Project Work 1(19AI702)

AI Powered Advanced Web Application Firewall

Submitted by:

V.SRIRAM (212222103002)

R.SUROTHAAMAN (212222103003)

M.PAVITHRA (212222100032)

2022-2026 Batch

TEAM NO: 58

Under the guidance of:

V.SWEDHA

NAME OF THE GUIDE

V.SWEDHA

Designation,Department of IT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(CS)

SAVEETHA ENGINEERING COLLEGE

(Autonomous Institution – UGC, Govt. of India)

(Affiliated to Anna University, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)

Saveetha Nagar, Thandalam, Chennai-602 105, TamilNadu, INDIA.

Agenda

1. Introduction
2. Statement of the Problem
3. Scope of the Project
4. Methodology
5. Architectural Diagram
6. System Flow
7. Algorithm Used
8. Design
   - Use Case Diagram
   - Class Diagram
   - Sequence Diagram
9. Implementation
10. Important Code Segments
11. Output and Results
12. Test Cases
13. Conclusion
14. Future Work
15. References

Introduction

- **AI-Powered Advanced Web Application Firewall (WAF)** is a security system designed to protect web applications from cyber threats.
- With the rapid growth of web applications, attacks such as **SQL Injection, Cross-Site Scripting (XSS), and DDoS** have become increasingly common.
- Traditional WAFs rely on **static, rule-based mechanisms**, which are often ineffective against evolving or unknown attacks.
- **Artificial Intelligence (AI) and Machine Learning (ML)** enable intelligent traffic analysis and **real-time threat detection**.
- This project integrates AI/ML techniques to create an **adaptive and intelligent WAF** that improves security dynamically.

Statement of the Problem

Develop an **AI-powered WAF** to protect web applications from cyber threats.
Use **machine learning models** to detect malicious web traffic.
Provide a **real-time monitoring dashboard** with alerts.
Improve **accuracy and efficiency** of threat detection.
Target Audience: Web developers, IT security teams, enterprises.

Deliverables:

- Trained AI/ML model for traffic analysis.
- Functional WAF system with monitoring dashboard.
- Documentation of architecture, model, and performance.

Scope of the project

Focus:

- AI-powered Web Application Firewall (WAF)
- Real-time detection and blocking of cyber threats
- Dashboard for monitoring traffic and alerts

Target Audience:

- Web developers & administrators
- Enterprises needing web security
- IT security professionals

Deliverables:

- AI/ML models detecting SQL Injection, XSS, DDoS
- Functional WAF with monitoring & alerts
- Documentation & scalable deployment

Scope of the project

Inclusions:

- Collect and preprocess web traffic logs.
- Extract features from HTTP requests and patterns.
- Build AI/ML detection models.
- Develop a user-friendly dashboard.
- Implement real-time threat blocking and alerts.

Exclusions:

- No hardware-level network security.
- Not for non-web or offline applications.
- No integration with third-party cloud security services.

Scope of the project

- **Data Collection:** Use web traffic logs with labeled normal and malicious requests.
- **Preprocessing:** Clean data and extract relevant features for ML analysis.
- **Limitations:**
  - Dataset may not include all types of web attacks.
  - Real-time performance may depend on server capacity.
  - AI model accuracy may vary with unseen traffic patterns.
  - Continuous updates needed for evolving threats.

Methodology

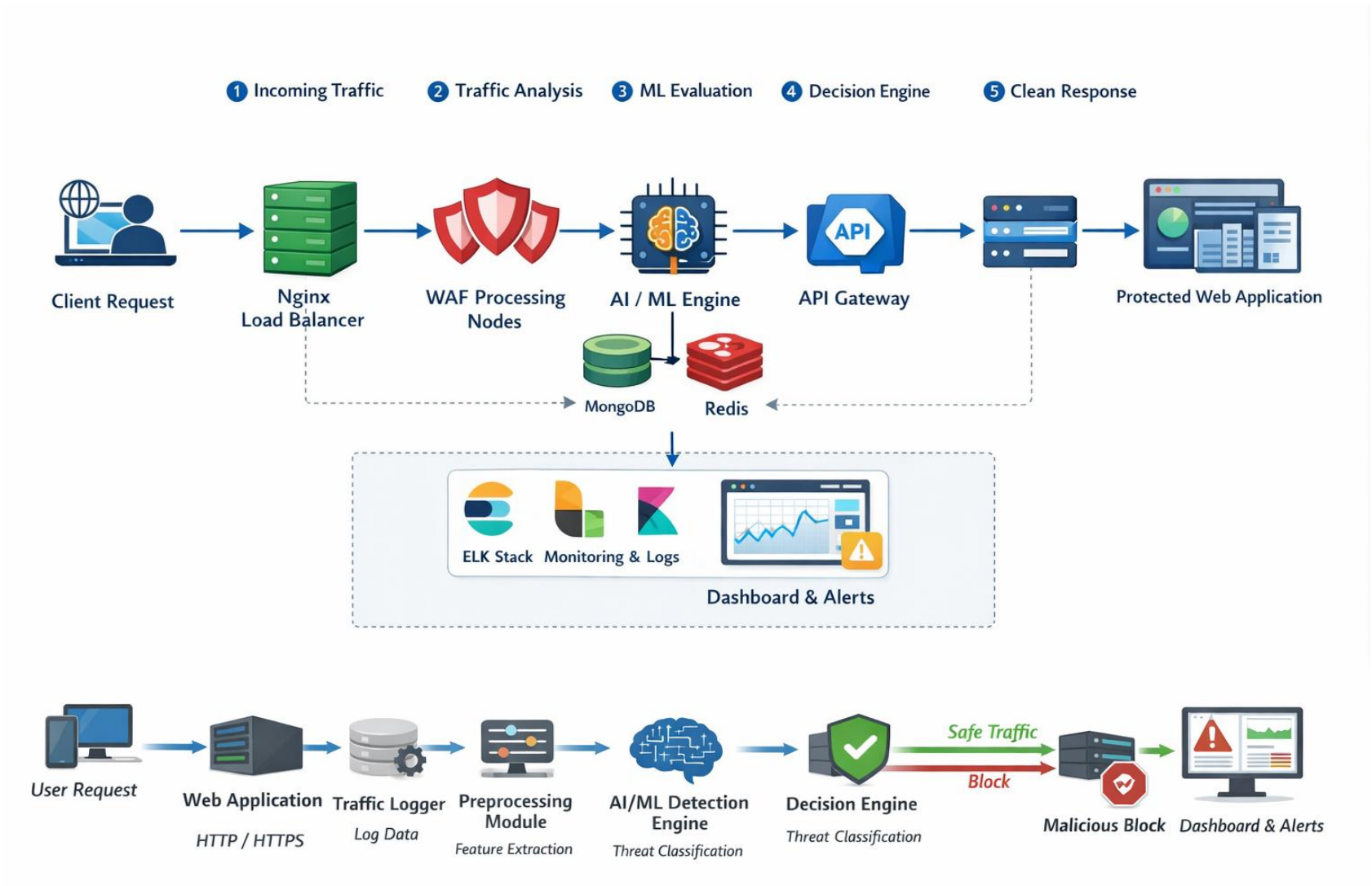
- **Data Collection:** Gather web traffic logs (normal and malicious requests).
- **Data Preprocessing:** Clean data and extract features from HTTP requests and headers.
- **AI/ML Model Development:** Train models to classify requests as safe or malicious.
- **Web Dashboard:** Build a user-friendly interface for monitoring traffic and alerts.
- **Integration & Testing:** Combine AI model with dashboard; validate accuracy and performance.

