Technical Guide

Firewall and Network Address Translation

Feature Overview and Configuration Guide

Introduction

This guide explains the firewall and NAT features of AlliedWare Plus UTM firewalls and secure VPN routers, and provides instructions for their configuration.

AlliedWare Plus firewalls offers security, flexibility, and ease of use. Unlike traditional firewalls, these adapt to the rapid changes in Internet-based applications, allowing enterprises to leverage web-based technology without facing costly security risks.

AlliedWare Plus firewalls also support 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 AlliedWare Plus firewalls, followed by examples illustrating how to configure them in various network situations.

Contents

Products and software version that apply to this guide

This Guide applies to all UTM firewalls and VPN routers that run the AlliedWare Plus OS.

Most features described in this document are supported from AlliedWare Plus 5.4.5 or later. These features apply from later releases:

- 5.5.2-1.1 or later: Tab completion for entity and application names
- 5.4.8-0.x or later: New firewall rules are needed when DPI is enabled and the firewall is accessing external services, including Update Manager.
- 5.4.7-2.4 or later: Configurable HTTP and HTTPS ports
- 5.4.7-1.x or later: Firewall connection logging
- 5.4.7-1.x or later: Configurable TCP established session timeout
- 5.4.7-0.1 or later: Subnet-based NAT
- 5.4.7-0.1 or later: Source and destination NAT
- 5.4.7-0.1 or later: Allowing partial sessions through a firewall (no state enforcement)
- 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:

- 5G Mobile UTM Firewall GUI Feature Overview and Configuration Guide
- Getting Started with the Device GUI on UTM Firewalls
- Getting Started with the Device GUI on VPN Routers
- Application Awareness Feature Overview and Configuration Guide
- Advanced Network Protection 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 and Web Control. The **Advanced Threat Protection** (ATP) license includes IP Reputation, and Advanced IPS.

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.

AlliedWare Plus 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, AlliedWare Plus 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.

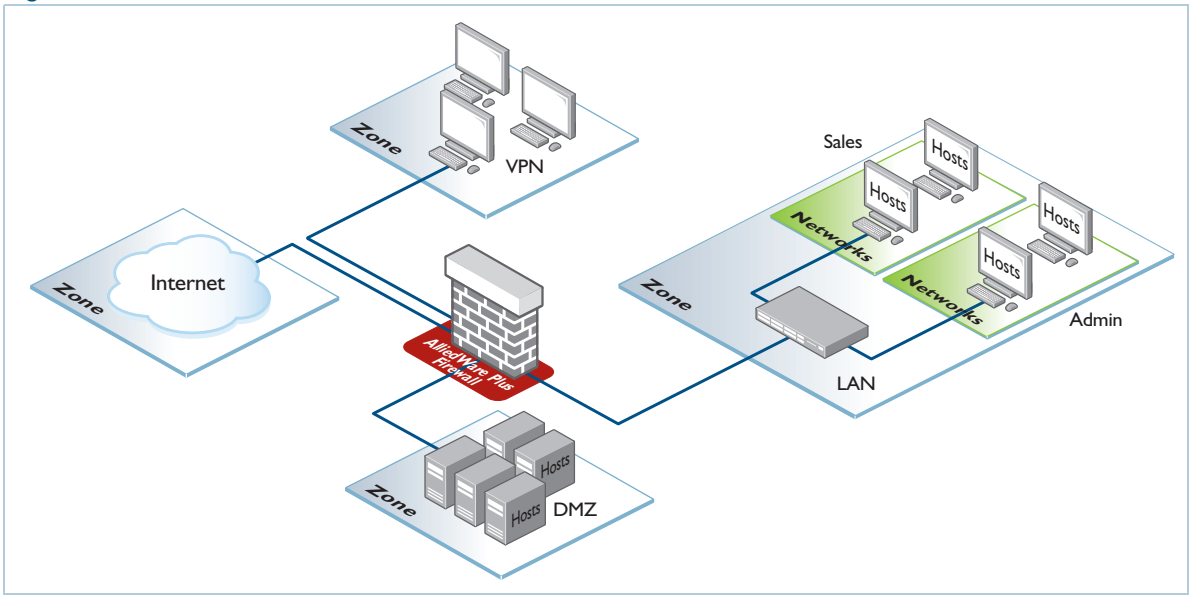
The AlliedWare Plus firewalls view the physical network in terms of zones, networks and hosts. Firewall rules can be applied to any level of this hierarchy, as shown in Figure 1 on page 6. See "Entities" on page 9 for entity definitions and usage.

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.

As data enters the firewall, it is first identified by the DPI application decoding engine. 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. By default, this engine contains a library of a few dozen common Internet-based applications that it is capable of identifying. 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 8 for application definition and usage.

Figure 1: Firewall zones, networks, hosts



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

The firewall GUI enables you to set up the firewall, configure entities such as zones, networks, and hosts, and create firewall, NAT, and traffic-control rules to manage traffic between them. It also includes features like Intrusion Prevention System (IPS) and URL Filtering, which help protect the network and control 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 firewall GUI

Your AlliedWare Plus firewall comes 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***.

HTTP and HTTPS GUI listen ports

By default, the firewall GUI uses the HTTP server listen port 80. The default HTTPS server listen port is 443. You can change the HTTPS port. From AlliedWare Plus version 5.4.7-2.4 you can also change the HTTP port, and disable listening on either the HTTP or HTTPS port.

This allows you to remap the GUI to use other ports and allow traffic using these HTTP (80) and HTTPS (443) ports to be forwarded through the device to another server, if required, instead of being terminated on the device. You may wish to change the HTTP port if port 80 needs to be used by another service at the same IP address in your network.

To change or disable the HTTP listen port, use the command:

```
awplus(config)# http port {<1-65535>|none}
```

To restore the HTTP port to its default (port 80), use the command:

```
awplus(config)# no http port <1-65535>
```

To change or disable the HTTPS listen port, use the command:

```
awplus(config)# http secure-port {<1-65535>|none}
```

Setting the port to none disables HTTP or HTTPS management.

Note that changing or disabling the HTTPS trusted port is not supported when using Vista Manager EX. If you are using Vista Manager EX and need to change the HTTPS trusted port, you must use

certificate-based authorization in Vista Manager EX. See the 'Vista Manager EX Installation and User Guide' for instructions.

To restore the HTTPS port to its default (port 443), use the command:

```
awplus(config)# no http secure-port
```

To check the settings for the HTTP and HTTPS (secure) ports, use the command:

```
awplus# show http
```

Applications

An application serves as a high-level abstraction to categorize packets within network traffic. Achieving traffic matching for applications involves various techniques, such as matching packets based on port numbers or identifying application signatures in packet flows. The device is capable of recognizing different types of applications, allowing configuration of source port, destination port, protocol, ICMP code, and ICMP type for each application.

For an application to be considered valid, its protocol, source, and destination must be appropriately configured. For instance, an application is invalid if it lacks a configured protocol, or if source and destination ports are assigned to protocols other than TCP, UDP, or SCTP.

There is an built-in library of many more applications that can be identified in traffic if Deep Packet Inspection (DPI) is enabled.

The extensive up-to-date library of applications maintained by Procera is available by subscription. When DPI is enabled, the device recognizes these applications.

You can use the **show application** and **show application detail** commands to display the detail of these applications.

If applications have the same name, precedence in all application-aware features is:

1. user-configured applications
2. applications identified by DPI
3. built-in predefined list

For information about applications and application awareness, see the Application Awareness Feature Overview and Configuration Guide.

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 (using the **application** and associated commands, described in step 4, "Configuring firewall and NAT rules for entities" on page 34.)

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

Allied Telesis UTM Firewalls and Secure VPN Routers 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, AlliedWare Plus 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)

GeoIP subnets for network entities

Overview

AlliedWare Plus version 5.5.4-2.1 onwards, supports the GeoIP (Geographic IP) feature.

The GeoIP feature is a network security and traffic management tool. It uses IP address geolocation to identify the approximate geographic location of clients or servers by mapping IP address blocks to specific countries or regions.

GeoIP is widely used for purposes such as security, access control, and traffic optimization. For instance, it can restrict access to a service to only those with source IP addresses from a specific region. Similarly, it can block all traffic originating from regions known for frequent malicious or nuisance network activities.

The feature enables you to dynamically add all networks assigned to a specific country to an entity with a single command. These networks are sourced from a third-party provider and updated automatically on a regular basis.

GeoIP is a best-effort service designed to conveniently limit or block traffic based on expected sources or destinations. However, it does not guarantee that all traffic associated with these

locations will be detected, nor does it prevent parties from obscuring their true location. You are encouraged to utilize additional features, such as Advanced IPS, IP Reputation, and Web Categorization, to enhance protection against malicious activity.

Configuration

The only requirements for this feature are that you have access to the Internet in order to obtain the GeoIP database. Multiple networks may be configured with GeoIP matches, and multiple matches per zone if required.

