INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-17 18:54 UTC

AWS Continuum: Security at Machine Speed and Amazon Bedrock AgentCore Harness Generally Available

SECURITY ANALYSIS | LOW

SCC Item ID	SCC-STY-2026-0219
Type	Security Analysis
Severity	LOW
Affected Products	Amazon Web Services (AWS), Amazon Bedrock AgentCore, AWS Continuum (cloud-native services, GA release)
Published	2026-06-16
Discovery Source	Gemini

Executive Summary

AWS has moved two AI-driven capabilities to general availability: AWS Continuum, which applies AI to accelerate vulnerability detection and remediation in cloud environments, and Amazon Bedrock AgentCore, a framework for building and deploying production-grade AI agents. These releases signal a broader industry shift toward machine-speed, agentic security operations where AI systems, not human analysts, initiate and execute security responses. For CISOs, the strategic question is no longer whether to adopt AI-augmented security, but how to govern autonomous agents operating within production cloud infrastructure.

Technical Analysis

AWS's dual general availability announcement reflects a maturation of two distinct but complementary AI security paradigms in cloud environments.

AWS Continuum is positioned as an AI-accelerated security platform targeting the vulnerability detection-to-remediation pipeline. In traditional cloud security operations, the gap between detection and remediation is measured in hours or days, creating windows that threat actors actively exploit. Continuum's architecture, as described in secondary source reporting, aims to compress that window by automating analysis and response actions at machine speed. This positions it against existing cloud-native tooling like Amazon Inspector, which focuses on automated vulnerability assessment, and extends toward active remediation rather than passive discovery.

Amazon Bedrock AgentCore addresses a different problem: the operational complexity of deploying AI agents in production environments. Building agentic workflows has historically required significant custom scaffolding, orchestration logic, and manual integration work. AgentCore reduces that overhead, lowering the barrier for development teams to ship AI agents that can take autonomous actions within AWS environments.

The security implications of AgentCore warrant particular attention. AI agents operating autonomously in cloud infrastructure introduce new attack surface considerations. An agent with permissions to modify infrastructure, query data stores, or invoke APIs represents a high-value target. Compromise of an agent's identity, its system prompt, or its tool-call chain could allow an adversary to abuse legitimate, credentialed cloud actions in ways that are difficult to distinguish from authorized behavior. This is a known risk class in the MITRE ATLAS framework for AI systems, though no MITRE ATT&CK technique mappings are available in the provided source data for these specific releases.

The source data for this story is grounded in secondary reporting via Gemini search. No primary AWS documentation, AWS blog announcements, or official technical architecture references were available in the research materials available for this story. All technical characterizations above reflect what the secondary source data supports; specific implementation details, pricing, and integration architecture should be verified against official AWS documentation before operational decisions are made.

Action Checklist

1. Step 1: Assess exposure, determine if your organization uses AWS cloud services, particularly Amazon Bedrock, AWS developer tooling, or existing cloud-native security platforms that may be displaced or complemented by Continuum and AgentCore.
2. Step 2: Review agent identity controls, if your organization is evaluating or has deployed AI agents in AWS environments, audit the IAM roles and permissions granted to those agents; apply least-privilege principles per CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) and CIS 6.1 (Establish an Access Granting Process).
3. Step 3: Establish logging and monitoring baselines for agentic activity, before deploying any AI agent in production, ensure audit logging is enabled across the environment per CIS 8.2 (Collect Audit Logs) and NIST SI-4 (System Monitoring), with specific attention to agent-initiated API calls, privilege escalations, and resource modifications.
4. Step 4: Update threat model, incorporate the emerging risk class of compromised AI agent identity and prompt injection into your threat register; document the conditions under which an AI agent operating in your environment could be abused to perform unauthorized cloud actions.
5. Step 5: Engage AWS account teams and review official GA documentation. Note: This story was researched from secondary sources. Locate the official AWS GA announcement via AWS Blog (aws.amazon.com/blogs) or your AWS account team, as primary documentation was not available during research. Verify capability scope, integration requirements, and security architecture directly from AWS before making adoption or procurement decisions.
6. Step 6: Brief leadership on agentic AI risk posture, frame the conversation around governance of autonomous systems, not just adoption of new tooling; emphasize that AI agents with cloud permissions require the same access lifecycle rigor as human privileged users per CIS 6.2 (Establish an Access Revoking Process).

