Continuous Monitoring and Threat Detection on AWS .

1. Problem Statement.

A SaaS organization required continuous threat detection across its AWS environment. Limited monitoring and delayed alerting increased security risk. A centralized detection framework was required.

I implemented this solution to improve visibility into API activity, network traffic, security misconfigurations, and threat indicators, while enabling rapid incident response.

2. Objectives

The primary objectives were to:

- Enable continuous security monitoring
- Detect suspicious IAM and API activity
- Identify network-based threats
- Centralize security findings
- Automate low-severity remediation
- Improve incident response readiness

3. Security Monitoring Architecture

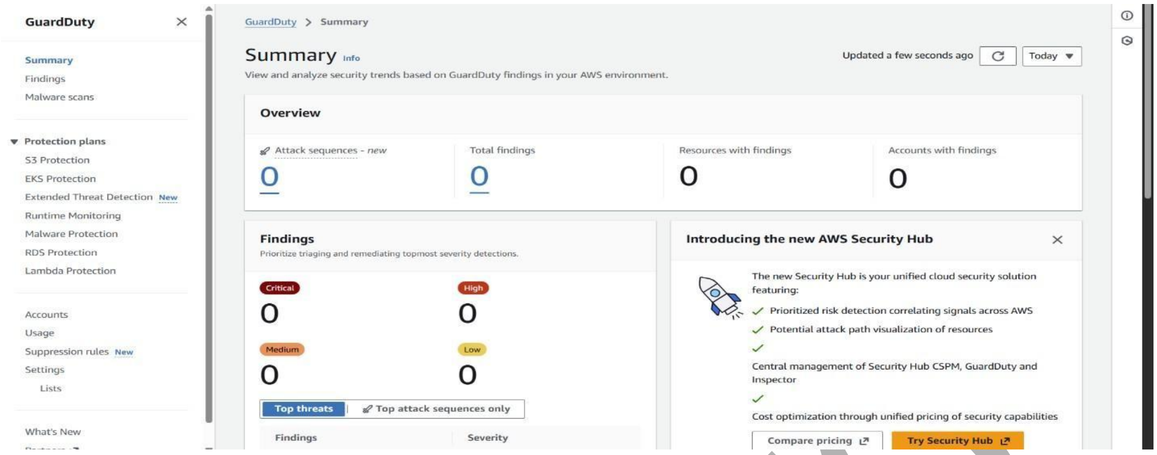
The monitoring framework was built using the following AWS services:

- Amazon GuardDuty
- AWS CloudTrail
- AWS Security Hub
- VPC Flow Logs
- Amazon CloudWatch
- Amazon SNS
- AWS Lambda

All security events and findings were centralized for analysis and response.

4. GuardDuty Configuration

I enabled Amazon GuardDuty to monitor:


GuardDuty ✕

Summary
Findings
Malware scans

▼ Protection plans
S3 Protection
EKS Protection
Extended Threat Detection New
Runtime Monitoring
Malware Protection
RDS Protection
Lambda Protection

Accounts
Usage
Suppression rules New
Settings
Lists

What's New

GuardDuty > Summary

Summary Info

View and analyze security trends based on GuardDuty findings in your AWS environment.

Updated a few seconds ago | Today ▼

Overview

Attack sequences - new	Total findings	Resources with findings	Accounts with findings
0	0	0	0

Findings

Prioritize triaging and remediating topmost severity detections.

Critical	High
0	0
Medium	**Low**
0	0

Top threats | Top attack sequences only

Findings | Severity

Introducing the new AWS Security Hub ✕

The new Security Hub is your unified cloud security solution featuring:

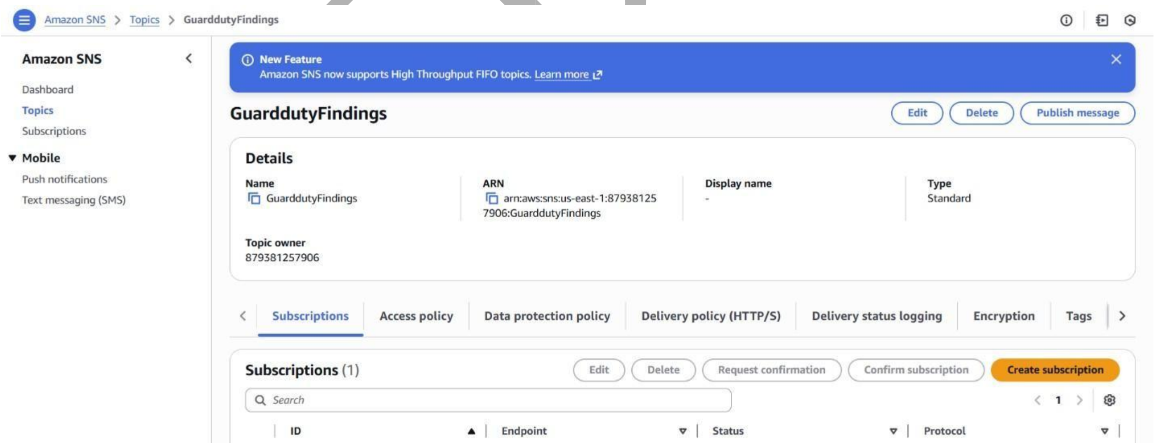
✓ Prioritized risk detection correlating signals across AWS
✓ Potential attack path visualization of resources
✓ Central management of Security Hub CSPM, GuardDuty and Inspector
✓ Cost optimization through unified pricing of security capabilities

