Agent Behavior Analytics: Securing the Autonomous Enterprise

Understanding how Exabeam detects, monitors, and investigates AI agent activity

Abstract

Artificial intelligence adoption is accelerating, and autonomous agents are moving from experimental tools into operational digital workers capable of executing complex tasks at scale. These systems interact with applications, invoke APIs, and perform actions within workflows, often independently and at machine speed. Traditional security models, designed for human identities, lack the visibility and behavioral context required to detect the risks these agents introduce.

Exabeam extends behavioral analytics to this emerging class of entities. By combining established user and entity behavior analytics (UEBA) with Agent Behavior Analytics (ABA), Exabeam detects activity from agents deployed as external services, integrated workflows, or fully embedded digital workers. UEBA identifies behavioral anomalies, credential misuse, and policy violations. ABA introduces detections for agent-specific behaviors such as prompt injection, guardrail violations, anomalous tool invocation sequences, and autonomous decision patterns.

Together, these capabilities provide visibility into how agents operate, how they interact with systems and data, and whether their behavior deviates from established baselines. Exabeam reconstructs activity throughout the full execution path, from user to agent to downstream systems, preserving chain of custody and supporting confident investigation.

This paper defines ABA within Exabeam, outlines the evolving role of agents, and explains how behavioral detection applies to current deployments, near-term capabilities, and the shift toward an autonomous workforce.

Agent Visibility Gap:Traditional vs Agent Reality

Traditional Security Model	Agent Reality
User → Application → Logs **Clear, direct attribution to a single user identity**	**User** **↓** **Agent** **↓** **Multiple systems (APIs, tools, credentials)** **Attribution lost across systems** **Fragmented visibility and identity**

Figure 1.

Identity attribution breaks down once agents execute across systems.

The Problem: Agent Visibility Gaps

Traditional security controls rely on a core assumption: All activity can be attributed to a named user. Firewalls monitor user traffic, identity systems authenticate users, and SIEM platforms baseline user behavior to detect anomalies. This model has been effective for decades. In agent-driven environments, it no longer applies.

Agents introduce structural gaps that existing models were not designed to handle. They do not appear as distinct identities. Instead, they operate with inherited credentials such as service accounts, API keys, or user tokens. Security tools interpret activity as originating from familiar identities, while the agent itself remains hidden.

This abstraction breaks attribution. As agents execute tasks, actions are recorded under whichever credential is used at each step rather than a unified entity. Logs capture what occurred but not which agent orchestrated the behavior, making it difficult to determine what performed an action.

Lifecycle dynamics further complicate detection. Many agents are short-lived, created for specific operations and then terminated. These ephemeral instances do not persist long enough to establish behavioral baselines, reducing the effectiveness of anomaly detection models that rely on historical context.

At the same time, not all agents are visible to security teams. Developers can deploy agents without formal governance, creating shadow agents that operate outside standard controls. These systems may interact with internal resources using delegated credentials while remaining outside monitoring frameworks.

Speed and scale amplify these challenges. Unlike human users, agents can execute large volumes of actions in rapid succession. Behavioral deviations can escalate quickly, and by the time anomalies are detected, significant impact may have already occurred.

Addressing Agent Visibility Gaps With Behavioral Analytics

ABA extends detection beyond identity-based attribution by treating agents as distinct actors. Rather than relying solely on usernames or credentials, Exabeam analyzes activity spanning systems, identities, and workflows to identify patterns that indicate risk.

This includes detecting prompt injection, guardrail violations, anomalous tool usage, and misuse of delegated credentials. By correlating execution activity end to end, Exabeam reconstructs chain of custody, from user to agent to downstream systems, allowing security teams to understand how actions were performed and where risk emerged.

Detection Challenges With Static Rules

Traditional detection models rely on deterministic rules built around expected sequences of activity, such as unusual logins or privilege escalation. These approaches assume behavior is predictable and deviations can be defined in advance.

Agent-driven workflows break this assumption. Agents dynamically reason through tasks, invoke tools, and adapt execution paths based on intermediate outcomes. Two executions of the same workflow may produce completely different sequences of actions.

As adoption grows, the number of potential execution paths expands rapidly. Maintaining rule coverage becomes complex and incomplete. Behavioral analytics addresses this challenge by shifting detection from predefined events to observed patterns of activity.

Instead of asking whether a specific event occurred, detection evaluates whether behavior diverges from expected patterns within a workflow. The focus shifts from isolated events to how activity evolves through execution.

