upwind

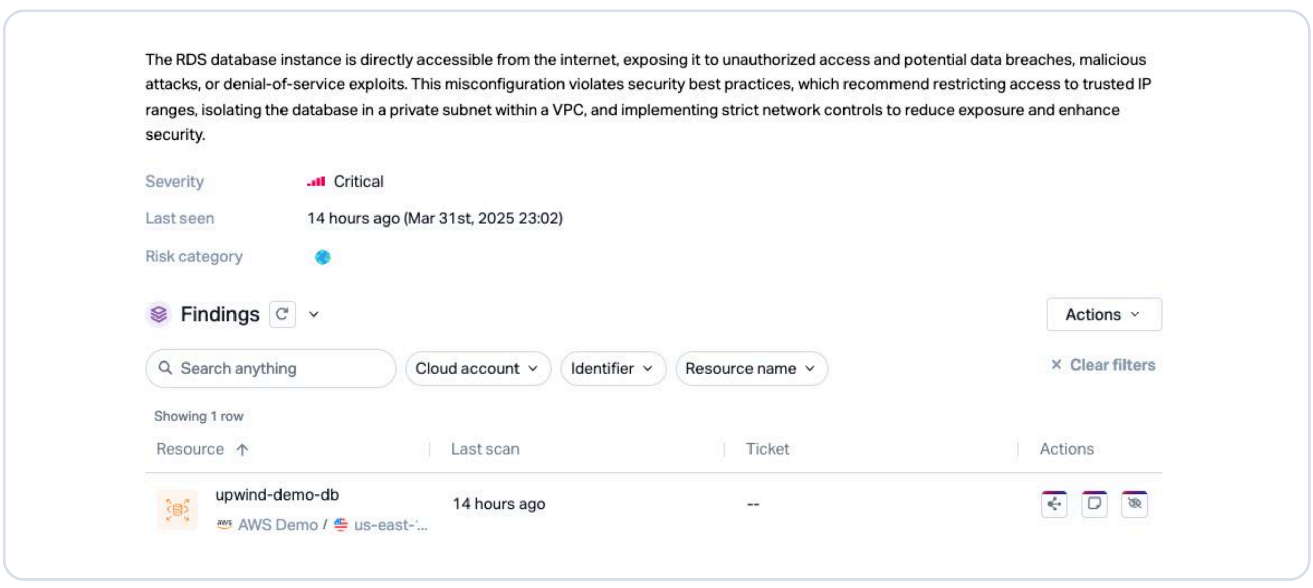
Cloud Security Assessment Report

Key Findings

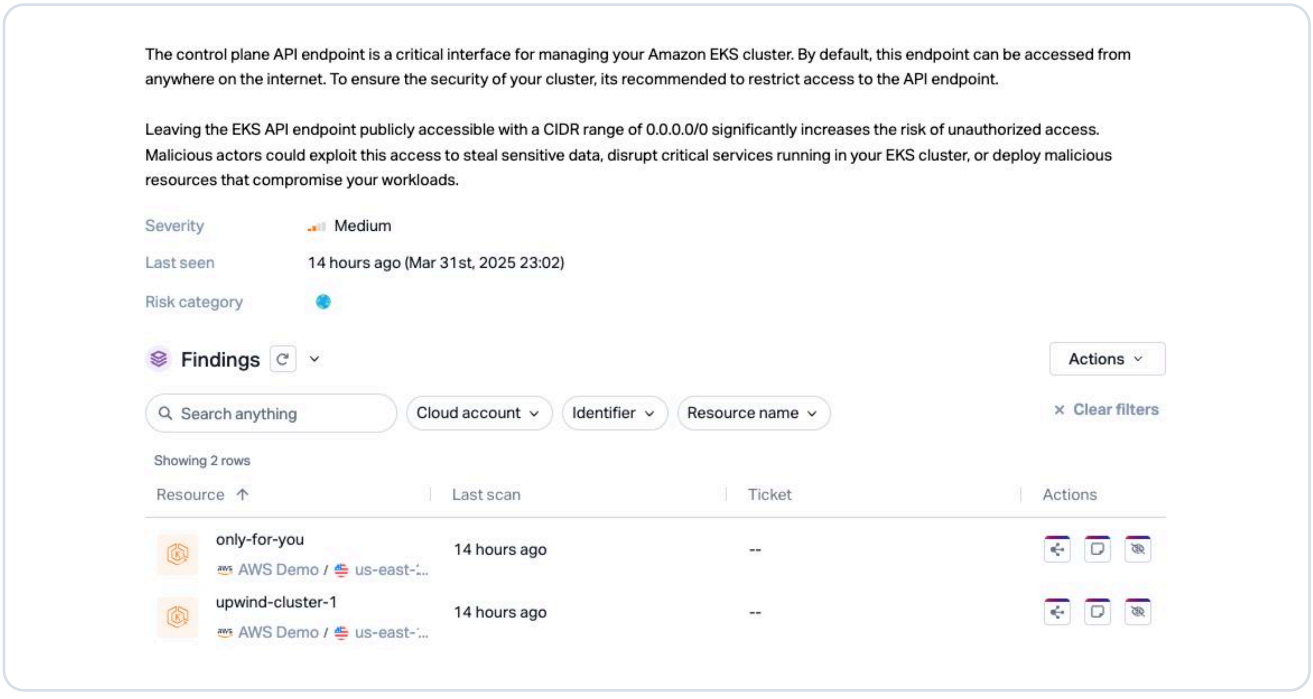
A) Cloud Security Posture Assessment (CSPM) – Basic Phase

Misconfigurations & Exposure:

