

**Analysing the Vulnerabilities
and Behaviour of a Linux-based EC2
Instance using Amazon Inspector**

Table of contents

1.	Architectural Overview of Amazon Inspector Assessment Runs
2.	Installing SSM and Inspector agent on the EC2 instance
3.	Troubleshooting the Unknown Agent status with AWS Support Team
4.	Monitoring Amazon Inspector Using Amazon CloudWatch
5.	Monitoring Amazon Inspector Using Amazon CloudTrail
6.	Report findings and Remediation
7.	Lambda for E-mail notification Automatic Updation
8.	Pricing Incurred in Task

1. Architectural overview of the Amazon Inspector assessment runs

Amazon Inspector performs security assessments of Amazon EC2 instances by using AWS managed rules packages such as the Common Vulnerabilities and Exposures (CVEs) package. Amazon Inspector is a service designed for analysing AWS resources in order to identify potential security issues.

By creating templates and specifying targets you can easily assess how secure your AWS resources are. During the assessment process a wide range of elements are scanned and all collected data is then analysed and compared to a set of security rules specified in the template. As a result you get a detailed report indicating what security problems your resources potentially suffer from and what should you do to fix them.

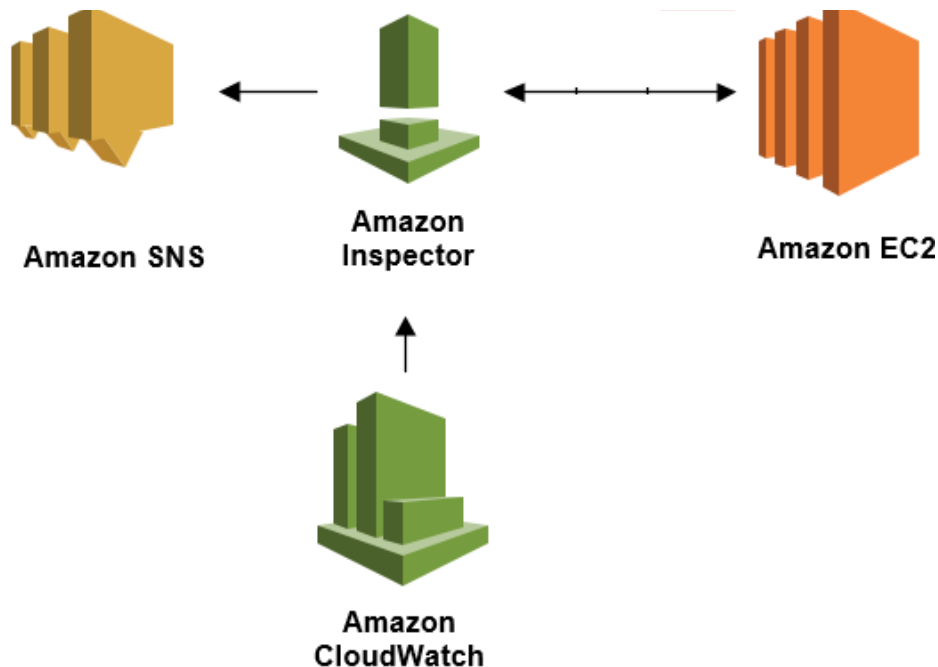


Figure 1. Architectural Overview

2. Installing SSM and Inspector agent on the EC2 instance

Amazon Inspector uses assessment targets to designate the AWS resources to evaluate and to create an assessment target and install a Systems Manager Agent and inspector agent on the EC2 instance using run command which will be restricted otherwise. To verify that the agent is installed and running, sign in to your EC2 instance and run the following command:

```
sudo /opt/aws/awsagent/bin/awsagent status
```

```
ubuntu@ip-10-0-0-7: ~  
ent start First:156 Last:757  
Enable 627:OperatingSystem, Count:1 (sent:1) , TotSize:342, Seconds from assess  
ment start First:0 Last:0  
Enable 628:SplitMsgBegin, Count:1 (sent:1) , TotSize:0, Seconds from assessment  
start First:79 Last:79  
Enable 629:SplitMsgEnd, Count:1 (sent:1) , TotSize:0, Seconds from assessment s  
tart First:79 Last:79  
Enable 630:MonitoringStart, Count:0 (sent:1)  
Enable 631:MonitoringEnd, Count:1 (sent:0) , TotSize:67, Seconds from assessmen  
t start First:877 Last:877  
Enable 632:Terminal, Count:73 (sent:73) , TotSize:8465, Seconds from assessment  
start First:0 Last:0  
Enable 635:ConfigurationInfo, Count:31 (sent:31) , TotSize:53341, Seconds from  
assessment start First:0 Last:0  
Disable 636:TcpV4ListeningPortClosure, Count:0 (sent:0)  
Disable 637:TcpV6ListeningPortClosure, Count:0 (sent:0)  
Disable 638:UdpV4ListeningPortClosure, Count:1 (sent:0) , TotSize:99, Seconds fr  
om assessment start First:835 Last:835  
Disable 639:UdpV6ListeningPortClosure, Count:0 (sent:0)  
Disable 645:SystemPerformance, Count:30 (sent:0) , TotSize:24182, Seconds from a  
ssessment start First:1 Last:873  
Disable 646:ProcessPerformance, Count:30 (sent:0) , TotSize:220044, Seconds from  
assessment start First:1 Last:873  
