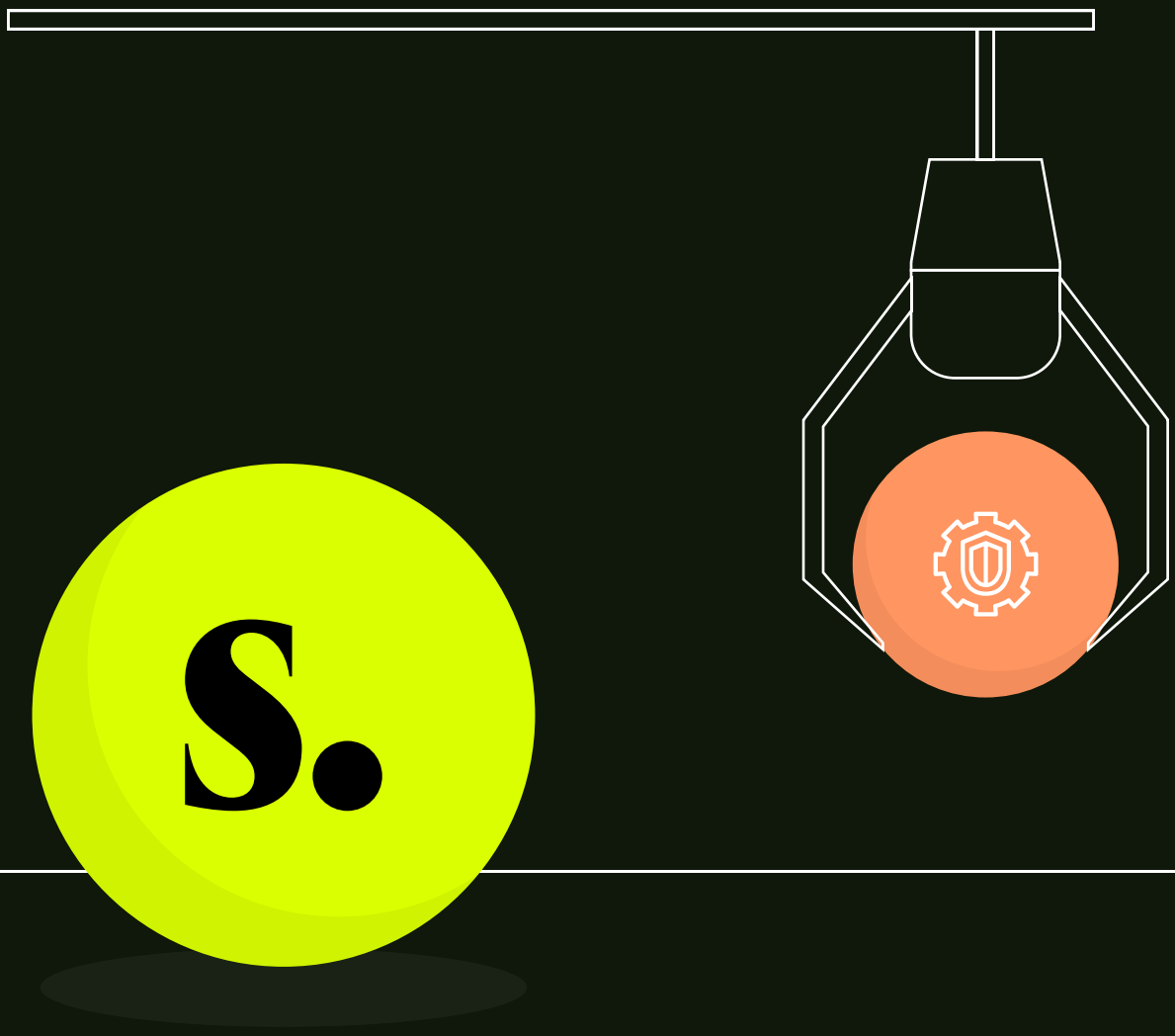
sweet.

Replacing AWS GuardDuty with Sweet Security: **Strategic Comparison Guide**



www.sweet.security

Introduction

Amazon GuardDuty is a widely used threat detection service that continuously monitors AWS accounts and workloads for malicious activity and anomalous behavior. While GuardDuty provides valuable insights, it may not align with the needs of organizations seeking deeper security visibility and real-time detection.

Sweet Security offers a modern, detection-first alternative with full-stack runtime monitoring, cross-cloud visibility, and built-in incident response. This document outlines the core differences between GuardDuty and Sweet Security, providing a strategic comparison across detection depth, response capabilities, EKS and container security, application visibility, and operational cost.

Core Capabilities Comparison

Capability	AWS GuardDuty	Sweet Security
Detection	Log-based, anomaly + intel driven	Full-stack runtime + cloud-layer behavioral detection
EKS Protection	Control plane logs + optional runtime agent	Deep runtime + in-cluster visibility + API activity
Attack Context	Individual findings	Correlated attack story with root cause & impact
API Visibility	None	Layer 7 monitoring, API misuse, data exposure alerts
False Positives	Configurable, but context-limited	High-fidelity, LLM-powered analysis reduces noise
Customization	Suppression filters only	Custom rules, policies, detection logic
Environment Coverage	AWS only	AWS, Azure, GCP, on-prem, hybrid cloud
Pricing Model	Usage-based (log volume dependent)	Predictable pricing based on assets

Where Sweet Security Goes Further

1 Full-Stack Threat Detection, Not Just Cloud Logs

GuardDuty relies on logs like CloudTrail, VPC Flow, and DNS. Sweet Security captures data from cloud events, workloads, and applications. It deploys lightweight sensors to monitor process execution, file access, network behavior, and API usage in real time—catching threats at every layer, not just the control plane.