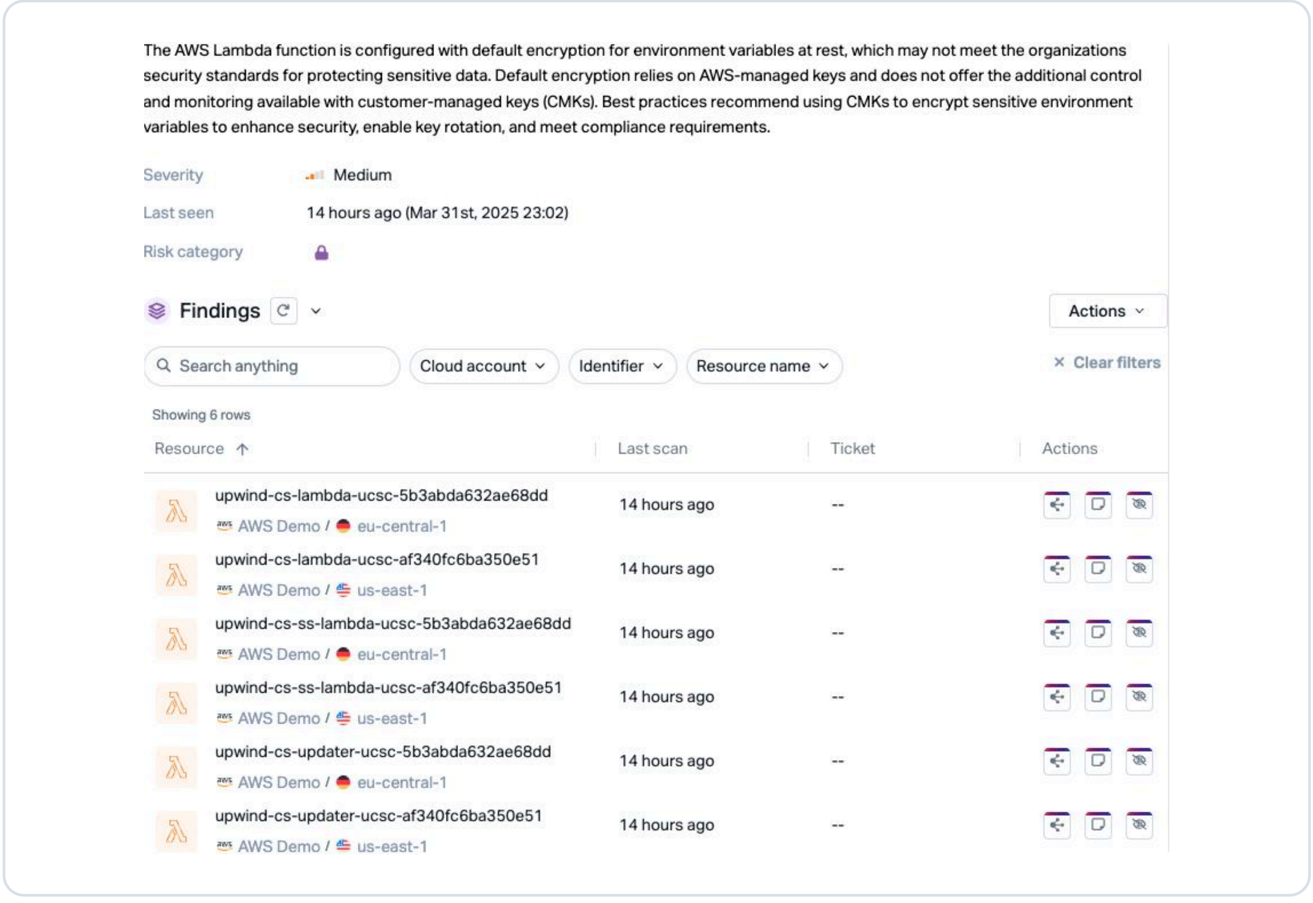
One RDS Instance has been identified as being directly accessible from the internet.



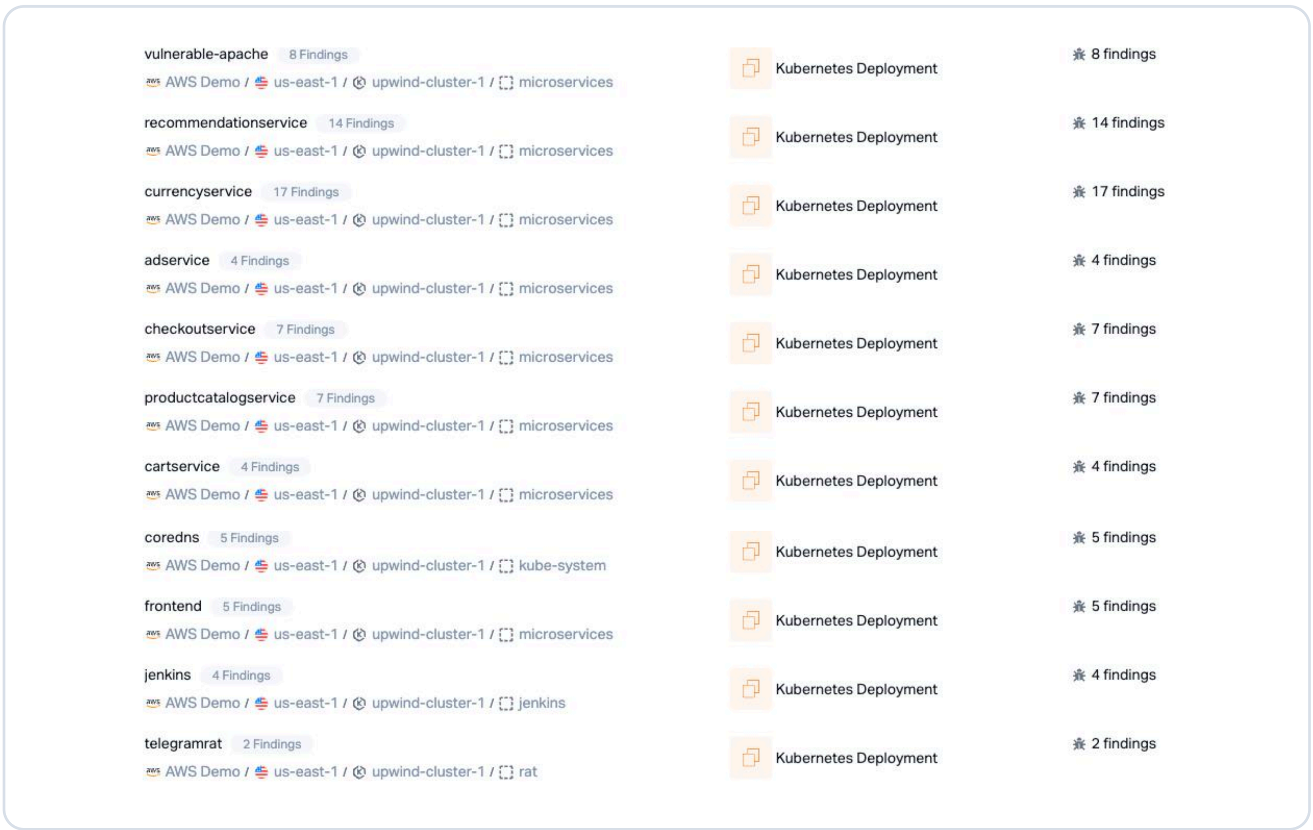
Two EKS API Endpoints have been identified as Internet Accessible.



