Introduction

This guide describes the firewall and NAT features on the AR-Series firewalls and how to configure them.

The firewall feature on the AR-Series UTM and VPN firewalls offers security, flexibility and ease of use. Unlike a traditional firewall, they will keep pace with rapid changes in Internet-based applications, enabling enterprises to see the benefits of web-based technology without costly security issues.

The AR-Series firewalls also supports Network Address Translation (NAT), allowing a single device to act as an agent between the public Internet and a local private network. With NAT, private (RFC1918) IPv4 addresses can be configured on devices located on the private side of the firewall. When those devices send traffic to the Internet, the firewall translates the private addresses to become one or more publicly-valid addresses. When the firewall receives traffic that is destined for those devices, it translates the public address back to the appropriate private address.

This document gives an overview of the firewall and NAT on AR-Series firewalls, followed by examples illustrating how to configure them in various network situations.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Products and software version that apply to this guide</td>
<td>3</td>
</tr>
<tr>
<td>Related documents</td>
<td>3</td>
</tr>
<tr>
<td>Advanced Feature Licences</td>
<td>3</td>
</tr>
<tr>
<td>The Firewall</td>
<td>4</td>
</tr>
<tr>
<td>Firewall GUI</td>
<td>6</td>
</tr>
<tr>
<td>Accessing the AR-Series Firewall GUI</td>
<td>6</td>
</tr>
<tr>
<td>Applications</td>
<td>7</td>
</tr>
<tr>
<td>Application Layer Gateways (ALG)</td>
<td>7</td>
</tr>
<tr>
<td>Entities</td>
<td>8</td>
</tr>
<tr>
<td>Network Address Translation (NAT)</td>
<td>9</td>
</tr>
<tr>
<td>Configuring Firewall and NAT Rules for Entities</td>
<td>11</td>
</tr>
<tr>
<td>Firewall with Dynamic IP Addressing</td>
<td>14</td>
</tr>
<tr>
<td>Configuring Firewall Rules to Interact with Update Manager</td>
<td>15</td>
</tr>
<tr>
<td>Firewall with High Availability</td>
<td>16</td>
</tr>
<tr>
<td>Configuring NAT Loopback with DMZ</td>
<td>17</td>
</tr>
<tr>
<td>Static ENAT rule</td>
<td>19</td>
</tr>
<tr>
<td>Dynamic ENAT rule</td>
<td>19</td>
</tr>
<tr>
<td>Configuring Static NAT with Proxy ARP</td>
<td>20</td>
</tr>
<tr>
<td>Source-based NAT with Secondary IP Addresses</td>
<td>22</td>
</tr>
<tr>
<td>Configuring Source and Destination NAT (Double NAT)</td>
<td>24</td>
</tr>
<tr>
<td>Interaction with firewall</td>
<td>25</td>
</tr>
<tr>
<td>Configuring Subnet-based NAT</td>
<td>26</td>
</tr>
<tr>
<td>Allowing Partial Sessions through a Firewall</td>
<td>30</td>
</tr>
</tbody>
</table>
Products and software version that apply to this guide

This Guide applies to the AR-Series firewalls running AlliedWare Plus version 5.4.5 or later:

- AR4050S UTM Firewall
- AR3050S UTM Firewall
- AR2050V VPN Firewall
- AR2010V VPN Firewall

Most features described in this document are supported from AlliedWare Plus 5.4.5 or later. These features are available in version 5.4.7-0.1 or later:

- Subnet-based NAT
- Source and destination NAT
- Allowing partial sessions through a firewall (no state enforcement)

This feature is available in version 5.4.6-2.1 or later:

- Firewall with High Availability (VRRP)

Related documents

The following documents provide information about related features on AlliedWare Plus products:

- Getting Started with the UTM Firewall GUI Feature Overview Guide
- Getting Started with the VPN Firewall GUI Feature Overview Guide
- The product’s Datasheet
- The product’s Command Reference

These documents are available from the links above or on our website at alliedtelesis.com

Advanced Feature Licences

Flexible subscription licensing options make it easy to choose the right combination of security features to best meet your business needs. The Advanced Firewall license includes Application Control, Web Control and URL Filtering. The Advanced Threat Protection (ATP) license includes IP Reputation, stream-based Malware Protection and (on the AR4050S only) proxy-based Antivirus.
The Firewall

A firewall, at its most basic level, controls traffic flow between a trusted network (such as a corporate LAN) and an untrusted or public network (such as the Internet). The most commonly deployed firewalls nowadays are port-based or packet filtering. These traditional firewalls determine the allowed traffic versus the disallowed traffic based on many characteristics of the packets, including their destination and source IP addresses and TCP/UDP port numbers. However, traditional network security solutions have failed to keep pace with changes to applications, threats, and the network landscape.

AR-Series firewalls are designed for the challenges facing modern networks. In contrast to traditional firewalls that lack the intelligence to discern network traffic in a world where network boundaries are disintegrating and Internet applications are exploding, AR-Series firewalls no longer talk about packets, IP addresses and ports. Instead they focus on applications, users and content. It classifies traffic by the application’s identity in order to enable visibility and control of all types of application.

The AR-Series firewalls view the physical network in terms of zones, networks and hosts. Firewall rules can be applied to any level of this hierarchy, as shown in Figure 1 on page 5. See “Entities” on page 8 for entity definitions and usage.

When the firewall is enabled, its default policy is to drop all applications from anywhere to anywhere. If no rule is explicitly configured, all traffic moving through the firewall is blocked.

The firewall filters traffic by identifying applications. The application-centric traffic classification identifies specific applications flowing across the network regardless of the port and protocol in use.