For example to add a zone with Fiji and both American Samoa and Western Samoa under the same grouping:

```
zone pacific
    network fiji
      ip subnet dynamic geoip FJ
    network samoa
      ip subnet dynamic geoip AS
      ip subnet dynamic geoip WS
```

This feature uses a resource containing the Geo IP data that is downloaded from the Update Server, as used by various UTM features. The firewall (if enabled) will need to be configured to allow update requests to be sent to the Update Server. This configuration is the same as for other features that use the Update Server.

Monitoring

Use the command **show resource** to show when the GeoIP database was last updated, and when it will next be updated. The command **show entity** will display the current set of networks for each Entity.

Dynamic addresses for host entities - FQDN lookup

Overview

In AlliedWare Plus, the Firewall, PBR, and Traffic Control rules (collectively known as UTM feature rules) rely on applications to match ingress traffic. These applications can be built-in or customized but can only match based on protocol, ports, DSCP value, and ICMP type/code. Alternatively, a DPI engine can identify applications, but it is limited to known applications, which may not include obscure or region-specific services.

Due to these limitations, it can be challenging to apply different routing policies to specific Internet services, as many web services use common ports and may not be reliably recognized by supported DPI providers. While fixed IP address configurations can be used as a workaround, maintaining an up-to-date list of IPs for dynamic Internet services is burdensome.

To address these challenges, from software version **5.4.8-1** onwards, FQDN lookup for entities is introduced as a new method to match traffic for UTM features, reducing configuration complexity and improving flexibility.

FQDN lookup for entities matches traffic in UTM features. It allows an entity to dynamically maintain a list of IP addresses updated via DNS. You can create firewall entities that specify FQDNs, and the device copies matching IP addresses (from A and AAAA records in its DNS cache) into the entity's IP address list. A records are for IPv4 addresses, and AAAA records are for IPv6 addresses.

This ensures that the IP addresses associated with a specific Internet service are always as up-to-date as the DNS records for that service.

The router needs to be configured with the DNS relay feature, so that all DNS requests sent by clients within the network are intercepted by the router itself.

When an FQDN is configured and a client make a DNS request for that FQDN, the router will copy the IP address(es) learned into the firewall entity IP address lists. Domains that are added into the cache can then be seen in the output of **show ip dns forwarding cache** and added into firewall entities as appropriate.

Configuration

In simple terms, to use this feature you need to:

1. Configure DNS Relay
2. Add FQDNs to Entities
3. Add firewall rules

Here’s a basic DNS Relay configuration:

```
ip name-server 192.168.1.1
ip domain-lookup via-relay
ip dns forwarding
ip dns forwarding cache size 1000 timeout 1800
```

Configuration examples

Table 1: FQDN Lookup, without any Firewall rules:

```
!
service password-encryption
!
hostname AR4050
!
no banner motd
!
username manager privilege 15 password 8 $1$bJoVec4D$JwOJGPr7YqoExA0GVasdE0
!
no service ssh
!
service telnet
!
no service http
!
no clock timezone
!
snmp-server
!
aaa authentication enable default local
aaa authentication login default local
!
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network fqdn
  host facebook
   ip address dynamic fqdn facebook.com
!
ip name-server 192.168.1.1
ip domain-lookup via-relay
!
no service dhcp-server
!
no ip multicast-routing
!
spanning-tree mode rstp
!
lacp global-passive-mode enable
no spanning-tree rstp enable
!
interface port1.0.1-1.0.8
 switchport
 switchport mode access
!
interface vlan1
 ip address 192.168.1.2/24
!
ip route 0.0.0.0/0 vlan1
!
ip dns forwarding
ip dns forwarding cache size 10000 timeout 1800
!
line con 0
line vty 0 4
!
end
```

Table 2: FQDN lookup with firewall and NAT configured

```
service password-encryption
!
hostname AR4050S
!
no banner motd
!
username manager privilege 15 password 8 $1$bJoVec4D$JwOJGPr7YqoExA0GVasdE0
!
!
no service ssh
!
no service telnet
!
no service http
!
no clock timezone
!
snmp-server
!
!
aaa authentication enable default local
aaa authentication login default local
!
!
zone private
 network lan
  ip subnet 192.168.1.0/24
  host user1
   ip address dynamic fqdn user1.test.com
  host user2
   ip address dynamic fqdn user2.test.com
!
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network eth1
  host eth1
   ip address dynamic interface eth1
!
zone public_2
 network all
  ip subnet 0.0.0.0/0 interface eth2
  ip subnet 192.168.2.0/24
  host eth2
!
firewall
 rule 20 deny any from private.lan.user2 to public_1.all <--- Deny user2 internet
access
 rule 30 permit any from private.lan.user1 to public_1.all <--- Permit user1
internet access
 rule 40 permit any from private.lan to private.lan
 rule 20 permit any from public_2.all to public_2.all
 protect
!
nat
 rule 10 masq any from private.lan to public_1.all
 enable
```

FQDN lookup with firewall and NAT configured (continued)

```
!
!
ip name-server 10.0.0.1
ip domain-lookup via-relay
!
ip dhcp pool test
 network 192.168.1.0 255.255.255.0
 range 192.168.1.2 192.168.1.10
 dns-server 192.168.1.1
 route 0.0.0.0/0 192.168.1.1 rfc3442
!
!
!
service dhcp-server
!
no ip multicast-routing
!
spanning-tree mode rstp
!
lacp global-passive-mode enable
no spanning-tree rstp enable
!
vlan database
 vlan 2 state enable
!
interface port1.0.1
 shutdown
 switchport
 switchport mode access
!
interface port1.0.2
 switchport
 switchport mode access
!
interface port1.0.3
 switchport
 switchport mode access
 switchport access vlan 2
!
interface port1.0.4-1.0.8
 switchport
 switchport mode access
!
interface eth1
 ip address 172.16.0.2/24
!
interface eth2
 ip address 192.168.2.1/24
!
interface vlan1
 ip address 192.168.1.1/24
!
interface vlan2
```

FQDN lookup with firewall and NAT configured (continued)

```
ip address 10.0.0.2/24
!
ip route 0.0.0.0/0 192.168.2.2 10
ip route 0.0.0.0/0 172.16.0.1
!
ip dns forwarding
ip dns forwarding timeout 600
ip dns forwarding cache size 1000 timeout 600
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
end
```

Table 3: FQDN lookup with PBR (no linkmon)

```
service password-encryption
!
hostname AR4050S
!
no banner motd
!
username manager privilege 15 password 8 $1$bJoVec4D$JwOJGPr7YqoExA0GVasdE0
!
!
no service ssh
!
no service telnet
!
no service http
!
no clock timezone
!
snmp-server
!
!
aaa authentication enable default local
aaa authentication login default local
!
!
zone private
 network lan1
  ip subnet 192.168.1.0/24
  host user1
   ip address dynamic fqdn user1.test.com
  host user2
ip address dynamic fqdn user2.test.com
 network lan2
  ip subnet 10.0.0.0/24
!
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network eth1
  host eth1
   ip address dynamic interface eth1
```

FQDN lookup with PBR (no linkmon) (continued)

```
!
zone public_2
 network all
  ip subnet 0.0.0.0/0 interface eth2
  ip subnet 192.168.2.0/24
  host eth2
!
firewall
 rule 10 permit any from private.lan1 to public_1.all
 rule 50 permit any from private.lan2 to private.lan2
 rule 60 permit any from private.lan1 to private.lan1
 rule 70 permit any from public_1.all to public_1.all
 rule 80 permit any from public_2.all to public_2.all
 protect
!
nat
 rule 10 masq any from private.lan1 to public_1.all
 enable
!
!
!
!
policy-based-routing
 ip policy-route 10 match http from private.lan1.user1 nexthop 172.16.0.1   <---
Web traffic from user1 is routed out eth1
 ip policy-route 20 match http from private.lan1.user2 nexthop 192.168.2.2  <---
Web traffic from user2 is routed out eth2
 policy-based-routing enable
!
ip name-server 10.0.0.1
ip domain-lookup via-relay
!
ip dhcp pool test
 network 192.168.1.0 255.255.255.0
 range 192.168.1.2 192.168.1.10
 dns-server 192.168.1.1
 route 0.0.0.0/0 192.168.1.1 rfc3442
service dhcp-server
!
no ip multicast-routing
!
spanning-tree mode rstp
!
lacp global-passive-mode enable
no spanning-tree rstp enable
!
vlan database
 vlan 2 state enable
!
interface port1.0.1
 shutdown
 switchport
 switchport mode access
!
interface port1.0.2
 switchport
 switchport mode access
```