2 Correlated Attack Stories vs. Disjointed Alerts

GuardDuty generates individual findings for each detected anomaly. Sweet Security automatically links related activities into a single incident timeline. Analysts see how the attack began, what it touched, and what actions were taken—without jumping between logs or tools. Sweet excels in incident storytelling, combining multiple findings across different layers into one concise, actionable narrative. This not only saves time but also ensures faster prioritization and remediation.

3 Real-Time Behavioral Analysis

Sweet uses LLM-powered analytics to understand normal behavior and flag deviations. Whether it's a new process spawning in a container, or an unusual series of API calls, Sweet correlates behaviors to identify advanced threats that log-only tools may miss.

4 Layer 7 API Security

GuardDuty lacks insight into API calls and user interactions at the app layer. Sweet adds Layer 7 inspection—tracking endpoint usage, authentication bypasses, and sensitive data exposure across REST and internal APIs.

5 Kubernetes and EKS Runtime Protection

GuardDuty offers EKS audit log monitoring and optional runtime agent support, limited to predefined detections. Sweet provides real-time monitoring of container behavior, lateral movement, privilege escalation, and more across all K8s platforms, not just EKS.

6 Multi-Cloud and Hybrid Support

GuardDuty is AWS-only. Sweet Security supports AWS, GCP, Azure, and on-prem workloads in one unified dashboard—ideal for teams operating across cloud providers or migrating workloads.

7 Predictable Cost and Less Noise

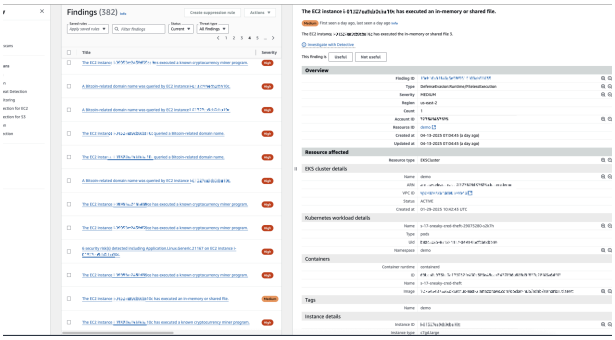
GuardDuty's pay-per-use pricing can be unpredictable as workloads scale. Sweet offers fixed-cost models and high-fidelity alerts, reducing operational overhead and eliminating alert fatigue.

Practical Example: Unified Threat Detection in Action

Scenario

An attacker executes a script directly from memory to avoid detection. The script probes security defenses, accesses files, and attempts to extract credentials — all without leaving traditional file system traces.

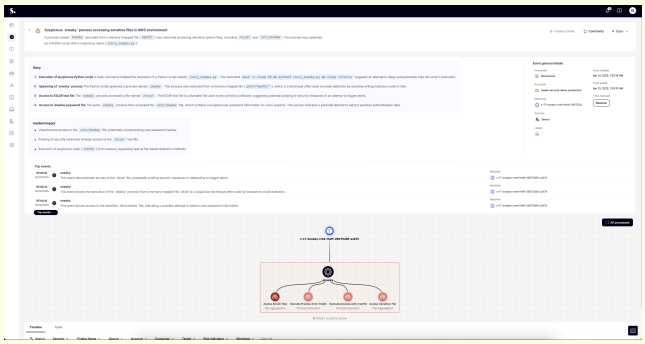
AWS GuardDuty



Amidst hundreds of noisy findings, GuardDuty surfaced a single relevant alert — an in-memory or shared file execution. But it stops there. There's no visibility into how the attack started, what script was used, or which files were accessed. The full behavior of the attacker or the sequence of events are not clear, and teams are left with isolated metadata and no broader context, making it difficult to understand impact or respond confidently.

VS.

Sweet Security



Sweet identifies the suspicious process, links related behavior, and builds a complete incident automatically. It surfaces the use of in-memory evasion techniques, shows which sensitive files were accessed, and maps out the entire attack in a clear timeline. Teams get full context — what happened, what was impacted, and how to respond — all in one place.

Sweet shows:

- Execution of Python script directly from memory
- Runtime access to files indicating security probing and credential theft
- Stealth techniques to evade file-based detection systems
- Linked activity and full incident timeline

The incident includes full context, impact analysis, and suggested remediation options.

This approach applies beyond EKS: similar cross-layer detection and response would apply to EC2 compromise, Azure workload privilege abuse, or API endpoint tampering in a GCP-hosted app.

Beyond GuardDuty:
Other AWS Services Sweet Can Replace

Security Hub

Replaced by our unified risk-based dashboard that prioritizes findings and toxic combinations across vulnerabilities, identities, secrets and misconfigurations.

Amazon Inspector

Replaced by Sweet's runtime vulnerability detection, including exploitability context runtime insights such as reachabilty and execution of packages into running workloads.

CloudTrail & Cloud Watch-based monitoring

We don't replace those, but ingest and enrich AWS logs natively, adding the right context to understanding the logs meaning, and in real-time.

Final Recommendations

If your team prioritizes:

- Real-time, correlated detection across cloud and runtime layers
- Fast, automated threat response
- Full-stack Kubernetes and container coverage
- API security, identity threat detection, and reduced alert noise

Sweet Security provides a broader, deeper alternative to GuardDuty.

If your goal is baseline AWS-native visibility with low operational maintenance, GuardDuty can still provide value. However, as threats evolve, the need for runtime visibility, cross-environment context, and faster response often outweighs the benefits of remaining AWS-native only.