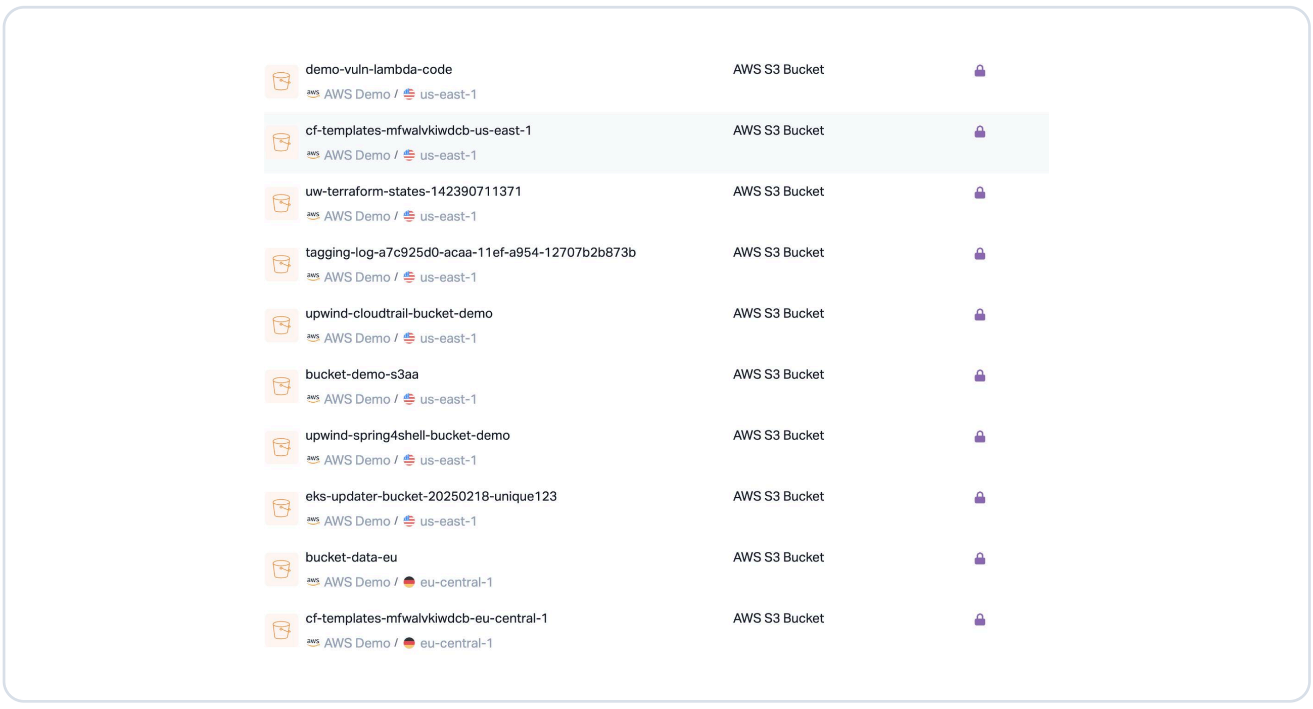
AWS Lambda Functions have been identified with standard default encryption for environment variables - we recommend using CMKs to encrypt these variables to meet compliance requirements.



All assets listed here are **publicly exposed assets** with critical/high vulnerabilities, and we recommend remediation immediately for 88 CVEs total.

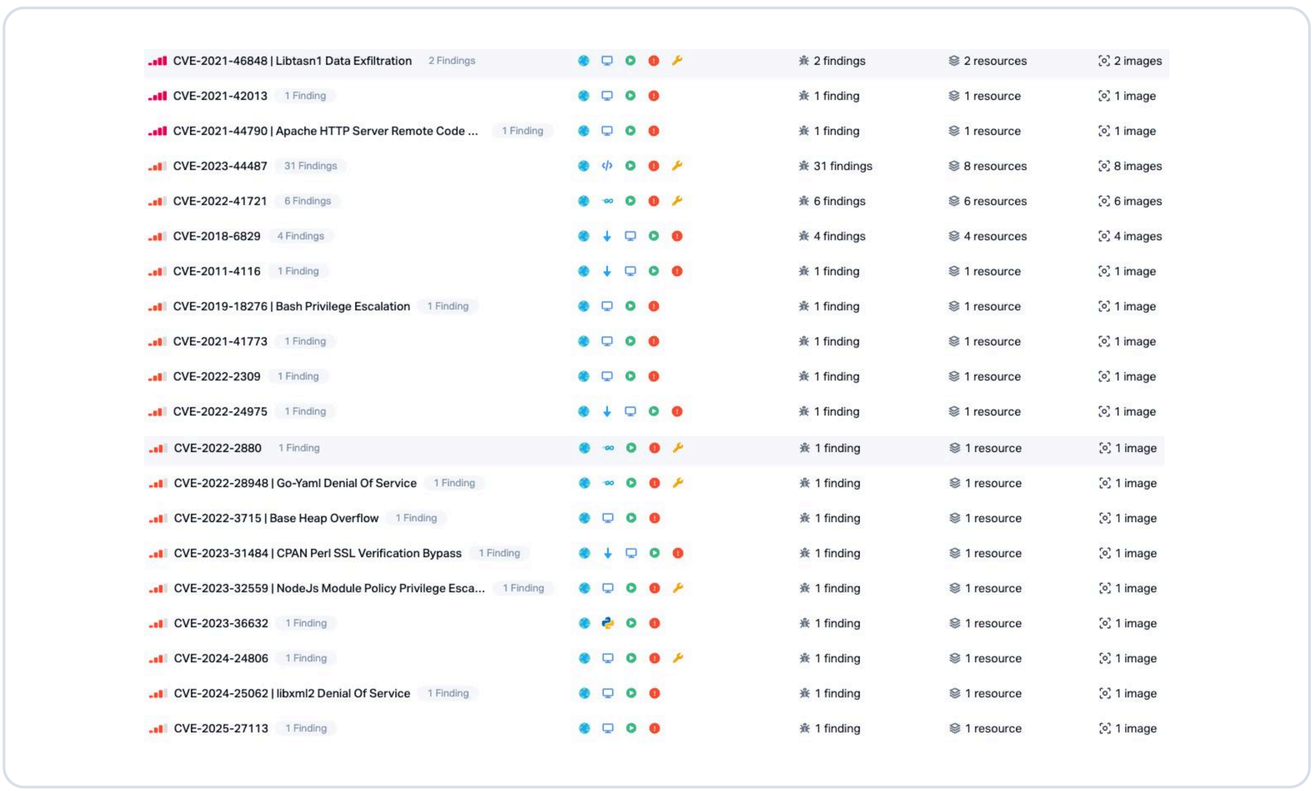


There are 11 detected insecure S3 buckets with Internet Exposure.



Critical Vulnerabilities with Internet Exposure:

We recommend remediating all findings associated with these CVEs, as these are all Critical or High CVEs with Internet exposure.