FQDN lookup with PBR (no linkmon) (continued)

```
!
interface port1.0.3
 switchport
 switchport mode access
 switchport access vlan 2
!
interface port1.0.4-1.0.8
 switchport
 switchport mode access
!
interface eth1
 ip address 172.16.0.2/24
!
interface eth2
 ip address 192.168.2.1/24
!
interface vlan1
 ip address 192.168.1.1/24
!
interface vlan2
 ip address 10.0.0.2/24
!
ip route 0.0.0.0/0 192.168.2.2 10
ip route 0.0.0.0/0 172.16.0.1
!
ip dns forwarding
ip dns forwarding timeout 600
ip dns forwarding cache size 1000 timeout 600
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
end
```

Table 4: FQDN lookup with PBR using Linkmon without load balancing

```
service password-encryption
!
hostname AR4050S
!
no banner motd
!
username manager privilege 15 password 8 $1$bJoVec4D$JwOJGPr7YqoExA0GVasdE0
!
!
no service ssh
!
no service telnet
!
no service http
!
no clock timezone
!
snmp-server
!
!
aaa authentication enable default local
aaa authentication login default local
```

FQDN lookup with PBR using Linkmon without load balancing (continued)

```
!
!
zone private
 network lan1
  ip subnet 192.168.1.0/24
  host user1
   ip address dynamic fqdn user1.test.com
  host user2
   ip address dynamic fqdn user2.test.com
 network lan2
  ip subnet 10.0.0.0/24
!
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network eth1
  host eth1
   ip address dynamic interface eth1
!
zone public_2
 network all
  ip subnet 0.0.0.0/0 interface eth2
  ip subnet 192.168.2.0/24
  host eth2
   ip address dynamic interface eth1

!
firewall
 rule 10 permit any from private.lan1 to public_1.all
 rule 50 permit any from private.lan2 to private.lan2
 rule 60 permit any from private.lan1 to private.lan1
 rule 70 permit any from public_1.all to public_1.all
 rule 80 permit any from public_2.all to public_2.all
 protect
!
nat
 rule 10 masq any from private.lan1 to public_1.all
 enable
!
!
!
!
policy-based-routing
 ip policy-route 10 match http linkmon-group internet_access linkmon-profile
internet_profile   <--- PBR is configured to send any web traffic to the FQDN of
wan1 or wan2 depending on the state of the link
 policy-based-routing enable
!
linkmon probe name internet_backup type icmp-ping
 destination wan2.test.com
 interval 1000
 enable
!
linkmon probe name internet_main type icmp-ping
 destination wan1.test.com
 interval 1000
 enable
!
linkmon group internet_access
 member 1 destination 172.16.0.1 probe internet_main
 member 2 destination 192.168.2.2 probe internet_backup
```

FQDN lookup with PBR using Linkmon without load balancing (continued)

```
!
ip name-server 10.0.0.1
ip domain-lookup via-relay
!
ip dhcp pool test
 network 192.168.1.0 255.255.255.0
 range 192.168.1.2 192.168.1.10
 dns-server 192.168.1.1
 route 0.0.0.0/0 192.168.1.1 rfc3442
!
!
!
service dhcp-server
!
no ip multicast-routing
!
spanning-tree mode rstp
!
lacp global-passive-mode enable
no spanning-tree rstp enable
!
vlan database
 vlan 2 state enable
!
interface port1.0.1
 shutdown
 switchport
 switchport mode access
!
interface port1.0.2
 switchport
 switchport mode access
!
interface port1.0.3
 switchport
 switchport mode access
 switchport access vlan 2
!
interface port1.0.4-1.0.8
 switchport
 switchport mode access
!
interface eth1
 ip address 10.0.0.2/24
!
interface eth2
 ip address 192.168.2.1/24
!
interface vlan1
 ip address 192.168.1.1/24
!
interface vlan2
 ip address 10.0.0.2/24
!
ip route 0.0.0.0/0 192.168.2.2 10
ip route 0.0.0.0/0 172.16.0.1
!
ip dns forwarding
ip dns forwarding timeout 600
ip dns forwarding cache size 1000 timeout 600
```

FQDN lookup with PBR using Linkmon without load balancing (continued)

```
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
end
```

Table 5: FQDN lookup with traffic control

```
service password-encryption
!
hostname AR4050S
!
no banner motd
!
username manager privilege 15 pa
ssword 8 $1$bJoVec4D$JwOJGPr7YqoExA0GVasdE0
!
!
no service ssh
!
no service telnet
!
no service http
!
no clock timezone
!
snmp-server
!
!
aaa authentication enable default local
aaa authentication login default local
!
!
zone private
 network lan1
  ip subnet 192.168.1.0/24
  host user1
   ip address dynamic fqdn user1.test.com
  host user2
   ip address dynamic fqdn user2.test.com
 network lan2
  ip subnet 10.0.0.0/24
!
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network eth1
  host eth1
   ip address dynamic interface eth1
!
zone public_2
 network all
  ip subnet 0.0.0.0/0 interface eth2
  ip subnet 192.168.2.0/24
  host eth2
   ip address dynamic interface eth2
```

FQDN lookup with traffic control (continued)

```
!
firewall
 rule 10 permit any from private.lan1 to public_1.all
 rule 50 permit any from private.lan2 to private.lan2
 rule 60 permit any from private.lan1 to private.lan1
 rule 70 permit any from public_1.all to public_1.all
 rule 80 permit any from public_2.all to public_2.all
 protect
!
nat
 rule 10 masq any from private.lan1 to public_1.all
 enable
!
!
!
traffic-control
 policy TEST priority
  class HIGH priority-level 15
  class LOW priority-level 5
 rule 10 match http from private.lan1.user1 to public_1.all policy TEST.HIGH<---Web
traffic from user1 is assigned to the policy TEST.HIGH
 rule 20 match http from private.lan1.user2 to public_2.all policy TEST.LOW <---Web
traffic from user2 is assigned to the policy TEST.LOW
 interface eth1 virtual-bandwidth 10mbit
 traffic-control enable
!
!
policy-based-routing
 ip policy-route 10 match http linkmon-group internet_access linkmon-profile
internet_profile
 policy-based-routing enable
!
!
ip name-server 10.0.0.1
ip domain-lookup via-relay
!
ip dhcp pool test
 network 192.168.1.0 255.255.255.0
 range 192.168.1.2 192.168.1.10
 dns-server 192.168.1.1
 route 0.0.0.0/0 192.168.1.1 rfc3442
!
!
!
service dhcp-server
!
no ip multicast-routing
!
spanning-tree mode rstp
!
lacp global-passive-mode enable
no spanning-tree rstp enable
!
vlan database
 vlan 2 state enable
!
```

FQDN lookup with traffic control (continued)

```
interface port1.0.1
 shutdown
 switchport
 switchport mode access
!
interface port1.0.2
 switchport
 switchport mode access
!
interface port1.0.3
 switchport
 switchport mode access
 switchport access vlan 2
!
interface port1.0.4-1.0.8
 switchport
 switchport mode access
!
interface eth1
 speed 10
 ip address 172.16.0.0.2/24
!
interface eth2
 ip address 192.168.2.1/24
!
interface vlan1
 ip address 192.168.1.1/24
!
interface vlan2
 ip address 10.0.0.2/24
!
ip route 0.0.0.0/0 192.168.2.2 10
ip route 0.0.0.0/0 172.16.0.1
!
ip dns forwarding
ip dns forwarding timeout 600
ip dns forwarding cache size 1000 timeout 600
!
line con 0
 exec-timeout 0 0
line vty 0 4
!
end
```

Monitoring

To see the currently configured FQDN entities, use the command: **show running-config entity**

```
AR4050#show running-config entity
zone public_1
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network fqdn
  host facebook
   ip address dynamic fqdn facebook.com
```

To see resolved IP addresses, use the command: **show entity**

```
AR4050#show entity
Zone:          public_1
 Network:      public_1.all
  Subnet:      0.0.0.0/0 via eth1
 Network:      public_1.fqdn
  Host:        public_1.fqdn.facebook
   FQDN IPv4:  facebook.com
   Address:    157.240.8.35 (dynamic)
```

To see the current DNS cache, use the command: s**how ip dns forwarding cache**

```
AR4050#show ip dns forwarding cache
IPv4 addresses in cache:     1
IPv6 addresses in cache:     0
Cache size: 10000
Host
 Address                                                 Expires Flags
facebook.com
 157.240.8.35                                                101
```

Troubleshooting notes

1. Services using a **CDN** (Content Delivery Network) often share IP addresses with other related or unrelated services, which can create challenges for network administrators. If one service needs to be allowed and another blocked, sharing the same IP address can cause issues—either both are permitted or both are blocked based on rule order.

   Examples include Office365 applications (e.g., Excel365 and Word365) sharing IPs and dissimilar services like Google Search (www.google.com) and You Tube (www.youtube.com). For instance, an administrator may want to block You Tube but allow Google Search, but shared IP addresses make this difficult to enforce.