Enable 647:TimeEvent, Count:28 (sent:29) , TotSize:2604, Seconds from assessmen  
t start First:36 Last:847  
Disable 648:FileInfo, Count:0 (sent:0)  
Enable 649:DirectoryInfo, Count:12 (sent:12) , TotSize:2838, Seconds from asses  
sment start First:0 Last:0  
Enable 650:Oval, Count:1 (sent:11) , TotSize:666996, Seconds from assessment st  
art First:79 Last:79  
Enable 651:Error, Count:0 (sent:0)  
Disable 652:PasswordPolicy, Count:0 (sent:0)  
Enable 653:RetrieverCompletionStatus, Count:3 (sent:3) , TotSize:4776, Seconds  
from assessment start First:156 Last:757  
Enable 654:ProbeResultMsg, Count:0 (sent:0)  
Disable 655:EventSubscriberStatusMsg, Count:3 (sent:0) , TotSize:726, Seconds fr  
om assessment start First:156 Last:757  
Enable 656:OpenPortsMsg, Count:1 (sent:1) , TotSize:1061, Seconds from assessme  
nt start First:6 Last:6  
Disable 657:ProbeInfoMsg, Count:1 (sent:0) , TotSize:241, Seconds from assessmen  
t start First:0 Last:0  
-----  
Dur Since last config load sec: 591724  
All Messages count: 7632  
All Messages size: 4645385  
Messages successfully sent:5316  
Health Message: Count:5045, Seconds from agent start: First:0 Last:591591  
{ "t":1556772754895, "proxy":142, "o":"Ubuntu 16.04.6 LTS\\", "k":"4.4.0-1074-aws  
", "s":5316, "d":0, "l":21, "m":1}  
ubuntu@ip-10-0-0-7:~$
```

Figure 2. Messages exchanged between agent and inspector

This command returns the status of the currently running agent, on checking the status it is observed in the screenshot that messages are being exchanged between the agent installed on ec2 machine and amazon inspector.

AWS Systems Manager Agent (SSM Agent) is Amazon software that can be installed and configured on an Amazon EC2 instance, an on-premises server, or a virtual machine (VM). SSM Agent makes it possible for Systems Manager to update, manage, and configure ec2 instances. SSM Agent is installed, by default in some but in some like the machine tested on it had to be manually installed.

	Start time	Status	Template name	Findings	Findings by severity	Exclusions
<input type="checkbox"/>	Today at 10:47 AM (GMT+5)...	Analysis complete	combined ALL	248	High Medium Low Info	0
<input type="checkbox"/>	04/25/2019 (GMT+5) (7 day...	Analysis complete	combined ALL	250	High Medium Low Info	0
<input type="checkbox"/>	04/25/2019 (GMT+5) (7 day...	Analysis complete	combined ALL	249	High Medium Low Info	0
<input type="checkbox"/>	04/24/2019 (GMT+5) (8 day...	Analysis complete	combined ALL	251	High Medium Low Info	0
<input type="checkbox"/>	04/22/2019 (GMT+5) (10 da...	Analysis complete	combined ALL	248	High Medium Low Info	0
<input type="checkbox"/>	04/19/2019 (GMT+5) (13 da...	Analysis complete	combined ALL	247	High Medium Low Info	0
<input type="checkbox"/>	04/17/2019 (GMT+5) (15 da...	Analysis complete	combined ALL	247	High Medium Low Info	0
<input type="checkbox"/>	04/17/2019 (GMT+5) (15 da...	Analysis complete	temp05	1	High Medium Low Info	0
<input type="checkbox"/>	04/17/2019 (GMT+5) (15 da...	Analysis complete	temp04	3	High Medium Low Info	0
<input type="checkbox"/>	04/17/2019 (GMT+5) (15 da...	Analysis complete	temp03	152	High Medium Low Info	0

