# Network Hardening Using Amazon Inspector and AWS Systems Manager

## Lab overview

Securing an infrastructure can be a challenge for any company. Companies use many tools to audit networks and find vulnerabilities in systems and applications. This process takes significant time and effort.

In this lab, you are a new security engineer for AnyCompany. You need to identify weak areas in the company's network security and update AnyCompany's environment for better efficiency and optimization. You will use Amazon Inspector to do this.

**Amazon Inspector** runs scans that analyze all your network configurations—such as security groups, network access control lists (network ACLs), route tables, and internet gateways—together to infer reachability. You don't need to send packets across the virtual private cloud (VPC) network or connect to Amazon Elastic Compute Cloud (Amazon EC2) instance network ports. It's like packetless network mapping and reconnaissance.

From Amazon Inspector, you will use the **network reachability package** to analyze your network configurations to find security vulnerabilities in your EC2 instances. The findings that Amazon Inspector generates also provide guidance about restricting access that is not secure.

## Objectives

- Configure Amazon Inspector
- Run an agentless network audit
- Investigate the scan results
- Update security groups
- Log in to an application server instance using AWS Systems Manager Session Manager

# Lab environment

The current environment has two EC2 instances. One instance is a bastion server named **BastionServer** in a public subnet. The other instance is an application server named **AppServer** in a private subnet.

Bastion servers are servers used to manage access to an internal or private network from an external network. They are sometimes called jump boxes or jump servers. Because bastion servers are often accessible from the internet, they typically run a minimum number of services to reduce their attack surface. They are also commonly used to proxy and log communications, such as Secure Shell (SSH) sessions.

# Task 1: View EC2 instances and add tags

To create an assessment target for Amazon Inspector Classic to assess, you start by tagging the EC2 instances that you want to include in your target. In this task, you tag the BastionServer instance.

Every AWS tag consists of a key and value pair of your choice. For example, you might choose to name your key **Name** and your value **MyFirstInstance**.

In the AWS Management Console, choose Services and select **EC2**.

If you see **New EC2 Experience** at the upper-left of your screen, confirm that **New EC2 Experience** is selected. This lab is designed to use the new Amazon EC2 console.

In the left navigation pane, choose **Instances**.
The running **BastionServer** and **AppServer** EC2 instances are listed.
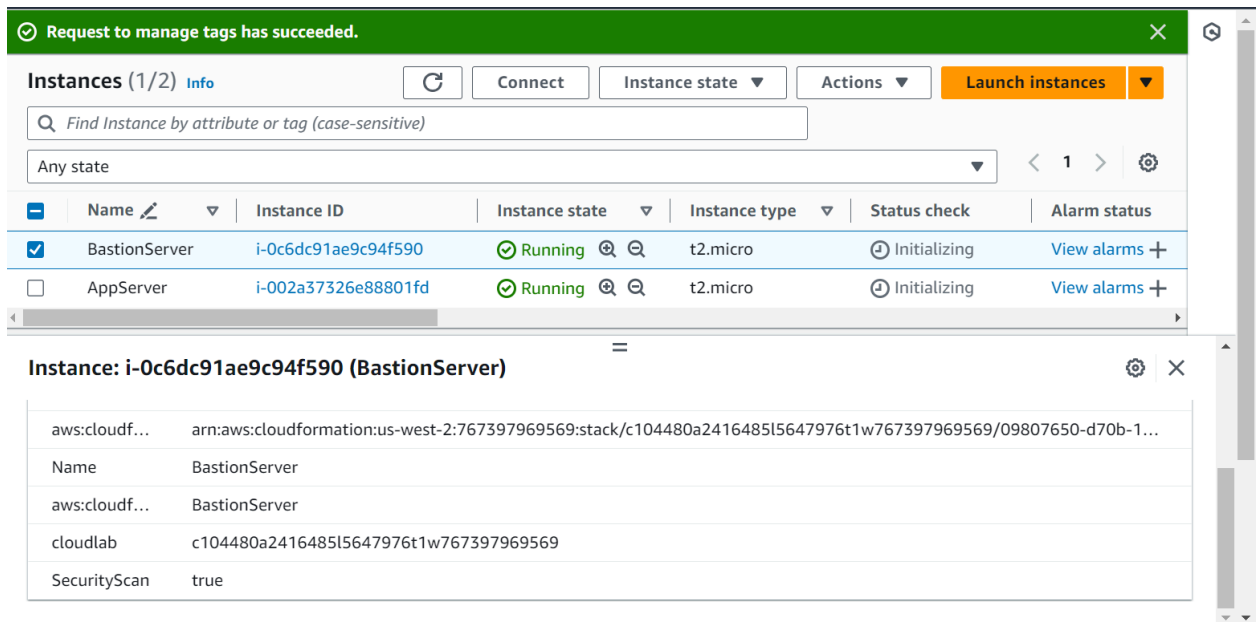
Choose the **BastionServer** instance.

Choose the **Tags** tab.

Choose **Manage tags**.