Compare pricing | Try Security Hub

- CloudTrail logs
- VPC Flow Logs
- DNS activity

GuardDuty was configured to generate findings for suspicious behavior and compromised credentials.

An SNS topic was created to distribute GuardDuty alerts to the security team.


Amazon SNS > Topics > GuarddutyFindings

Amazon SNS

Dashboard
Topics
Subscriptions
▼ Mobile
Push notifications
Text messaging (SMS)

New Feature
Amazon SNS now supports High Throughput FIFO topics. Learn more

GuarddutyFindings

Edit | Delete | Publish message

Details

Name	ARN	Display name	Type
GuarddutyFindings	arn:aws:sns:us-east-1:87938125 7906:GuarddutyFindings	-	Standard

Topic owner
879381257906

Subscriptions | Access policy | Data protection policy | Delivery policy (HTTP/S) | Delivery status logging | Encryption | Tags

Subscriptions (1)

Edit | Delete | Request confirmation | Confirm subscription | Create subscription

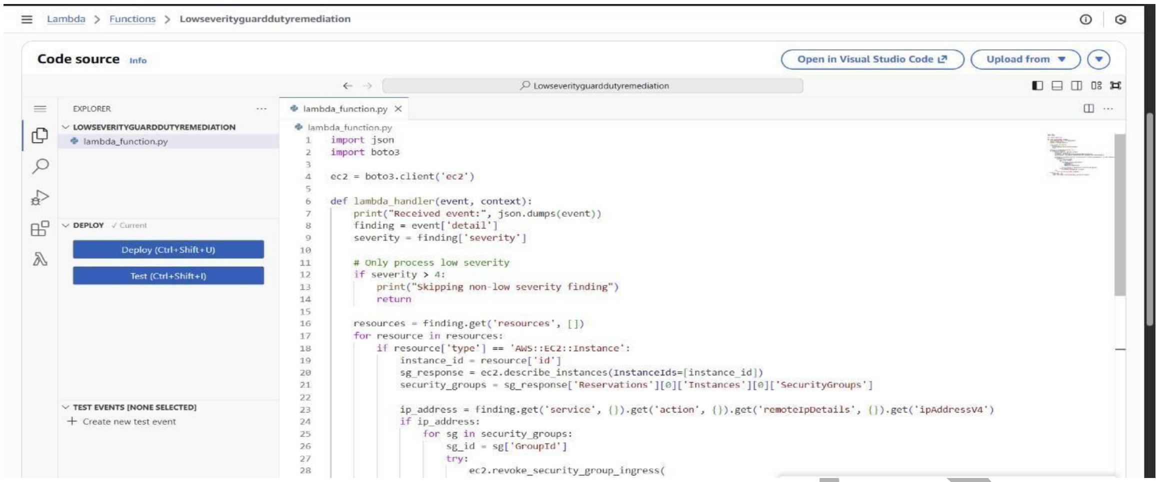
Search

ID	Endpoint	Status	Protocol

Sample findings were generated to validate alert delivery.