The firewall identifies applications through a database of regularly updated application signatures. Deep Packet Inspection (DPI) is used by the firewall to match packets against these signatures and provide layer 7 filtering for firewall rules. See “Applications” on page 7 for application definition and usage.
The firewall provides the following features:

- **Stateful inspection** maintains the status of active connections through the firewall to dynamically allow inbound replies to outbound connections.
- **Robust application identification and inspection** enables granular control of the flow of sessions through a firewall, based on the specific applications that are being used.
- **Rules** allow specified traffic to be matched and the appropriate action applied.
- **Network Address and Port Translation** permits multiple hosts on a LAN to be mapped to a single public IP address and hides details of the internal network.
- **OpenVPN integration** provides secure remote access to intranet resources.
- **Application Layer Gateway (ALG)** inspects the application layer payload of a packet and understands the application control messages, and performs Network Address Translation processing if necessary.
- **Logs** allow retrieval of all event details for later analysis.
- **Reports of network usage and statistics** give network managers the information they need to effectively manage their networks.
Firewall GUI

If you want to you can use the Firewall GUI to monitor and configure your AR-Series firewall.

The firewall GUI provides setup of the firewall, enabling the configuration of entities (zones, networks and hosts) and then creating firewall, NAT and traffic-control rules for managing traffic between these entities. Features such as the Intrusion Prevention System (IPS) and URL Filtering help protect the network, and manage website access.

The GUI also supports a DHCP server, interface management, VLAN management, system tools, a CLI window and a dashboard for network monitoring. The dashboard shows interface and firewall traffic, system and environmental information, and the security monitoring widget lets you view and manage rules and security features.

Accessing the AR-Series Firewall GUI

If your AR-Series firewall came with the GUI pre-installed, perform the following steps to browse to the GUI:

1. Connect to any of the LAN switch ports
2. Open a web browser and browse to https://192.168.1.1. This is the pre-configured IP address of VLAN1. The default username is manager and the default password is friend.

If your AR-Series firewall did not come with the GUI pre-installed, perform the following steps through the command-line interface:

3. Create one or more IP interfaces and assign them IP addresses, including configuring WAN connectivity. For information about configuring PPP, see the PPP Feature Overview and Configuration Guide. For information about configuring IP, see the IP Feature Overview and Configuration Guide.

4. If you plan to enable the firewall functionality, first create firewall rules to allow both DNS and HTTPS traffic from the Update Manager to pass through the firewall. This is needed because AR-series firewalls block all traffic by default. The following figure shows a recommended example configuration, when WAN connectivity is through ppp0:

```
zone public
network wan
    ip subnet 0.0.0.0/0 interface ppp0
host ppp0
    ip address dynamic interface ppp0
firewall
    rule 10 permit dns from public.wan.ppp0 to public.wan
    rule 20 permit https from public.wan.ppp0 to public.wan
    protect
```
5. Use the following command to download and install the GUI:
   ```
   awplus# update webgui now
   ```

6. Enable the HTTP service:
   ```
   awplus# configure terminal
   awplus(config)# service http
   ```

7. Log into the GUI.
   Start a browser and browse to the firewall's IP address, using HTTPS. You can access the GUI via any reachable IP address on any interface.
   The GUI starts up and displays a login screen. Log in with your username and password.

**Applications**

An application is a high level abstraction of application packets being transported by network traffic. Traffic matching for applications can be achieved through the firewall by using several techniques, for example, matching packets to port numbers or searching for application signatures in flows of packets. You can configure source port, destination port, protocol, ICMP code and ICMP type for the application. An application is invalid if its protocol, source or destination are not properly configured, for example, if an application has no protocol configured, or source and destination ports are applied to protocols that are not TCP, UDP or SCTP.

By default, there are a number of predefined applications with protocols, source and destinations ports. You can use the `show application` command to display the detail of these applications.

**Application Layer Gateways (ALG)**

To determine the protocol associated with a given packet, the firewall typically looks at the IP protocol number and/or the source and destination TCP/UDP port numbers. This works well for most protocols. However, there are some protocols which use different port/IP protocol numbers at different points during communication. An example of this is FTP, which uses the well-known port 21 for negotiation but either uses the well-known port 20 or ephemeral ports for the associated data transfer.

The Application Layer Gateway (ALG) identifies data streams associated with these protocols to be processed correctly by the firewall.
The following protocols are supported by the ALG and are included in the default (predefined) application list:

- FTP
- IRC
- PPTP

The following protocols are supported by the ALG but are not included in the default application list: SNMP, GRE, SCTP, TFTP, H323 and SIP.

The protocols not included on the default application list require that a custom application be created for them (application command and associated commands, see step 4 in "Configuring Firewall and NAT Rules for Entities" on page 11.)

Alternatively, with an Advanced Firewall subscription license, you can utilize the Application Control feature which adds automatic support for thousands of applications to the application list.

**Entities**

AR-Series firewalls support application and entity-based security policies. For example, firewall and Network Address Translation (NAT) rules are applied to applications among different zone entities.

An entity is a high level abstraction of an individual network device, an individual network, or a group of networks or subnets. It is the instance that firewall and NAT policies can be applied to. There are three types of entity:

- Zone
- Network
- Host

Zone is a high level abstraction for a logical grouping or segmentation of physical networks. This is the highest level of partitioning that firewall and NAT policy can be applied to. Zone establishes the security border of your networks. A zone defines a boundary where traffic is subjected to policy restrictions as it crosses to another region of your networks. The minimum zones normally implemented would be a trusted zone for the private network behind the firewall and a untrusted zone for the Internet. Other common zones are a Demilitarized Zone (DMZ) for publicly visible web servers and a Virtual Private Network (VPN) zone for remote access users or tunnels to other networks.

