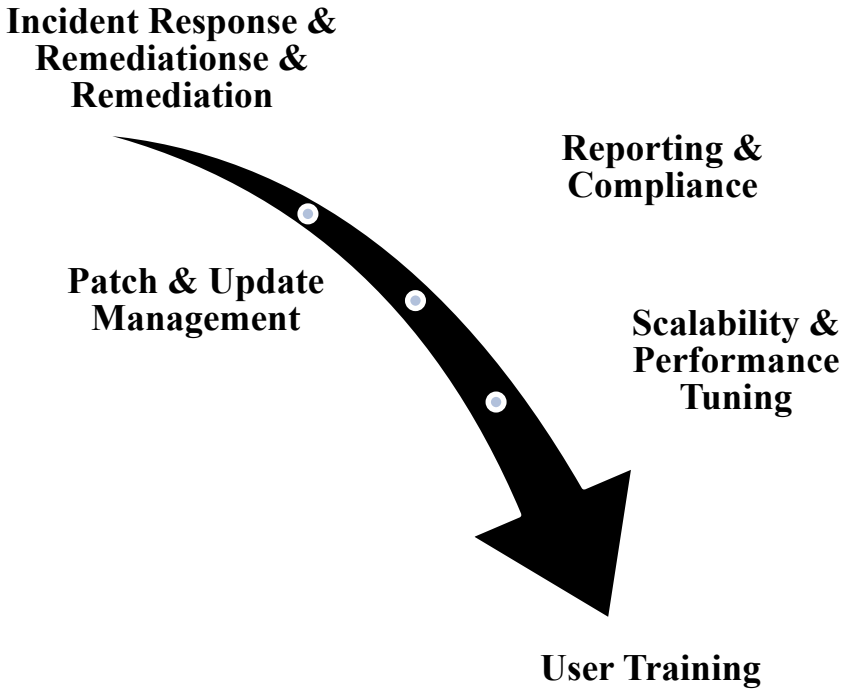
2. **Deployment & Provisioning Phase**

This is where the actual system is rolled out:



(Flow chart 3: Steps in Deployment & Provisioning Phase phase of endpoint security monitoring system deployment)

3. **Post-Deployment Phase (Operations & Maintenance)**



(Flow chart 4: Steps in Post-Deployment Phase of endpoint security monitoring system deployment)

4. **Provisioning Approaches**

1. **On-Premises**
    a. Installed on company servers.
    b. More control but requires more IT resources.
2. **Cloud-Based**
    a. Managed by a provider (e.g., CrowdStrike, SentinelOne, Microsoft Defender for Endpoint).
    b. Easy to scale, less infrastructure overhead.
3. **Hybrid**
    a. Combines on-prem and cloud.
    b. Useful for large organizations with regulatory constraints.

g. When and where should we deploy an endpoint security monitoring system?

1. During Initial Infrastructure Setup:

Endpoint security monitoring should be implemented when the organization is first building its IT infrastructure. This ensures devices are secured before they are exposed to threats.

2. When Onboarding New Devices:

New laptops, desktops, mobile devices, or IoT systems must be secured with endpoint monitoring before connecting to the network.

3. Before Handling Sensitive or Regulated Data:

Deploy monitoring systems before processing sensitive financial, healthcare, or customer data to meet compliance requirements like GDPR, HIPAA, or PCI DSS.

4. After Security Incidents:

If an organization experiences malware infections, phishing attacks, or insider threats, endpoint monitoring should be deployed immediately to provide real-time detection.

5. In Remote/Hybrid Work Scenarios:

Remote employees using personal devices or public Wi-Fi are high-risk, so monitoring is critical in such cases.

h. Which one to use?

1. Wazuh (Open Source)

Wazuh is a free and open-source security platform that unifies XDR and SIEM protection for endpoints and cloud workloads.

a. Intrusion detection

Wazuh agents scan the monitored systems looking for malware, rootkits and suspicious anomalies. They can detect hidden files, cloaked processes or unregistered network listeners, as well as inconsistencies in system call responses. In addition to agent capabilities, the server component uses a signature-based approach to intrusion detection, using its regular expression engine to analyze collected log data and look for indicators of compromise.

b. Log data analysis

Wazuh agents read operating system and application logs and securely forward them to a central manager for rule-based analysis and storage. When no agent is deployed, the server can also receive data via syslog from network devices or applications. The Wazuh rules help make you aware of application or system errors, misconfigurations, attempted and/or successful malicious activities, policy violations and a variety of other security and operational issues.

c. File integrity monitoring

Wazuh monitors the file system, identifying changes in content, permissions, ownership, and attributes of files that you need to keep an eye on. In addition, it natively identifies users and applications used to create or modify files. File integrity monitoring capabilities can be used in combination with threat intelligence to identify threats or compromised hosts. In addition, several regulatory compliance standards, such as PCI DSS, require it.

d. Vulnerability detection

Wazuh agents pull software inventory data and send this information to the server, where it is correlated with continuously updated CVE (Common Vulnerabilities and Exposure) databases, in order to identify well-known vulnerable software. Automated vulnerability assessment helps you find the weak spots in your critical assets and take corrective action before attackers exploit them to sabotage your business or steal confidential data.

e. Configuration assessment

Wazuh monitors system and application configuration settings to ensure they are compliant with your security policies, standards and/or hardening guides. Agents perform periodic scans to detect applications that are known to be vulnerable, unpatched, or insecurely configured. Additionally, configuration checks can be customized, tailoring them to properly align with your organization. Alerts include recommendations for better configuration, references and mapping with regulatory compliance.

f. Incident response

Wazuh provides out-of-the-box active responses to perform various countermeasures to address active threats, such as blocking access to a system

from the threat source when certain criteria are met. In addition, Wazuh can be used to remotely run commands or system queries, identifying indicators of compromise (IOCs) and helping perform other live forensics or incident response tasks.

g. Regulatory compliance

Wazuh provides some of the necessary security controls to become compliant with industry standards and regulations. These features, combined with its scalability and multi-platform support help organizations meet technical compliance requirements. Wazuh is widely used by payment processing companies and financial institutions to meet PCI DSS (Payment Card Industry Data Security Standard) requirements. Its web user interface provides reports and dashboards that can help with this and other regulations (e.g. GPG13 or GDPR).

h. Cloud security

Wazuh helps monitoring cloud infrastructure at an API level, using integration modules that are able to pull security data from well-known cloud providers, such as Amazon AWS, Azure or Google Cloud. In addition, Wazuh provides rules to assess the configuration of your cloud environment, easily spotting weaknesses. IN addition, Wazuh lightweight and multi-platform agents are commonly used to monitor cloud environments at the instance level.

i. Containers security

Wazuh provides security visibility into your Docker hosts and containers, monitoring their behavior and detecting threats, vulnerabilities and anomalies. The Wazuh agent has native integration with the Docker engine allowing users to monitor images, volumes, network settings, and running containers. Wazuh continuously collects and analyzes detailed runtime information. For example, alerting for containers running in privileged mode, vulnerable applications, a

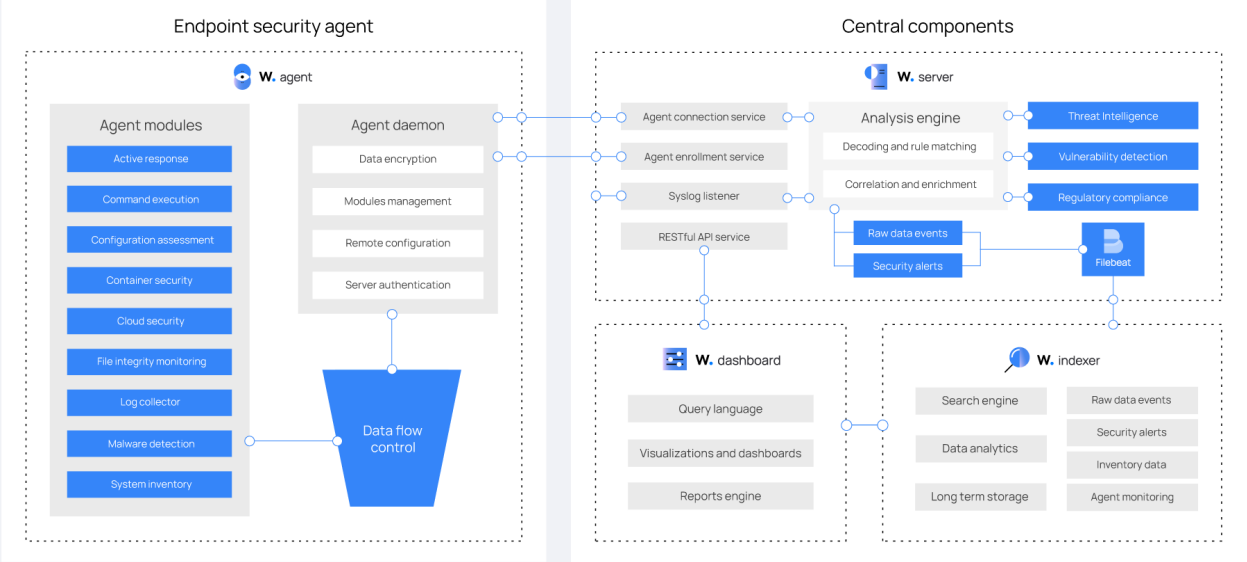
shell running in a container, changes to persistent volumes or images, and other possible threats.

j. How thread detection works in Wazuh

- **Signature-Based Detection:**
    - Malware Detection: Wazuh agents monitor files on endpoints and can detect malicious files by comparing their signatures (e.g., hashes) against known malware threat intelligence indicators stored in CDB lists.
    - Rootkit Detection: The Root check module, enabled by default on monitored endpoints, periodically scans for anomalies and known rootkit signatures at both kernel and user space levels.
- **Behavioral and Anomaly Detection:**
    - Rootkit Behavior Detection: Beyond signatures, root check analyzes system behavior for patterns indicative of rootkit activity, such as modifications to critical system files or unusual system calls.
    - Log Data Analysis: Wazuh collects and analyzes security telemetry from various sources (endpoints, network devices, applications). This data is then compared against predefined rules and decoders to identify suspicious activities and potential threats.
    - Anomaly Detection: Wazuh can leverage integrations like the OpenSearch Anomaly Detection plugin to identify deviations from established baselines in endpoint or network behavior, potentially uncovering new or evolving threats.
- **File Integrity Monitoring (FIM):**
    - Wazuh monitors critical files and configurations for unauthorized or unexpected changes, alerting administrators to potential tampering or compromise.

- **Security Configuration Assessment (SCA):**
  - Wazuh assesses the security posture of endpoints by comparing their configurations against predefined policies and benchmarks, identifying misconfigurations that could expose systems to threats.
- **Vulnerability Detection:**
  - The Vulnerability Detection module analyzes software inventory data from endpoints and correlates it with vulnerability information from the Wazuh Cyber Threat Intelligence (CTI) platform to identify known vulnerabilities (CVEs) on monitored systems.
- **Custom Rules and Decoders:**
  - Users can create custom decoders and rules to extend Wazuh's detection capabilities, enabling the identification of specific threats or malicious activities tailored to their environment.

k. Wazuh Workflow Diagram



(Figure 1: Workflow diagram of Wazuh)

I. Endpoint Security Agent (on each device)

- Agent Modules (blue boxes on the left): These are mini-tools that run on endpoints (like laptops, servers, containers, cloud VMs). They do things like:
    - Detect malware
    - Monitor files for tampering
    - Collect logs
    - Run configuration and compliance checks
    - Provide system inventory
    - Trigger active responses (like blocking processes)
- Agent Daemon: Works in the background to:
    - Encrypt and send data securely
    - Manage and update modules
    - Authenticate with the server
    - Handle remote configuration
- Data Flow Control: Collects and forwards all this information to the central server.

II. Central Server (Wazuh Server)

- Agent Services:
    - Connects and registers new agents
    - Listens for incoming logs (Syslog)
    - Provides REST API for communication
- Analysis Engine:
    - Decodes incoming data
    - Matches against rules (e.g., detect suspicious activity)
    - Correlates multiple events to detect threats
    - Enriches data with context (like IP reputation, known threats)

- Threat Intelligence / Vulnerability Detection / Compliance: Extra features to check if activity matches known attacks, vulnerabilities, or compliance rules (like HIPAA, PCI-DSS, etc.).

III. Dashboard (Visualization Layer)

- Let's analysts search logs, visualize alerts, and generate reports.
- Provides custom dashboards (e.g., one for healthcare, one for academics, one for restaurants in your case).

IV. Indexer (Storage and Search Engine)

- Stores all collected data (raw logs, alerts, inventory info).
- Provides **fast searching and data analytics**.
- Keeps long-term records for audits and compliance.

2. CrowdStrike Falcon

- **Best for:** Enterprises needing strong cloud-native EDR (Endpoint Detection & Response).
- **Pros:**
  - Lightweight, cloud-delivered agent
  - Real-time threat intelligence and detection
  - Strong EDR and threat hunting capabilities
- **Cons:**
  - Subscription-based, expensive for small businesses

3. Microsoft Defender for Endpoint

- **Best for:** Companies already in the Microsoft ecosystem.
- **Pros:**
  - Deep integration with Microsoft 365
  - Good at malware detection and vulnerability management
  - Automated investigation and remediation

- **Cons:**
    - Limited visibility outside Microsoft ecosystem
    - Costs can add up with licensing

4. SentinelOne

- **Best for:** Organizations wanting AI-driven endpoint protection.
- **Pros:**
    - Autonomous AI-powered EDR/XDR
    - Strong ransomware defense
    - Rollback features (undo malicious changes)
- **Cons:**
    - Higher cost than open-source options
    - Some false positives

5. Sophos Intercept X

- **Best for:** SMBs and mid-sized businesses.
- **Pros:**
    - Good ransomware protection
    - Root cause analysis
    - Cloud-managed dashboard
- **Cons:**
    - Requires additional modules for full SIEM functionality

i. Endpoint Security Challenges

Table :1

Endpoint Security Challenges	
Diverse Endpoint Devices	○ Organizations must secure a wide variety of devices such as laptops, mobile phones, tablets, IoT devices, and servers. ○ Each device type has different operating systems, applications, and security requirements, making consistent protection difficult.
Remote Work and BYOD (Bring Your Own Device)	○ With employees working remotely and using personal devices, endpoints are often outside the secure corporate perimeter. ○ This increases risks of insecure Wi-Fi, unpatched systems, and lack of visibility for security teams.
Patch Management and Software Updates	○ Delayed or inconsistent patching leaves endpoints vulnerable to exploitation. ○ Attackers often exploit known vulnerabilities that remain unpatched.
Malware and Ransomware	○ Endpoints are frequent targets for malware, phishing, and ransomware attacks. ○ Advanced malware can bypass traditional antivirus tools using fileless techniques or obfuscation.

Data Loss and Insider Threats	○ Endpoints are often the first-place sensitive data is stored, transferred, or accessed. ○ Lost or stolen devices can lead to major data breaches. ○ Malicious or careless insiders can intentionally or accidentally compromise endpoint security.
Visibility and Monitoring	○ ○ Security teams often lack real-time visibility into all endpoint activity, especially across distributed environments. ○ Without centralized monitoring, it becomes difficult to detect anomalies and respond quickly.
Scalability of Security Tools	○ As organizations grow, scaling endpoint protection across thousands of devices can be resource-intensive. ○ Ensuring uniform policies and enforcement across multiple regions is a challenge.
Integration with Other Security Systems	○ Endpoints must integrate with SIEMs, firewalls, and intrusion detection systems. ○ Poor integration can result in blind spots and slow incident response.

h. Advantages & Disadvantages of Traditional Systems

Advantages

1. **Proven and Widely Used**
    - Traditional systems like antivirus and firewalls have been industry standards for decades.
    - They are well understood by IT teams, with established best practices.
2. **Basic Threat Detection**
    - Effective at identifying **known malware** and preventing common attacks.
    - Signature-based detection can quickly stop threats with existing definitions.
3. **Low Cost and Availability**
    - Many solutions are affordable, with both free and commercial options.
    - Easy to deploy across desktops, laptops, and servers.
4. **Regulatory Alignment**
    - Basic endpoint protection (like antivirus) is often required for compliance with **HIPAA, PCI DSS, GDPR**, and other standards.
5. **Low Learning Curve**
    - Since they've been around for a long time, IT staff generally require little specialized training to operate them.

Disadvantages

1. **Signature Dependency**
    - Traditional antivirus relies on **known signatures** and cannot effectively detect **zero-day threats** or advanced malware.
2. **Limited Visibility and Response**
    - Focuses mainly on prevention, with little to no **endpoint detection and response (EDR)** capability.

- Provides limited context for incident investigations.

3. **Inadequate Against Modern Threats**
   - Struggles with **fileless malware, polymorphic viruses, and advanced persistent threats (APTs)**.
   - Attackers easily bypass these tools using evasion techniques.
4. **Performance Impact**
   - Signature scanning can slow down devices, consuming CPU and memory resources.
5. **Poor Adaptability**
   - Designed for traditional, perimeter-based networks, not modern **cloud, remote work, or IoT** environments.
   - Does not scale effectively for hybrid infrastructures.

i. Comparative Analysis: Traditional vs ML-based detection

Table :2

Feature / Aspect	Traditional Detection (Signature- or Rule-based)	ML-based Detection (Anomaly, Behavior, Learning Models)
Known vs Unknown Threats	Very good at detecting *known threats* using signatures/rules. Poor / blind to new/zero-day attacks. (MDPI)	Better at detecting novel or previously unseen threats by learning "normal" behavior and spotting anomalies. (MDPI)
False Positives / False Negatives	Usually low false positives for known threats, but misses when attackers change signatures. Rules are rigid. (MDPI)	Can detect unknown threats but often produce more false positives / need lots of training/tuning. (Frontiers)
Adaptability / Scalability	Less adaptive. Need manual updates of signatures or rules. Doesn't scale well with evolving	More scalable and adaptive. ML algorithms can retrain, adjust, work with large data volumes. (MDPI)

	threats. (MDPI)	
Computational Resources & Complexity	Lower resource requirement. Less computational overhead. Easier to interpret. (MDPI)	Higher resource usage (training, feature extraction, inference), more complex to deploy and maintain. Model interpretability is harder. (Frontiers)
Real-Time / Near Real-Time Detection	Can be fast for known threats (signature matching), but often delayed for emerging ones.	ML can potentially offer near real-time detection, depending on data, model, and infrastructure. (MDPI)
Data Requirements & Training	Needs signature databases/rules maintained by humans. Less data needed.	Needs large labeled and unlabeled datasets (both benign and malicious), data preprocessing, feature engineering. (SpringerOpen)
Interpretability & Trust	Very interpretable — rule-based decisions are easier to trace. Analysts can see which rule triggered.	Sometimes opaque ("black-box" models), which can make trust, validation, and regulatory compliance challenging. (Frontiers)

j. How endpoints are attacked

1. Attack stages

1. **Reconnaissance** — Attacker gathers info about users, software versions, open services, public IPs, posted credentials, etc.
2. **Initial access / Delivery** — Malicious email attachment, phishing link, drive-by download, compromised website, malicious USB, or exploiting an exposed service.
3. **Exploitation** — Attacker exploits a vulnerability (unpatched OS/app, misconfigured service) or succeeds with credential-based access (stolen/password reuse).

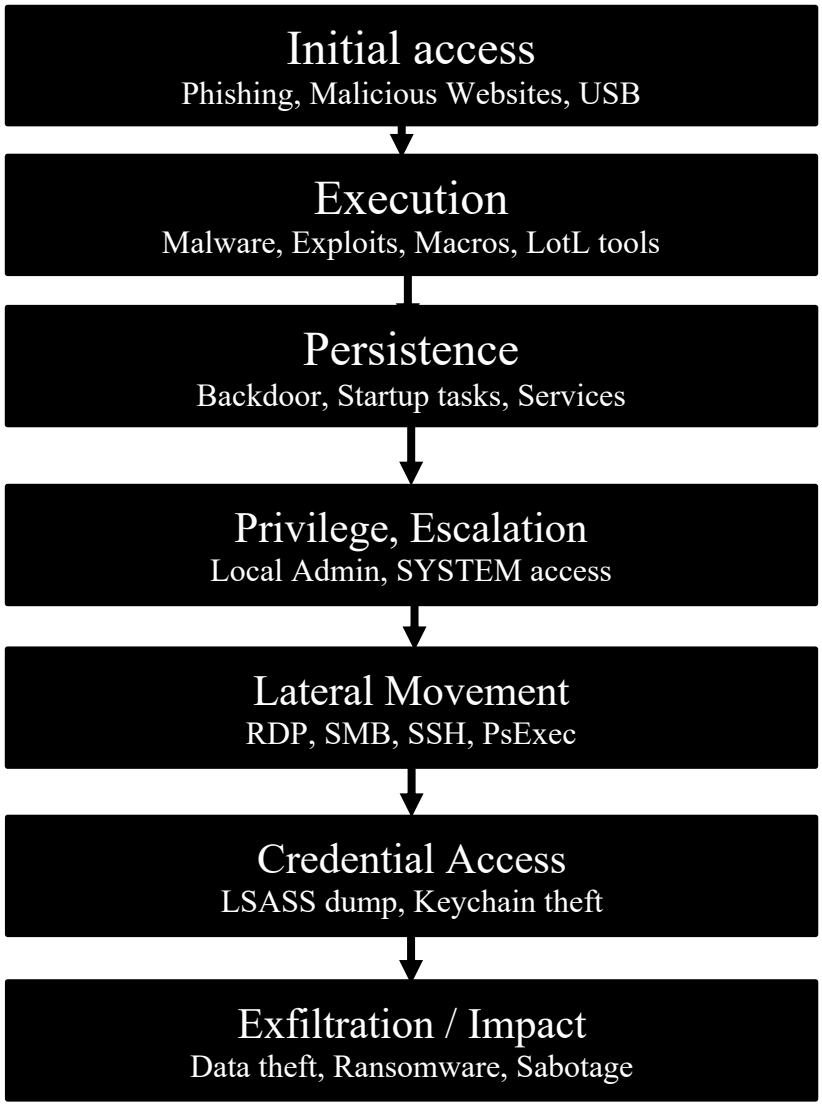
4. **Execution** — Malicious code runs on the endpoint (ransomware binary, script, macro, shellcode, living-off-the-land tool like PowerShell).
5. **Persistence** — Attacker installs backdoors, scheduled tasks, services, or modifies startup items so access survives reboots.
6. **Privilege Escalation** — Attacker gains higher privileges (local admin, SYSTEM) using exploits or credential theft.
7. **Lateral Movement** — Using harvested credentials, Remote Desktop/SMB/SSH, PsExec, WMI, etc., the attacker moves to other endpoints.
8. **Discovery & Credential Access** — Attacker enumerates network resources, user lists, credential stores (LSASS, keychains).
9. **Exfiltration / Impact** — Data is compressed/encrypted and exfiltrated or systems are encrypted (ransomware) / sabotage begins.
10. **Covering Tracks** — Logs are tampered with, malware obfuscated, tools removed to delay detection.