Defining Agents: A Three-Tier Model

AI agents operate along a spectrum of integration and autonomy. Their execution model and level of system access determine both the attack surface and detection approach. Exabeam defines a three-tier model to describe this range, with each tier introducing its own set of challenges.

Three-Tier Agent Framework

Visibility complexity and detection

Tier 1	Tier 2	Tier 3
Enterprise Chatbots	**Semi-Autonomous Agents**	**Digital Workers**
Entity Model **User behavior in agent interactions**	**Entity Model** **Agent behavior across credential chains**	**Entity Model** **Agent as formal user**
Detection Focus • **Token anomalies** • **Guardrail violations** • **Unusual tool invocation**	**Detection Focus** • **Task execution anomalies** • **Scope violations** • **Credential inconsistency**	**Detection Focus** • **Full UEBA + AI rules** • **Baseline deviation** • **Lateral movement + exfiltration**
Examples: ChatGPT, Gemini, Copilot	Examples: Claude Code, custom agents, workflow tools	Examples: OpenClaw, Devin, identity-based workers

User With Agents → Agent Task Execution → Agent As User →

Figure 2.

Three-tier framework for modeling agent behavior and defining detection strategy

Tier 1: Isolated Agents

Chatbots like ChatGPT, Google Gemini, and Microsoft Copilot operate as external services on vendor-managed infrastructure and are isolated from enterprise systems by default. Integration into workflows occurs only when organizations explicitly connect these platforms to internal systems through APIs, webhooks, browser extensions, or other interfaces using provisioned credentials.

User Identity and Log Consistency Challenge

The representation of user identity in Tier 1 activity varies significantly by platform. OpenAI ChatGPT Compliance API includes user email and user ID within conversation logs, enabling clear attribution. In contrast, Google Gemini Enterprise, when collected through pub/sub, does not consistently expose user identity in conversation data by default. Other platforms follow different logging patterns with varying levels of identity fidelity and context.

This inconsistency creates a fundamental challenge for detection. Security teams cannot rely on a uniform representation of user behavior and must implement platform-specific parsing, normalization, and correlation strategies to establish reliable attribution.

Compromise and Detection

If an attacker compromises a Tier 1 agent session, for example by obtaining a session token or API key, the impact is defined by the scope of access granted to that credential. Because interactions occur within trusted user context, malicious activity can leverage legitimate access paths to retrieve data, invoke tools, or execute workflows.

Exabeam detects compromised Tier 1 interactions through behavioral analysis of user activity within agent platforms. This includes identifying deviations such as unusual token consumption, spikes in prompt volume, repeated guardrail violations, and anomalous tool invocation behavior. Detection focuses on deviations in user interaction patterns, indicating potential misuse or session compromise. This model applies to ChatGPT Enterprise conversations, Google Gemini interactions, and similar external AI usage where user activity is the primary detection surface.

Detection Focus

Detection centers on user-initiated interaction patterns rather than the agent itself. Key signals include guardrail violations, prompt injection attempts, unusual token usage, anomalous tool invocation patterns, and changes in user behavior, such as first-time access or unexpected query patterns.

Effectiveness depends on the fidelity of available telemetry. Platforms that expose richer interaction data support stronger behavioral correlation, while limited visibility constrains detection depth.

Tier 2: Integrated Service Agents and Task Harnesses

Tier 2 agents operate within enterprise systems and execute tasks spanning multiple applications, APIs, and data sources using enterprise credentials. These include task agents and harnesses such as Claude Code, Model Context Protocol (MCP) integrations, self-hosted or open-source models like Ollama, and custom LLM-powered workflows.

These agents are deployed on enterprise infrastructure, including cloud platforms, on-premises systems, CI/CD pipelines, and application layers. They have direct access to enterprise resources such as APIs, databases, file storage, systems and identity services.

Unlike user-driven conversational agents, these systems execute predefined business functions. They review code, generate and update documents, process tickets, automate workflows, and orchestrate multi-step operations across systems. In many cases, they are shared resources with autonomous or semi-autonomous execution.

These agents often authenticate using service account credentials, may be invoked by multiple users, and frequently function with task-specific permissions that are not linked to a single user identity. This execution model allows broad operational reach but complicates attribution, as activity is distributed across multiple credentials and systems rather than a single observable entity.

Identity and Attribution Challenge

From a detection perspective, Tier 2 introduces fragmented visibility. Agent activity is distributed across multiple credentials and systems during execution. A single workflow may authenticate to one system using an API key, interact with another using a service account, and access additional resources through separate methods.