2. Some services, particularly those hosted by CDNs, may start with a connection to a single FQDN but require additional requests to new IP addresses not listed under the original FQDN in DNS. This can cause issues if PBR rules are set to route all traffic for the service through a specific interface, as subsequent traffic to these new IPs may be routed incorrectly, leading to communication errors.

   Administrators can identify missing FQDNs by checking the output of **show ip dns forwarding cache** and updating their configurations accordingly. However, since domains change over time, they must regularly refresh and monitor the list of FQDNs. Office365 is a common example of a service that exhibits this behavior.

3. Firewall rules must be configured to prevent clients from using external DNS servers. This ensures that client devices use the router as their DNS server, so future requests to the service's IP addresses consistently resolve to the same IP addresses that the router has resolved for that service.

4. Some services may have IP addresses registered in DNS, but upon accessing that IP address, the client is sent a redirect to a secondary IP address where all communication then proceeds with. Because the secondary IP address was not included in the DNS reply, the firewall entity will not be able to correctly match traffic sent to that address.

5. Because the device relies on DNS queries in order to populate the list of IP addresses to match traffic against, any Internet resources that are accessed directly by IP address from network clients and therefore don't generate DNS requests, won't be able to be matched by the device. A workaround is to manually configure explicit PBR rules to match against those IP addresses.
6. Any DNS requests that are not sent using standard unencrypted DNS queries to port 53, cannot be intercepted by the device and so traffic destined to FQDNs that have been resolved via these protocols won't be able to be matched by the device as it will have no record of the IP addresses used by the domain. These alternative DNS protocols do not yet have widespread adoption, but are under active development at present, e.g. DNS over HTTPS, DNS over TLS or DNSCrypt.
7. IP addresses matched by FQDN entities are only updated when the DNS records are updated by DNS request. Expired DNS records still exist in the DNS cache but are not displayed. When a DNS request is made, the DNS cache is traversed for expired entries and deleted then. For example, a user performs DNS requests for "facebook.com" and "google.com" so these are added to the DNS cache. After some time these will expire and they are not deleted from the DNS cache yet. When a DNS request is made for "google.com", the record for "google.com" is updated and the record for "facebook.com" is now deleted.

Default flow with firewall enabled

The following section describes the default behaviors for various Layer 2 and Layer 3 data-plane and control-plane protocols.

If the firewall is enabled by the **protect** command, a default deny policy drops all traffic that does not match configured rules that is being processed via the firewall software.

L3 data plane

All Layer 3 IP data plane messages are subject to firewall screening, so if the firewall is enabled and there are no firewall permit rules explicitly configured to allow the associated applications, Layer 3 data plane messages are dropped. This includes protocols like ICMP, general customer TCP, UDP and multicast network traffic.

L3 control plane

By default, the firewall blocks both reception and transmission of Layer 3 control plane messages (L3CP) if corresponding firewall rules are not configured.This also include remote management protocols like SSH and Telnet.

For IPv6 and IPv4 routing protocols to operate to allow transmission and reception via the reserved multicast address range, corresponding firewall permit rules must be configured. This applies to routing protocols such as PIM, BGP, OSPF, OSPFv3, RIP and RIPng.

L2 control plane

However, Layer 2 control plane (L2CP) protocols (embedded within Ethernet frames) are processed shortly after ingress, **before** being processed by the firewall, so associated firewall permit rules are not required.

These L2CP frames include all of IEEE Std 802.1D and IEEE Std 802.1Q Reserved Addresses used by LACP, STP, LLDP, and 802.1x MAC control. To remain IEEE802.1 compliant, they do not typically ingress one interface and egress another. These protocols operate independently of the firewall—they continue to be transmitted and received whether or not the firewall is enabled unless the Layer

2 feature is explicitly disabled in the device configuration. The individual features themselves are, however, designed to detect and drop malformed control plane messages which might be used to form some kind of DOS attack, so there is still some inherent protection provided.

AMF messages

AMF messages are a special case, and the firewall treats them as Layer 2 control plane messages processed independently of the firewall. AMF virtual link messages are however transported as IP UDP packets, so corresponding allow rules must be configured.

IPv4 ARP and IPv6 ND

With the firewall enabled, IPv4 ARP and IPv6 RA, RS, NA, and NS are all permitted without the need to configure firewall rules.

VRRP messages

From AlliedWare 5.4.6-2.x, VRRP behavior is under firewall control. High Availability uses VRRP. If High Availability is used, then firewall permit rules must be configured to allow VRRP multicast messages to be received. VRRP messages are still transmitted from the firewall without a corresponding firewall permit rule, but the incoming VRRP control plane IPv4/IPv6 messages will be blocked by the firewall before being processed by VRRP feature.

In AlliedWare Plus versions 5.4.6-1.x and earlier, incoming and outgoing VRRP messages bypass the firewall, so no corresponding firewall permit rules are required.

L2 bridged traffic

Additionally, Layer2 bridged Ethernet frames (bridged from one interface to another) are also not subject to the firewall application rules, so they will continue to flow unimpeded if the firewall is enabled. The bridge itself does, however, inspect the embedded IP data fields contained in the Layer 2 Ethernet frames, and so will also drop malformed packets if the encapsulated data is corrupt.

Firewall filtering and logging

This section describes the filtering and logging performed when the firewall feature on an AlliedWare Plus firewall is enabled. A firewall rule specifies the action (Table 6) to take for traffic that matches other parameters in the rule.

Table 6: Firewall actions and log message dispositions

ACTION	MEANING
Permit	The matched packets are permitted to egress from the firewall.
Deny	The matched packets are silently dropped by the firewall. No explicit notification is sent to the source of the packets
Reject	The matched packets are rejected by the firewall and an attempt is made to cleanly close the connection. The source of the packets is notified where possible, for instance, a TCP RST packet is returned for a TCP session, or ICMP packets such as destination/port unreachable are sent to the source.
Log	The matched packets are logged, and will continue to be processed by subsequent firewall rules, which may eventually permit, deny or reject the packets.

Default filtering behaviour

When enabled, the firewall has some default attack protection and filtering rules installed that are not configurable by the user. When packets are dropped by these default filters, log messages are generated to record the reason for the drop. In order to prevent the device from being overloaded by generating log messages in response to an attack, the generation of logs is rate-limited, depending on the reason for the packet being dropped.

Smurf attack protection

The firewall has smurf attack protection enabled by default, and it cannot be disabled. A smurf attack is an ICMP ping that is sent with a broadcast IP address as the destination IP address. The firewall will silently discard all pings that are directed at the broadcast address and will not log the packet.

Invalid TCP flags

The firewall, when enabled, protects against TCP packets with illegal flag combinations set. When dropping these illegal packets, the firewall will generate at most one log message per second regardless of the number of packets dropped by the rule. The logs generated for these illegal TCP flag combinations will begin with the prefix:

```
Firewall: DENY probe <illegal-flags>
```

followed by the packet data.

The firewall will also drop new TCP connections that have not been properly started with a SYN flag set. The prefix for these log messages is

```
Firewall: DENY no SYN
```

with a maximum logging rate of one per second.

Connection tracking of permitted packets

The firewall performs stateful packet inspection as part of its general filtering process. TCP, UDP or ICMP packets that successfully match a PERMIT rule and are identified as matching an existing ESTABLISHED connection or are part of a NEW connection are subjected to flood protection filtering (see "Flood protection filtering" on page 28). If the permitted packets cannot be correctly matched to an existing connection, are not related to an existing connection, are invalid for starting a new connection, or invalid for another reason, the packets are considered to be invalid and will be dropped. Dropped invalid packets will produce a log with the prefix:

```
Firewall: DENY INVALID
```

at a maximum rate of one log message per second.

Some criteria for packets to be considered invalid are:

- The total maximum number of connections has been exceeded. The maximum for each AlliedWare Plus firewall model is 100 000 connections.
- Packet is short/truncated/malformed or has a bad checksum.

- For TCP packets:
  - The sequence number is not as expected; for instance, an ACK is received for data that has not yet been transmitted.
  - The connection tracking has become out of sync with the actions of the client and server; marking the packets as invalid and dropping will force the client to initiate a new connection.

Flood protection filtering

Flood-protection filtering acts as an additional layer of defense, and applies only to traffic that has already been permitted by a firewall rule. The flood-protection rate-limiting depends on the model and protocol.

Table 7: Flood-protection rate-limiting

MODEL	TCP SYN CONNECTIONS PER SECOND	SYN BURST	UDP CONNECTIONS PER SECOND	UDP BURST	ICMP CONNECTIONS PER SECOND	ICMP BURST
AR2010V	3333	6000	3333	6000	1000	2000
AR2050V	3333	6000	3333	6000	1000	2000
AR3050S	3333	6000	3333	6000	1000	2000
AR4050S, AR4050S-5G	10000	12000	10000	10000	1000	2000

The filtering works as a standard single-rate traffic meter. The 'per second' figure is the number of new connection attempts per second that will be allowed to connect to the device on each individual UDP/TCP port or per ICMP type. When connection attempts exceed this rate, the excess packets will be matched against the 'burst' bucket until this is exhausted. When the per-second rate is exceeded and the burst bucket is exhausted, all excess packets will be dropped and a maximum of one log message per second will be generated regardless of the number of packets dropped. Logs generated when packets have been dropped by this process will be prefixed with one of:

```
DENY UDPLIMIT reach.
DENY SYNLIMIT reach.
DENY ICMPLIMIT reach.
```

for UDP, TCP or ICMP packets respectively.