2. Common attack vectors

- **Phishing & Malicious Attachments** — Email with weaponized Office macro or malicious link.
- **Drive-by / Watering-hole** — User visits a legitimate site that's been compromised and downloads malware.
- **Exploit of Unpatched Vulnerability** — CVE in an endpoint app or OS used to run code remotely.
- **Credential Theft / Password Reuse** — Reused password from a breached site or harvested via phish.
- **Removable Media (USB)** — Autorun/malicious binary on USB stick.
- **Supply-chain compromise** — Malicious update from a third-party app.

- **Living-off-the-Land (LotL) Abuse** — Using built-in tools (PowerShell, WMI, certutil) to avoid AV detection.
- **Malicious Containers / Misconfigured Cloud Agents** — Compromised container images or cloud VMs with exposed creds.

3. Endpoint Attack Lifecycle



(Flow chart 4: Steps in endpoint attack lifecycle)

4. **Attack examples mapped to real techniques**

- **Phishing → Macro-enabled doc → PowerShell download** (Execution via LOTL).
- **Unpatched SMB/Remote Desktop exploit → Remote code execution → Ransomware**.
- **Credential stuffing (reused passwords) → Remote login → Data theft**.
- **Compromised developer package → supply-chain implant → widespread endpoint compromise**.

These map directly to MITRE ATT&CK categories such as Initial Access, Execution, Persistence, Privilege Escalation, Lateral Movement, Credential Access, and Exfiltration.

5. **How attackers evade detection**

- Use obfuscation and packers to hide malware signatures.
- Use fileless techniques (scripts, memory-only implants) to avoid disk-based scanning.
- Use legitimate admin tools (LotL) to blend in with normal activity.
- Throttle exfiltration or blend with normal traffic to bypass anomaly thresholds.
- Delete or tamper logs to delay forensic analysis.

6. what you should implement on endpoints

Table: 3

Attack Stage	Practical Controls (endpoint-focused)
Reconnaissance / Initial Access	Phishing awareness training; email filtering; web filtering; URL sandboxing.
Exploitation	Timely patch management; application whitelisting; vulnerability scanning.
Execution	EDR with behavioral detection; restrict/monitor PowerShell, scripting hosts; application control.
Persistence	Monitor autoruns/startup items; restrict service installs; use EDR to detect new persistence artifacts.
Privilege Escalation	Enforce least privilege; remove local admin from users; MFA for privileged actions.
Lateral Movement	Segment networks; restrict admin tools; log and alert on lateral protocols (SMB/RDP).
Credential Access	Credential protection (LSASS protection); rotate credentials; use password vaults; MFA.
Exfiltration	DLP controls; network egress monitoring; anomaly detection on data flows.
Post-incident / Forensics	Centralized logging (SIEM); immutable logs; endpoint snapshots; incident playbooks.

k. How to secure Endpoints

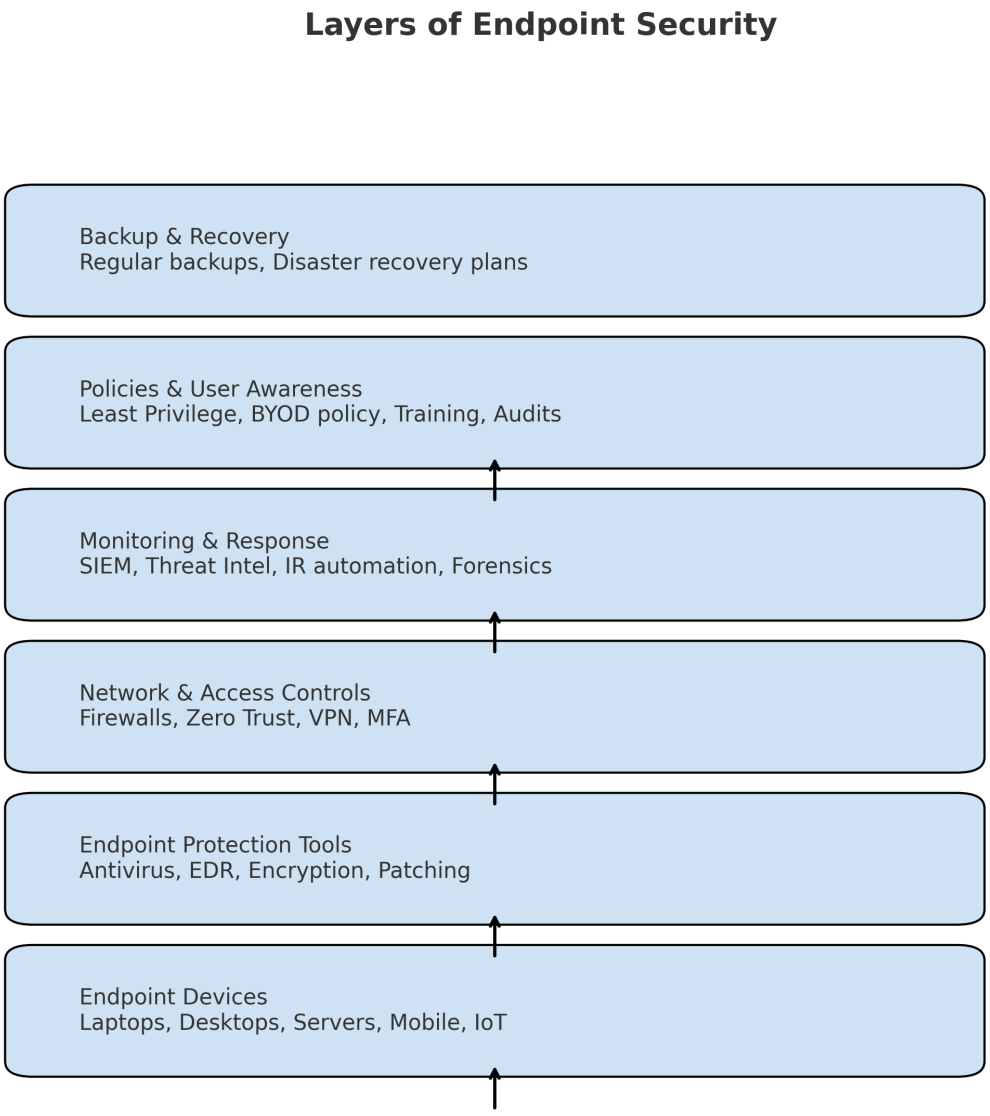


Figure 2: Ai generated flow chart of layers of endpoint Security

l. Different Business Needs for Endpoint Security Monitoring Systems

The security requirements of organizations vary widely depending on their size, operations, and regulatory environment. Each business sector faces different challenges and risks, making it necessary to tailor endpoint security monitoring systems accordingly.

Small and medium-sized businesses (SMBs) typically operate with limited IT budgets and staff. Their primary focus is on affordability, simplicity, and automation. They require endpoint solutions that offer centralized control, real-time malware detection, and easy deployment without extensive hardware or configuration. Cloud-based endpoint security platforms are ideal for such organizations as they reduce maintenance costs while providing continuous protection.

Large enterprises**,** in contrast, manage thousands of endpoints distributed across multiple networks and geographical regions. They need scalable systems that support advanced features such as endpoint detection and response (EDR), threat intelligence integration, behavioral analysis, and automated incident response. These businesses often integrate their endpoint monitoring systems with broader Security Information and Event Management (SIEM) tools for comprehensive visibility and reporting.

Healthcare organizations demand strong endpoint protection due to the sensitivity of patient data and compliance with privacy regulations like HIPAA**.** Their systems must include encryption, intrusion prevention, and continuous monitoring to prevent data breaches. Similarly, financial institutions prioritize fraud detection, encryption of financial records, and protection from phishing and ransomware attacks.

Educational institutions require flexible endpoint monitoring to secure open networks shared by students and faculty, often across multiple campuses. Meanwhile, government agencies and critical infrastructure sectors such as energy, transportation, and defense—need advanced protection against sophisticated and persistent cyber threats. These sectors rely on multi-layered security architectures with strict access control and real-time alerting.

Ultimately, an effective endpoint security monitoring system must adapt to these diverse business environments, providing tailored protection, regulatory compliance, and scalability to meet evolving cybersecurity needs

m. Alternative Solution if Wazuh is Unavailable

In the absence of Wazuh, several alternative tools can be implemented to maintain effective security monitoring, threat detection, and incident response. One viable option is OSSEC**,** an open-source host-based intrusion detection system that provides comprehensive log analysis, file integrity monitoring, rootkit detection, and active response capabilities. As the original foundation for Wazuh, OSSEC can deliver many of the same core security functions.

Another robust alternative is Security Onion**,** a Linux-based platform that integrates tools such as Zeek**,** Suricata**,** and the Elastic Stack (Elasticsearch, Logstash, Kibana**)** to provide advanced network and host monitoring features. This solution enables both real-time traffic analysis and centralized event management.

For organizations focusing primarily on intrusion detection, tools such as Snort or Suricata can serve as reliable IDS/IPS solutions by inspecting network packets and identifying potential threats. Additionally, Graylog and Splunk can be

utilized for centralized log management, data visualization, and correlation of security events.

These alternatives collectively ensure that even without Wazuh, the organization can maintain visibility, integrity, and control over its cybersecurity operations.

n. Why Endpoint Security Monitoring Is Important Across Sectors

1. Healthcare Sector

A. Why Endpoint Security Monitoring Is Needed

Hospitals and clinics manage highly sensitive data such as patient medical histories, prescriptions, and billing details. Most of these systems are interconnected, meaning a single compromised endpoint could expose thousands of patient records. Without endpoint monitoring, attackers can exploit vulnerabilities, install malware, or steal data unnoticed for weeks.

B. Common Cyberattacks

- Ransomware: Encrypts patient data and demands payment for decryption.
- Phishing: Targets doctors or staff with fake emails to steal login credentials.
- Data Breaches: Unauthorized access to electronic health records (EHRs).
- Insider Threats: Employees misusing access privileges.
- IoT Device Exploits: Attackers targeting medical devices connected to hospital networks.

C. Recent Real-World Attacks

- Change Healthcare Ransomware Attack (2024): A ransomware attack on a U.S. healthcare payment system caused nationwide billing outages.
- HCA Healthcare Data Breach (2023): Over 11 million patients' data leaked online, including contact info and appointments.

- Scripps Health Ransomware Attack (2021): Forced the shutdown of hospital systems and delayed surgeries.

D. Key Features Needed for Healthcare

- Real-time malware and ransomware detection
- HIPAA compliance tracking
- Endpoint activity logs and access control
- Data encryption and secure communication
- Automated alerts for abnormal activities
- IoT device monitoring

2. Academic Sector

A. Why Endpoint Security Monitoring Is Needed

Universities and schools handle student data, research work, and financial records. Many of them use shared systems with limited IT resources, making them vulnerable to cyberattacks. Students and faculty often connect personal devices to the same network, creating multiple weak points for attackers.

B. Common Cyberattacks

- Phishing and Credential Theft: Targeting staff and students to access systems.
- Ransomware: Locks access to research data or student management systems.
- DDoS (Distributed Denial-of-Service): Disrupts online classes or exam portals.
- Data Breaches: Leaks of grades, financial data, or student records.
- Malware from Unauthorized Downloads: Students downloading infected content.

C. Recent Real-World Attacks

- Toronto District School Board (2023): Suffered a cyberattack that compromised internal systems.
- University of Michigan (2023): Shut down its campus network after a significant security incident.
- Lincoln College (Illinois, 2022): Closed permanently after a ransomware attack blocked access to critical data.

D. Key Features Needed for Academia

- Real-time endpoint and user activity monitoring
- FERPA compliance support for student data
- Anti-phishing protection and web filtering
- Secure login and MFA enforcement
- Endpoint usage reports and alerts
- Safe browsing control for students

3. Restaurant and Hospitality Sector

A. Why Endpoint Security Monitoring Is Needed

Restaurants and hotels rely on POS systems, reservation portals, and payment gateways, which constantly process credit card data and personal details. Most small businesses do not have dedicated IT security teams, making them easy targets for cybercriminals. Without endpoint monitoring, attackers can exploit outdated systems or steal customer payment data.

B. Common Cyberattacks

- POS (Point-of-Sale) Malware: Steals credit card information.
- Ransomware: Locks access to customer databases or booking systems.
- Phishing Attacks: Targets staff through fake vendor or delivery emails.
- Insider Data Theft: Employees accessing customer or payment data.
- Wi-Fi Exploits: Attacks through unsecured public Wi-Fi networks.

C. Recent Real-World Attacks

- Marriott International Data Breach (2022): Hackers stole data of over 20 GB, including customer payment info.
- McMenamins Restaurants (2021): Hit by ransomware affecting employee and payroll data.
- POS Malware Campaign (2023): Multiple restaurant chains were targeted in North America using malware to skim credit card data.

D. Key Features Needed for Restaurants

- PCI-DSS compliance monitoring for secure transactions
- POS and network activity tracking
- File integrity monitoring for customer databases
- Real-time alerts for suspicious payment behavior
- Threat detection for IoT and Wi-Fi devices
- Data backup and quick recovery tools

4. Value Proposition

While the market is saturated with robust endpoint security monitoring platforms like CrowdStrike, Symantec, and Microsoft Defender, most existing solutions are primarily designed for large enterprises, require advanced technical knowledge, and often involve significant licensing costs. Our system differentiates itself in several keyways, making it particularly suitable for small businesses and non-technical users:

a. **Sector-Specific Dashboards:** Unlike traditional endpoint monitoring systems that provide a generic view of threats and alerts, our system offers tailored dashboards for specific sectors, such as clinics, academic institutions, and restaurants. Each dashboard displays relevant metrics, alerts, and recommendations that align with the sector's unique security challenges. For example, a clinic dashboard highlights patient data access attempts and HIPAA compliance alerts, while a restaurant dashboard emphasizes POS system security and payment data protection.

b. **User-Friendly Interface:** Our platform prioritizes accessibility for non-technical users. The dashboards use simple visualizations, color-coded alerts, and plain-language explanations for security events. This ensures that even small business owners or staff without a cybersecurity background can monitor their endpoints effectively, reducing reliance on dedicated IT teams.

c. **Cost-Effective Solution:** Many enterprise-grade endpoint security platforms require expensive licenses and extensive hardware resources. Our system leverages Wazuh, an open-source security platform, to provide robust monitoring capabilities at minimal cost. This makes it financially viable for small businesses, clinics, and educational institutions to maintain a high level of endpoint security without the burden of large upfront investments.

d. **Lightweight and Flexible Deployment:** Our system is designed to be lightweight and flexible, capable of running on modest hardware and easily integrated into existing IT infrastructure. This contrasts with traditional solutions that often require complex configurations, specialized servers, and extensive IT support.
e. **Actionable Alerts and Reports:** Beyond detecting threats, our system generates actionable alerts and recommendations that guide the user in resolving security incidents. This proactive approach reduces response time and improves overall endpoint security posture without overwhelming users with technical jargon or unnecessary data.
f. **Focus on Small Businesses and SMEs:** While most existing platforms focus on large organizations, our system is specifically built to address the security needs of small businesses. By offering low-cost, easy-to-use monitoring tools with sector-specific insights, our solution fills a gap in the market for accessible, practical cybersecurity tools for small-scale operations.

In summary, our Endpoint Security Monitoring System bridges the gap between advanced security technology and small business accessibility. By providing tailored dashboards, user-friendly interfaces, and cost-effective deployment, it empowers non-technical users to actively protect their digital environments, making high-quality endpoint security achievable for organizations of any size.

5. Value Diagram

Client Requirements

- **Real-time detection of security threats and system anomalies.**
- **Centralized monitoring of multiple devices.**
- **Simple dashboards understandable by non-technical users.**
- **Compliance with data protection standards (HIPAA, FERPA, PCI-DSS).**
- **Low setup cost and minimal technical maintenance.**

Our Company's Work

- **Installed and configured Wazuh Manager on a secure virtual environment (Kali Linux).**
- **Deployed Wazuh agents on client systems to collect logs and send alerts.**
- **Built sector-specific dashboards for clinics, academic institutions, and restaurants.**
- **Integrated real-time alerting, reporting, and compliance tracking features.**
- **Designed a secure login system with individual IDs and passwords for clients.**
- **Developed an admin panel for subscription management, billing, and live alerts.**

Achieved Value

- **24/7 visibility**
- **Improved security awareness**
- **Faster incident response**
- **Regulatory compliance**
- **Reduced operational costs**
- **Enhanced trust and data protection**

a. Real-world fraud incidents caused by endpoint weaknesses

1. **Target Data Breach (2013) –** Hackers accessed the network through a third-party contractor's weak endpoint. Lack of monitoring allowed malware installation, exposing 40 million customers' payment data (Krebs, 2014).
2. **Equifax Breach (2017) –** An unpatched server endpoint vulnerability let attackers steal the personal data of 147 million people due to poor patch management (GAO, 2018).
3. **Colonial Pipeline Attack (2021) –** A compromised VPN endpoint without MFA led to a major fuel supply shutdown and fraud risks across the U.S. (U.S. House of Representatives, 2021).
4. **Bangladesh Bank Heist (2016) –** Malware on employee endpoints stole SWIFT credentials, leading to $81 million in fraudulent transfers (BBC News, 2016).

b. Existing fraud detection projects (banks, e-commerce, etc.)

1. **PayPal –** Uses machine learning and neural networks to detect unusual transaction patterns and prevent fake seller activities, saving millions annually.
2. **American Express –** Employs AI models and behavior analytics to monitor transactions in real time and reduce false fraud alerts.
3. **Amazon Fraud Detector (AWS) –** A cloud-based ML service that identifies fake reviews, account abuse, and suspicious transactions for e-commerce users.
4. **HSBC –** Uses big data and ML to detect money laundering and fraudulent transfers through pattern and network analysis.

7. System Design / Technical Details

a. Overview

TenshiGuard is a modular, multi-tenant endpoint security monitoring ecosystem designed to provide real-time threat visibility, alert management, and automated reporting for organizations of varying sizes and sectors. The system is composed of three primary modules: the Client Interface, the Company Dashboard, and the Company Website. Each module operates independently but connects through a shared authentication base, unified database schema, and REST API layer. The platform was built with scalability and cost-effectiveness in mind, ensuring that small institutions such as schools, clinics, and small businesses can access standard-grade security monitoring tools.

The detection backbone of the system is powered by Wazuh, an open-source SIEM and XDR framework. Wazuh was selected for its transparent architecture, broad operating system support, and ease of integration with other components such as Elasticsearch and Kibana. Alternative enterprise solutions such as CrowdStrike Falcon, Microsoft Defender for Endpoint, and SentinelOne were reviewed. Although these systems offer robust commercial-grade features, they are expensive and lack the open customization required for academic, experimental, and pilot projects. Wazuh's open-source model, flexible APIs, and modular scalability make it the most practical and adaptable option for TenshiGuard.

b. Key Modules

The Client Interface is the organization-facing portal that enables users to register their institution, manage devices, and monitor endpoint activity. It provides real-time dashboards showing CPU and memory utilization, agent connectivity, and detected threat patterns. It also includes tools for managing user roles, subscription status, and reporting. The interface is implemented using Flask and styled with Bootstrap and

Chart.js for responsive visual analytics. Its architecture is optimized for scalability and security, enabling multiple organizations to access their respective dashboards independently through a shared backend.

The Company Dashboard acts as the administrative and analytic platform for TenshiGuard's operations team. It provides cross-organizational visibility into subscription management, revenue tracking, threat event analytics, and system performance. It uses the same database schema and authentication logic as the Client Interface but operates under elevated privileges. The dashboard integrates background workers through Celery to manage report generation and asynchronous computations, ensuring that resource-intensive analytics do not interfere with live user sessions.