A network is a high level abstraction of a logical network in a zone. This consists of the IP subnets and interfaces over which it is reachable. Subnets are grouped into networks to apply a common set of rules among the subnets.
Host is a high level abstraction of a single node in a network. This is commonly used if a particular device, for example a server, has a static IP address that needs to be specified in a firewall policy.

In addition to supporting network address translation for TCP and UDP traffic, AR-Series firewalls also support VPN pass-through. Network services that use the following protocols can traverse a NAT device.

- ESP (Encapsulation Security Payload)
- PPTP (Point to Point Tunneling Protocol)
- L2TP (Layer 2 Tunneling Protocol)
- GRE (Generic Routing Encapsulation)

**Network Address Translation (NAT)**

NAT, defined in RFC 1631, provides a solution to one of the major problems facing the Internet—IP address depletion. IP address space is limited and obtaining a large block of registered addresses is difficult. Although you can use private IP address (RFC 1918) in your internal network, private IP addresses are not routable through the Internet.

A router can act as an agent between the Internet and a local network. When you use NAT, you assign private IP addresses to hosts on the private side of the router. When those hosts send traffic, the router translates the private addresses to one or more public and valid addresses before routing the traffic. When the router receives traffic that is destined for those hosts, it translates the public addresses back to the appropriate private addresses.

AR-Series firewalls support two basic modes of NAT:

- Masquerading: Devices with non-global addresses are able to access the public network by sharing the IP address of an external facing interface. The source IP address of an outgoing packet is translated to the interfaces of external interface. The source port (TCP or UDP) is translated to a new value in order for the packet flow to be uniquely distinguishable.

- Port Forwarding: Servers on a private network are made accessible to the public network by aliasing an externally facing interface’s IP to the server’s IP address. The destination address of an incoming packets is translated from the external interface’s IP to the private server’s IP. This is an address-only translation.

AR-Series firewalls also support Enhanced NAT (ENAT) which gives you the ability to

- Configure the global address used in Masquerading and Port Forwarding.
- Perform port translations in Port Forwarding configurations.
AR-Series firewalls support the following additional methods of network address translation.

- **Static NAT**: This is a one-to-one, address-only translation. For packets originating in the private zone and destined for the public zone, the source IP address is translated. For packets originating in the public zone and destined for the NAT device's globally routable address, the destination address is translated.

- **Static ENAT**: This is a one-to-one address and port translation for packet flows initiated by a host in a public zone that is mapped through to a host in a private zone. This has a number of possible uses. For example, a difference in destination port, with the same address in the public zone can be used to distinguish between two different servers in the private zone. For whatever reason, the server in the private zone may be listening on a different port to the one advertised in the public zone.

- **Dynamic ENAT**: This is a many-to-one address translation where multiple hosts in the private zone share a globally routable address in the public zone. Source-port translation is used to provide uniqueness in the connect tracking so that return packets can be forwarded to the correct host in the private zone.

By default, NAT is disabled. You can use the `enable (NAT)` command to explicitly enable this functionality. If firewall protection is enabled, you need to configure firewall rules that allow the application matching its source and destination entities to pass through the firewall. Portfwd rules (actions) are applied before any other firewall rules and masq rules (actions) are applied after any other firewall rules. To configure NAT rules, you can use the `rule (NAT)` command.

**Figure 2: Network Address Translation**

- Public IP addresses are translated to private addresses
- Private IP addresses are translated to public address

Static NAT (public to private)  
Static ENAT (public to private)  
Static NAT (private to public)  
Dynamic ENAT (private to public)
Configuring Firewall and NAT Rules for Entities

Firewall rules are constructed as follows:

\[
\text{rule } [1-65535] \{\text{permit|deny|reject|log}\} <\text{application-name}> \text{ from } <\text{source-entity}> \text{ to } <\text{destination-entity}> \text{ [no-state-enforcement] [log]}
\]

Port forwarding and masquerade NAT rules are constructed as follows:

\[
\text{rule } [1-65535] \text{ portfw } <\text{application-name}> \text{ from } <\text{source-entity}> \text{ [to } <\text{destination-host-entity}>\text{] with dst } <\text{destination-host-entity}> \text{ [dport } 1-65535]\text{]}
\]
\[
\text{rule } [1-65535] \text{ masq } <\text{application-name}> \text{ from } <\text{source-entity}> \text{ to } <\text{destination-entity}> \text{ [with src } <\text{source-host-entity}>\text{]}
\]

The source and destination entities referenced within the rule can match a zone (\text{zone}), or a network nested within a zone (\text{zone.network}), or an individual host nested within a network (\text{zone.network.host}).

The following example shows you how to configure the firewall. The figure below shows the network topology and zone partition used by the example.

**Figure 3: Network topology and zone partition**

<table>
<thead>
<tr>
<th>Step 1: Configure DMZ zone.</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{awplus#configure terminal}</td>
</tr>
<tr>
<td>\texttt{awplus(config)#zone dmz}</td>
</tr>
<tr>
<td>\texttt{awplus(config-zone)#network servers}</td>
</tr>
<tr>
<td>\texttt{awplus(config-network)#ip subnet 172.16.0.0/24 interface eth1}</td>
</tr>
<tr>
<td>\texttt{awplus(config-host)#host ftp}</td>
</tr>
<tr>
<td>\texttt{awplus(config-host)#ip address 172.16.0.2}</td>
</tr>
<tr>
<td>\texttt{awplus(config-host)#host web-server}</td>
</tr>
<tr>
<td>\texttt{awplus(config-host)#ip address 172.16.0.10}</td>
</tr>
</tbody>
</table>
**Step 2: Configure private zone.**
awplus(config-host)#zone private
awplus(config-zone)#network lan
awplus(config-network)#ip subnet 192.168.1.0/24 interface vlan1

