



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.

The RDS database instance is directly accessible from the internet, exposing it to unauthorized access and potential data breaches, malicious attacks, or denial-of-service exploits. This misconfiguration violates security best practices, which recommend restricting access to trusted IP ranges, isolating the database in a private subnet within a VPC, and implementing strict network controls to reduce exposure and enhance security.

Severity Critical
Last seen 14 hours ago (Mar 31st, 2025 23:02)
Risk category Medium

Findings Cloud account Identifier Resource name Clear filters

Showing 1 row

Resource	Last scan	Ticket	Actions
upwind-demo-db AWS Demo / us-east-1	14 hours ago	--	

Two EKS API Endpoints have been identified as Internet Accessible.

The control plane API endpoint is a critical interface for managing your Amazon EKS cluster. By default, this endpoint can be accessed from anywhere on the internet. To ensure the security of your cluster, its recommended to restrict access to the API endpoint.

Leaving the EKS API endpoint publicly accessible with a CIDR range of 0.0.0.0/0 significantly increases the risk of unauthorized access. Malicious actors could exploit this access to steal sensitive data, disrupt critical services running in your EKS cluster, or deploy malicious resources that compromise your workloads.

Severity Medium
Last seen 14 hours ago (Mar 31st, 2025 23:02)
Risk category Medium

Findings Cloud account Identifier Resource name Clear filters

Showing 2 rows

Resource	Last scan	Ticket	Actions
only-for-you AWS Demo / us-east-1	14 hours ago	--	
upwind-cluster-1 AWS Demo / us-east-1	14 hours ago	--	

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.

The AWS Lambda function is configured with default encryption for environment variables at rest, which may not meet the organizations security standards for protecting sensitive data. Default encryption relies on AWS-managed keys and does not offer the additional control and monitoring available with customer-managed keys (CMKs). Best practices recommend using CMKs to encrypt sensitive environment variables to enhance security, enable key rotation, and meet compliance requirements.

Severity Medium
Last seen 14 hours ago (Mar 31st, 2025 23:02)
Risk category Low

Findings Cloud account Identifier Resource name Clear filters

Showing 6 rows

Resource	Last scan	Ticket	Actions
upwind-cs-lambda-ucsc-5b3abda632ae68dd AWS Demo / eu-central-1	14 hours ago	--	
upwind-cs-lambda-ucsc-af340fc6ba350e51 AWS Demo / us-east-1	14 hours ago	--	
upwind-cs-lambda-ucsc-5b3abda632ae68dd AWS Demo / eu-central-1	14 hours ago	--	
upwind-cs-lambda-ucsc-af340fc6ba350e51 AWS Demo / us-east-1	14 hours ago	--	
upwind-cs-updater-ucsc-5b3abda632ae68dd AWS Demo / eu-central-1	14 hours ago	--	
upwind-cs-updater-ucsc-af340fc6ba350e51 AWS Demo / us-east-1	14 hours ago	--	

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

vulnerable-apache 8 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 8 Findings
recommendationservice 14 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 14 Findings
currencyservice 17 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 17 Findings
adservice 4 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 4 Findings
checkoutservice 7 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 7 Findings
productcatalogservice 7 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 7 Findings
cartservice 4 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 4 Findings
coredns 5 Findings AWS Demo / us-east-1 / upwind-cluster-1 / kube-system	Kubernetes Deployment 5 Findings
frontend 5 Findings AWS Demo / us-east-1 / upwind-cluster-1 / microservices	Kubernetes Deployment 5 Findings
jenkins 4 Findings AWS Demo / us-east-1 / upwind-cluster-1 / jenkins	Kubernetes Deployment 4 Findings
telegrafmat 2 Findings AWS Demo / us-east-1 / upwind-cluster-1 / rat	Kubernetes Deployment 2 Findings

There are 11 detected insecure S3 buckets with Internet Exposure.

demo-vuh-lambda-code AWS Demo / us-east-1	AWS S3 Bucket	
cf-templates-mfwskwdb-us-east-1 AWS Demo / us-east-1	AWS S3 Bucket	
use-terraform-states-142380711371 AWS Demo / us-east-1	AWS S3 Bucket	
tagging-log-a7c25d0-acaa-1ef-a954-12707b2b873b AWS Demo / us-east-1	AWS S3 Bucket	
upwind-cloudtrail-bucket-demo AWS Demo / us-east-1	AWS S3 Bucket	
bucket-demo-s3aa AWS Demo / us-east-1	AWS S3 Bucket	
upwind-springshell-bucket-demo AWS Demo / us-east-1	AWS S3 Bucket	
eks-updater-bucket-20250218-unique123 AWS Demo / us-east-1	AWS S3 Bucket	
bucket-data-eu AWS Demo / eu-central-1	AWS S3 Bucket	
cf-templates-mfwskwdb-eu-central-1 AWS Demo / eu-central-1	AWS S3 Bucket	

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.