Figure 3. Assessment runs on target machine

3. Troubleshooting the unknown agent status with AWS Support Team

I contacted AWS Support because the agent status for egsp linux instance "i-083952770d8693fd1" was being shown as "UNKNOWN". We performed the following troubleshooting actions from within the instance:

- 1) Checked the kernel version which comes out be 4.9.32-15.41.amzn1.x86_64 (lies in kernel versions list that are compatible with an Amazon Inspector agent running on Linux)
- 2) Installed inspector agent again.
- 3) `sudo /opt/aws/awsagent/bin/awsagent status`
- 4) `telnet arsenal.us-west-2.amazonaws.com 443`
- 5) `curl -vk https://arsenal.ap-south-1.amazonaws.com`
- 6) `telnet s3.dualstack.us-west-2.amazonaws.com 443`
- 7) `curl -vk https://s3.dualstack.ap-south-1.amazonaws.com`

Since, we were not able to figure out the possible issue causing this error, I reached out to our internal service team to further troubleshoot the issue. As per the internal team, there wasn't seem to be an issue with our connection to arsenal. Hence, they would require the Agent logs of the instance with this issue which were captured using the following steps:

- 1) Copy the agent.cfg file located at "/opt/aws/awsagent/etc/agent.cfg" to "/opt/aws/awsagent/etc/agent_bckup.cfg".
- 2) Stop the agent with command: `sudo /etc/init.d/awsagent stop`
- 3) Edit the agent.cfg file located at "/opt/aws/awsagent/etc/agent.cfg".
- 4) Use the below configuration OR append these settings to your existing configuration:


```
{
  "SubSystems" : "ALL",
  "LogLevels" : "LogAll",
  "LogFile" : "/tmp/agent_test.log"
}
```
- 5) Restart the agent so that agent reads the correct configuration using the following command: `sudo /etc/init.d/awsagent start`
- 6) Wait 15 minutes or run a 15 minute assessment
- 7) After 15 minutes or the 15 minute assessment is over, stop the agent once again with: `sudo /etc/init.d/awsagent stop`
Collect the file /tmp/agent_test.log and send it back to us.
Copy back "/opt/aws/awsagent/etc/agent_bckup.cfg" to "/opt/aws/awsagent/etc/agent.cfg"
- 8) Restart the agent so that agent reads the original configuration again:
`sudo /etc/init.d/awsagent start`

The agent logs obtained by following the steps mentioned above helped us narrow down the issue and then we captured a packet trace at agent start using the following steps:

- 1) Check if tcpdump is installed on the system: `which tcpdump`
- 2) If tcpdump is not installed then: `sudo yum install -y tcpdump`
- 3) Stop the agent with command: `sudo /etc/init.d/awsagent stop`
- 4) Start network capture to a file: `sudo tcpdump -i any -w agent.pcap`
- 5) Start the agent: `sudo /etc/init.d/awsagent start`
- 6) Wait 30 seconds
- 7) Stop network capture: `Ctrl+C`
- 8) Attach agent.pcap to this support case.

The case is still open and the issue remains unresolved as of now but I came across a blog dated Jan 2019 which states that CIS Operating System Configuration findings is currently not supported for Amazon Linux versions in Inspector yet.

4. Monitoring Amazon Inspector Using Amazon CloudWatch

The Amazon Inspector namespace includes the following metrics. And can be monitored for real-time metrics using Amazon CloudWatch, which collects and processes raw data into readable. By default, Amazon Inspector sends metric data to CloudWatch in 5-minute periods. And can be used with the AWS Management Console, the AWS CLI, or an API to view the metrics that Amazon Inspector sends to CloudWatch. Here, console is used.