IR / Forensic Enrichment

Triage Priority	DEFERRED
Escalation Criteria	Escalate to urgent if CloudTrail logs reveal that an existing Bedrock AgentCore agent has executed IAM privilege escalation actions (AttachRolePolicy, PutRolePolicy, CreateAccessKey) autonomously, or if a prompt injection indicator is discovered in Bedrock model invocation logs triggering unauthorized resource modifications — either condition warrants immediate IR engagement and potential breach notification assessment.
Recovery Notes	Because this story describes a GA product announcement rather than an active exploitation event, recovery actions are adoption-gated: do not deploy AgentCore or Continuum in production until the IAM least-privilege audit (Step 2) and logging baseline (Step 3) are complete and verified. If agents are already deployed, treat any discovered over-privileged role as a misconfiguration incident — revoke excess permissions, rotate the agent's execution role credentials, and monitor CloudTrail for `bedrock-agent.amazonaws.com` events for a minimum of 30 days post-hardening to confirm no residual unauthorized activity. Validate that Bedrock model invocation logging captures input and output payloads to enable retrospective prompt injection detection.
Forensic Artifacts	AWS CloudTrail logs filtered on eventSource `bedrock-agent.amazonaws.com` and `bedrock.amazonaws.com` — captures all AgentCore InvokeAgent, CreateAgent, UpdateAgent, and action group execution API calls with caller identity, source IP, and timestamp; primary evidence source for unauthorized agentic actions | Bedrock model invocation logs (enabled via PutModelInvocationLoggingConfiguration) stored in S3 or CloudWatch Logs — contains the actual prompt input and model output for each agent invocation, which is the primary artifact for detecting prompt injection attempts against AgentCore agents | IAM credential report and CloudTrail `iam.amazonaws.com` events for agent execution roles — specifically CreateAccessKey, AttachRolePolicy, PutRolePolicy, and AssumeRole events tied to Bedrock agent role ARNs, evidencing any agent-driven privilege escalation attempts | AWS Config configuration history for IAM roles associated with AgentCore — provides a point-in-time record of permission changes to agent execution roles, useful for establishing whether a role was modified after initial deployment in a way that expanded the agent's blast radius | Amazon S3 server access logs and CloudTrail S3 data events for buckets accessible to AgentCore knowledge bases or action groups — captures any data exfiltration or unauthorized read/write operations that an abused agent identity could perform against connected data sources

Per-Action IR Details

Step 1: Assess exposure — determine if your organization uses AWS cloud services, particularly Amazon Bedrock, AWS developer tooling, or existing cloud-native security platforms that may be displaced or complemented by Continuum and AgentCore.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR capability through asset awareness and environment baselining

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory)

Compensating: Run `aws iam list-roles --query 'Roles[?contains(RoleName, `Bedrock`) || contains(RoleName, `Agent`)]'` and `aws bedrock list-foundation-models` (AWS CLI, free) to enumerate Bedrock usage. Cross-reference with `aws cloudtrail lookup-events --lookup-attributes AttributeKey=EventSource,AttributeValue=bedrock.amazonaws.com` to identify any existing AgentCore or Continuum

API calls in CloudTrail logs.

Evidence: This is a pre-deployment assessment step; no live system state is being altered. Capture current IAM role listings and CloudTrail event history for `bedrock.amazonaws.com` and `bedrock-agent.amazonaws.com` API sources before any configuration changes are made, to establish a pre-adoption baseline.

Step 2: Review agent identity controls — if your organization is evaluating or has deployed AI agents in AWS environments, audit the IAM roles and permissions granted to those agents; apply least-privilege principles per CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts) and CIS 6.1 (Establish an Access Granting Process).