Default deny

If a packet is processed by the firewall and does not match any of the permit, deny or reject action rules, it will hit the final default deny rule, and produce a log with the prefix:

```
Firewall: DENY in policy
```

to a maximum rate of 20 log messages per second.

Logging for user-configured rules

There are two ways to log firewall events. The first is to configure the rule with the terminating **log** parameter. When packets are logged in this way, the action (deny, permit, or reject) is applied and a log message is also generated each time the rule is hit. The disposition for these log messages is 'PERMIT', 'DENY' or 'REJECT' according to the action of the rule.

The second way is to configure the firewall rule with **log** as the action. When packets are logged in this way, they continue to be processed by subsequent firewall rules, which may eventually permit, deny or reject the packets. The disposition for these log messages is 'LOG'. Because this action does not affect the traffic, it may be more useful for diagnostic purposes.

Note that it is possible to configure both methods in one rule, but this would result in duplicated log messages.

Some log messages that should be generated when packets match these rules may be dropped by the system under heavy traffic loads.

Firewall log messages

Firewall log messages are logged with facility 'local5', and have severity level 'info' (6). The message part includes information in the following format:

```
Firewall [rule <rule>]: <action> IN=<input-interface> OUT=<output-
interface> SRC=<source-ip> DST=<dest-ip> MARK=<mark> ...
```

Table 8: Elements in firewall log messages

Message element	Description
<rule>	The number of the firewall rule applied. If a packet is dropped by the default deny policy, there is no rule number.
<action>	The action applied to the packet or flow by the firewall; one of DENY, LOG, PERMIT or REJECT.
<input-interface>	The interface via which the traffic was received by the firewall.
<output-interface>	The interface via which the traffic was to be transmitted by the firewall.
<source-ip>	The source IP address of the packet.
<dest-ip>	The destination IP address of the packet.
<mark>	The DPI mark—the last 3 digits are the DPI application index in hexadecimal.
...	Any other packet details available.

Output 1: Example firewall log messages

```
2022 Jul 28 23:26:34 local5.info awplus ulogd[432]: Firewall rule 10: PERMIT IN=
OUT=eth0 SRC=192.168.5.2 DST=192.168.5.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=7935
DF PROTO=ICMP TYPE=8 CODE=0 ID=2406 SEQ=1
2022 Jul 25 14:10:38 local5.info awplus ulogd[432]: Firewall: DENY probe FIN
IN=vlan1 OUT=eth1 MAC=00:00:cd:38:00:bc:52:54:6b:6b:0f:1e:08:00 SRC=192.168.1.1
DST=172.16.1.2 LEN=40 TOS=0x00 PREC=0x00 TTL=63 ID=54219 PROTO=TCP SPT=6000 DPT=21
WINDOW=512 RES=0x00 UG PSH FIN URGP=0
2022 Jul 25 18:38:36 local5.info awplus ulogd[432]: Firewall rule 20: PERMIT
IN=eth1 OUT=vlan1 MAC=00:00:cd:38:00:96:52:54:78:36:8f:a6:08:00 SRC=172.16.1.2
DST=192.168.1.1 LEN=239 TOS=0x00 PREC=0x00 TTL=63 ID=20563 DF PROTO=TCP SPT=80
DPT=46254 WINDOW=905 RES=000 ACK PSH URGP=0 MARK=0x1053
```

Firewall connection logging

This feature is supported from AlliedWare Plus version 5.4.7-1.

Firewall connection logging can be enabled to provide additional logs that show the start and end of connections passing through the firewall. These messages are assigned facility local5. They have severity ‘info’ (6).

To enable logging of new connections, closed connections, or both passing through the firewall, use the commands:

```
awplus# configure terminal
awplus(config)# connection-log events {new|end|all}
```

To show the configuration of firewall connection logging, use the following command:

```
awplus# show connection-log events
```

Output 2: Example output from show connection-log events

```
awplus#show connection-log events
Log new connection events:    Disabled
Log connection end events:    Enabled
```

New connection log messages includes information in the following format for a newly started firewall connection:

```
NEW proto={tcp|udp|icmp|...|<number>} orig_src={<ipv4-addr>|<ipv6-addr>}
orig_dst={<ipv4-addr>|<ipv6-addr>} [orig_sport=<source-port>]
[orig_dport=<dest-port>] reply_src={<ipv4-addr>|<ipv6-addr>}
reply_dst={<ipv4-addr>|<ipv6-addr>} reply_sport=<source-port>
reply_dport=<dest-port>
```

Closed connection log messages includes information in the following format for a firewall connection that has ended:

```
END proto=[tcp|udp|icmp|...|<protocol-number>] orig_src={<ipv4-addr>|
<ipv6-addr>} orig_dst={<ipv4-addr>|<ipv6-addr>} [orig_sport=<source-port>]
[orig_dport=<dest-port>] orig_pkts=<packets> orig_bytes=<bytes>
reply_src={<ipv4-addr>|<ipv6-addr>} reply_dst={<ipv4-addr>|<ipv6-addr>}
reply_sport=<source-port> reply_dport=<dest-port> reply_pkts=<number>
reply_bytes=<number>
```

Table 9: Elements in firewall connection log messages

Message elements	Description
`proto={tcp|udp|icmp|<protocol>|<number>}`	The protocol or protocol number for the connection.
`orig_src={<ipv4-addr>|<ipv6-addr>}`	The source IPv4 or IPv6 address of the packet originating the connection.
`orig_dst={<ipv4-addr>|<ipv6-addr>}`	The destination IPv4 or IPv6 address for the packet originating the connection.
`orig_sport=<source-port>`	The source port number of the originating packet.
`orig_dport=<dest-port>`	The destination port number of the originating packet.
`orig_pkts=<packets>`	The total number of packets passed in the originating direction.
`orig_bytes=<bytes>`	The total number of bytes passed in the originating direction.
`reply_src={<ipv4-addr>|<ipv6-addr>`	The source IPv4 or IPv6 address of the returning packets.
`reply_dst={<ipv4-addr>|<ipv6-addr>`	The destination IPv4 or IPv6 address of the returning packets.
`reply_sport=<source-port>`	The source port number of the returning packets.
`reply_dport=<dest-port>`	The destination port number of the returning packets.
`reply_pkts=<number>`	The total number of returning packets.
`reply_bytes=<number>`	The total number of returning bytes.

Note that the original source and destination addresses and ports may differ from the reply source address and destination addresses and ports depending on whether NAT is applied and the type of NAT.

Output 3: Example connection log messages for TCP connection

```
NEW proto=TCP orig_src=192.168.1.100 orig_dst=192.168.1.1 orig_sport=55532
orig_dport=80 reply_src=192.168.1.1 reply_dst=192.168.1.100 reply_sport=80
reply_dport=55532

END proto=TCP orig_src=192.168.1.100 orig_dst=192.168.1.1 orig_sport=55532
orig_dport=80 orig_pkts=7 orig_bytes=522 reply_src=192.168.1.1
reply_dst=192.168.1.100 reply_sport=80 reply_dport=55532 reply_pkts=4
reply_bytes=811
```

Output 4: Example connection log messages for ICMP connection

```
NEW proto=ICMP orig_src=192.168.1.1 orig_dst=192.168.1.100
reply_src=192.168.1.100 reply_dst=192.168.1.1

END proto=ICMP orig_src=192.168.1.1 orig_dst=192.168.1.100 orig_pkts=2
orig_bytes=168 reply_src=192.168.1.100 reply_dst=192.168.1.1 reply_pkts=2
reply_bytes=168
```

To configure an AlliedWare Plus firewall to generate log messages for log events based on facility local5, including these firewall connection events, and send them to a syslog server at IP address 192.168.1.1, use the commands:

```
awplus# configure terminal
awplus(config)# log host 192.168.1.1 facility local5
```

For more information about logging on the AlliedWare Plus firewalls, see the following documents:

- Logging Feature Overview and Configuration Guide
- Log Message Reference for AlliedWare Plus™

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.

AlliedWare Plus 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.