Choose **Add tag**, and then enter the following information:

○ **Key**: `SecurityScan`
○ **Value**: `true`

Choose **Save**.



You have successfully applied tags for the BastionServer instance, which allows the security scan to find and scan this instance.

# Task 2: Configure and run Amazon Inspector

In this task, you learn how to run an agentless network audit on your EC2 instances using Amazon Inspector. For this lab, you use the network reachability rules package.

**Use case:** It might not be possible to install agents on all hosts in your deployment. Not all types of operating systems support Amazon Inspector agents. Using this method, you will be able to run network audits on all hosts.

In the **AWS Management Console**, choose the **Services** menu. Then choose **Security, Identity, & Compliance** and choose **Inspector**.

To open the navigation pane, choose on the left.

Choose **Switch to Inspector Classic**

Choose **Get started**.

Choose **Advanced setup**.

In the **Define an assessment target** section, configure the following options:

- For **Name**, enter `Network-Audit`
- Clear the check box for **All Instances**.
- For **Tags: Key**, choose **SecurityScan**.
- For **Tags: Value**, choose **true**.
- Clear the check box for **Install Agents**.

Choose **Next**.

In the **Define an assessment template** section, configure the following options:

- For **Name**, enter `Assessment-Template-Network`
- For **Rules packages**, leave **Network Reachability-1.1** selected, but choose the **x** next to each of the other packages to remove them.
- For **Duration**, choose **15 Minutes**.
- Clear the check box for **Assessment Schedule**.

Choose **Next**.

Choose **Create**.
You should see a **SUCCESS** notification, which confirms that the assessment run was initiated. It takes about 3-5 minutes to complete.

Check the status of the scan:

- In the left navigation pane, choose **Assessment runs**.
- In the **Amazon Inspector - Assessment Runs** section, choose the in the row for the run that you initiated to expand it and access more options for your run.
- To see the status of the run, choose **Show status**. If you do not see **Show status**, choose at the top.
- To close and return to the previous screen, choose **Close**.

Once the status changes to **Analysis complete**, choose **Findings** in the left navigation pane.

## Amazon Inspector - Findings

Findings are potential security issues discovered after Amazon Inspector runs an assessment against a specified assessment target. Learn more.

Add/Edit attributes                    Last updated on February 29, 2024 7:49:44 PM (2m ago)

Filter                                                                    Viewing 1-11 of 11

| | Severity ⓘ ▾ | Date ▾ | Finding | Target | Template |
|---|---|---|---|---|---|
| ▸ | High | Today at 7:4… | On instance i-0c6dc91ae9c94f590, TCP port 23 wh… | Network-Audit | Assessment-Temp… |
| ▸ | High | 01/23/2024 … | On instance i-060fa5449f657ea7d, TCP port 23 wh… | Network-Audit | Assessment-Temp… |
| ▸ | High | 01/23/2024 … | On instance i-060fa5449f657ea7d, TCP port 23 wh… | Network-Audit | Assessment-Temp… |
| ▸ | Medium | Today at 7:4… | On instance i-0c6dc91ae9c94f590, TCP port 22 wh… | Network-Audit | Assessment-Temp… |
| ▸ | Medium | 01/23/2024 … | On instance i-060fa5449f657ea7d, TCP port 22 wh… | Network-Audit | Assessment-Temp… |
| ▸ | Medium | 01/23/2024 … | On instance i-060fa5449f657ea7d, TCP port 22 wh… | Network-Audit | Assessment-Temp… |
| ▸ | Medium | 01/23/2024 … | On instance i-060fa5449f657ea7d, TCP port 22 wh… | Network-Audit | Assessment-Temp… |
| ▸ | Informational | Today at 7:4… | Aggregate network exposure: On instance i-0c6dc… | Network-Audit | Assessment-Temp… |
| ▸ | Informational | 01/23/2024 … | Aggregate network exposure: On instance i-060fa5… | Network-Audit | Assessment-Temp… |
| ▸ | Informational | 01/23/2024 … | Aggregate network exposure: On instance i-060fa5… | Network-Audit | Assessment-Temp… |

In this task, you created an assessment target (a collection of the AWS resources that you want Amazon Inspector Classic to analyze). Then you created an assessment template (a blueprint that you use to configure your assessment). You used the template to start an assessment run, which is the monitoring and analysis process that results in a set of findings.

# Task 3: Analyze Amazon Inspector findings

The findings that these rules generate show whether your ports are reachable from the internet through an internet gateway (including instances behind Application Load Balancers or Classic Load Balancers), a VPC peering connection, or a virtual private network (VPN) through a virtual gateway. These findings also highlight network configurations that allow for potentially malicious access, such as mismanaged security groups, ACLs, and internet gateways.

Choose to expand the high-severity finding. You should see the following key details:

- ○ **AWS agent ID** shows you the affected EC2 instance.
- ○ **Description** shows the reason for the finding. In this case, TCP port 23, which is associated with Telnet, is reachable from the internet.
- ○ **Recommendation** provides remediation suggestions.