Methodology

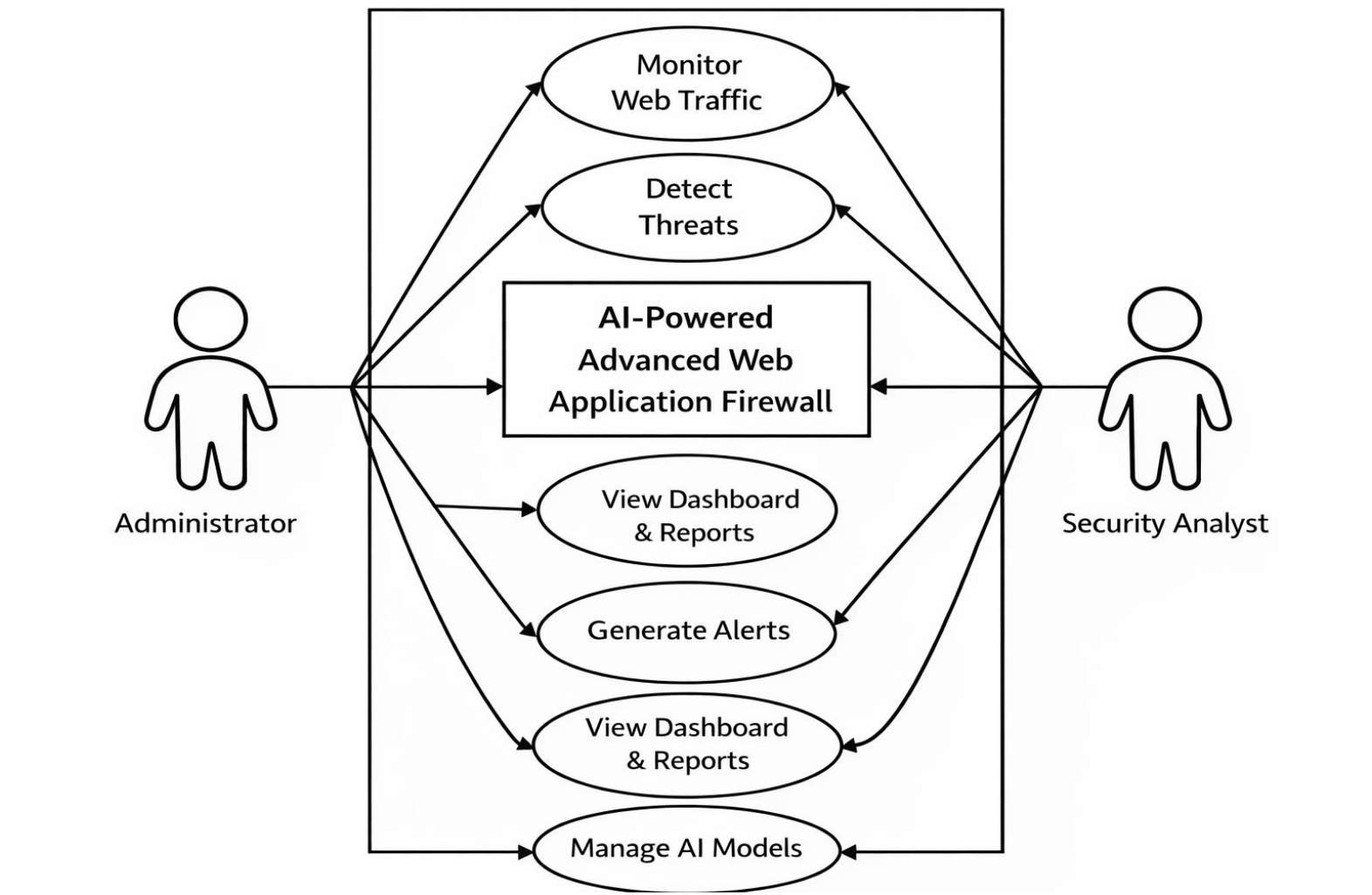
Traffic Analysis & AI Integration:

- **Collect Traffic Data:** Gather normal and malicious web requests.
- **Preprocess Data:** Clean, normalize, and extract features from HTTP requests.
- **Develop AI/ML Model:** Train models to classify requests as safe or malicious.
- **Web Dashboard:** Create a user-friendly interface for monitoring and alerts.
- **Integration & Testing:** Combine AI model with dashboard; test for accuracy and performance.