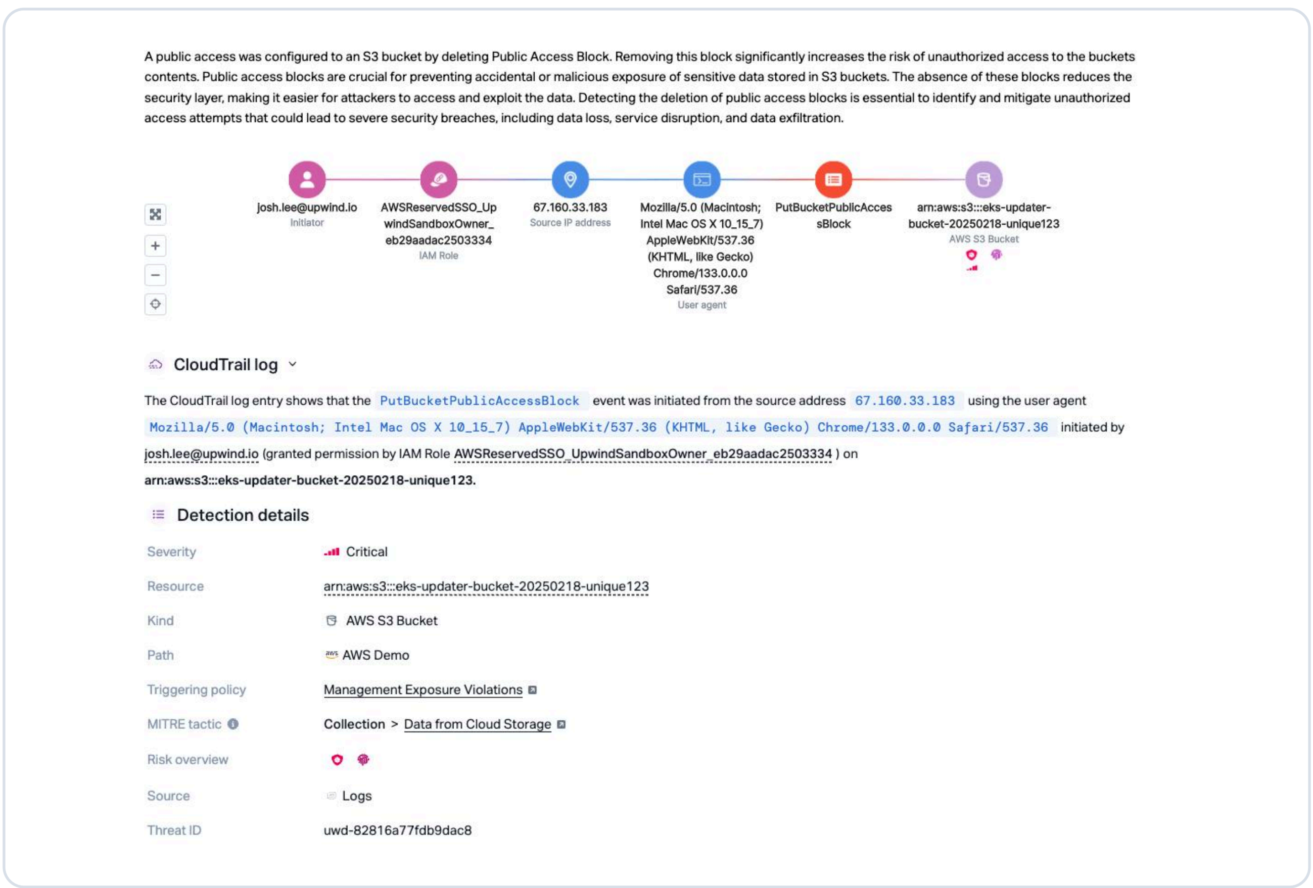
Data Insights:

Detect generative AI platforms being used.

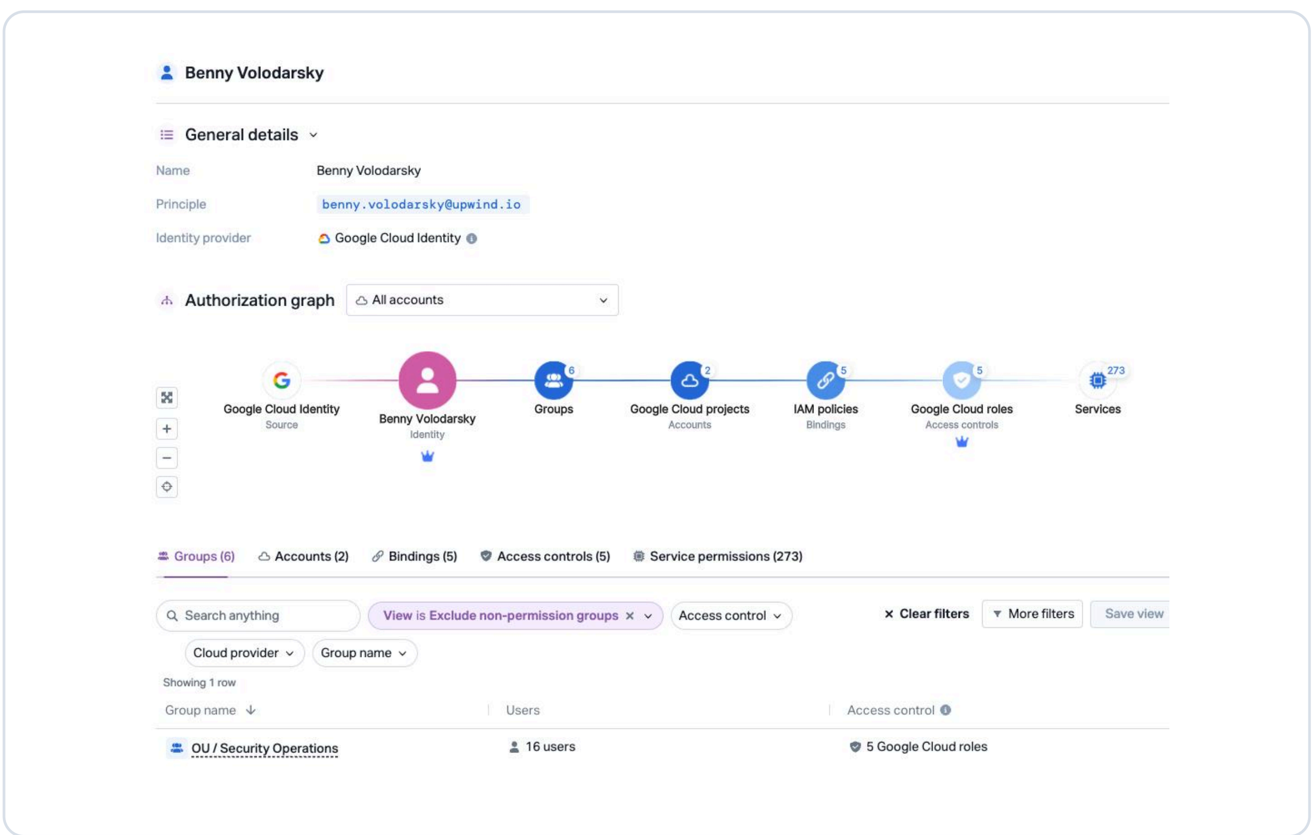
None found at this time.

Identify suspicious CloudTrail activity:

Access to an S3 Bucket was found



Identity Insights:

