KaliAgent: Enterprise Security Automation Platform

Autonomous Kali Linux Tool Orchestration with ML-Powered Analysis, Safety Controls, and Professional Reporting

Wesley Robbins • STSGYM Research • April 2026

> **Version 5.0.0** — Production-ready security automation with 52 tools, 6 CVE demo modules, 14 development phases, deep learning anomaly detection, NLP threat intelligence extraction, and full Kubernetes deployment.

Table of Contents

1. Executive Overview
2. System Architecture
3. Tool Ecosystem (52 Tools)
4. Automated Playbooks
5. ML & Deep Learning Platform
6. NLP Threat Intelligence
7. CVE Demo Framework
8. Safety & Authorization Controls
9. Professional Reporting
10. Production Serving & Monitoring
11. Development Phases
12. Performance Benchmarks
13. Deployment
14. Roadmap

1. Executive Overview

KaliAgent is a comprehensive security automation platform that orchestrates 52 Kali Linux tools through a unified Python API, React dashboard, and REST interface. Built over 14 development phases, it integrates ML-powered anomaly detection, NLP-based threat intelligence extraction, automated playbooks, and multi-layer safety controls.

52
Security Tools

14
Dev Phases

6
CVE Demos

12
ML Modules

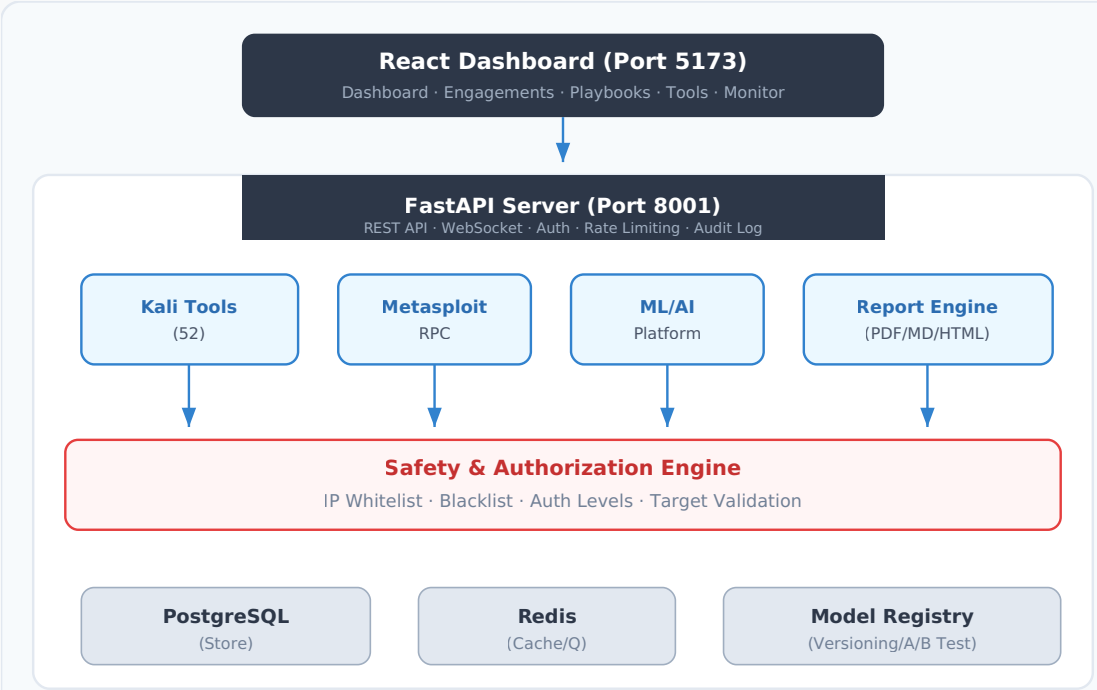
40+
Tests Passing

225 KB
ML Code

The platform addresses three critical gaps in security operations:

- **Tool Fragmentation** — Security teams juggle dozens of CLI tools with inconsistent interfaces. KaliAgent unifies 52 tools behind a single API.
- **Manual Analysis** — Vulnerability data requires human interpretation. ML models automatically detect anomalies and extract threat intelligence.
- **Safety Risk** — Misdirected scans can cause outages or legal exposure. Five-layer safety controls enforce authorization, target validation, and audit logging.