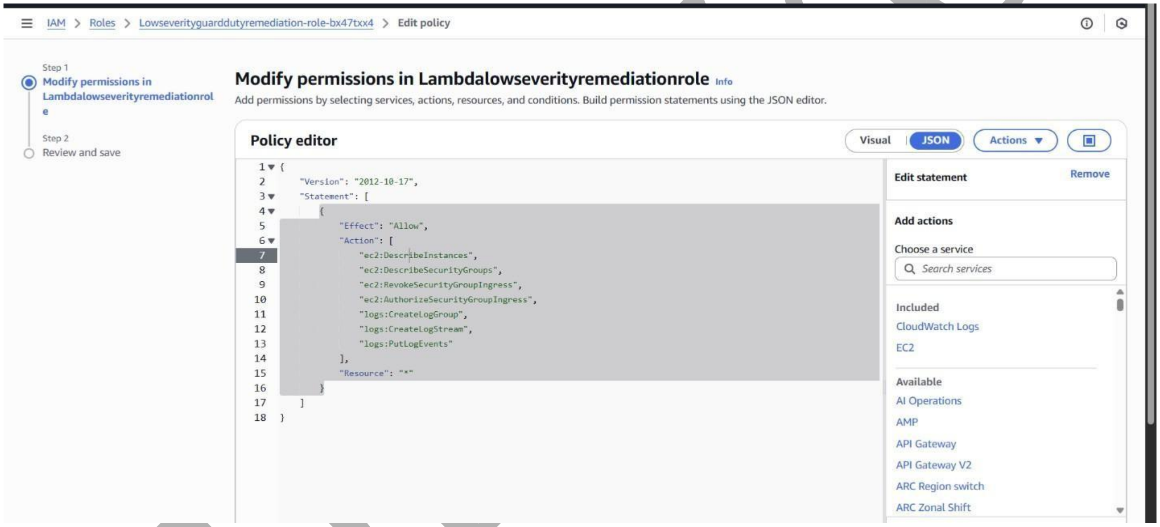
5. Automated Remediation for Low-Severity Findings

I implemented automated remediation for low-severity findings using AWS Lambda.


Lambda > Functions > Lowseverityguarddutyremediation
Code source Info
Open in Visual Studio Code
Upload from
Lowseverityguarddutyremediation
EXPLORER
LOWSEVERITYGUARDDUTYREMEDIATION
lambda_function.py
DEPLOY ✓ Current
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
TEST EVENTS [NONE SELECTED]
+ Create new test event
lambda_function.py

```python
import json
import boto3

ec2 = boto3.client('ec2')

def lambda_handler(event, context):
    print("Received event:", json.dumps(event))
    finding = event['detail']
    severity = finding['severity']

    # Only process low severity
    if severity > 4:
        print("Skipping non-low severity finding")
        return

    resources = finding.get('resources', [])
    for resource in resources:
        if resource['type'] == 'AWS::EC2::Instance':
            instance_id = resource['id']
            sg_response = ec2.describe_instances(InstanceIds=[instance_id])
            security_groups = sg_response['Reservations'][0]['Instances'][0]['SecurityGroups']

            ip_address = finding.get('service', {}).get('action', {}).get('remoteIpDetails', {}).get('ipAddressV4')
            if ip_address:
                for sg in security_groups:
                    sg_id = sg['GroupId']
                    try:
                        ec2.revoke_security_group_ingress(
```

A dedicated IAM role was created with permissions to modify tagged security groups.