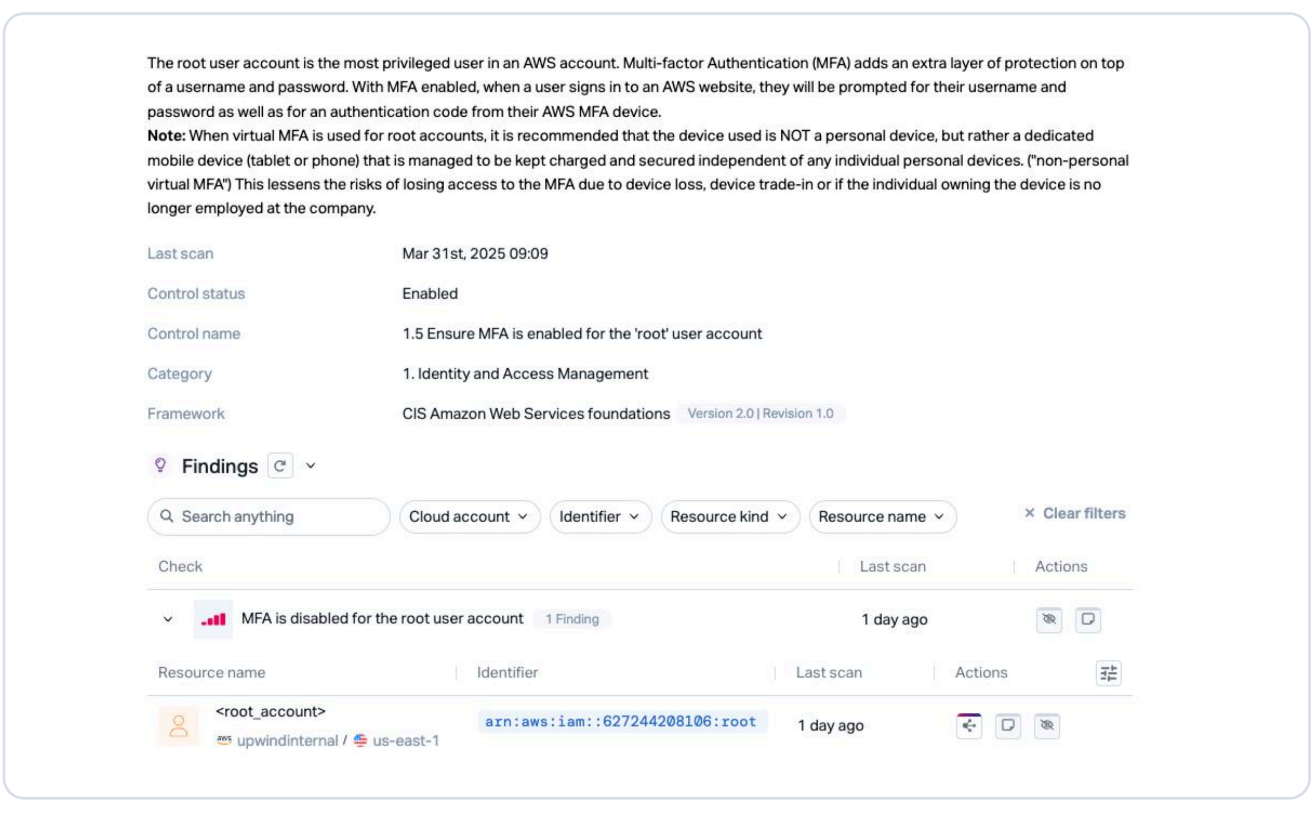


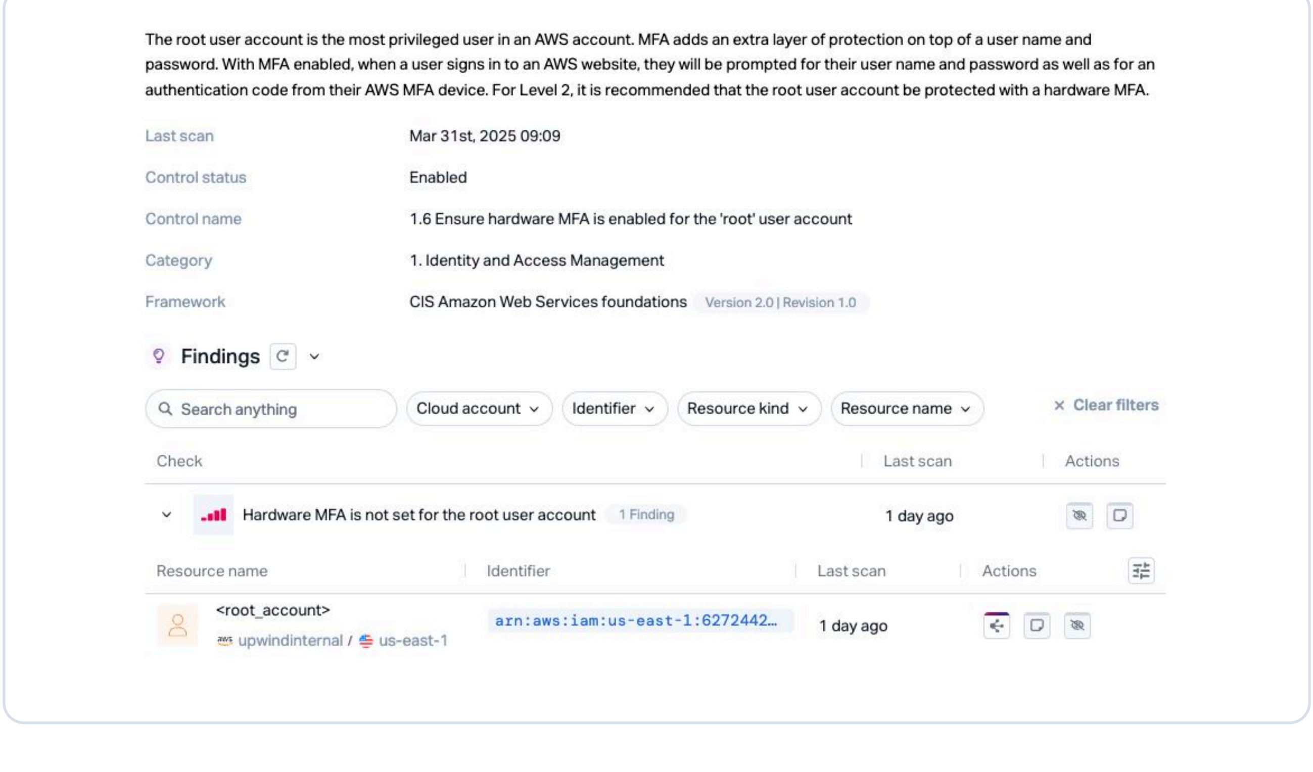
Feature coming:

- Enforcement of Principle of Least Privilege
- Identifying accounts that have not been used in 180 days+
- Identifying accounts with permissions that have never been used
- Identifying System Accounts with unused permissions