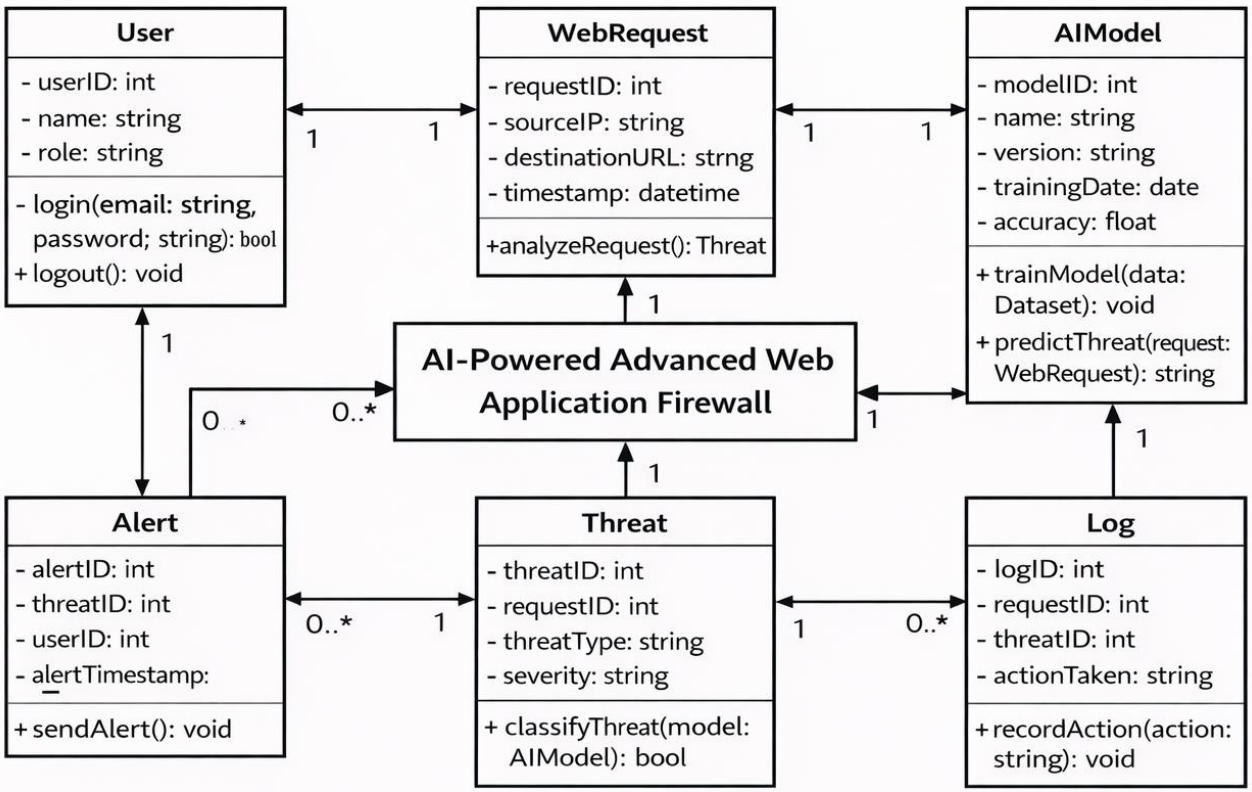
Architecture Diagram/Flow



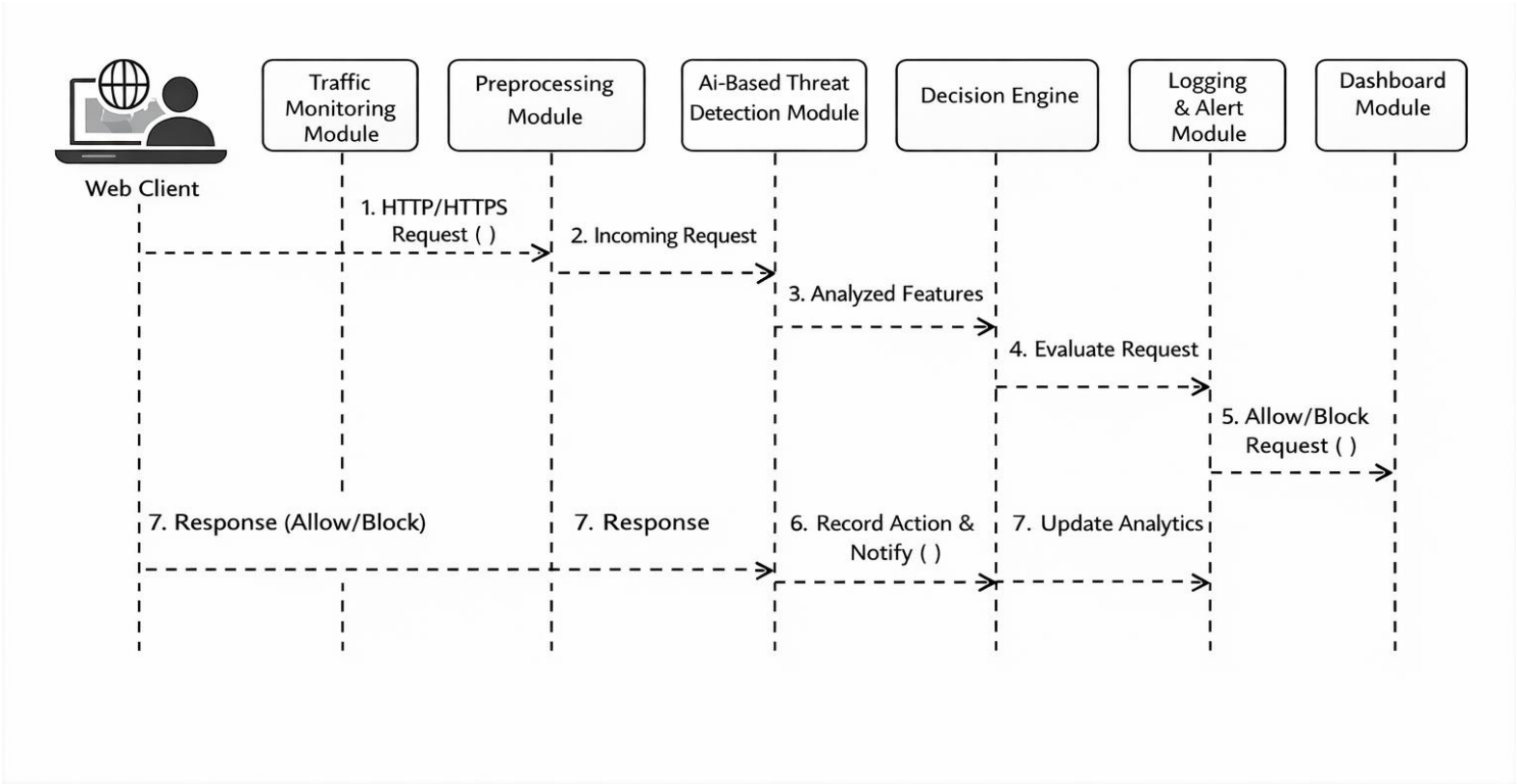
Design-Use Case Diagram



Design-Class Diagram



Sequence Diagram



Algorithms used

Machine Learning–Based Threat Detection Model:

- The Machine Learning–based Threat Detection Model is used to analyze incoming web traffic in real time.
- It classifies HTTP requests as normal or malicious based on extracted features from headers, URLs, and payloads.
- It is trained using labeled web traffic data containing both legitimate and attack requests.
- It improves detection accuracy without increasing response latency.
- It enables adaptive learning by identifying new attack patterns and reducing false positives during request filtering.