Modify permissions in Lambdalowseverityremediationrole Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

```json
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:RevokeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupIngress",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

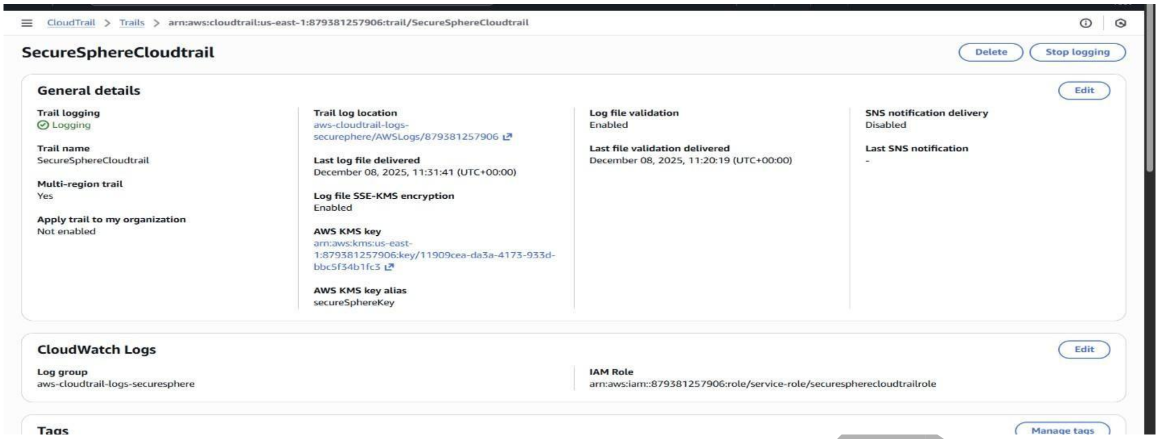
In production environments, permissions would be restricted to specific ARNs and tagged resources to enforce least privilege.

A Lambda function was deployed and triggered using EventBridge when low-severity findings occurred.

6. CloudTrail and API Monitoring

I configured a centralized CloudTrail trail to capture all AWS API activity.

Logs were forwarded to CloudWatch Logs for near real-time analysis.


CloudTrail > Trails > arn:aws:cloudtrail:us-east-1:879381257906:trail/SecureSphereCloudtrail
Delete
Stop logging
Edit
Edit
Manage tags

SecureSphereCloudtrail

General details

Trail logging
Logging

Trail name
SecureSphereCloudtrail

Multi-region trail
Yes

Apply trail to my organization
Not enabled

Trail log location
aws-cloudtrail-logs-securephere/AWSLogs/879381257906

Last log file delivered
December 08, 2025, 11:31:41 (UTC+00:00)

Log file SSE-KMS encryption
Enabled

AWS KMS key
arn:aws:kms:us-east-1:879381257906:key/11909cea-da3a-4173-933d-bbc5f34b1fc3

AWS KMS key alias
secureSphereKey

Log file validation
Enabled

Last file validation delivered
December 08, 2025, 11:20:19 (UTC+00:00)

SNS notification delivery
Disabled

Last SNS notification
-

CloudWatch Logs

Log group
aws-cloudtrail-logs-securesphere

IAM Role
arn:aws:iam::879381257906:role/service-role/securespherecloudtrailrole

Tags

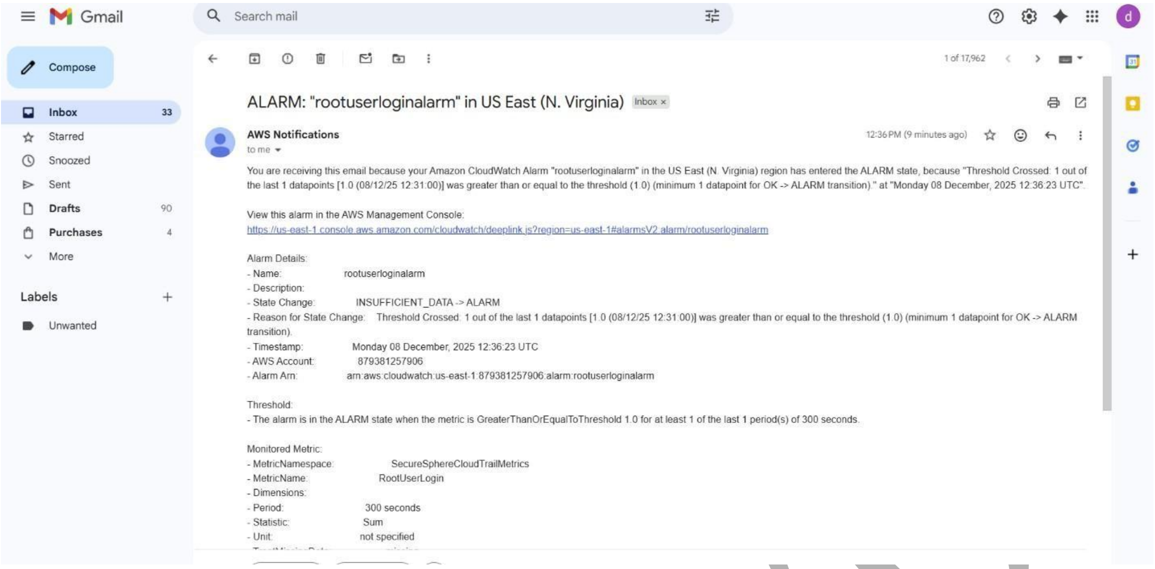
