Security Group Guardrails for AWS

Turbot HQ, Inc
1732 1st Ave #20232
New York, NY 10128 USA

info@turbot.com
https://turbot.com
+1 973 922 0297
Overview

Security Groups are a critical control point in applications on AWS. Turbot provides both full management of security groups and automated guardrails to protect customer managed groups. This document outlines the key design decisions and policies used in AWS Security Group management.

Prerequisite: Network Guardrails for AWS

Security Group Guardrails builds on the definitions and capabilities outlined in Network Guardrails for AWS. Please read that document first and use it as a reference throughout this guide.

Introduction to AWS Security Groups

Security groups act as virtual firewalls for instances launched within an AWS VPC. Instances can be associated with up to 5 security groups and because security groups act at the instance level and not the subnet level, different sets of security groups can be applied to instances inside the same subnet – allowing creation of resource zones.

Security group characteristics:

- Security groups specify allow rules, but not deny rules.
- Security groups specify separate rules for inbound and outbound traffic.
- By default, a security group has no inbound rules. Therefore, no inbound traffic is allowed until inbound rules are added to the security group.
- If a security group has no outbound rules, no outbound traffic is allowed. (Warning: AWS adds a default egress all rule to all new security groups.)
- Security groups are stateful — when a request is sent from an instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to approved inbound traffic are allowed to flow out, regardless of outbound rules.
- Instances associated with a security group can’t talk to each other unless rules are added allowing it.
- Security groups are associated with network interfaces. Changing the security groups associated with the instance changes the security groups associated with the primary network interface (eth0). Instances can also be associated with additional network interfaces with separate sets of security groups.
Security Group Automation & Control in Turbot

Control requirements over security groups vary widely between customers, applications and even application environments. In some cases, they are strictly controlled with no modification available to application teams. Other times, security groups are managed by the application team directly within guardrails defined centrally.

To support these cases, Turbot offers multiple degrees of automation:

<table>
<thead>
<tr>
<th>Type</th>
<th>Defined by</th>
<th>Managed by</th>
<th>Guardrails by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbot Security Groups</td>
<td>Turbot</td>
<td>Turbot</td>
<td>Turbot</td>
</tr>
<tr>
<td>Turbot Managed Security Groups</td>
<td>Customer</td>
<td>Turbot</td>
<td>Turbot</td>
</tr>
<tr>
<td>Security Groups</td>
<td>Customer</td>
<td>Customer</td>
<td>Turbot</td>
</tr>
</tbody>
</table>

Turbot Security Group Architecture

Overview

Common, simple security groups are intended to provide a secure, easily understood starting point for boundary protection and dynamic isolation of application components. Common security groups do not aim to solve every scenario, applications are expected to create specific, custom security groups according to their specific application requirements & security needs.

Turbot’s security group model should make it simple to implement core security design goals:
- Full application isolation, even inside the intranet.
- Within-application security across outward and inward facing subnets.
- Bastion level access by administrators.
- Trusted sub-networks.

Security Groups – Purpose

Turbot provides standard security groups to meet common purposes – e.g. web, app and database patterns. This flexible, best practice core suits many application teams and provides a great balance between security and ease of use.

Note that the Purpose is consistent with Subnet Purpose as covered in Turbot Network Guardrails for AWS. This relationship allows Turbot to make appropriate automation and guardrail decisions both in the creation and use of security groups – e.g. web_* security groups can only be used with resources in subnets with a web purpose.
Turbot encourages the use of standard ports by default – this reduces application complexity and the learning curve for developers. Here is our recommendation which covers most use cases:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Protocol</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>web</td>
<td>Web services</td>
<td>TCP</td>
<td>80</td>
<td>http</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>443</td>
<td>https</td>
</tr>
<tr>
<td>app</td>
<td>Tomcat/Weblogic and other app servers</td>
<td>TCP</td>
<td>8080-8083</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>8443-8446</td>
<td></td>
</tr>
<tr>
<td>db</td>
<td>Typical enterprise databases</td>
<td>TCP</td>
<td>1433</td>
<td>SQL Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>1521</td>
<td>Oracle</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>3306</td>
<td>MySQL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>5432</td>
<td>Postgres</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP</td>
<td>5439</td>
<td>Redshift</td>
</tr>
</tbody>
</table>

Custom security groups can be highly specific, but standard security groups should be simple and flexible enough to understand. Otherwise users reduce security through arbitrary application of rules.

**Security Groups – Source**

The Purpose maps out functionality, but to be effective security groups need to carefully manage the source of access. For standard security groups, Turbot uses the Network Security Zones as defined in Turbot Network Guardrails for AWS:

- Public – CIDR ranges for the Internet
- Intranet – CIDR ranges for the organizations Intranet
- Trusted – Dynamically calculated list of CIDR ranges for Subnets in this VPC and it’s peered VPCs that are not publicly accessible (see Private below)
- Private – Dynamically calculated list of CIDR ranges for Subnets in the VPC that are not publicly accessible (i.e. all except DMZ and Public Subnet Types).
- Bastion – CIDR ranges for trusted bastion servers
- Pair – Security group reference, allowing paired resources to communicate

**Turbot Security Groups – {purpose}_{source}**

Turbot automatically manages a standard set of security groups for the VPC.

