Autonomous Defense Transformers: Security-Native Reasoning for Digital Infrastructure

A Research Paper on AI-Driven Infrastructure Security

David Idris

Glemad, Inc.

AI research and product company developing the ADT model family since 2020

Lagos, Nigeria

david@glemad.com

ABSTRACT

Modern digital infrastructure is defended by systems that are fundamentally reactive. Telemetry is collected after actions occur, detections trigger after damage begins, and response is gated by human triage operating under time pressure. This architecture fails against AI-speed adversaries whose attack loops operate orders of magnitude faster than human decision cycles. We introduce Autonomous Defense Transformers (ADT), a security-native model class designed to reason continuously over live infrastructure state, interpret threats under uncertainty, validate defensive actions against explicit constraints, and generate audit-grade evidence as a first-class output. ADT is defined by five core design principles: defense-first pretraining, continuous model-level reasoning, integrated actuation under constraints, zero-trust alignment, and guardrailed learning. We present a complete system architecture separating context ingestion, threat interpretation, action validation, actuation, and audit trail generation. We provide a technical comparison with SIEM, SOAR, rule engines, and LLM-wrapper approaches, and define an evaluation framework focused on containment correctness, evidence completeness, and cost-weighted false positives. Deployment results from the PulseADT production system demonstrate 359x faster detection (0.8 min MTTD vs. 287 min industry average), 200x faster response (2.1 min MTTR vs. 420 min industry average), and 95% false positive reduction (1.2% vs. 23.5% industry average) across 680,000 protected assets. We conclude by discussing implications for enterprise resilience, regulatory enforcement, and national infrastructure security, with particular attention to African computing contexts.

Keywords: *autonomous defense, infrastructure security, transformer reasoning, compliance enforcement, safety gating, auditability, African cybersecurity*

INTRODUCTION

Digital infrastructure has evolved from relatively static systems into continuously changing environments composed of programmable identities, API-driven control planes, ephemeral workloads, automated deployment pipelines, and distributed configuration state (Rose et al., 2020; Syed et al., 2022). Security threats have evolved in parallel. Modern attackers increasingly operate through legitimate interfaces, automate reconnaissance and exploitation, and adapt behavior dynamically in response to partial containment (Sun et al., 2023).

Despite this shift, most deployed security systems remain architecturally human-paced. Detection pipelines may be automated, but interpretation, prioritization, and response are still gated by human analysts and brittle rule logic (Nguyen & Reddi, 2021). The resulting latency creates a structural mismatch between attacker speed and defender response. In practice, the most damaging modern incidents rarely hinge on the absence of

alerts. They hinge on the inability to interpret intent, verify constraints, and act quickly without causing operational harm (Sarker, 2023).

This paper argues that the failure is architectural rather than operational. Existing systems are designed to observe and alert, not to reason continuously or enforce invariants over infrastructure state (Repetto et al., 2021).

We introduce Autonomous Defense Transformers (ADT) as a security-native model class and system architecture intended to close this gap. ADT treats security and compliance as a continuous reasoning-and-control problem: build a persistent model of infrastructure state, update hypotheses about attacker intent under uncertainty, validate actions against explicit constraints, execute bounded enforcement, and generate audit-grade evidence as a first-class product of the control loop (Sewak et al., 2023).

Non-Goals

ADT is not designed to:

- replace organizational governance or legal accountability,
- prevent malicious insiders acting fully within granted authority,
- secure infrastructure in the absence of meaningful telemetry,
- eliminate all operational risk.

ADT addresses runtime infrastructure defense under uncertainty, not organizational failure modes.

CONTRIBUTIONS

This paper makes the following contributions:

1. **Problem reframing:** We formalize infrastructure security and compliance as continuous reasoning and control problems rather than alerting and documentation problems (Gafni & Levy, 2024).
2. **ADT definition:** We introduce Autonomous Defense Transformers as security-native models designed for threat interpretation, constraint-validated action, and auditability.
3. **System architecture:** We present an end-to-end architecture separating context ingestion, threat interpretation, action validation, actuation, and audit trail generation.
4. **Production validation:** We report deployment metrics from the PulseADT system protecting 680,000 assets, demonstrating 359x faster detection and 95% false positive reduction compared to industry averages.
5. **Threat model:** We define explicit adversary and defender assumptions aligned with modern control-plane and identity-driven attacks (MITRE, 2021).

6. **Action taxonomy and gating:** We formalize action classes and gating thresholds as a safety boundary for bounded autonomy.
7. **Evaluation plan:** We define an initial evaluation framework centered on containment correctness, evidence completeness, and cost-weighted false positives.
8. **Safety and governance:** We specify policy authority, update testing, separation of duties, and rollback guarantees.
9. **Limitations:** We document the dominant failure modes (telemetry gaps, context manipulation, action risk under uncertainty) and non-goals.
10. **African context:** We discuss the relevance of autonomous defense systems for African enterprises and critical infrastructure.

PROBLEM DEFINITION

Why Human-Led Security Fails at AI Speed

Human-led security fails because its control loop is bounded by attention rather than compute. Even when detections are accurate, response is delayed by alert queues, manual context reconstruction, coordination overhead, and approval workflows (Aminu et al., 2024). Modern attacks exploit this latency by operating entirely within the response gap. The result is not merely missed alerts, but failure to maintain invariants such as least privilege, approved network paths, and controlled access to sensitive resources.

If security is defined as maintaining invariants over infrastructure state, then a human-paced response loop becomes structurally insufficient. The correct framing is not how do we detect more, but how do we reason and enforce constraints continuously (Hammar & Stadler, 2020).

Limitations of Rule-Based Detection

Rule engines fail under modern conditions due to combinatorial behavior spaces, adversarial adaptation, fragmented context across tools, and surface-level semantics that do not capture intent (Dixit & Silakari, 2021). Rules encode known patterns. They do not reason about evolving system state, multi-step sequences, or the difference between legitimate automation and attacker-controlled automation using the same interfaces (Girdhar et al., 2023).

Compliance as a Reasoning Problem, Not Documentation

Compliance requirements encode behavioral constraints on systems. Encryption, least privilege, access review, and change control must hold continuously. Documentation alone cannot satisfy runtime enforcement. Compliance therefore becomes a reasoning problem: infer whether current infrastructure state

satisfies constraints, produce evidence at decision time, and ensure remediation actions are safe, authorized, and auditable (Formosa et al., 2021).

ADT DESIGN PRINCIPLES

Defense-First Pretraining

ADT models are trained to internalize infrastructure semantics, attacker tradecraft, and policy constraints as first-class concepts. The training objective is not only language fluency, but security-native structure: identities, roles, permissions, dependencies, change windows, and control-plane sequences (Maleki & Pourmoazemi, 2024).

Continuous Model-Level Reasoning

ADT maintains persistent state and updates threat hypotheses over time rather than reacting to isolated events. The core unit of analysis is the state transition and its relationship to policy constraints and historical baselines (Taye, 2023).

Integrated Actuation Under Constraints

ADT closes the loop from interpretation to action while validating every action against safety, policy, blast radius, and reversibility constraints. This is a central difference between AI-assisted security and autonomous defense (Kiely et al., 2023).