SNS notifications were configured to alert the security team when suspicious events were detected.

7. CloudWatch Metric Filters and Alarms

Metric filters were created for critical security events, including:

- Unauthorized API calls
- Console login without MFA
- Root account usage - IAM policy changes
- CloudTrail configuration changes
- AWS Config changes
- S3 bucket policy changes

Each metric was linked to a CloudWatch alarm and SNS notification.



ALARM: "rootuserloginalarm" in US East (N. Virginia)

AWS Notifications
to me

12:36 PM (9 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "rootuserloginalarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [1.0 (08/12/25 12:31:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Monday 08 December, 2025 12:36:23 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/rootuserloginalarm

Alarm Details:
- Name: rootuserloginalarm
- Description:
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [1.0 (08/12/25 12:31:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Monday 08 December, 2025 12:36:23 UTC
- AWS Account: 879381257906
- Alarm Arn: arn:aws:cloudwatch:us-east-1:879381257906:alarm:rootuserloginalarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:
- MetricNamespace: SecureSphereCloudTrailMetrics
- MetricName: RootUserLogin
- Dimensions:
- Period: 300 seconds
- Statistic: Sum
- Unit: not specified

Manual test events were generated to validate metric activation and alert delivery.

8. Security Hub Integration

AWS Security Hub was enabled to aggregate findings from multiple services, including:

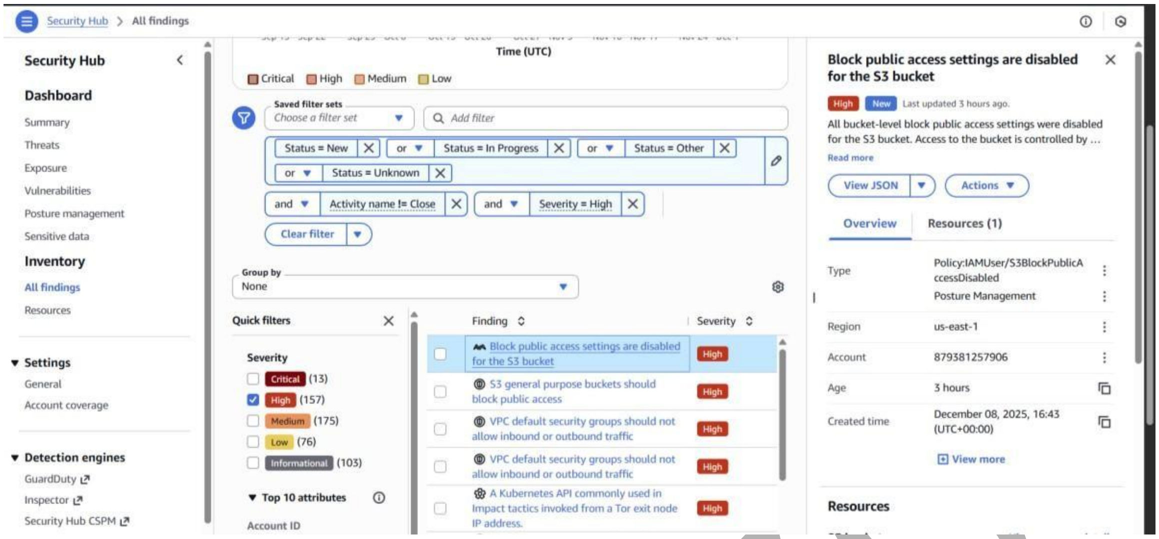
- AWS Config
- GuardDuty
- Inspector
- Macie

Security standards were reviewed and findings were prioritized based on risk level.