By combining the Purpose and the Source, and considering the Subnet Types used in the VPC, a standard set of security groups is calculated and applied. For example:
- **web_intranet**: Access to web ports (80/443) from within the intranet.
- **web_public**: Access to web ports (80/443) from the Internet.
- **db_private**: Access to the database from within the VPC.

Application stacks can then use combinations of these to create secure configurations. For example, a common Intranet application:

- Intranet facing load balancer: `web_intranet`
- Intranet facing web server: `web_intranet`
- Application server only accessed within a VPC: `app_private`
- Database server only accessed within a VPC: `db_private`

**Turbot Security Groups – (purpose)_pair**

In some cases, we wish to pair servers for access, making them more specifically connected than the functional groups provide. For example, a DMZ based Elastic LoadBalancer can be paired with an Intranet based Web server for specific access.

Turbot provides pair security groups for this purpose:

- **web_pair**: All resources with the `web_pair` security groups allow inbound access to web ports (80,443) from other instances in the `web_pair` security group.
- **db_pair**: e.g. All Oracle databases sharing `db_pair` can communicate with each other on port 1521.
- **app_pair**: e.g. Would allow an application load balancer to talk to all app server instances sharing the `app_pair` security group.

The pair security group is defined with ingress & egress rules to itself for the specific function. For example, `web_pair` is defined as:

- Allow ingress on TCP-80 from the `web_pair` security group.
- Allow ingress on TCP-443 from the `web_pair` security group.
- Allow egress on TCP-80 to the `web_pair` security group.
- Allow egress on TCP-443 to the `web_pair` security group.

An outward facing web site can make use of pairing to allow communication across the DMZ. Note that without the `web_pair` connection this outward facing load balancer has no access to any resources in the VPC.

- Public facing load balancer in outward facing subnets: `web_public & web_pair`.
- Web server in inward facing subnets: `web_pair`.
- App server in inward facing subnets: `app_private`.
- DB server in inward facing subnets: `db_private`.

Pairs can be used throughout the stack to prevent any unwanted access. For example, to prevent the Web server having any possible access to the DB server use this model:
• Public facing load balancer in outward facing subnets: web_public & web_pair.
• Web server in inward facing subnets: web_pair & app_pair.
• App server in inward facing subnets: app_pair & db_pair.
• DB server in inward facing subnets: db_pair.

Pairs are simple, yet powerful. This functionality is enabled by the Turbot best practice of separating each application into their own VPCs.

**Turbot Security Group Policies**

Per the architecture above, Turbot defines and manages common, best practice security groups. These groups are:

- Continuously enforced, preventing any drift from desired state.
- Defined centrally and automatically configured across accounts and VPCs.
- Automatically managed according to the subnet purpose.
- Connected with Turbot guardrails for use by appropriate resources only.

**Policies – Turbot Security Groups**

Turbot Managed Security Groups are managed per:

- AWS > VPC > Security Group Standard Rules

**Policies – Advanced Definitions**

Turbot Security Group definitions are built in to Turbot, no policies are available to modify them.

**Policies – Advanced Security Group Management**

Optionally, to help prevent conflicts and clarify management, each security group managed by Turbot can have a defined prefix. By default, the prefix is empty to produce standard, short form names.

- AWS > VPC > Security Group Turbot Name Prefix

The security group templates in Turbot leverage keywords for both traffic sources and destinations – e.g. vpc, internet, intranet, bastion. Turbot automatically interprets and expands these keywords into CIDR ranges or Security Group sources relative to the location of the security group. For example, vpc may become 10.1.0.0/24 in account aab while vpc may become 172.28.0.0/16 in account aac. As with the VPC policies, CIDR Range definitions default back to the Turbot > Network settings, but they can be set specifically in advanced cases with:
• VPC > Security Group Bastion CIDR Ranges
• VPC > Security Group Internet CIDR Ranges
• VPC > Security Group Intranet CIDR Ranges

Policies – Purpose Types

Customization of the ports for each Purpose is made available in
• {purpose} > Ports – Ingress ports to open.
• {purpose} > Sources – List of Subnet Types to enable for this Purpose.

Turbot Managed Security Group Policies

Security Groups defined by the Customer can be automatically managed by Turbot – providing extensibility to the Turbot Security Groups model above.

Policies – Turbot Managed Security Groups

Turbot Managed Security Groups are managed per:
• AWS > VPC > Security Groups

Turbot provides a simple YAML format to define security groups and manage them by policy. Customers may add as many rules as required, and they will be automatically adjusted and deployed by Turbot for each VPC:
• VPC > Security Group Extended Rules

Policies – Advanced Security Group Management

Please see Turbot Security Groups, Policies – Advanced Security Group Management. The process and rules for managed security groups are shared.