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Hardening environment and access controls before agentic workloads reach production

Controls: CIS 5.4 (Restrict Administrator Privileges to Dedicated Administrator Accounts), CIS 6.1 (Establish an Access Granting Process), NIST IR-4 (Incident Handling)

Compensating: Use `aws iam get-role --role-name ` and `aws iam list-attached-role-policies --role-name ` to enumerate permissions for every IAM role associated with Bedrock AgentCore execution. Use AWS IAM Access Analyzer (free tier) to flag any role with overly permissive policies (`*` actions or resources) granted to agent execution roles. Document findings in a spreadsheet with role ARN, attached policies, and last-used timestamp from `aws iam get-role` output.

Evidence: Before modifying or restricting any IAM role assigned to an existing AI agent, export the current IAM policy JSON (`aws iam get-role-policy` and `aws iam get-policy-version`) and capture CloudTrail entries for `iam.amazonaws.com` events (CreateRole, AttachRolePolicy, PutRolePolicy) tied to the agent role ARN. These records document the pre-hardening permission state and are volatile in the sense that policy changes overwrite the active configuration.

Step 3: Establish logging and monitoring baselines for agentic activity — before deploying any AI agent in production, ensure audit logging is enabled across the environment per CIS 8.2 (Collect Audit Logs) and NIST SI-4 (System Monitoring), with specific attention to agent-initiated API calls, privilege escalations, and resource modifications.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Configuring logging infrastructure and monitoring capability prior to production deployment of agentic workloads

Controls: CIS 8.2 (Collect Audit Logs), NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), NIST SI-4 (System Monitoring)

Compensating: Enable AWS CloudTrail with S3 data events and enable Amazon Bedrock model invocation logging via `aws bedrock put-model-invocation-logging-configuration` (both free within CloudTrail and S3 storage costs). For a 2-person team, configure a CloudWatch Logs Insights query filtering on `eventSource = bedrock-agent.amazonaws.com` and `errorCode IS NOT NULL` or `eventName IN (InvokeAgent, CreateAgentActionGroup)` to surface anomalous agent API calls without a SIEM. Export results weekly.

Evidence: This step configures monitoring infrastructure and does not alter live system state. Before enabling logging (which may itself generate a configuration-change CloudTrail event), snapshot the current CloudTrail trail configuration (`aws cloudtrail get-trail-status`) and existing Bedrock logging configuration to document the pre-baseline state and confirm no prior logging gaps that could obscure earlier agentic activity.

Step 4: Update threat model — incorporate the emerging risk class of compromised AI agent identity and prompt injection into your threat register; document the conditions under which an AI agent operating in your environment could be abused to perform unauthorized cloud actions.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Developing and maintaining threat models and incident criteria for novel attack classes including agentic AI abuse

Controls: NIST IR-8 (Incident Response Plan), NIST IR-4 (Incident Handling)

Compensating: Document threat scenarios using a simple threat register spreadsheet: for each Bedrock AgentCore action group, record the IAM permissions it holds, the data sources it can access, and the worst-case unauthorized action it could execute if its system prompt were injected or its identity token were stolen. Use the MITRE ATLAS framework (free, atlas.mitre.org) as a reference taxonomy for ML-specific attack patterns such as prompt injection and model inversion to structure scenario narratives.

Evidence: This is a documentation and planning activity that does not alter live system state; no volatile evidence capture is required. However, as part of threat modeling, retrieve and preserve current Bedrock agent definitions (`aws bedrock-agent list-agents`, `aws bedrock-agent get-agent --agent-id `) and their associated action group ARNs as baseline evidence of the production agent configuration at the time the threat model is constructed.

Step 5: Engage AWS account teams and review official GA documentation — secondary source reporting is the basis for this story; verify capability scope, integration requirements, and security architecture directly from AWS before making adoption or procurement decisions.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Intelligence sharing, external coordination, and verification of information before operational decisions are made

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives), NIST IR-6 (Incident Reporting)

Compensating: Subscribe to the AWS Security Bulletins RSS feed (https://aws.amazon.com/security/security-bulletins/) and the AWS What's New feed for Bedrock and AI/ML services. File a formal inquiry through your AWS TAM or Solutions Architect documenting specific security architecture questions about AgentCore's IAM execution model, prompt injection mitigations, and Continuum's automated remediation scope before signing off on any internal adoption decision.