9. S3 Security Monitoring and Remediation

A test S3 bucket was created with public access enabled to validate detection capabilities.

GuardDuty and Security Hub detected the misconfiguration.



Status = New or Status = In Progress or Status = Other or Status = Unknown
and Activity name != Close and Severity = High

Group by: None

Severity
☐ Critical (13)
☑ High (157)
☐ Medium (175)
☐ Low (76)
☐ Informational (103)

Finding	Severity
Block public access settings are disabled for the S3 bucket	High
S3 general purpose buckets should block public access	High
VPC default security groups should not allow inbound or outbound traffic	High
VPC default security groups should not allow inbound or outbound traffic	High
A Kubernetes API commonly used in Impact tactics invoked from a Tor exit node IP address.	High

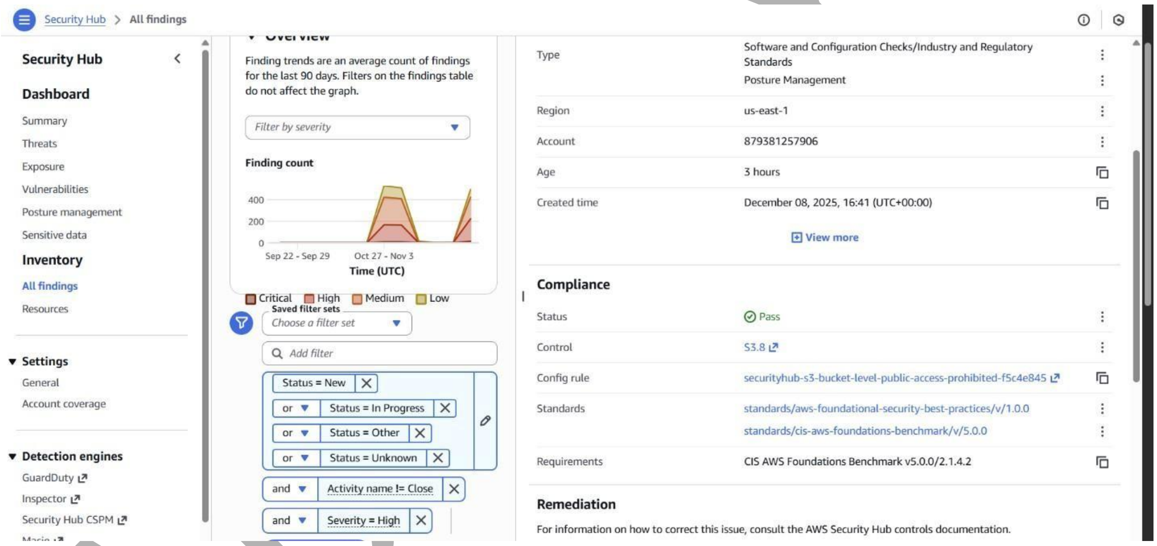
Block public access settings are disabled for the S3 bucket

High New Last updated 3 hours ago.

All bucket-level block public access settings were disabled for the S3 bucket. Access to the bucket is controlled by ... Read more

Type	Policy:IAMUser/S3BlockPublicAccessDisabled Posture Management
Region	us-east-1
Account	879381257906
Age	3 hours
Created time	December 08, 2025, 16:43 (UTC+00:00)

Public access was removed to remediate the high severity finding.



Finding trends are an average count of findings for the last 90 days. Filters on the findings table do not affect the graph.

Status = New or Status = In Progress or Status = Other or Status = Unknown
and Activity name != Close and Severity = High

Type	Software and Configuration Checks/Industry and Regulatory Standards Posture Management
Region	us-east-1
Account	879381257906
Age	3 hours
