

Introduction to Secure Firewall ASA-Firewall Services

Firewall services are those ASA features that are focused on controlling access to the network, including services that block traffic and services that enable traffic flow between internal and external networks. These services include those that protect the network against threats, such as Denial of Service (DoS) and other attacks.

The following topics provide an overview of firewall services.

- How to Implement Firewall Services, on page 1
- Basic Access Control, on page 2
- URL Filtering, on page 2
- Threat Protection, on page 2
- Firewall Services for Virtual Environments, on page 3
- Network Address Translation, on page 3
- Application Inspection, on page 4
- Use Case: Expose a Server to the Public, on page 4

How to Implement Firewall Services

The following procedure provides a general sequence for implementing firewall services. However, each step is optional, needed only if you want to provide the service to your network.

Before you begin

Configure the ASA according to the general operations configuration guide, including at minimum basic settings, interface configuration, routing, and management access.

Procedure

Step 1 Implement access control for the network. See Basic Access Control, on page 2.

Step 2 Implement URL filtering. See URL Filtering, on page 2.

Step 3 Implement threat protection. See Threat Protection, on page 2.

Step 4 Implement firewall services that are tailored to virtual environments. See Firewall Services for Virtual Environments, on page 3.

Step 5 Implement Network Address Translation (NAT). See Network Address Translation, on page 3.

Step 6 Implement application inspection if the default settings are insufficient for your network. See Application Inspection, on page 4.

Basic Access Control

Access rules, applied per interface or globally, are your first line of defense. You can drop, upon entry, specific types of traffic, or traffic from (or to) specific hosts or networks. By default, the ASA allows traffic to flow freely from an inside network (higher security level) to an outside network (lower security level).

You can apply an access rule to limit traffic from inside to outside, or allow traffic from outside to inside.

Basic access rules control traffic using a “5-tuple” of source address and port, destination address and port, and protocol. See Access Rules and Access Control Lists.

You can augment your rules by making them identity aware. To implement identity control, install Cisco Identity Services Engine (ISE) on a separate server to implement Cisco Trustsec. You can then add security group criteria to your access rules. See ASA and Cisco TrustSec.

URL Filtering

URL filtering denies or allows traffic based on the URL of the destination site.

To implement URL filtering, subscribe to the Cisco Umbrella service, where you configure the Enterprise Security policy to block malicious sites based on the fully-qualified domain name (FQDN). For FQDNs that are considered suspicious, you can redirect user connections to the Cisco Umbrella intelligent proxy, which performs URL filtering. The Umbrella service works by handling users' DNS lookup requests, returning the IP address for a block page or the IP address of the intelligent proxy. The service returns the real IP address for an FQDN for allowed domains. See Cisco Umbrella.

Threat Protection

You can implement a number of measures to protect against scanning, denial of service (DoS), and other attacks. A number of ASA features help protect against attacks by applying connection limits and dropping abnormal TCP packets. Some features are automatic, others are configurable but have defaults appropriate in most cases, while others are completely optional and you must configure them if you want them.

Following are the threat protection services available with the ASA.

- IP packet fragmentation protection—The ASA performs full reassembly of all ICMP error messages and virtual reassembly of the remaining IP fragments that are routed through the ASA, and drops fragments that fail the security check. No configuration is necessary.
- Connection limits, TCP normalization, and other connection-related features—Configure connection-related services such as TCP and UDP connection limits and timeouts, TCP sequence number

randomization, TCP normalization, and TCP state bypass. TCP normalization is designed to drop packets that do not appear normal. See Connection Settings.

For example, you can limit TCP and UDP connections and embryonic connections (a connection request that has not finished the necessary handshake between source and destination). Limiting the number of connections and embryonic connections protects you from a DoS attack. The ASA uses the embryonic limit to trigger TCP Intercept, which protects inside systems from a DoS attack perpetrated by flooding an interface with TCP SYN packets.

- Threat detection—Implement threat detection on the ASA to collect statistics to help identify attacks. Basic threat detection is enabled by default, but you can implement advanced statistics and scanning threat detection. You can shun hosts that are identified as a scanning threat. See Threat Detection.

Firewall Services for Virtual Environments

Virtual environments deploy servers as virtual machines, for example, in VMware ESXi. The firewalls in a virtual environment can be traditional hardware devices, or they can also be virtual machine firewalls, such as the ASA virtual.

Traditional and next-generation firewall services apply to virtual environments in the same way that they apply to environments that do not use virtual machine servers. However, virtual environments can provide additional challenges, because it is easy to create and tear down servers.

Additionally, traffic between servers within the data center might require as much protection as traffic between the data center and external users. For example, if an attacker gains control of a server within the data center, that could open up attacks on other servers in the data center.

Firewall services for virtual environments add capabilities to apply firewall protection specifically to virtual machines. Following are the firewall services available for virtual environments:

- Attribute-based access control—You can configure network objects to match traffic based on attributes, and use those objects in access control rules. This lets you decouple firewall rules from network topology. For example, you can allow all hosts with the Engineering attribute to access hosts with the Lab Server attribute. You could then add/remove hosts with these attributes and the firewall policy would be applied automatically without the need for updating access rules. For more information, see Attribute-Based Access Control.