The Company Website serves as the public entry point into the ecosystem. It presents information about the TenshiGuard platform, its modules, and its pricing model. The website allows institutions to register and submit organizational details through secure forms that send data to the backend Flask API. It also serves as a hub for product documentation, tutorials, and onboarding materials. The site is built with Node.js, using RESTful communication to integrate with the backend database and authentication services.

c. Integration Plan

Section	Short Summary
Integration Architecture	Service-oriented design with separate UI, backend, database, and telemetry layers. All modules run in Docker containers and communicate through Flask APIs. Website manages registration, Client Interface shows alerts, and Dashboard provides admin control. Flask pulls data from Wazuh, caches in Redis, stores in PostgreSQL.
Data Synchronization	Unified database with key tables (Organization, User, Device, SecurityEvent). SQLAlchemy ensures safe transactions. All queries filtered by organization_id. Celery syncs Wazuh telemetry and updates PostgreSQL/Redis.
Security & Access Control	JWT-based authentication, role-based permissions, and subscription checks. All communication is over HTTPS/TLS 1.3 via Nginx. Secrets stored in .env files. Route-level decorators enforce access.
Testing & Deployment	Integration tested with PyTest and Postman. Docker Compose manages containers and health checks. CI builds images, runs tests, and deploys to staging before production.
Automation & Monitoring	Celery handles scheduled tasks (reports, telemetry sync, alerts). Wazuh monitors security events. Prometheus + Grafana show system health and performance metrics.

d. Roadmap

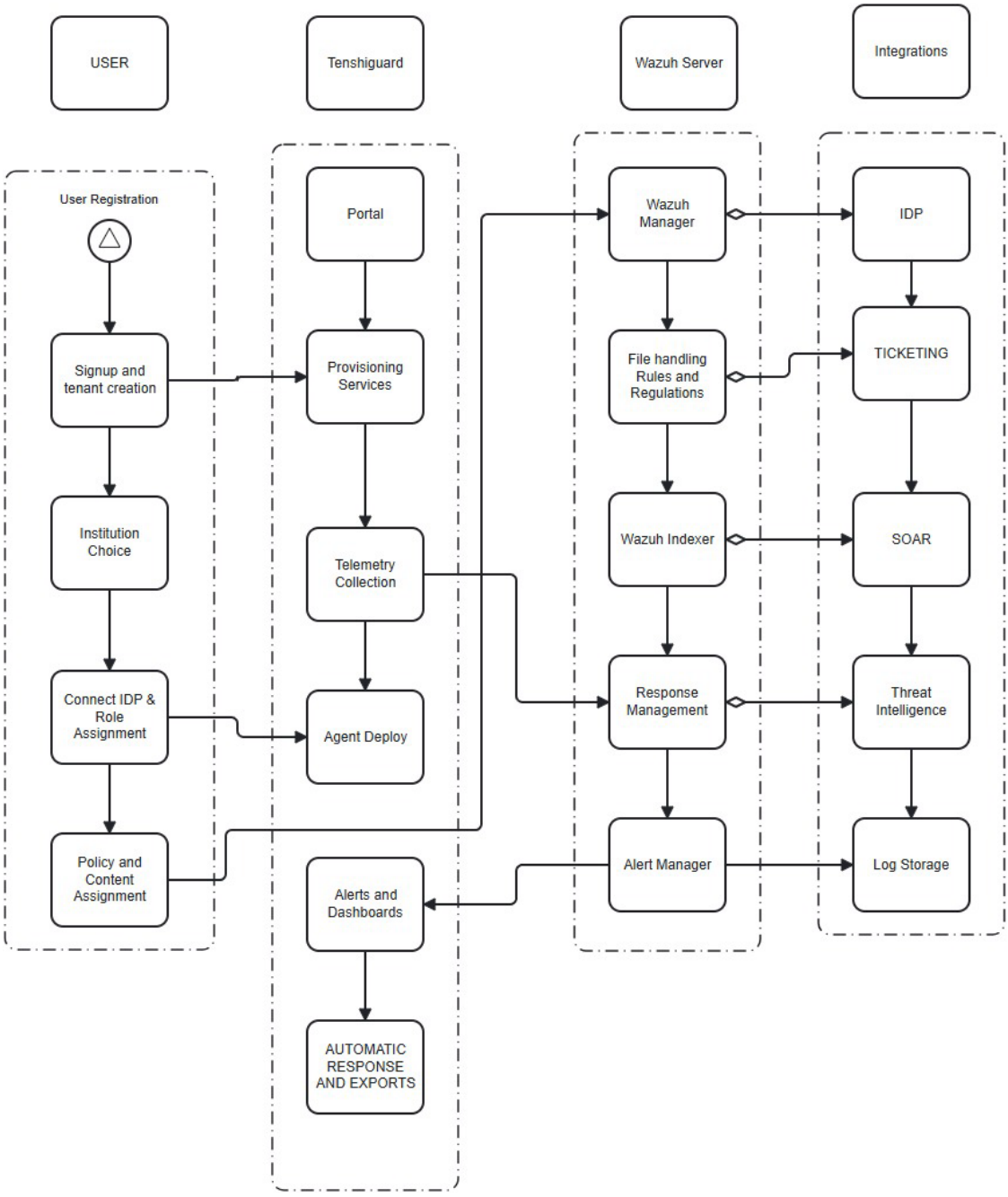
Phase	Short Description
Phase 1: Core Stability	Build the main system: Flask backend, database models, login/signup, and basic dashboards. JWT authentication, CRUD features, and Bootstrap UI created. Unit tests ensure everything works. Result: a stable working prototype.
Phase 2: Sector-Specific Dashboards	Create custom dashboards for industries like education, healthcare, and hospitality. Uses sector_config.json to change KPIs and themes. Redis improves speed. Testing ensures correct dashboard loads for each sector.
Phase 3: Subscription-Based Features	Add Basic, Pro, and Enterprise plans. Access to features depends on subscription. Celery checks subscription status automatically. Company Dashboard shows subscription details. Tests ensure users only access allowed features.
Phase 4: Real-Time Wazuh + Elastic Integration	Connect to live Wazuh telemetry. TenshiGuard pulls endpoint logs securely and indexes them in Elasticsearch. Dashboards display threat levels, heatmaps, and uptime. Celery handles data ingestion and Redis caches recent results. Stress tests ensure the system works for many clients.
Phase 5: Advanced Add-ons	Add automation, intelligent alerts, and compliance tools. Machine learning detects unusual behavior. System creates PDF reports automatically. Mobile-friendly design and multi-language support added.

e. Development Control Plan

Category	Details
Version Control and Branching	GitHub hosts the source repo. Branching model: main (stable), dev (integration), feature/* (new modules). Commits follow standardized format with module tags & issue references. PRs require two peer reviews. Versioning follows Semantic Versioning (MAJOR.MINOR.PATCH).
Environment & Configuration Management	Environments: development, staging, production. Config uses 12-factor principles; secrets stored in .env files. Docker Compose manages Flask, PostgreSQL, Redis, Celery, Nginx. Production runs Gunicorn behind Nginx, SSL via Certbot. Nightly AES-256 encrypted DB backups.
CI/CD (Continuous Integration & Deployment)	GitHub Actions automates testing, linting, building, and deploying. CI includes flake8, PyTest, Selenium. Only passing builds go to staging; production deployments require manual approval. Rollbacks done via redeploying previous image tag.
Testing & Quality Assurance	Includes unit, integration, performance, security testing. PyTest for backend, Locust for performance load testing, OWASP ZAP for vulnerability scans. Minimum 85% test coverage required. Regression & acceptance tests ensure backward compatibility.

Documentation & Governance	All technical docs stored in Confluence. Changes tracked via CRFs (Change Request Forms) detailing impact & justification. Approved changes map to GitHub issues. Weekly coordination meetings between dev, integration, and QA teams.
Deployment & Monitoring	Uses blue-green deployment to avoid downtime. Nginx load-balances between active & idle containers. Monitoring via Prometheus (metrics) and Wazuh (security log monitoring). Post-deployment monitoring lasts 48 hours with immediate escalation for issues.
Governance & Compliance	Aligns with ISO/IEC 27001 standards. All data encrypted in transit. MFA required for developer access. Complete audit logs for accountability. Focus on privacy, scalability, and resilience across development lifecycle.

f. Architecture of Endpoint Security Monitoring System



g. Tools & Technologies Used (Python, ML libraries, SIEM tools, EDR, etc.)

Technology	Description	Role in TenshiGuard Project
Flask (Python)	A lightweight, flexible Python micro-framework for building web applications and APIs. It provides essential tools without unnecessary complexity, supports RESTful API development, and integrates well with security and cryptography libraries.	Acts as the core backend framework. Handles routing, API communication, authentication, database interaction, user request processing, and security features like role-based access control.
SQLite / PostgreSQL	Relational database management systems (RDBMS). SQLite is lightweight and file-based, ideal for development. PostgreSQL is robust, scalable, and supports complex queries, concurrency, and enterprise-grade security.	SQLite is used for development/testing; PostgreSQL is used for production to support large datasets, multi-user environments, and secure organizational data management. Stores users, devices, alerts, logs, and configuration data.
Bootstrap 5	A modern front-end framework providing responsive layouts, pre-built UI components, and customizable styles. Lightweight and no jQuery dependency.	Ensures responsive, professional, and mobile-friendly UI across dashboards and pages. Speeds up UI development and maintains

		visual consistency throughout the TenshiGuard Client Interface.
Jinja2	A templating engine for Python that embeds logic within HTML and supports loops, conditions, and template inheritance.	Dynamically renders backend data (alerts, device status, user info) into web pages. Links Flask backend with UI, enabling interactive dashboards and data-driven pages. Improves code reuse and maintainability.

8. Methodology

The methodology adopted for the TenshiGuard project was designed to ensure a systematic, structured, and technically sound development process capable of producing a fully functional endpoint monitoring and analysis system. The approach combined both software engineering principles and iterative development cycles to enable continuous improvement, modular scalability, and integration with open-source security tools.

This methodology provided a framework for planning, design, implementation, testing, and deployment. Each phase was executed following strict development control procedures to maintain quality assurance, ensure reproducibility, and minimize integration risks. The project was guided by the principles of the Agile software development lifecycle and structured around the Waterfall-based documentation model for academic clarity.

a. Research and Design Approach

The project followed a hybrid Agile methodology. The initial stage focused on system research, requirement gathering, and feasibility analysis. This was followed by modular design and iterative development cycles aimed at producing incremental deliverables that could be independently tested and integrated.

During the design phase, both conceptual architecture and logical system models were defined. The architecture emphasized separation of concerns through multi-tier layering:

- **Presentation Layer:** Front-end components (Client Interface and Dashboard) for visualization and user interaction.
- **Application Layer:** Flask backend serving APIs, authentication, and orchestration logic.

- **Data Layer:** PostgreSQL database and Redis in-memory cache for persistence and performance.
- **Telemetry Layer:** Wazuh Manager handling event collection and agent communication.

This layered design allowed individual modules to be developed, tested, and deployed independently while maintaining full interoperability.

b. System Development Process

The TenshiGuard development followed a continuous cycle of design, build, test, and deploy. Each iteration incorporated stakeholder feedback, integration testing, and documentation updates.

i. Requirement Analysis

The requirement phase involved collecting both functional and non-functional requirements.

Functional requirements included:

- User authentication and registration
- Real-time dashboard visualization
- Wazuh agent integration
- Subscription management and role-based access control
- Alert and report generation

Non-functional requirements addressed scalability, security, maintainability, and performance optimization.

Tools such as Microsoft Teams, GitHub Projects, and Trello were used to track progress and milestones.

I. System Design

The design phase focused on defining how each module would interact with others. Data flow diagrams, Entity Relationship Diagrams (ERDs), and system architecture charts were produced to visualize interactions between the Client Interface, Company Dashboard, and backend components.

System security design included encryption protocols, token-based authentication, and role-based access control. Docker containerization was chosen for consistent deployment across environments. Flask Blueprints were used for modular routing, ensuring a clean separation of API endpoints and web interfaces.

II. Implementation

Implementation involved coding, integration, and unit testing. Flask served as the core backend framework, PostgreSQL handled structured data storage, and Redis enabled caching and task brokering.

The front-end was built with Bootstrap 5 and Chart.js to ensure responsive design and interactive visualization. Celery was configured to execute asynchronous background tasks such as scheduled data synchronization, automated reporting, and email notifications.

Each development sprint targeted specific deliverables:

- Sprint 1: Core backend setup, authentication, and database schema creation.
- Sprint 2: Client Interface dashboard and mock data integration.
- Sprint 3: Wazuh API integration and agent data handling.
- Sprint 4: Company Dashboard, analytics caching, and automated report generation.
- Sprint 5: Deployment, optimization, and system documentation.

Version control was enforced using GitHub with a strict branching model (main, dev, and feature/*). Continuous integration workflows ran automated tests for every commit to maintain system stability.

c. Testing Methodology

Testing was conducted at multiple levels to verify functionality, performance, and reliability.

I. Unit Testing

Each module underwent unit testing using PyTest to ensure individual functions performed as expected. Core tests included user registration, JWT token validation, and API response accuracy.

II. Integration Testing

Integration testing validated the interaction between the Flask backend, PostgreSQL, and Wazuh. Postman scripts and automated CI jobs verified data flow between services. Docker-based staging environments replicated production conditions to ensure environment consistency.

III. System and Performance Testing

System-level testing simulated multi-user operations using tools such as Locust. Metrics such as response time, concurrency limits, and error rate were measured to evaluate performance under load. Redis caching and Celery worker scaling were fine-tuned based on test results to enhance performance.

IV. Security Testing

Security testing was critical due to the sensitive nature of endpoint monitoring. Tools such as OWASP ZAP and Bandit were used to detect vulnerabilities. Access control lists were reviewed to ensure strict adherence to least-privilege principles. SQL injection, CSRF, and session hijacking tests were performed to validate defenses.

d. Deployment Strategy

The deployment strategy was designed to ensure repeatability, scalability, and resilience. Docker Compose orchestrated all containers, including Flask, PostgreSQL, Redis, Celery, and Nginx.

Three environments were maintained:

- **Development Environment:** Used for feature prototyping with debug configurations.
- **Staging Environment:** Used for integration and pre-production testing.
- **Production Environment:** Used for live operation, secured with SSL and real telemetry data.

A **blue-green deployment** strategy was implemented to eliminate downtime during updates. Nginx handled load balancing and SSL termination, while Gunicorn managed concurrent API requests.

System monitoring and logging were handled through Wazuh, which provided continuous visibility into infrastructure health and detected anomalies in real time.

e. Documentation and Version Control

Comprehensive documentation accompanied each phase of development.

- Technical documentation outlined architecture, database schema, and API routes.
- User manuals guided installation, configuration, and usage procedures.
- Developer notes tracked dependencies, configuration changes, and known issues.

Version control ensured that all code changes were tracked and reviewed. Git commits were signed and linked to project issues. Pull requests required approval

before merging into the dev or main branches, maintaining transparency and accountability.

f. Quality Assurance and Governance

The TenshiGuard project adopted a proactive quality assurance approach to ensure that both technical and organizational objectives were met. QA reviews were conducted after every major sprint to identify code inefficiencies, performance bottlenecks, and integration errors.

Governance was implemented through internal review boards, where progress was assessed weekly. Compliance with software engineering standards, cybersecurity best practices, and open-source licensing was maintained throughout the project lifecycle.

System logs, commit histories, and deployment artifacts were archived to support traceability and academic evaluation.

9. Implementation / Demonstration

a. Screenshots (Dataset preprocessing, Model training, Testing results)

The image shows the TenshiGuard Admin Dashboard, a local cybersecurity monitoring tool. It displays one user, five active alerts, and a threat graph peaking on Thursday and Saturday. The dashboard gives admins a clear, real-time view of system activity and security risks.

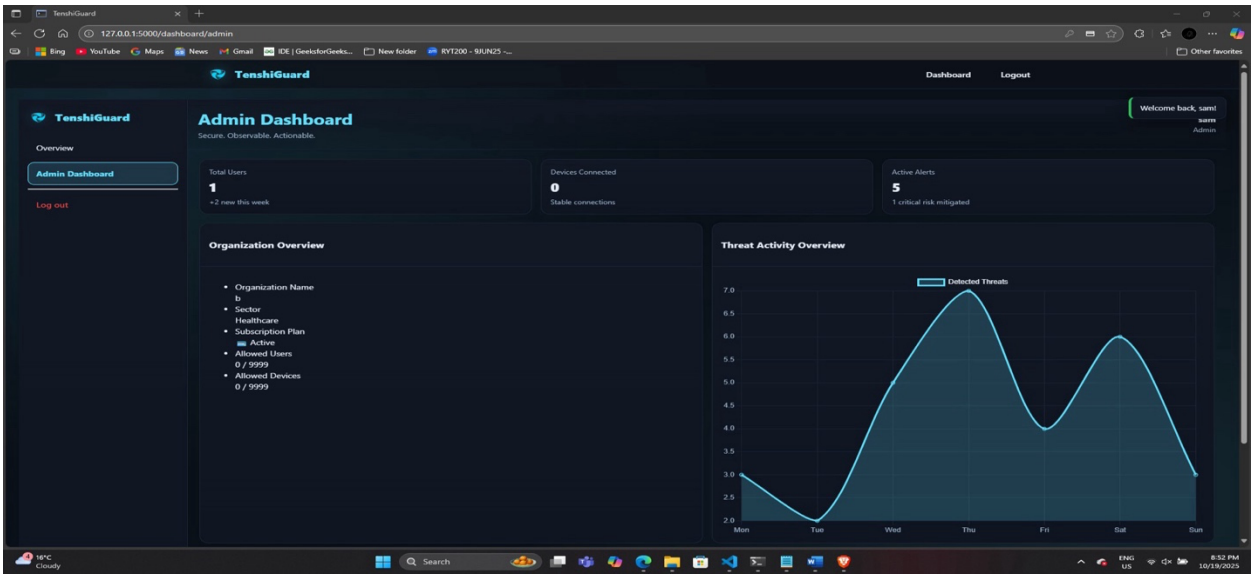


Figure 3: Dashboard of Admin Interface

The image shows the TenshiGuard Complete Payment page for the Enterprise plan priced at $199 per month. It's a mock payment setup running locally at 127.0.0.1, allowing users to test entering card details. This page demonstrates a secure and user-friendly payment design during development.

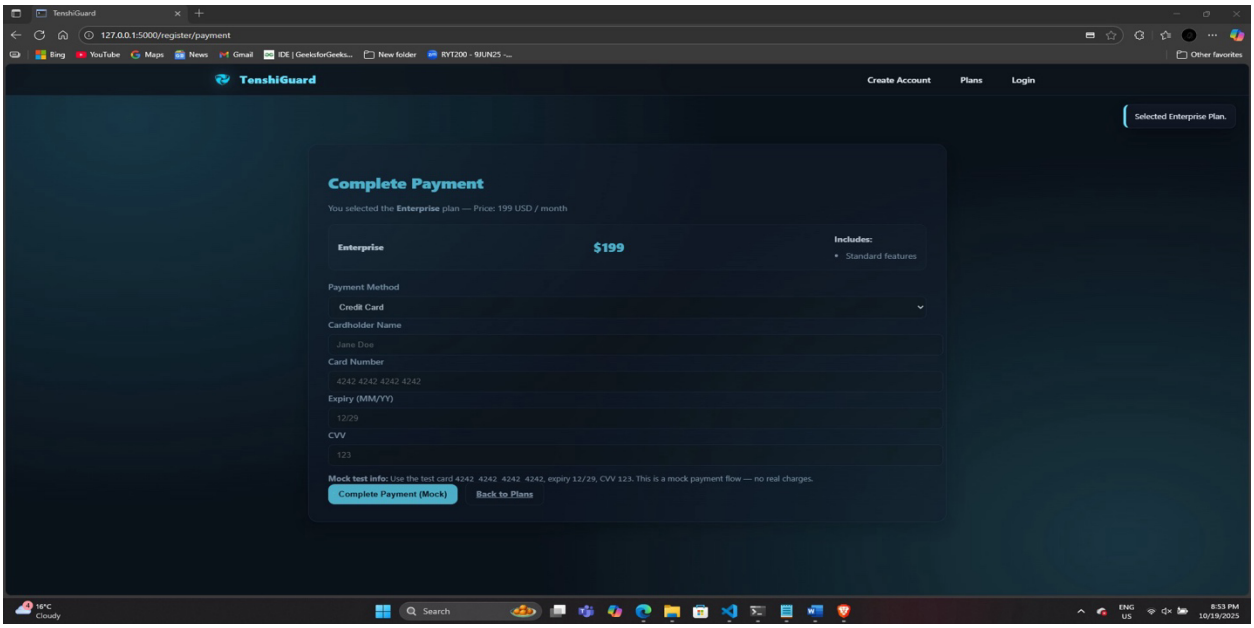


Figure 4: Complete Payment

The image shows the Create Admin Account page of the TenshiGuard platform, displayed after a successful payment. The user can set up an admin username, email, and password, and choose which security alerts to receive, such as for malware or suspicious logins. This step finalizes account setup and alert preferences before accessing the main dashboard.

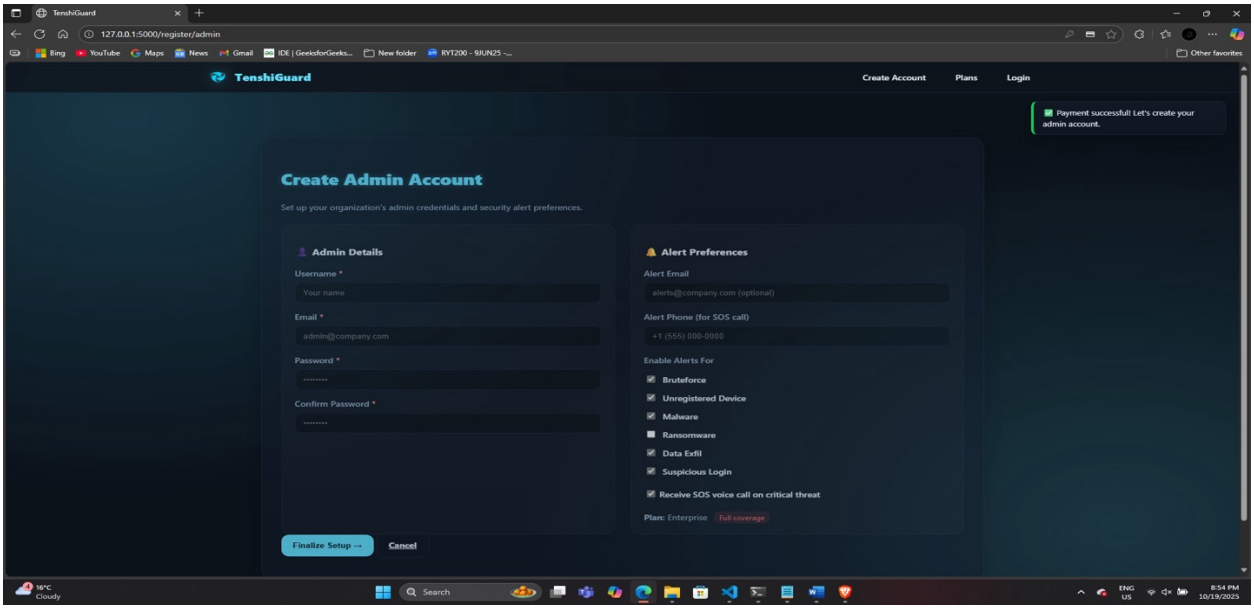


Figure 5: Create Admin Account

The image shows the Marketing Analytics dashboard, where the admin can track and manage all client payments. It displays total monthly revenue, payment statuses, and details for each company's plan and transactions. Quick action buttons help send invoices, reminders, or suspend overdue accounts for efficient subscription management.