CVE-2021-46848 (Libsani Data Extrition) 2 Findings	2 Findings	2 resources	2 images
CVE-2021-42013 1 Finding	1 finding	1 resource	1 image
CVE-2021-44790 (Apache HTTP Server Remote Code ... 1 Finding	1 finding	1 resource	1 image
CVE-2023-44487 31 Findings	31 findings	8 resources	8 images
CVE-2022-41721 6 Findings	6 findings	6 resources	6 images
CVE-2011-84829 4 Findings	4 findings	4 resources	4 images
CVE-2011-4116 1 Finding	1 finding	1 resource	1 image
CVE-2019-18276 (Bash Privilege Escalation) 1 Finding	1 finding	1 resource	1 image
CVE-2021-41773 1 Finding	1 finding	1 resource	1 image
CVE-2022-2309 1 Finding	1 finding	1 resource	1 image
CVE-2022-24975 1 Finding	1 finding	1 resource	1 image
CVE-2022-2880 1 Finding	1 finding	1 resource	1 image
CVE-2022-28948 (Go-Ventl Denial Of Service) 1 Finding	1 finding	1 resource	1 image
CVE-2022-3715 (Base Heap Overflow) 1 Finding	1 finding	1 resource	1 image
CVE-2023-31484 (CPAN Perl SSL Verification Bypass) 1 Finding	1 finding	1 resource	1 image
CVE-2023-32559 (NodeJS Module Policy Privilege Escal... 1 Finding	1 finding	1 resource	1 image
CVE-2023-36332 1 Finding	1 finding	1 resource	1 image
CVE-2024-24066 1 Finding	1 finding	1 resource	1 image
CVE-2024-25062 (Ibmtd Denial Of Service) 1 Finding	1 finding	1 resource	1 image
CVE-2025-27113 1 Finding	1 finding	1 resource	1 image

Data Insights:

Detect generative AI platforms being used.

None found at this time.

Identify suspicious CloudTrail activity:

Access to an S3 Bucket was found

A public access was configured to an S3 bucket by deleting Public Access Block. Removing this block significantly increases the risk of unauthorized access to the buckets contents. Public access blocks are crucial for preventing accidental or malicious exposure of sensitive data stored in S3 buckets. The absence of these blocks reduces the security layer, making it easier for attackers to access and exploit the data. Detecting the deletion of public access blocks is essential to identify and mitigate unauthorized access attempts that could lead to severe security breaches, including data loss, service disruption, and data exfiltration.

CloudTrail log

The CloudTrail log entry shows that the PutBucketPublicAccessBlock event was initiated from the source address 67.168.33.183 using the user agent Mozilla/5.0 (Macintosh; Intel Mac OS X 18_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36 initiated by josh.lay@upwind.io (AccessControl) on arn:aws:s3::eks-updater-bucket-20250218-unique123. Initiated by josh.lay@upwind.io (AccessControl) on arn:aws:s3::eks-updater-bucket-20250218-unique123.

Detection details

Severity Critical
Resource arn:aws:s3::eks-updater-bucket-20250218-unique123
Kind AWS S3 Bucket
Path AWS Demo
Triggering policy Management Exposure Violations
MITRE tactic Collection - Data from Cloud Storage
Risk overview High
Source Logs
Threat ID urid-82816a7770b9da08

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.

The root user account is the most privileged user in an AWS account. Multi-factor Authentication (MFA) adds an extra layer of protection on top of a username and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their username and password as well as for an authentication code from their AWS MFA device.

Note: When virtual MFA is used for root accounts, it is recommended that the device used is NOT a personal device, but rather a dedicated mobile device (tablet or phone) that is managed to be kept charged and secured independent of any individual personal devices. (Non-personal virtual MFA) This lessens the risks of losing access to the MFA due to device loss, device trade-in or if the individual owning the device is no longer employed at the company.

Last scan Mar 31st, 2025 09:09
Control status Enabled
Control name 1.5 Ensure MFA is enabled for the 'root' user account
Category 1. Identity and Access Management
Framework CIS Amazon Web Services foundations Version 2.0 | Revision 1.0

Findings Cloud account Identifier Resource kind Resource name Clear filters

Check 1 day ago Actions

Resource name	Identifier	Last scan	Actions
<root_account> upwindinternal / us-east-1	arn:aws:iam::627244208106:root	1 day ago	

The root user account is the most privileged user in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their user name and password as well as for an authentication code from their AWS MFA device. For Level 2, it is recommended that the root user account be protected with a hardware MFA.

