CONTROLLING VPC EGRESS TRAFFIC

"How do I implement VPC egress controls?"

Overview

Amazon Web Services (AWS) offers customers different methods for securing resources in their Amazon Virtual Private Cloud (Amazon VPC) networks. One important security measure is to effectively control inbound (*ingress*) and outbound (*egress*) VPC network traffic in order to distinguish between legitimate and illegitimate requests. If internal servers are compromised, they can pose a threat to a larger network of resources—especially when attempting to steal sensitive data or communicate with command and control systems. This document provides AWS customers with best practices and common approaches for controlling egress traffic as part of a holistic network security strategy. See the VPC Security Capabilities Solution Brief for broader network security recommendations.

The following sections address key considerations and recommendations for managing network egress traffic from VPCs and assume some basic knowledge of Amazon VPC, Amazon Elastic Compute Cloud (Amazon EC2), Virtual Private Network (VPN) connections, high availability (HA) remote network connectivity,[1] network addressing, subnetting, routing, and AWS Direct Connect.

General Best Practices

Consider the following universal network design principles when implementing network egress controls:

- Network security controls should meet or exceed application availability and scalability requirements.
- Network egress controls should address risks associated with internal users or systems communicating with unauthorized external hosts.
- A defense-in-depth strategy should leverage a combination of host-based, network, and monitoring controls.

Application on AWS

When deciding on an appropriate network egress control design, it is important to leverage AWS native routing and firewall capabilities to control network traffic. It is also important to consider how to complement AWS network security features with third-party network gateway and host-based tools. The following sections provide an overview of both AWS and third-party options for controlling VPC egress traffic.

AWS Network Egress Control Capabilities

The following sections describe key Amazon VPC features that AWS customers can use to control egress traffic.

Subnet-Level Routing Rules

Each VPC has subnet route tables with defined rules that control the flow of traffic out of the VPC, and each individual subnet can have different traffic routing rules. When designing VPC subnets, consider which public-facing EC2 instances require direct access to the internet and which do not. For example, public load balancers, proxy servers, and network gateways require direct internet access while internal load balancers, application servers, and database servers typically do not. Place all EC2 instances that do not require direct access to the internet in private subnets so their egress traffic can be directed to outbound network gateways with routing rules. This reduces a network's attack surface and allows for additional network controls such as network address translation (NAT) or deep packet inspection.

Security Groups, Network ACLs, and VPC Flow Logs

All VPC network traffic is subject to stateful EC2 instance security group rules and stateless subnet network access control list (ACL) rules. Leverage outbound security group rules (and optionally network ACLs) to control which external hosts, ports, and networks an EC2 instance is authorized to contact. This controls egress traffic by restricting unauthorized outbound network connectivity. Additionally, VPC Flow Logs provide visibility into both authorized and

[1] See the *Resources* section for Solution Briefs that offer additional HA guidance.

unauthorized network connections. Flow log analysis can help identify unexpected traffic or connection attempts indicating attempts to circumvent network egress controls. See Resources for additional Solution Briefs describing security groups, network ACLs, and network monitoring in more detail.

VPC Endpoints

VPC endpoints are horizontally scaled, redundant, highly available virtual devices that provide private connectivity between EC2 instances in a VPC and supported AWS services or VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. VPC endpoints enable customers to use routing policies, security groups, endpoint policies, and service-specific policies to control traffic destined for supported AWS services. Route table configuration allows VPC endpoint routes to be made available to specific subnets, while not to others. Security groups allow outbound traffic, such as HTTPS traffic, to be restricted to specific VPC endpoints. Both routing policies and security groups leverage service prefix lists, which logically represent the range of public IP addresses used by the service. Endpoint and service policies enable customers to implement granular service access policies, further restricting which endpoints or service resources may be accessed.

For example, a VPC endpoint for Amazon Simple Storage Service (Amazon S3) allows you to carefully control EC2 instance access to the service. Customers can use VPC route tables and security groups to restrict outbound HTTPS access and limit web traffic to a VPC endpoint. The endpoint, in turn, can restrict Amazon S3 access to specific S3 buckets. And, finally, S3 bucket policies can restrict access to specific S3 endpoints.

AWS NAT Gateway

Customer EC2 instances in a private subnet sometimes need to communicate with the public Internet. A NAT device enables this connection, replacing internal servers' private IP addresses with public IP addresses on the way out of the network, and retranslating response IP addresses on the way back in. AWS offers two types of NAT options: NAT gateways and NAT instances. NAT gateways are AWS managed while customers are responsible for managing NAT instances. NAT gateways provide better availability and bandwidth over individual NAT instances, however customers can leverage multiple NAT instances to increase availability and network performance. Third-party options provide additional network traffic control and monitoring capabilities. See In-Line Gateways for more information about this option.

