**Automated AWS Security Group Compliance Check and Remediation**

|  |  |
| --- | --- |
| Document Owner | |
| Author | Position / Responsibility |
| Shambo Basu | App/Cloud Support Specialist |
|  |  |

|  |  |  |
| --- | --- | --- |
| Document Reviewer | | |
| Name | Title / Role | Signature |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Document Version Revisions | | |
| Date | Version | | Editor | Signature |
| 12/2/2021 | 1.0 | | Shambo Basu |  |
|  |  | |  |  |
|  |  | |  |  |

|  |  |  |
| --- | --- | --- |
| Document Approver | | |
| Name | Title / Role | Signature |
|  |  |  |
|  |  |  |
|  |  |  |

**Pre-requisites**

To create custom AWS Config rules below are the pre-requisites with links for installation:

1. AWS Programmatic Access – *Once this setup is completed, please remove the programmatic access as that will be flagged in ISD.*
2. AWS CLI 2 - https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2-windows.html
3. Python 3.8 - https://www.c-sharpcorner.com/article/how-to-install-python-3-8-in-windows/
4. Boto 3 - https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html
5. RDK – See below.

**RDK Installation**

If you want to use the RDK, go ahead and install it using pip:

$ pip install rdk

**To deploy the AWS Config rule**

1. Download ZIP attached here and extract.
2. Run the following RDK command.

rdk deploy SECURITY\_GROUP\_REMEDIATION

1. After the code is deployed, sign in to the AWS Management Console and navigate to AWS Config.
2. Navigate to [rules](https://console.aws.amazon.com/config/home#/rules), and you should see the AWS Config rule SECURITY\_GROUP\_REMEDIATION. After the rule is deployed, it will automatically trigger an evaluation, and you should see evaluation results shortly after deployment.
3. If you have any noncompliant resources, you should see them appear. If not, you can create a noncompliant security group by adding in an internet-accessible port.
   1. [Create a new security group](https://console.aws.amazon.com/ec2/v2/home#CreateSecurityGroup) and add an inbound rule with the Source as Anywhere or 0.0.0.0/0 with an open port of your choosing.

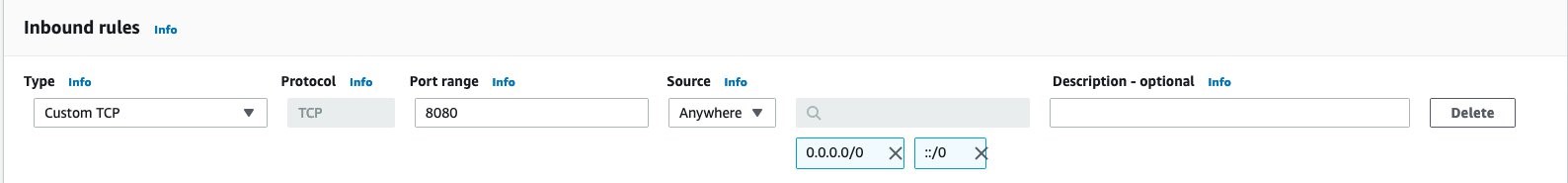


Figure 1: Configure a noncompliant security group

* 1. After the security group rule is created or modified. Within a few minutes, the change you made should trigger the AWS Config rule you deployed and you’ll see the EC2 Secuirty Group as noncompliant.

**Create a remediation action**

As previously mentioned, AWS Config remediation actions are declared in Systems Manager automation documents, which are invoked for identified noncompliant resources. This means that in order to create a custom remediation action, you’ll need to create a custom Systems Manger automation document. As in the previous steps, you’ll use Python as the language of choice for the remediation action in this blog post.

In the folder you’ll find the SECURITY\_GROUP\_REMEDIATION.yaml file, which contains the template used for the automation document.