Zero-Trust Alignment

No input, retrieval, model output, or external tool output is trusted implicitly. Every decision is justified, every action is constraint-checked, and every enforcement step is recorded for audit (Bertino, 2021; Azad et al., 2024).

Guardrailed Learning

Learning and updates occur under governance. Model and policy changes are tested, staged, monitored, and rollbackable. Autonomous defense cannot depend on ungoverned drift (Tyagi & Seranmadevi, 2024).

MODEL AND SYSTEM ARCHITECTURE

ADT is designed as a closed-loop reasoning and enforcement system. It is not a monolithic model. It is a coordinated architecture with explicit boundaries between (i) observation and state construction, (ii)

hypothesis-based threat interpretation, (iii) action proposal, (iv) constraint-validated gating, (v) actuation, and (vi) audit evidence generation.

Transformer Backbone and Reasoning Kernel

The transformer serves as a reasoning kernel, not as a chat interface. It integrates heterogeneous context, reasons over sequences and dependencies, maintains competing hypotheses, and proposes candidate actions with justification (Vaswani et al., 2017). ADT does not require a novel transformer architecture. The novelty is the security-native embedding: persistent state, constraint validation, auditable outputs, and bounded actuation.

Context Ingestion and Unified State

The ingestion layer normalizes signals into canonical typed objects with preserved provenance. Inputs include (non-exhaustive): cloud audit logs, IAM policies and deltas, identity session logs, network flow summaries, workload metadata, CI/CD events, configuration drift signals, vulnerability context, and compliance policies with exceptions.

Outputs include:

- an entity graph (identities, resources, dependencies),
- event timelines (ordered state transitions),
- configuration snapshots (current state, deltas),
- policy objects (constraints, authority rules),
- provenance metadata (source, time, integrity).

Threat Interpretation Layer

Threat interpretation converts observed state and sequences into intent-level hypotheses under uncertainty. ADT maintains a threat belief state over hypotheses, confidence, and expected evidence for confirmation and refutation. Uncertainty is preserved, not collapsed into a single label, because aggressive action under false certainty is a primary safety failure mode in autonomous defense (Ding et al., 2023).

Action Validation and Audit Generation

Actions are proposed, not executed. Every action is validated against policy admissibility, safety constraints, reversibility, blast radius, and required confidence thresholds. Every decision produces an evidence bundle designed for independent review: what was observed, what was inferred, what constraints were checked, what action was taken, and what verification confirmed outcomes (Noel et al., 2016).

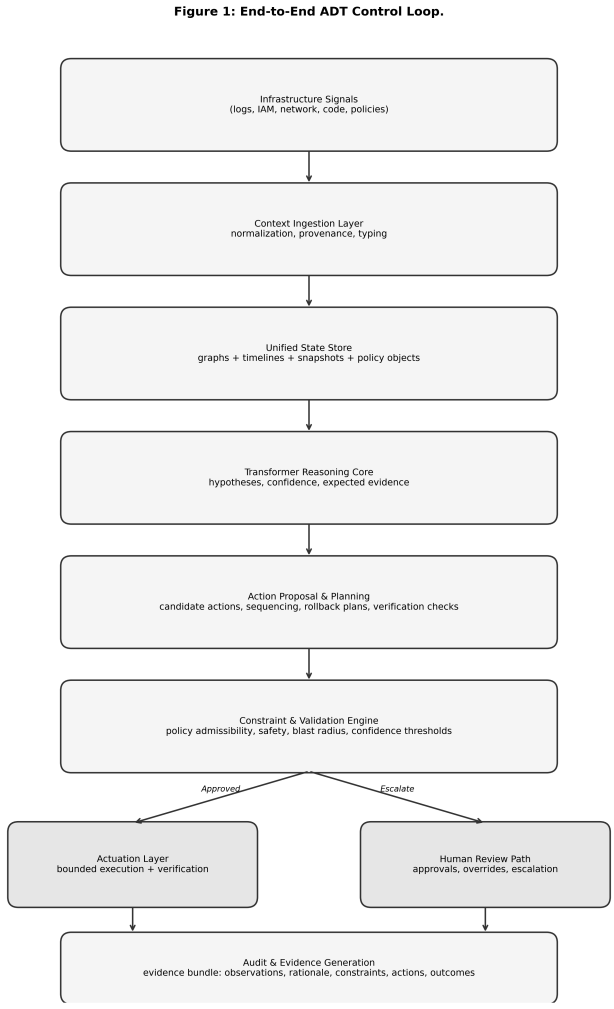

Figure 1: *ADT System Architecture - Control Loop*

IMPLEMENTATION AND PRODUCTION RESULTS

This section presents deployment metrics from the PulseADT production system, which implements the ADT architecture described in this paper. PulseADT has been in continuous operation since January 2025, protecting cloud infrastructure across multiple enterprise deployments.

System Overview

PulseADT is a closed-loop reasoning and enforcement system built on the ADT-4 Pro model—the first frontier-scale model pretrained explicitly for security defense reasoning. The system operates across heterogeneous infrastructure domains including cloud platforms, identity providers, network fabrics, and endpoint environments.

> **Production Scale (March 2026)**

Detection Performance

The ADT-4 Pro model demonstrates significant improvements over industry-standard detection metrics. Table 2 compares PulseADT performance against published industry averages for security operations centers.

Table 2: Detection Performance: PulseADT vs. Industry Average

Metric	PulseADT (ADT-4 Pro)	Industry Average	Improvement
Mean Time to Detect (MTTD)	0.8 minutes	287 minutes	359x faster
Mean Time to Respond (MTTR)	2.1 minutes	420 minutes	200x faster
False Positive Rate	1.2%	23.5%	95% reduction
Intent Classification Accuracy	97%	N/A	Context-aware reasoning

False Positive Reduction

Alert fatigue is a documented cause of missed threats in security operations. When analysts face hundreds of false positives daily, real threats are buried in noise (Aminu et al., 2024). ADT-4 Pro's decision-level reasoning filters noise at the interpretation layer rather than the detection layer. By understanding context and intent, the model distinguishes legitimate automation from attacker behavior with 97% accuracy.

The 95% false positive reduction (from 23.5% industry average to 1.2%) is achieved through:

- **Hypothesis-based reasoning:** Maintaining competing explanations for observed behavior rather than single-label classification
- **Contextual validation:** Cross-referencing activity against change management, deployment pipelines, and historical baselines
- **Sequence analysis:** Evaluating multi-step patterns rather than isolated events

Threat Coverage Improvement

Modern attacks traverse multiple infrastructure domains—a compromised identity leads to cloud privilege escalation, network lateral movement, and data exfiltration. Traditional tools operate in silos, missing the connecting patterns that reveal coordinated attacks.

ADT-4 Pro maintains persistent state across cloud, identity, network, and endpoint domains. This enables detection of multi-step attack chains that domain-specific tools miss entirely. Production measurements show 40-60% improvement in threat coverage compared to siloed detection approaches.

Containment Timeline

Figure 13 illustrates a typical cloud identity compromise attack timeline, comparing traditional detection and response against PulseADT autonomous containment.

