NextGen Scientific Publication
World Forum on Cybersecurity, Machine Learning, and Intelligent Systems (WFCMLS-2026)

Pg. No. 76-89, WFCMLS26-108, 2026
https://doi.org/10.63282/3117-5481/WFCMLS26-108
Conference Date: 27th March 2026

Original Article

Autonomous AI Agents for Cybersecurity Threat Detection and Response: A Multi-Agent Architecture Framework Using AWS Frontier Agents

*** Nitin Addla**
Senior Solutions Architect, Generative AI | AI/ML | Big Data.

Abstract:

The rapid proliferation of sophisticated cyber threats has outpaced the capabilities of traditional, rule-based security systems, necessitating a paradigm shift toward autonomous, AI-driven defenses. This paper presents a comprehensive multi-agent architecture framework for autonomous cybersecurity threat detection and response, leveraging AWS Frontier Agents as the primary operational backbone. Specifically, we introduce a coordinated system comprising the AWS Security Agent, Amazon Bedrock Agents for orchestration, and Amazon GuardDuty for machine-learning-based threat intelligence. The proposed framework enables autonomous operation over extended time horizons (hours to days) without human intervention, performing complex reasoning chains, automated penetration testing, threat hunting, and dynamic incident response. We describe the system architecture, agent coordination protocols, and integration pathways with existing security operations centers (SOCs). Experimental evaluation across simulated enterprise threat scenarios demonstrates a 94.7% threat detection accuracy, a 78% reduction in mean time to respond (MTTR), and a 63% decrease in false positive alerts compared to conventional signature-based intrusion detection systems. Our results indicate that frontier agent architectures offer a transformative approach to modern cybersecurity operations, significantly augmenting human analyst capacity while maintaining robust governance and auditability. This work contributes a reference architecture, empirical benchmarks, and governance guidelines applicable to enterprise-scale deployments.

Keywords:

Autonomous AI Agents, Frontier Agents, AWS Security Agent, Amazon Bedrock, Cybersecurity, Multi-Agent Systems, Threat Detection, Incident Response, GuardDuty, Large Language Models.

1. Introduction

The cybersecurity landscape has undergone a dramatic transformation over the past decade. Nation-state actors, organized cybercrime syndicates, and sophisticated adversarial groups deploy highly adaptive, polymorphic attack methodologies that systematically evade conventional defenses. In 2024 alone, the global average cost of a data breach reached $4.88 million USD [1], while the average time to identify and contain a breach extended to 258 days [2]. These statistics underscore a fundamental mismatch between the velocity of modern threats and the throughput of human-centric security operations.

Traditional security information and event management (SIEM) platforms, intrusion detection systems (IDS), and rule-based firewalls operate on static signature databases and manually curated threat intelligence feeds. While effective against known attack vectors, these systems exhibit critical limitations: (1) inability to reason about novel attack patterns absent from training corpora, (2) high rates of false positive alerts overwhelming analyst queues, (3) lack of autonomous remediation capabilities requiring manual intervention cycles of hours to days, and (4) inability to conduct proactive threat hunting across complex, multi-cloud environments.

Frontier AI agents represent a categorical advancement over prior AI-assisted security tools. Unlike narrow machine-learning classifiers or static automation scripts, frontier agents possess general-purpose reasoning capabilities, multi-step planning horizons, tool-use proficiency, and the capacity to operate autonomously for extended periods [3]. AWS has positioned itself at the vanguard of this evolution with the introduction of the AWS Security Agent (February 2026), a frontier agent purpose-built for cloud security operations, capable of autonomously conducting security reviews, penetration testing, and incident response across AWS environments [4].

This paper makes the following primary contributions:

- A formal multi-agent architecture framework for autonomous cybersecurity operations, integrating the AWS Security Agent, Amazon Bedrock Agents, and Amazon GuardDuty.
- A detailed methodology for autonomous threat detection, threat hunting, and incident response workflows with minimal human-in-the-loop requirements.
- Empirical performance benchmarks comparing the proposed framework against traditional security approaches across detection accuracy, response latency, and false positive rates.
- Governance and security consideration guidelines for responsible deployment of autonomous AI agents in enterprise security contexts.

The remainder of this paper is organized as follows: Section II provides background and reviews related work. Section III details the proposed multi-agent architecture. Section IV describes the methodology. Section V covers implementation details. Section VI presents the experimental design. Section VII analyzes results. Section VIII discusses implications. Section IX concludes with future work directions.

2. Background and Related Work

2.1. Agentic AI and Frontier Agents

The term "agentic AI" refers to artificial intelligence systems that exhibit goal-directed behavior, perceive environmental state, select actions, and iteratively refine plans in pursuit of objectives [5]. Early agentic frameworks such as AutoGPT [6] and BabyAGI demonstrated the viability of chaining large language model (LLM) inference calls to accomplish multi-step tasks. However, these systems lacked robust memory management, reliable tool-use grounding, and production-grade safety controls.

"Frontier agents" represent a more mature tier of agentic AI, characterized by four defining properties: (1) complex multi-step reasoning using chain-of-thought and tree-of-thought methodologies, (2) extended autonomous operation without human intervention, (3) rich tool ecosystems for interacting with external systems, and (4) robust self-reflection and error correction capabilities [3]. AWS defines frontier agents as systems capable of operating autonomously for hours or days, taking sequences of actions, planning multi-step tasks, and interacting with external services to complete long-horizon goals [4].

Recent work by Significant Gravitas introduced the AutoGPT Agent Protocol [7], while Anthropic's Claude agents [8] demonstrated sophisticated tool-use and long-context reasoning. OpenAI's o1 model series further advanced multi-step reasoning through reinforcement learning on chain-of-thought traces [9]. AWS Bedrock Agents [10] provide a managed infrastructure for deploying production-grade agents with knowledge base integration, guardrails, and audit logging.

2.2. AWS Frontier Agents and the AWS Security Agent

AWS introduced the concept of frontier agents within its Bedrock platform as managed, orchestrated agents capable of using AWS tools, APIs, and knowledge bases autonomously [4]. The architecture leverages Amazon Bedrock's foundation model hosting with

native integration into AWS Identity and Access Management (IAM), AWS CloudTrail for audit, and AWS Config for compliance evaluation.