**To create the remediation action and accompanying infrastructure**

1. In the directory with the YAML remediation document, run the following command to create the automation document.
2. aws ssm create-document --content file://SECURITY\_GROUP\_REMEDIATION.yaml --name "security-group-remediation-quarantine" --document-type "Automation" --document-format YAML
3. After the document is created, you can view it in your Systems Manager Documents in the console, or by running the following CLI command.
4. aws ssm list-documents --filters Key=Owner,Values=Self
5. Before you move on to setting up the AWS Config rule to use this document, you need to give Systems Manager the proper IAM permissions to be able to run the commands specified in the document. You should see the file security-group-remediation-quarantine-policy.json in the cloned GitHub repository. That policy document contains the necessary permissions. To create a new policy with those permissions, run the following command.
6. aws iam create-policy --policy-name security-group-remediation-quarantine-policy --policy-document file://security-group-remediation-quarantine-policy.json
7. Take note of the Arn key in the response, because you’ll need this value in step 8 below.
8. You also need to create an IAM role and assign the newly created policy to it. You can do that by running the following command (note that the required trust policy document is also provided, named security-group-remediation-quarantine-trust-policy.json).

aws iam create-role --role-name security-group-remediation-quarantine-role --assume-role-policy-document file://security-group-remediation-quarantine-trust-policy.json

1. Take note of the Arn key in the response, because you’ll need it in step 5 of the next section.
2. Lastly, assign the created policy to your newly created role (replace *{POLICY\_ARN}* with the policy Arn value you noted in step 6):

aws iam attach-role-policy --role-name 'security-group-remediation-quarantine-role' --policy-arn *{POLICY\_ARN}*

You’ve set up all the infrastructure you need! Moving on to the next section, you’ll configure your AWS Config rule to auto-remediate by using your newly created Systems Manager automation document.

**Configure the AWS Config rule to automatically remediate**

You need to associate the automation document with your AWS Config rule and configure auto-remediation. This will cause any noncompliant resources to be automatically remediated after they’re identified as noncompliant.

**To configure the rule for auto-remediation**

1. In the AWS Config console, navigate to your AWS Config rules and select the recently created rule.
2. At the top right, choose Edit, and scroll down to Choose remediation action.
3. In the Remediation action field, select the recently created Systems Manager automation document. Be sure to turn on Auto remediation.
4. Set the Resource ID parameter to ResourceId to indicate to the Systems Manager document which parameter is the noncompliant resource ID.
5. Lastly, set the ConfigRuleName parameter to the name of the AWS Config rule (SECURITY\_GROUP\_REMEDIATION) and set the AutomationAssumeRole parameter to the Arn you took note of the role creation response, from step 8 in the above section.

Your configuration should look like the one in Figure 2.

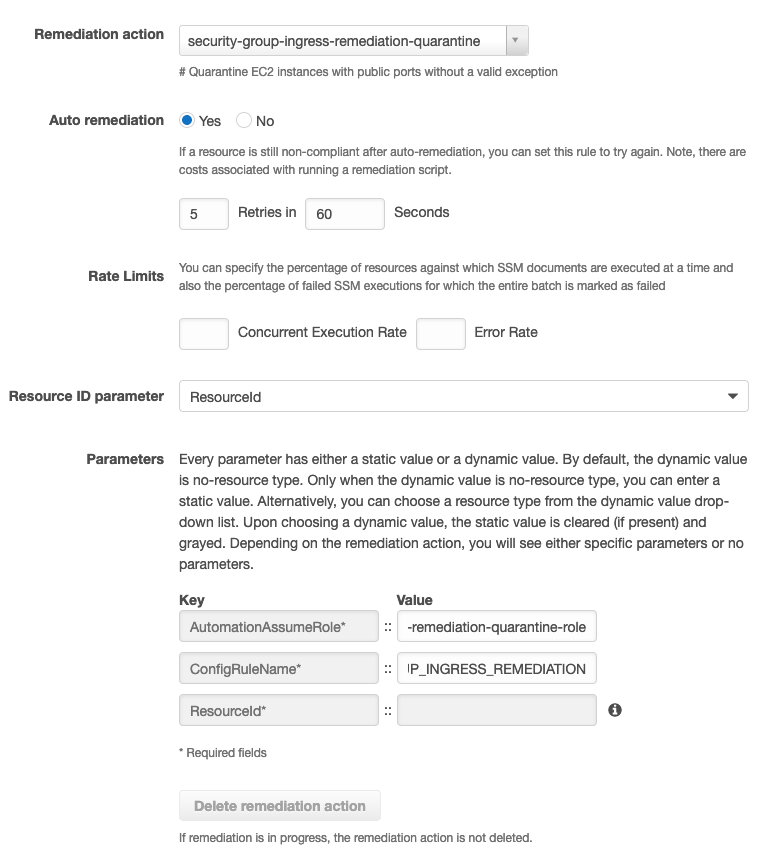


Figure 2: AWS Config rule remediation configuration

1. Save your configuration, that’s all.