Algorithms used

Anomaly Detection and Dynamic WAF Rule Generation Algorithm:

- This algorithm identifies abnormal behavior by comparing incoming traffic patterns with normal baseline traffic.
- It detects attacks such as SQL Injection, Cross-Site Scripting (XSS), and brute-force attempts.
- Based on detected threats, it dynamically generates and updates Web Application Firewall rules.
- It automatically blocks malicious requests and IP addresses in real time.
- It ensures continuous protection by adapting firewall rules according to evolving attack behaviors.

Hardware and software selection

Hardware:

- The project uses the following hardware components:
- Server or cloud system for hosting the WAF
- Load balancer (Nginx) for traffic distribution
- Storage for logs and ML models
- Network infrastructure for HTTP/HTTPS traffic

Software:

- The project uses the following software components:
- Python programming language
- Flask / FastAPI framework
- Machine learning using Scikit-learn
- Nginx web server and load balancer
- Docker and Docker Compose
- Prometheus and Grafana for monitoring
- Linux operating system

Implementation

- The system is implemented using Python-based web technologies and containerized services.
- Nginx is used as a load balancer to route incoming traffic to multiple WAF nodes.
- HTTP requests are intercepted and analyzed using a security middleware.
- Relevant features are extracted and passed to the machine learning engine.
- The ML model classifies requests as normal or malicious.
- Based on the prediction, dynamic WAF rules are generated to allow or block traffic.
- All security events and metrics are logged and visualized using monitoring tools.

Important Code segments

Dynamic WAF Rule Generation (AI Driven)

```
def generate_rules_from_threats(self, threats):
    if len(threat_list) >= 3:
        rule = self._generate_rule_for_threat_type()
```

Description:

- Uses ML threat predictions to create **dynamic WAF rules**
- Generates rules for **SQL Injection, XSS, Bot attacks**
- Automatically blocks **IPs, URLs, User-Agents**
- Prevents duplicate rules and manages rule expiry

Important Code segments

Nginx Rule Conversion & Validation

```
def to_nginx_config(self):
    return f"deny {self.condition};"
def _validate_nginx_config(self, config):
    return syntax_check and security_check
```

Description:

- Converts AI-generated rules into **valid Nginx configuration**
- Validates:
  - Syntax correctness
  - Regex patterns
  - Security directives
- Ensures **safe deployment without server crash**

Important Code segments

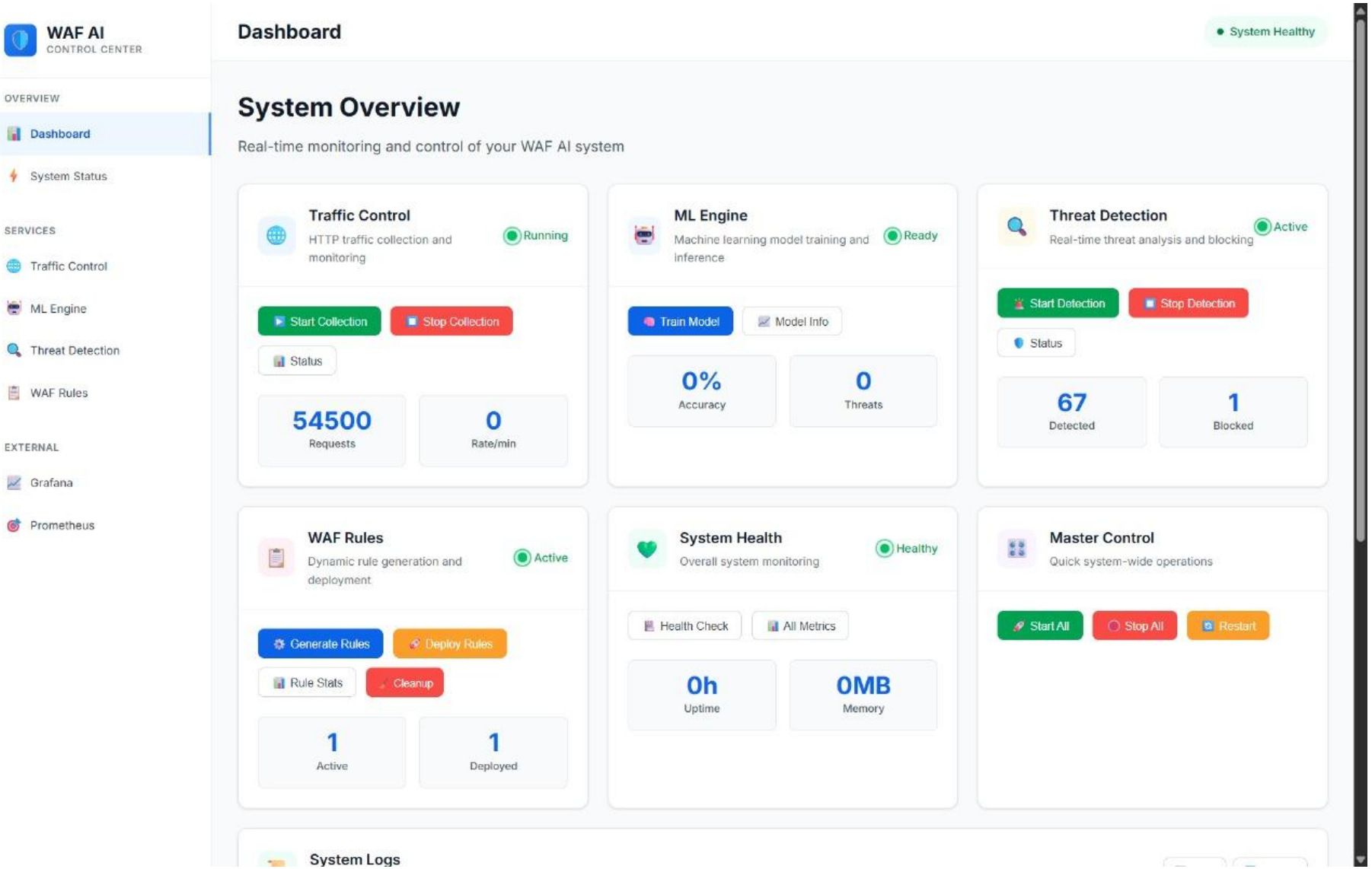
Secure Rule Deployment Across Nginx Nodes

```
async def deploy_rules_to_all_nodes(self, config):
    await self._deploy_to_node(node, config)
```