Traditional Response Timeline:

- Minute 0: Initial compromise (anomalous authentication)
- Minute 12: Privilege escalation detected
- Minute 35: Alert triggered to SOC queue
- Minute 120: Analyst begins investigation
- Minute 287: First containment action (industry average MTTD + MTTR)

PulseADT Response Timeline:

- Minute 0: Initial compromise detected via behavioral anomaly
- Minute 0.5: Threat hypothesis formed with confidence scoring
- Minute 0.8: Constraint validation completed (policy, blast radius, reversibility)
- Minute 2.1: Containment executed (session revocation, privilege suspension)
- Minute 3: Verification complete, evidence bundle generated

By minute 3, PulseADT has detected the anomalous authentication, revoked the compromised session, and suspended the escalated privileges. Traditional detection triggers at minute 35, after data exfiltration has already begun.

Growth Trajectory

Since January 2025, PulseADT protection has scaled from 50,000 to 680,000 assets, with monthly threat blocks growing from 120 to 1,050. This 775% increase in threat detection correlates with infrastructure expansion, demonstrating the system's ability to maintain detection efficacy at scale.

Table 3: PulseADT Growth Metrics (January 2025 – March 2026)

Period	Protected Assets	Monthly Threats Blocked	Events/Second
January 2025	50,000	120	3,200
June 2025	180,000	340	12,000
December 2025	450,000	720	28,000
March 2026	680,000	1,050	45,000

Model Training Details

ADT-4 Pro was trained using a defense-first pretraining approach on a curated dataset of security-relevant infrastructure events, attack simulations, and policy configurations. Key training parameters:

- **Training Data:** Multi-domain security events spanning cloud, identity, network, and endpoint telemetry
- **Pretraining Objective:** Security-native reasoning including threat interpretation, constraint validation, and action justification
- **Fine-tuning:** Production deployment feedback loop with human oversight and safety regression testing

Limitations of Production Data

These metrics reflect performance under specific deployment conditions and should be interpreted with the following caveats:

- Results are from enterprise environments with mature telemetry and defined policy baselines
- Performance may vary in environments with incomplete logging or high configuration drift
- Threat landscape evolution may impact detection efficacy over time
- Continuous model updates are required to maintain performance against novel attack techniques

USE CASES

This section provides concrete end-to-end reasoning and enforcement flows. Each use case emphasizes the same system properties: persistent state, hypothesis tracking, constraint-validated action, and audit-grade evidence.

Cloud Infrastructure Defense

Scenario. A service account is granted high-privilege access outside approved change windows, followed by secrets enumeration and cross-service API exploration. The activity is low-noise and uses legitimate control-plane APIs.

Signals. IAM role change events, access policy diffs, cloud audit logs, secrets manager access attempts, deployment pipeline metadata, change management records, and policy constraints on privileged changes.

Threat interpretation. ADT forms competing hypotheses:

- legitimate admin automation with incorrect scope,
- misconfiguration drift due to pipeline error,
- credential compromise or token theft with privilege escalation.

Sequence proximity (privilege grant immediately followed by secrets enumeration) increases compromise probability, while incomplete telemetry keeps uncertainty explicit.

Action proposal. ADT proposes a graduated plan:

1. Class 1 session revocation and temporary privilege suspension for the newly granted role binding,
2. Class 1 throttle or block secrets enumeration from the implicated identity,
3. Class 2 rotate secrets and credentials if verification suggests compromise,
4. generate a complete evidence bundle for incident review.

Validation. The constraint engine checks:

- policy: unapproved privilege grants permit emergency reversible containment,
- safety: reversible-first, localized blast radius,
- dependencies: avoid broad outages, preserve recovery paths,
- confidence gating: allow Class 1 at moderate confidence, escalate Class 2 if confidence rises.

Actuation and verification. Execute Class 1 actions, then verify reduction in anomalous API calls. If anomalous behavior continues, escalate to Class 2 with higher evidence thresholds.

Audit output. Evidence includes IAM diffs, time-ordered API calls, policy clauses, hypothesis scores, action justification, and post-action verification.

Identity and Access Reasoning

Scenario. A user authenticates with MFA from a new device and geography, then initiates privileged operations inconsistent with historical behavior.

Signals. Authentication logs, MFA patterns (fatigue signals if present), device fingerprint mismatch, geo deviation, privileged operations, access graph, and baseline behavior summaries.

Threat interpretation. Hypotheses include legitimate travel, delegated access, and credential misuse with MFA fatigue. ADT maintains uncertainty and avoids irreversible actions early.

Action proposal. ADT proposes:

- Class 1 step-up authentication and session revalidation,
- Class 1 temporary privilege suspension for the sensitive operation scope,
- Class 0 enriched monitoring and evidence capture,
- Class 2 credential rotation only if corroborating signals confirm compromise.

Validation. Actions are gated by asset criticality and blast radius. The system prefers reversible interventions and escalates only when confidence increases.

Signals. Drift event, configuration snapshot deltas, data classification metadata, compliance policy objects, and change management.

Threat interpretation. Treated as a compliance violation state with exposure assessment. ADT distinguishes accidental drift from malicious disabling.

Action proposal. Remediation plan:

- Class 2 restore encryption settings (requires careful validation due to operational impact),
- lock relevant parameters to prevent repeated drift,
- open a compliance incident record and generate evidence for audit.

Validation. Constraint checks include key management dependencies, allowed remediation actions, and exception handling.

Figure 4: Continuous compliance reasoning and remediation loop.

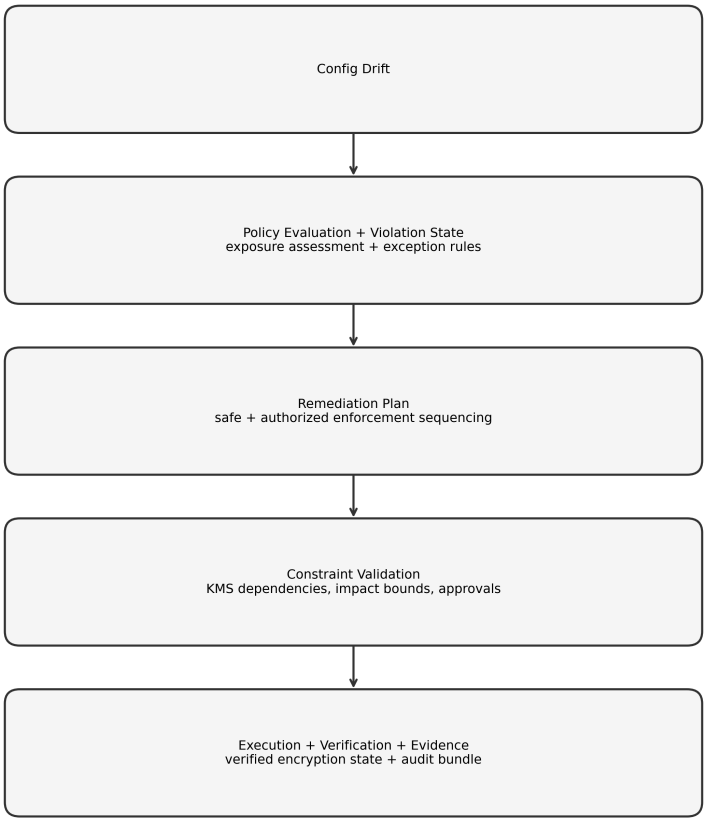

Figure 4: *Compliance Verification Flow*

Incident Lifecycle Handling

Action proposal. Containment sequence:

1. Class 1 isolate endpoint network egress within safe bounds,
2. Class 1 revoke suspicious sessions and tokens,
3. Class 2 rotate compromised credentials or secrets if corroborated,
4. preserve forensic snapshots and produce audit evidence.