As a result, actions appear as separate, unrelated entities rather than as part of a cohesive execution. Without explicit instrumentation or identity normalization, tracking agent behavior end to end is difficult. Detection systems may identify anomalies within individual credentials or systems, but correlating those signals into a complete view introduces complexity.

Compromise and Detection

If a Tier 2 agent or its credentials are compromised, attackers can exploit its ability to operate within systems and execute workflows with legitimate access. Because activity is distributed across identities, malicious behavior may appear normal unless evaluated in context.

The Exabeam detects these scenarios by analyzing workflow execution patterns, identifying anomalies in tool and API sequences, and correlating signals throughout systems to reconstruct agent activity. Detection also incorporates attribution back to the originating user or system when relationships can be established.

Effective detection in this tier depends on correlating activity across fragmented signals. Exabeam supports this through telemetry investigation and is developing agent-specific instrumentation that tags operations with agent identity, enabling attribution beyond user or service account context.

Detection Focus

Detection centers on behavioral analysis of task execution across systems and identities. Key signals include deviations in workflow patterns, credential usage inconsistencies, unauthorized tool invocation, and unexpected data access. Models focus on linking API calls, credentials, and system activity back to originating behavior.

Unlike Tier 1, detection evaluates how work is performed through execution sequences, making cross-system correlation essential.

Examples

Claude Code supports code review and generation, while custom LLM-powered task orchestrators handle document and ticket processing. Workflow agents are also used for data transformation and automation pipelines. These implementations represent common Tier 2 deployment patterns.

Tier 3: Fully Embedded Agents (Digital Workers)

Digital workers such as OpenClaw, Hermes, and Devin operate fully integrated identities within enterprise identity systems. They are assigned credentials, permissions, and authentication capabilities equivalent to human users. These agents often handle operational responsibilities such as infrastructure management, financial transactions, and vendor integrations, executing tasks independently without continuous human input.

From a detection perspective, Tier 3 agents are modeled as first-class identities. Their activity appears in logs the same form as human activity, including authentication events, system access, and workflow execution. These agents often handle operational responsibilities such as infrastructure management, financial transactions, and vendor integrations, executing tasks independently without continuous human input.

Compromise and Detection

Exabeam detects Tier 3 activity using the full UEBA rule library, establishing behavioral baselines and identifying deviations in access, data usage, and privilege activity. Because digital workers operate as standard identities, a compromise carries the same risk as a user account, including abnormal access, privilege escalation, lateral movement, and data exfiltration.

Exabeam is also expanding coverage to include lifecycle events such as agent creation, assignment, and termination, as well as multi-step attack patterns where users create or modify agents to escalate access or perform unauthorized actions. Ongoing work includes techniques to distinguish human and agent behavior, enabling identification of patterns specific to autonomous systems.

Detection Focus

Detection applies behavioral analytics directly to agent identities. Signals include baseline deviations, anomalous system or data access, unexpected resource usage, and correlated activity patterns. Consistent identity representation allows full behavioral correlation and high-confidence detection.

Examples

OpenClaw, Hermes, Devin, and similar digital workers are integrated into identity systems and operate as user accounts with assigned credentials, permissions, and persistent access to enterprise services.

Detection Coverage by Tier

Tier 1	Tier 2	Tier 3
User-Centric AI	**Agent & Credential Layer**	**Full Identity Coverage**
Primary Signal • User behavior • Token usage • Guardrail signals	**Primary Signal** • Task execution • Credential usage • Cross-system activity	**Primary Signal** • Full UEBA behavior • Baseline anomalies • AI + behavioral correlation
Coverage Limited to user + AI interaction layer	**Coverage** Fragmented visibility (identity gap)	**Coverage** Complete coverage with UEBA + AI layering

Figure 3.

Detection capability varies based on how agents are deployed and represented within enterprise environments.

Implications for Detection Strategy

The three-tier model organizes agents by integration and autonomy, directly shaping both the attack surface and the detection approach. As agents progress from external interaction to internal workflow execution and full identity integration, detection shifts from user-centric monitoring to behavioral analysis of identities, workflows, and systems.

Tier 1 detection is limited to user interactions with external agents. Tier 2 expands visibility to workflow execution but introduces fragmentation across identities and credentials. Tier 3 enables complete behavioral detection by representing agents as unified identities and applying full UEBA coverage.

Across all tiers, the guiding principle remains consistent: Detection must focus on behavior. In agent-driven environments, behavior becomes the primary detection surface.