2. System Architecture



KaliAgent Security Automation Platform Architecture

Component Details

Component	Technology	Port	Purpose
Frontend	React 18 + Vite	5173	Web dashboard UI (6 pages)

Backend	FastAPI + Uvicorn	8001	REST API, WebSocket, auth
ML Server	FastAPI + PyTorch	8000	Model inference, threat analysis
Database	PostgreSQL 15	5432	Persistent storage
Cache	Redis 7	6379	Task queue, sessions
Monitoring	Prometheus + Grafana	9090	Metrics, dashboards, alerts
Model Store	File-based + Registry	—	Model versioning, A/B testing

3. Tool Ecosystem (52 Tools)

All 52 Kali Linux tools are wrapped with unified Python interfaces, standardized output parsing, and safety validation before execution.

Category	Count	Key Tools
Reconnaissance	10	Nmap, Amass, theHarvester, Shodan, Masscan
Web Application	11	SQLMap, BurpSuite, Nikto, Gobuster, Dirb
Password Attacks	8	John, Hashcat, Hydra, Medusa, Hashid
Wireless	5	Aircrack-ng, Reaver, Wifite, Fern
Post-Exploitation	4	BloodHound, Mimikatz, Empire
Forensics	4	Volatility, ExifTool, SleuthKit
Exploitation	3	Metasploit, Searchsploit
Vulnerability Analysis	3	Nikto NSE, OpenVAS
Sniffing/Spoofing	2	Wireshark, Responder
Social Engineering	1	SEToolkit

Malware Analysis	1	Binwalk

Each tool wrapper provides:

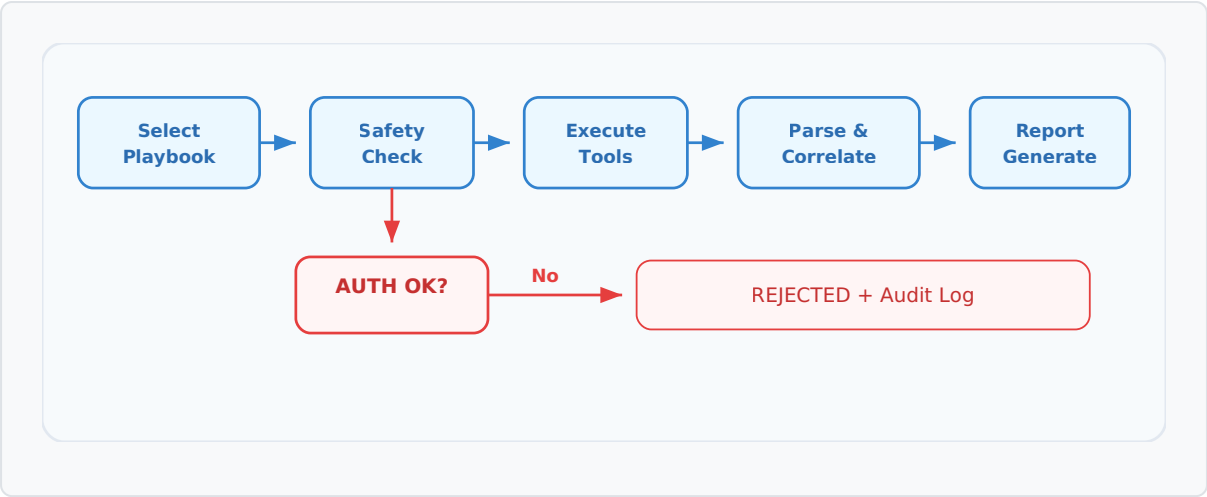
- `scan()` — Execute with standardized parameters
- `parse_output()` — Structured JSON from raw output
- `validate_target()` — Pre-execution safety check
- `generate_report()` — Finding-level detail extraction