1) AssessmentTargetARN metrics:

Metric	Description
TotalMatchingAgents	Number of agents that match this target
TotalHealthyAgents	Number of agents that match this target that are healthy
TotalAssessmentRuns	Number of assessment runs for this target
TotalAssessmentRunFindings	Number of findings for this target

2) AssessmentTemplateARN metrics:

Metric	Description
TotalMatchingAgents	Number of agents that match this template
TotalHealthyAgents	Number of agents that match this template that are healthy
TotalAssessmentRuns	Number of assessment runs for this template
TotalAssessmentRunFindings	Number of findings for this template

3) Aggregate metrics

Metric	Description
TotalAssessmentRuns	Number of assessment runs in this AWS account

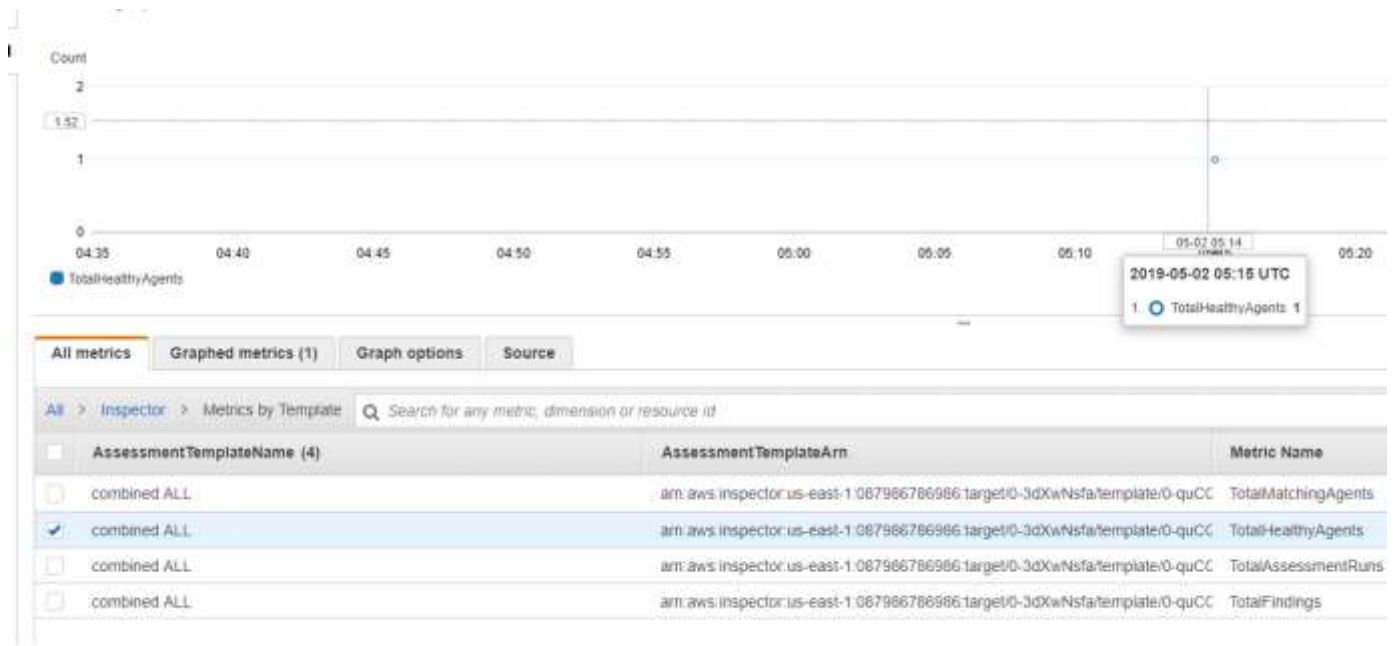


Figure 4. Graph metrics for total healthy agents

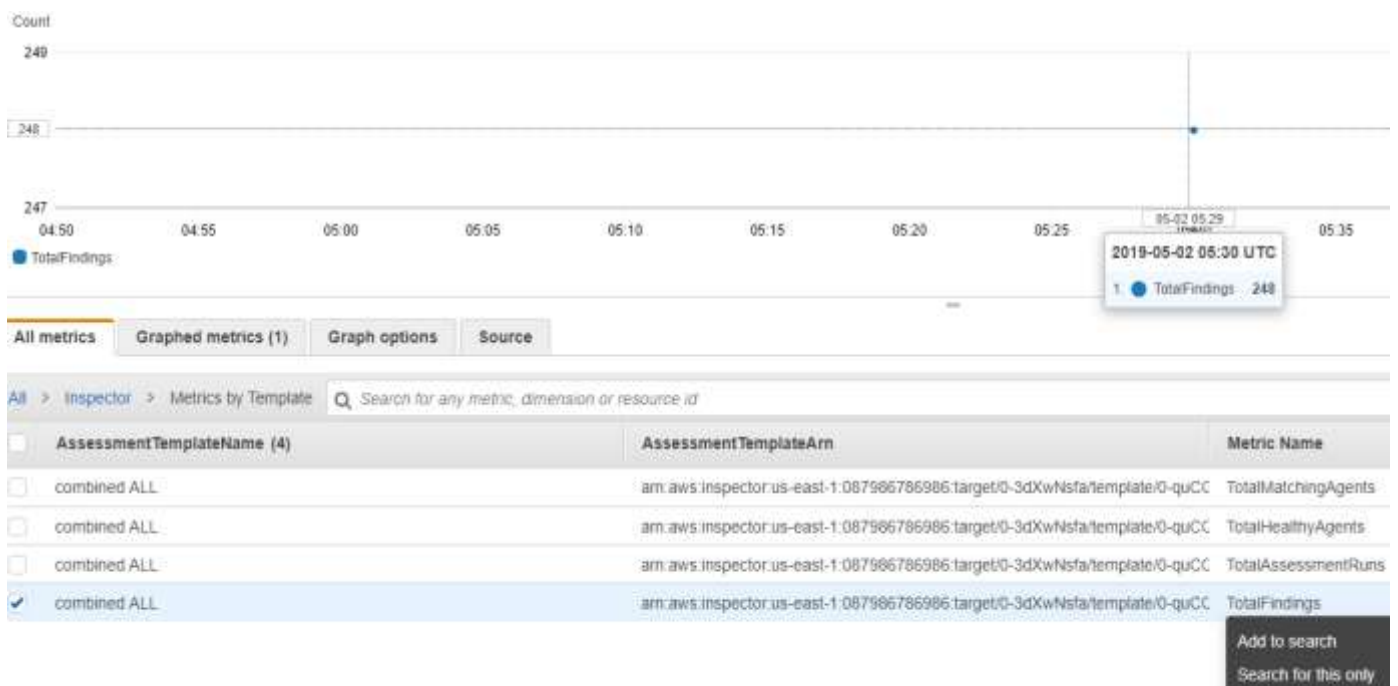


Figure 5. Graph metrics for total findings

5. Monitoring Amazon Inspector Using Amazon CloudTrail

Amazon Inspector is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Amazon Inspector. CloudTrail captures all API calls for Amazon Inspector as events, including calls from the Amazon Inspector console and code calls to the Amazon Inspector API operations.

The major difference noticed between CloudWatch and CloudTrail monitoring is that Cloudwatch logs focus on what is happening, which resources and services are being used. Whereas CloudTrail focusses on revealing who did the activity and when was it done.

Event history

Your event history contains the activities taken by people, groups, or AWS services in [supported services](#) in your AWS account. By def

You can view the last 90 days of events. Choose an event to view more information about it. To view a complete log of your CloudTrail

Filter:		Time range:		
User name	prachi	Select time range		
	Event time	User name	Event name	Resource type
▶	2019-05-02, 12:32:13 PM	prachi	DescribeConfigurationRecorders	
▶	2019-05-02, 12:32:09 PM	prachi	LookupEvents	
▶	2019-05-02, 12:31:26 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:26 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:26 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:25 PM	prachi	DescribeAssessmentTemplates	
▶	2019-05-02, 12:31:25 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	DescribeAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:24 PM	prachi	ListAssessmentRuns	
▶	2019-05-02, 12:31:23 PM	prachi	DescribeResourceGroups	