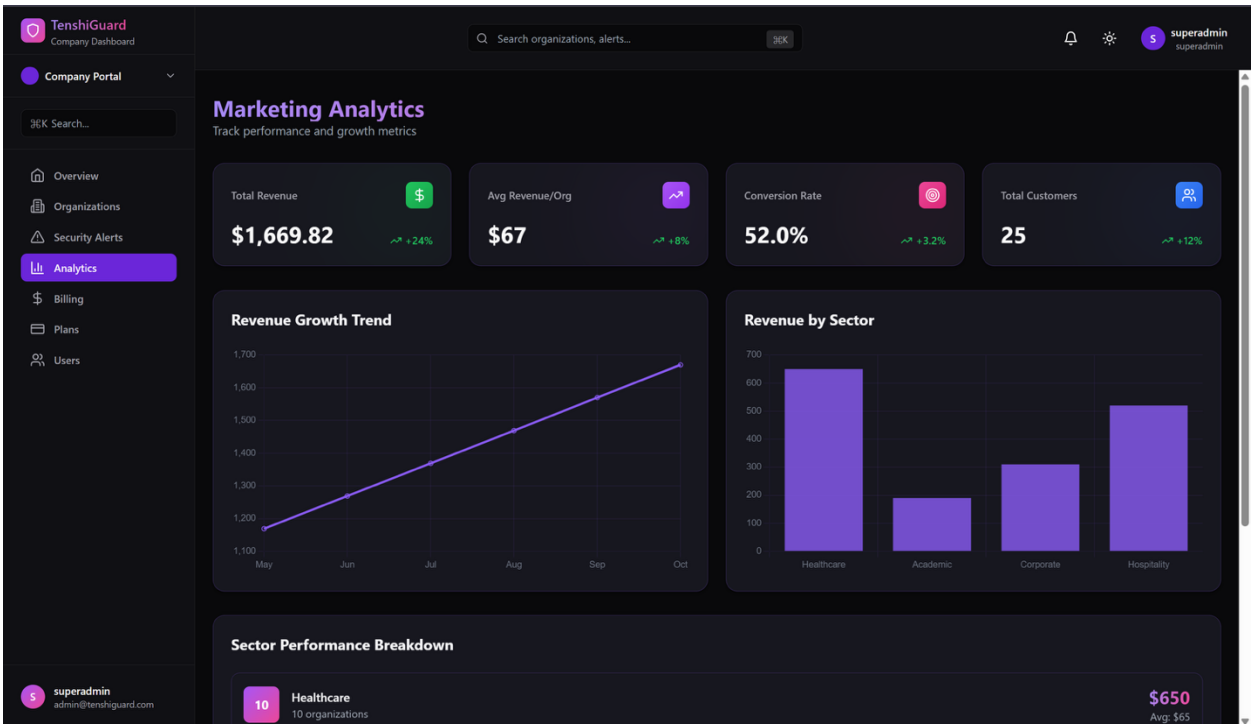


Figure 6: Marketing Analytics

The image shows the Billing & Revenue section of the TenshiGuard Company Dashboard, which tracks payments and subscriptions. It highlights total monthly revenue of $1,669.82 with growth stats, along with paid and pending organizations. The table below shows each company's plan, sector, and payment status, giving admins a clear overview of billing and subscription activity.

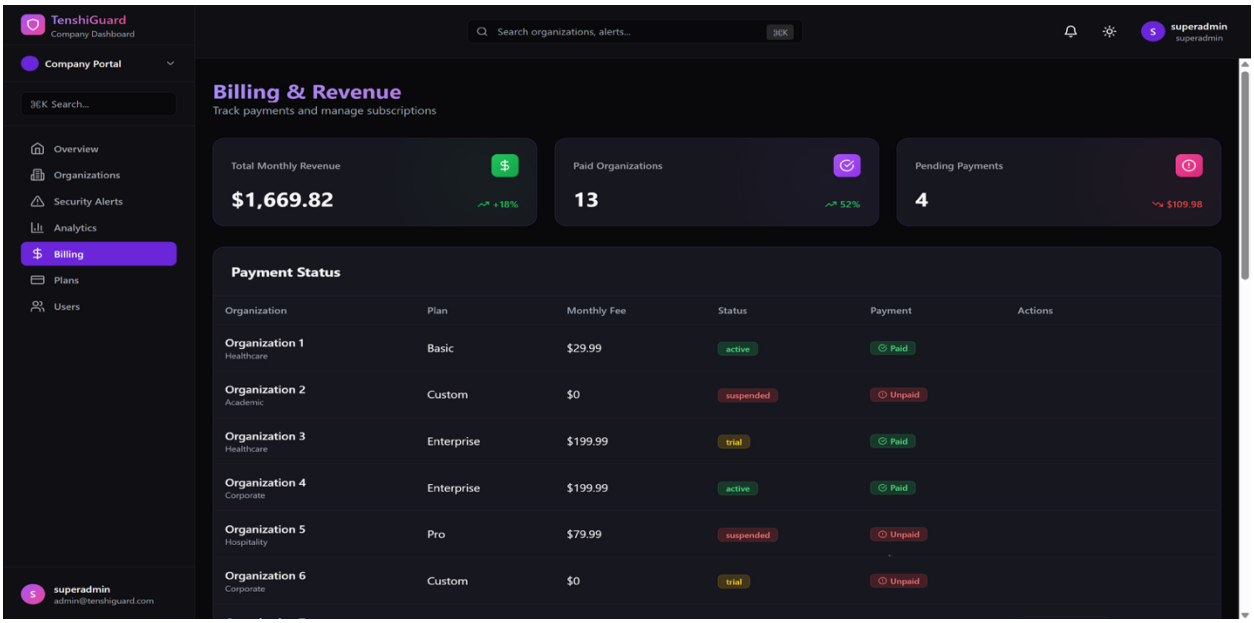


Figure 7: Billing and Revenue

The image shows the Subscription Management Dashboard of the TenshiGuard system, where admins can oversee all client subscriptions. It displays totals for organizations, active and expired plans, and pending payments. The table provides subscription and billing details, helping admins manage renewals and client accounts efficiently.

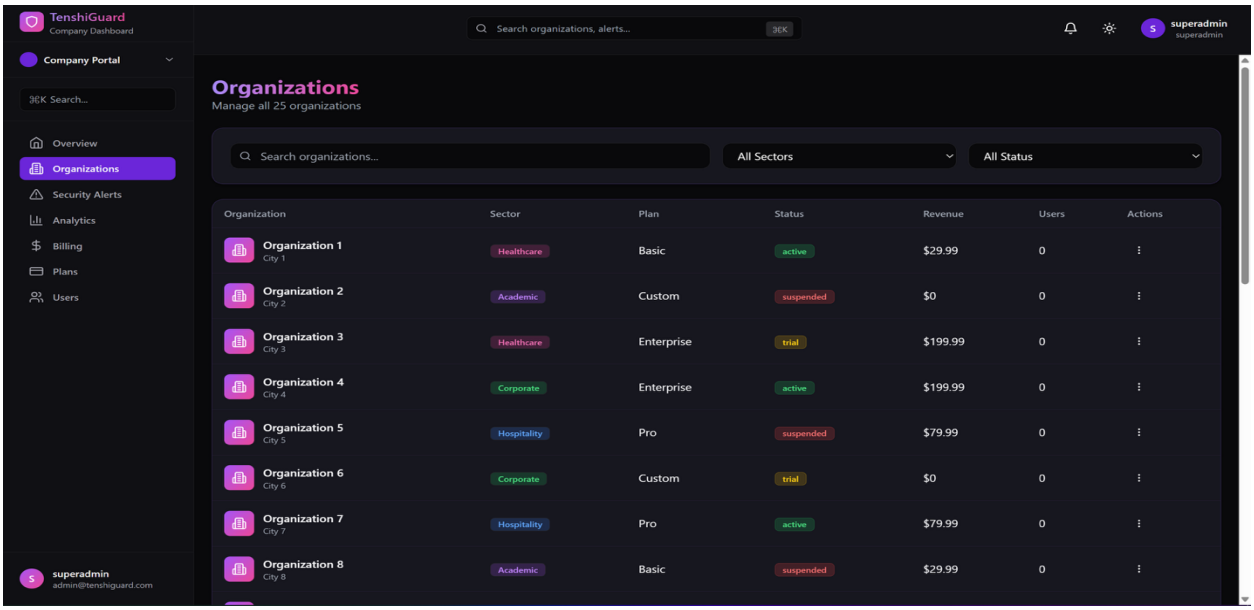


Figure 8: Subscription Management Dashboard

The image shows the Live Security Alerts Dashboard of the TenshiGuard system, displaying real-time alerts by severity high, medium, and low. It includes details like alert type, time, and description, along with options for email or SMS notifications. This dashboard helps admins quickly detect and respond to security threats across all connected endpoints.

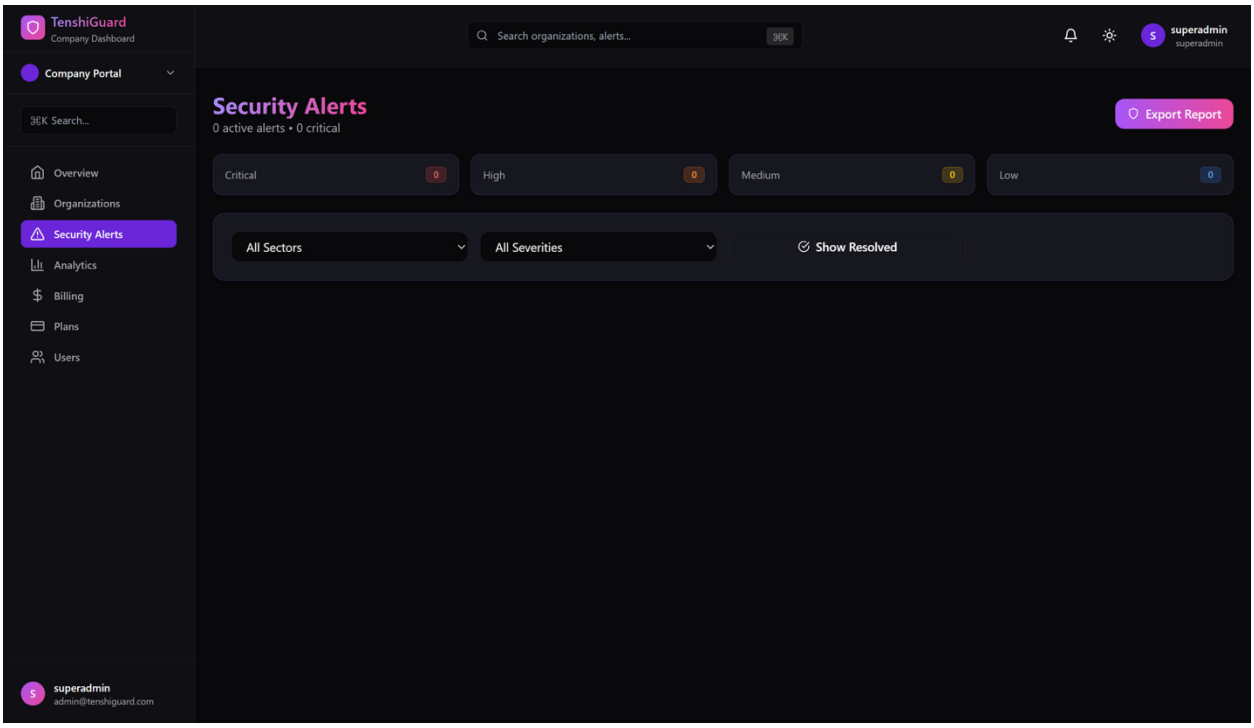


Figure 9: Live Security Alerts Dashboard Admin page

The image shows the Live Security Alerts Client Dashboard of TenshiGuard – Enterprise Security. It displays real-time alerts by severity, with all categories currently at zero, and includes settings for email or SMS notifications and login thresholds. The page also has sections for active alerts and recent activities, which are currently empty.

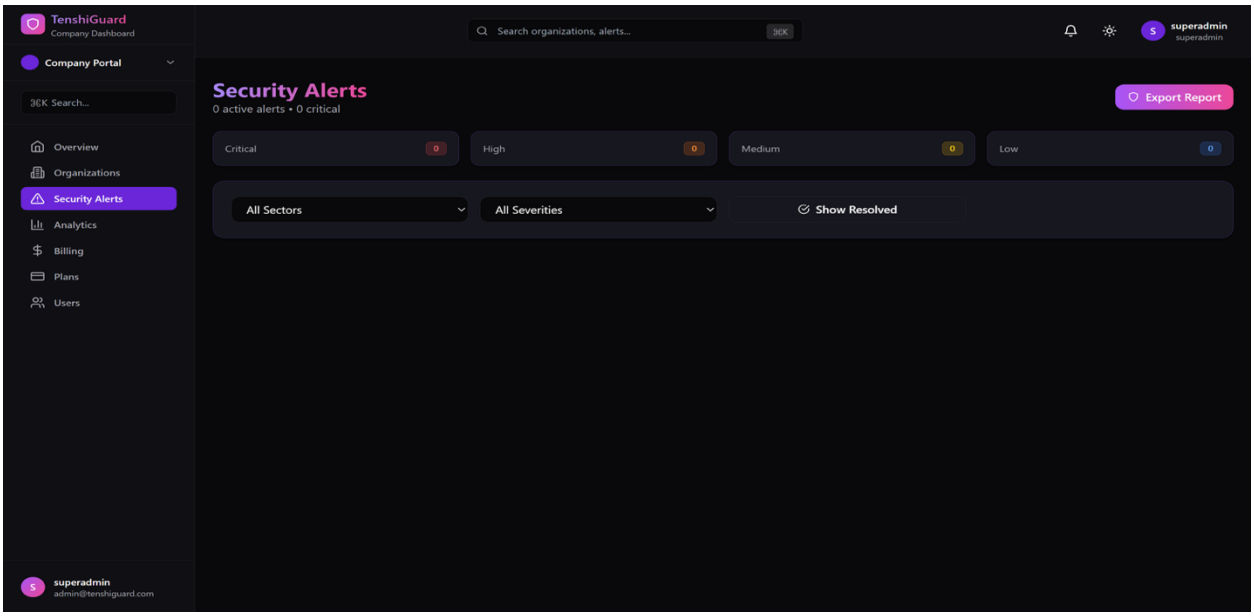


Figure 11: Live Security Alerts Dashboard Client page

The image shows the Live Endpoint Monitoring page of TenshiGuard Enterprise Security on the client side. It displays a table for tracking active endpoints with details like name, IP, OS, status, and threat level, though no data is currently shown. The page, accessed by the healthcare admin, includes navigation to the Dashboard, Live Alerts, Reports, and Compliance modules.

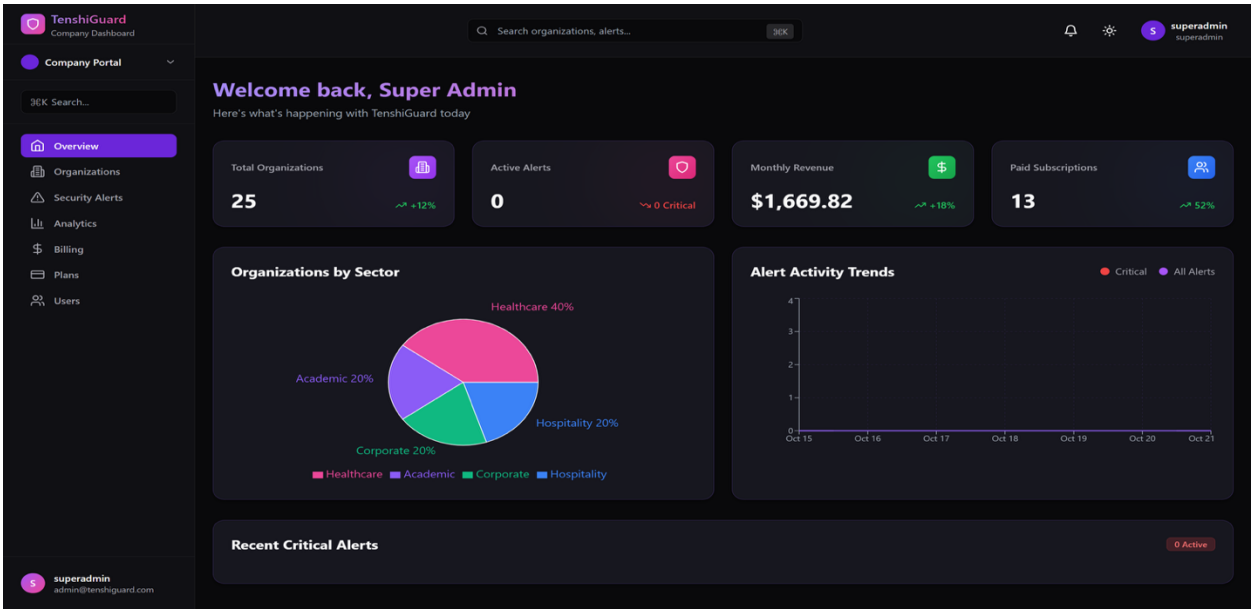


Figure 12: Live Endpoint Monitoring

The image shows the Login Page of the TenshiGuard Enterprise Security Platform. It includes fields for username, password, and options to remember login details or recover credentials. A banner confirms a successful logout, and demo accounts for different roles are listed below in a clean blue-and-white design.

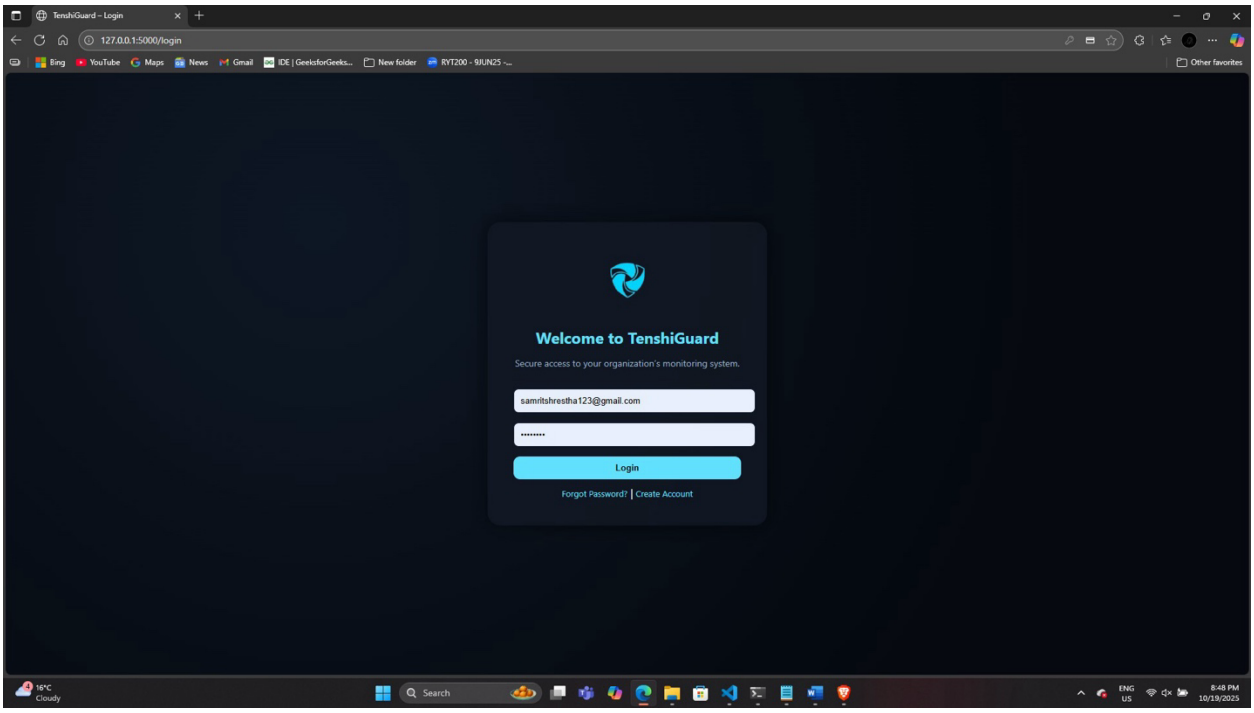


Figure 13: Login Page

The image shows the Recover Account page of the TenshiGuard Enterprise Security Platform. It lets users reset their username or password via Email or Phone Recovery, with a field to enter their registered email and send a recovery code. The clean blue-and-white layout includes quick links to log in or register.

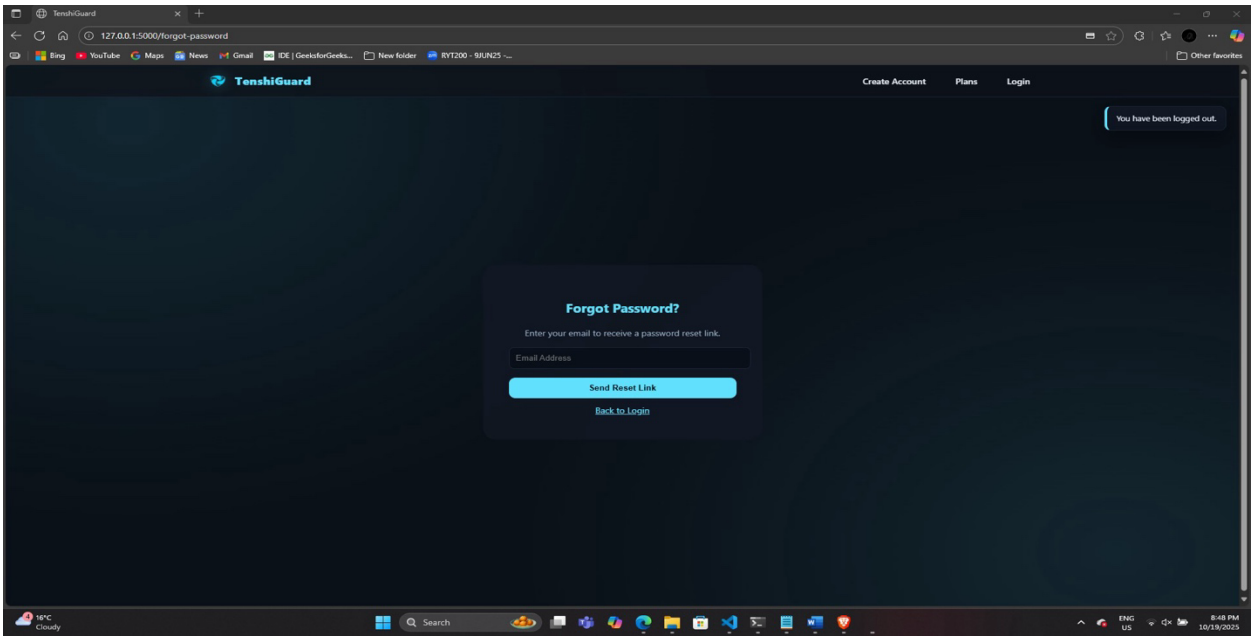


Figure 14: Account Recovery Page

The image shows the Register Your Company page of the TenshiGuard Enterprise Security Platform. It's the first step in a three-part registration process, collecting company and admin details like name, email, and password. The clean blue-and-white layout includes a Continue to Plan Selection button and a link for existing users to log in.