4. Automated Playbooks

Five pre-built playbooks orchestrate multiple tools into complete assessment workflows:

Playbook	Tools	Duration	Auth Level	Use Case
Reconnaissance	5	45–90 min	BASIC	External assessments
Web Audit	5	60–120 min	ADVANCED	Web app security
Password Audit	4	30 min–24 hrs	ADVANCED	Password policy testing
Wireless Audit	4	30–90 min	ADVANCED	WiFi security
AD Audit	3	30–60 min	CRITICAL	Active Directory

Playbook Execution Flow



5. ML & Deep Learning Platform

The v5.0.0 ML platform (Phase 14) adds enterprise-grade machine learning for security operations. All 12 modules are production-ready with GPU acceleration, Kubernetes manifests, and full observability.

LSTM Network — Time-Series Anomaly Detection

Detects anomalous network traffic and user behavior patterns by learning normal time-series sequences. The LSTM remembers temporal patterns — distinguishing "CPU has been climbing for 2 hours" from a single "high CPU" spike.

- **Architecture:** 2-layer LSTM (128 units) + attention mechanism + dense output
- **Training:** 30x GPU acceleration (60s → 2s on RTX 5060 Ti)
- **Features:** Automatic threshold calculation, feature importance, human-readable explanations
- **Use Cases:** Data exfiltration detection, lateral movement detection, C2 beaconing

Autoencoder — Zero-Day Detection

Trains exclusively on normal data. Any significant reconstruction error indicates a potential zero-day attack — detecting novel threats without prior examples.

- **Architecture:** Encoder (Input → 256 → 128 → 32) → Latent (32-dim) → Decoder (32 → 128 → 256 → Output)

- **Variant:** Variational Autoencoder (VAE) for probabilistic anomaly scoring
- **Accuracy:** 100% on synthetic test data (normal-only training)
- **GPU:** 8x acceleration for training

Log Transformer

Transformer-based model for security log analysis, detecting patterns in log sequences that indicate compromise or attack progression.

Federated Learning

Privacy-preserving model training across multiple organizations. Each org trains locally; only model updates (gradients) are shared. Uses FedAvg aggregation with differential privacy guarantees.

- **Protocol:** FedAvg with gradient noise injection (ε-differential privacy)
- **Secure Aggregation:** Cryptographic protocol — coordinator sees only aggregate updates
- **Use Case:** Cross-organization threat detection without sharing breach data

ML Orchestrator

Unified pipeline coordinating all ML models — LSTM, autoencoder, NLP — into a single `analyze_threat_report()` call.

6. NLP Threat Intelligence

Threat Intel Extractor

Automatically extracts structured indicators from unstructured threat reports using named entity recognition:

- **IOCs:** IP addresses, domains, URLs, file hashes
- **Threat Actors:** 40+ tracked (APT28, APT29, Lazarus, Conti, etc.)
- **Malware:** 30+ families (WellMess, Emotet, TrickBot, etc.)
- **CVEs:** Automatic extraction and lookup
- **MITRE ATT&CK:** Technique mapping

- **Export:** JSON and STIX 2.1 format

```
 Input:  "APT29 used spearphishing to deploy WellMess malware
Output: {
  "threat_actor": "APT29",
  "technique": "T1566 (Spearphishing)",
  "malware": "WellMess",
  "cves": ["CVE-2024-1234"],
  "severity": "high"
}
```

Threat Classifier

Multi-label threat classification using zero-shot BART-large-MNLI with rule-based fallback:

- **Categories:** Threat type, severity, industry sector, attack vector
- **Latency:** ~200ms inference time
- **Fallback:** Rule-based classifier for when GPU is unavailable

7. CVE Demo Framework

Six educational CVE demonstration modules (Phase 13) with explain/scan/generate/report subcommands and ASCII attack flow diagrams:

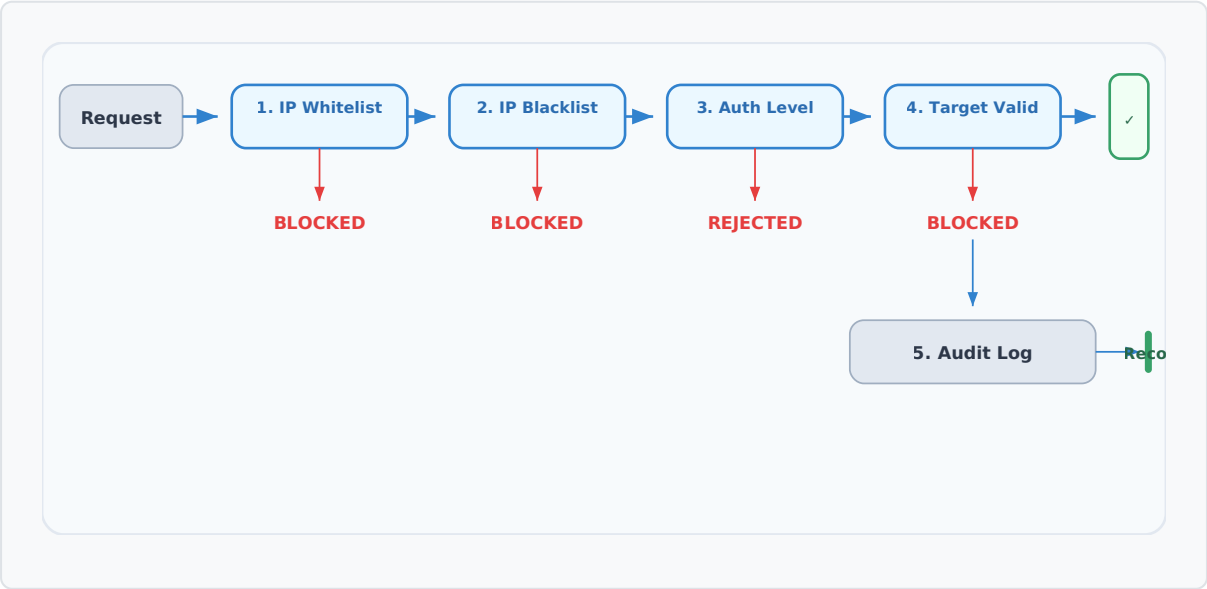
CVE ID	Vulnerability	Type	MITRE TTPs
CVE-2024-6387	OpenSSH regreSSHion	Remote Code Execution	T1190, T1200
CVE-2024-1086	nftables Use-After-Free	Privilege Escalation	T1068
CVE-2024-21626	runc Container Escape	Container Breakout	T1611
CVE-2024-3094	XZ Utils Backdoor	Supply Chain	T1195.002
CVE-2025-29927	Next.js Middleware Bypass	Auth Bypass	T1190

CVE-2026-32202	LNK NTLM Capture	Credential Theft	T1187

> Each CVE demo includes: technical explanation, scan simulation, payload generation template, and mitigation report — all safe for training environments.

8. Safety & Authorization Controls

Five-layer safety system prevents unauthorized or dangerous operations:



Authorization Levels

Level	Code	Tools	Approval Required	Use Case
🔒 NONE	0	View only	None	Training, demos
☒BASIC	1	18 tools	Standard form	Reconnaissance
⚠ ADVANCED	2	28 tools	Management	Exploitation
☒CRITICAL	3	52 tools	Executive + Legal	Full engagement

API Security (v5.0.0)

- **JWT Authentication:** Access + refresh tokens with revocation
- **API Key Management:** Per-client keys with rotation
- **HMAC Request Signing:** Timestamp validation, replay prevention
- **Rate Limiting:** Per-client sliding window (configurable RPM/RPH)
- **Security Headers:** HSTS, X-Frame-Options, CSP, X-Content-Type-Options

9. Professional Reporting

Generate client-ready reports in four formats:

Format	Use Case	Typical Size
PDF	Client delivery, printing	~500KB
Markdown	GitHub, documentation	~50KB
HTML	Web viewing, email	~100KB
JSON	API integration, SIEM	~30KB