Network Address Translation

One of the main functions of Network Address Translation (NAT) is to enable private IP networks to connect to the Internet. NAT replaces a private IP address with a public IP address, translating the private addresses in the internal private network into legal, routable addresses that can be used on the public Internet. In this way, NAT conserves public addresses because you can advertise at a minimum only one public address for the entire network to the outside world.

Other functions of NAT include:

- Security—Keeping internal IP addresses hidden discourages direct attacks.
- IP routing solutions—Overlapping IP addresses are not a problem when you use NAT.

- Flexibility—You can change internal IP addressing schemes without affecting the public addresses available externally; for example, for a server accessible to the Internet, you can maintain a fixed IP address for Internet use, but internally, you can change the server address.
- Translating between IPv4 and IPv6 (Routed mode only)—If you want to connect an IPv6 network to an IPv4 network, NAT lets you translate between the two types of addresses.

NAT is not required. If you do not configure NAT for a given set of traffic, that traffic will not be translated, but will have all of the security policies applied as normal.

See:

- Network Address Translation (NAT)
- NAT Examples and Reference

Application Inspection

Application inspection engines are required for services that embed IP addressing information in the user data packet or that open secondary channels on dynamically assigned ports. These protocols require the ASA to do a deep packet inspection, to open the required pinholes and to apply network address translation (NAT).

The default ASA policy already applies inspection globally for many popular protocols, such as DNS, FTP, SIP, ESMTP, TFTP, and others. The default inspections might be all you require for your network.

However, you might need to enable inspection for other protocols, or fine-tune an inspection. Many inspections include detailed options that let you control packets based on their contents. If you know a protocol well, you can apply fine-grained control on that traffic.

You use service policies to configure application inspection. You can configure a global service policy, or apply a service policy to each interface, or both.

See:

- Service Policy
- Getting Started with Application Layer Protocol Inspection
- Inspection of Basic Internet Protocols
- Inspection for Voice and Video Protocols
- Inspection for Mobile Networks.

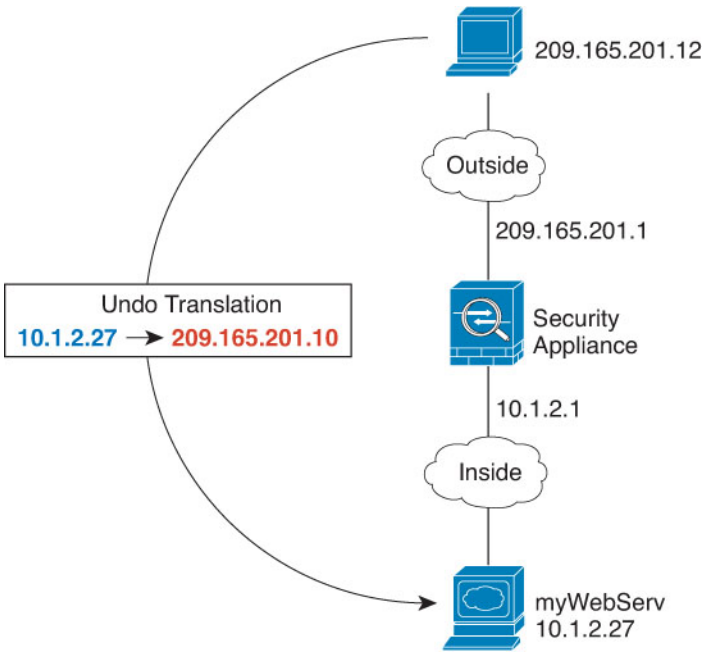
Use Case: Expose a Server to the Public

You can make certain application services on a server available to the public. For example, you could expose a web server, so that users can connect to the web pages but not make any other connections to the server.

To expose a server to the public, you typically need to create access rules that allow the connection and NAT rules to translate between the server's internal IP address and an external address that the public can use. In addition, you can use port address translation (PAT) to map an internal port to an external port, if you do not want the externally exposed service to use the same port as the internal server. For example, if the internal web server is not running on TCP/80, you can map it to TCP/80 to make connections easier for external users.

The following example makes a web server on the inside private network available for public access.

Figure 1: Static NAT for an Inside Web Server



Procedure

Step 1 Create a network object for the internal web server.

```
hostname(config)# object network myWebServ
hostname(config-network-object)# host 10.1.2.27
```

Step 2 Configure static NAT for the object:

```
hostname(config-network-object)# nat (inside,outside) static 209.165.201.10
```

Step 3 Add an access rule to the access group attached to the outside interface to permit web access to the server.

```
hostname(config)# access-list outside_access_in line 1 extended
permit tcp any4 object myWebServ eq http
```

Step 4 If you do not already have an access group on the outside interface, apply it using the access-group command:

```
hostname(config)# access-group outside_access_in in interface outside
```