**Step 3: Configure public zone.**
awplus(config-host)#zone public
awplus(config-zone)#network internet
awplus(config-network)#ip subnet 0.0.0.0/0 interface eth2

**Step 4: Configure application.**
awplus(config)#application tftp
awplus(config-application)#protocol udp
awplus(config-application)#dport 69

**Step 5: Configure firewall rules.**
awplus(config)#firewall
awplus(config-firewall)#rule 100 permit ping from public to dmz
awplus(config-firewall)#rule 200 permit ping from private to dmz
awplus(config-firewall)#rule 300 permit ftp from public to dmz.servers.ftp
awplus(config-firewall)#rule 400 permit tftp from public to dmz.servers.ftp
awplus(config-firewall)#rule 500 permit http from public to dmz.servers.web-server
awplus(config-firewall)#rule 600 permit any from private to private
awplus(config-firewall)#rule 700 permit any from dmz to dmz
awplus(config-firewall)#rule 800 permit any from private to public
awplus(config-firewall)#rule 900 permit any from dmz to public

**Step 6: Enable firewall protection.**
Enable firewall protection and apply the firewall rules. This also ensures that the network administrator is not prematurely locked out of the device.
awplus(config-firewall)#protect

**Step 7: Configure Network Address Translation (NAT) rules.**
awplus(config)#nat
awplus(config-nat)#rule 10 masq any from private to public
awplus(config-nat)#rule 20 masq any from dmz to public
awplus(config-nat)#rule 30 portfwd ftp from public with dst dmz.servers.ftp
awplus(config-nat)#rule 40 portfwd http from public with dst dmz.servers.web-server
Step 8: Enable NAT to apply the NAT rules.

```bash
awplus(config-nat)#enable
```

Step 9: Configure interfaces.

```bash
awplus(config)#interface eth2
awplus(config-if)#ip address 128.0.0.1/24
awplus(config-if)#interface eth1
awplus(config-if)#ip address 172.16.0.1/24
awplus(config-if)#exit
awplus(config)#vlan database
awplus(config-vlan)#vlan 1
awplus(config-vlan)#exit
awplus(config)#interface vlan1
awplus(config-if)#ip address 192.168.1.1/24
```

Step 10: Verify Firewall configuration.

```bash
awplus#show running-config firewall
```

**Output 1: Example output from the console**

```
awplus#show running-config firewall
firewall
  rule 100 permit ping from public to dmz
  rule 200 permit ping from private to dmz
  rule 300 permit ftp from public to dmz.servers.ftp
  rule 400 permit tftp from public to dmz.servers.ftp
  rule 500 permit http from public to dmz.servers.web-server
  rule 600 permit any from private to private
  rule 700 permit any from dmz to dmz
  rule 800 permit any from private to public
  rule 900 permit any from dmz to public
  protect!
```

Step 11: Verify Entity configuration.

```bash
awplus#show entity
```

**Output 2: Example output from the console**

```
awplus#show entity
Zone:       dmz
  Network:   dmz.servers
  Subnet:    172.16.0.0/24 via eth1
  Host:      dmz.servers.ftp
    Address:  172.16.0.2
  Host:      dmz.servers.web-server
    Address:  172.16.0.10

Zone:       private
  Network:   private.lan
  Subnet:    192.168.1.0/24 via vlan1

Zone:       public
  Network:   public.internet
  Subnet:    0.0.0.0/0 via eth2
```
Step 12: Verify NAT configuration.

awplus# show nat rule

Output 3: Example output from the console

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>From</th>
<th>With (dst/src)</th>
<th>Entity</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>masq</td>
<td>private</td>
<td>-</td>
<td>public</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>masq</td>
<td>public</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>portfwd</td>
<td>public</td>
<td>dmz.servers.ftp</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>portfwd</td>
<td>public</td>
<td>dmz.servers.web-server</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Firewall with Dynamic IP Addressing

A WAN interface may obtain its IP address dynamically. For example, this might be an Ethernet interface configured as a DHCP client, or a PPP interface.

Entities and their associated rules can be configured to allow for this.

The following firewall configuration extract shows how to allow ping traffic to originate from a PPPoE WAN that has been assigned an IP address dynamically.

```configuration
! zone public
  network wan
    ip subnet 0.0.0.0/0 interface ppp1
  host router
    ip address dynamic interface ppp1
! firewall
  rule 10 permit ping from public.wan.router to public
  protect
! interface eth1
  encapsulation ppp 1
! interface ppp1
  ip address negotiated
!```
Configuring Firewall Rules to Interact with Update Manager

The Update Manager is a tool to enable an AlliedWare Plus device to be kept up to date with the latest available software components and resources. When Firewall protection is enabled, you need to create Firewall rules to permit the Update Manager traffic to be sent. For more information about the Update Manager, see the Update Manager Feature Overview and Configuration Guide.

Step 1: Configure network entity.

You can create a network entity for the Update Manager which is located on the Internet assuming that the Internet is reachable over interface ETH2.

```
awplus#configure terminal
awplus(config)#zone public
awplus(config-zone)#network INTERNET
awplus(config-zone)#ip subnet 0.0.0.0/0 interface eth2
```

Step 2: Configure entity for the Update Manager source traffic.

You can create an entity for the Update Manager source traffic which is from the interface that connects to the Internet.