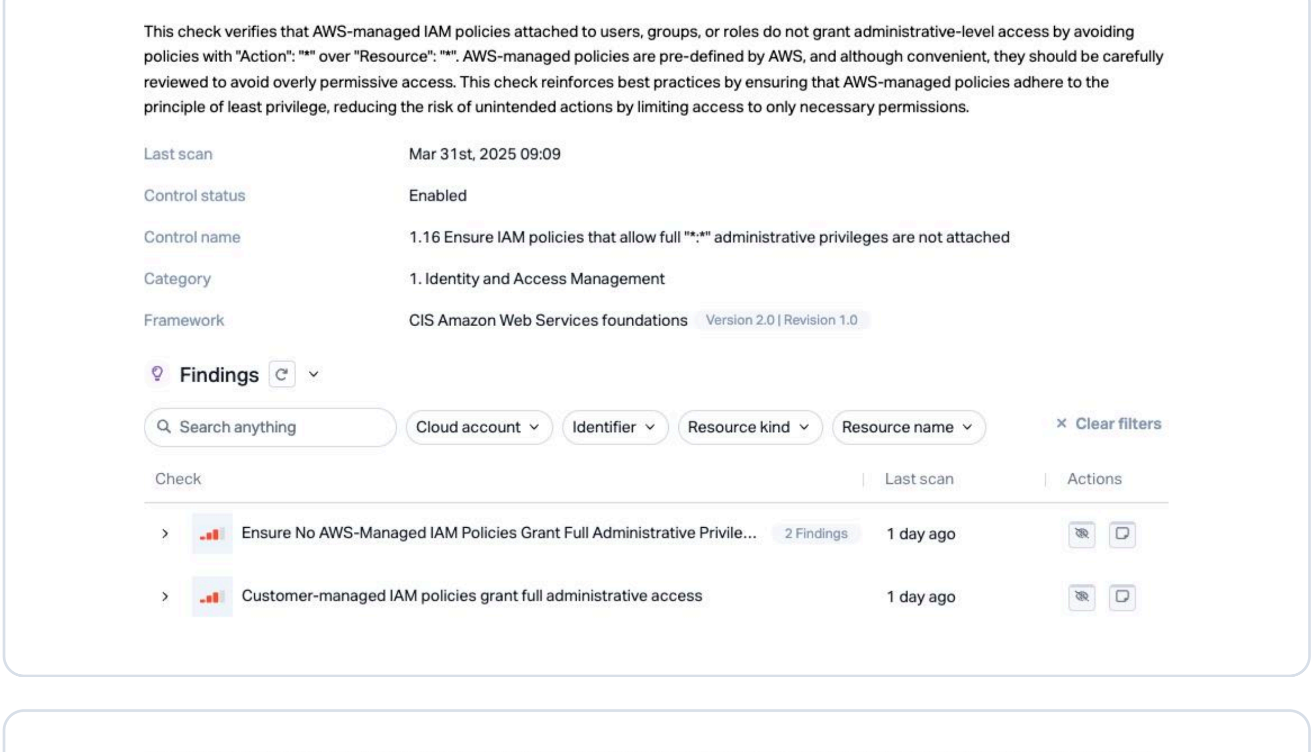
Compliance Posture CIS Benchmark:

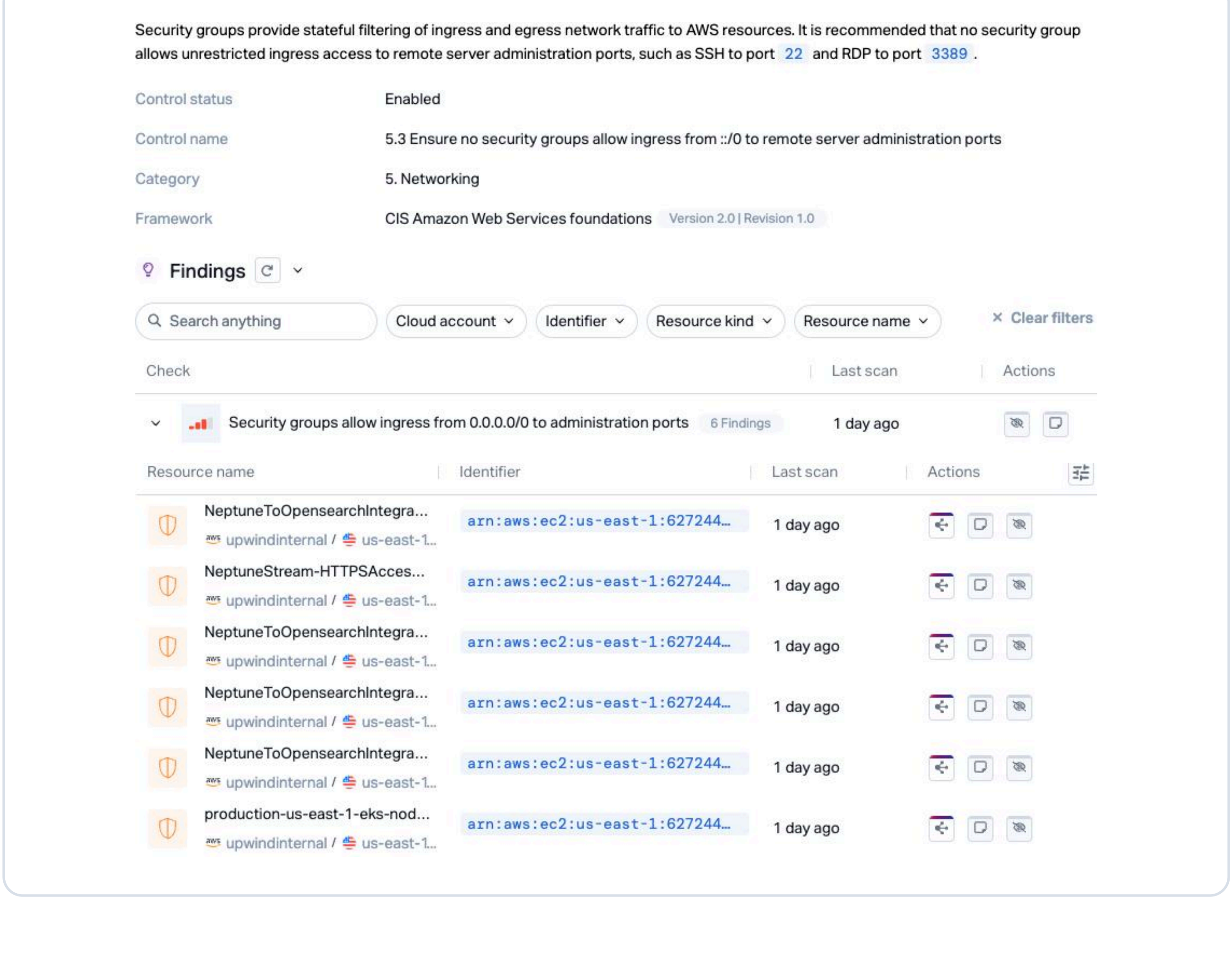
2 Critical CIS Benchmark Findings have been identified - both related to MFA for the root user account. Highly recommend remediating both findings immediately.





16 High Findings have been found for various CIS Benchmarks.





Key Findings

B) Kubernetes Security Posture Assessment (KSPM) – Advanced Phase

K8s Misconfigurations:

Disable anonymous authentication on the Kubelet server to ensure that only authenticated requests are processed. This prevents unauthorized access by treating unauthenticated requests as invalid, enhancing security for your cluster.