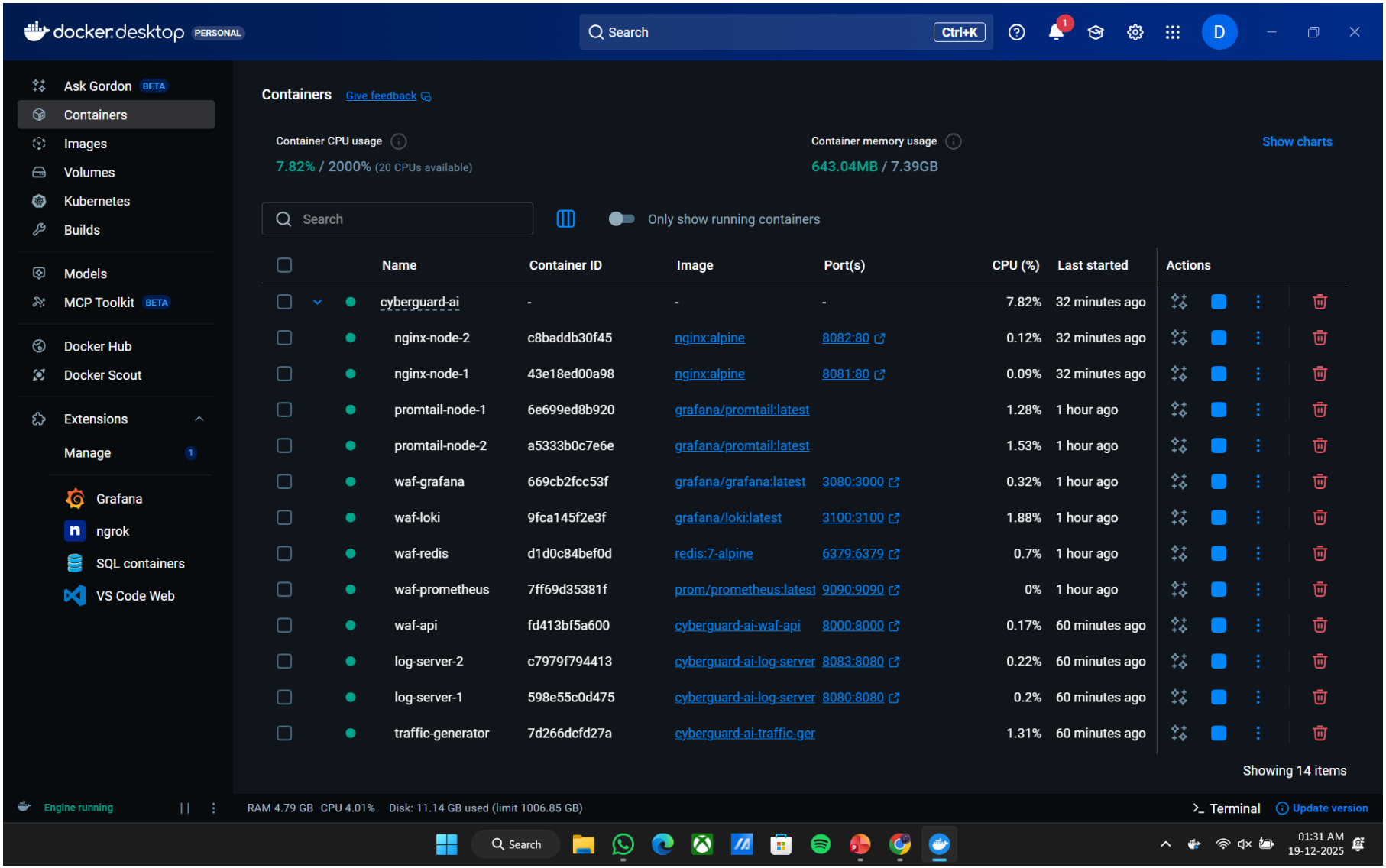
Description:

- Deploys WAF rules to **multiple Nginx nodes**
- Supports **Docker, API, and SSH-based deployment**
- Includes **backup, rollback, and verification**
- Ensures **high availability and fault tolerance**