Evidence: No live system state is altered by this step. Preserve copies of all AWS GA announcement documentation, release notes, and any written responses from AWS account teams as dated records in your security decision log — these serve as provenance evidence for adoption decisions if questions arise later during audits or regulatory review.

Step 6: Brief leadership on agentic AI risk posture — frame the conversation around governance of autonomous systems, not just adoption of new tooling; emphasize that AI agents with cloud permissions require the same access lifecycle rigor as human privileged users per CIS 6.2 (Establish an Access Revoking Process).

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons learned dissemination and policy/governance updates driven by emerging threat class awareness

Controls: CIS 6.2 (Establish an Access Revoking Process), NIST IR-1 (Policy and Procedures), NIST IR-8 (Incident Response Plan)

Compensating: Prepare a one-page executive brief using concrete examples: map each Bedrock AgentCore IAM role to an equivalent human privileged role (e.g., 'this agent has the equivalent of a sysadmin credential for your S3 environment') to make the risk tangible for non-technical leadership. Include a table of current agent role ARNs, their effective permissions, and whether a formal access review and offboarding process exists — built from the `aws iam` CLI outputs gathered in Step 2.

Evidence: This is a governance communication step and does not alter live system state. Attach the IAM role audit output from Step 2 and the threat register from Step 4 as supporting exhibits to the briefing materials, ensuring leadership decisions are grounded in verified environment data rather than vendor marketing.

Detection Guidance

For organizations evaluating or deploying AWS Continuum or Amazon Bedrock AgentCore, detection priorities should focus on the behavior of autonomous agents operating within your cloud environment rather than on

indicators specific to these GA releases.

Log sources to baseline and monitor per NIST AU-2 (Event Logging) and AU-6 (Audit Record Review, Analysis, and Reporting):

- AWS CloudTrail: Capture all API calls made by Bedrock agent execution roles. Flag any agent-initiated calls to IAM, S3, EC2, or Lambda that were not part of the agent's documented workflow.

- AWS CloudWatch: Alert on anomalous invocation rates, unexpected resource creation or deletion events, or out-of-hours agent execution patterns.

- VPC Flow Logs: Monitor for unexpected outbound connections initiated by agent compute resources.

Behavioral anomalies to hunt for:

- Agent identity (IAM role) performing actions outside its documented tool-call scope.

- Privilege escalation attempts originating from Bedrock execution roles.

- Repeated access denials followed by successful calls from the same agent identity, which may indicate probing for permission boundaries.

- Modifications to agent system prompts or knowledge base configurations by accounts other than designated administrators.

Gap audit: Review whether your SIEM or cloud detection platform has rules tuned to distinguish authorized agentic actions from anomalous ones. Standard behavioral baselines built for human users may not capture the high-velocity, repetitive API call patterns that legitimate AI agents generate, creating blind spots.

No specific IOCs are available for these GA releases. Applicable D3FEND countermeasures include D3-UAP (User Account Permissions) to restrict agent IAM scope, D3-LAM (Local Account Monitoring) adapted to cloud identity monitoring for agent roles, and D3-CRO (Credential Rotation) applied to agent execution credentials and API keys.

Framework Mappings

NIST-800-53R5

- **SI-4** — System Monitoring

CIS-V8

- **8.2** — Collect Audit Logs

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.23** — Information security for use of cloud services

Sources

Source	URL	Tier
Vulnerability Reporting - Amazon Web Services (AWS)	https://aws.amazon.com/security/vulnerability-reporting/	T3
Cloud Security – Amazon Web Services (AWS)	https://aws.amazon.com/security/	T3
Top 10 AWS Security Issues You Need to Know - SentinelOne	https://www.sentinelone.com/cybersecurity-101/cloud-security/aws-se...	T3
Automated Software Vulnerability Management - Amazon Inspector	https://aws.amazon.com/inspector/	T3
Overview of Vulnerability Assessment for AWS | Security Command ...	https://docs.cloud.google.com/security-command-center/docs/vulnerab...	T3

Generated 2026-06-17 18:54 UTC by TJS Security Command Center