Figure 6. CloudTrail event history for Amazon Inspector

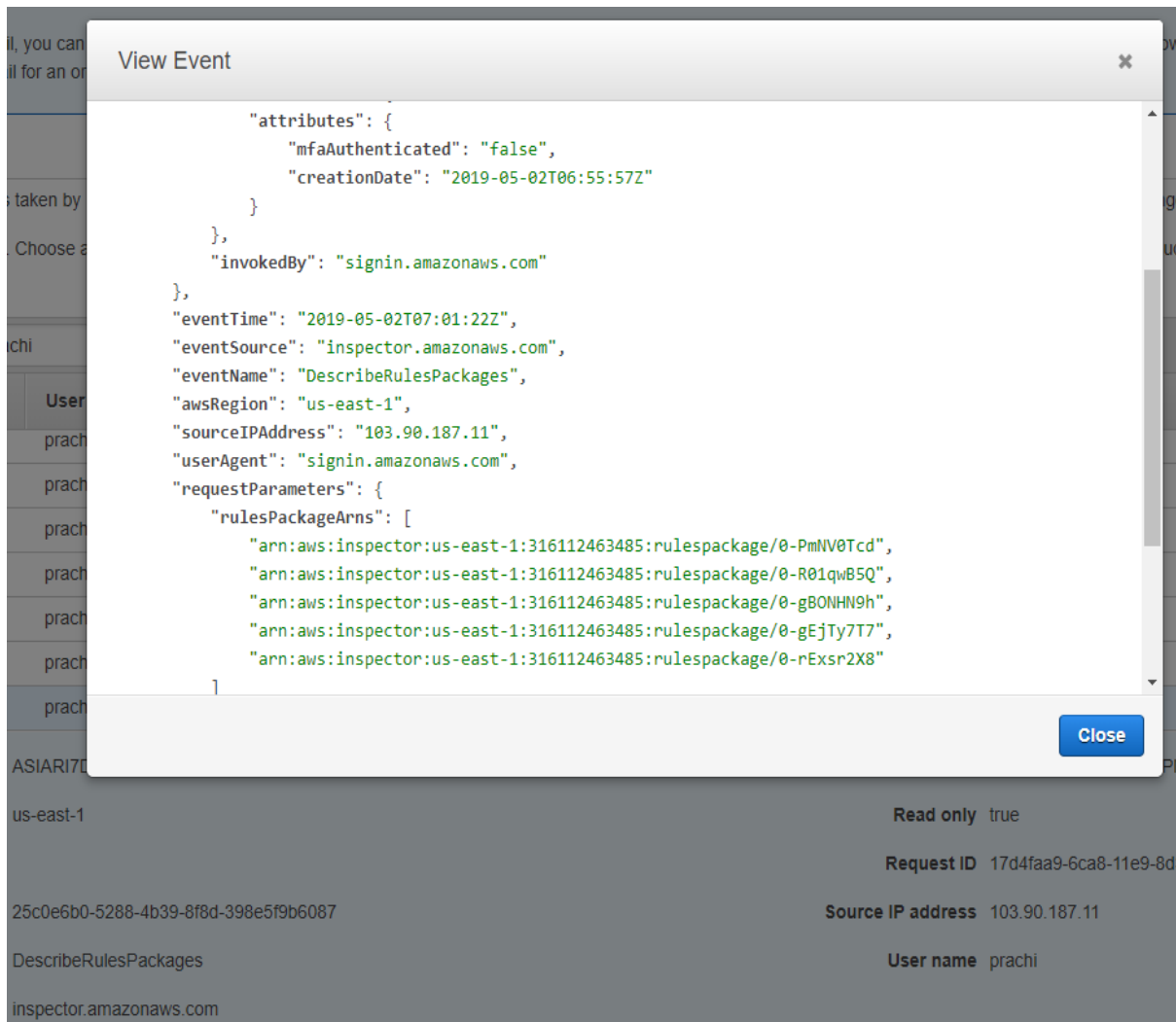


Figure 7. CloudTrail event triggered details

6. Report findings and Remediation

The reports generated were studied and the remediations were segregated and proceeded as per below priorities:

Priority 1: Critical Risk Profile and can be addressed Quick in Time

Priority 2: High Risk Profile and can be addressed Quick in Time

In parallel of above, exercise is performed to logically group all the findings irrespective of their Risk Profile rating and Time to fix. (Here preferences should be given which are exposed over internet and the vulnerability is old in age – can be extracted from CVE info)

Amazon Inspector - Findings

Findings are potential security issues discovered after Amazon Inspector runs an assessment against a specified assessment target. [Learn more.](#)

✖ Filters: [{"assessmentRunArns":["arn:aws:inspector:us-east-1:887986786986:target/0-3dXmNsfa/template/0-quCQLROc/run/0-o25aqDg1"]}]

Add/Edit attributes Last up

Filter

<input type="checkbox"/>	Severity	Date	Finding	Target	Template	Rules Package
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit
<input type="checkbox"/>	High	Today at 11:...	Instance i-0be80745fdee529db is not compliant wit...	test01	combined ALL	CIS Operating System Securit

Figure 8. Operating system vulnerabilities

For e.g.