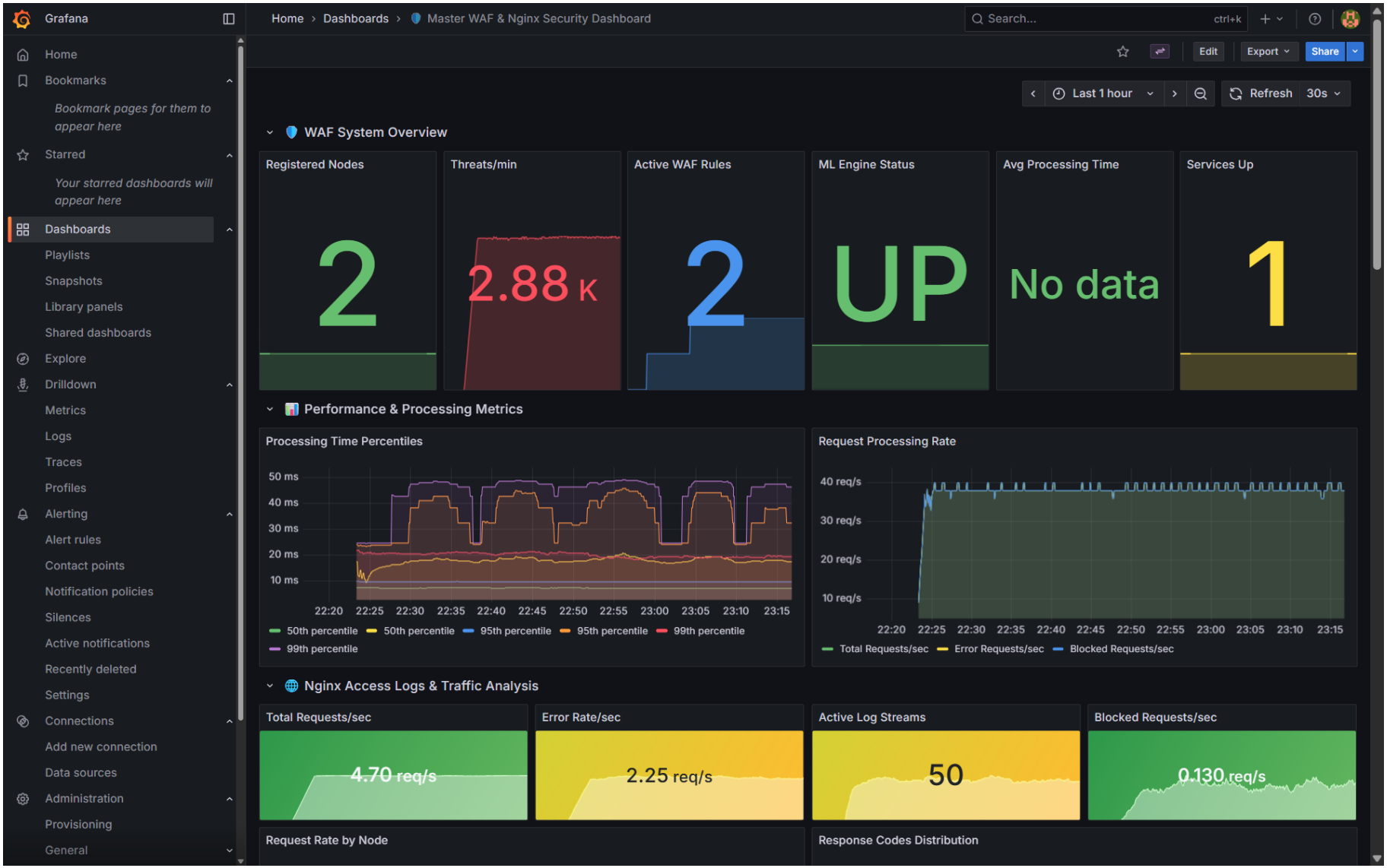
Output


WAF AI
CONTROL CENTER
OVERVIEW
Dashboard
System Status
SERVICES
Traffic Control
ML Engine
Threat Detection
WAF Rules
EXTERNAL
Grafana
Prometheus
Dashboard
System Healthy
System Overview
Real-time monitoring and control of your WAF AI system
Traffic Control
HTTP traffic collection and monitoring
Running
Start Collection
Stop Collection
Status
54500
Requests
0
Rate/min
ML Engine
Machine learning model training and inference
Ready
Train Model
Model Info
0%
Accuracy
0
Threats
Threat Detection
Real-time threat analysis and blocking
Active
Start Detection
Stop Detection
Status
67
Detected
1
Blocked
WAF Rules
Dynamic rule generation and deployment
Active
Generate Rules
Deploy Rules
Rule Stats
Cleanup
1
Active
1
Deployed
System Health
Overall system monitoring
Healthy
Health Check
All Metrics
0h
Uptime
0MB
Memory
Master Control
Quick system-wide operations
Start All
Stop All
Restart
System Logs