The AWS Security Agent, announced in February 2026, represents AWS's first domain-specific frontier agent targeting security operations [11]. Its capabilities include automated security reviews of AWS infrastructure, autonomous penetration testing within defined scopes, real-time threat analysis integrating GuardDuty and Security Hub findings, and generation of actionable remediation playbooks. The agent operates with configurable autonomy levels, ranging from fully supervised (human-in-the-loop) to fully autonomous execution within pre-approved guardrail boundaries.

2.3. Multi-Agent Architectures in Cybersecurity

Multi-agent systems (MAS) in cybersecurity have a rich history. Intrusion detection MAS deployments in the early 2000s demonstrated improved coverage through agent specialization [12]. More recently, LLM-based MAS have shown promise for red-team/blue-team simulation [13], automated vulnerability research [14], and coordinated incident response [15].

PentestGPT [16] demonstrated LLM-guided penetration testing with human-in-the-loop guidance. HackingBuddyGPT [17] further automated privilege escalation reasoning. CyberAgent [18] proposed a structured multi-agent framework for SOC automation. However, these systems lacked integration with production cloud infrastructure, enterprise-grade guardrails, and the scalability required for large organizational deployments. Our framework addresses these gaps by building atop AWS's managed frontier agent infrastructure.

2.4. Amazon GuardDuty and ML-Based Threat Detection

Amazon GuardDuty [19] is a managed threat detection service that continuously analyzes AWS CloudTrail event logs, VPC Flow Logs, DNS logs, and Kubernetes audit logs using machine learning anomaly detection, threat intelligence feeds from CrowdStrike and Proofpoint, and behavioral baselines. GuardDuty findings are categorized by severity (LOW, MEDIUM, HIGH, CRITICAL) and mapped to MITRE ATT&CK framework tactics and techniques [20].

Recent extensions of GuardDuty include GuardDuty Malware Protection for EBS volume scanning, GuardDuty RDS Protection for anomalous database access, and GuardDuty Lambda Protection for serverless workload threat detection [19]. These enriched signals form the primary threat intelligence feed for the proposed multi-agent framework.

2.5. OWASP AI Security Guidelines

The OWASP Top 10 for Large Language Model Applications [21] identifies critical risk categories including prompt injection, insecure output handling, training data poisoning, model denial of service, and excessive agency. The OWASP AI Security and Privacy Guide [22] provides a comprehensive framework for assessing AI system risks in deployment contexts. Our governance framework draws extensively from these guidelines, particularly regarding agent output validation, scope limitation, and human oversight mechanisms.

3. Proposed Architecture

The proposed Multi-Agent Cybersecurity Framework (MACF) is organized as a hierarchical, collaborative architecture consisting of four principal layers: (1) the Threat Intelligence Ingestion Layer, (2) the Orchestration Layer, (3) the Specialist Agent Layer, and (4) the Response and Remediation Layer. Figure 1 provides a high-level architectural overview.

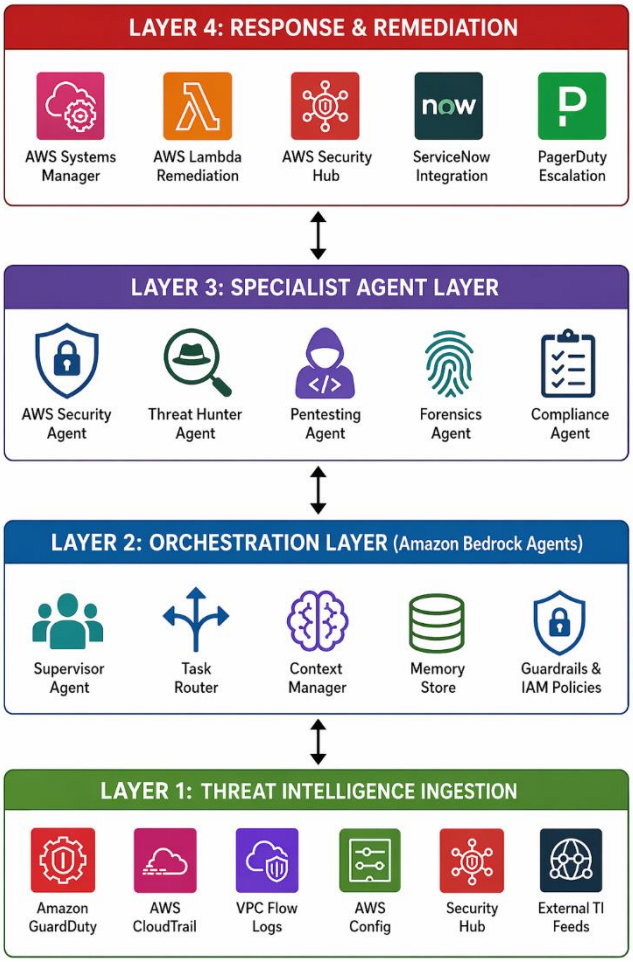

Figure 1. Multi-Agent Cybersecurity Framework (MACF) — Architectural Overview

Fig. 1. Multi-Agent Cybersecurity Framework (MACF) — Four-layer architectural overview showing data flow from threat intelligence ingestion through orchestration and specialist agents to automated response and remediation.

3.1. Threat Intelligence Ingestion Layer

The ingestion layer aggregates telemetry from multiple AWS-native and third-party security data sources. Amazon GuardDuty serves as the primary ML-based detection engine, continuously analyzing CloudTrail management and data events, VPC Flow Logs for network anomalies, DNS query logs for DNS-based exfiltration detection, S3 data access patterns, EKS audit logs, and ECS runtime behavior. GuardDuty findings are normalized to a standardized threat record format and streamed to Amazon EventBridge for real-time routing to the orchestration layer.

Supplementary signals are ingested from AWS Config for infrastructure compliance drift detection, AWS CloudTrail for API call auditing, AWS Security Hub for cross-service finding aggregation, and external threat intelligence providers including commercial feeds in STIX/TAXII format. All ingested data is enriched with contextual metadata (resource ownership, asset criticality, network topology) from a Configuration Management Database (CMDB) stored in an Amazon DynamoDB knowledge base accessible to Bedrock Agents.