Telnet is a text-based terminal emulation utility that is part of the TCP/IP suite of protocols. It allows a system to connect to a remote host to perform commands as if you were on the console of the remote machine.

Choose to expand the medium-severity findings and analyze the details.

- ○ For the medium-severity finding, TCP port 22, which is associated with SSH, is reachable from the internet.

SSH, like the Telnet utility, gives a user the ability to log in to a remote machine and perform commands as if they were on the console of that system. Telnet, however, is insecure because its data isn't encrypted when communicated. SSH provides a secure, encrypted tunnel to access another system remotely.

# Task 4: Update security groups

In this task, you see a few remediation options for the security findings that Amazon Inspector discovered. The first option shows how to lock down port 22 to specific IP addresses.

Choose to expand the details of the high-severity finding.

In the **Recommendation** section, choose the link to the security group. The link should look similar to the following example: **sg-0b2dc685cd6e6e706**.

When the link opens, you can see the **BastionServerSG** security group that is attached to the **BastionServer** that has produced findings within Amazon Inspector.

Choose the **Inbound rules** tab.

These are the current inbound rules for this security group. They are also the high and medium findings that Amazon Inspector caught.

Choose **Edit inbound rules**.

For the inbound rule associated with port range **23**, choose **Delete**.

> Port 23 Telnet is vulnerable to security attacks, and the SSH protocol helps you to overcome many security issues of Telnet. SSH is now the only major protocol to access the network devices and servers over the internet.

For the **SSH** rule, remove the current inbound IP address of **0.0.0.0/0** by choosing the **X** next to it to update the resource.

The 0.0.0.0/0 IP address for inbound rules means that port 22 is accessible from anyone on the internet.

> You can adjust the inbound rules so that only your IP address is able to access port 22. Although this option is much more secure, it still has vulnerabilities. For example, someone could access the computer that is associated with that IP address and gain access.

> For **Source**, choose the **Custom** dropdown list, and then select **My IP**.

> Choose **Save rules**.

# Re-scan the environment

> Navigate to the browser tab that has Amazon Inspector open. In the left navigation pane, choose **Assessment templates**.

Select the check box next to **Assessment-Template-Network**, and choose **Run**. This step runs the same scan from earlier in the lab and produces findings from the security group updates.

> **Note** The scan takes approximately 30-60 seconds to complete.

> In the left navigation pane, choose **Assessment runs**, and refresh every 10-15 seconds until the **Status** changes to **Analysis complete**.

In the left navigation pane, choose **Findings**, and then choose **Date** to sort by most-recent findings.

> The high-severity finding is now gone, but the medium-severity finding remains. Although port 22 was scoped down to allow access to only your IP address, port 22 is still technically open to the internet outside the VPC.

In this task, you updated the security group attached to the BastionServer so that it allows traffic from only your IP address instead of the open internet and removed the wide-open and no-longer-needed Telnet port.

# Task 5: Replace BastionServer with Systems Manager

In this task, you replace the BastionServer instance, which has primarily used SSH to connect to the AppServer within the private subnet. Instead, you use Session Manager via Systems Manager.

**Systems Manager** is a secure end-to-end management solution for hybrid cloud environments. Systems Manager is the operations hub for your AWS applications and resources and consists of four core feature groups.

In the AWS Management Console, choose Services and select **EC2**.

In the left navigation pane, choose **Security Groups**.

Choose the **Security group ID** for **BastionServerSG**.

Choose **Edit inbound rules**.

Choose **Delete**, and then choose **Save rules** to remove the SSH inbound rule.

In the left navigation pane, choose **Instances**.

Select the check box for **BastionServer**. Then choose the **Instance state** dropdown list, and choose **Stop instance**.

In the confirmation dialog, choose **Stop**.
Next, connect to the AppServer directly using Session Manager.
With **Session Manager**, you can quickly and securely access your EC2 instances through an interactive one-click browser-based shell or through the AWS Command Line Interface (AWS CLI) without the need to open inbound ports, maintain bastion hosts, or manage SSH keys.

Select the check box next to **AppServer**, and then choose **Connect**.

You are now connected directly to the **AppServer**.

Enter in the following Linux commands to change the directory and to view the current working directory of the AppServer.

```
cd ~
pwd
```

The output should look like the following: **/home/ssm-user**

# Final scan of the environment

Go to your browser tab that has **Amazon Inspector** open.

In the left navigation pane, choose **Assessment runs**.

Select the check box for the previously run assessment, and then choose **Run**.

Wait for the **Status** to show **Analysis complete**, and choose to expand the details of the most recent assessment run.

Verify that there are zero **Findings**.

You have successfully improved the network security by adding an IAM role to the AppServer and removing the SSH inbound rule within the Bastion security group while making it even easier to connect using Session Manager provided by Systems Manager.

# Conclusion

You now have successfully:

- Configured Amazon Inspector
- Run an agentless network audit

- Investigated the scan results
- Updated security groups
- Logged in to an application server using Session Manager