Identity and Tracking: How Exabeam Sees Agents

The foundational challenge is that agents are not consistently represented as first-class identities. Their activity is attributed to parent users or inherited service accounts, creating ambiguity.

Exabeam addresses this through telemetry ingestion, normalization, and behavioral correlation to reconstruct agent activity across systems and identity contexts.

Cloud LLM Platforms

Exabeam provides native telemetry ingestion for major cloud-based LLM platforms, delivering visibility into user interaction with external agents.

For ChatGPT, the Compliance Platform API exposes detailed telemetry, including conversation logs, model usage, token counts, and tool invocations. The native Exabeam collector ingests this data through the REST API, normalizes it into the Common Information Model (CIM), and applies behavioral analytics to identify anomalous patterns in user and agent activity.

For Google Gemini, agent activity is surfaced through a pub/sub messaging interface. Exabeam connects to these streams, ingests events in near real time, and applies the same behavioral analytics used for traditional user activity.

Microsoft Copilot presents a different challenge. Copilot does not currently provide a comprehensive telemetry API for full activity visibility. Exabeam can surface detections for custom agents developed in Copilot Studio, but native Copilot activity remains only partially observable.

Emerging capabilities such as Microsoft Entra Agent ID, currently in preview, are expected to improve visibility by exposing identity, authentication, and governance signals for agent activity. These signals will strengthen attribution and detection as the ecosystem evolves.

Custom Agents and Task-Driven Agents

Custom agents and task-driven systems often lack standardized logging, making visibility into their behavior significantly more difficult. These agents are typically built using diverse frameworks and deployed within internal workflows, resulting in inconsistent telemetry and limited observability into execution.

To address this, Exabeam is developing agent telemetry libraries designed to capture the full execution context of agent activity. These libraries instrument the agent decision chain, collecting detailed information that includes prompts and reasoning steps, tool invocations and function calls, API usage and associated credentials, MCP integrations, token consumption, and execution outcomes.

This level of instrumentation provides visibility not only into what actions were performed, but also how decisions were made and how execution progressed. By capturing both inputs and outcomes, these telemetry streams support deeper behavioral analysis and improve detection of anomalous or unsafe agent behavior.

These capabilities are currently in active development. When available, organizations will be able to deploy telemetry libraries and sidecar components to instrument custom agents directly. This instrumentation generates standardized, CIM-aligned logs that feed into the Exabeam detection engine, supporting consistent behavioral analysis for agent activity that would otherwise lack native observability.

Digital Workers

The most effective model for agent detection is to onboard agents as formal user identities within enterprise identity systems such as Active Directory (AD) or Okta. In this model, digital workers are assigned credentials and operate as user entities, allowing behavioral detection to apply directly.

Identity consistency is a key advantage, allowing existing UEBA models to operate without modification. Behavioral baselining, anomaly detection, and cross-system correlation function in the same way as they do for human users, providing complete visibility into agent-driven activity.

This model is not practical for every agent type, particularly in Tier 2 scenarios where agents operate using multiple credentials. However, for Tier 3 digital workers, and in any case where credential standardization is possible, it provides the strongest foundation for detection.

The effectiveness of this approach is straightforward. The closer agent behavior aligns with established identity constructs, the more effectively existing detection infrastructure can be applied.

Log Sources and Ingestion

Exabeam reconstructs agent activity by ingesting telemetry from multiple data sources that capture both interaction signals and execution behavior.

These sources include cloud platform audit logs such as AWS CloudTrail, Azure Activity Logs, and Google Cloud audit logs, which record API activity and resource access. Identity logs from systems like Okta and Azure AD provide visibility into authentication events, credential usage, and service account behavior associated with agents.

AI platform telemetry, including ChatGPT Compliance logs, OpenAI API activity, Google Gemini audit trails, and logs from platforms such as Claude Code, captures conversations, token usage, and tool interactions. HTTP and API gateway logs provide additional context, where user-agent strings, API keys, and request patterns can be used to infer agent behavior and correlate execution across services. Application-level logs from workflow platforms, RPA systems, and custom applications capture agent-driven execution within business processes.

By normalizing and correlating these data sources into the CIM, Exabeam reconstructs agent activity spanning systems, identities, and workflows. This allows detection models to evaluate complete execution sequences rather than isolated events.

Agent Lifecycle and Attribution: Chain of Custody

Preserving chain of custody is critical for understanding and investigating agent-driven activity. When an agent executes an action, security teams must be able to trace the full execution path through multiple layers:

user → agent → tool → downstream system

This requires visibility not only into individual events, but into how actions propagate through systems, identities, and workflows. Without this context, attribution is incomplete and the origin of activity remains unclear.