Last scan	Mar 31st, 2025 09:06
Control status	Enabled
Control name	4.2.1 Ensure anonymous-auth argument is disabled for kubelet
Category	4.2. Kubelet
Framework	CIS Kubernetes Version 1.9

Findings

Search anything | Cloud account | Identifier | Resource kind | Resource name | × Clear filters

Check	Last scan	Actions
Ensure anonymous-auth argument is disabled for kubelet (5 Findings)	1 day ago	

Resource name	Identifier	Last scan	Actions
ip-10-11-19-178.ec2.internal upwindinternal / us-east-1...	`0697b420-7ae4-4c5c-89c0-e9e4...`	1 day ago	
ip-10-11-22-46.ec2.internal upwindinternal / us-east-1...	`65615162-350e-42b7-8f8a-f0a7...`	1 day ago	
ip-10-11-20-189.ec2.internal upwindinternal / us-east-1...	`8d326cd1-ce63-4a9c-a2d1-b673...`	1 day ago	
ip-10-11-31-224.ec2.internal upwindinternal / us-east-1...	`c667e165-d2e7-4321-bcde-33ee...`	1 day ago	
ip-10-11-60-198.ec2.internal upwindinternal / us-east-1...	`e01ce9ba-144e-47e7-99d9-3209...`	1 day ago	

Kubernetes stores secrets that may contain sensitive information, such as service account tokens or credentials used by workloads. To reduce the risk of privilege escalation and unauthorized access, permissions to retrieve `get`, `list`, or `watch` secrets should be restricted to only those users and processes that require them. Limiting access to secrets within both Roles and ClusterRoles enhances security by preventing unintended exposure of sensitive data across the cluster.

Control status	Enabled
Control name	5.1.2 Minimize access to secrets
Category	5.1. RBAC and Service Accounts
Framework	CIS Kubernetes Version 1.9

Findings

Search anything | Cloud account | Identifier | Resource kind | Resource name | × Clear filters

Check	Last scan	Actions
Minimize access to secrets in Kubernetes Roles (11 Findings)	1 day ago	
Minimize access to secrets in Kubernetes ClusterRoles (20 Findings)	1 day ago	

Resource name	Identifier	Last scan	Actions
admin upwindinternal / us-east-...	`admin::7be4c420-8183-4381-bf...`	1 day ago	
argocd-notifications-controller upwindinternal / us-east-...	`argocd-notifications-control...`	1 day ago	

Kubernetes Roles and ClusterRoles define access permissions to resources and actions within the cluster. The use of the `*` wildcard in these roles grants unrestricted permissions across all resources, which can lead to unintended privilege escalation, especially as new resources are introduced over time. To follow the principle of least privilege, explicit resource and action definitions should be used instead of wildcards to ensure controlled and secure access management.

Control status	Enabled
Control name	5.1.3 Minimize wildcard use in Roles and ClusterRoles
Category	5.1. RBAC and Service Accounts
Framework	CIS Kubernetes Version 1.9

Findings

Search anything | Cloud account | Identifier | Resource kind | Resource name | × Clear filters

Check	Last scan	Actions
Minimize wildcard use in Kubernetes Roles	1 day ago	
Minimize Wildcard Use in Kubernetes Cluster Roles (13 Findings)	1 day ago	

argocd-server upwindinternal / us-east-...	`argocd-server::7be4c420-8183...`	1 day ago	
cluster-admin upwindinternal / us-east-...	`cluster-admin::7be4c420-8183...`	1 day ago	
eks:addon-manager upwindinternal / us-east-...	`eks:addon-manager::7be4c420-...`	1 day ago	
eks:cloud-controller-manager upwindinternal / us-east-...	`eks:cloud-controller-manage...`	1 day ago	
eks:service-operations upwindinternal / us-east-...	`eks:service-operations::7be4...`	1 day ago	
system:controller:generic-gar... upwindinternal / us-east-...	`system:controller:generic-ga...`	1 day ago	
system:controller:horizontal-p... upwindinternal / us-east-...	`system:controller:horizontal...`	1 day ago	
system:controller:namespace... upwindinternal / us-east-...	`system:controller:namespace-...`	1 day ago	
system:controller:resourcequ... upwindinternal / us-east-...	`system:controller:resourcequ...`	1 day ago	

Kubernetes automatically assigns the default service account to pods if no specific service account is provided. Using default service accounts can complicate auditing and increase security risks by granting unnecessary permissions. To ensure better auditing and security, specific service accounts should be created and assigned to pods that require access to the Kubernetes API.

Last scan	Mar 31st, 2025 09:06
Control status	Enabled
Control name	5.1.5 Ensure that default service accounts are not actively used
Category	5.1. RBAC and Service Accounts
Framework	CIS Kubernetes Version 1.9

Findings

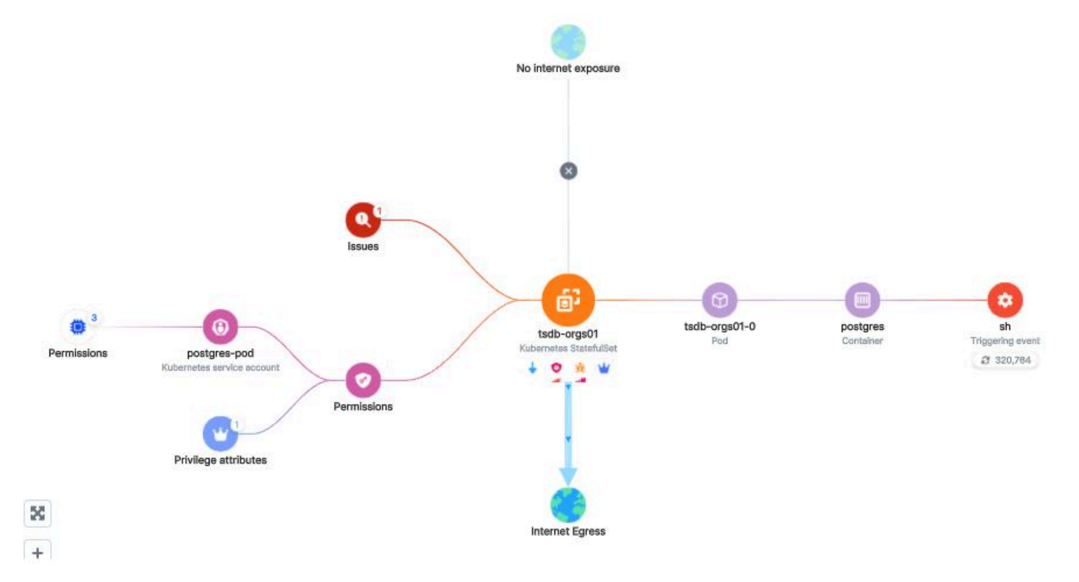
Search anything | Cloud account | Identifier | Resource kind | Resource name | × Clear filters

Check	Last scan	Actions
Ensure that default service accounts are not actively used (9 Findings)	1 day ago	

Resource name	Identifier	Last scan	Actions
default upwindinternal / us-east-1...	`default::amazon-guardduty::7...`	1 day ago	
default upwindinternal / us-east-1...	`default::default::7be4c420-8...`	1 day ago	
default upwindinternal / us-east-1...	`default::devops::7be4c420-81...`	1 day ago	
default upwindinternal / us-east-1...	`default::karpenter::7be4c420...`	1 day ago	
default upwindinternal / us-east-1...	`default::kube-node-lease::7b...`	1 day ago	
default upwindinternal / us-east-1...	`default::kube-public::7be4c4...`	1 day ago	
default upwindinternal / us-east-1...	`default::kube-system::7be4c4...`	1 day ago	
default upwindinternal / us-east-1...	`default::monitoring::7be4c42...`	1 day ago	
default upwindinternal / us-east-1...	`default::twingate::7be4c420-...`	1 day ago	

Runtime Threats:

A database management process has been detected creating a shell, which deviates from its standard operational profile. This behavior is unusual because database processes typically do not need to create shells for normal operations. This poses a significant threat as it may indicate attempts to execute unauthorized system commands by exploiting a vulnerability.