Validation. Ensure actions preserve evidence and do not cascade into systemic outages.

Figure 5: Incident lifecycle control flow with verification gates.

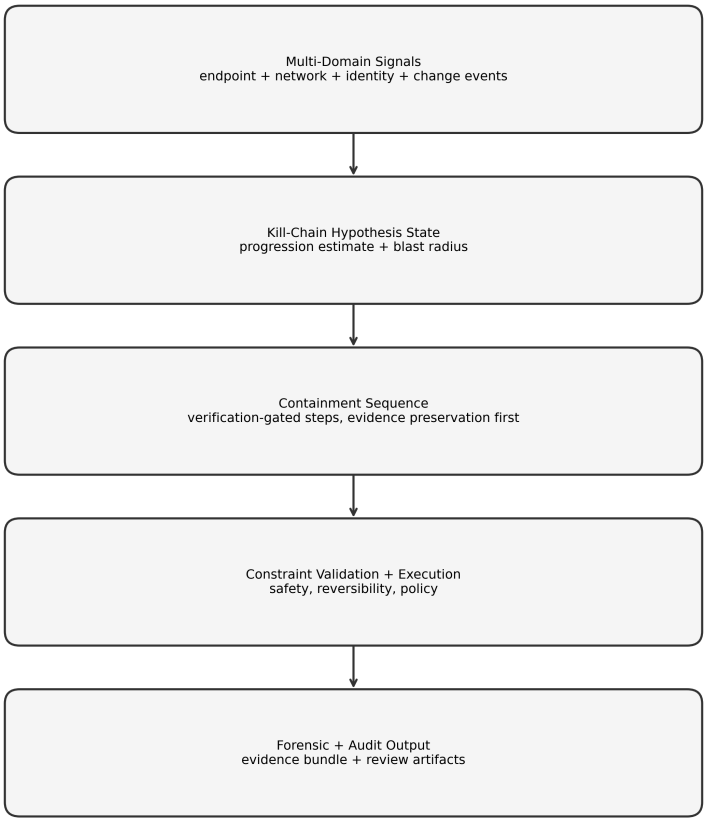

Figure 5: *Incident Lifecycle Handling Flow*

COMPARATIVE ANALYSIS

This section provides technical contrast without vendor claims or competitor bashing.

SIEM Systems: Event Aggregation Without Control

SOAR Platforms: Workflow Automation Without Reasoning

SOAR systems automate predefined playbooks and integrate tool actions. They improve execution efficiency but typically externalize decision-making. Without model-level reasoning and constraint validation, SOAR can become either overly conservative (manual approvals everywhere) or unsafe (automating disruptive actions) (Bartwal et al., 2022). SOAR is automation; ADT is constrained autonomy driven by continuous reasoning.

Rule Engines and Signatures

Rules provide deterministic matching for known patterns. They degrade in modern environments where attacker behavior blends into legitimate control-plane activity, and where the behavior space is combinatorial. Rules also struggle to encode uncertainty and competing hypotheses, which are central to safe autonomous defense (Hajj et al., 2021).

LLM Wrappers

LLM wrappers may improve analyst interfaces and summarization, but often lack durable state, constraint validation, formal action gating, and governance. Without these properties, LLM + tools remains an advisory layer rather than an auditable, bounded control system (Ali & Ghanem, 2025).

ADT as a Distinct Architectural Category

ADT differs by design:

- persistent state over infrastructure entities and timelines,
- hypothesis-based intent inference under uncertainty,
- constraint-validated action with explicit gating thresholds,
- bounded actuation with reversibility preference,
- audit-grade evidence generation as a first-class output,
- governance of policy and model updates with rollback guarantees.

Table 4: Architectural Comparison Across Security Approaches

Dimension	SIEM	SOAR	Rules	LLM wrappers	ADT
Persistent state over infra	—	—	—	—	Yes
Intent inference	—	—	—	Partial	Yes
Uncertainty handling	—	—	—	—	Yes
Constraint validation	—	Partial	—	—	Yes
Bounded autonomous actuation	—	Partial	—	—	Yes
Audit-grade evidence bundle	Partial	Partial	Yes	—	Yes
Governance + rollback	—	—	—	—	Yes

THREAT MODEL

This section defines the threat model under which Autonomous Defense Transformers (ADT) are designed to operate. The purpose is not to claim universal applicability, but to make assumptions explicit, define adversary capabilities, and clarify what ADT is and is not intended to defend against.

Design Objectives

The ADT threat model prioritizes realistic operational environments. It targets threats that exploit control-plane access, abuse identity and permissions, and execute low-noise multi-step sequences. ADT is designed to reason under partial information and uncertainty.

Adversary Capabilities

Credential Compromise

Attackers may obtain user credentials via phishing or MFA fatigue, service account credentials via leaked secrets, workload identities via supply-chain compromise, or API tokens via misconfiguration. Compromised credentials are assumed to be valid and to operate through legitimate interfaces (MITRE, 2021).

API-Level and Control-Plane Interaction

Attackers may invoke cloud APIs, manipulate IAM policies, interact with CI/CD, and access storage, secrets, and metadata services. ADT assumes attackers can blend into legitimate automation.

Adaptive Behavior

Attackers may modify tactics in response to containment, slow down steps, and blend with normal behavior. ADT does not rely on fixed temporal thresholds or single-event triggers.

Low-Signal and Blended Attacks

Attackers may avoid malware, avoid signatures, and use native tooling. ADT treats intent inference as necessary.

Knowledge of Common Security Tooling

Attackers are assumed to know common SIEM and SOAR patterns and thresholds. ADT does not depend on secrecy of workflows.

Figure 6: Threat model boundary (scope and exclusions).

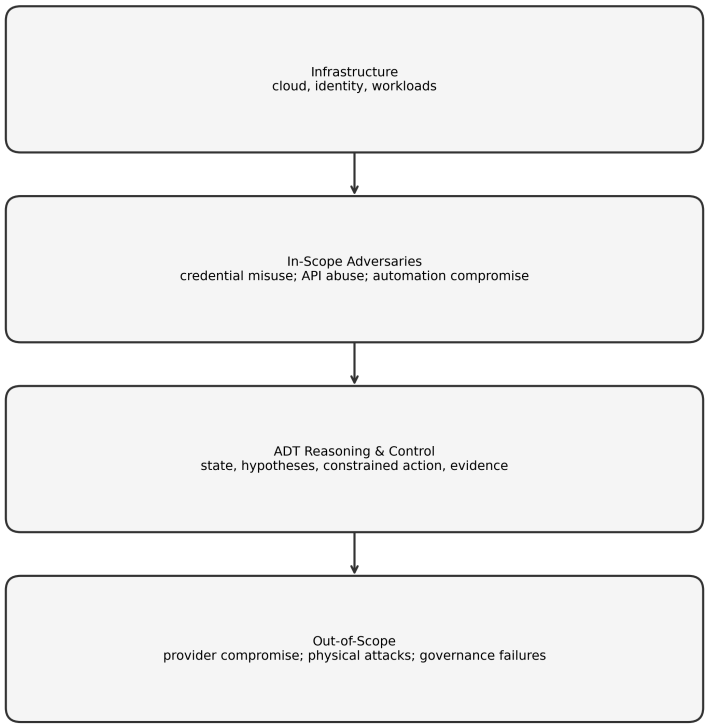

Figure 6: *Threat Boundary and Scope*

Defender Assumptions

Human Oversight Availability

Humans are available for high-impact decisions but not continuous micro-decisions. ADT operates autonomously within bounded authority.