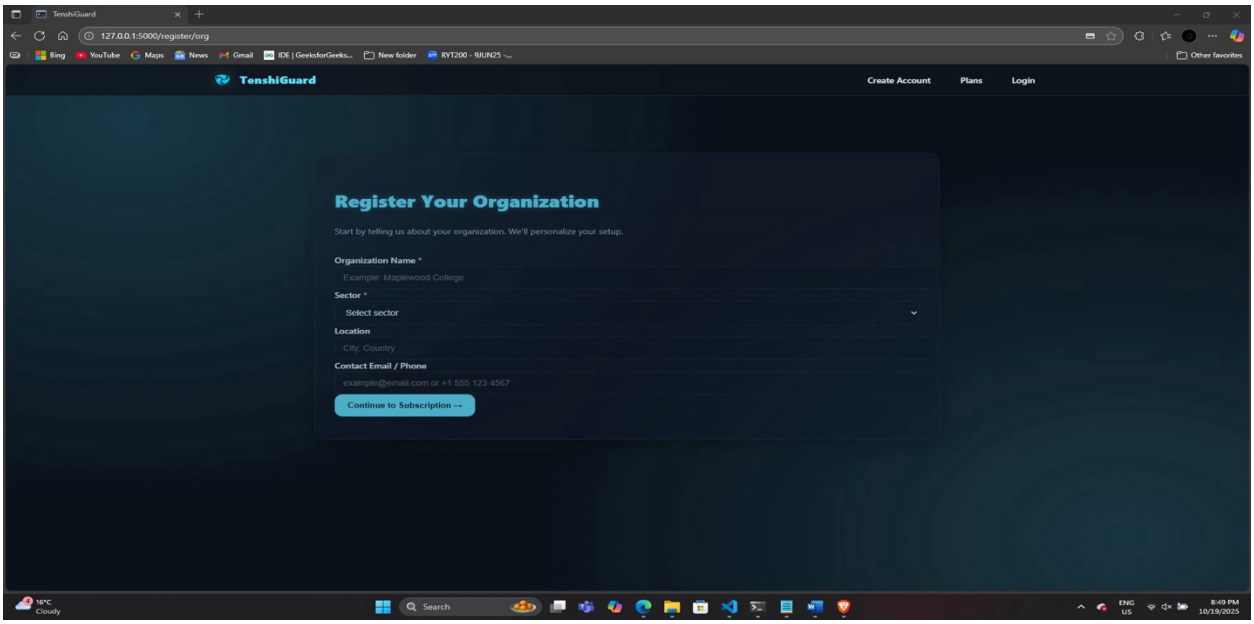


Figure 15: Registration page

The image shows the Subscription Page of TenshiGuard, where new organizations can register and choose a plan. It offers Starter ($19/month), Professional ($49/month), and Enterprise ($99/month) options with varying features like monitoring, reporting, and support. The page makes plan comparison and activation quick and easy for clients.

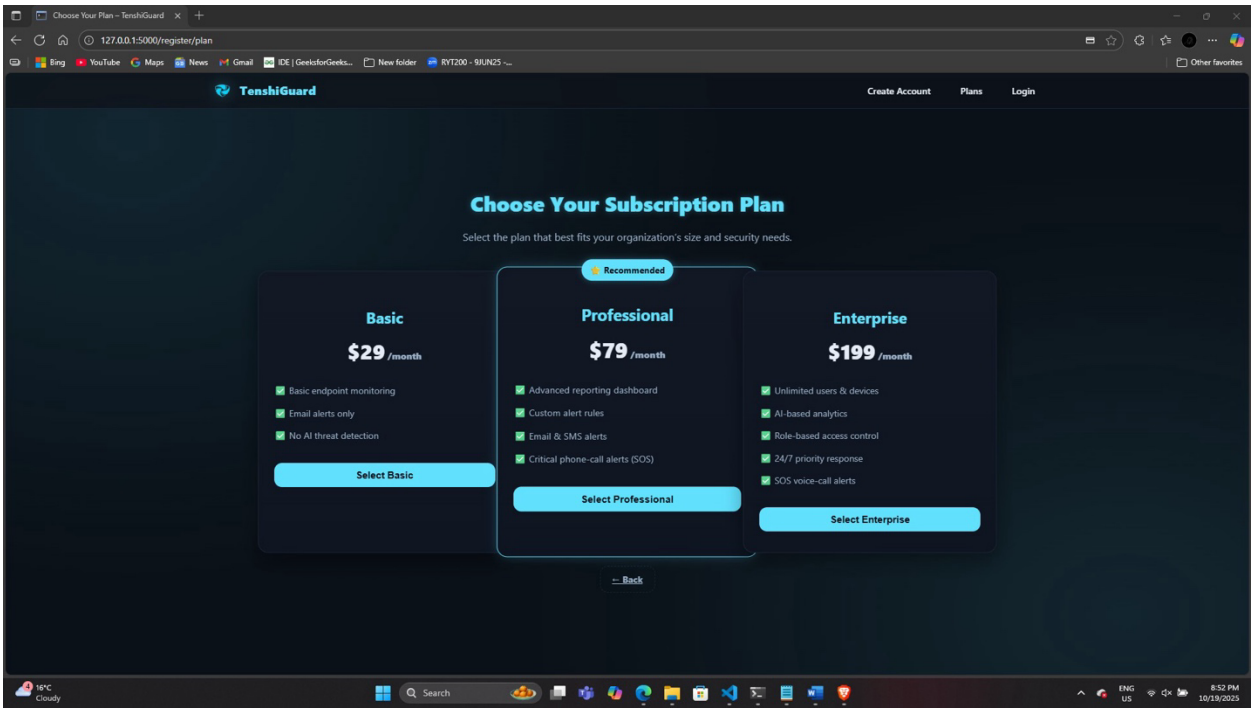


Figure 17: Subscription Page

The image shows the Subscription & Billing page of TenshiGuard, displaying the current plan, billing info, and payment history. It highlights the Enterprise Plan ($99/month), devices in use, and renewal date, with options to view past transactions or manage receipts. Users can also upgrade or downgrade plans, making it easy to manage subscriptions and payments.

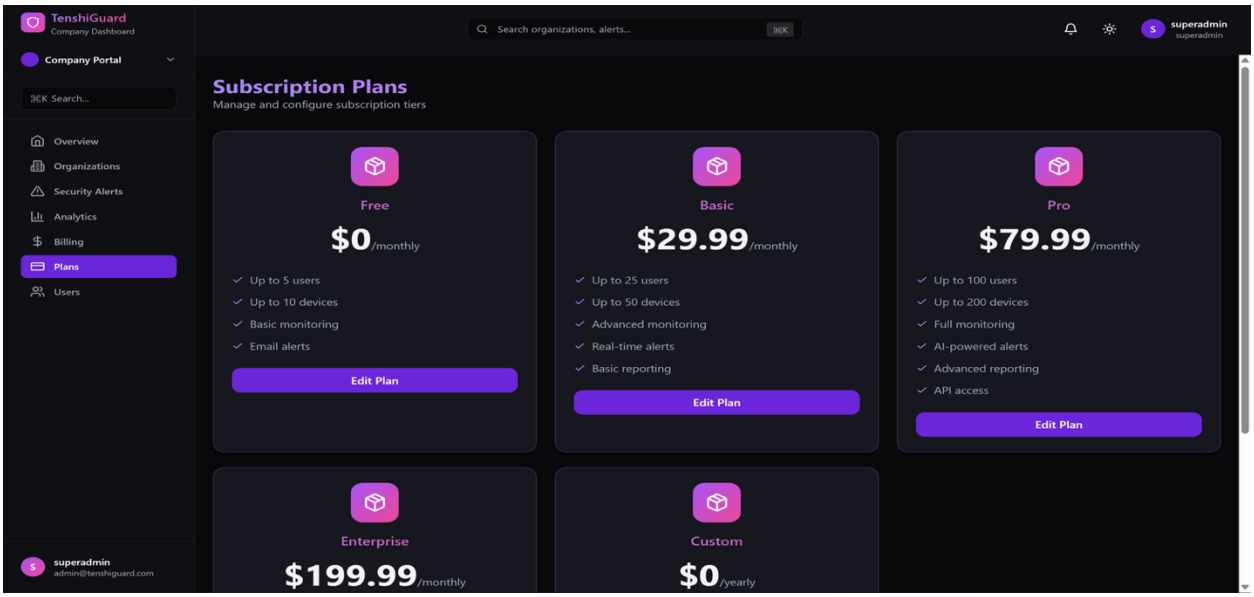


Figure 20: Subscription & Billing Explanation

The image shows the TenshiGuard Admin Dashboard for Test Org 10 on a Professional plan, providing a central interface for managing endpoint security. It displays panels for device status and performance trends, though all currently show zero connected agents. The dashboard is in Simulation Mode for testing alerts and notifications, with navigation for users, devices, alerts, and subscription management.

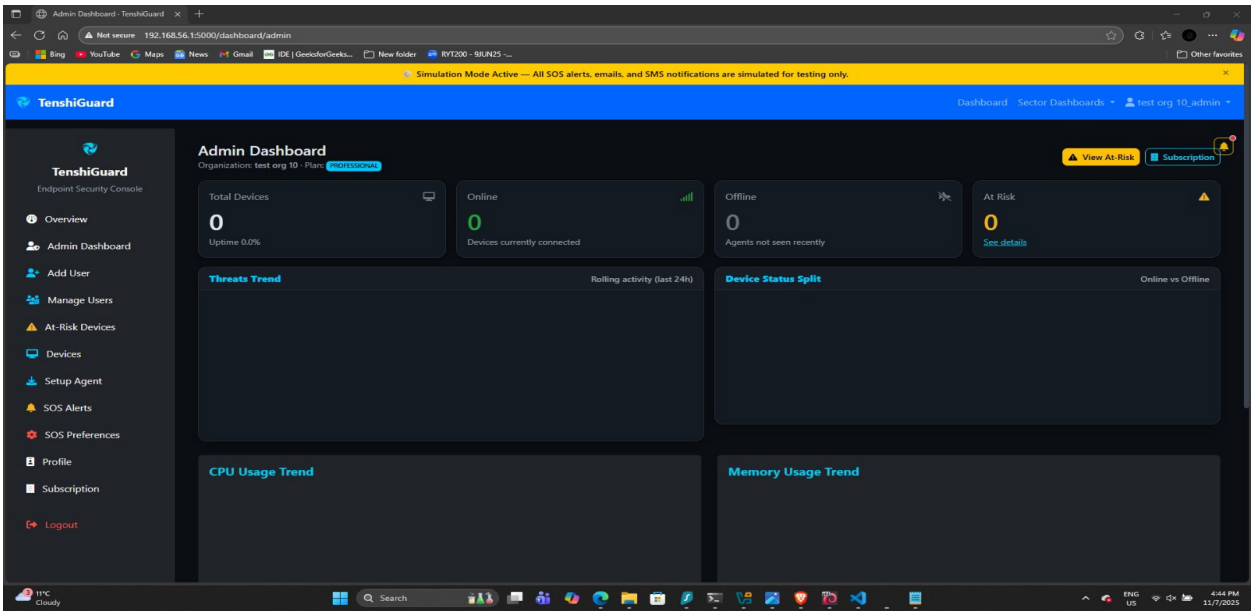


Figure 21: TenshiGuard Admin Dashboard Overview

The image shows the SOS Preferences page of the TenshiGuard Endpoint Security Console, where admins configure alert settings. Alerts are set to trigger at medium severity or higher for selected categories like System, Network, Login, and Malware, with notifications sent via Email and SMS. A simulation banner indicates that all alerts are currently in test mode, and options to save preferences or view alerts help admins manage notifications efficiently.

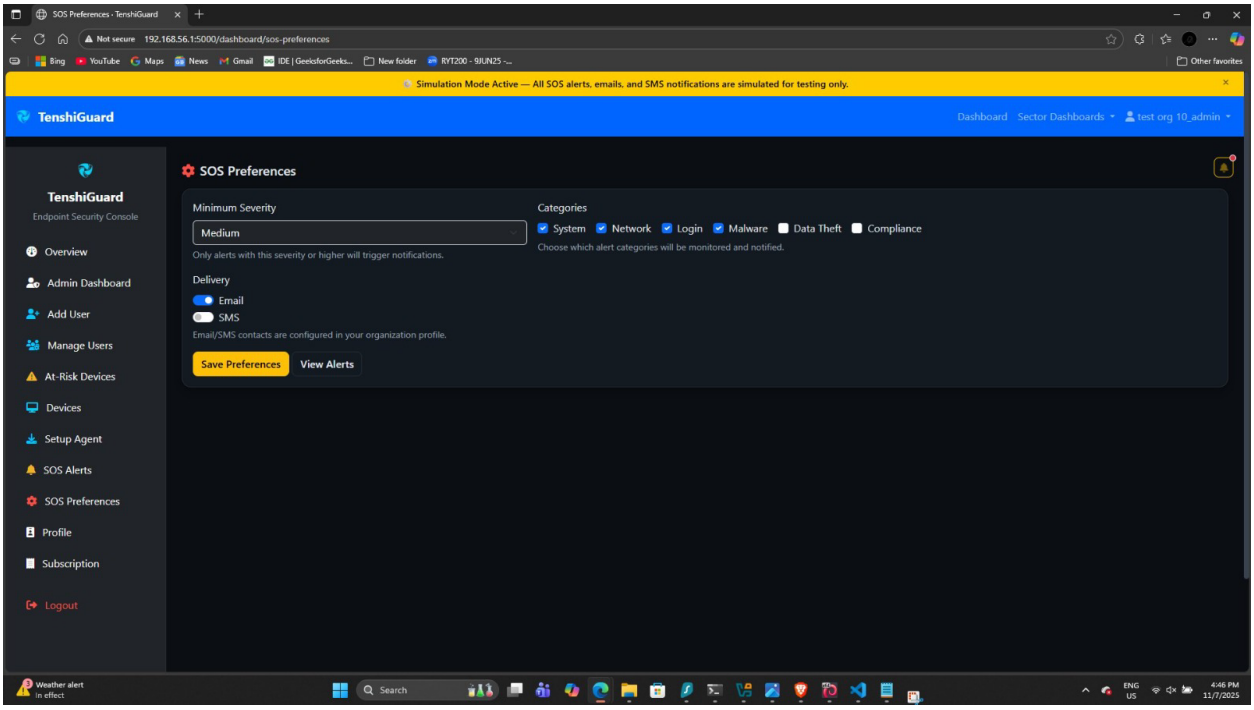


Figure 22: TenshiGuard SOS Preferences Configuration

The image shows the SOS Alerts section of the TenshiGuard Endpoint Security Console, where real-time security alerts are monitored and managed. The dashboard lists alerts by time, severity, category, and status, though no active alerts are currently shown. A simulation banner indicates test mode, with options to customize preferences or generate demo alerts for practice.

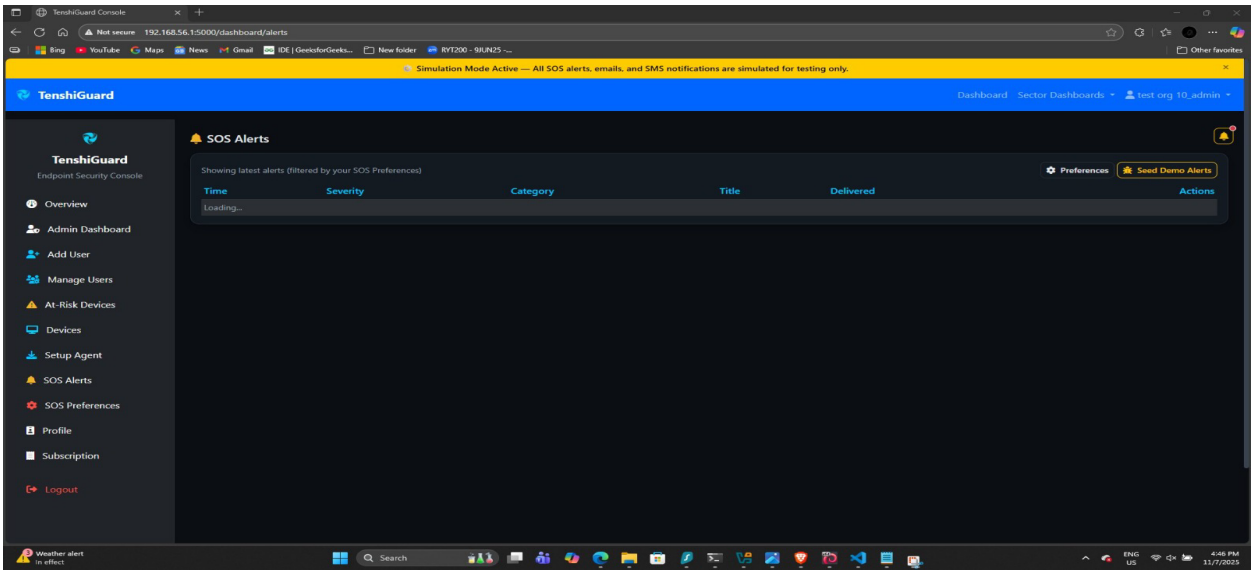


Figure 23: TenshiGuard SOS Alerts Dashboard

The image shows the TenshiGuard Agent Deployment page, where admins can deploy security agents across platforms like Linux, macOS, Windows, Cloud, and Android. Each option includes a View Guide for installation instructions, covering servers, desktops, routers, and cloud instances. A simulation banner indicates test mode, and the navigation menu provides quick access to key sections for managing devices and alerts.

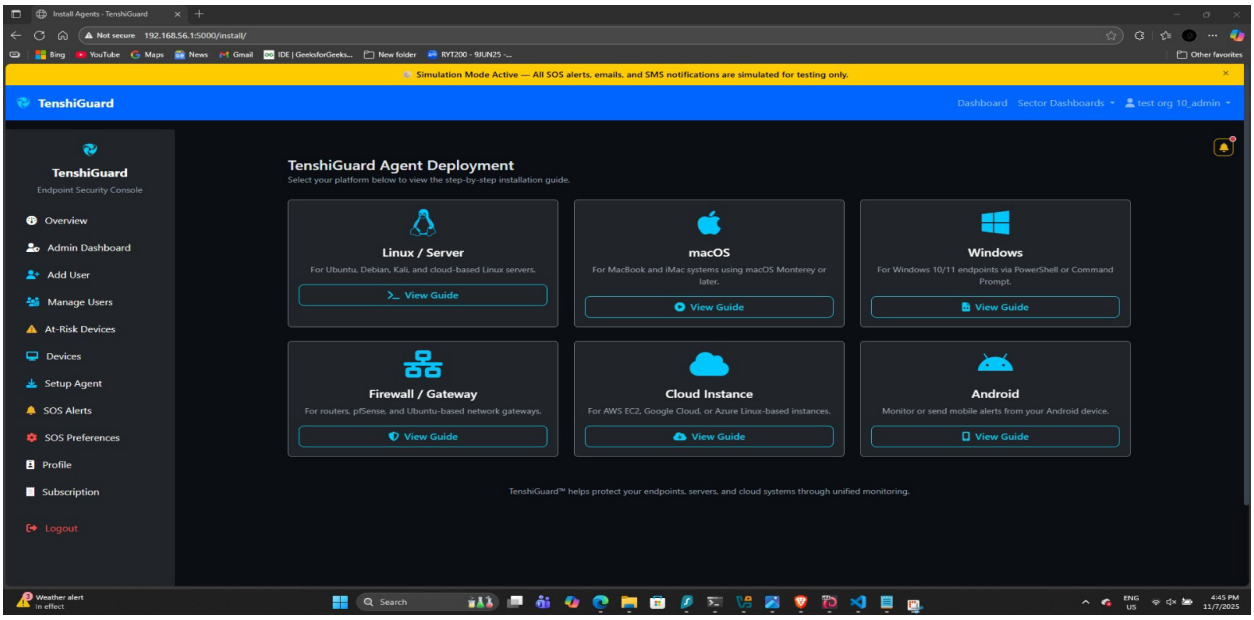


Figure 24: TenshiGuard Agent Deployment Interface

The image shows the Connected Devices page of the TenshiGuard Endpoint Security Console, providing an overview of all registered devices for monitoring. Currently, no devices are listed, and admins can Setup New Agent or Clean Duplicates. A simulation banner indicates test mode, and the navigation panel gives access to other sections like the Dashboard and SOS Alerts.