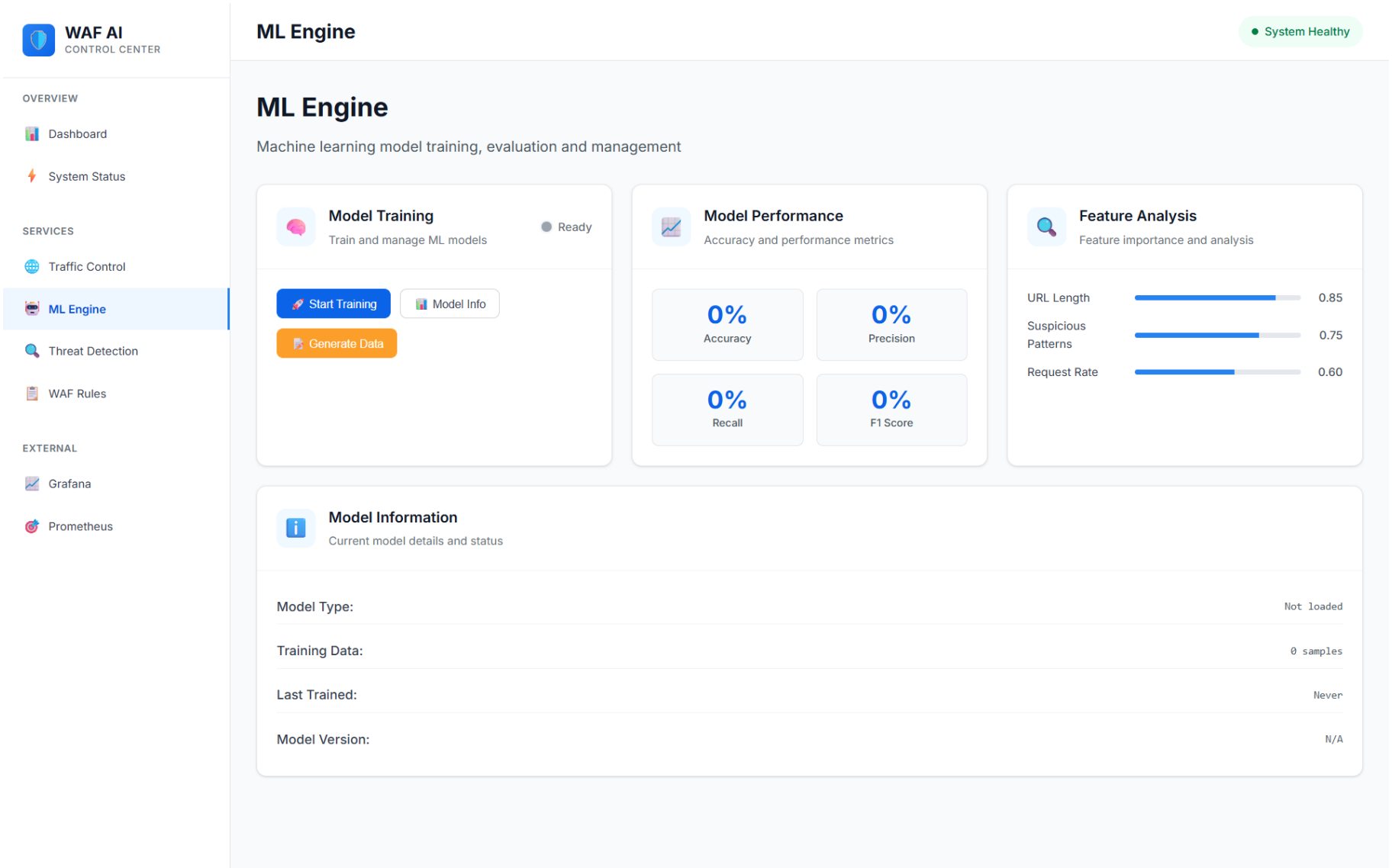
Output


docker desktop PERSONAL
Search Ctrl+K
Ask Gordon BETA
Containers
Images
Volumes
Kubernetes
Builds
Models
MCP Toolkit BETA
Docker Hub
Docker Scout
Extensions
Manage
Grafana
ngrok
SQL containers
VS Code Web
Containers Give feedback
Container CPU usage
7.82% / 2000% (20 CPUs available)
Container memory usage
643.04MB / 7.39GB
Show charts
Search
Only show running containers
Name Container ID Image Port(s) CPU (%) Last started Actions
cyberguard-ai - - - 7.82% 32 minutes ago
nginx-node-2 c8baddb30f45 nginx:alpine 8082:80 0.12% 32 minutes ago
nginx-node-1 43e18ed00a98 nginx:alpine 8081:80 0.09% 32 minutes ago
promtail-node-1 6e699ed8b920 grafana/promtail:latest 1.28% 1 hour ago
promtail-node-2 a5333b0c7e6e grafana/promtail:latest 1.53% 1 hour ago
waf-grafana 669cb2fcc53f grafana/grafana:latest 3080:3000 0.32% 1 hour ago
waf-loki 9fca145f2e3f grafana/loki:latest 3100:3100 1.88% 1 hour ago
waf-redis d1d0c84bef0d redis:7-alpine 6379:6379 0.7% 1 hour ago
waf-prometheus 7ff69d35381f prom/prometheus:latest 9090:9090 0% 1 hour ago
waf-api fd413bf5a600 cyberguard-ai-waf-api 8000:8000 0.17% 60 minutes ago
log-server-2 c7979f794413 cyberguard-ai-log-server 8083:8080 0.22% 60 minutes ago
log-server-1 598e55c0d475 cyberguard-ai-log-server 8080:8080 0.2% 60 minutes ago
traffic-generator 7d266dcfd27a cyberguard-ai-traffic-ger 1.31% 60 minutes ago
Showing 14 items
Engine running
RAM 4.79 GB CPU 4.01% Disk: 11.14 GB used (limit 1006.85 GB)
Terminal
Update version
Search
01:31 AM
19-12-2025

Output


Grafana
Home
Bookmarks
Bookmark pages for them to appear here
Starred
Your starred dashboards will appear here
Dashboards
Playlists
Snapshots
Library panels
Shared dashboards
Explore
Drilldown
Metrics
Logs
Traces
Profiles
Alerting
Alert rules
Contact points
Notification policies
Silences
Active notifications
Recently deleted
Settings
Connections
Add new connection
Data sources
Administration
Provisioning
General
Home › Dashboards › Master WAF & Nginx Security Dashboard
Search... ctrl+k
Edit
Export
Share
Last 1 hour
Refresh 30s
WAF System Overview
Registered Nodes
2
Threats/min
2.88 K
Active WAF Rules
2
ML Engine Status
UP
Avg Processing Time
No data
Services Up
1
Performance & Processing Metrics
Processing Time Percentiles
50 ms
40 ms
30 ms
20 ms
10 ms
22:20 22:25 22:30 22:35 22:40 22:45 22:50 22:55 23:00 23:05 23:10 23:15
50th percentile
50th percentile
95th percentile
95th percentile
99th percentile
99th percentile
Request Processing Rate
40 req/s
30 req/s
20 req/s
10 req/s
22:20 22:25 22:30 22:35 22:40 22:45 22:50 22:55 23:00 23:05 23:10 23:15
Total Requests/sec
Error Requests/sec
Blocked Requests/sec
Nginx Access Logs & Traffic Analysis
Total Requests/sec
4.70 req/s
Error Rate/sec
2.25 req/s
Active Log Streams
50
Blocked Requests/sec
0.130 req/s
Request Rate by Node
Response Codes Distribution

Output


WAF AI
CONTROL CENTER
OVERVIEW
Dashboard
System Status
SERVICES
Traffic Control
ML Engine
Threat Detection
WAF Rules
EXTERNAL
Grafana
Prometheus
ML Engine
System Healthy
ML Engine
Machine learning model training, evaluation and management
Model Training
Train and manage ML models
Ready
Start Training
Model Info
Generate Data
Model Performance
Accuracy and performance metrics
0%
Accuracy
0%
Precision
0%
Recall
0%
F1 Score
Feature Analysis
Feature importance and analysis
URL Length
0.85
Suspicious Patterns
0.75
Request Rate
0.60
Model Information
Current model details and status
Model Type:
Not loaded
Training Data:
0 samples
Last Trained:
Never
Model Version:
N/A