AlliedWare Plus firewalls also support **Enhanced NAT** (ENAT) which gives you the ability to

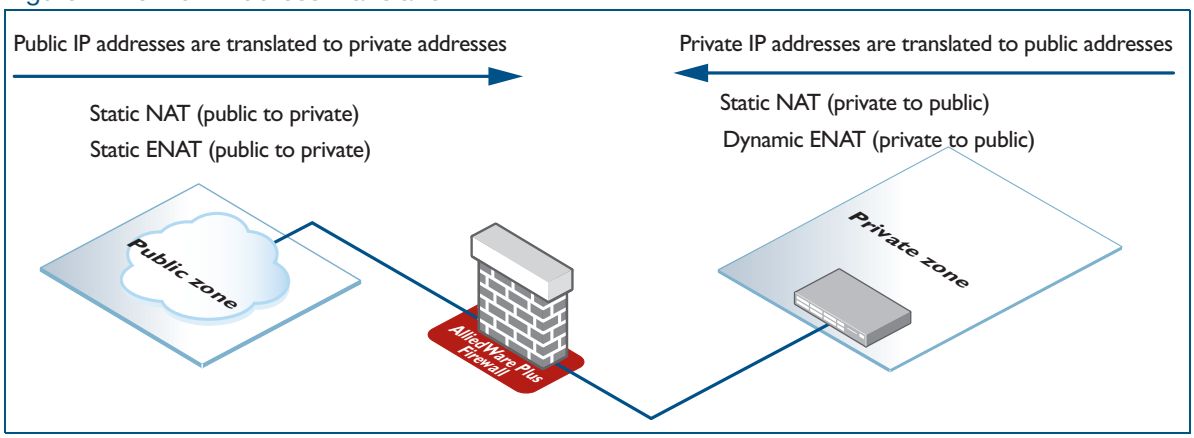
- Configure the global address used in Masquerading and Port Forwarding.
- Perform port translations in Port Forwarding configurations.

AlliedWare Plus 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



Configuring firewall and NAT rules for entities

Firewall rules are constructed as follows:

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

Port forwarding and masquerade NAT rules are constructed as follows:

```
rule [<1-65535>] portfw <application-name> from <source-entity> [to
<destination-host-entity>] with dst <destination-host-entity> [dport
<1-65535>]
```

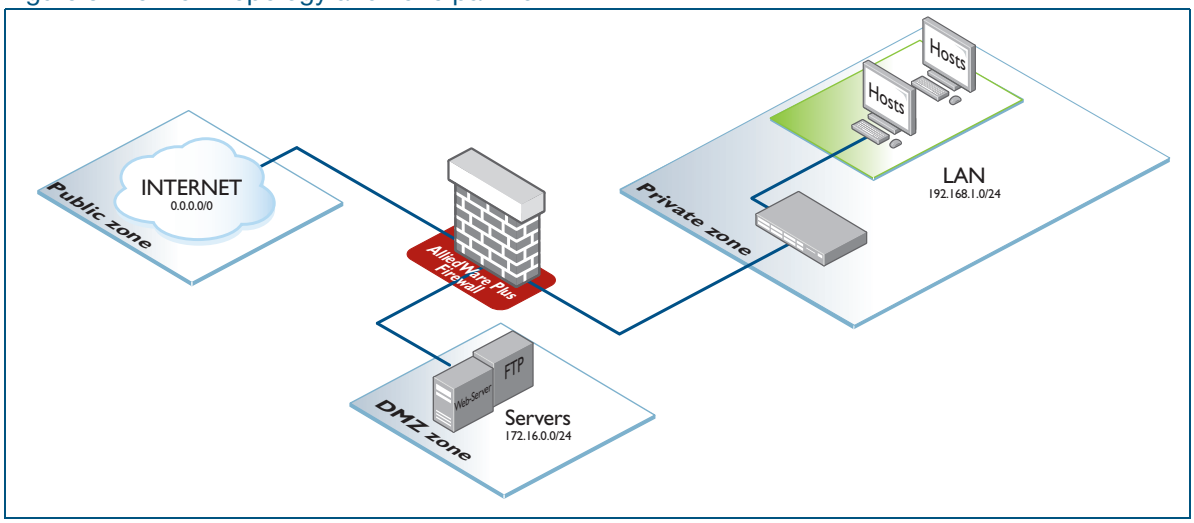
```
rule [<1-65535>] masq <application-name> from <source-entity> to
<destination-entity> [with src <source-host-entity>]
```

From version 5.5.2-1.1 onwards, you can use the tab key to auto-complete application and entity names. This makes it easier to specify the name of an existing DPI application or firewall entity.

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



Step 1: **Configure DMZ zone.**

```
awplus#configure terminal
awplus(config)#zone dmz
awplus(config-zone)#network servers
awplus(config-network)#ip subnet 172.16.0.0/24 interface eth1
awplus(config-host)#host ftp
awplus(config-host)#ip address 172.16.0.2
awplus(config-host)#host web-server
awplus(config-host)#ip address 172.16.0.10
```

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.**

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

Step 9: **Configure interfaces.**

```
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.**

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

Output 5: 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.**

```
awplus#show entity
```

Output 6: 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 7: Example output from the console

```
awplus#show nat rule

[* = Rule is not valid - see "show nat rule config-check"]
-------------------------------------------------------------------------
  ID      Action      From                 With (dst/src) Entity   Hits
          App         To                   With dport
-------------------------------------------------------------------------
  10      masq        private              -                       0
          any         public               -
  20      masq        dmz                  -                       0
          any         public               -
  30      portfwd     public               dmz.servers.ftp         0
          ftp         -                    -
  40      portfwd     public               dmz.servers.web-server  0
          http        -                    -              -
```

Note that there is a configurable maximum of 500 NAT and/or Firewall rules combined. However, the practical limit may reduce as additional features are configured and used on the device, and depending on the system resources available.

Temporary Layer 3 outage when enabling firewall protection

When you enable firewall protection, a temporary L3 outage occurs. If you have a logging host configured, you will see an error on the command terminal and in the log when this happens. The messages look like this:

```
awplus(config-firewall)#protect

16:24:56 awplus IMISH[3883]: [manager@ttyS0]protect

awplus(config-firewall)#16:24:58 awplus firewalld[526]: FW: Disable L3
traffic

16:24:58 awplus syslog-ng[329]: I/O error occurred while writing; fd='64',
error='Operation not permitted (1)'

16:24:58 awplus syslog-ng[329]: Syslog connection broken; fd='64',
server='AF_INET(192.168.251.10:514)', time_reopen='60'
```

You do not need to take any action if you get these messages. Layer 3 routing will restart automatically.

Determining the number of source and destination ports used in rules

You can include TCP and UDP ports in your application definitions, using the **sport** and **dport** parameters.

For example:

```
awplus(config)#application tftp

awplus(config-application)#protocol udp

awplus(config-application)#dport 69
```

You can specify these source or destination ports by specifying one of the following:

- a single port, for example **sport 50010**
- a range of ports, for example **dport 50010 to 50070**
- the keyword **any**, for example **sport any**

The maximum number of source and destination ports per application is 15 (15 source ports and 15 destination ports). This is counted as follows:

- a single **sport** or **dport** counts as 1 port
- a range counts as 2 ports
- the keyword **any** counts as 2 ports.

For example, you can successfully create the following application, because it has only two ports:

```
awplus(config)#application example-1
awplus(config-application)#protocol tcp
awplus(config-application)#dport 50001 to 50016
```

In contrast, in the following configuration, the last entry (**dport 50016**) will fail because the port limit has been reached:

```
awplus(config)#application example-2
awplus(config-application)#protocol tcp
awplus(config-application)#dport 50001
awplus(config-application)#dport 50002
awplus(config-application)#dport 50003
awplus(config-application)#dport 50004
awplus(config-application)#dport 50005
awplus(config-application)#dport 50006
awplus(config-application)#dport 50007
awplus(config-application)#dport 50008
awplus(config-application)#dport 50009
awplus(config-application)#dport 50010
awplus(config-application)#dport 50011
awplus(config-application)#dport 50012
awplus(config-application)#dport 50013
awplus(config-application)#dport 50014
awplus(config-application)#dport 50015
awplus(config-application)#dport 50016 <-- This command fails
```

Configuring NAT rules with DPI

You can configure firewall rules to allow or deny specific application traffic to flow from one entity to another. And most commonly, when using DPI in combination with NAT, it is sufficient to configure a single rule to masq any traffic from LAN to WAN without the need to configure NAT rules for each application. You may also configure a few NAT port forwarding rules to allow external traffic from the Internet to the public IP address to be translated to reach the internal addresses of internal servers.

For example:

```
awplus(config)# nat
awplus(config-nat)# enable
awplus(config-nat)# rule masq any from lan to wan
awplus(config-nat)# exit
awplus(config)# exit
```

However, if you configure NAT rules to selectively apply address translation to specific application traffic only, you may find that the application traffic matching the NAT rules will not be forwarded even with DPI enabled. This is because the DPI engine cannot positively identify the application until after the first few packets associated with the application flow have been seen. Therefore, NAT does not know what to do with the initial packets of a new flow, as they will not match any defined application-specific NAT rules.

There are two solutions to this problem:

Solution 1: Create a new custom definition

The first alternative for allowing DPI-permitted traffic through NAT rules is to create a new custom definition for the application for the NAT rule.

Step 1: **Create a new custom application definition.**

Create a new custom definition for the application for the NAT rule. For example:

```
awplus(config)# application customapp
awplus(config-application)# protocol tcp
awplus(config-application)# sport 300 to 65535
awplus(config-application)# dport 45
```

Step 2: **Apply this application to NAT rules.**

```
awplus(config)# nat
awplus(config-nat)# enable
awplus(config-nat)# rule masq customapp from lan to wan
awplus(config-nat)# exit
awplus(config)# exit
```

Confirm that the NAT rules with the specified application are valid.