Created time	December 08, 2025, 16:41 (UTC+00:00)

Compliance

Status	Pass
Control	S3.8
Config rule	securityhub-s3-bucket-level-public-access-prohibited-f5c4e845
Standards	standards/aws-foundational-security-best-practices/v/1.0.0 standards/cis-aws-foundations-benchmark/v/5.0.0
Requirements	CIS AWS Foundations Benchmark v5.0.0/2.1.4.2

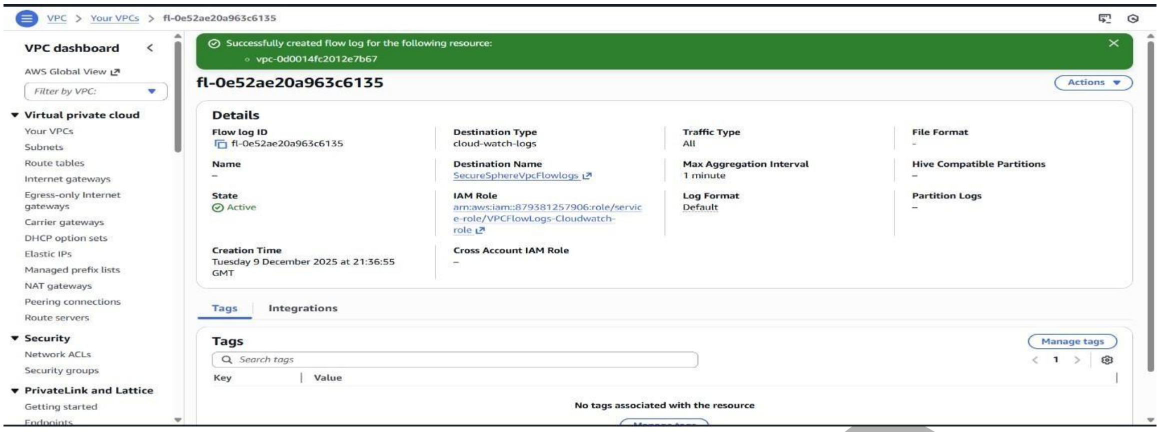
Remediation

For information on how to correct this issue, consult the AWS Security Hub controls documentation.

Server access logging was enabled to improve audit visibility.

10. Network Security and VPC Flow Logs

VPC Flow Logs were enabled for all production VPCs.



Successfully created flow log for the following resource:
- vpc-0d0014fc2012e7b67

fl-0e52ae20a963c6135

Details

Flow log ID	Destination Type	Traffic Type	File Format
fl-0e52ae20a963c6135	cloud-watch-logs	All	-
Name	**Destination Name**	**Max Aggregation Interval**	**Hive Compatible Partitions**
–	SecureSphereVpcFlowlogs	1 minute	–
State	**IAM Role**	**Log Format**	**Partition Logs**
Active	arn:aws:iam::879381257906:role/service-role/VPCFlowLogs-Cloudwatch-role	Default	–
Creation Time	**Cross Account IAM Role**		
Tuesday 9 December 2025 at 21:36:55 GMT	–		

Tags

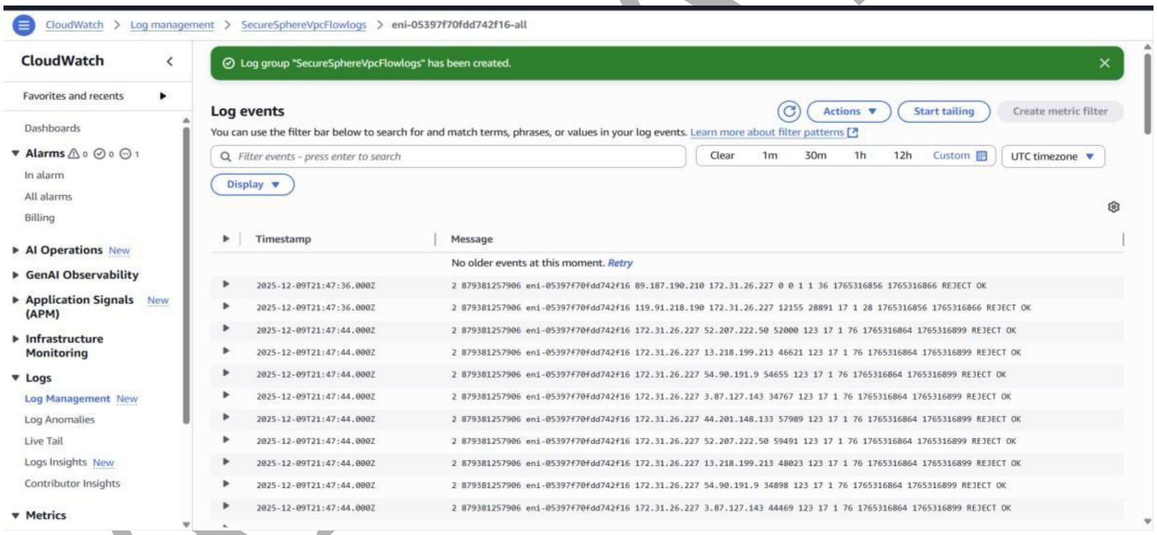
Key	Value

No tags associated with the resource

Logs were sent to CloudWatch Log Groups for monitoring and analysis.

An EC2 instance was used to generate network traffic for validation.

Both ACCEPT and REJECT traffic was verified in log streams.



Log group "SecureSphereVpcFlowlogs" has been created.

Log events

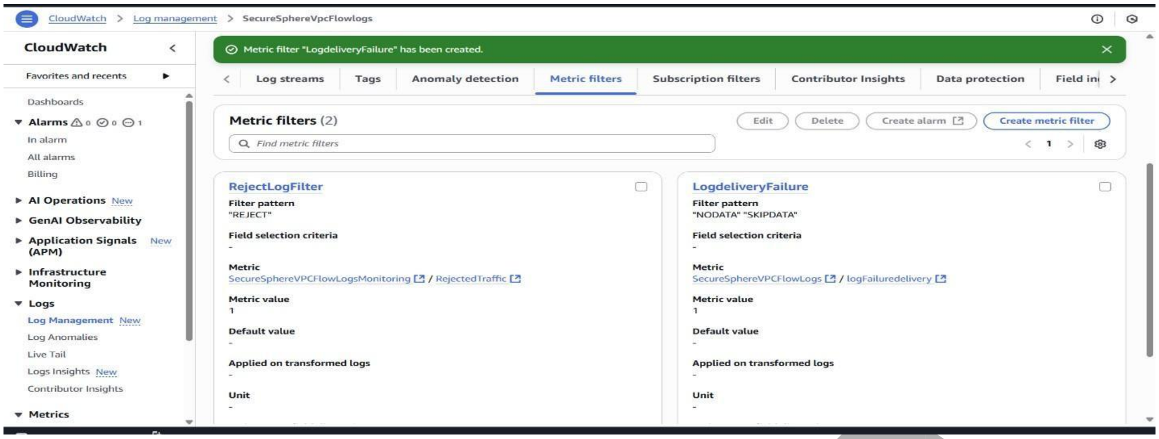
You can use the filter bar below to search for and match terms, phrases, or values in your log events. Learn more about filter patterns