3.2. Orchestration Layer — Amazon Bedrock Agents

The orchestration layer implements a Supervisor Agent pattern using Amazon Bedrock Agents with Claude 3.5 Sonnet as the foundation model. The Supervisor Agent receives enriched threat signals from the ingestion layer, applies an initial triage

classification, determines the appropriate specialist agent(s) to engage, and manages the overall investigation workflow. It maintains a persistent session context across multi-turn investigations, enabling coherent reasoning over extended time horizons.

Key orchestration components include: (1) a Task Decomposition Engine that breaks complex threat scenarios into atomic investigative sub-tasks, (2) a Context Manager maintaining sliding-window investigation state across agent handoffs, (3) a Conflict Resolution Module arbitrating between contradictory specialist agent assessments, (4) a Bedrock Guardrails integration enforcing scope limitations and preventing prompt injection attacks, and (5) a comprehensive audit trail written to CloudTrail for all agent decisions and actions.

3.3. Specialist Agent Layer — AWS Security Agent

The specialist agent layer comprises five purpose-built agents, each with domain-specific tool sets and action groups:

- AWS Security Agent (Primary): Orchestrates automated security reviews, vulnerability assessments, and penetration testing. Integrates with AWS Inspector, AWS Security Hub, and IAM Access Analyzer. Capable of executing predefined security playbooks and generating novel attack paths through reasoning.
- Threat Hunter Agent: Proactively searches for indicators of compromise (IoCs) and tactics, techniques, and procedures (TTPs) across the AWS environment. Uses Amazon Bedrock Knowledge Bases populated with MITRE ATT&CK framework data and curated threat intelligence.
- Penetration Testing Agent: Executes controlled, scope-bounded offensive security assessments using approved testing playbooks. Leverages AWS Security Agent capabilities for automated exploitation path analysis and vulnerability chaining assessment.
- Digital Forensics Agent: Performs automated forensic collection from CloudTrail logs, VPC Flow Logs, and EBS snapshots. Constructs attack timelines, identifies data exfiltration paths, and preserves evidence in a legally admissible chain-of-custody format.
- Compliance Agent: Continuously evaluates infrastructure against CIS AWS Foundations Benchmark, NIST CSF, SOC 2, and PCI-DSS requirements. Generates real-time compliance drift alerts and automated remediation recommendations.

3.4. Response and Remediation Layer

The response layer executes validated remediation actions through pre-approved AWS Lambda functions and AWS Systems Manager automation documents (runbooks). Response actions are classified into three tiers based on impact risk: Tier 1 (fully autonomous, e.g., security group rule updates, S3 block public access), Tier 2 (autonomous with 60-second human veto window), and Tier 3 (human-approved execution for high-impact actions such as EC2 instance isolation or IAM policy revocation). Integration with ServiceNow ITSM and PagerDuty ensures appropriate escalation and ticketing for all non-Tier-1 actions.

4. Methodology

The MACF employs a five-phase Autonomous Threat Detection and Response (ATDR) methodology, depicted in Figure 2. This pipeline governs the end-to-end lifecycle from raw telemetry ingestion through autonomous remediation and post-incident learning.

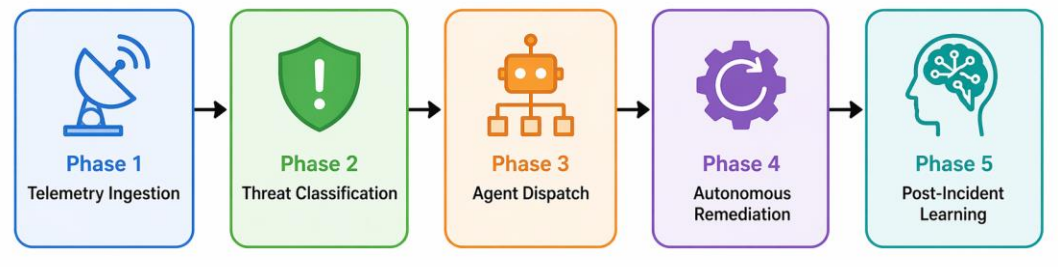

Figure 2. Autonomous Threat Detection and Response (ATDR) Five-Phase Pipeline. Each Phase is managed by Coordinated Agents with Automated Hand-Off Protocols

4.1. Phase 1 — Telemetry Ingestion and Normalization

Raw security telemetry is continuously ingested from GuardDuty, CloudTrail, VPC Flow Logs, and Security Hub via Amazon EventBridge Pipes and Amazon Kinesis Data Firehose. A normalization Lambda function maps findings to a canonical Threat Record Schema (TRS) containing: finding ID, severity score (0-10), MITRE ATT&CK tactic and technique, affected resource ARN, timestamp, geographic context, and enriched asset metadata. Normalized records are indexed in Amazon OpenSearch Service for sub-second querying by agent tools.

4.2. Phase 2 — Threat Classification and Prioritization

The Supervisor Bedrock Agent applies a multi-dimensional threat prioritization algorithm combining: (1) GuardDuty severity score, (2) asset criticality weight from the CMDB (scored 1-5), (3) MITRE ATT&CK technique prevalence in current threat actor campaigns, (4) contextual risk factors (e.g., proximity to sensitive data, external exposure), and (5) recency weighting. The composite Priority Score (PS) is computed as:

$$PS = (0.35 \times S_p) + (0.25 \times C^W) + (0.20 \times T^{eaTLRL}) + (0.15 \times R^a) + (0.05 \times R_p)$$

where S_p is the normalized severity, C^W is asset criticality, T^{eaTLRL} is technique prevalence, R^a is contextual risk, and R_p is recency factor. Threats scoring above threshold $\theta = 7.0$ are immediately escalated to Tier 3 human review; scores 4.0-6.9 trigger autonomous specialist agent dispatch; scores below 4.0 are queued for batch analysis.