Agent Creation and Interaction

For cloud LLMs, lifecycle events are exposed as discrete, auditable activities. These include agent creation, modification, sharing, invocation, as well as interaction events such as token usage, guardrail violations, and conversation activity. Each action generates distinct log entries that can be ingested and analyzed.

Exabeam ingests these events through native collectors and applies behavioral analytics immediately, enabling detection of abnormal patterns in how agents are created, configured, and used. This forms the initial layer of visibility into how agents enter and operate within the environment.

Agent Invocation and Tool Execution

Once deployed, agents are invoked to execute tasks. These tasks typically involve calling tools, interacting with APIs, querying databases, or triggering workflows that span multiple systems. Each step generates logs at the system level.

Where available, correlation identifiers such as session IDs, conversation IDs, and request IDs can link these actions back to the originating agent.

In native platforms like ChatGPT and Google Gemini, APIs expose agent or conversation identifiers that allow events to be correlated across interactions.

For custom agents, correlation depends on instrumentation. Without it, activity appears as isolated system events. Exabeam agent telemetry libraries, currently in development, standardize correlation by tagging execution with agent identifiers.

For digital workers onboarded as users, execution is inherently tied to a single identity. Tool calls and API activity are logged under that entity, provided credential usage remains consistent.

The completeness of this correlation determines how effectively agent behavior can be reconstructed from individual system events.

Credential Consistency and Entity Correlation

The effectiveness of the agent-as-user model depends on consistent identity representation. In practice, this consistency often breaks down.

Agents frequently operate across multiple platforms using different credential types. A single agent may call a code repository using an API key, access a database through a service account, authenticate to cloud APIs using OAuth tokens, and interact with internal tools using additional credentials. Each action is recorded under a different identity.

For example, a GitHub API call may appear as "API_KEY_12345," a database login as "svc-agent-bot," and an outbound email as "agent.service@company.com." From the perspective of the detection engine, these appear as separate entities and cannot be reliably linked without additional context.

This results in fragmented visibility. Individual detections may trigger on specific credentials or systems, but the unified view of agent behavior is incomplete.

Exabeam addresses this challenge in stages. First, telemetry normalization improves visibility into what the agent did. Agent telemetry libraries and sidecar instrumentation convert disparate event formats into standardized, CIM-aligned data streams that can be tagged with agent identifiers, supporting correlation.

However, this does not fully unify the underlying credential identities. Linking multiple credentials back to a single agent remains an area of active research and development. Current efforts focus on extending telemetry to include agent identifiers, credential metadata, and API key references, allowing reconstruction of a unified entity.

In the interim, organizations can improve attribution by standardizing credential usage, instrumenting agent telemetry, and using identity platforms such as Okta or Azure AD to track relationships between agents and credentials.

Agent-as-User Model and Relationship Tracking

When agents are modeled as users within Exabeam, two capabilities become critical for effective attribution.

The first is establishing relationships between users and agents. Agents are often created, configured, or invoked by specific individuals or systems. Identity platforms such as Microsoft Entra, Okta, and AD can record these relationships through agent creation and assignment events. Exabeam is exploring integrations to ingest this data and build relationship graphs, for example linking "User A created Agent X" or "User B invoked Agent X." This provides provenance during investigation.

The second capability is resolving credential fragmentation. Without consistent identity mapping, agents continue to appear as multiple independent entities. Exabeam is evaluating approaches that extend telemetry to associate credentials with agent identifiers and formalize agent entity types within the Attack Surface Insights (ASI) entity management system. This would allow agents to be recognized and tracked alongside users and devices as first-class entities.

For Tier 3 digital workers that are fully integrated into identity systems, these challenges are significantly reduced. A single authoritative identity provides consistent attribution throughout execution. For Tier 2 agents operating with multiple credentials and no formal identity representation, attribution remains incomplete, making this a primary focus of ongoing development.

Baseline, Detection, and Noise Management

UEBA establishes a baseline of normal behavior for each entity and identifies deviations from that baseline. This approach is effective when identity representation is consistent and activity is sustained. In agent driven environments, these conditions are not always met, introducing challenges for baseline-dependent detection.

Baseline Maturity

UEBA builds a behavioral baseline using observed activity for each entity. This baseline represents patterns, such as access behavior, authentication activity, and resource usage that are leanred before anomaly-based detection rules are allowed to trigger.