```
awplus(config)#zone ROUTER
awplus(config-zone)#network EXTERNAL
awplus(config-network)#ip subnet 49.1.2.0/24 interface eth2
awplus(config-host)#host EXTERNAL_INT
awplus(config-host)#ip address 49.1.2.3
```

Step 3: Configure Firewall rules.

The Update Manager traffic uses HTTPS protocol. You can create a Firewall rule to allow HTTPS application.

```
awplus(config-host)#end
awplus#configure terminal
awplus(config)#firewall
awplus(config-firewall)#rule permit https from ROUTER.EXTERNAL.EXTERNAL_INT to public
```

Similarly, you can create a rule to allow DNS resolution of the Update Server's URL if the DNS server is reachable via the WAN interface.

```
awplus(config-firewall)#rule permit dns from ROUTER.EXTERNAL.EXTERNAL_INT to public
```
Firewall with High Availability

Firewall control of received IPv4 VRRP packets is supported from AlliedWare Plus version 5.4.6-2.1.

If you are using VRRP and you have the firewall enabled, you need to create a firewall rule to allow IPv4 VRRP packets. High Availability (HA) uses VRRP, so if you are using High Availability and the firewall, you also need to create a firewall rule to allow IPv4 VRRP packets.

The rule needs to permit packets to IP subnet 224.0.0.18/32, which is the VRRP multicast address. You can limit the rule so that it only applies to the VRRP application (protocol 112).

For example, if the firewall is enabled, and VRRP is configured on vlan1, and vlan1 has an IP address in the 172.20.10.0/24 subnet, the following configuration will allow VRRP packets to be received:

```plaintext
application vrrp
  protocol 112
  zone private
  network vlan1
    ip subnet 172.20.10.0/24 interface vlan1
  network vrrp_subnet
    ip subnet 224.0.0.18/32
firewall
  rule 10 permit vrrp from private.vlan1 to private.vrrp_subnet
  protect
```

Note that the firewall only controls incoming VRRP packets. Outgoing VRRP packets are not processed by the firewall. They will be sent regardless of the firewall configuration.
Configuring NAT Loopback with DMZ

The scenario applies when private zone clients use an external DNS (no internal DNS) and wish to access services located within a DMZ as if they were outside the office.

This example shows a three-zone network (public, private and DMZ zones) with associated firewall and NAT rules. A client is located within a private zone, and the server is located within the DMZ.

A client initiated DNS request to the domain name associated with the service resolves to the public IP address of the AR-Series firewall.

The client then sends its HTTP request to the public IP address of the AR-Series firewall. A static ENAT port forwarding rule is used to translate the destination IP to become the IP address of the server located in the DMZ ("Static ENAT rule" on page 19).

The service is accessed by sending request to the public IP address of the AR-Series firewall and that request is internally ‘looped back’ towards the DMZ server IP address via the destination address translation.

The internal IP address of the server located within the DMZ zone is also accessible when the user is physically located outside of the office and accesses the service directly via the Internet.

An optional dynamic ENAT masquerade rule can allow direct access from the server in the DMZ to hosts in the private zone. This optional rule can be used in the case where there is a need for connections to be initiated directly from a server located within the DMZ to reach private zone clients ("Dynamic ENAT rule" on page 19).

Figure 4: Physical network
Firewall and Network Address Translation

Accessing the AR-Series Firewall GUI

Figure 5: AR-Series firewall entity map

Figure 6: Configuration: NAT loopback with DMZ

```
zone dmz
    network dmz
    ip subnet 192.168.10.0/24 interface vlan2
    ip subnet 49.1.2.3/32
    host http-server
        ip address 192.168.10.10
    host router
        ip address 49.1.2.3
!
zone private
    network lan
        ip subnet 192.168.1.0/24 interface vlan1
!
zone public
    network wan
        ip subnet 0.0.0.0/0 interface eth1
!
firewall
    rule 10 permit any from private.lan to public
    rule 20 permit any from private to private
    rule 40 permit http from public to dmz.dmz.http-server
    rule 50 permit any from private.lan to dmz.dmz
    rule 60 permit http from dmz.dmz.http-server to private
    protect
!
nat
    rule 20 portfwd http from public with dst dmz.dmz.http-server
    rule 40 masq any from private.lan to public
    rule 50 masq any from dmz.dmz to public
    enable
!
vlan database
    vlan 2 state enable
!
interface port1.0.2
    switchport access vlan 2
!
interface eth1
    ip address 49.1.2.3/24
!
interface vlan1
    ip address 192.168.1.254/24
!
interface vlan2
    ip address 192.168.10.254/24
!
ip route 0.0.0.0/0 49.1.2.100
```
Static ENAT rule

Static ENAT (port-forwarding) NAT rule 10 is configured to allow traffic initiated from hosts located within the private zone to be able to access a Web Server located within the DMZ. This rule also allows associated return traffic from the web server to reach the private hosts.

Figure 7: Configuration for static ENAT port forwarding option

```
# Allow HTTP traffic going from PRIVATE.LAN (192.168.1.0/24) to DMZ.DMZ.ROUTER (49.1.2.3),
# and forward to DIPA DMZ.DMZ.HTTP-SERVER (192.168.10.10)
rule 10 portfwd http from private.lan to dmz.dmz.router with dst dmz.dmz.http-server
```

Figure 8: Static ENAT—Port Forwarding

Dynamic ENAT rule

Additionally, dynamic ENAT (masquerade) NAT rule 30 can be optionally configured, to allow traffic directly initiated from the web server located in the DMZ to reach hosts in the private zone.

Figure 9: Configuration for dynamic ENAT with masquerade option