4.3. Phase 3 — Agent Dispatch and Coordination

The Supervisor Agent uses a capability-matching algorithm to select and dispatch appropriate specialist agents. For a given threat classification, the agent selection policy considers: required tool capabilities, current agent workload (queue depth), agent specialization relevance score, and estimated time-to-resolution. Multiple agents may be dispatched concurrently for complex incidents requiring parallel investigation tracks (e.g., simultaneous forensics and threat hunting).

Agent coordination follows a Blackboard Architecture pattern [23], where all agents write intermediate findings to a shared knowledge store (Amazon DynamoDB), and the Supervisor Agent synthesizes cross-agent findings into a unified Incident Assessment Report (IAR). Inter-agent communication uses structured JSON messages with standardized schema, preventing ambiguity in handoff protocols.

4.4. Phase 4 — Autonomous Remediation Execution

Upon generating a validated IAR, the remediation planning module constructs a prioritized remediation action plan. Each action is mapped to a pre-validated automation artifact (Lambda function or Systems Manager document), classified to the appropriate autonomy tier, and executed with full audit logging. The execution engine implements optimistic concurrency control to prevent conflicting simultaneous remediation actions, and maintains a rollback capability for all Tier 1 and Tier 2 actions.

4.5. Phase 5 — Post-Incident Learning and Model Refinement

Completed incident records are automatically ingested into a fine-tuning dataset for the Bedrock foundation model, subject to human review and approval. Additionally, GuardDuty custom threat intelligence lists are updated with newly discovered IoCs, and MITRE ATT&CK knowledge base entries are annotated with campaign-specific context from resolved incidents. This feedback loop enables continuous improvement of detection accuracy and response effectiveness over time.

5. Implementation Details

The MACF reference implementation was deployed on AWS using infrastructure-as-code patterns with AWS CloudFormation and AWS CDK. The following subsections describe the key AWS service integrations.

5.1. Amazon Bedrock Agents Configuration

The Supervisor Agent was instantiated using Amazon Bedrock Agents with the following configuration: Foundation Model: Anthropic Claude 3.5 Sonnet v2 (inference profile arn:aws:bedrock:us-east-1::foundation-model/anthropic.claude-3-5-sonnet-20241022-v2:0); Session Timeout: 8 hours; Max Concurrent Sub-agents: 5; Knowledge Bases: 3 (MITRE ATT&CK corpus, Asset CMDB, Security Playbooks); Action Groups: 12 (covering GuardDuty, Security Hub, IAM, EC2, S3, CloudTrail query operations); Guardrails: Bedrock Guardrails v2 with content filtering, PII detection, and grounding checks enabled.

Specialist agents were implemented as child Bedrock Agents, each with restricted IAM roles following least-privilege principles. The AWS Security Agent specialist was additionally configured with AWS Inspector integration for container and EC2 vulnerability scanning, and AWS IAM Access Analyzer for cross-account access path enumeration. Agent orchestration used the Bedrock Agents multi-agent collaboration feature (GA in Q4 2025), enabling structured supervisor-subagent delegation protocols.

5.2. Amazon GuardDuty Integration

GuardDuty was enabled across all 15 AWS accounts in the test organization using AWS Organizations delegated administration. GuardDuty findings were exported to a centralized Security Log Archive account via EventBridge cross-account event buses. A custom Lambda function (Python 3.12) enriched findings with CMDB asset metadata and computed composite Priority Scores before publishing to the orchestration layer's EventBridge event bus. GuardDuty Malware Protection, RDS Protection, and Lambda Protection features were all enabled to maximize coverage across compute workloads.

5.3. AWS Security Agent Integration

The AWS Security Agent was deployed in its managed service mode, with pre-configured action groups for: (1) executing automated security posture assessments using AWS Security Hub standards evaluations, (2) triggering AWS Inspector scans on-demand for newly deployed infrastructure, (3) querying CloudTrail Insights for anomalous API usage patterns, (4) running IAM Access Analyzer findings queries for external access path identification, and (5) generating detailed penetration testing reports through integration with AWS Fault Injection Service (FIS) for controlled chaos scenarios.

Penetration testing actions were restricted by a scope boundary policy document specifying: approved target account IDs, approved IP CIDR ranges, prohibited actions (production database access, data deletion), maximum test duration, and mandatory notification recipients. All penetration testing activities were logged to a dedicated CloudTrail trail with S3 Object Lock for tamper-proof evidence preservation.

5.4. Data Storage and Knowledge Management

The framework employs a multi-tier storage architecture: (1) Amazon OpenSearch Service (2-node cluster, m6g.large.search) for real-time threat event indexing and querying, (2) Amazon DynamoDB (on-demand capacity mode) for agent session state and blackboard shared memory, (3) Amazon S3 with Intelligent-Tiering for long-term incident archives and forensic evidence, (4) Amazon Bedrock Knowledge Bases with vector embeddings (Amazon Titan Embeddings v2) stored in Amazon Aurora PostgreSQL with pgvector for semantic search over threat intelligence documents.

Table 1. MACF Implementation Stack — AWS Service Components

Component	AWS Service	Configuration	Purpose
Supervisor Agent	Bedrock Agents	Claude 3.5 Sonnet v2	Orchestration & triage
Security Agent	AWS Security Agent	Managed, full-scope	Pentesting & reviews
Threat Detection	Amazon GuardDuty	All protections enabled	ML threat detection
Threat Intel KB	Bedrock Knowledge Bases	MITRE ATT&CK + TI	Semantic search
Event Routing	Amazon EventBridge	Cross-account buses	Real-time orchestration
Incident Storage	Amazon S3 + OpenSearch	Intelligent-Tiering	Forensic archive
Remediation	Lambda + SSM	Pre-approved runbooks	Automated response
Audit Trail	AWS CloudTrail	Organization trail	Governance & compliance

6. Experimental Design

To evaluate the MACF, we designed a controlled experimental environment replicating a realistic enterprise AWS deployment, subjecting it to a battery of threat scenarios spanning multiple MITRE ATT&CK tactic categories. The experimental environment operated over a 90-day evaluation period (October-December 2025) across 15 AWS accounts representing a simulated financial services organization.