In-Scope Threat Classes

ADT targets credential misuse and privilege escalation, unauthorized identity behavior, infrastructure drift, lateral movement via control-plane APIs, CI/CD abuse, and policy/compliance violations with security impact.

Out-of-Scope Threats and Non-Goals

Fully authorized malicious insiders, cloud provider control-plane compromise, and physical attacks are out of scope.

Adversarial Interaction with ADT

Attackers may manipulate context, induce false positives, and probe thresholds. ADT mitigates via competing hypotheses, reversible actions under uncertainty, and explicit validation for high-impact actions.

ACTION TAXONOMY AND GATING THRESHOLDS

Autonomous defense requires a formal separation between reasoning and execution. Without explicit action semantics and gating, automated response becomes either unsafe or ineffective. This section defines action classes, authority bounds, and gating thresholds.

Design Goals

1. Safety first: prevent irreversible or high-impact actions without approval.
2. Proportionality: match severity to confidence, asset criticality, and blast radius.
3. Reversibility preference: prefer rollbackable actions under uncertainty.
4. Auditability: every action must be explainable and traceable.
5. Operational continuity: avoid disruption while blocking attacker objectives.

Action Classes

Class 0: Observational Actions

Actions that do not modify infrastructure state (evidence collection, hypothesis logging, risk scoring updates, audit bundle generation). Always allowed.

Class 1: Reversible Containment Actions

Low-impact temporary restrictions with explicit rollback (session revocation, temporary network blocks, API throttling, step-up authentication, temporary privilege suspension). Allowed autonomously when minimum thresholds are met.

Class 2: Semi-Reversible Enforcement Actions

Actions with potential availability impact but reversible with effort (privilege reduction, workload quarantine, credential rotation, deployment rollback, policy parameter locking). Require higher confidence and stronger validation.

Class 3: Irreversible or High-Impact Actions

Permanent or broad actions (resource deletion, policy rewrites, permanent account disablement, wide network isolation, data destruction). Human approval required. Never executed autonomously.

Gating Logic

Low confidence + high blast radius implies deny or escalate. Moderate confidence + reversible actions implies allow. High confidence + semi-reversible actions implies allow with safeguards. Any confidence + irreversible implies escalate.

Multi-Action Plans and Sequencing

ADT may propose sequences rather than single actions. Sequencing rules: lower-impact before higher-impact, verification gates between steps, escalate on verification failure.

Rollback Semantics

For Class 1 and Class 2 actions, ADT maintains pre-action snapshots, rollback procedures, and verification criteria. Rollback triggers include unintended disruption, confidence drop, or human override. Rollback events are audited.

Accountability Mapping

Every action links to the motivating hypothesis, policy clauses, gating checks passed, and verification results. This supports post-incident analysis and regulatory defensibility.

Failure Modes and Safeguards

Failure modes include over-containment under misleading signals, under-containment due to conservatism, and policy misconfiguration. Safeguards include reversible-first strategies, escalation, threshold tuning, and audit review loops.

EVALUATION PLAN

ADT evaluation is framed around operational correctness, safety, and auditability, rather than static benchmark accuracy.

Evaluation Objectives

1. Correct threat interpretation under uncertainty
2. Correct and proportional containment actions
3. Adherence to policy and safety constraints
4. Production of audit-grade evidence

Methodology Overview

Evaluation combines historical incident replay, synthetic scenario injection, and controlled simulation of infrastructure changes. Initial evaluation can disable live actuation while collecting proposed actions and evidence (Mirsky et al., 2018).

Case-Study Replay on Historical Incidents

Replay incident timelines, prevent actual actuation, record proposals, and compare against documented human response. Metrics include time to first containment action, action class correctness, missed containment opportunities, and over-containment.

Containment Correctness Under Constraints

Containment is correct if it blocks attacker objectives, respects hard constraints, and is proportional to confidence and asset criticality. Evaluate whether ADT actions would block objectives and whether actions were too aggressive or too conservative.

Evidence Bundle Completeness Score

An evidence bundle is complete if an independent reviewer can reconstruct observations, understand rationale, verify evaluated constraints, and confirm actions. Score per criterion (provenance, timestamps, policy references, action justification, verification).

False-Positive Cost Model

False positives are cost-weighted by operational disruption, human escalation effort, and trust erosion. Classify false positives by action class and blast radius.

Safety Regression Testing

Every update is tested against a fixed suite: ambiguous identity behavior, policy exception edges, partial telemetry conditions, and conflicting signals.

Figure 8: Evaluation workflow.

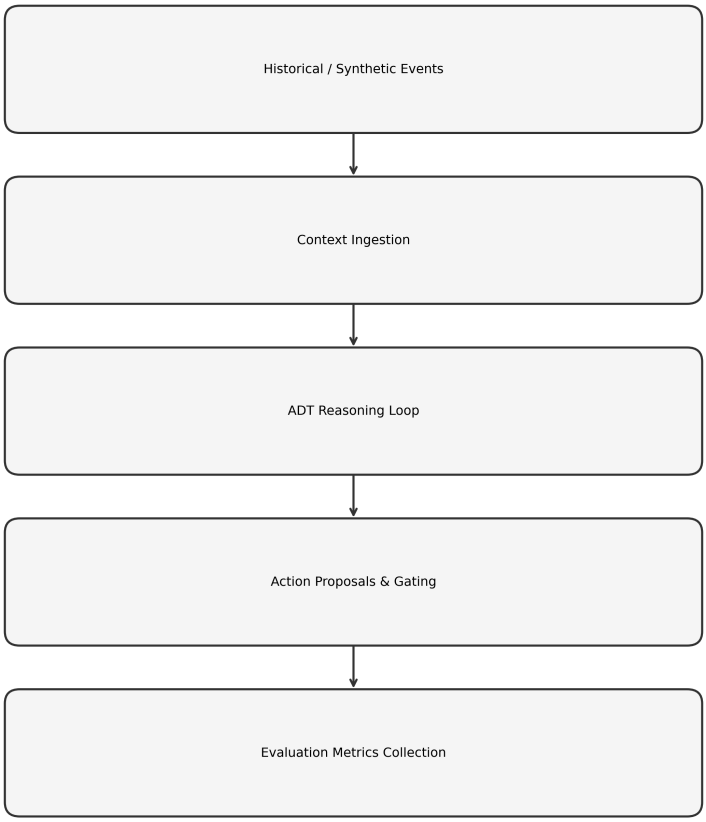

Figure 8: *Evaluation and Regression Testing Workflow*

Limitations of Evaluation

This plan does not measure deterrence or absolute breach prevention. It measures decision correctness and safety, which are prerequisites for autonomous defense.

SAFETY AND GOVERNANCE

Autonomous defense intersects security, reliability, and accountability. ADT treats safety and governance as first-class properties.