```
# NAT HTTP traffic going from DMZ.DMZ.HTTP-SERVER (192.168.10.10) to PRIVATE.LAN (192.168.1.0/24)
# with SIPA DMZ.DMZ.ROUTER (49.1.2.3)
rule 30 masq http from dmz.dmz.http-server to private.lan with src dmz.dmz.router
```

Figure 10: Dynamic ENAT—Masquerade
Configuring Static NAT with Proxy ARP

In the following example, an AR-Series firewall is configured with a private zone and a public zone, and a web server is located in the private zone. The public eth1 interface of the firewall is configured with IP address 172.22.0.1/24. Web traffic from a client located on the Internet is routed to a different IP address (172.22.0.3) in order to reach the web server.

Via a port-forwarding NAT rule, traffic is then NATed in order to reach the internal IP address of the web server (172.22.200.3) located in the private zone.

The AR-Series firewall is additionally configured to send proxy-ARP responses to requests to the public IP address (172.22.0.3) of the web server. To restrict the public interface to only sending these proxy-ARP responses for a limited number of specified IP addresses, it uses the `ip limited-local-proxy-arp` command. The IP addresses to which it will respond are specified with the `local-proxy-arp <address>` command.

The proxy-ARP responses use the firewall’s own public interface MAC address (eth1).

Figure 11: Static NAT with proxy-ARP
! Create a private zone for the HTTP server with address 172.22.200.3:
zone private
  network vlan1
    ip subnet 172.22.200.0/24
    host http_server
      ip address 172.22.200.3

! Create a public zone for the HTTP server with address 172.22.0.3:
zone public
  network eth1
    ip subnet 0.0.0.0/0 interface eth1
    host http_server
      ip address 172.22.0.3 <-------- HTTP traffic will be destined for this address

! Create a NAT rule to map from the public zone to the private zone server:
nat
  rule 10 portfwd http from public.eth1 to public.eth1.http_server with dst
do private.vlan1.http_server
  enable

! Configure eth1. It has a different public address than the HTTP server:
interface eth1
  enable the limited local proxy ARP feature:
    ip limited local-proxy-arp
    ip address 172.22.0.1/24

! Configure vlan1:
interface vlan1
  ip address 172.22.200.5/24

! Configure the device to respond to ARPs for the HTTP server public address:
  local-proxy-arp 172.22.0.3/32
**Source-based NAT with Secondary IP Addresses**

In the example below, the link between the AR-Series firewall and the ISP router is using a private IP subnet (192.168.73.0/24). This situation can arise if the ISP does not have enough public IPv4 addresses available that it can allocate to its customers, and has not yet upgraded to an IPv6 network infrastructure.

The ISP has allocated a single public IP address for use by the AR-Series firewall. To achieve this, the ISP's router is configured to route traffic to the single public host IP address 10.0.22.13/32 via the private network address (192.168.73.253) allocated to the WAN address of the AR-Series firewall.

All traffic originating from the AR-Series firewall to the Internet needs to have its source IP address translated to appear to come from the public IP address 10.0.22.13 to be routable via the Internet.

In order to achieve this, the AR-Series firewall is configured with a NAT masquerade rule appended with the `with src` configuration option to translate the source IP address of all traffic egressing the eth1 WAN interface from the private IP address 192.168.73.253, to the public IP address 10.0.22.13.

Without this NAT rule, all traffic would use the private IP address allocated to the WAN interface of the AR-Series firewall. This rule allows traffic to be NATed to an address that is different to the configured WAN interface IP address.

Figure 13: Example: source-based NAT
Figure 14: Configuration for source-based NAT

```plaintext
! zone wan
     network eth1
        ip subnet 192.168.73.253/32
     network eth1-1
        ip subnet 10.0.22.13/32
!
zone internet
     network wan01
        ip subnet 0.0.0.0/0 interface eth1
!
nat
    rule 90 masq any from wan.eth1 to internet.wan01 with src wan.eth1-1 enable
!
interface eth1
    ip address 192.168.73.253/24
    ip address 10.0.22.13/30 secondary
!
ip route 0.0.0.0/0 192.168.73.254
!
```

Configuring Source and Destination NAT (Double NAT)

Source and destination NAT is supported from 5.4.7-0.1.

Source and destination NAT (also known as double NAT) translates both the source address and the destination address of incoming and outgoing connections. To configure it, the usual NAT rules are used with port forwarding and masquerade options.

In this example, the AR-Series firewall is configured with a network in a public zone and a network in a private zone with appropriate subnets and a private zone host to port-forward traffic to.

Incoming traffic to the firewall from the Internet destined for the server has:
- source: 192.0.2.50
- destination: 192.0.2.254 (router public IP address)

With this configuration, NAT rule 10 port-forwards any public-public traffic to private.lan.server. NAT rule 20 matches the port-forwarded traffic and masquerades the source IP address as the AR-Series firewall’s private IP address.

Outgoing traffic from the firewall to the server now has:
- source: 172.16.1.254 (router private IP address)
- destination: 172.16.1.10
Interaction with firewall

NAT **portfw** rules (actions) are applied before any other firewall rules and NAT **masq** rules (actions) are applied after any other firewall rules. When firewall protection is enabled, all traffic is blocked by default. Configure the firewall to allow the same application, source and destination entities that you configure for the NAT rules, using the **rule (Firewall)** command.
Configuring Subnet-based NAT

Subnet-based NAT is supported from 5.4.7-0.1.

Subnet-based NAT translates just the network portion of a packet’s source or destination IP address to a different network address—the host portion of the address is unchanged. There is a one-to-one mapping from addresses in one subnet to the other. Subnet-based NAT allows a user to perform NAT translation on all hosts between two network entities.

Configuring a NAT rule with the netmap option, you can modify the source subnet or destination subnet for a range of addresses, by using the following command:

```
rule [<1-65535>] netmap <application-name> from <source-subnet-entity> to <destination-subnet-entity> with {src|dst} <translated-subnet-entity>
```

For example, subnet-based NAT has been used in a network where all the LANs use the same subnet (192.168.1.0/24). The LAN in each of the premises has a corresponding 172.16.X.0/24 subnet that the device performs subnet-based NAT translation on.

For a two-device topology, the same entity configuration can be used. Firewall-B uses subnet-based NAT to translate the source IP addresses to appear as public.wan2. Firewall-B will change the destination IP addresses from public.wan1 to private.lan. This allows hosts on both 192.168.1.0/24 networks to communicate with remote premises. This example shows configuration to translate addresses for traffic from the client via Firewall-B to Firewall-A to the server.

**Figure 16: Example: subnet-based NAT**

In this example, each firewall has traffic for their 172.16.X.0/24 network routed to them for subnet-based NAT (netmap) translation.

The client (IP address 192.168.1.10) thinks it is connecting to 172.16.1.20. Packets sent by the client have:

- Source 192.168.1.10
- Destination 172.16.1.20
Firewall-B uses subnet-based NAT (netmap option) to translate the source address of this traffic from the 192.168.1.0/24 network to the 172.16.2.0/24 network. The traffic now has:

- Source 172.16.2.10
- Destination 172.16.1.20

Firewall-A uses subnet-based NAT (netmap option) to translate the destination address of this traffic from the 172.16.1.0/24 network to the 192.168.1.0/24 network. The traffic now has:

- Source 172.16.2.10
- Destination 192.168.1.20

The server (IP address 192.168.1.20) receives traffic from 172.16.2.10.

The return path traffic from the server to the client will be reverse-path translated by the connection tracking tables of Firewalls A and B. Bi-directional rules can be created to allow either side to initiate the traffic (see "Bi-directional configuration for subnet NAT" on page 29).

**Figure 17: Example: subnet NAT configuration for Firewall-A**

```plaintext
zone private
network lan
  ip subnet 192.168.1.0/24
!
zone public
network wan1
  ip subnet 172.16.1.0/24
network wan2
  ip subnet 172.16.2.0/24
!
net
rule 10 netmap any from public.wan2 to public.wan1 with dst private.lan enable
```

**Figure 18: Example: subnet-based NAT configuration for Firewall-B**

```plaintext
zone private
network lan
  ip subnet 192.168.1.0/24
!
zone public
network wan1
  ip subnet 172.16.1.0/24
network wan2
  ip subnet 172.16.2.0/24
!
net
rule 10 netmap any from private.lan to public.wan1 with src public.wan2 enable
```
These rules will allow any 192.168.1.X hosts to masquerade as 172.16.2.X hosts when exiting Firewall-B. When traffic to 172.16.1.X arrives at Firewall-A the destination IP address will be changed to 192.168.1.X, allowing both client LANs to use the same local addressing.

Verifying configuration

Source and destination NAT and subnet-based NAT rules and translations can be verified by checking the rule tables and firewall connection tables.

---

**Firewall-B # show nat rule**

```
[* = Rule is not valid - see "show nat rule config-check"]

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>From</th>
<th>With (dst/src)</th>
<th>Entity</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>netmap-src</td>
<td>private.lan</td>
<td>public.wan2</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>any</td>
<td>public.wan1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

---

**Firewall-A # show nat rule**

```
[* = Rule is not valid - see "show nat rule config-check"]

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>From</th>
<th>With (dst/src)</th>
<th>Entity</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>netmap-dst</td>
<td>public.wan2</td>
<td>private.lan</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>any</td>
<td>public.wan1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

---

**Firewall-A # show firewall connections**

```
icmp src=172.16.2.20 dst=172.16.1.10 type=8 code=0 id=2349 packets=5
bytes=420 src=192.168.1.10 dst=172.16.2.20 type=0 code=0 id=2349
packets=5
```
The following two configurations include a second rule to allow bi-directional translation, so that traffic can be initiated from either end.

**Figure 19: Firewall-A configuration for bi-directional subnet NAT**

```
hostname Firewall-A
!
zone private
  network lan
    ip subnet 192.168.1.0/24
!
zone public
  network wan1
    ip subnet 172.16.1.0/24
  network wan2
    ip subnet 172.16.2.0/24
!
nat
  rule 10 netmap any from private.lan to public.wan2 with src public.wan1
  rule 20 netmap any from public.wan2 to public.wan1 with dst private.lan
  enable
!
interface eth1
  ip address 10.0.0.1/24
!
interface eth2
  ip address 192.168.1.254/24
!
ip route 172.16.2.0/24 10.0.0.2
```

**Figure 20: Firewall-B configuration for bi-directional subnet NAT**