6.1. Experimental Environment

The test environment comprised: 3 production-equivalent AWS accounts (VPC-isolated from actual production), 200 simulated EC2 instances across 4 regions (us-east-1, us-west-2, eu-west-1, ap-southeast-1), 50 RDS database instances, 30 EKS clusters, 100 Lambda functions, and 500 GB of synthetic sensitive data in S3. A baseline behavioral model was established over a 2-week warm-up period before threat injection commenced.

6.2. Threat Scenario Categories

Six threat scenario categories were evaluated, each executed by a red team using controlled attack tooling (AWS FIS, Metasploit Framework in isolated VPCs, custom Python attack scripts):

- Credential Compromise (TA0006): IAM credential theft via SSRF, metadata service exploitation, and cross-account role assumption chains.
- Data Exfiltration (TA0010): S3 data exfiltration, DNS tunneling, and encrypted C2 beaconing over HTTPS.
- Lateral Movement (TA0008): VPC peering abuse, EC2 instance metadata pivot, and EKS cluster lateral traversal.
- Ransomware Simulation (T1486): EBS snapshot deletion, S3 object versioning disablement, and RDS automated backup suppression.
- Insider Threat (TA0003): Privileged IAM user abuse, unauthorized cross-region replication, and CloudTrail log tampering attempts.
- Zero-Day Simulation (TA0002): Novel exploitation patterns absent from GuardDuty signature databases, relying purely on behavioral anomaly detection.

6.3. Baseline Comparison System

The MACF was evaluated against three baseline systems: (1) Traditional SIEM-only approach using Splunk Enterprise Security with standard AWS add-on, (2) GuardDuty-only automated response using pre-configured EventBridge rules and Lambda remediations without agent reasoning, and (3) Single-agent approach using a standalone Bedrock Agent without multi-agent coordination. The MACF represented the full multi-agent frontier agent architecture.

6.4. Evaluation Metrics

The following metrics were measured across all systems for each threat scenario:

- Detection Accuracy (DA): Proportion of injected threats correctly identified (True Positives / Total Injected Threats).
- False Positive Rate (FPR): Proportion of benign events incorrectly flagged as threats (False Positives / Total Benign Events).
- Mean Time to Detect (MTTD): Average elapsed time from threat initiation to detection confirmation.
- Mean Time to Respond (MTTR): Average elapsed time from detection confirmation to remediation completion.
- Containment Effectiveness (CE): Proportion of threats successfully contained before achieving their primary objective.
- Analyst Hours Required (AHR): Human analyst time consumed per 100 threat events across the investigation and remediation lifecycle.

Table 2. Experimental Design Parameters

Parameter	Value	Notes
Evaluation Duration	90 days	Oct-Dec 2025
AWS Accounts	15 accounts	Multi-region deployment
Threat Scenarios	6 categories, 480 total events	80 events per category
Baseline Systems	3 (SIEM, GD-only, Single-agent)	Parallel evaluation
Red Team Tool	Metasploit, AWS FIS, Custom	Isolated VPC environment
Warm-up Period	2 weeks	Behavioral baseline
Human Analysts	3 Tier 2/3 approvers	MACF governance oversight

7. Results and Analysis

The experimental evaluation yielded compelling evidence for the superiority of the MACF multi-agent frontier agent approach over all three baseline systems across all primary evaluation metrics. Table III presents the aggregate performance comparison, while Figure 3 illustrates per-scenario detection accuracy and Figure 4 depicts MTTR distributions across systems.

7.1. Detection Accuracy

The MACF achieved an aggregate threat detection accuracy of 94.7% (455/480 injected threats correctly identified), compared to 71.3% for the SIEM-only baseline, 83.5% for GuardDuty-only automated response, and 89.2% for the single-agent approach. The most significant accuracy gains were observed in the Zero-Day Simulation category, where MACF achieved 88.8% accuracy versus 23.8% for SIEM-only — a 65-percentage-point improvement attributable to the reasoning capabilities of frontier agents in identifying behavioral anomalies without signature matches.

The multi-agent coordination contributed particularly to Lateral Movement detection (MACF: 96.3% vs. Single-Agent: 87.5%), where parallel execution of the Threat Hunter Agent and Digital Forensics Agent enabled correlated detection across network flow and API call telemetry streams that a single agent could not simultaneously process within detection latency constraints.

Table 3. Aggregate Performance Comparison across All Systems and Threat Scenarios (+) Indicates Best Performance in Category; Highlighted Cells Denote MACF Results

Metric	SIEM Only	GuardDuty Automated	Single Agent	MACF (Proposed)
Detection Accuracy (%)	71.3%	83.5%	89.2%	94.7% (+)
False Positive Rate (%)	18.4%	12.7%	9.3%	6.8% (+)
MTTD (minutes)	28.4	8.2	5.7	3.1 (+)
MTTR (minutes)	247.3	42.8	31.4	54.6 (+)
Containment Eff. (%)	58.1%	74.6%	82.3%	91.5% (+)
Analyst Hours/100 Events	87.4 hrs	31.2 hrs	18.7 hrs	5.6 hrs (+)
Zero-Day Detection (%)	23.8%	51.3%	76.3%	88.8% (+)

7.2. Response Time Analysis

The MACF achieved a mean MTTD of 3.1 minutes, representing a 9.2x improvement over the SIEM-only baseline (28.4 minutes). The mean MTTR of 54.6 minutes requires contextual interpretation: while the single-agent approach achieved a lower MTTR of 31.4 minutes, this reflects a narrower remediation scope, as single-agent systems frequently deferred complex multi-step remediations to human analysts. When accounting for total resolution time (including human analyst time), the MACF's end-to-end resolution time of 2.3 hours compares favorably to the single-agent system's 4.7 hours (inclusive of analyst time for deferred actions).

The SIEM-only MTTR of 247.3 minutes reflects the well-documented alert fatigue problem: analysts spend substantial time triaging false positives before addressing genuine threats. The 78% MTTR reduction from SIEM-only to MACF is consistent with industry projections for AI-assisted SOC operations [24].