Detection details

Severity	High
Resource	tsdb-orgs01
Kind	Kubernetes StatefulSet
Path	upwindinternal / us-east-1 / production-us-east-1-eks /...
MITRE tactic	Execution > Command and Scripting Interpreter
Risk overview	
Source	Sensor
Threat ID	uwd-25ee1e738fca2a81

Resource risk analysis

- The resource has an active internet egress communication, last seen 2 minutes ago. View connections
- The resource has no additional threat detections.
- The resource has 7 critical-severity and 93 other-severity vulnerabilities.
- The resource is highly privileged due to its Kubernetes security context.

Show more

Last executed process

Name	sh
Command	`sh -c envdir "/run/etc/wal-e.d/env" wal-g wal-push "pg_wal/0000001B000073DF000000CD"`
State	Terminated
Process tree	View process tree

Runtime Exploit Funnel:

Prioritize vulnerabilities based on exploitability.

Productivity funnel

Total CVEs	Critical CVEs	Upwind Prioritized CVEs
4,216	98	5

93 investigations & patches saved x **4** hours per investigation =

372 Hours → 16 Days → 1 Month

95% Saving time

API Catalog:

Elasticsearch Bulk API Receiving Excessive Payloads — DevOps insights

Endpoint | 2 endpoints
CWE-770 | Allocation of Resources Without Limits or Throttling

Excessive Payloads Impacting Elasticsearch Performance

Multiple requests to the Elasticsearch Bulk API are exceeding the maximum payload size, resulting in `413 Request Entity Too Large` errors. This indicates potential issues with data ingestion processes and can significantly impact Elasticsearch performance and stability.

Evidence

Requests with route IDs `rt-27e3e597dbe70fda` and `rt-d1567338bc6aebb1` to the `/bulk` endpoint have `Content-Length` headers exceeding 100MB, leading to `413` errors in the responses.

Remediation

1. **Investigate data ingestion processes**: Identify the source of these large payloads and optimize data flow to reduce individual request sizes.
2. **Adjust Elasticsearch configuration**: Consider increasing the `http.max_content_length` setting in Elasticsearch to accommodate larger payloads if necessary. However, ensure this change aligns with your infrastructure capabilities and security policies.
3. **Implement data chunking**: Break down large payloads into smaller chunks and send them as separate requests to the Bulk API.
4. **Monitor Elasticsearch performance**: Regularly monitor Elasticsearch performance metrics, including indexing throughput and latency, to identify and address potential bottlenecks.

Unauthorized API Access — Security insights

Endpoint | POST /cluster-components/register
CWE-306 | Missing Authentication for Critical Function

Unauthorized Access to Sensitive API Endpoints

Multiple attempts to access the `/cluster-components/register` API endpoint resulted in 401 Unauthorized errors. This indicates potential attempts by unauthorized entities to register cluster components, posing a significant security risk.

Evidence

7 requests to `/cluster-components/register` resulted in 401 Unauthorized errors. Example request ID: `bea0be8c-e3c0-4083-834f-0a6bb9824c2d`

Remediation

Strengthen authentication mechanisms for the `/cluster-components/register` endpoint. Implement robust authorization checks to ensure only legitimate entities can register cluster components. Consider using multi-factor authentication and strong password policies.

Unauthorized Access to Sensitive API — Security insights

Endpoint | 7 endpoints
CWE-287 | Improper Authentication

Unauthorized Access to Sensitive API

Multiple requests to the `/host-components/register` API resulted in 401 Unauthorized errors. This indicates potential attempts to access the API without proper authorization. It is crucial to ensure that only authorized clients can access this sensitive endpoint.

Evidence

The following requests to `/host-components/register` resulted in 401 Unauthorized errors:

- Request with X-Request-Id: 68e9bd2d-4c69-4e27-9df3-cabc6fc9556b
- Request with X-Request-Id: cff891bf-4995-45ba-8227-ecbf0e809bb3
- Request with X-Request-Id: 66e76704-673f677d57bde55278d18a77
- Request with X-Request-Id: 66de9293-7fd9806019495e306b7420b2
- Request with X-Request-Id: 66d9dfc6-6b091ce24125a6463b94e48e
- Request with X-Amzn-Trace-Id: Root=1-66efad4b-2879f5bd689aa9397f63a7e0
- Request with X-Amzn-Trace-Id: Root=1-66f1b8c7-1ecb3f010945ad4b2a849ee4

Remediation

Review and strengthen the authorization mechanism for the `/host-components/register` API. Consider implementing the following:

- **Strong authentication:** Use robust authentication methods such as OAuth 2.0 or API keys to verify client identity.
- **Role-based access control (RBAC):** Implement RBAC to restrict access based on user roles and permissions.
- **Input validation:** Validate all input parameters to prevent injection attacks.
- **Regular security audits:** Conduct regular security audits to identify and address potential vulnerabilities.

Unauthorized Access to GCP Container Clusters Endpoint — Security insights

Endpoint | GET /v1/organizations/{-}/container/projects/{-}/clusters
CWE-287 | Improper Authentication

The `/v1/organizations/{orgId}/container/projects/{projectId}/clusters` API endpoint is vulnerable to unauthorized access.

Multiple requests to this endpoint, which is responsible for retrieving information about GCP container clusters, were successful despite lacking proper authentication headers. This indicates a critical security vulnerability as it allows unauthorized users to access sensitive information about your Kubernetes clusters, potentially leading to further attacks.

Evidence

Multiple requests to `/v1/organizations/{orgId}/container/projects/{projectId}/clusters` were made without any authentication headers, and some of these requests received successful responses (200 OK).

Remediation

Immediately implement proper authentication mechanisms for the `/v1/organizations/{orgId}/container/projects/{projectId}/clusters` API endpoint. Consider using industry-standard authentication methods such as API keys, OAuth 2.0, or JWT to ensure that only authorized clients can access this endpoint.

Compliance and Risk Prioritization

Frameworks Evaluated:

- CIS Benchmarks (CSPM).
- PCI, HIPAA, or NIST (optional extensions).

Risk Severity Scoring:

- Categorized by critical, high, medium, and low severity.

Attack Path Analysis:

- Map attack paths from misconfigurations and vulnerabilities to potential exploitation.

Next Steps and Recommendations

Immediate Remediation Suggestions:

- Actions to reduce public exposure.
- Patching guidance for critical CVEs.
- Misconfiguration fixes.

Ongoing Monitoring:

- Recommend continuous runtime and posture monitoring.

AWS MP Integration:

- Guidance on how to transact for $0/$1 via AWS MP.

✅ Final Output

This document provides a standardized Cloud Security Assessment format that clearly defines scope, deliverables, and actionable outcomes. It is designed for easy consumption by channel partners and end-customers, while highlighting Upwind's capabilities and cloud-native strengths.