```
Timestamp                 Message
                          No older events at this moment. Retry
2025-12-09T21:47:36.000Z  2 879381257906 eni-05397f70fdd742f16 89.187.190.210 172.31.26.227 0 0 1 1 36 1765316856 1765316866 REJECT OK
2025-12-09T21:47:36.000Z  2 879381257906 eni-05397f70fdd742f16 119.91.218.190 172.31.26.227 12155 28891 17 1 28 1765316856 1765316866 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 52.207.222.50 52000 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 13.218.199.213 46621 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 54.90.191.9 54655 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 3.87.127.143 34767 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 44.201.148.133 57989 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 52.207.222.50 59491 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 13.218.199.213 48023 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 54.90.191.9 34898 123 17 1 76 1765316864 1765316899 REJECT OK
2025-12-09T21:47:44.000Z  2 879381257906 eni-05397f70fdd742f16 172.31.26.227 3.87.127.143 44469 123 17 1 76 1765316864 1765316899 REJECT OK
```

11. Network Monitoring Alerts

Metric filters were created for:

- Rejected traffic events
- Log delivery failures



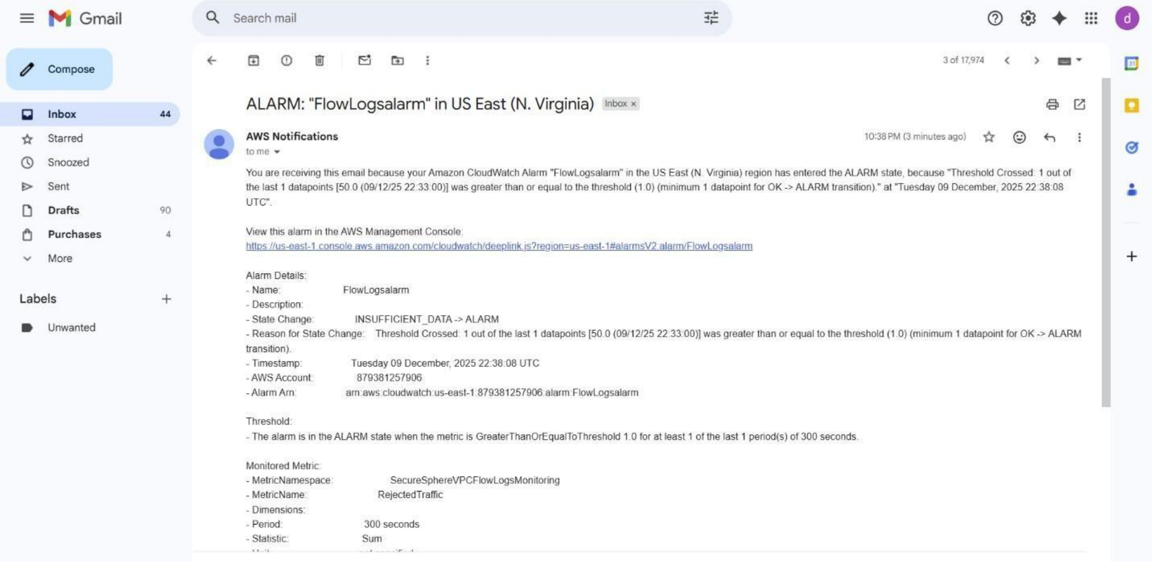
Metric filter "LogdeliveryFailure" has been created.

Metric filters (2)

RejectLogFilter
Filter pattern
"REJECT"
Field selection criteria
-
Metric
SecureSphereVPCFlowLogsMonitoring / RejectedTraffic
Metric value
1
Default value
-
Applied on transformed logs
-
Unit
-

LogdeliveryFailure
Filter pattern
"NODATA" "SKIPDATA"
Field selection criteria
-
Metric
SecureSphereVPCFlowLogs / logFailuredelivery
Metric value
1
Default value
-
Applied on transformed logs
-
Unit
-

CloudWatch alarms were configured to notify the security team of network anomalies.



ALARM: "FlowLogsalarm" in US East (N. Virginia)

AWS Notifications
to me
10:38 PM (3 minutes ago)

You are receiving this email because your Amazon CloudWatch Alarm "FlowLogsalarm" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [50.0 (09/12/25 22:33:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 09 December, 2025 22:38:08 UTC".

View this alarm in the AWS Management Console:
https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/FlowLogsalarm

```
Alarm Details:
- Name:                 FlowLogsalarm
- Description:
- State Change:         INSUFFICIENT_DATA -> ALARM
- Reason for State Change:    Threshold Crossed: 1 out of the last 1 datapoints [50.0 (09/12/25 22:33:00)] was greater than or equal to the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp:            Tuesday 09 December, 2025 22:38:08 UTC
- AWS Account:          879381257906
- Alarm Arn:            arn:aws:cloudwatch:us-east-1:879381257906:alarm:FlowLogsalarm

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:
- MetricNamespace:           SecureSphereVPCFlowLogsMonitoring
- MetricName:                RejectedTraffic
- Dimensions:
- Period:                    300 seconds
- Statistic:                 Sum
```

12. Documentation and Governance

All configurations, alerting workflows, and remediation procedures were documented.

This documentation supported audits, incident response, and knowledge transfer.

13. Outcomes and Impact

This implementation delivered the following improvements:

- Centralized threat detection - Faster incident response
- Improved network visibility

- Automated remediation
- Reduced security misconfigurations
- Enhanced compliance monitoring

14. Conclusion

This monitoring and threat detection framework provides comprehensive visibility into cloud security risks.

Through centralized logging, automated response, and continuous analysis, the environment supports secure and resilient operations.