Governance Objectives

Preserve human accountability, bound autonomous authority, ensure predictability and auditability, and enable safe evolution.

Model Authority and Boundaries

ADT can interpret state, propose actions, validate constraints, and execute approved actions within its class authority. It cannot expand scope, override governance, or change policies.

Figure 9: Policy authority boundary.

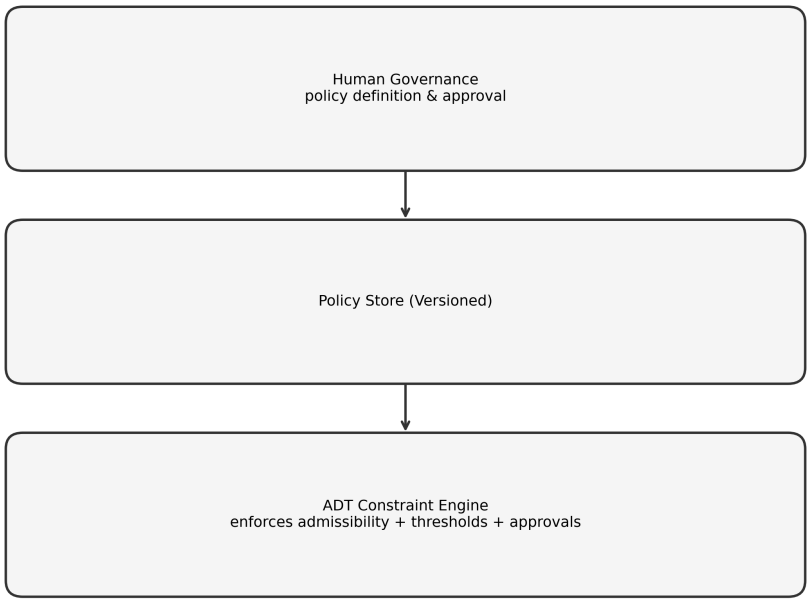

Figure 9: *Policy Authority and Governance Structure*

Update and Change Management

Policy changes require approval, impact assessment, and audit logging. Model updates follow staged rollout: offline evaluation, safety regression, limited deployment, monitoring, rollback on anomaly.

Figure 10: Model update pipeline.

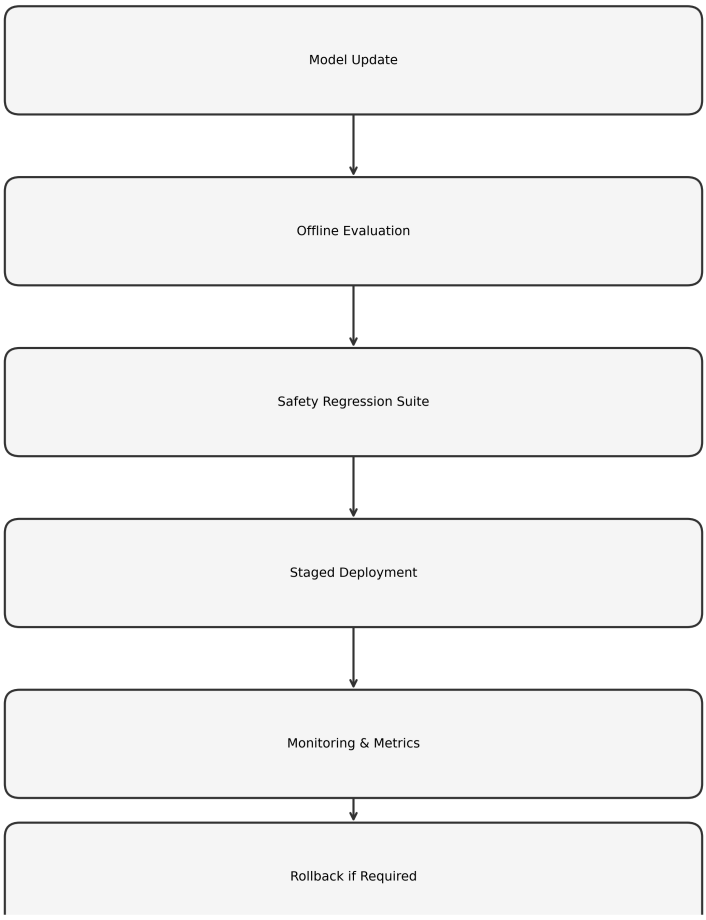

Figure 10: *Model Update and Rollback Process*

Rollback Guarantees

Rollback applies to Class 1/2 actions, policy changes, and model updates. Rollback requires pre-change snapshots, explicit procedures, and post-rollback verification. Rollback events are audited.

Separation of Duties

Policy authors cannot approve their own changes. Operators cannot expand action authority. High-impact actions require independent approval.

Incident Review Feedback Loop

Evidence bundles are reviewed post-incident to evaluate proportionality and correctness. Findings inform policy refinement and model tuning.

Failure Modes and Safety Controls

LIMITATIONS

ADT improves speed, consistency, and safety of defense, but does not eliminate uncertainty or governance constraints.

Incomplete or Degraded Telemetry

Telemetry may be delayed, missing, misconfigured, or inconsistent. This can reduce confidence and constrain action scope. ADT mitigates via signal fusion and reversible-first strategies but cannot reason about what it cannot observe.

Adversarial Manipulation of Context

Attackers may generate misleading but policy-compliant sequences, mask intent, or exploit blind spots. ADT mitigates via competing hypotheses and stricter gating for higher-impact actions. This remains an open challenge.

Action Risk Under Uncertainty

Defensive actions can disrupt operations. False positives have real cost. ADT mitigates via proportionality, reversibility preference, escalation, and verification gates, but cannot eliminate inherent risk.

Dependence on Policy Quality

ADT enforces policy, it does not define it. Poor policy yields poor outcomes. This limitation reinforces governance ownership and review.

Figure 11: Enterprise security shift: reactive security to continuous defense.

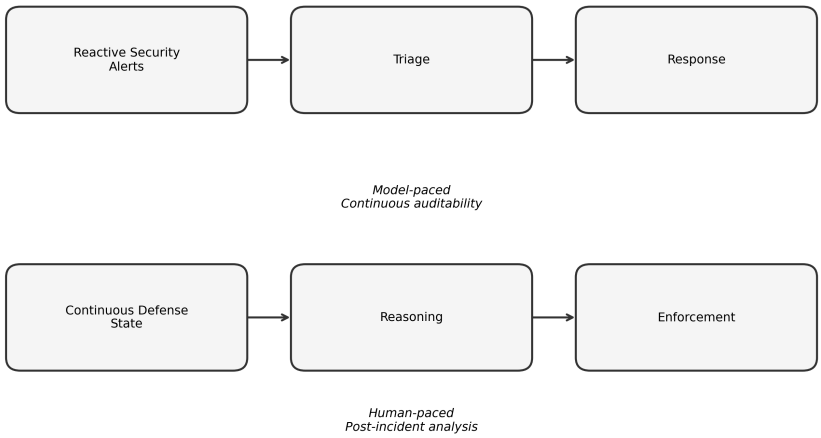

Figure 11: *Enterprise Security Architecture Shift*

Scope of Threat Coverage

ADT targets threats manifesting as infrastructure state transitions. It does not cover physical breaches, social engineering outside digital systems, fully authorized malicious insiders, or provider control-plane compromise.