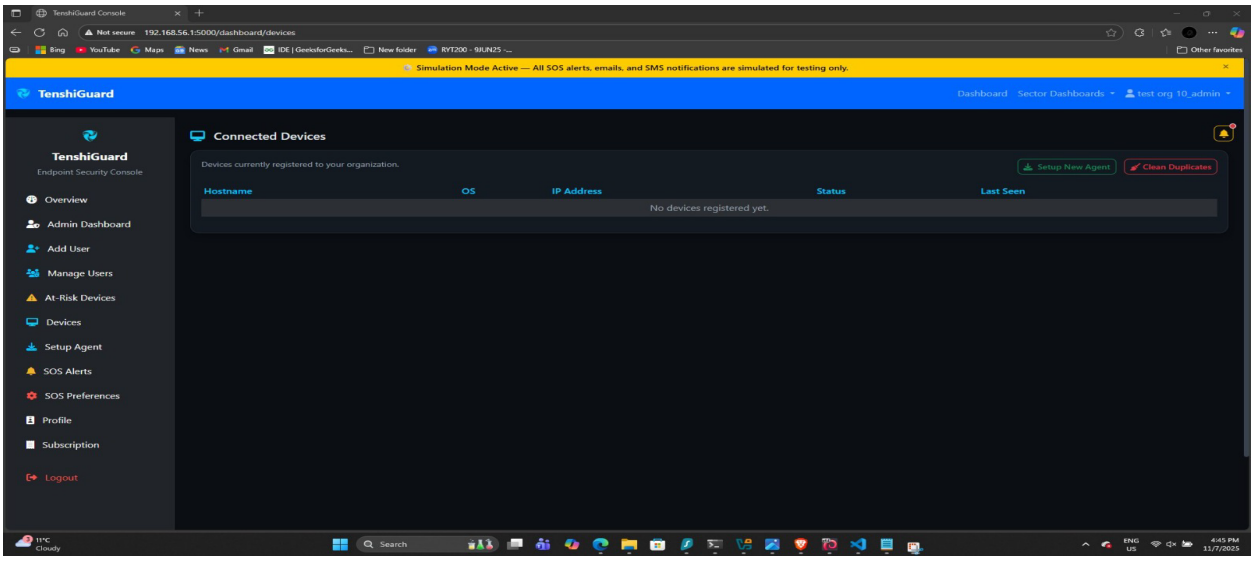


Figure 25: TenshiGuard Connected Devices Dashboard

The image shows the At-Risk Devices section of the TenshiGuard Endpoint Security Console, where admins monitor devices for potential threats. Currently, no devices are flagged as risky. A simulation banner indicates test mode, and the table would normally show device details, risk levels, and suggested mitigation steps, with navigation to other management sections on the sidebar.

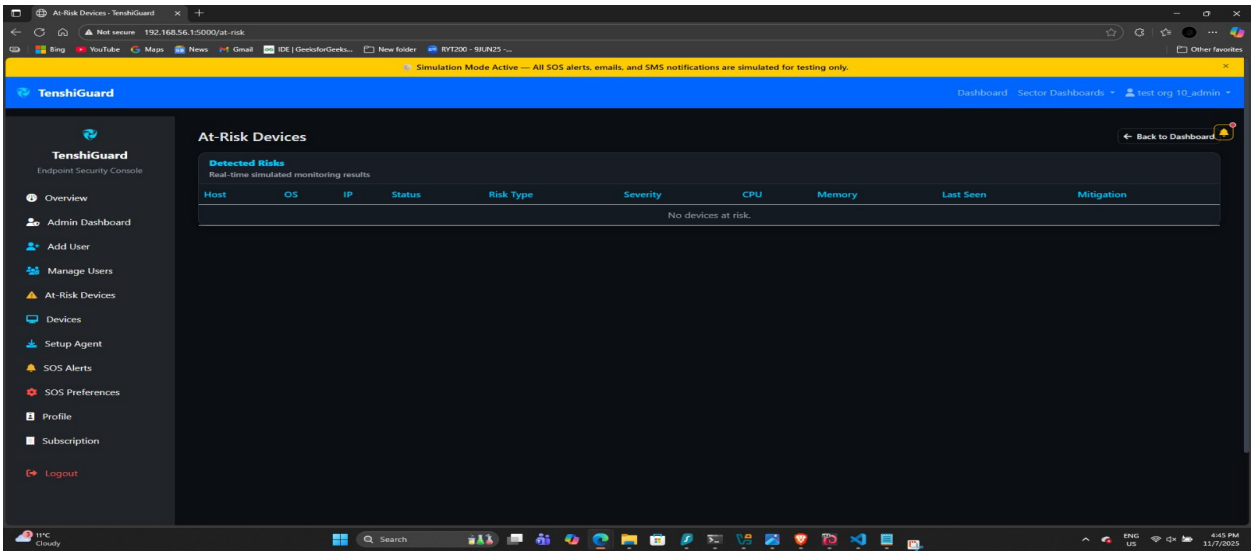


Figure 26: TenshiGuard At-Risk Devices Dashboard Overview

The image shows the Manage Users section of the TenshiGuard Endpoint Security Console, where admins can view and manage user access. It lists users with details like username, role, and status, and provides buttons to edit, delete, or add users. A simulation banner indicates test mode, and the sidebar allows navigation to other management sections.

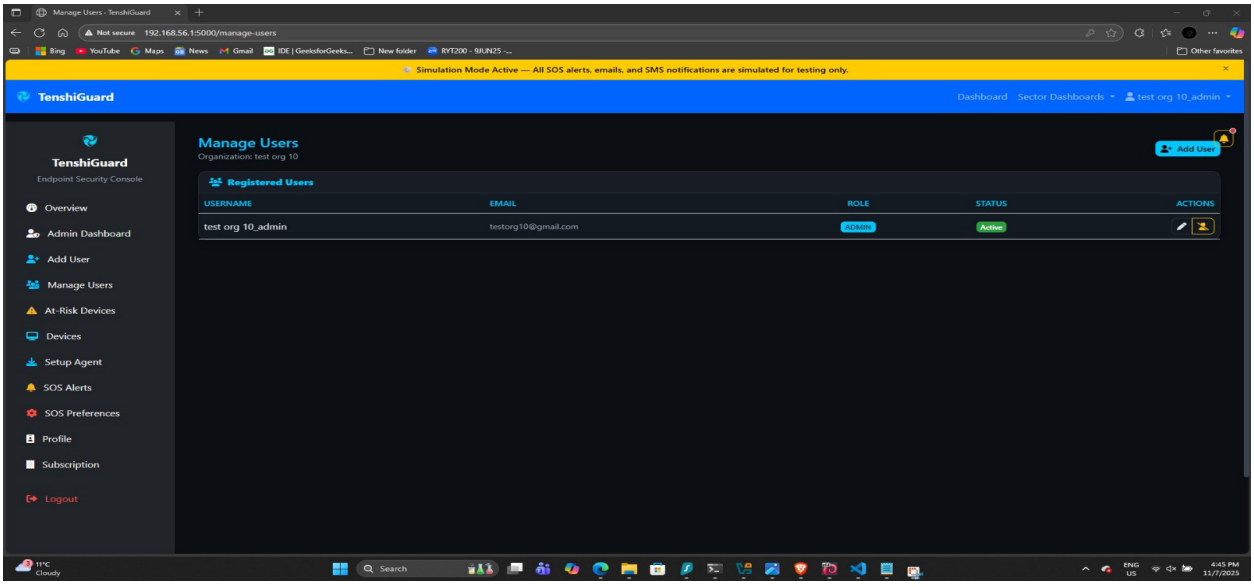


Figure 27: TenshiGuard User Management Panel

The image shows the Add New User page of the TenshiGuard Endpoint Security Console, accessed locally at 192.168.56.1:5000. Admins can create new accounts by entering username, email, password, organization, and sector (set to “academic”). A simulation banner indicates test mode, with navigation options for managing users, devices, alerts, and subscriptions.

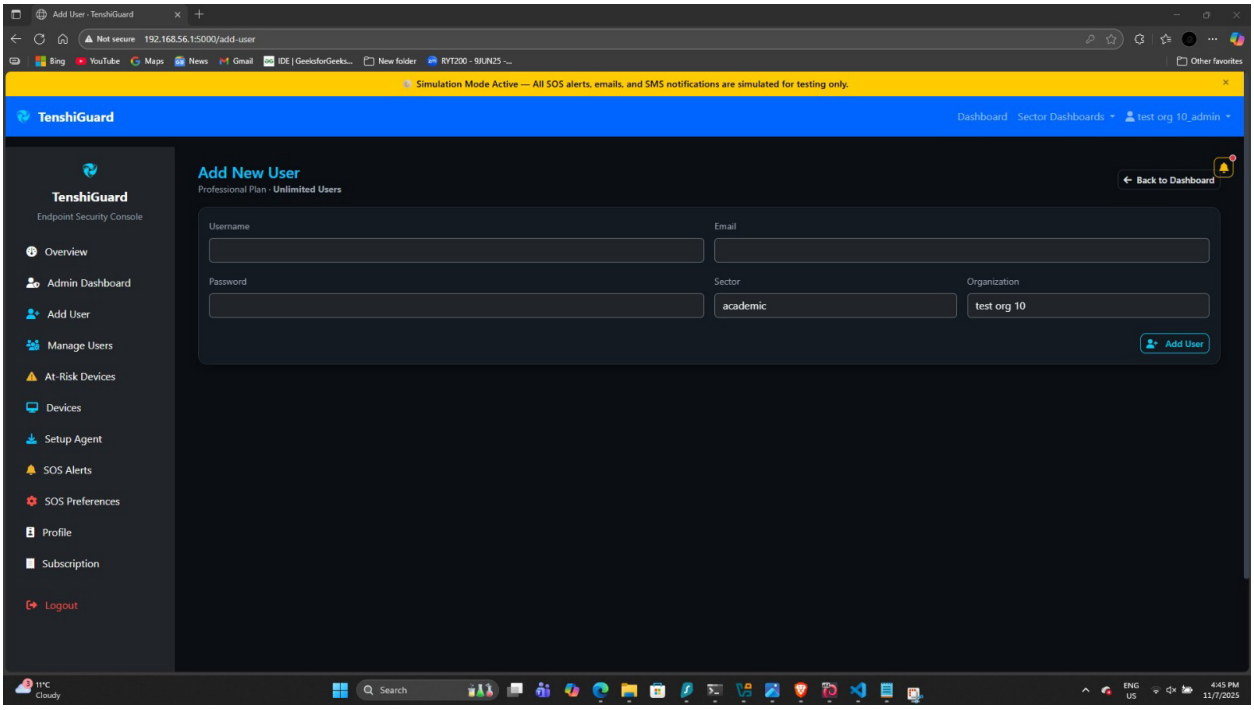


Figure 28: Adding a New User in TenshiGuard Console

The image shows the Profile Overview page of the TenshiGuard Endpoint Security Console, displaying user and organization details. The logged-in user is test org 10_admin with a Professional plan, and SOS alerts are enabled but no contacts are configured. A simulation banner indicates test mode, with navigation for managing users, devices, alerts, and other security settings.

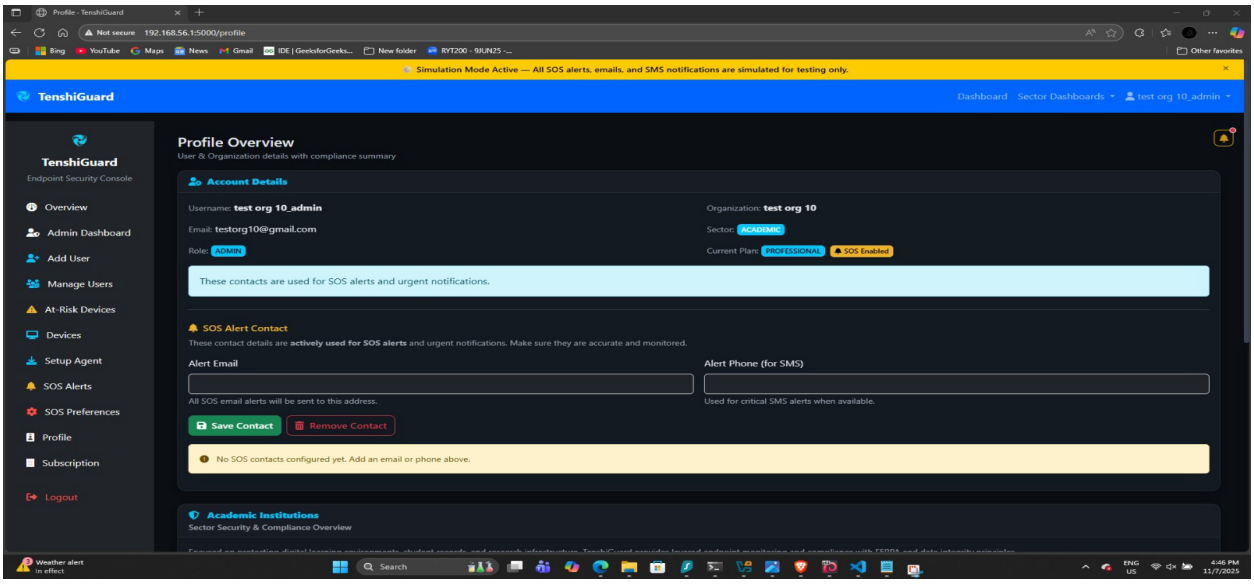


Figure 29: TenshiGuard Profile Overview Screen

The image shows the TenshiGuard Agent installation page for Linux or server systems. It provides step-by-step instructions to download, install, and start the agent, as well as commands to verify the installation. A simulation banner indicates that all alerts and messages are in test mode.

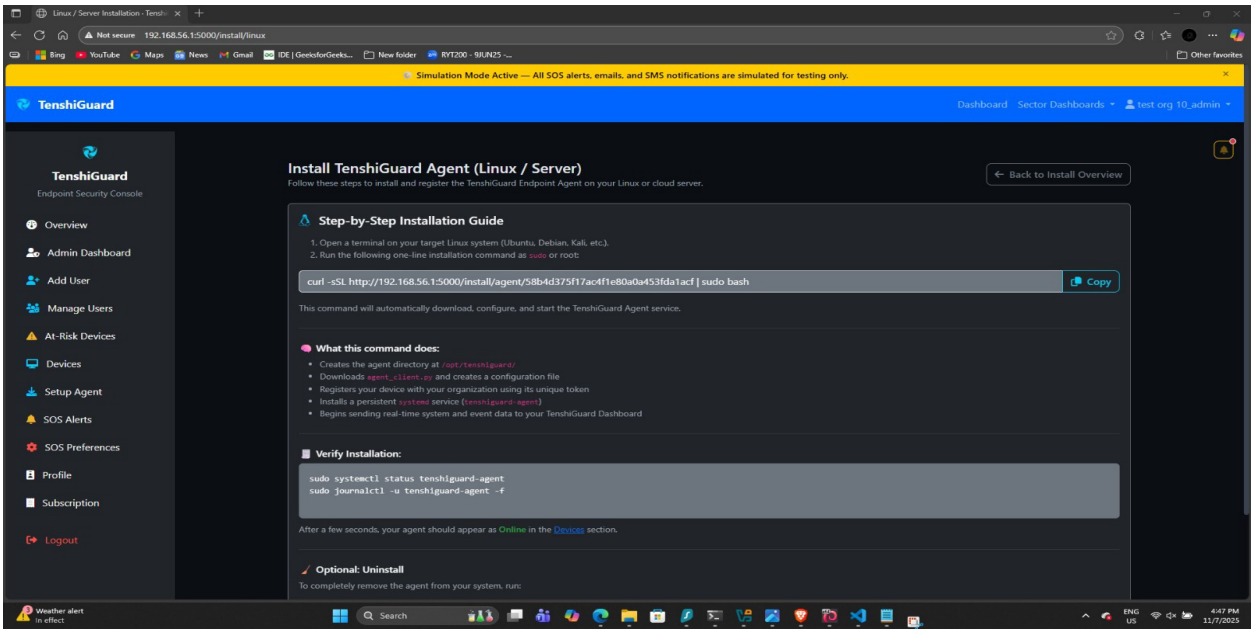


Figure 30: Installing TenshiGuard Agent on a Linux System

The image shows the Subscription Plan page of the TenshiGuard Endpoint Security Console for test org 10, using the Professional plan with SOS alerts enabled. It compares Basic, Professional, and Enterprise plans, highlighting features like real-time monitoring, patch management, and weekly reports. The page also lists supported compliance standards, including FERPA, SOC 2 Type II, and ISO 27001.

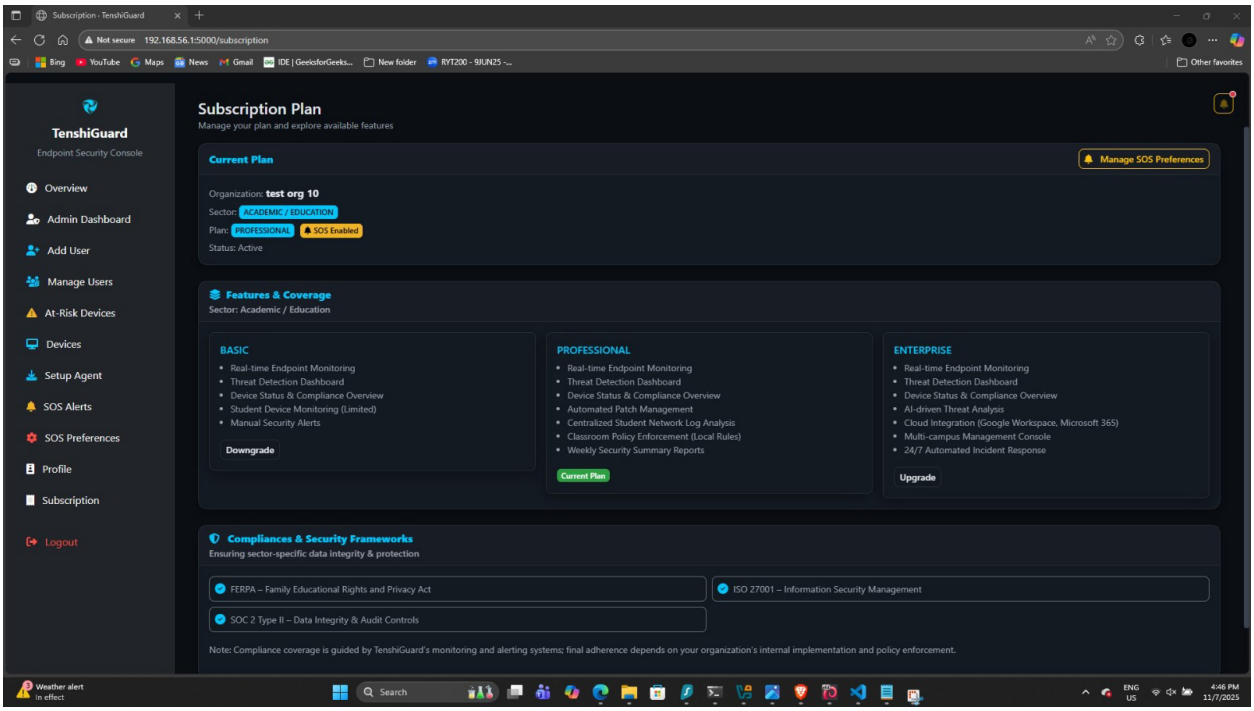


Figure 31: TenshiGuard Subscription Plan Overview

The image shows the Compliance & Security Overview page of the TenshiGuard Endpoint Security Console for academic institutions. It highlights compliance with frameworks like FERPA, ISO 27001, PIPEDA, and GDPR, showing statuses and overall progress toward full compliance. The dashboard helps monitor classrooms, labs, and research data to maintain cybersecurity and regulatory standards.

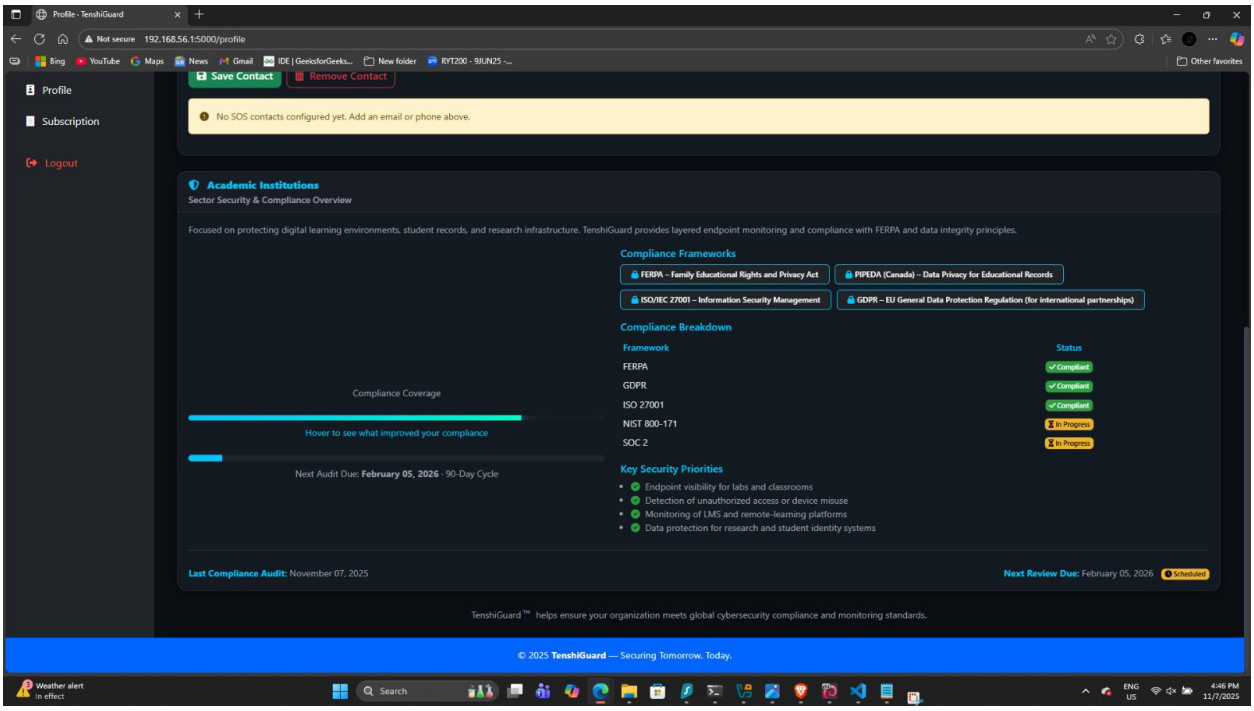


Figure 32: TenshiGuard Compliance and Security Overview

This screenshot shows the Live Security Events page in the TenshiGuard security console. It lists recent login attempts, agent activity, and alerts so the admin can see what's happening on the system in real time. The page helps the admin quickly spot failed logins, new agent registrations, and other important security events.