Last scan Mar 31st, 2025 09:09
Control status Enabled
Control name 1.6 Ensure hardware MFA is enabled for the 'root' user account
Category 1. Identity and Access Management
Framework CIS Amazon Web Services foundations Version 2.0 | Revision 1.0

Findings Cloud account Identifier Resource kind Resource name Clear filters

Check 1 day ago Actions

Resource name	Identifier	Last scan	Actions
<root_account> upwindinternal / us-east-1	arn:aws:iam:us-east-1:6272442...	1 day ago	

16 High Findings have been found for various CIS Benchmarks.

This check verifies that AWS-managed IAM policies attached to users, groups, or roles do not grant administrative-level access by reviewing policies with "Action": "*" over "Resource": "*". AWS-managed policies are pre-defined by AWS, and although convenient, they should be carefully reviewed to avoid overly permissive access. This check reinforces best practices by ensuring that AWS-managed policies adhere to the principle of least privilege, reducing the risk of unintended access by limiting access to only necessary permissions.

Last scan Mar 31st, 2025 09:09
Control status Enabled
Control name 1.16 Ensure IAM policies that allow full "*" administrative privileges are not attached
Category 1. Identity and Access Management
Framework CIS Amazon Web Services foundations Version 2.0 | Revision 1.0

Findings Cloud account Identifier Resource kind Resource name Clear filters

Check 1 day ago Actions

Resource name	Identifier	Last scan	Actions
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneStream-HTTPSAcces... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
production-us-east-1-eks-nod... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to remote server administration ports, such as SSH to port 22 and RDP to port 3389.

Control status Enabled
Control name 5.3 Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports
Category 5. Networking
Framework CIS Amazon Web Services foundations Version 2.0 | Revision 1.0

Findings Cloud account Identifier Resource kind Resource name Clear filters

Check 1 day ago Actions

Resource name	Identifier	Last scan	Actions
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneStream-HTTPSAcces... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
NeptuneToOpenSearchIntegra... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	
production-us-east-1-eks-nod... upwindinternal / us-east-1	arn:aws:ec2:us-east-1:627244...	1 day ago	

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: 5 Findings

Search anything: Cloud account Identifier Resource kind Resource name Clear filters

Check: Last scan Actions

Resource name	Identifier	Last scan	Actions
ip-10-11-19-178.ec2.internal	0697b428-7ae4-4c5c-89c0-e9e4...	1 day ago	
ip-10-11-22-46.ec2.internal	65615162-350e-42b7-8f8a-f9a7...	1 day ago	
ip-10-11-20-189.ec2.internal	8d326cd1-ce63-4a9c-e2d1-b673...	1 day ago	
ip-10-11-31-224.ec2.internal	c667e165-d2e7-4321-bdc6-33ee...	1 day ago	
ip-10-11-60-198.ec2.internal	e01ce9ba-144e-47e7-99d9-3289...	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: 20 Findings

Search anything: Cloud account Identifier Resource kind Resource name Clear filters

Check: Last scan Actions

Resource name	Identifier	Last scan	Actions
admin	admin::7be4c420-8183-4381-bf...	1 day ago	
argocd-notifications-controller	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: 13 Findings

Search anything: Cloud account Identifier Resource kind Resource name Clear filters

Check: Last scan Actions

Resource name	Identifier	Last scan	Actions
argocd-server	argocd-server::7be4c420-8183...	1 day ago	
cluster-admin	cluster-admin::7be4c420-8183...	1 day ago	
eks:addon-manager	eks:addon-manager::7be4c420-...	1 day ago	
eks:cloud-controller-manager	eks:cloud-controller-manage...	1 day ago	
eks:service-operations	eks:service-operations::7be4...	1 day ago	
system:controller-generic-gar...	system:controller:generic-ga...	1 day ago	
system:controller-horizontal-p...	system:controller:horizontal...	1 day ago	
system:controller:namespace...	system:controller:namespace...	1 day ago	
system:controller:resourcequ...	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: 9 Findings

Search anything: Cloud account Identifier Resource kind Resource name Clear filters

Check: Last scan Actions

Resource name	Identifier	Last scan	Actions
default	default::amazon-guardduty::7...	1 day ago	
default	default::default::7be4c420-8...	1 day ago	
default	default::devops::7be4c420-81...	1 day ago	
default	default::karpenter::7be4c420...	1 day ago	
default	default::kubernetes-lease::7b...	1 day ago	
default	default::kubernetes-public::7be4c4...	1 day ago	
default	default::kubernetes-system::7be4c4...	1 day ago	
default	default::monitoring::7be4c42...	1 day ago	
default	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: High

