ForensIQ Copilot

AI-Driven Log Investigation & Forensic Intelligence Platform

THE SMURFS|VIT VELLORE |CSIC 1.0

Problem Statement

Core Insight: *Automated, explicable investigation and attack reconstruction—rather than detection—remain the unresolved problem.*

The Real-World Cybersecurity Challenge

Large amounts of security logs are produced by modern organizations from:

- Endpoints (Linux/Windows)
- Applications and servers
- Network devices (proxies, firewalls)
- Cloud platforms (storage, API, and IAM)

The investigation phase is still essentially flawed, despite the maturity of log collection and alerting.

When Present System Fail

1. Alert-Centric: Manual correlations are made between isolated alerts from various attack stages.
2. Fragmented Correlation: Static rules and silos eliminate cross-entity and temporal context.
3. Incapacity to Explain: ML systems' incapacity to defend risk scores erodes analyst confidence.
4. No Attack Reconstruction: Events are displayed as isolated incidents rather than attacker timelines.
5. Weak Forensic Integrity: Evidence is unreliable when logs are altered or overwritten.

Proposed Solution - Conceptual Overview

What are we Building

The **ForensIQ Copilot** is a forensic intelligence and log investigation platform powered by **AI** that converts unprocessed security logs into:

- Related incidents
- Attack routes rebuilt
- MITRE ATT&CK-aligned methods
- Explanations supported by evidence
- Reports prepared for investigation
- Fine tuned LLM chatbot

| From Raw Logs To Forensic-gradeattack narratives.

Key Shift in Thinking

Proposed Solution & Technical Solution

ForensIQ Copilot

AI-Driven Log Investigation Engine

- ➔ Unified log ingestion → normalized schema
- ➔ Hybrid anomaly detection (statistical + behavioral)
- ➔ Graph-based entity correlation
- ➔ MITRE ATT&CK technique mapping
- ➔ Tamper-proof forensic evidence storage

Investigation Intelligence

- **Explainable AI** for analyst trust
- **Reconstructed attack paths** & timelines
- **Investigation** ready reports.

Investigation-first system that operates **after alerts,**
integrates with existing tools, and produces **attack narratives** - **not scores.**

System Architecture

Overview & Core Processing

- Architecture Overview
  - Scalable forensic analysis made possible by a modular, layered architecture
- Log Ingestion Layer:
  - Purpose: Gather logs from various sources
  - Tech: FastAPI, REST APIs
  - Reason: Scalable and lightweight
- Parsing & Normalization:
  - Purpose: Create a unified schema from logs
  - Tech: Python, Regex, JSON validation
- Feature Engineering:
  - Purpose: Extract textual and numerical characteristics
  - Tech: Pandas, NumPy, scikit-learn
- Detection Engine:
  - Hybrid detection using Rules, Isolation Forest, LOF, N-grams