```
awplus#show nat rule

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

-----------------------------------------------------------------------
  ID     Action     From                  With (dst/src) Entity   Hits
         App        To                    With dport
-----------------------------------------------------------------------
  10     masq       lan                   -                       0
         customapp  wan                   -
```

Solution 2: Override the DPI definition

The second alternative for allowing DPI-permitted traffic through NAT rules is to statically configure an application with the same name as the DPI application. The statically configured application overrides any previously defined DPI-based settings. For example:

```
awplus(config)# application mail
awplus(config-application)# protocol tcp
awplus(config-application)# sport 500 to 10000
awplus(config-application)# dport 50
awplus(config-application)# exit
awplus(config)# nat
awplus(config-nat)# rule masq mail from lan to wan
awplus(config-nat)# end
```

Confirm that the NAT rules with the specified application are valid.

```
awplus#show nat rule

[* = Rule is not valid - see "show nat rule config-check"]
-----------------------------------------------------------------------
  ID     Action     From                  With (dst/src) Entity   Hits
         App        To                    With dport
-----------------------------------------------------------------------
  10     masq       lan                   -                       0
         mail       wan
```

When DPI is enabled, because there is a user-defined application called ‘mail’, it will not be replaced by the DPI definition. The user-defined application has priority.

For more information about DPI, see the Application Awareness Feature Overview and Configuration Guide.

Configuring the 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.

```
!
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 a firewall rule for external services

In addition to forwarding packet flows between interfaces, AlliedWare Plus firewalls often need to initiate packets flows of their own to the Internet in order to provide various services.

Some common examples are:

- DNS lookups and DNS relay
- Update Manager
- Web Control queries
- Routing protocols

When using the firewall, you will need rules that allow traffic from each of these services to egress the device. You can use one of these methods to permit this traffic:

- Configure a single firewall rule that allows any flow initiated from the device to egress. Flows initiated by the device can be trusted, so firewall rules for them do not need to be as selective as rules for other traffic. We recommend this method. Follow the configuration below.
- Alternatively, you can configure specific rules to allow each protocol originating from the AlliedWare Plus firewall to egress. However, to do this you need to understand the protocols each service uses to operate, and some of these are proprietary. (For instance, for Update Manager traffic, see "Configuring firewall rules with update manager" on page 42.)

Also, from version 5.4.8-0.x onwards, flows initiated by the device are processed by DPI (when enabled) before being processed by the firewall. So if DPI is enabled, you also need a rule to allow the first 'undecided' packets in a locally initiated flow before DPI has identified their application.

Configuration

Step 1: Configure network entity

You can create a network entity for the services, which are 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 services' source traffic

You can create an entity for the services' 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
awplus(config-host)#end
```

Step 3: Configure a firewall rule

Then, to configure a rule to allow services originating from the AlliedWare Plus firewall to egress (with or without DPI), you can use a command like this:

```
awplus(config-firewall)# rule permit any from ROUTER.EXTERNAL.EXTERNAL_INT
to public
```

Configuring firewall rules with update manager

The Update Manager is a tool to enable an AlliedWare Plus device to keep 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 traffic between the Update Manager and the Update Server. You can use one of these methods to permit this traffic:

- Create several firewall rules to selectively permit traffic between the Update Manager and the Update Server, as described in this section.
- Use the simpler and less restrictive configuration described in "Configuring a firewall rule for external services" on page 41).

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
awplus(config-host)#end
```

Firewall rules

Step 3: Configure firewall rules

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

```
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
```

With DPI enabled

If Deep Packet Inspection (DPI) is enabled, then you will need to configure a rule to allow the initial ‘undecided’ traffic in a new flow before DPI has identified which application it belongs to:

```
awplus(config-firewall)#rule permit undecided from ROUTER.EXTERNAL.EXTERNAL_INT
to public
```

DPI with built-in library

If DPI is enabled with the internal (built-in) library, then the IP traffic originating from the Update Manager will be classified as SSL. You will need a rule to permit the SSL traffic.

```
awplus(config-firewall)#rule permit ssl from ROUTER.EXTERNAL.EXTERNAL_INT
to public
```

DPI with Procera

If DPI is enabled with the external library (the **provider procera** command), then the IP traffic from the Update Manager will initially be identified as TCP (from AlliedWare Plus version 5.4.8-0.1), then as SSL, and then as HTTPS. In addition to the rules above allowing HTTPS and DNS traffic, you will also need to allow this TCP and SSL traffic through the firewall. You can do this by one of these methods:

- Either, configure firewall rules to permit TCP and SSL traffic originating from the WAN interface:

```
awplus(config-firewall)#rule permit tcp from ROUTER.EXTERNAL.EXTERNAL_INT to public
awplus(config-firewall)#rule permit ssl from ROUTER.EXTERNAL.EXTERNAL_INT to public
```

- Or, configure a custom application for the Update Manager, and add TCP to it:

```
awplus(config-host)#
awplus#configure terminal
awplus(config)#application update_manager
awplus(config-application)# protocol tcp
awplus(config-application)# dport 443
awplus(config-application)#exit
awplus(config)#firewall
awplus(config-firewall)#rule permit update_manager from ROUTER.EXTERNAL.EXTERNAL_INT to public
```

For more information about Application Awareness and DPI, see Application Awareness Feature Overview and Configuration Guide.

Configuring firewall rules with subscription licensing

AlliedWare Plus devices configured with features such as AMF and OpenFlow use subscription-based licensing. These devices could be located within a private firewall zone, accessing the subscription service located in the Internet, via the AlliedWare Plus firewall.

In order to allow access to the subscription licensing services from a private zone to the Internet, firewall allow rules need to be created. For more information about Subscription Licensing, see the Licensing Feature Overview and Configuration Guide.

In order to allow the AlliedWare Plus firewall itself to access subscription licensing services, see "Configuring firewall rules with update manager" on page 42.

Step 1: Configure an entity for the public zone attached to the Internet

Configure a public zone attached to the Internet, where 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 an entity for the private zone**

You can create a private zone, which is associated with the internal network accessed via interface vlan1

```
awplus(config)#zone private
awplus(config-zone)#network INTERNAL
awplus(config-network)#ip subnet 10.1.1.0/24 interface vlan1
```

Step 3: **Configure firewall rules**

Subscription services are accessed using HTTPS protocol. You can create a firewall rule to allow HTTPS application to flow through the AlliedWare Plus firewall from the private to public zones.

```
awplus(config-host)#end
awplus#configure terminal
awplus(config)#firewall
awplus(config-firewall)#rule permit https from private to public
```

Similarly, you can create a rule to allow DNS resolution of the subscription service URL if the DNS server is reachable via the WAN interface from devices located within the private zone.

```
awplus(config-firewall)#rule permit dns from private to public
```

If the AlliedWare Plus firewall is also performing NAT, then corresponding NAT-based masquerade rules for HTTPS and DNS will also need to be configured.

Configuring TCP established session timeout

By default, when a TCP session is successfully established through the AlliedWare Plus firewall, when the session goes idle it automatically times out of the firewall connection tracking table after 3600 seconds. In some situations it may be beneficial to time out unused established TCP sessions earlier.

For example, in a busy environment where there is an excessive number of sessions being established, the firewall connection tracking table could become oversubscribed, with new connections being blocked until older sessions are timed out.

From release 5.4.7-1.x onwards, the following command is available to set a non-default TCP session timeout for established idle sessions:

```
ip tcp timeout established <1-31536000>
```

Configuring UDP/TCP connection limiting (per entity)

From software version 5.5.0-1.1 onwards, firewall session limiting rules apply to both UDP and TCP connections, whereas previously, the limiting rules only applied to TCP sessions.

You can configure a limit for the number of allowed firewall connections associated with a particular entity (Zone/Network/Host). A limit applied to an entity with multiple hosts will apply the limit to each individual host address, not the total cumulative connections for the entity. The limit will be applied to both IPv4 and IPv6 connections. For example, a limit of 1000 would allow up to 1000 IPv4 and 1000 IPv6 connections.

The feature applies to new UDP connections and TCP syn packets. When an entity exceeds the limit for new TCP/UDP connections, the connection will be denied. By default, connections are not limited (up to the maximum total number of allowed connections).