Note that security groups cannot be directly associated with a NAT gateway. Instead, customers can use EC2 instance security groups outbound rules to control authorized network destinations or leverage a network ACL associated with the NAT gateway's subnet to implement subnet-level controls over NAT gateway traffic.

Additional Egress Control Options

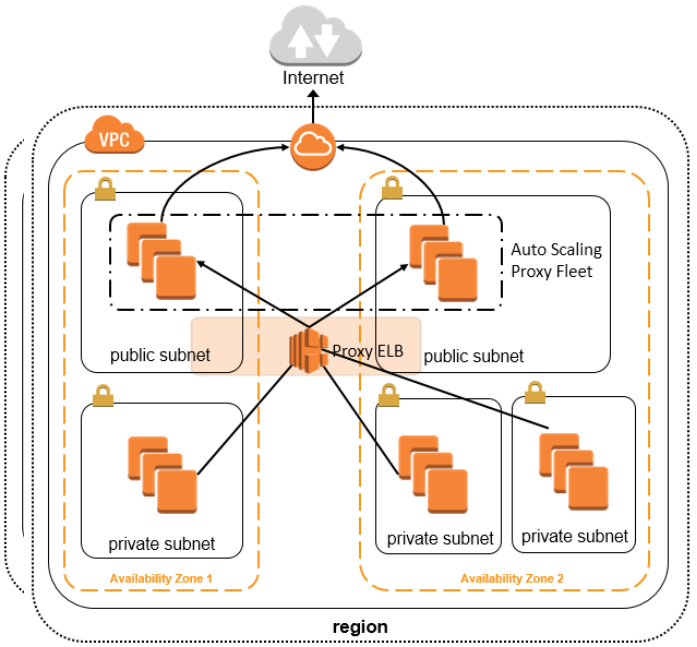
In addition to AWS-provided network controls, AWS partners provide additional tools to control both ingress and egress traffic. These tools provide network security capabilities that are often required by customers looking to perform additional levels of packet inspection for network intrusion detection and prevention (IDP), implement data loss prevention services (DLP), or perform application-level traffic filtering. These solutions can be grouped in to the following categories: host-based solutions, forward proxy servers, and in-line gateways.

Host-Based Solutions

Host-based network security solutions include OS-native capabilities, such as *iptables* or Windows Firewall, and third-party security software providing firewall, deep packet inspection, IDP, DLP, and system integrity checking capabilities. Many AWS customers leverage host-based network security tools because this approach scales linearly as EC2 instances are added or removed from server farms, does not introduce potential network single-points-of-failure or bottlenecks, and does not require application level configuration changes such as configuring proxy settings. AWS Partner Network members Alert Logic and TrendMicro provide fully managed network security services for support installing, configuring, and managing network security data collected from their host-based agents. Additional third-party options include Sourcefire Snort, Tripwire, OSSEC, and McAfee HIDS.

Forward Proxy Servers

A forward proxy server acts as an intermediary for requests from internal users and servers, often caching content to speed up subsequent requests. Companies usually implement proxy solutions to provide URL and web content filtering, IDS/IPS, data loss prevention, monitoring, and advanced threat protection. AWS customers often use a VPN or AWS Direct Connect connection to leverage existing corporate proxy server infrastructure, or build a forward proxy farm on AWS using software such as Squid proxy servers with internal Elastic Load Balancing (ELB).



Web proxy servers are the most common type of proxy server used today. Web proxies control HTTP and HTTPS traffic and have ubiquitous support from web clients such as web browsers, web command line tools, programming tools, and web application servers. SOCKS proxy servers, although less common than web proxies, leverage custom SOCKS proxy clients to control any type of IP network traffic. In either case, each EC2 instance must be configured (typically through initial instance bootstrapping or application deployment and configuration) to leverage the proxy solution at either the OS or application level.

In-Line Gateways

A traditional approach to network security is to direct network traffic through specific network filtering and monitoring gateways. Amazon VPC enables customers to route outbound network traffic through in-line gateway appliances on EC2 instances, which in turn route the traffic to its intended destination. AWS customers use this feature to control outbound traffic in a variety of ways, from providing simple NAT services to implementing next generation firewalls and Universal Threat Management (UTM) gateways. AWS Marketplace partners offering in-line network security appliances include Aviatrix, Check Point, FortiNet, Palo Alto Networks, and Sophos.