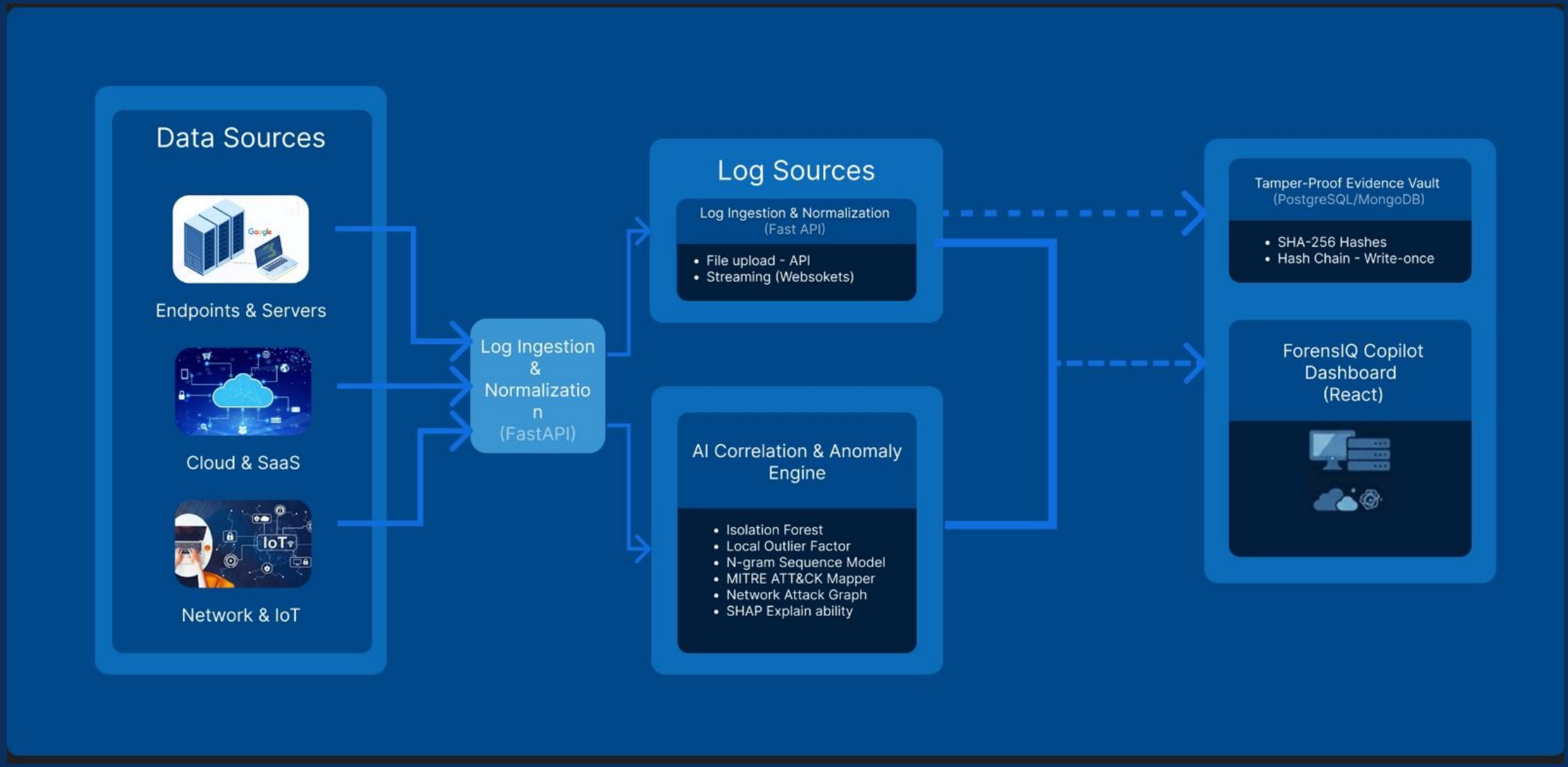
System Architecture

Intelligence, Interface & Forensics

- Correlation Engine:
  - Purpose: Connect alerts to potential attacks
  - Tech: NetworkX
- LLM Explanation Layer:
  - Purpose: Give investigators an explanation of alerts
  - Fine-tuned LLaMA model
- Dashboard:
  - Purpose: Analyst-facing visualization
  - Tech: React, Chart.js, D3.js
- Forensic Evidence Vault:
  - Purpose: Maintain the integrity of the evidence
  - Tech: PostgreSQL, SHA-256 hash chaining

- Design Principle:
  - Modular, pluggable, independently scalable components
- Threat Intelligence Enrichment:
  - AbuseIPDB for malicious IP reputation lookup
  - Used during detection and correlation to validate suspicious IPs

System Architecture



Log Ingestion & Normalization Layer

Ingestion & Processing

- Supports structured and semi-structured logs.
- Handles heterogeneous sources (OS, app, network, cloud)
- Parses and normalizes logs into a **common event schema**

"Without normalization, correlation is mathematically meaningless".

Normalized Event Schema

Core extracted fields:

- `timestamp`
- `host`
- `user`
- `service`
- `template_id`
- `parameters`

Used consistently across all downstream analysis

Threat Enrichment

- **IP reputation**
- Geo-location context
- Known abuse and threat intelligence sources
- Adds **external context** before correlation and inference.