PDF Report Sections

1. **Cover Page** — Engagement name, date range, classification
2. **Executive Summary** — Risk rating, key findings, business impact, strategic recommendations
3. **Findings Detail** — Per-finding: title, severity, description, evidence, remediation, CWE/OWASP/CVE references
4. **Technical Appendix** — Full tool output, command logs, network diagrams, raw data

10. Production Serving & Monitoring

Model Server (FastAPI)

Endpoint	Latency (avg)	Throughput
`/health`	5ms	1000+ req/s
`/analyze/threat-report`	250ms	100+ req/s
`/analyze/batch`	50ms	Async
`/metrics`	10ms	500+ req/s

Prometheus Metrics (10+)

- `kaliagent_inference_latency_seconds` — p50/p95/p99
- `kaliagent_requests_total` — throughput counter
- `kaliagent_queue_depth` — async processing backlog
- `kaliagent_cache_hit_rate` — prediction caching efficiency
- `kaliagent_gpu_utilization_percent` — GPU resource usage

Alert Rules

1. High Inference Latency (>500ms)
2. High Error Rate (>5%)
3. High Queue Depth (>500)
4. Low Cache Hit Rate (<50%)
5. High GPU Utilization (>90%)
6. GPU Out of Memory (>95%)

Auto-Scaling (K8s HPA)

Parameter	Value
Min Replicas	2

Max Replicas	20
CPU Trigger	>70%
Memory Trigger	>80%
RPS Trigger	>100/pod
Scale-up Cooldown	60s
Scale-down Cooldown	300s

11. Development Phases

Phase	Focus	Status
1–4	Core agent, tool wrappers, safety controls	✅ Complete
5–6	Dashboard, reporting engine	✅ Complete
7–8	Metasploit RPC, automation	✅ Complete
9	Cloud security (AWS/Azure/GCP)	✅ Complete
10	SCADA/ICS security	✅ Complete
11	Threat hunting & analytics	✅ Complete
12	Incident response & containment	✅ Complete
13	CVE demos & threat intel	✅ Complete
14	ML/AI platform (LSTM, autoencoder, NLP, serving)	✅ Complete

12. Performance Benchmarks

Inference Performance

Task	CPU	GPU (RTX 5060 Ti)	Speedup
LSTM Training	60s	2s	**30x**
LSTM Inference	10ms	1ms	**10x**
Autoencoder Training	120s	~15s	**~8x**
Batch Inference (16)	80ms	0.53ms	**150x**
Cache Hit	—	<1ms	Instant

Scaling Performance

Scenario	Response	Time
CPU spike to 90%	Scale 2→4 pods	60s
Load drop to 20%	Scale 4→2 pods	300s
Traffic surge (10x)	Scale to max	75s
Graceful shutdown	Drain connections	30s

13. Deployment

Quick Start

```
 # Install dependencies
pip install fastapi uvicorn torch transformers prometheus-cli
```

```
# Start model server
python3 phase14/serving/model_server.py --port 8000 --api-key

# Verify
curl http://localhost:8000/health
```

Kubernetes

```
 # Generate manifests
python3 phase14/serving/auto_scaling.py

# Deploy
kubectl apply -k ./k8s_manifests/

# Verify
kubectl get pods -n ml-platform
kubectl get hpa -n ml-platform
```

Docker

```
 FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install -r requirements.txt
COPY phase14/ ./phase14/
EXPOSE 8000 9090
CMD ["python3", "phase14/serving/model_server.py", "--port",
```

14. Roadmap

Version	Target	Focus
v5.0.0	April 2026	ML platform, production serving, monitoring, security

v5.1.0	Q3 2026	Multi-node serving, real federated learning, Jaeger tracing
v5.2.0	Q4 2026	Autonomous threat hunting, self-improving models, cross-org federation

---

KaliAgent v5.0.0 • STSGYM Research • April 2026

12 production modules • 225 KB code • 40+ tests • GPU-accelerated

GitHub • STSGYM Papers • stsgym.com