1. There are several vulnerabilities in CVE section in relation to PHP which can be fixed by updating it to the latest version, to eliminate multiple linked vulnerabilities.
2. Multiple vulnerabilities due to older Linux version, on updating the version and it will fix the multiple linked vulnerabilities.
3. There are multiples ports and services being set as open by Amazon, and are being highlighted by them as vulnerable so I checked with the application owners that what ports and services are required open to enable the accessibility of application and rest of the ports and services can be shut to reduce down the list of vulnerabilities

These are just examples, and such relations can be found out throughout the list of vulnerabilities being generated. After addressing the P1 and P2, the ec2 instances was rescanned and reconciled output had lesser number of vulnerabilities left.

Further, it is important to know what all applications and services are running on the EC2 instance being monitored because accordingly the priority of remediations will change. The contact us eventifyd instance had a tomcat server and sql server majorly running to host a webpage. . Port number 8080 and 81 were open so they were closed. 8080 was also not required because the application had tomcat hosted on port 80. And the webpage <http://contactus.minddemo.cloud> is not hosted on https.

Section 1: Executive Summary

This is an Inspector assessment report for an assessment started on 2019-04-24 11:38:16 UTC for assessment template 'combined ALL'. The assessment target included 1 instances, and was tested against 5 Rules Packages.

The assessment target is defined using the following EC2 tags

Key	Value
Name	contactus-mind-cloud-services

The following Rules Packages were assessed. A total of 250 findings were created, with the following distribution by severity:

Rules Package	High	Medium	Low	Informational
CIS Operating System Security Configuration Benchmarks-1.0	78	0	0	11
Common Vulnerabilities and Exposures-1.1	80	74	2	0
Network Reachability-1.1	0	0	0	2
Runtime Behavior Analysis-1.0	0	0	0	2
Security Best Practices-1.0	0	1	0	0

Figure 9. Findings report summary

7. Lambda for email notification automatic updation

1) Create an SNS topic to which Amazon Inspector will publish messages:

Amazon SNS uses *topics*, communication channels for sending messages and subscribing to notifications. An SNS topic was created for this solution to which Amazon Inspector publishes messages whenever there is a security finding. Then a Lambda function is created that subscribes to this topic and receives a notification whenever a new security finding is generated.