AI Correlation & Inference Engine

Detection → Understanding → Decision *(No black boxes.)*

Anomaly Ensemble (Detection Layer)

Flag events that *deserve attention.*

Input: Parsed & normalized logs

Models (classical ML):

- Isolation Forest → rare behavior detection
- LOF → local deviation from peer behavior
- N-gram sequence model → command / action sequence breaks

Output (per event):

- Anomaly score
- Feature-level reason (what deviated)

Statistical anomaly detection

Correlation Engine (Understanding Layer)

Merge anomalies into one incident story

a) Deterministic correlation logic

- Same host, user, or IP.
- Same process or service.
- Same time window (for example, ≤ 5 minutes).
- Same MITRE ATT&CK phase.

b) Threat intelligence enrichment

- External IPs checked against AbuseIPDB.
- Reputation score used as contextual signal, not sole decision factor.

c) Graph construction

- Nodes: host, user, process, IP.
- Edges: login → exec → lateral movement → exfiltration.

Logical correlation — zero guessing

Inference Scoring (Decision Layer)

Assess severity *after* full context is known.

Signals combined:

- Count & strength of anomalies
- MITRE technique severity
- Kill-chain progression depth
- Cross-host / cross-user movement
- AbuseIPDB score.

Outputs:

- Incident risk score
- Confidence level
- Evidence list

Explainable scoring

Proposed Solution & Technical Solution

Models Used & What they Capture

Isolation Forest
Detects global statistical outliers

→ **Volume anomalies**
Sudden spikes or rare events

Local Outlier Factor (LOF)
Detects peer-group deviations

→ **Contextual anomalies**
Behavior deviating from similar entities

Sequence / N - gram models
Detect abnormal event orderings

→ **Behavior anomalies**
Unusual execution or access sequences

Why Hybrid Matters

- **Attacks** are subtle
- No single model captures:
  - **Volume**
  - **Context**
  - **Behavior**

Attack Correlation & Graph Reasoning

Graph Construction

- ★ **Build dynamic** graph from logs
- ★ **Nodes:** users , hosts, IPs, processes
- ★ **Edges:** authentication, execution, network flows
- ★ **Time aware, graph updates**

Correlation & Reasoning

- ★ **Sling-time window** correlation
- ★ **Cross-log** entity linking
- ★ Context propagation across hops
- ★ Noise reduction through relationship strength

System Output

- ★ **Reconstructed attack paths**
- ★ **Multi-step intrusion narratives**
- ★ Clear start ⟶ lateral movement ⟶ impact

MITRE-ATT&CK Technique and Mapping

Mapping Logic

- Correlated attacks behavior mapped to ATT&CK techniques
- Mapping operates on post - correlation events
- Outputs standardized technique IDs (Txxx)
- Supports tactic - level and technique-level attribution

Confidence & Validation

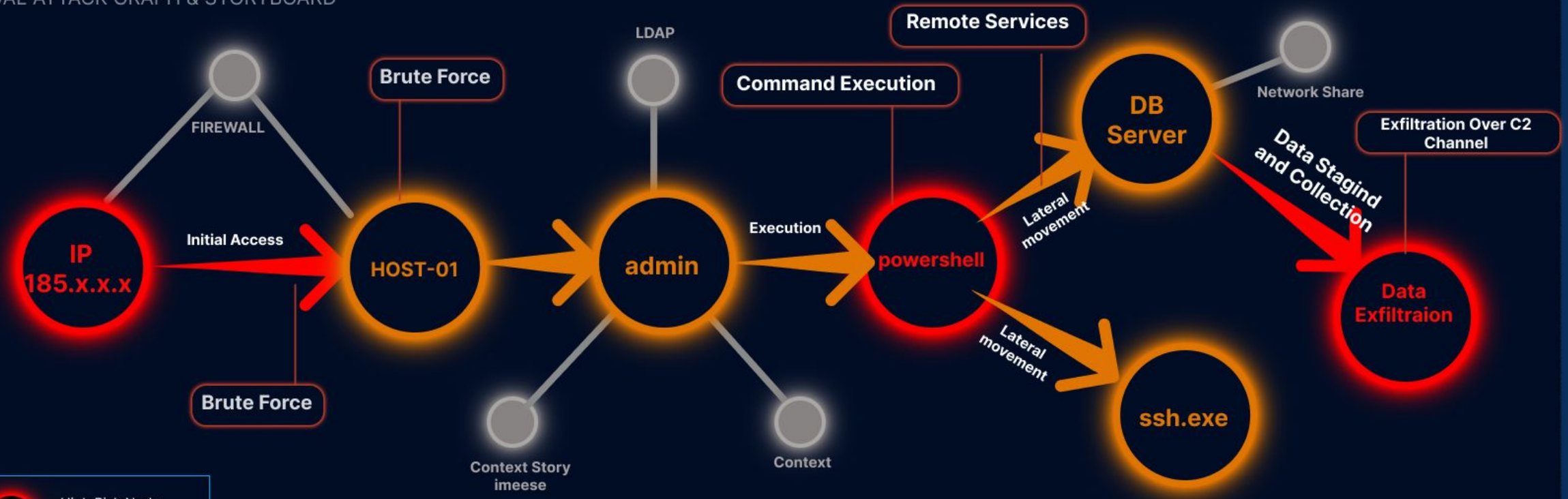
- **Confidence increases when:**
  - **ML anomaly detection** triggers
  - Rule - based ATT&CK pattern matches
- Hybrid scoring reduces false positives
- Explicit confidence attached to each technique