Baseline formation depends on historical training data and entity maturity requirements. In most cases, baselines are established over roughly 14 days of activity, although requirements vary by detection rule. Some rules do not require maturity and can trigger immediately. Others require extended observation periods, such as 7, 14, or 28 days, to ensure deviations are evaluated against a stable behavioral profile and to reduce false positives.

Once an entity reaches maturity, anomaly-based rules begin evaluating activity against the established baseline. Deviations trigger alerts that indicate behavior inconsistent with the learned profile.

For digital workers that are onboarded as user identities, this process mirrors that of human users. During the baseline development period, anomaly-based detections that depend on maturity do not trigger. Once the baseline is established, the full UEBA rule library applies without modification. This is a calibration phase required to establish reliable detection thresholds rather than a limitation of the model.

Short-Lived Agents and Detection Gaps

Agents lifecycle characteristics introduce a key limitation: Many agents are short-lived.

Task-specific agents may run for hours or days and terminate before sufficient activity is collected. In these cases, the entity never reaches baseline maturity, and detections that require maturity will not trigger.

Detection still occurs through rules that do not depend on learned behavior, but overall coverage is reduced. This creates a known gap for ephemeral agents, where limited behavioral context constrains evaluation of deviations.

This limitation is innate to any detection model based on historical baselining and highlights the need for complementary approaches that do not rely on sustained identity activity.

Detection Categories

Exabeam identifies agent-related risk through a combination of behavioral and rule-based detections that evaluate identity activity, task execution, and workflow behavior.

Behavioral anomaly detection focuses on deviations from established patterns, including spikes in token consumption, abnormal API call rates, unusual tool invocation sequences, and indicators of data exfiltration. These signals reflect changes in how agents or users interact with systems relative to learned behavior.

In addition to behavioral anomalies, Exabeam detects prompt injection and model abuse attempts, including jailbreak techniques, malicious prompt structures, attempts to expose system prompts, and inputs designed to bypass model constraints.

Guardrail violations are another class of detections, identifying breakdowns in platform-level safety controls such as those enforced by systems like Google Model Armor or similar mechanisms.

Policy and privilege violations are also monitored, including the use of elevated credentials, access to data outside defined scope, and invocation of unauthorized tools.

Traditional Insider threat indicators remain highly relevant in agent-driven environments. Thes include bulk data access, persistence changes, and exfiltration patterns that may indicate misuse or compromise of agent capabilities.

Observability and Visibility Gaps

Exabeam provides extensive visibility into agent activity, but detection capability depends on available telemetry. Understanding where signals exists and where they do not is essential for evaluating coverage.

Minimum Observability Surface

Detection requires at least one observable signal. This may include platform telemetry such as logs from ChatGPT, Copilot, or Gemini APIs; authentication events such as service account logins or API key usage; downstream system logs generated by agent activity; or proxy or gateway logs that capture request patterns, user-agent strings, or API signatures.

If no such signal exists, for example in fully isolated or air-gapped systems, agent activity cannot be observed or analyzed.

Known Visibility Gaps and Research Areas

Despite broad coverage, several gaps remain.

On-premises LLM deployments, including self-hosted models such as Ollama or Llama-based systems, are difficult to instrument due to limited native telemetry. Work is ongoing to address this through API gateway wrappers and custom instrumentation approaches.

Custom internal RAG systems present similar challenges. These implementations often lack standardized logging and require bespoke ingestion pipelines.

Agents operating in isolated or air-gapped systems may generate no observable signals if they do not interact with monitored services. These scenarios require explicit logging or telemetry injection to enable detection.

Agents that use multiple credentials across systems remain a persistent challenge. Without consistent identity mapping or instrumentation, activity appears as fragmented entities instead of a unified behavioral model. Addressing this requires credential standardization or explicit agent telemetry to allow correlation.

These areas remain active research and development priorities and are central to expanding detection coverage in agent-driven environments.

Current Detections and Real-World Examples

Exabeam introduced a core set of AI-specific behavioral and factual detections in Q1 2026. These detections are delivered through Threat Detection Management and continue to expand with each quarterly release. In addition, hundreds of existing UEBA rules apply directly to agent activity when agents are modeled as identities within the environment.

Sample AI-Specific Detections

Exabeam provides a growing library of agent-specific detections that extend traditional behavioral analytics. These detections identify both early-stage activity and high-risk behaviors in agent interactions.