In-line gateways force egress traffic through predefined network choke-points, which provide additional security capabilities but can also introduce potential single points of failure or bandwidth bottlenecks into the network. Therefore, it is important to design gateway-based solutions that leverage multiple subnets and Availability Zones (AZs), Auto Recovery for EC2, or availability monitoring and recovery scripts.

Fully Qualified Domain Name (FQDN) Filtering

For typical AWS workloads, the destination of outbound traffic is known. For example, an application may connect with a hosted database service or a file-sharing service. For this type of workload, specifying policies by IP may not be practical because domain names can often be translated to many different IP addresses, and maintaining security groups at each server instance can be challenging.

Filtering outbound traffic by an expected list of domain names can be an efficient way to secure egress traffic from a VPC because the hostnames of these services are typically known at deployment, the list of hosts that need to be accessed by an application is small and does not change often, and hostnames rarely change. Filtering traffic by a list of domain names also enables companies to centralize the maintenance and deployment of rules.

For more information on how to build a highly available, secure FQDN Egress Filtering service on the AWS Cloud in minutes, see the Aviatrix FQDN Egress Filtering Quick Start.

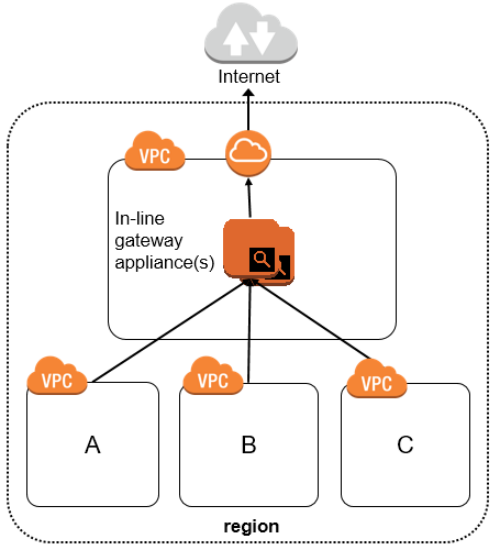
High Availability Options

VPC subnets support a single default route destination, which means customers can only scale in-line gateways horizontally by adding additional subnets, each supported by a different EC2 instance gateway appliance. AWS recommends, at a minimum, implementing one in-line gateway per AZ to provide additional availability and capacity, and independently control traffic from each AZ. If additional outbound capacity is needed, create additional private subnets with a default outbound route directing traffic to additional in-line gateways.

Physical hardware failures or software issues within the virtual machine hosting the gateway appliance can affect network egress availability for the individual subnet that it supports. For critical infrastructure components such as egress gateways, use Auto Recovery to automatically recover EC2 instances in the event of underlying hardware failures or a non-responsive OS. Additionally, consider incorporating AWS network APIs into custom network monitoring scripts to automatically recover failed gateway appliances or redirect routing paths to healthy instances. We also recommend checking with the appliance provider for any high availability mechanisms they have built for their product; you may not need to write your own custom monitoring scripts.

Multi-VPC Design Considerations

As a company expands its deployment to multiple VPCs, it must consider traffic latency, performance, and overall cost when implementing egress controls. It can be complex and cost prohibitive to deploy in-line gateways for each subnet or VPC. An alternative solution is to route outbound network traffic from all VPCs in the same AWS Region to a transit egress VPC hosting redundant security appliances. This method creates a regional hub-and-spoke network topology that uses virtual gateways (VGWs) or EC2 network appliances in each VPC to forward traffic to regional egress gateways. This design minimizes the total number of in-line security appliances and enables centralized management and operation of these devices. AWS Marketplace partners, including Aviatrix, Check Point, FortiNet, Juniper, Palo Alto Networks, and Sophos offer different solutions for implementing this design. For additional guidance on implementing hub-and-spoke networks on AWS, see AWS Global Transit Network.



Resources

VPC Security Capabilities	https://d0.awsstatic.com/aws-answers/VPC_Security_Capabilities.pdf
VPC Network Management and Monitoring	https://d0.awsstatic.com/aws-answers/VPC_Network_Management_and_Monitoring.pdf
VPC Endpoints Documentation	http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html
NAT Gateway Documentation	http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html
NAT Instances Documentation	http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_NAT_Instance.html

EC2 Auto Recovery Documentation http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-recover.html