Why this matters

- **Industry - standard** threat classification
- **Improved analyst trust** and explanaibility
- **Audit** - and compliance **ready** investigation reports

FORENSIQ COPILOT

VISUAL ATTACK GRAPH & STORYBOARD



Attack Storyboard

14:02:15	14:05:45	14:10:30	14:12:50	14:15:20	14:20:10	14:25:05
Initial Access	T1110 Brute Force	admin compromised	T1059 Powershell executed	T1021 Lateral Movement to DB-Server	Data Collected	T1041 Data Exfiltration

Explainability and Analyst Copilot

Human-in-the-loop reasoning — not automated decision-making

Analyst Queries
(NL → Structured Retrieval)

Analyst questions converted into structured graph traversals + SQL queries

Query execution spans:

- Incident graph (entities, edges, timelines)
- Anomaly metadata (scores + features)
- MITRE ATT&CK technique mappings

LLM is a query interface, not a reasoning engine.

Evidence-Backed Explanations
(Anti-Blackbox)

- Model-level evidence exposed:
  - Isolation Forest → score deviation
  - LOF → density shift
  - N-gram → rare sequence
- Normal vs observed behavior shown
- Every explanation tied to:
  time • entity • technique • model signal

No text without evidence

Incident Summaries & Timelines (LLM Scoped Use)

Summaries generated from:

- Graph structure
- Event order
- MITRE kill-chain phase

Outputs:

- SOC summary
- Forensic timeline
- PDF report

No detection or scoring by LLM

LLM explains decisions — it does not make them

LLM Fine-Tuning

Base model: LLaMA 2–3B (small, controllable, on-prem deployable)
Fine-tuning method: LoRA (parameter-efficient, low-risk adaptation)

Analyst Queries

- 1,000+ analyst-style cyber incident Q&A pairs.
- Data from sanitized SOC notes, MITRE ATT&CK techniques, and graph-based incident timelines.
- Each sample includes:
  - Analyst question.
  - Retrieved evidence (events, entities, scores).
  - Ground-truth explanation linked to evidence IDs.
- Key innovation: evidence-structured training data instead of free-text

Training Objective

Teach the LLM to:

- Reference only provided evidence
- Use correct security terminology
- Explain *why* a risk score exists, not invent causes

Penalize outputs that:

- Introduce unseen facts
- Speculate beyond evidence
- Use vague security language

Training Objective

Teach the LLM to:

- Reference only provided evidence
- Use correct security terminology
- Explain *why* a risk score exists, not invent causes

Penalize outputs that:

- Introduce unseen facts
- Speculate beyond evidence
- Use vague security languages

The LLM is trained to explain evidence, not to invent

Tamper-proof Forensic Evidence Vault

Technical Safeguards

- Hash-chained log storage
- Write-once, immutable records
- Cryptographic integrity verification