Examples include:

- First successful invocation of an AI agent tool for a user, establishing a behavioral baseline
- Abnormal volume of AI requests for a user or organization, indicating a velocity anomaly
- Abnormal aggregate token usage across AI interactions, indicating potential misuse
- Detection of guardrail violations enforced by AI platforms
- Prompt injection attempts designed to manipulate agent behavior
- Attempts to expose system prompts or underlying model instructions
- First-time creation of an AI agent by a user, indicating a new entity introduction
- First-time sharing of an AI agent by a user, indicating expansion of access
- Initial attempts to execute scripts or commands through AI interactions, indicating execution risk

These detections operate within a broader behavioral model. Individual signals may be low confidence, but correlation produces meaningful detection outcomes.

Three Attack Scenarios by Tier

Representative attack scenarios illustrating how behavioral detections surface across agent deployment models

Tier 1	Tier 2	Tier 3
Compromised ChatGPT Session	**Compromised Code Agent**	**Rogue Digital Worker**
Situation User account abused to extract internal data through AI interactions.	**Situation** Agent approves malicious changes using leaked credentials across systems.	**Situation** Autonomous agent exfiltrates sensitive data outside its normal baseline.
What Fires • **Token spike** • **Guardrail violations** • **Off-scope tool invocation**	**What Fires** • **Out-of-scope execution** • **Credential misuse across systems** • **Off-pattern workflow sequences**	**What Fires** • **Baseline deviation** • **Lateral movement** • **Data exfiltration**

Figure 3. **Representative attack scenarios** and detection signals across agent deployment tiers

Scenario 1: Compromised Cloud LLM Account (Tier 1)

Situation

A user's ChatGPT account is compromised through credential theft. The attacker uses the session to systematically query internal documents uploaded to ChatGPT, leveraging legitimate access to bypass traditional controls.

What Exabeam Detects

Behavioral analysis identifies a shift in account usage, including a sharp increase in prompt and conversation volume, abnormal token consumption, and unusual patterns of document analysis tool usage. Repeated guardrail violations indicate attempts to extract sensitive information, while geographic anomalies reveal access from unexpected locations.

Investigation Insight

Correlated signals show a transition from normal conversational use to structured data extraction. Timeline reconstruction illustrates the progression of activity, allowing analysts to distinguish between user error and compromise based on behavioral change.

Scenario 2: Compromised Code Review Agent (Tier 2)

Future-State with Agent Telemetry and Credential Correlation

Situation

An organization deploys a code review agent using Claude Code with GitHub API access and pull request approval permissions. The agent's credentials are compromised, allowing an attacker to approve malicious pull requests containing backdoors while bypassing human review.

What Exabeam Detects Today

Detection is based on anomalies in service account or agent behavior, including first-time or unusual tool invocation, spikes in API or token usage, and deviations from expected usage patterns.

What Exabeam Is Building Toward

With agent telemetry and credential correlation, detection expands to execution-level analysis. Exabeam identifies abnormal task sequences, such as pull request approvals without preceding code analysis, and scope violations where the agent interacts with repositories outside its intended domain. Credential-to-agent linking associates API keys with originating agents, allowing detection of credential misuse across contexts. Additional signals include inconsistent credential usage from unexpected endpoints. Chain-of-custody reconstruction provides visibility into which user deployed the agent, which credentials were used, and which systems were accessed.

Why This Gap Exists

Today, agent credentials appear as separate identities in logs. The agent itself is not directly represented, and activity remains disconnected across systems. Telemetry instrumentation and credential correlation unify these signals, allowing detection to progress from isolated anomalies to full workflow visibility.

Scenario 3: Rogue Digital Worker (Tier 3)

Situation

An organization deploys a digital worker as a fully onboarded user identity within AD and Okta. The agent is authorized to process support tickets and update internal systems. A malicious insider modifies the agent's configuration to extract sensitive customer data to an external destination.

What Exabeam Detects Using UEBA

Because the digital worker operates as a formal identity, behavioral baselining applies immediately. Detection identifies changes in activity, including increased database query volume, access to previously unused sensitive fields, and interaction with systems outside the agent's normal scope. Data exfiltration signals emerge through bulk transfer patterns, unusual destinations, and high-frequency access to sensitive records. Additional indicators include lateral movement and privilege escalation inconsistent with the agent's defined role.

Why This Detection Is Effective

Because the agent is modeled as a user, activity is fully captured within established behavioral models. Deviations from baseline in access, resource usage, and system interaction trigger detections using existing UEBA capabilities. AI-specific detections extend this coverage by identifying patterns unique to autonomous behavior.