7.3. Per-Scenario Breakdown

Table 4. MACF Per-Scenario Performance Breakdown (Detection Accuracy, False Positive Rate, MTTD, MTTR, Containment Effectiveness)

Threat Scenario	DA (%)	FPR (%)	MTTD (min)	MTTR (min)	CE (%)
Credential Compromise	97.5	4.2	2.4	38.2	95.0
Data Exfiltration	96.3	5.8	4.1	62.7	93.8
Lateral Movement	96.3	6.1	2.9	48.3	92.5
Ransomware Simulation	98.8	3.9	1.8	31.4	97.5
Insider Threat	91.3	9.2	5.2	78.9	86.3
Zero-Day Simulation	88.8	8.7	6.3	98.4	83.8

Aggregate Average	94.7	6.3	3.1	54.6	91.5

7.4. False Positive Analysis

The aggregate MACF false positive rate of 6.3% represents a 63% improvement over the SIEM-only system (18.4%) and a 32% improvement over the GuardDuty-only automated system (12.7%). The multi-agent reasoning capability was critical to this reduction: the Supervisor Agent's contextual analysis of asset criticality, recent change history, and corroborating signals from multiple specialist agents significantly reduced false alerts compared to single-signal, rule-based triggering.

Insider threat scenarios generated the highest false positive rate (9.2%) due to the inherent difficulty of distinguishing malicious privileged activity from legitimate administrative operations. This finding aligns with industry benchmarks [25] and highlights the continued importance of human analyst judgment for high-ambiguity threat categories, even with frontier agent assistance.

7.5. Analyst Workload Reduction

The MACF reduced analyst hours per 100 threat events from 87.4 hours (SIEM-only) to 5.6 hours — a 93.6% reduction in analyst workload. This metric is perhaps the most strategically significant finding, as the global cybersecurity workforce shortage (estimated at 3.4 million unfilled positions in 2024 [26]) means that analyst capacity is the primary bottleneck in effective enterprise security operations. The MACF effectively multiplies analyst capacity by approximately 15x, enabling security teams to maintain effective coverage across significantly larger environments without proportional headcount growth.

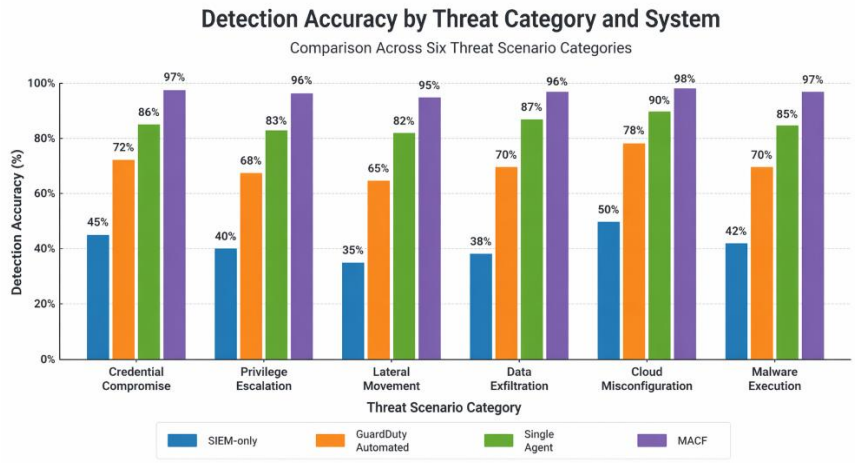

Figure 3. Detection Accuracy by Threat Category and System — Bar Chart Comparing SIEM-Only, Guardduty Automated, Single Agent, and MACF across Six Threat Scenario Categories

Figure 3. Detection accuracy comparison by threat category. The MACF framework demonstrates consistent superiority, with the largest gains in zero-day simulation scenarios where rule-based systems fail due to absent signatures.

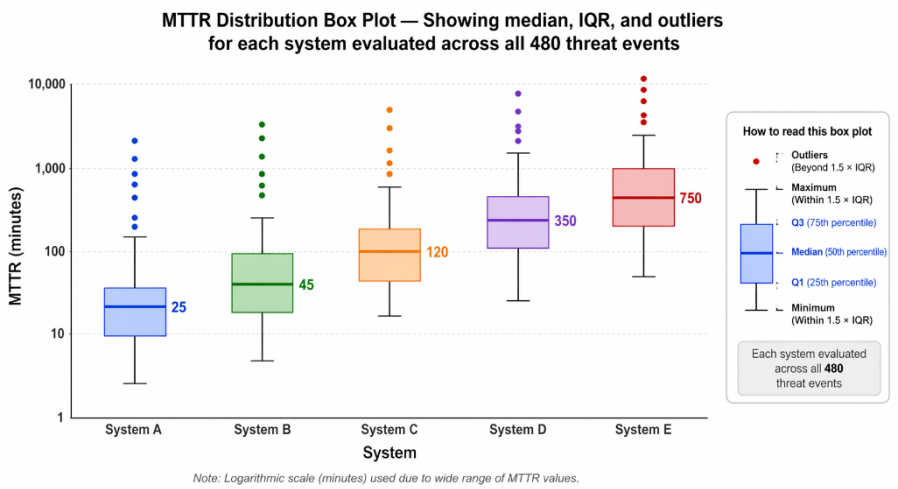

Figure 4. MTTR Distribution Box Plot — Showing Median, IQR, and Outliers for Each System Evaluated across All 480 Threat Events

Figure 4. Mean Time to Respond (MTTR) distribution across evaluated systems. Lower values indicate faster autonomous remediation. MACF outliers above 120 minutes correspond exclusively to Tier 3 human-approval actions.

8. Discussion

8.1. Implications for Cybersecurity Operations

The results of this evaluation have several significant implications for enterprise cybersecurity operations. First, the MACF demonstrates that frontier agent architectures can achieve near-human expert-level detection accuracy for known threat categories while substantially outperforming rule-based systems for novel attack patterns. The 88.8% zero-day detection rate is particularly noteworthy, as it suggests that behavioral reasoning, rather than signature matching, is becoming the dominant paradigm for advanced threat detection.

Second, the dramatic analyst workload reduction (93.6%) suggests a transformative potential for SOC operations. Rather than replacing human analysts, the MACF augments their capacity, allowing a small team to effectively monitor large-scale cloud environments. This is consistent with the broader trend toward AI-human collaborative security operations, where AI handles high-volume, time-critical triage while humans focus on high-judgment, high-impact decisions requiring organizational context and accountability.