Test Cases

S.No	Module	Test Case	Input	Expected Output	Actual Output	Status
1	Authentication	Valid login	Admin username & password	Returns JWT token	JWT token returned, e.g., eyJhbGciOiJIUzI1NiIsInR5cCI...	Pass
2	Authentication	Invalid login	Wrong credentials	401 Unauthorized	401 Unauthorized	Pass
3	User Management	Add new user	Admin token + user data	201 Created	201 Created	Pass
4	User Management	List users	Admin token	List of users	[{"id":1,"name":"John"},{"id":2,"name":"Alice"}]	Pass
5	ML Engine	Train model	Sample training JSON	Training completes, model saved	Training completed successfully, model saved to models/	Pass
6	ML Engine	Predict threat	Test HTTP request data	Correct threat label	Threat label: SQL Injection	Pass
7	ML Engine	Incremental training	New threat data	Model updated	Model updated with new threat data	Pass
8	Traffic Collector	Start collection	Node URLs	Collector starts, logs traffic	Collector started on nodes: [node1, node2]	Pass
9	Traffic Collector	Stop collection	Active nodes	Collection stops	Collection stopped for all nodes	Pass
10	Traffic Stats	Fetch stats	Node ID	Traffic metrics JSON	{"requests":1200,"blocked":35,"anomalies":5}	Pass
11	WAF Rules	Generate rules	Threat prediction data	Nginx rules file generated	nginx_rules.conf generated	Pass
12	WAF Rules	Deploy rules	Node IDs + rules	Rules deployed to nodes	Rules deployed successfully to nodes [node1,node2]	Pass

Test Cases

13	WAF Rules	Rollback rules	Node IDs	Rules reverted	Rules rolled back successfully	Pass
14	Nginx Node	Add node	Node JSON	Node added, status OK	Node added: node3, status OK	Pass
15	Nginx Node	Node status	Node ID	Node connectivity and health	Node node1 is UP, latency 20ms	Pass
16	Control Panel	Service status	Access panel	Real-time status shown	Dashboard shows all services UP	Pass
17	Control Panel	Start/Stop service	Service name	Service starts/stops successfully	Service WAF started successfully	Pass
18	Monitoring	Grafana metrics	Access Grafana	Metrics displayed	Grafana dashboard displays CPU, memory, requests	Pass
19	Monitoring	Prometheus metrics	/metrics endpoint	Correct metrics output	{"http_requests_total":1200,"threats_detected":35}	Pass
20	Alerts	Slack/email alert	Threat detected	Alert sent	Alert sent to Slack/email for SQL Injection	Pass
21	Security	Emergency shutdown	Admin token	All services stop	All services stopped, nodes [node1,node2]	Pass
22	Security	Block IP range	CIDR + reason	IP blocked in WAF	IP range 192.168.1.0/24 blocked	Pass
23	API	Rate limiting	High request volume	Requests limited	429 Too Many Requests	Pass
24	API	Unauthorized access	No token	401 Unauthorized	401 Unauthorized	Pass
25	Logging	Access logs	HTTP requests	Logs stored and filtered	Logs stored in MongoDB, filtered by threat type	Pass

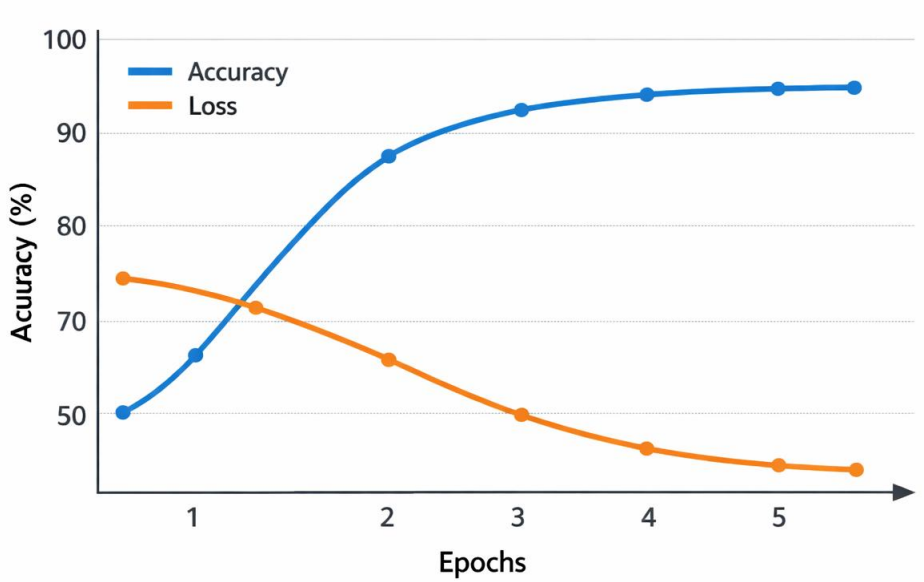
Results

Machine Learning Performance

- **Accuracy:** 90%
- **Precision:** 93%
- **Recall:** 92%

Inference

- Fast detection
- Reliable predictions
- Low false positives
- Real-time analysis
- Improved system security



Conclusion

- The **AI-Powered Advanced Web Application Firewall** successfully demonstrates the use of machine learning to monitor and analyze web traffic in real time.
- The system intelligently detects malicious requests, blocks potential threats, and dynamically generates security rules.
- By combining automation with AI, the firewall reduces manual intervention, improves detection accuracy, and enhances overall web application security.
- This project proves that AI-based WAF solutions are effective, scalable, and suitable for modern web environments.

Future Work

- In future, the system can be enhanced by training the machine learning model with larger and more diverse real-world traffic datasets to improve detection accuracy.
- Advanced deep learning techniques can be integrated to identify zero-day and evolving cyber attacks more effectively.
- The firewall can be extended to support cloud-native and distributed environments with auto-scaling capabilities.
- Integration with threat intelligence feeds and SIEM tools can further strengthen real-time threat response and security analytics.

References

1. A. Patcha and J. M. Park,
   An Overview of Anomaly Detection Techniques, Computer Networks, 2007.
2. N. Moustafa and J. Slay,
   UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection, IEEE, 2015.
3. Z. Alom et al.,
   Intrusion Detection Systems Using Deep Learning Techniques, IEEE Access, 2019.
4. Project GitHub Repository:
   https://github.com/Darkwebnew/AI-Powered-Advanced-Web-Application-Firewall.git
5. **Project Demo Video**:
   Demonstration of AI-Powered Advanced Web Application Firewall (Self-Recorded)

Questions

Thank You