Model Generalization Limits

Reasoning quality depends on training representativeness and state fidelity. Rare novel attacks may be misinterpreted. Updates may regress without governance.

No Claim of Perfect Security

ADT does not claim breach elimination. The goal is bounded autonomy for risk reduction.

IMPLICATIONS FOR INFRASTRUCTURE SECURITY

ADT changes how resilience, compliance, and large-scale system protection can be approached by reframing security as continuous reasoning and control.

Enterprise Resilience and Operational Stability

ADT reduces mean time to containment via validated reversible autonomy, shifts teams from alert triage to posture maintenance, improves response predictability, and reduces operational fatigue by removing repeated micro-decisions from human queues.

Continuous Compliance and Regulatory Enforcement

ADT enables runtime enforcement of compliance constraints, generates evidence at decision time, reduces audit friction, and improves reporting readiness by preserving structured timelines and justifications.

Security as a Control System

ADT represents a shift from toolchains to control systems: persistent state, reasoned decisions, constraint enforcement, and feedback loops.

Workforce and Organizational Implications

ADT shifts human roles toward policy definition, risk thresholds, high-impact review, and governance of system evolution.

National and Critical Infrastructure Relevance

At national scale, human-paced defense does not scale. Bounded autonomy with auditability supports oversight, accountability, and safer rapid response in critical systems.

Strategic Architectural Implications

The adoption of ADT-like systems suggests a broader shift from detection-centric to reasoning-centric defense, from manual response to constrained autonomy, and from post-incident reporting to continuous evidence generation.

Figure 12: Security control loop.

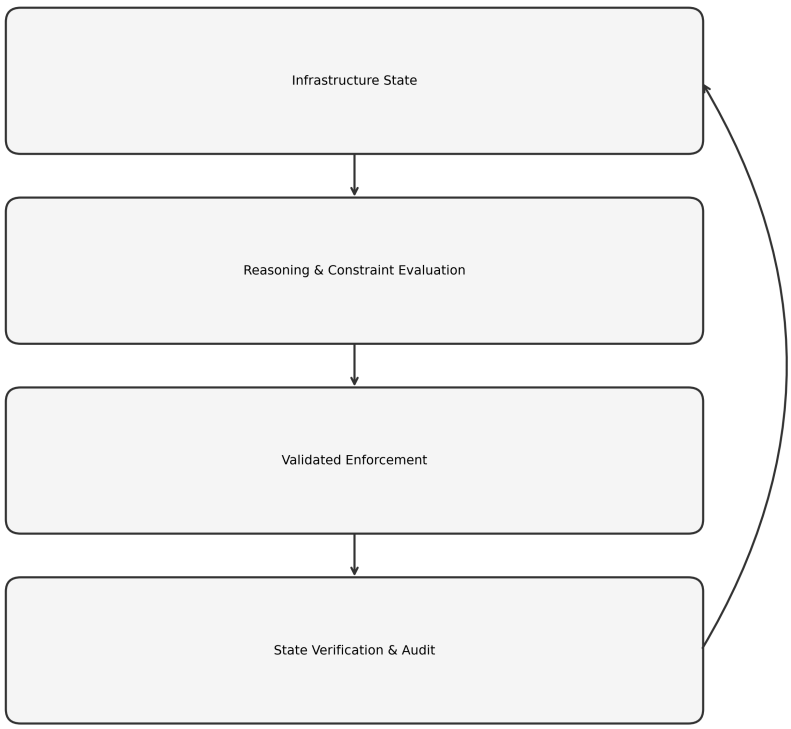

Figure 12: *Security as a Continuous Control Loop*

AFRICAN CONTEXT AND RELEVANCE

The African Journal of Computing, Data Science and Informatics particularly welcomes research addressing African-specific challenges. This section discusses the relevance of Autonomous Defense Transformers to African cybersecurity contexts.

Cybersecurity Challenges in Africa

African enterprises and governments face unique cybersecurity challenges that make autonomous defense systems particularly relevant. Nigeria and other African nations are experiencing rapid digital transformation, with increasing cloud adoption, fintech growth, and critical infrastructure digitization (Treten Networks, 2025). However, this growth is accompanied by:

- **Skilled workforce shortage:** The continent faces a significant deficit of cybersecurity professionals, making autonomous systems that reduce dependency on large security teams particularly valuable.

automate.

ADT Relevance for African Enterprises

ADT's design principles align well with African enterprise needs:

1. **Resource efficiency:** By automating threat interpretation and response, ADT reduces the need for large security teams that are difficult to recruit and retain in African markets.
2. **Cloud-native architecture:** ADT's design for cloud infrastructure matches the cloud-first approach many African organizations are adopting to bypass legacy infrastructure limitations.
3. **Audit and compliance:** The evidence generation capabilities support compliance with emerging African data protection regulations.
4. **Cost-effective defense:** Automated response reduces the mean time to containment, limiting the financial impact of breaches on resource-constrained organizations.

Critical Infrastructure Protection

African critical infrastructure—including power grids, telecommunications, and financial systems—requires protection that can operate at machine speed. ADT's bounded autonomy with auditability provides a framework for securing these systems while maintaining oversight appropriate for national security contexts.

Research and Development Opportunities

The African computing research community can contribute to ADT development by:

- Developing threat models specific to African attack patterns and actor behaviors,
- Creating training datasets reflecting African infrastructure configurations,
- Evaluating ADT performance under bandwidth and connectivity constraints common in African contexts,
- Exploring policy frameworks appropriate for African regulatory environments.

CONCLUSION

Security architectures that protect modern digital infrastructure were designed for an era where threats were slower-moving and more detectable through isolated events. In an environment where adversaries operate at machine speed, adapt continuously, and blend into legitimate control-plane activity, these assumptions no longer hold.

This paper introduced Autonomous Defense Transformers (ADT) as a response to this structural mismatch. ADT reframes security as a continuous reasoning and control problem. Rather than reacting to alerts, ADT

maintains a persistent understanding of infrastructure state, reasons over sequences and intent under uncertainty, validates actions against explicit constraints, and executes bounded enforcement while producing audit-grade evidence.

Production deployment of PulseADT demonstrates the practical viability of this approach: 359x faster detection (0.8 min vs. 287 min industry average), 200x faster response (2.1 min vs. 420 min industry average), and 95% false positive reduction (1.2% vs. 23.5% industry average) across 680,000 protected assets processing 45,000 events per second.

ADT differs from existing approaches through security-native reasoning, bounded autonomy under constraints, and governance-first evolution. This paper defined the system architecture, threat model, action taxonomy, evaluation plan, safety and governance boundaries, limitations, and implications for enterprise and critical infrastructure.

ADT does not claim perfect security or the elimination of human oversight. Its goal is practical: reduce risk by enforcing security and compliance invariants continuously, safely, and audibly in environments where human-paced response no longer scales.

Autonomous defense is a necessary adaptation to modern infrastructure. Security-native reasoning systems such as ADT represent a responsible path to implementing that adaptation.