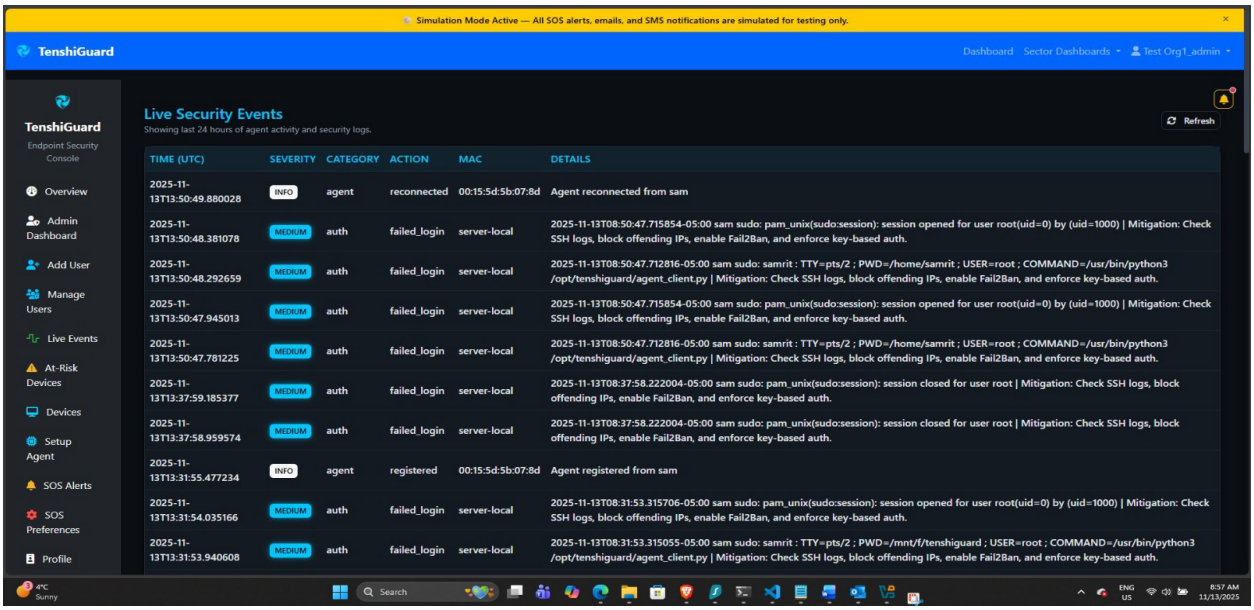


Figure 33: TenshiGuard – Live Security Events Page

This screenshot shows the Real-Time SOS Alerts window in the TenshiGuard console. It displays live alerts such as failed login attempts and agent reconnections, helping the admin quickly spot suspicious activity. The panel updates automatically, giving real-time visibility into urgent security events.

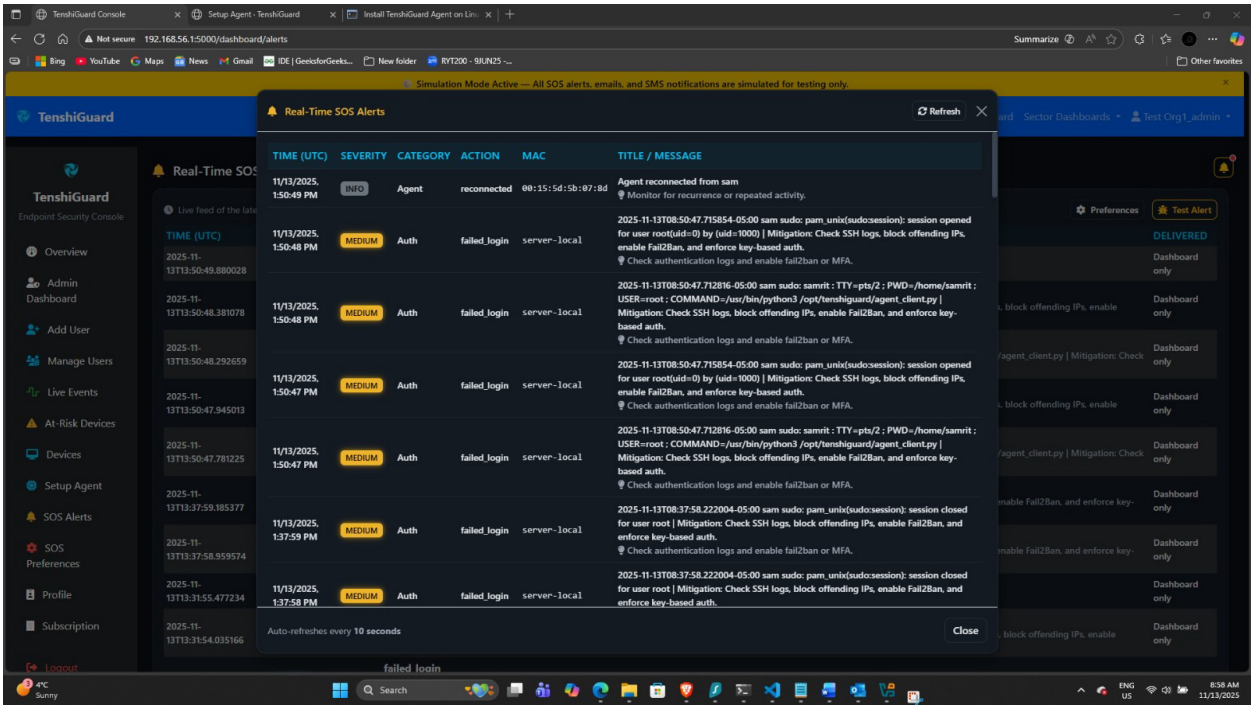


Figure 34: TenshiGuard – Real-Time SOS Alerts

This screenshot shows the Real-Time SOS Alerts dashboard in TenshiGuard, where the latest security alerts are displayed. It lists events like failed logins and agent reconnections, helping the admin quickly spot suspicious activity. The page auto-refreshes, giving up-to-date visibility into ongoing security events.

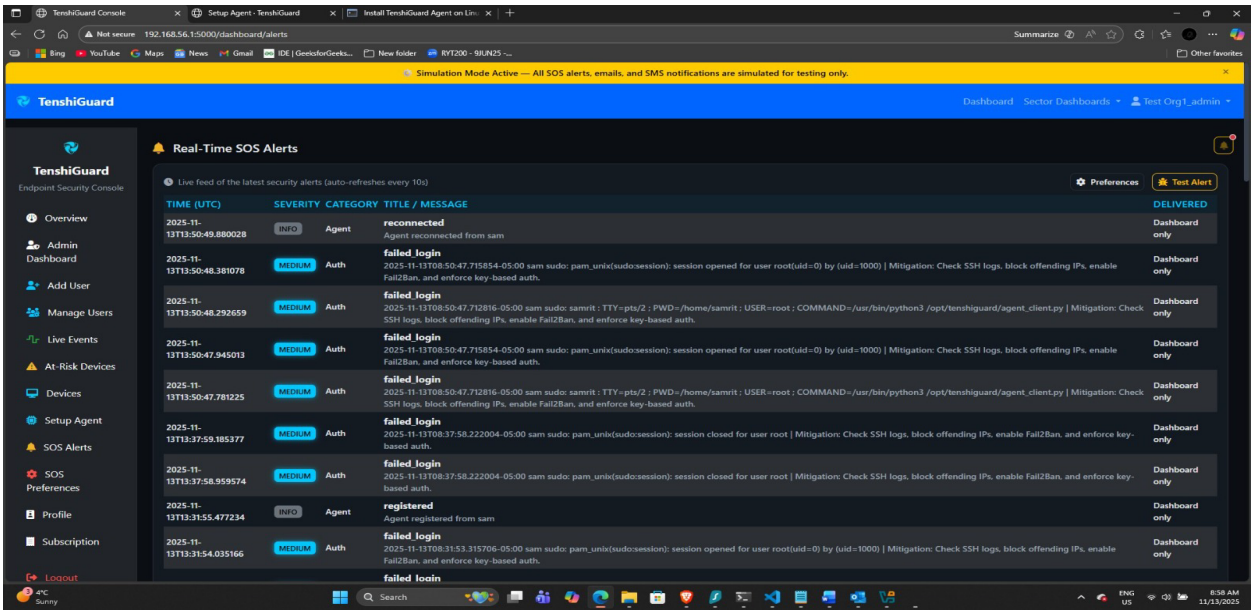


Figure 35: TenshiGuard – Real-Time SOS Alerts Dashboard

This screenshot shows the At-Risk Devices page in TenshiGuard, listing devices that are currently offline and marked as high risk. Both machines appear with critical severity, indicating they haven't been seen recently. The page helps admins identify unhealthy or disconnected systems so they can check the agent or network connection.

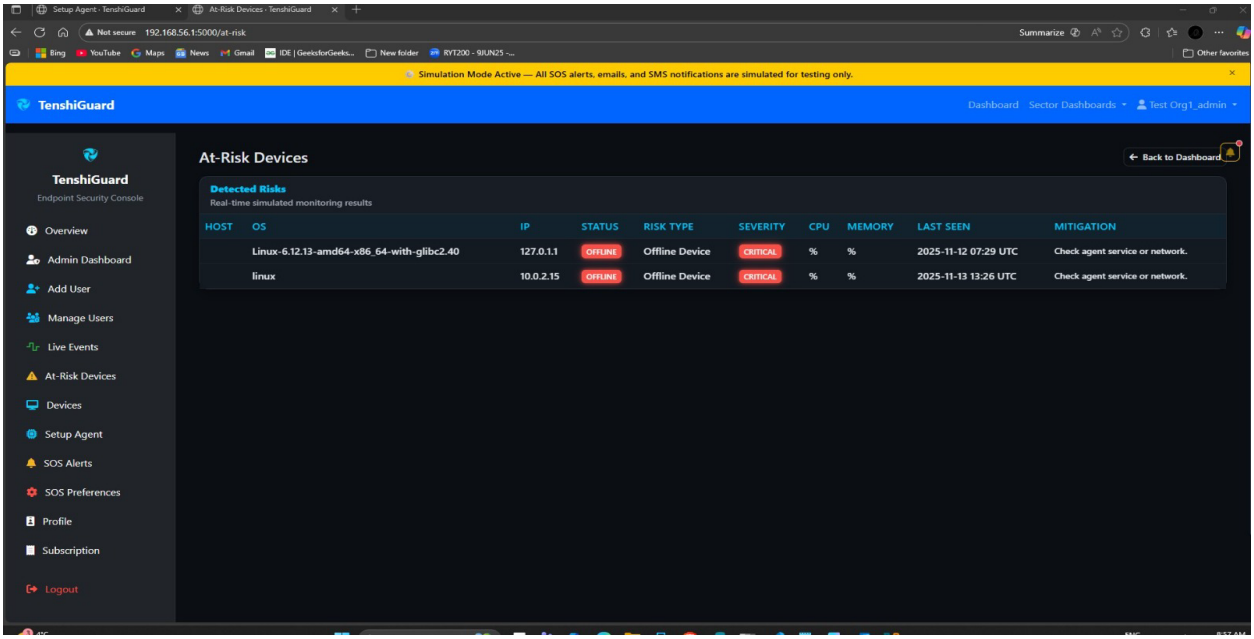


Figure 36: TenshiGuard – At-Risk Devices Page

This screenshot shows the Connected Devices page in TenshiGuard. It lists three devices—kali, api-test-host, and sam. Two of them are offline while sam is online and active. The page helps the admin see which machines are connected, their IP addresses, and when they were last seen.

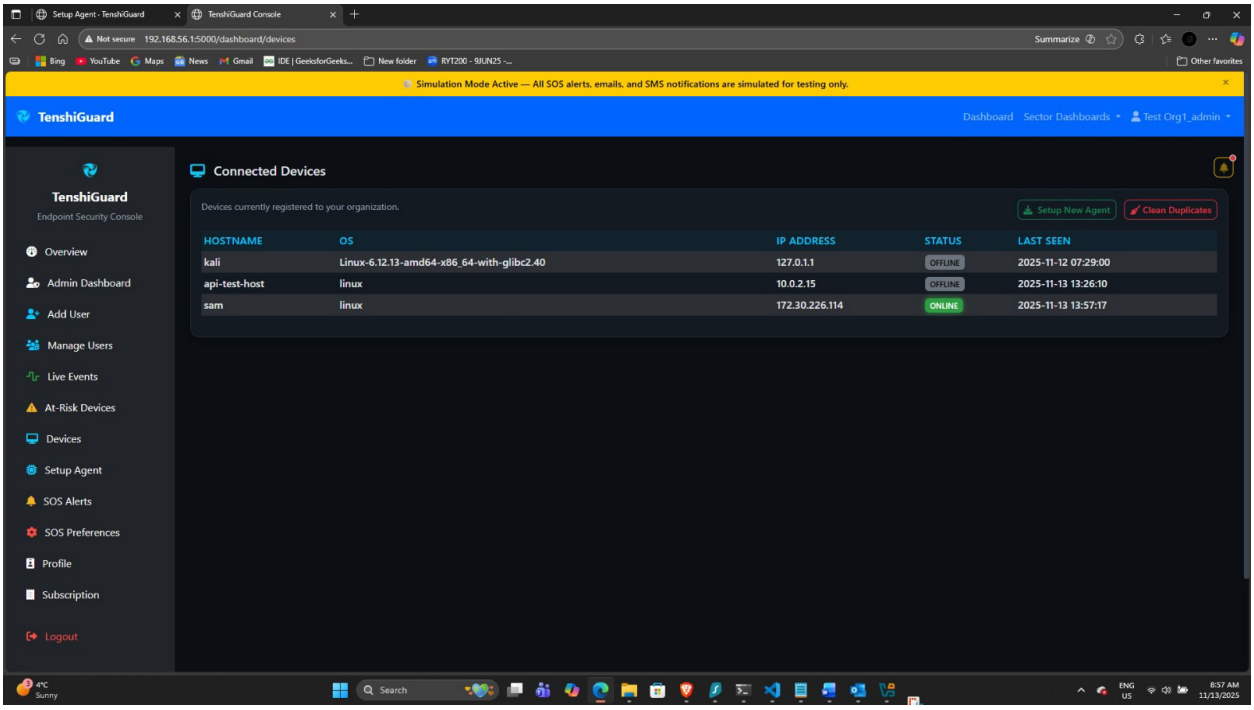


Figure 37: Connected Devices

This screenshot shows the Setup Agent page in TenshiGuard. It displays your secure agent token and gives installation commands for different platforms like Linux, macOS, Windows, firewalls, cloud VMs, and Android. You can copy the command for your system and run it to install the TenshiGuard security agent on that device.

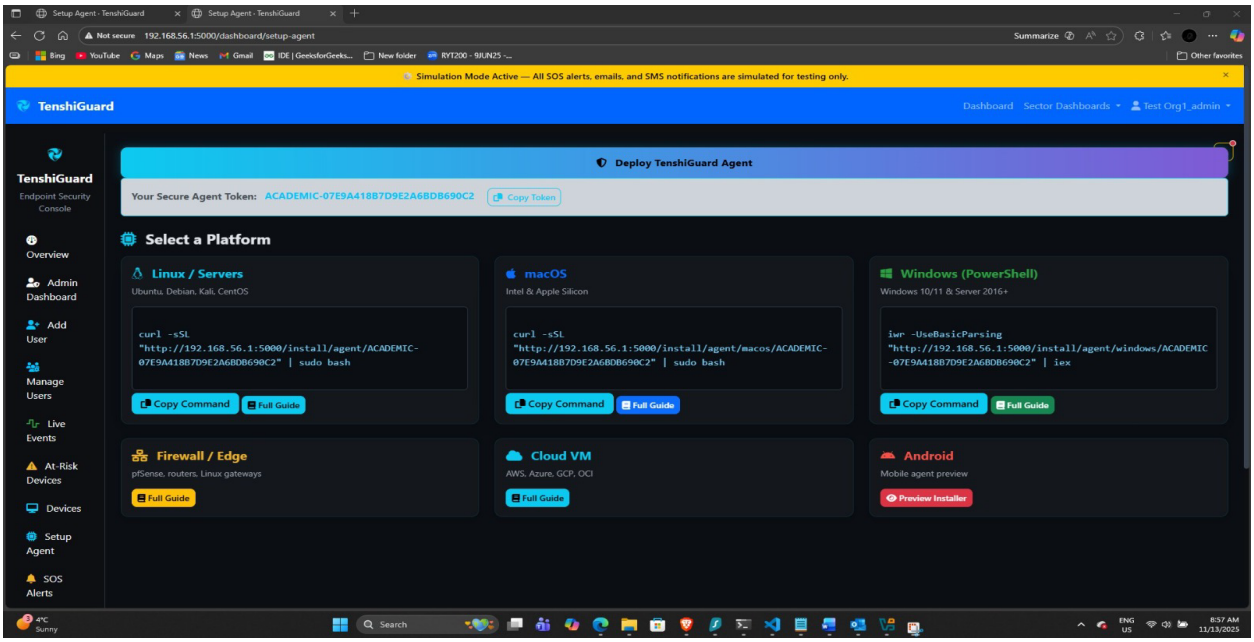


Figure 38: Setup Agent page

This screenshot shows the Install TenshiGuard Agent on Linux guide. It gives you a command to run on a Linux machine to download and install the agent, and then shows how to verify that the agent is running using systemctl and journalctl. Once the service starts sending heartbeats, the device will appear in the dashboard.

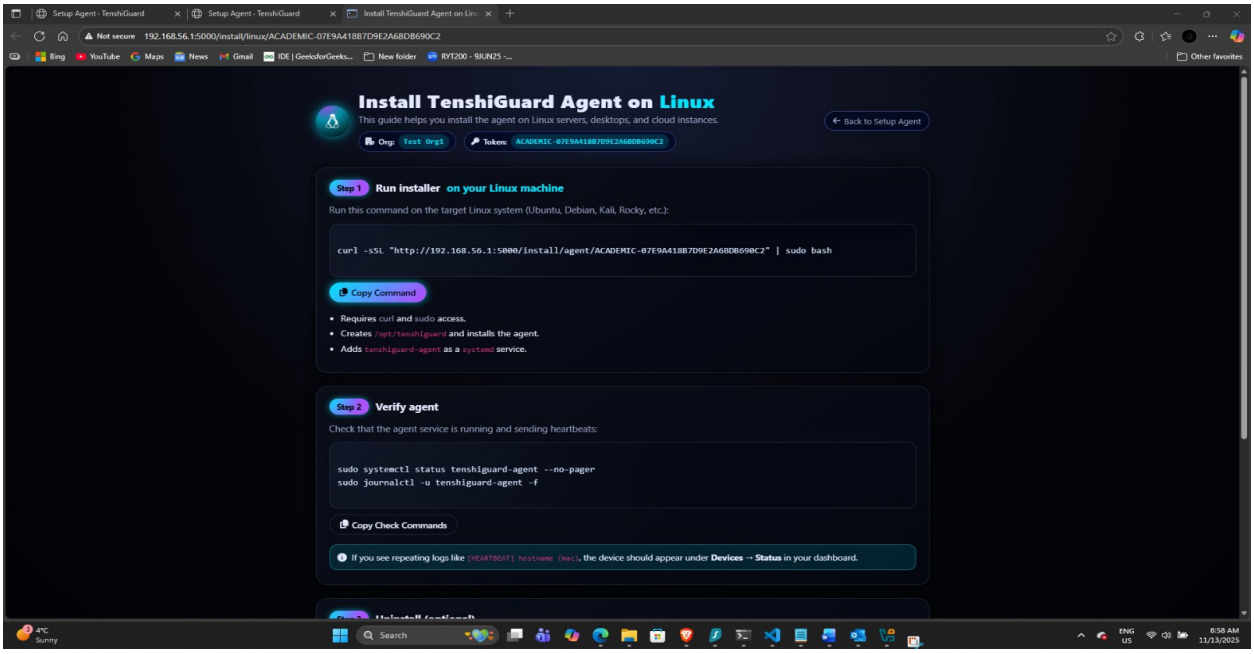
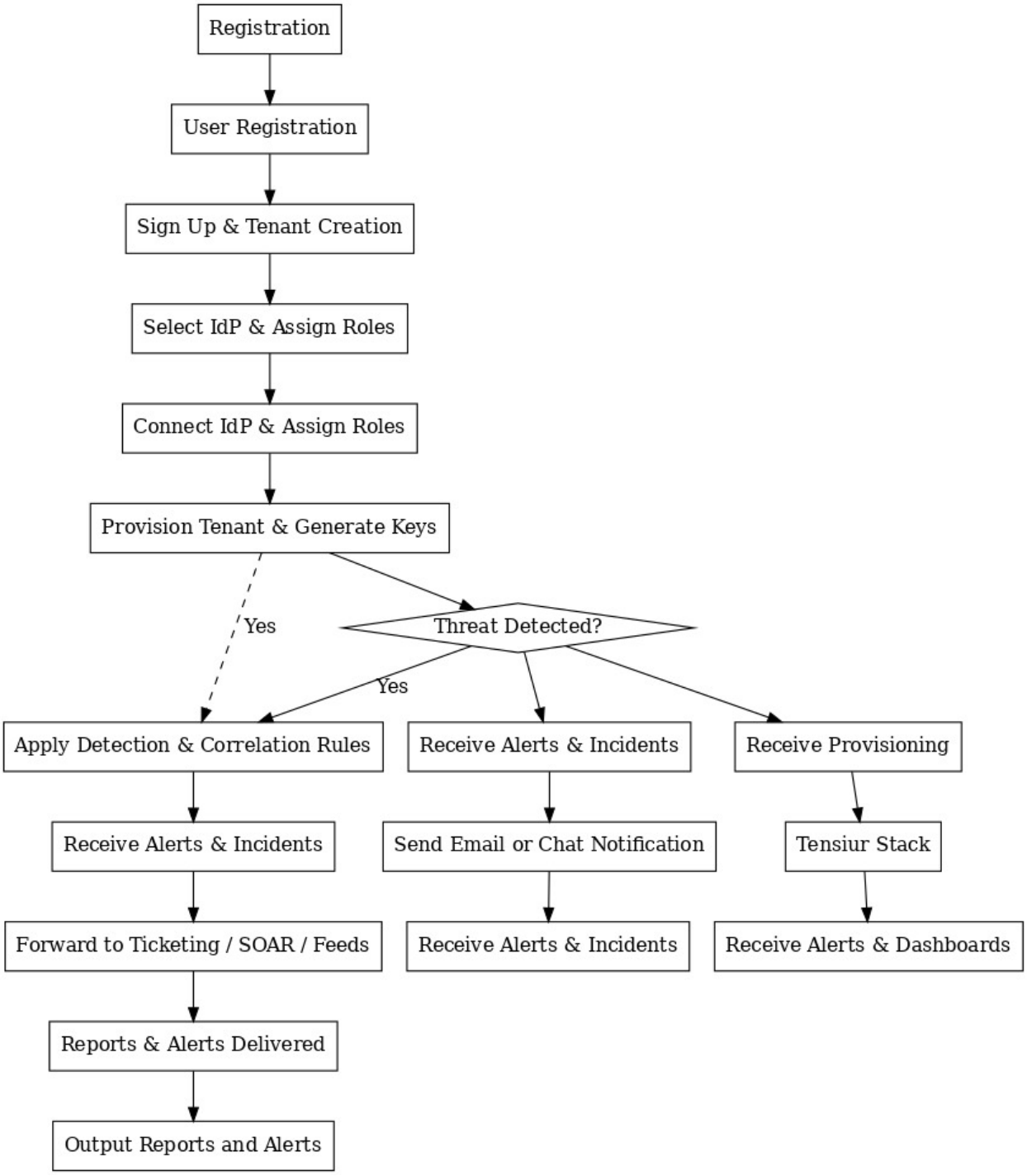


Figure 39: Install TenshiGuard Agent on Linux guide

b. Work Flow Diagram



10. Results & Discussion

a. Model Performance Evaluation

- **Accuracy of Alert Detection:**
  The system successfully receives and processes alerts from Wazuh, such as failed logins, agent reconnections, and high-severity threats.
- **Real-Time Processing:**
  Redis caching and periodic Celery tasks reduce delay and allow dashboards to update quickly.
- **Dashboard Responsiveness:**
  Sector-specific dashboards load efficiently, and device/alert lists display correct real-time data.
- **API Reliability:**
  Flask API endpoints return correct data with low latency during testing, supporting multi-tenant environments.