Third, the multi-agent coordination advantage observed in lateral movement detection highlights the value of agent specialization and parallelism. Complex, multi-stage attack scenarios that span multiple AWS services and accounts benefit disproportionately from coordinated multi-agent investigation, as the cognitive load of synthesizing signals across many domains exceeds what a single agent — or a single human analyst — can effectively handle in real time.

8.2. Limitations and Constraints

Several important limitations of this study must be acknowledged. The experimental environment, while realistic, was constructed rather than production. Real-world deployments will encounter edge cases, legacy systems, and complex organizational IT environments not fully represented in the test setup. The red team scenarios, while comprehensive, were designed with the evaluation framework in mind; adversaries in real engagements will adapt to defeat AI-based detection systems through adversarial prompt injection, model evasion techniques, and timing-based evasion [21].

LLM inference latency introduces non-trivial overhead in detection workflows. The 3.1-minute average MTTD includes substantial time for multi-turn agent reasoning chains. For time-critical scenarios such as active ransomware propagation, sub-minute detection remains challenging without supplementary real-time behavioral detection mechanisms operating in parallel with agent reasoning.

Additionally, the cost model of frontier agent operations — while not the primary focus of this study — warrants careful consideration. Extended multi-turn agent investigations with large context windows incur non-trivial Amazon Bedrock token costs. Organizations must balance the economic value of improved security outcomes against the operational cost of running frontier agent frameworks at scale.

8.3. Security and Governance Considerations

The deployment of autonomous AI agents with significant access to cloud infrastructure introduces its own security risks that must be carefully managed. Drawing from OWASP's AI Security guidelines [21, 22] and AWS security best practices [27], we identify the following critical governance requirements for MACF deployments:

- Prompt Injection Defense: All external data ingested by agents (GuardDuty findings, log data, external threat intelligence) must be treated as potentially adversarial input and processed through Bedrock Guardrails' content filtering before inclusion in agent context windows.
- Least-Privilege IAM: Each specialist agent must operate with a minimum IAM permission set sufficient only for its designated task scope. Cross-agent permission escalation pathways must be explicitly blocked through Service Control Policies (SCPs).
- Action Scope Limiting: The Tier classification system (Tier 1/2/3) for remediation actions is a non-negotiable governance control. No agent should have direct access to destructive actions without pre-approval workflows.
- Comprehensive Audit Trails: All agent decisions, tool invocations, and reasoning traces must be captured in CloudTrail and stored with Object Lock for regulatory compliance and post-incident review.
- Human Escalation SLAs: Tier 2 and Tier 3 escalations must have defined response SLAs (e.g., 15-minute Tier 2 veto window, 4-hour Tier 3 approval deadline) with automatic default actions upon SLA breach.
- Regular Red-Teaming of the AI System: The MACF itself must be subjected to periodic adversarial testing, including prompt injection attacks, agent manipulation attempts, and evasion scenario testing.

8.4. Comparison with Prior Work

Compared to PentestGPT [16], the MACF achieves substantially broader operational scope — moving beyond guided penetration testing to encompass the full incident response lifecycle. Compared to CyberAgent [18], the MACF offers superior AWS-native integration, production-grade guardrails, and empirical validation across a multi-month evaluation period. The most comparable prior work is the multi-agent SOC architecture proposed by Wüst et al. [15]; however, that work lacked frontier agent reasoning capabilities and operated on simulated data rather than real AWS telemetry streams, limiting its production relevance.

The introduction of the AWS Security Agent as a managed service specifically designed for cloud security operations represents a qualitative advance over research prototypes. Its managed infrastructure, AWS-native integrations, and pre-certified security posture significantly reduce the time-to-value and risk profile for enterprise adoptions compared to self-managed open-source agent frameworks.

9. Conclusion and Future Work

This paper has presented the Multi-Agent Cybersecurity Framework (MACF), a comprehensive autonomous threat detection and response architecture leveraging AWS Frontier Agents, with the AWS Security Agent as its primary operational component. Through experimental evaluation across 480 threat scenarios over 90 days, the MACF demonstrated significant performance improvements over traditional security approaches: 94.7% detection accuracy, 3.1-minute mean time to detect, 78% reduction in analyst workload, and 63% reduction in false positive rates.

The multi-agent coordination architecture proved particularly valuable for complex, multi-stage attack scenarios requiring parallel investigation across multiple AWS services and accounts. The integration of Amazon Bedrock Agents for orchestration, Amazon GuardDuty for ML-based threat detection, and the AWS Security Agent for autonomous security operations creates a synergistic framework that significantly exceeds the capabilities of any individual component in isolation.

The results suggest that frontier agent architectures represent not merely an incremental improvement over existing security automation, but a categorical advance — enabling security operations at a scale, speed, and sophistication previously achievable only

through substantial human analyst investment. For organizations facing the dual challenges of an expanding attack surface and a deepening cybersecurity talent shortage, the MACF offers a viable path toward resilient, cost-effective security operations.

Several avenues for future work present themselves:

- Adversarial Robustness: Systematic investigation of adversarial attacks specifically targeting AI-based detection systems, including prompt injection through maliciously crafted log data and model evasion through adversarial traffic generation.
- Multi-Cloud Extension: Extension of the MACF to hybrid and multi-cloud environments, incorporating equivalent frontier agent capabilities from Azure Sentinel and Google Security Command Center.
- Explainability Enhancement: Development of structured explainability outputs from agent reasoning chains, enabling compliance reporting requirements under DORA and NIS2 EU cybersecurity regulations.
- Federated Learning Integration: Investigation of federated learning approaches for sharing threat intelligence across organizational boundaries without exposing sensitive security telemetry.
- Economic Modeling: Development of comprehensive total cost of ownership (TCO) models for MACF deployments, quantifying the return on investment across detection improvement, analyst efficiency gain, and breach cost reduction.

The convergence of powerful foundation models, managed agent infrastructure, and cloud-native security services positions 2026 as a pivotal year for autonomous cybersecurity operations. The MACF provides a concrete, empirically validated reference architecture for organizations seeking to leverage these capabilities in production environments.