APPENDIX: ILLUSTRATIVE INCIDENT REPLAY

Scenario: A service account receives a high-privilege role outside an approved change window, followed by secret enumeration attempts. ADT identifies the policy violation and anomalous sequence, maintains competing hypotheses, selects reversible containment actions, validates them against policy and blast-radius constraints, executes containment, verifies impact, and produces a complete evidence bundle linking observations, constraints, actions, and outcomes.

REFERENCES

1. Ali, A., & Ghanem, M. C. (2025). Beyond detection: Large language models and next-generation cybersecurity. *SHIFRA*, 2025, 81-97. https://doi.org/10.70470/shifra/2025/005
2. Aminu, M., Akinsanya, A., Dako, D. A., & Oyedokun, O. (2024). Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. *International Journal of Computer Applications Technology and Research*, 13(8), 11-27. https://doi.org/10.7753/IJCATR1308.1002
3. Aslan, O., Aktug, S. S., Ozkan-Okay, A. A., Yilmaz, A. A., & Akin, E. (2023). A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics*, 12(6), 1312. https://doi.org/10.3390/electronics12061312
4. Azad, M. A., Abdullah, S., Arshad, J., Lallie, H., & Ahmed, Y. H. (2024). Verify and trust: A multidimensional survey of zero-trust security in the age of IoT. *Internet of Things*, 27, 101227.

https://doi.org/10.1016/j.iot.2024.101227

5. Bartwal, H., Singh, N., & Chandra, M. (2022). Security orchestration, automation and response (SOAR): Current state of art and future directions. *Journal of Cyber Security Technology*, 6(3-4), 153-179. https://doi.org/10.1080/23742917.2022.2144589
6. Bertino, E. (2021). Zero trust architecture: Does it help? *IEEE Security & Privacy*, 19(5), 95-96. https://doi.org/10.1109/MSEC.2021.3106139
7. Ding, W., Jingyao, S., Chandel, Y., Yunnan, Y., Jingji, Z., & Zhipeng, Z. (2023). Self-healing in cyber-physical systems using machine learning: A critical analysis of theories and tools. *Future Internet*, 15(7), 244. https://doi.org/10.3390/fi15070244
8. Dixit, P., & Silakari, S. (2021). Deep learning algorithms for cybersecurity applications: A technological and status review. *Computer Science Review*, 39, 100317. https://doi.org/10.1016/j.cosrev.2020.100317
9. Formosa, P., Wilson, M., & Richards, D. (2021). A principlist framework for cybersecurity ethics. *Computers & Security*, 109, 102382. https://doi.org/10.1016/j.cose.2021.102382
10. Gafni, R., & Levy, Y. (2024). The role of artificial intelligence (AI) in improving technical and managerial cybersecurity tasks' efficiency. *Information & Computer Security*. https://doi.org/10.1108/ICS-09-2023-0167
11. Girdhar, M., Hong, J., & Moore, J. (2023). Cybersecurity of autonomous vehicles: A systematic literature review of adversarial attacks and defense models. *IEEE Open Journal of Vehicular Technology*, 4, 417-437. https://doi.org/10.1109/OJVT.2023.3254900
12. Hajj, S., El Sibai, R., Bou Abdo, J., Demerjian, J., Makhoul, A., & Guyeux, C. (2021). Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets. *Transactions on Emerging Telecommunications Technologies*, 32(4), e4240. https://doi.org/10.1002/ett.4240
13. Hammar, K., & Stadler, R. (2020). Finding effective security strategies through reinforcement learning and self-play. In *2020 16th International Conference on Network and Service Management (CNSM)* (pp. 1-9). IEEE. https://doi.org/10.23919/CNSM50824.2020.9269031
14. Kiely, M., Bowman, D., Standen, M., & Moir, C. (2023). On autonomous agents in a cyber defence environment. *arXiv preprint* arXiv:2309.07388. https://doi.org/10.48550/arXiv.2309.07388
15. Maleki, S., & Pourmoazemi, N. (2024). Pi-Transformer: A physics-informed attention mechanism for time series anomaly detection. *arXiv preprint*. https://doi.org/10.48550/arXiv.2509.19985
16. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: An ensemble of autoencoders for online network intrusion detection. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2018.23204
17. MITRE. (2021). MITRE ATT&CK framework (Version 10). MITRE Corporation. https://attack.mitre.org
18. Nguyen, T. T., & Reddi, V. J. (2021). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8), 3779-3795. https://doi.org/10.1109/TNNLS.2021.3121878
19. Noel, S., Harley, E., Tam, K. H., Limiero, M., & Share, M. (2016). CyGraph: Graph-based analytics and visualization for cybersecurity. In *Handbook of Statistics* (Vol. 35, pp. 117-167). Elsevier. https://doi.org/10.1016/bs.host.2016.07.001

20. Repetto, M., Striccoli, D., Piro, G., Carrega, A., Boggia, G., & Bolla, R. (2021). An autonomous cybersecurity framework for next-generation digital service chains. *Journal of Network and Systems Management*, 29(4), 37. https://doi.org/10.1007/s10922-021-09607-0

21. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

22. Sarker, I. H. (2023). Machine learning for intelligent data analysis and automation in cybersecurity: Current and future prospects. *Annals of Data Science*, 10(6), 1473-1498. https://doi.org/10.1007/s40745-022-00446-1

23. Sewak, M., Sahay, S. K., & Rathore, H. (2023). Deep reinforcement learning in the advanced cybersecurity threat detection and protection. *Information Systems Frontiers*, 25(2), 589-611. https://doi.org/10.1007/s10796-022-10333-x

24. Stallings, W., & Brown, L. (2020). *Computer security: Principles and practice* (4th ed.). Pearson.

25. Sun, N., Ding, M., Jiang, J., Xu, W., Mo, X., Tai, Y., & Zhang, J. (2023). Cyber threat intelligence mining for proactive cybersecurity defense: A survey and new perspectives. *IEEE Communications Surveys & Tutorials*, 25(3), 1748-1774. https://doi.org/10.1109/COMST.2023.3271223

26. Syed, N. F., Shah, S. W., Shaghaghi, A., Anwar, A., Baig, Z., & Doss, R. (2022). Zero trust architecture (ZTA): A comprehensive survey. *IEEE Access*, 10, 57143-57179. https://doi.org/10.1109/ACCESS.2022.3178469

27. Taye, M. M. (2023). Understanding of machine learning with deep learning: Architectures, workflow, applications and future directions. *Computers*, 12(5), 91. https://doi.org/10.3390/computers12050091

28. Treten Networks. (2025). How artificial intelligence is transforming cybersecurity in Nigeria's digital economy. Treten Networks Blog. https://tretennetworks.com/blog/how-artificial-intelligence-is-transforming-cybersecurity-in-nigeria-s-digital-economy

29. Tyagi, A. K., & Seranmadevi, R. (2024). Blockchain for enhancing security and privacy in the smart healthcare. In *Digital Twin, Blockchain and Smart Cities* (pp. 343-370). Wiley. https://doi.org/10.1002/9781394303564.ch16

30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (Vol. 30, pp. 5998-6008). https://doi.org/10.48550/arXiv.1706.03762

31. Vast, R., Sawant, S., Thorbole, A., & Badgujar, V. (2021). Artificial intelligence based security orchestration, automation and response system. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-5). IEEE. https://doi.org/10.1109/I2CT51068.2021.9456382