```
hostname Firewall-B
!
zone private
  network lan
    ip subnet 192.168.1.0/24
!
zone public
  network wan1
    ip subnet 172.16.1.0/24
  network wan2
    ip subnet 172.16.2.0/24
!
nat
  rule 10 netmap any from private.lan to public.wan2 with src public.wan2
  rule 20 netmap any from public.wan2 to public.wan1 with dst private.lan
  enable
!
interface eth1
  ip address 10.0.0.2/24
!
interface eth2
  ip address 192.168.1.254/24
!
ip route 172.16.1.0/24 10.0.0.1
```
Allowing Partial Sessions through a Firewall

Firewall no-state-enforcement rules are supported from 5.4.7-0.1.

The no-state-enforcement rules illustrated by this example should only be used when asymmetric routing design causes the firewall to only see partial sessions, so that the firewall may otherwise block required traffic. When the firewall detects an out-of-sequence session, it permits the session from that point onwards.

This option only applies to firewall permit rules, and cannot be used with NAT rules.

**Stateful inspection**
During normal AR-Series firewall operation, application-based rules are used to identify the first packet in a connection, to permit matching connections to proceed and to deny other connections. Stateful inspection is used to permit packets for an already permitted connection to pass through the firewall. Packets are denied if they do not match a permit rule (that is, if they do not matching the application, to and from addresses and interfaces) or do not match an existing connection.

**Problem**
However, in some networks there may be a firewall that does not ‘see’ all the traffic in a connection. In this example, an enterprise network has multiple offices connected via multiple private VPN links. Traffic from office A to office B is routed via office C but traffic from office B to office A is routed via office D. Firewalls at C and D are also configured to secure office traffic and access to the Internet. Stateful inspection does not allow the firewalls at C and D to permit traffic transiting between offices A and B because they only ever see part of the connection traffic.

**Solution**
The best solution for such a network is often to resolve the routing issues by changing the network topology to ensure the firewall can see and track sessions in their entirety to apply full stateful inspection. For cases where this is not possible, this example maintains the routing configuration and effectively disables stateful inspection for traffic matching particular firewall rules. A firewall rule is configured with a no-state-enforcement option to permit traffic from the connection source to the connection destination.

**Note:** This feature applies to firewall permit rules only. It cannot be applied via NAT rules, as NAT requires full stateful tracking of the entire session in order to maintain network address and port translations for data flows.
Figure 21: Example: partial sessions through firewall

![Diagram of network setup](image)

Figure 22: Example: partial sessions through firewall—configuration for Firewall C

```plaintext
zone Transit
network 2
  ip subnet 10.0.0.0/8 interface vlan2
network 4
  ip subnet 10.0.0.0/8 interface vlan4
!
firewall
rule 10 permit any from Transit to Transit no-state-enforcement
protect
!
interface vlan2
  ip address 10.0.2.2/24
!
interface vlan4
  ip address 10.0.4.2/24
!
ip route 10.0.0.0/8 10.0.4.1
```

Interaction with firewall | Page 31
How it works
The following steps show the process of permitting and establishing the TCP connection between Host A at Office A and Server B at Office B.

1. Host A at Office A requests an HTTP URL from Server B at Office B.
2. Host A sends a TCP SYN from 10.1.1.100:1024 to 10.4.1.100:80.
3. Firewall A forwards the SYN to Firewall C.
4. Firewall C matches this TCP SYN to rule 10 ("permit any from transit to transit").
5. Firewall C forwards the packet to Firewall B which routes it to Server B.
7. Firewall B forwards the SYN/ACK to Firewall D.
8. Firewall D matches this SYN/ACK packet to its rule 10, due to the no-state-enforcement option.
9. Firewall D forwards the SYN/ACK to Firewall B which forwards it to Host A.
10. Host A sends the ACK and HTTP request to Server B.
11. Firewall C counts this as a rule 10 match due to the no-state-enforcement option.
13. Firewall D permits this as a connection match for the traffic flow that was permitted by rule 10 in step 8.

```plaintext
zone Transit
    network 3
        ip subnet 10.0.0.0/8 interface vlan3
    network 5
        ip subnet 10.0.0.0/8 interface vlan5

firewall
    rule 10 permit any from Transit to Transit no-state-enforcement

interface vlan3
    ip address 10.0.3.2/24

interface vlan5
    ip address 10.0.5.2/24

ip route 10.0.0.0/8 10.0.3.1
```
The firewall rule used to permit half-completed sessions supports the following:

- The syntax uses the `no-state-enforcement` option:
  
  ```
  rule [1-65535] permit <application> from <entity-1> to <entity-2> no-state-enforcement [log]
  ```

- Only the `permit` action is supported with no-state-enforcement rules.

- Rules are configured to permit traffic from the connection source to the connection destination.

- The `log` option can be configured with the `no-state-enforcement` option.

- Deep Packet Inspection (DPI) applications are not supported for `no-state-enforcement` rules.

- Other applications (not DPI) and entities can be specified as in other firewall rules.

- However, this rule is expected to be used to permit all traffic between interfaces on the firewall regardless of the state.

This configuration extract illustrates these points:

```plaintext
zone Transit
  network 3
    ip subnet 10.0.0.0/8 interface vlan3
  network 5
    ip subnet 10.0.0.0/8 interface vlan5
!
firewall
  rule 10 permit any from Transit to Transit no-state-enforcement
```

The output displayed by the following commands on each of the devices ....

```
Firewall-C#show firewall rule

[* = Rule is not valid - see "show firewall rule config-check"]

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>App</th>
<th>From</th>
<th>To</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>permit</td>
<td>any</td>
<td>Transit</td>
<td>Transit</td>
<td>10</td>
</tr>
</tbody>
</table>
```

```
Firewall-C#show firewall connections
tcp SYN_SENT src=10.1.1.100 dst=10.4.1.100 sport=48348 dport=80
packets=1 bytes=60 [UNREPLIED] src=10.4.1.100 dst=10.1.1.100 sport=80
dport=48348 packets=0 bytes=0
```

Note that the `show firewall rule` output displays more than one rule hit for every connection, where a normal connection-based rule would show 1 hit per connection.
For the return traffic, DUT-2 shows:

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>App</th>
<th>From</th>
<th>To</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>permit</td>
<td>any</td>
<td>Transit</td>
<td>Transit</td>
<td>4</td>
</tr>
</tbody>
</table>
```

```
tag CLOSE_WAIT src=10.4.1.100 dst=10.1.1.100 sport=80 dport=48348 packets=4 bytes=844 [UNREPLIED] src=10.1.1.100 dst=10.4.1.100 sport=48348 dport=80 packets=0 bytes=0
```