Security Group Policies

Security groups can be used and managed like any other resource. Application teams are granted permission to use security groups, with their configuration subject to guardrails defined in Turbot.

Policies – Permissions for Security Group Management

Please see Managing Security Groups at Account level with VPC/* Rights below.
The “default” Security Group

So far in this guide we have focused on security groups with a specific purpose – e.g. web_intranet to allow web traffic from the intranet.

The default security group is created by AWS and cannot be deleted. Turbot recommends it is carefully configured to act as a true default security group for your VPC resources. In many environments, it’s enforced as a required security group for EC2 servers.

In our tightly restricted network model, the default security group provides:

- Common egress rules (e.g. Active Directory, web).
- Security based ingress rules (e.g. security scanners, SSH, RDP).

Turbot provides guardrails to ensure that the default security groups are always used appropriately, for example:

- EC2 instances in inward facing subnets MUST have the default security group.

Policies – Default Security Group

The default security group rules are initially defined by Turbot, but usually require customization to the specific customer environment using:

- VPC > Security Group Default Rules
- VPC > Security Group Rules Manage Default

Managing Security Groups at Account level with VPC/* Rights

Policies – Permissions for Security Group Management

Security Groups are a difficult resource to define. Sometimes they are managed with resources (e.g. EC2 instances); and sometimes they are centrally controlled (e.g. Network Team).

Permission management for VPC Security Groups is defined by:

- AWS > VPC > Enabled
- AWS > VPC > Rights
- AWS > VPC > Security Group Management
But, because Security Groups are a subset of VPC, it’s also important to consider the impact of these policies on other VPC resource types. Management policies for other VPC resources must also be considered and reviewed:

- AWS > VPC > {resource} Management

**Policies – Permissions if VPC & SG are both centrally managed**

Block all VPC management by the application team, including management of security groups. Security groups can be used by the application team for their resources, but cannot create or edit them. For this scenario, a best practice setup is:

- AWS > VPC > Enabled – Required: Disabled
- AWS > VPC > Rights – Required: Enabled if AWS > VPC > Enabled

Although the Management policies for the VPC are not used in this case (Rights is effectively disabled), it’s good practice to set them to match the AWS > VPC > Enabled posture:

- AWS > VPC > {resource} Management – Required: Enabled if AWS > VPC > Enabled

**Policies – Permissions if VPC centrally managed, but SG managed by app team**

This case requires all VPC management to be denied to the application team, except for security groups. To achieve this, we need to enable rights management but deny most resources from being managed.

- AWS > VPC > Enabled – Required: Disabled
- AWS > VPC > Rights – Required: Enabled
- AWS > VPC > Security Group Management – Required: Enabled

All other VPC resources must be denied management. As above, this is simplified through the reference to AWS > VPC > Enabled (which is set to Disabled):

- AWS > VPC > {resource} Management – Required: Enabled if AWS > VPC > Enabled

**Policies – VPC & SG managed by app team**

Per the definition in Network Guardrails for AWS, VPC permissions can be granted at Account level with:

- AWS > VPC > Enabled – Required: Enabled
- AWS > VPC > Rights – Required: Enabled if AWS > VPC > Enabled
- AWS > VPC > {resource} Management – Required: Enabled if AWS > VPC > Enabled
Approved Security Groups

Guardrails for customer managed Security Groups are defined with standard Turbot resource policies and guardrails.

Note: All Turbot Security Groups and Turbot Managed Security Groups are skipped for the Approved guardrails. They are managed exactly through their definition instead.

Policies – Approved

Existence of the actual security group is subject to:
- VPC > Security Group Approved
- VPC > Security Group Approved Usage

Rules within the security group are subject to:
- VPC > Security Group Rules Approved
- VPC > Security Group Rules Approved Egress Blacklisted Ports
- VPC > Security Group Rules Approved Egress Minimum Bitmask
- VPC > Security Group Rules Approved Egress Restrict Maximum Port Range
- VPC > Security Group Rules Approved Egress Restrict To Sources
- VPC > Security Group Rules Approved Ingress Blacklisted Ports
- VPC > Security Group Rules Approved Ingress Minimum Bitmask
- VPC > Security Group Rules Approved Ingress Restrict Maximum Port Range
- VPC > Security Group Rules Approved Ingress Restrict To Sources

Tag management

Policies - Tags

As usual, Turbot provides a standard model for automatic tagging of Security Group resources:
- VPC > Security Group Tags Template: Define the tags to be set on the resource.
- VPC > Security Group Tags: Skip, Check or Enforce the Template.
About Turbot

Our mission at Turbot is to unlock the value of public cloud for enterprises through innovation, insight and speed. Turbot has built a strong brand and reputation by hiring the right people, continually innovating and enhancing the product, and providing unparalleled customer service.

Contact Us

Email: info@turbot.com
Twitter: @turbothq
LinkedIn: https://www.linkedin.com/company/turbot
Phone: +1 973 922 0297
Address: https://turbot.com

Turbot HQ, Inc
1732 1st Ave #20232
New York, NY 10128 USA