Legal & Compliance Impact

- **Provides legal defensibility**
- **Meets regulatory compliance requirements**
- **Ensures post-incident forensic reliability**

Why Tamper-Proof Design Matters

- **Log evidence remains verifiably intact and tamper-proof**
- **Provides admissible records for court and auditors**
- **Builds a defensible timeline of secure events**

Layer of assurance for incident investigation and audit readiness

Expected Outcomes

Measured Impact

- Investigation time reduced from hours to minutes
- Reduced analytic fatigue
- Clear attack timelines
- Forensically defensible reports

Key metric

Mean Time to Investigation (MTTI)

Benchmarking Against Existing Solutions

Traditional SIEM / SOAR / Splunk like Platforms

- Centralized log collection and storage
- Rule-based alerting and dashboards
- Basic MITRE tagging and risk scores
- Reactive incident notifications

What ForensIQ Copilot Adds?

- Cross-log correlation across systems and users
- Automated attack reconstruction and timelines
- Explainable AI-driven anomaly reasoning
- Forensic-grade evidence integrity and reporting

Detailed Feature Comparisons

Feature	Legacy SIEM	SOAR	ForensIQ Copilot
Log Storage	High	Low	**Optimized Forensic Lake**
Detection Style	Rule-Based	Workflow-Based	**AI Anomaly + Correlation**
Root Cause Analysis	Manual	Scripted	**Autonomous Attack Graph**
NLP Support	None	Limited	**Full Native Copilot**
Investigation Speed	**Hours/Days**	**Mins/Hours**	**Seconds/Mins**

“From alerts and isolated events → to explainable attack narratives”

Differentiation and Unique Value Proposition

Traditional Security Tools

- Alerts and risk scores
- Isolated log events
- Manual correlation
- Query-heavy workflows
- Limited forensic explainability

ForensIQ Copilot

- Investigation-first design
- Hybrid ML + graph reasoning
- Explainable attack narratives
- Analyst-centric workflows
- Forensic-grade evidence handling

From alerts to attack stories —
explainable, evidence-backed, forensic-ready.

"Built for SOCs, MSSPs, and DFIR teams"

Expected Outcomes & Impact

Operational Outcomes

- Faster incident investigations
- Automated attack reconstruction
- Reduced analyst workload
- Clear forensic timelines
- Explainable AI-driven insights

Business & Risk Impact

- Reduced mean time to detect and respond
- Lower operational and breach costs
- Improved compliance and audit readiness
- Higher confidence in forensic decisions
- Better ROI on security operations

60-70% reduction in Investigation time
30-40% improvement in detection accuracy

Target End Use Cases

SOC Investigations: Rapid triage of security alerts using automated correlation, timelines, and anomaly explanations to reduce investigation time and analyst fatigue.

Post-Incident Forensics: Reconstruction of complete attack storylines after a breach, enabling investigators to identify root cause, affected assets, and attacker movement.

Insider Threat Detection: Detection of abnormal user behavior, privilege misuse, and policy violations through behavior-based analysis and cross-log correlation.

Compliance & Audit Readiness: Generation of tamper-proof, court-admissible forensic reports with verified timelines, evidence integrity, and chain of custody.

Product Roadmap

Phase 1: Core

- foundation for enterprise SOC teams and incident responders.
- log ingestion, parsing, and normalization.
- adding secure data aggregation and backend API services.
- docker-based setup for easy deployment.
- basic search and filtering features.

Phase 2: Forensics

- develop advanced intelligence and forensics tools for CERTs and digital forensic teams.
- tamper-proof forensic vault to store evidence securely.
- anomaly detection using isolation forest, LOF, and N gram model.
- correlation engine with MITRE ATT&CK mapping.

Phase 3: Insight

- visual intelligence
- graph-based attack visualizations and timeline reconstructions.
- interactive dashboard and frontend for analysts.

Phase 4: Automation

- automation and AI assistant for interaction
- Explainability engine and an LLM-powered investigative copilot.
- One-click forensic reporting and performance benchmarking.

THE SMURFS

Thank You

BHANU MAHESH CHEKURI

MIT KUMAR

JOHN POLY

RITWIK PANDEY

SHAUNAK DHAR