Customer-Facing Attribution and Explainability

When Exabeam surfaces agent-related detection, analysts must be able to quickly understand what occurred, how the activity progressed, and which entities were involved. This includes answering a set of fundamental questions: what happened, which identities were involved, how the agent operated, and what type of risk is represented.

Exabeam addresses this through structured, evidence-based case summaries that reconstruct activity across identities, systems, and workflows.

Case Architecture

Exabeam organizes detections into cases that provide a unified investigation view across related signals and activity.

Individual detections are aggregated into cases when multiple detections occur within a defined time window and collectively exceed a risk threshold. This grouping enables analysis of behavior in context, rather than as isolated events.

Each case includes an AI-generated summary produced by Exabeam Nova, which synthesizes detections, highlights relevant indicators, and recommends next investigative actions. This provides an interpretation layer while preserving access to underlying evidence.

Cases are supported by a chronological timeline that reconstructs the sequence of events, including agent creation, invocation, and activity across systems. This allows analysts to understand not just what occurred, but how activity progressed over time.

Entity-level context is also provided, including details on users, agents, devices, IP addresses, and systems involved in the activity. Supporting detections are included alongside each case, exposing the raw detection logic, triggering conditions, and reasoning behind each alert.

Agent Identification in Investigation

At present, agents are not universally represented as a formal entity type across all environments. As a result, agent activity is surfaced through multiple signals rather than a single unified identity.

When available, agent identifiers embedded in log data, such as vendor or product identifiers (for example, “anthropic+claude”), are used to identify agent activity directly. Exabeam Nova can recognize these patterns and incorporate them into case context.

Detection rules themselves also provide context. AI-specific detections are labeled to reflect agent-related behavior, and these signals contribute to identifying when activity is associated with agent execution rather than traditional activity.

In environments where explicit identifiers are not present, analyst reconstruct agent activity through manual investigation, correlating timestamps, tool invocation patterns, API usage, and behavioral sequences to infer the role of an agent in the activity.

Evidence and Threat Attribution

Exabeam does not attempt to infer intent. Instead, it presents evidence-based behavioral patterns aligned with known threat classes. Attribution is derived through correlation of multiple signals. Individual detections, such as token spikes, guardrail violations, or new tool invocations, are typically low confidence on their own. When evaluated together, they form patterns that indicate potential misuse, compromise, or policy violations.

Behavioral anomalies are assessed in combination, including deviations in token usage, data access patterns, and first-time or unusual activity. Policy violations are identified when agents operate outside defined constraints, including guardrail breaches or unauthorized resource access. Threat-specific indicators, such as prompt injection attempts, exposure of system prompts, and persistence-related behaviors, add further context for classification.

The strength of this model lies in correlation. Weak signals gain significance when evaluated together within execution sequences, system interactions, and identity activity. This approach allows analysts to construct a clear, evidence-based explanation of what occurred and why it represents risk.

The Path Forward

Agents are already in production use. They support customer-facing chatbots, execute workflows, and increasingly operate as autonomous digital workers. Traditional security controls, designed for human identities and predictable activity patterns, lack the visibility and behavioral context required to detect the risks these systems introduce.

Exabeam extends behavioral analytics to this emerging class of entities. By combining UEBA with ABA, Exabeam delivers visibility into cloud LLM platforms, integrated task agents, and embedded digital workers. Detection covers behavioral anomalies, prompt injection attempts, policy and guardrail violations, and insider threat indicators. Activity is correlated across systems and identities, preserving chain of custody and enabling confident attribution.

This approach allows security teams to move beyond event-based monitoring to analysis of behavior within complete execution paths. Agent activity becomes observable, measurable, and explainable within existing detection and investigation workflows.

At the same time, the agent ecosystem continues to evolve. Detection coverage will expand as telemetry improves, agent instrumentation becomes more standardized, and identity models for non-human entities mature. Exabeam is advancing in each of these areas through continued detection expansion, telemetry integration, and deeper attribution capabilities.

For security teams, this represents a fundamental shift. Agent-driven risk is no longer opaque. It can be identified through behavioral patterns, correlated across systems, and investigated using evidence-based workflows. As adoption of autonomous systems grows, the ability to understand and detect agent activity becomes a core requirement for modern security operations.

About Exabeam

Exabeam is the leader in behavior intelligence for the agentic enterprise. As organizations deploy digital workers and confront machine-speed adversaries, Exabeam delivers flexible, industry-proven solutions for insider threat coverage of humans and agents and faster, more accurate threat detection, investigation, and response (TDIR). Learn more at www.exabeam.com.

exabeam™

Learn more at
www.exabeam.com →