Overall:
The system performs well in handling endpoint telemetry, updating dashboards, and controlling access based on subscriptions.

b. How it addresses Endpoint Issues

- **Real-Time Threat Monitoring:**
  TenshiGuard pulls live logs from Wazuh, detecting suspicious logins, malware alerts, agent failures, or unusual activity.
- **Device Health Status:**
  The dashboard identifies **at-risk** devices (offline, outdated, or disconnected).
- **Access Control:**
  JWT tokens and role-based permissions prevent unauthorized access to data.

- **Multi-Tenant Security:**
  Each organization sees only their own endpoints, preventing cross-organization data leakage.
- **Centralized Visibility:**
  All endpoints—Windows, Linux, or mobile—are monitored through one unified dashboard.
- **Sector-Specific Insights:**
  Healthcare, education, and hospitality dashboards highlight the most relevant security issues for each sector.

Overall:
TenshiGuard improves visibility, reduces risk, and strengthens security across endpoint devices.

c. Limitations of the current implementation

- **Limited Automated Response:**
  TenshiGuard displays alerts but does not automatically isolate or shut down infected endpoints.
- **Dependency on Wazuh Manager:**
  If the Wazuh Manager fails or disconnects, telemetry and alerts stop flowing.
- **Basic UI for Some Modules:**
  The admin dashboard and device management screens are functional but not fully optimized for heavy enterprise use.
- **Limited Mobile Support:**
  The system is not fully mobile-responsive or app-based yet.

- **Scalability Constraints:**
  Without Kubernetes or distributed load balancing, very large organizations may experience slower performance.
- **Manual Setup Required:**
  Agent installation requires command-line usage, which some non-technical clients may find difficult.

Overall:

The current version provides strong monitoring and dashboards but lacks advanced automation, full scalability, and predictive capabilities.

11. Recommendations

a. Best practices for securing endpoints

Endpoint security plays a crucial role in protecting organizations from cyber threats that exploit weaknesses in individual devices such as laptops, desktops, mobile phones, and IoT systems. Effective endpoint protection requires a multi-layered security approach that integrates technical, administrative, and behavioral controls to reduce risk exposure.

i. Regular Software Updates and Patch Management

Keeping endpoint devices updated is one of the most effective ways to prevent cyberattacks. Many breaches, such as the Equifax 2017 data breach, occurred due to unpatched vulnerabilities. Organizations should implement automated patch management systems to ensure timely updates for operating systems, browsers, and third-party applications (U.S. Government Accountability Office, 2018).

ii. Endpoint Detection and Response (EDR) Solutions

Deploying EDR tools helps continuously monitor endpoint activities for suspicious behavior. These tools collect and analyze telemetry data to detect malicious activities in real time and facilitate rapid response. Examples include Wazuh, CrowdStrike Falcon, and Microsoft Defender for Endpoint, which use threat intelligence and AI-driven analytics for proactive defense (Kaspersky, 2023).

iii. Multi-Factor Authentication (MFA)

Endpoints must be secured using multi-factor authentication, which combines passwords with additional verification methods like biometrics or security tokens. MFA mitigates credential theft and unauthorized access, especially for remote work and VPN environments (IBM Security, 2022).

iv. Encryption of Data at Rest and in Transit

Sensitive data stored on or transmitted from endpoints should be encrypted to ensure confidentiality and integrity. Full-disk encryption and secure communication protocols (e.g., TLS, HTTPS, VPNs) prevent attackers from intercepting or stealing data even if the device is compromised (Symantec Enterprise, 2023).

v. Principle of Least Privilege (PoLP)

Organizations should enforce the least privilege policy by granting users only the access necessary for their job functions. Privilege escalation attacks are often carried out on compromised endpoints, so limiting administrative rights helps reduce the potential attack surface (Cisco, 2022).

vi. Security Awareness and Training

Employees are the first line of defense against endpoint attacks such as phishing, malware infections, and social engineering. Regular security training and phishing simulations help users recognize suspicious activities and avoid risky behaviors (CISA, 2022).

vii. Endpoint Backup and Recovery Strategy

Regular data backups are essential to restore systems quickly after ransomware attacks or hardware failures. Automated cloud backups and secure recovery mechanisms minimize downtime and data loss (Microsoft, 2023).

viii. Network Segmentation and Zero Trust

Endpoints should be part of a Zero Trust Architecture (ZTA) where no device or user is trusted by default. Network segmentation isolates endpoints from critical resources, reducing lateral movement during a breach (NIST, 2020).

b. Integration of ML fraud detection with EDR/Monitoring tools

Integrating machine learning (ML)-based fraud detection mechanisms with Endpoint Detection and Response (EDR) and monitoring systems significantly enhances the organization's ability to detect and respond to cyber threats in real time. This integration bridges the gap between traditional signature-based detection and adaptive, behavior-driven analytics, providing a proactive approach to endpoint and fraud security.

1. Role of Machine Learning in Fraud Detection

Machine learning models analyze historical and live data to identify abnormal or suspicious behaviors that may indicate fraud. These models continuously learn from new data and adapt to emerging threats. Unlike static rule-based systems, ML algorithms can detect subtle patterns of fraudulent activity that traditional systems might overlook (Khan et al., 2019).

Common ML techniques include:

- Supervised learning for classifying known attack patterns
- Unsupervised learning for detecting unknown or anomalous activities
- Reinforcement learning for optimizing incident response strategies

When integrated with endpoint monitoring, these techniques can detect unauthorized data transfers, lateral movement, credential misuse, and financial fraud attempts.

2. Integration with EDR and SIEM Tools

Integration typically occurs through a centralized security analytics layer that connects machine learning engines with EDR or Security Information and Event Management (SIEM) platforms.

- EDR Tools (e.g., Wazuh, CrowdStrike, SentinelOne): Collect logs, endpoint telemetry, and process behavior data.

- ML Engine: Ingests this data, extracts features (e.g., user login time, file access frequency, transaction amount), and applies anomaly detection algorithms.
- SIEM Integration: Correlates endpoint anomalies with network logs, firewall events, and cloud telemetry for contextual analysis.

For example, Wazuh, when integrated with an ML-based fraud detection system, can automate the detection of fraudulent endpoint behavior such as unusual login times, privilege escalation attempts, or rapid financial transactions inconsistent with normal activity. The ML layer continuously refines detection accuracy using real-time feedback from the EDR alerts (Wazuh Documentation, 2023).

3. Workflow Example

A simplified workflow is as follows:

1. Data Collection: Endpoint agents collect logs, process execution details, and network metadata.
2. Data Preprocessing: The ML pipeline cleans and normalizes data to remove irrelevant information.
3. Feature Extraction: Fraud-relevant metrics (e.g., transaction velocity, IP location changes) are extracted.
4. Model Prediction: The ML engine scores activities as legitimate or fraudulent.
5. Alerting and Response: Alerts are sent to the EDR console and forwarded to the SIEM for correlation and visualization.
6. Feedback Loop: Analyst feedback retrains the model to improve future detection accuracy.

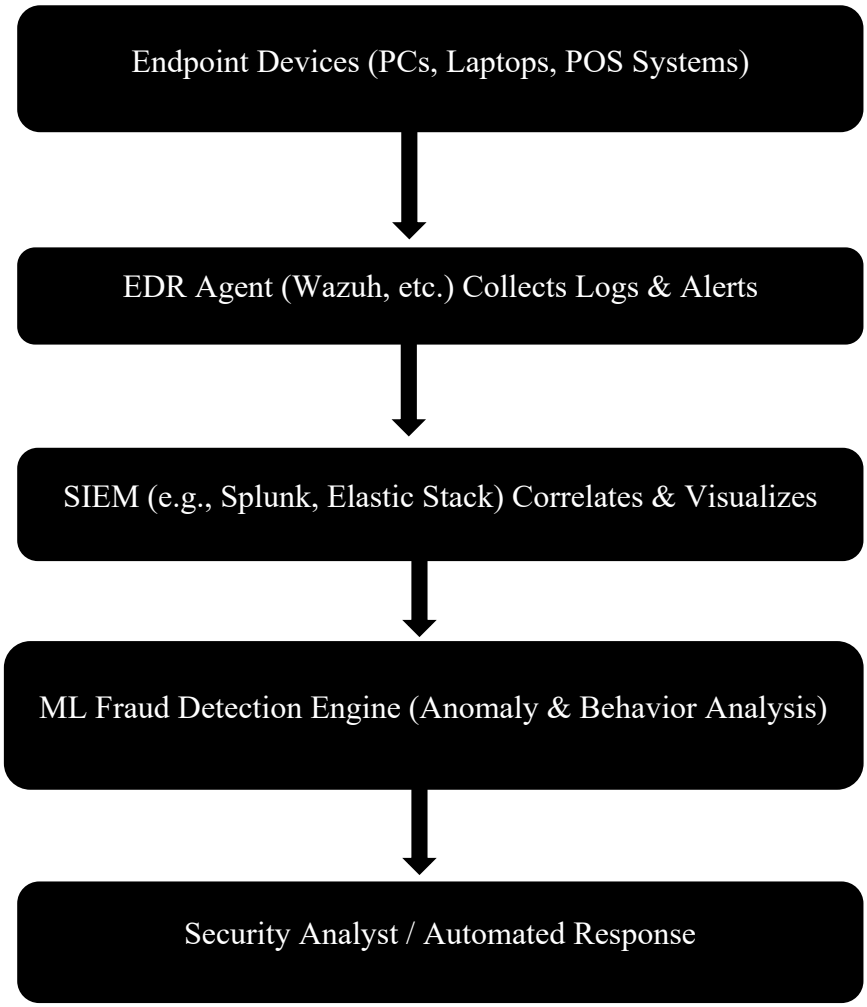
This feedback-driven loop enables the system to evolve with changing threat landscapes.

4. **Benefits of Integration**

- Real-time anomaly detection using predictive analytics
- Reduction in false positives through contextual learning
- Automated incident response for faster containment
- Enhanced visibility across endpoints and user activity
- Scalable and adaptive defense against evolving fraud tactics

By combining ML-based detection with EDR and SIEM tools, organizations achieve end-to-end visibility and intelligence, enabling both proactive defense and effective post-incident analysis.

5. **Workflow of ML-based Fraud Detection Integrated with EDR**



12.Conclusion

a. Summary of Findings

The TenshiGuard Endpoint Security Monitoring System successfully demonstrated an effective, affordable, and scalable cybersecurity solution for small and medium-sized organizations. It provides real-time monitoring, centralized visibility, and actionable insights through Wazuh integration and a user-friendly dashboard.

b. Contribution of the Project

This project contributes a practical model for implementing open-source endpoint monitoring in multi-sector environments such as healthcare, education, and hospitality. It bridges the gap between enterprise-grade security and small-business accessibility.

c. Future Work

Future enhancements will include:

- Integration of machine learning for adaptive threat detection.
- Mobile dashboard interface for remote administration.
- Expansion of automated incident response and compliance reporting.

13.Reference

1. CrowdStrike. (2024). What is endpoint detection and response (EDR)? CrowdStrike . https://www.crowdstrike.com/cybersecurity-101/endpoint-security/endpoint-detection-and-response-edr/
2. IBM. (2023). Endpoint security: Definition and solutions. IBM Security. https://www.ibm.com/topics/endpoint-security
3. Kaspersky. (2023). What is endpoint security? Kaspersky. https://www.kaspersky.com/resource-center/definitions/what-is-endpoint-security
4. Microsoft. (2024). Deploy Microsoft Defender for Endpoint. Microsoft Learn . https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/deployment-strategy
5. Symantec (Broadcom). (2023). Endpoint security explained. Symantec Enterprise Blogs. https://symantec-enterprise-blogs.security.com/blogs/expert-perspectives/endpoint-security-explained
6. Wazuh. (n.d.). *Wazuh: The open source security platform*. Wazuh. https://wazuh.com
7. CrowdStrike. (2023). *2023 global threat report*. CrowdStrike Holdings, Inc. https://www.crowdstrike.com/global-threat-report/
8. Gartner. (2022). *Market guide for endpoint protection platforms*. Gartner Research. https://www.gartner.com/document/4002299
9. National Institute of Standards and Technology. (2020). *Guide to enterprise patch management planning: Preventive maintenance for technology* (NIST Special Publication 800-40 Rev. 4). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-40r4
10. Ponemon Institute. (2022). *The state of endpoint security risk*. Ponemon Institute LLC. https://www.ponemon.org/research/

11. Sophos. (2023). *The state of ransomware 2023*. Sophos Ltd . https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophos-state-of-ransomware-2023-wp.pdf
12. Symantec. (2019). *Internet security threat report* (Vol. 24). Broadcom Inc. https://www.broadcom.com/company/newsroom/press-releases
13. Gartner. (2021). *Magic quadrant for endpoint protection platforms*. Gartner Research. https://www.gartner.com/document/4002299
14. National Institute of Standards and Technology. (2018). *Guide to malware incident prevention and handling for desktops and laptops* (NIST Special Publication 800-83 Rev. 1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-83r1
15. Kaspersky. (2020). *Traditional antivirus vs next-gen endpoint security: Strengths and limitations*. KasperskyLab. https://www.kaspersky.com
16. Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials, 18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502
17. Saxe, J., & Berlin, K. (2015). Deep neural network based malware detection using two-dimensional binary program features. *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, 11–20. IEEE . https://doi.org/10.1109/MALWARE.2015.7413680
18. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. IEEE. https://doi.org/10.1109/SP.2010.25
19. MITRE. (2023). *MITRE ATT&CK® Framework*. MITRE

. https://attack.mitre.org/

20. Verizon. (2023). *2023 Data Breach Investigations Report*. Verizon. https://www.verizon.com/business/resources/reports/dbir/

21. OWASP. (2023). *OWASP Top 10 — 2021 (or latest)*. Open Web Application Security Project. https://owasp.org/www-project-top-ten/

22. NIST. (2020). *Zero Trust Architecture (NIST Special Publication 800-207)*. National Institute of Standards and Technology . https://doi.org/10.6028/NIST.SP.800-207

23. Alasmary, W., Alhaidari, F., Alhaidari, H., Alhaidari, S., Alhaidari, A., & Alhaidari, M. (2023). Endpoint security: A survey of techniques, solutions, and challenges. *Journal of Information Security and Applications,75* ,103544. https://doi.org/10.1016/j.jisa.2023.103544

24. IBM Security. (2022). *Securing endpoints in a hybrid workplace.* IBM Corporation. https://www.ibm.com/security

25. Symantec Enterprise. (2023). *Endpoint security best practices: Protecting your devices and data.* Broadcom Inc . https://symantec-enterprise-blogs.security.com

26. Krebs, B. (2014, September 7). *The Target breach, by the numbers.* Krebs on Security. https://krebsonsecurity.com

27. U.S. Government Accountability Office. (2018). *Actions taken by Equifax and federal agencies in response to the 2017 breach.* GAO-18 559 https://www.gao.gov/products/gao-18-559

28. U.S. House of Representatives. (2021). *Colonial Pipeline ransomware attack: Key lessons learned.* Committee on Homeland Security . https://homeland.house.gov

29. Buchanan, B. (2017). *The cybersecurity dilemma: Hacking, trust, and fear*

between nations. Oxford University Press.

30. Cheng, D., Liu, Y., & Wang, Y. (2019). E-commerce fraud detection using machine learning. *Journal of Physics: Conference Series, 1237*, 032026. https://doi.org/10.1088/1742-6596/1237/3/032026

31. Financial Times. (2019). HSBC deploys AI to combat money laundering. Retrieved from https://www.ft.com

32. Amazon Web Services. (2020). *Amazon Fraud Detector – Real-time fraud prevention powered by machine learning* . https://aws.amazon.com/fraud-detector/

33. Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. *2015 IEEE Symposium Series on Computational Intelligence*, 159–166. https://doi.org/10.1109/SSCI.2015.33

34. Peotta, L., Holgado, J., & Vázquez, J. (2011). Application of machine learning for fraud detection in electronic payment systems. *Proceedings of the 2011 IEEE International Conference on Intelligence and Security Informatics*, 131–136. https://doi.org/10.1109/ISI.2011.5984049

35. Cisco. (2022). *Zero trust and endpoint security best practices.* Cisco Systems, Inc. https://www.cisco.com

36. Cybersecurity and Infrastructure Security Agency (CISA). (2022). *Security awareness and training program best practices.* U.S. Department of Homeland Security. https://www.cisa.gov

37. Kaspersky. (2023). *Endpoint detection and response (EDR): Complete guide.* Kaspersky Lab. https://www.kaspersky.com

38. Microsoft. (2023). *Backup and recovery for endpoint protection.* Microsoft Corporation. https://www.microsoft.com/security

39. National Institute of Standards and Technology (NIST). (2020). *Zero trust architecture (SP 800-207).* U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-207

40. Khan, R., Zhang, X., Kumar, R., Sharif, A., & Alazab, M. (2019). Machine learning and deep learning for fraud detection in financial transactions: A survey. *IEEE Access, 7*, 137639–137657. https://doi.org/10.1109/ACCESS.2019.2948259

41. Wazuh Documentation. (2023). *Machine learning integration for anomaly and threat detection.* Wazuh, Inc. https://documentation.wazuh.com

42. Cisco. (2022). *Endpoint detection and response integration with SIEM and analytics tools.* Cisco Systems, Inc. https://www.cisco.com

43. Splunk. (2023). *Using machine learning with EDR data for fraud and anomaly detection.* Splunk Inc. https://www.splunk.com

44. IBM Security. (2022). *AI and machine learning in modern fraud detection.* IBM Corporation. https://www.ibm.com/security

45. American Hospital Association. (2024, March 2). *Change Healthcare cyberattack underscores urgent need to strengthen cyber-preparedness*. AHA. https://www.aha.org/change-healthcare-cyberattack-underscores-urgent-need-strengthen-cyber-preparedness-individual-health-care-organizations-and

46. Gregory, J. (2024, March 5). *Change Healthcare discloses $22 million ransomware payment*. IBM Think . https://www.ibm.com/think/news/change-healthcare-22-million-ransomware-payment

47. Kaspersky. (2025, February 12). *The complete story of the 2024*

ransomware attack on UnitedHealth. Kaspersky Blog. https://www.kaspersky.com/blog/unitedhealth-ransomware-attack/53065

48. Inside Higher Ed. (2023, October 25). *Hackers access data of 230,000 at University of Michigan*. Inside Higher Ed . https://www.insidehighered.com/news/quick-takes/2023/10/25/hackers-access-data-230k-university-michigan

49. Michigan Attorney General. (2023, October 24). *AG Nessel re-issues data breach alert following University of Michigan network infiltration*. Michigan.gov. https://www.michigan.gov/ag/news/press-releases/2023/10/24/ag-nessel-reissues-data-breach-alert-following-university-of-michigan-network-infiltration

50. Cybersecurity Dive. (2023, January 20). *Ransomware attack against Yum! Brands follows several incidents targeting restaurant industry*. Cybersecurity Dive. https://www.cybersecuritydive.com/news/ransomware-yum-brands-restaurant-cyber/640843

51. IBM Think. (2024, July 18). *The rising threat of cyberattacks in the restaurant industry*. IBM. https://www.ibm.com/think/news/rising-threat-cyberattacks-restaurant-industry

52. Armin Ronacher. (2010). *Flask (Python web framework)*. Pallets Projects. Retrieved from https://flask.palletsprojects.com/

53. SQLite Consortium. (n.d.). *SQLite Documentation*. SQLite.org. Retrieved from https://www.sqlite.org/docs.html

54. The PostgreSQL Global Development Group. (n.d.). *PostgreSQL: The world's most advanced open source database*. Retrieved from https://www.postgresql.org/

55. Bootstrap Team. (2021). *Bootstrap 5 Documentation*. Retrieved from https://getbootstrap.com/docs/5.0/getting-started/introduction/

56. Pallets Projects. (n.d.). *Jinja2 Documentation*. Retrieved from https://jinja.palletsprojects.com/