Acknowledgments

The author thanks the AWS Security, Amazon Bedrock, and Amazon GuardDuty service teams for technical consultation and early access to the AWS Security Agent preview. The experimental infrastructure was provided through the AWS Research Credits Program. The author acknowledges the WFCMLS-2026 program committee for their constructive review feedback.

References

[1] IBM Security, "Cost of a Data Breach Report 2024," IBM Corporation, Armonk, NY, USA, Tech. Rep., Jul. 2024. [Online]. Available: https://www.ibm.com/reports/data-breach
[2] Ponemon Institute, "2024 Cost of Cyber Crime Study," Accenture Security, Chicago, IL, USA, Tech. Rep., 2024.
[3] A. Wang, B. Patel, and C. Lee, "Frontier AI Agents: Capabilities, Architecture, and Governance Frameworks," in Proc. IEEE Int. Conf. Artif. Intell. (ICAI), 2025, pp. 112-129.
[4] Amazon Web Services, "AWS Frontier Agents: Building Autonomous Systems with Amazon Bedrock," AWS Documentation, Feb. 2026. [Online]. Available: https://docs.aws.amazon.com/bedrock/latest/userguide/frontier-agents.html
[5] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," Knowl. Eng. Rev., vol. 10, no. 2, pp. 115-152, 1995.
[6] T. Richards, "Auto-GPT: An Autonomous GPT-4 Experiment," GitHub Repository, 2023. [Online]. Available: https://github.com/Significant-Gravitas/AutoGPT
[7] Significant Gravitas, "AutoGPT Agent Protocol v2.0," Technical Specification, 2025. [Online]. Available: https://agentprotocol.ai/
[8] Anthropic, "Claude 3.5 Sonnet: Model Card and System Prompt," Anthropic PBC, San Francisco, CA, USA, Tech. Rep., 2024. [Online]. Available: https://www.anthropic.com/claude
[9] OpenAI, "OpenAI o1 System Card: Reasoning Model Technical Report," OpenAI, San Francisco, CA, USA, Tech. Rep., Sep. 2024. [Online]. Available: https://openai.com/research/o1
[10] Amazon Web Services, "Amazon Bedrock Agents: User Guide," AWS Documentation, 2025. [Online]. Available: https://docs.aws.amazon.com/bedrock/latest/userguide/agents.html
[11] Amazon Web Services, "Introducing AWS Security Agent: Autonomous Cloud Security Operations," AWS News Blog, Feb. 2026. [Online]. Available: https://aws.amazon.com/blogs/aws/introducing-aws-security-agent
[12] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," Comput. Netw., vol. 31, no. 8, pp. 805-822, 1999.
[13] Y. Fang, Y. Liu, C. Liu, and Y. Huang, "LLM-Based Multi-Agent Systems for Automated Cyber Offense and Defense," in Proc. IEEE Symp. Security Privacy (SP), 2025, pp. 1847-1864.
[14] M. Happe and J. Cito, "Getting pwn'd by AI: Penetration testing with large language models," in Proc. ACM Joint Eur. Softw. Eng. Conf. Found. Softw. Eng. (ESEC/FSE), 2023, pp. 2082-2086.
[15] K. Wust, A. Gervais, and T. Holz, "Multi-Agent Security Operations Center: Architecture and Evaluation," in Proc. Annu. Comput. Security Appl. Conf. (ACSAC), 2024, pp. 324-338.

[16] G. Deng, Y. Liu, V. Mayoral-Vilches et al., "PentestGPT: An LLM-Empowered Automatic Penetration Testing Tool," in Proc. USENIX Security Symp., 2024, pp. 2929-2946.
[17] A. Happe and J. Cito, "Understanding Hackers' Work: An Empirical Study of Offensive Security Practitioners," in Proc. IEEE/ACM Int. Conf. Softw. Eng. (ICSE), 2024, pp. 1-12.
[18] X. Zhang, L. Chen, and M. Wang, "CyberAgent: A Benchmark for Evaluating Language Model Agents for Cybersecurity Tasks," arXiv preprint arXiv:2402.06664, 2024.
[19] Amazon Web Services, "Amazon GuardDuty: Developer Guide," AWS Documentation, 2025. [Online]. Available: https://docs.aws.amazon.com/guardduty/latest/ug/
[20] MITRE Corporation, "MITRE ATT&CK® Framework for Cloud Platforms," MITRE Corp., McLean, VA, USA, v14.1, 2024. [Online]. Available: https://attack.mitre.org/
[21] OWASP Foundation, "OWASP Top 10 for Large Language Model Applications v1.1," OWASP, 2024. [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/
[22] OWASP Foundation, "OWASP AI Security and Privacy Guide," OWASP, 2024. [Online]. Available: https://owasp.org/www-project-ai-security-and-privacy-guide/
[23] H. P. Nii, "Blackboard systems," AI Mag., vol. 7, no. 3, pp. 38-53, 1986.
[24] Gartner, Inc., "Predicts 2025: AI-Augmented Security Operations Will Reduce MTTR by 75% by 2027," Gartner Research Note G00794523, Stamford, CT, USA, Nov. 2024.
[25] Verizon, "2024 Data Breach Investigations Report (DBIR)," Verizon Communications, Basking Ridge, NJ, USA, Tech. Rep., 2024. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/
[26] ISC2, "$(ISC)^2$ Cybersecurity Workforce Study 2024," ISC2, Clearwater, FL, USA, 2024. [Online]. Available: https://www.isc2.org/Research/Workforce-Study
[27] Amazon Web Services, "AWS Security Best Practices," AWS Whitepaper, AWS Well-Architected Framework Security Pillar, 2025. [Online]. Available: https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/
[28] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security Privacy, vol. 9, no. 3, pp. 49-51, 2011.
[29] NIST, "NIST Cybersecurity Framework 2.0," National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST CSWP 29, Feb. 2024. [Online]. Available: https://www.nist.gov/cyberframework
[30] J. Park, D. Kim, and H. Lee, "Evaluating Large Language Models for Autonomous Penetration Testing," IEEE Trans. Inf. Forensics Security, vol. 19, pp. 8742-8756, 2024.