Source: Sensor

Threat ID: uwd-25ee1e739fca2a81

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

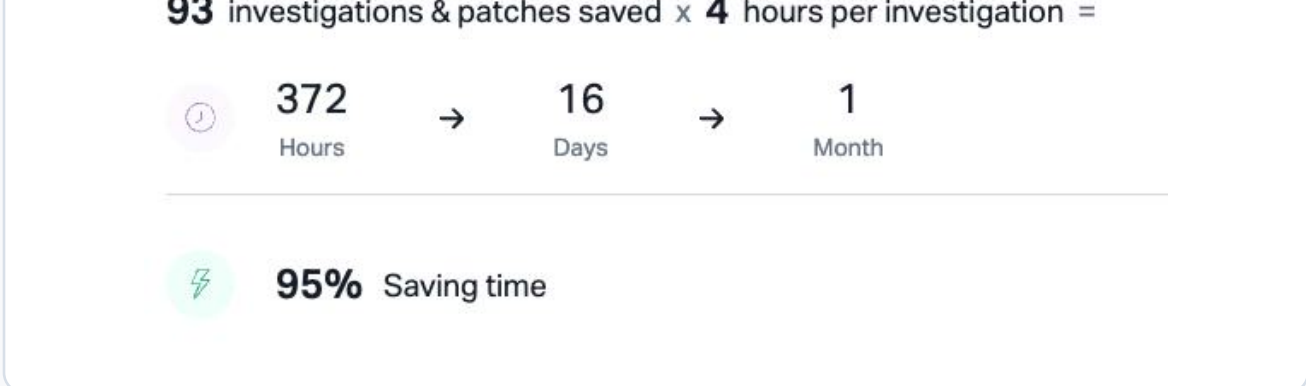
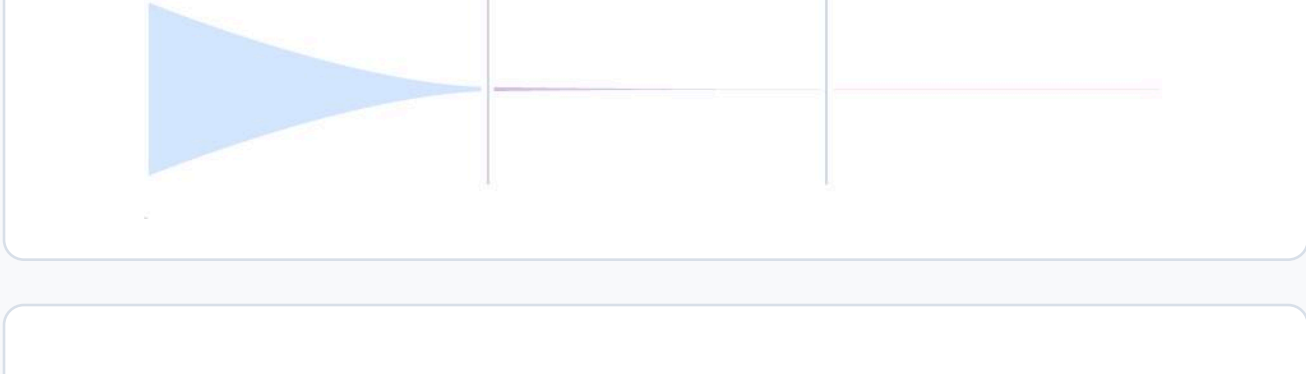
Command: sh -c envdir "/run/etc/wal-e.d/env" wal-g wal-push "pg_wal/0000001B000073DF000000CD"

State: Terminated

Process tree: [View process tree](#)

Runtime Exploit Funnel:

Prioritize vulnerabilities based on exploitability.



API Catalog:

Elasticsearch Bulk API Receiving Excessive Payloads [DevOps insights](#)

Endpoint: `POST /_bulk`

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-27e3e597d7b7fda` and `rt-d1567338bc6aebb1` to the `/_bulk` endpoint have `Content-Length` headers exceeding 100MB, leading to `413` errors in the responses.

Remediation

- Investigate data ingestion processes: Identify the source of these large payloads and optimize data flow to reduce individual request sizes.
- 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.
- Implement data chunking: Break down large payloads into smaller chunks and send them as separate requests to the Bulk API.
- 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-e3c0-4883-834f-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: `GET /v1/organizations/{orgId}/container/projects/{projectId}/clusters`

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-cabcf6c9556b`
- Request with X-Request-Id: `cf891bf1-4995-45ba-8227-ecbf0e809bb3`
- Request with X-Request-Id: `66e76704-673f677d57be55278d18a77`
- Request with X-Request-Id: `66de9293-7fd9806019495e306b7420b2`
- Request with X-Request-Id: `66d9dfc6-6b091ce24125a6463b94e48e`
- Request with X-Amzn-Trace-Id: `Root=1-66efad4b-2879750d69a8a397f63a7e0`
- 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/{orgId}/container/projects/{projectId}/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.