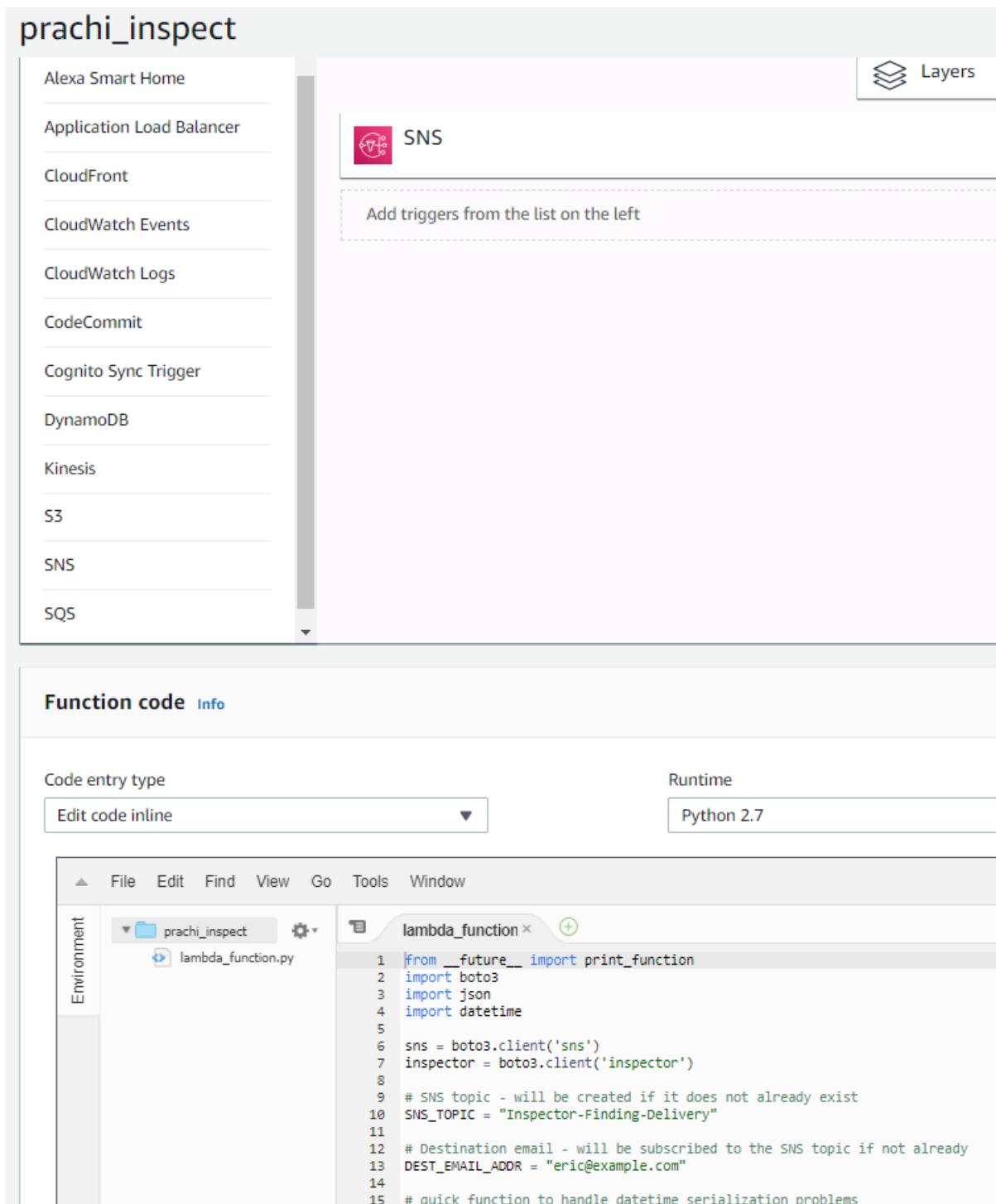


Figure 10. Lambda for automated updation

2) Configure an Amazon Inspector assessment template to post finding notifications to the SNS topic:

An *assessment template* is a configuration that tells Amazon Inspector how to construct a specific security evaluation. For example, an assessment template can tell Amazon Inspector which EC2 instances to target and which rules packages to evaluate. You can configure a

template to tell Amazon Inspector to generate SNS notifications when findings are identified. In order to enable automatic remediation, you either create a new template or modify an existing template to set up SNS notifications to the SNS topic that you just created.

3) Create the Lambda auto-remediation function:

Now, create a Lambda function that listens for Amazon Inspector to notify it of new security findings, and then tells the EC2 SSM agent to run the appropriate system update command.

8. Pricing Incurred in task

On calculating the bill using simple AWS monthly calculator the following is the cost estimated for the task.

1) Amazon Inspector Assessment runs-

Amazon Inspector assessments with the network reachability rules packages are priced per instance per assessment (instance-assessment) per month. For example, if you run 1 assessment against 1 instance, that is 1 instance-assessment. If you run 1 assessment against 10 instances, that is 10 instance-assessments. The pricing starts at \$0.15 per instance-assessment per month but the first 250 assessment runs are free and around 15 runs were made so the costing stands at zero.

2) CloudWatch-

Detailed ec2 monitoring is not enabled so the price \$2.10 per instance per month and goes down to \$0.14 per instance at the lowest priced tier.

3) CloudTrail-








There was not any need to create additional trail so the costing stands at zero.

4) Lambda, SNS service and AWS Business Support is also utilized.

5) EC2 Instances-

The below is the monthly cost and they were used for 5 days so it adds up to 1.5 dollars

Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
	contact us instance	1	10 Hours/Day ▼	Linux on t2.micro 	On-Demand (No Cor 	\$ 3.79
	bastion host	1	10 Hours/Day ▼	Windows on t2.micro 	On-Demand (No Cor 	\$ 5.19
	Add New Row					