Changes to limits only affect new connections. TCP TIME_WAITS are not included when counting the current number of connections associated with a host. Setting a lower limit will not affect existing connections. If a connection limit is removed, any active connections are not affected.

For TCP syn packets and new UDP flows, the connection limits are 50 per second for the address.

Configuration commands

These commands are available in firewall configuration mode.

- `connection-limit <id> from <entity> with limit <1-4096>`
- `no connection-limit <id>`
- `no connection-limit all`
- `show firewall connections limits`
- `show firewall connections limits config-check`

For example, to apply a TCP/UDP session limit for the entity private, use the following command:

```
awplus(config-firewall)# connection-limit 1 from private with limit 1000
```

Configuring firewall rules 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:

```
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 firewall rules to allow ICMP error messages when DPI is enabled

When using Firewall without DPI, ICMP error messages associated with a permitted flow (such as ‘fragmentation needed’, ‘host unreachable’, etc) are also permitted through the firewall. However, when DPI is enabled, these messages are not permitted. If you require ICMP errors messages to be passed through the Firewall, you can create applications and rules to allow it. We recommend making this as restricted as possible.

As as example, suppose there is a web server on the internal network that can be accessed by hosts in the sales network. Some of those hosts may connect via VPN, which has a reduced MTU. Routers leading to those hosts need to be able to send ICMP fragmentation needed messages back to the web server if it sends packets that are too big for a VPN tunnel. When DPI is enabled, you can achieve this with the following steps.

These steps assume that the firewall is already enabled and configured so that the sales hosts can reach the web server. This includes the definition of the associated entities sales.sales_net and internal.servers.web.

Step 1: Define a custom application for the ICMP fragmentation needed messages (type 3 Destination Unreachable, code 4 Too Big)

```
awplus#configure terminal
awplus(config)#application frag_needed
awplus(config-application)#protocol icmp
awplus(config-application)#icmp-type 3
awplus(config-application)#icmp-code 4
awplus(config-application)#exit
```

Step 2: **Configure an entity for the private zone**

You can create a private zone, which is associated with the internal network accessed via interface vlan1

```
awplus(config)#firewall
awplus(config-firewall)#rule permit frag_needed from sales.sales_net to
internal.servers.web
```

Alternatively, if you need to support the whole range of ICMP destination unreachable messages, you can define an application with ICMP-type 3 but no ICMP code:

```
awplus#configure terminal
awplus(config)#application unreach
awplus(config-application)#protocol icmp
awplus(config-application)#icmp-type 3
```

Configuring NAT loopback with DMZ

NAT loopback can be used 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.

In simple terms, this means that NAT loopback allows devices inside a private network (like your office) to access services within the network's DMZ (a secure part of the network) using the same public address or domain name that someone outside the office would use. This is useful when the internal devices don't have their own internal DNS (domain name system) and rely on the same external addresses as external users.

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

Firewall rules 10, 20, 50 (Figure 6 on page 50) are configured to allow traffic from clients within the private zone to access the Internet and the DMZ zone. Firewall rule 40 is to allow only HTTP traffic from the Internet to reach the web server in the DMZ. Firewall rule 60 and 70 are included to allow HTTP traffic initiated from the web server access to the private zone and the public zone. If traffic is not initiated from the web server, then rules 60 and 70 are not required.

A client initiated DNS request to the domain name associated with the service resolves to the public IP address of the AlliedWare Plus firewall.

The client then sends its HTTP request to the public IP address of the AlliedWare Plus firewall. A static ENAT port forwarding rule (NAT rule 20 in Figure 6) is used to translate the destination IP address to become the IP address of the server located in the DMZ ("Configuring a Static ENAT rule" on page 51).

The service is accessed by sending a request to the public IP address of the AlliedWare Plus 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 in the DMZ zone is also accessible when the user is physically located outside of the office and accesses the service directly from the Internet via the same ENAT port forwarding rule (NAT rule 20 in Figure 6).

The source IP of traffic to the Internet from clients located within the private zone are translated to become the public IP used on eth1 (NAT rule 40).

An optional dynamic ENAT masquerade rule (NAT rule 50 in Figure 6) 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 in the DMZ to reach private zone clients, and should not be required if the server only send reply traffic. ("Configuring a Dynamic ENAT rule" on page 51).

Figure 4: Physical network

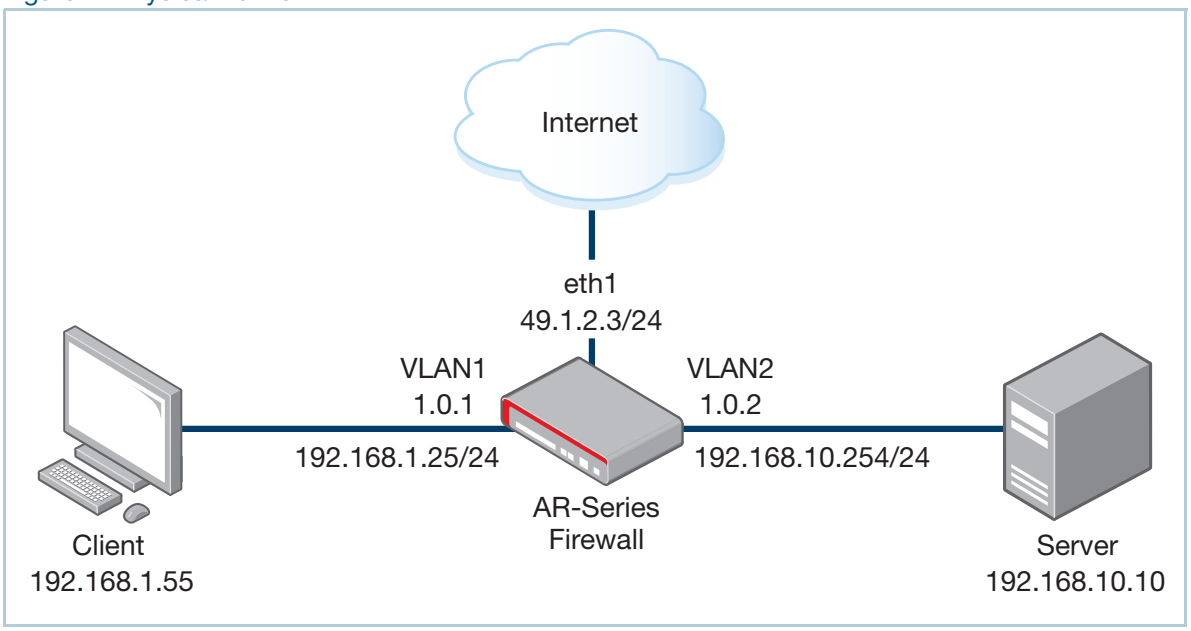


Figure 5: AlliedWare Plus firewall entity map

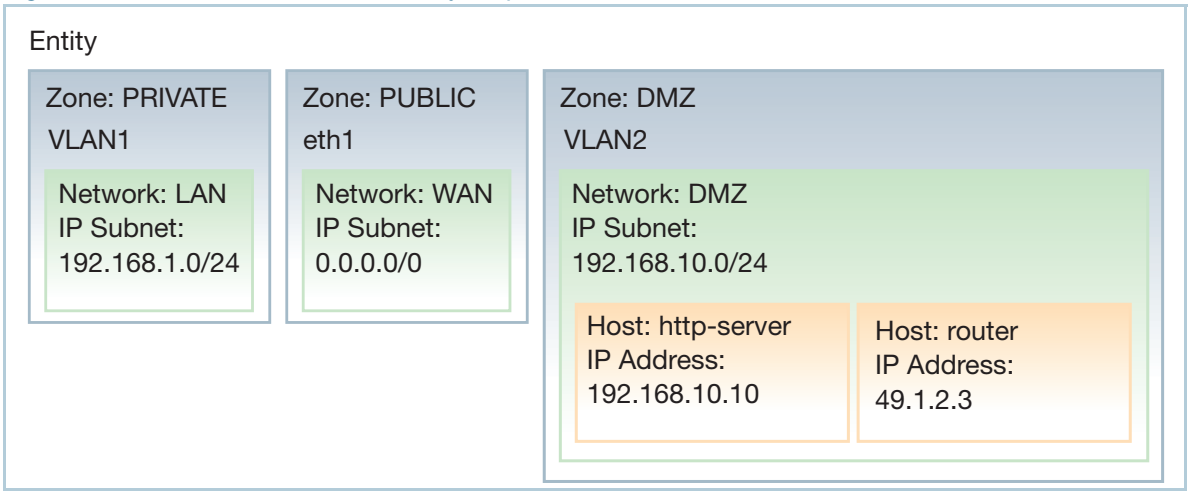


Figure 6: Configuration: NAT loopback with DMZ

```
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
 #rule 70 permit http from dmz.dmz.http-server to public
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
```

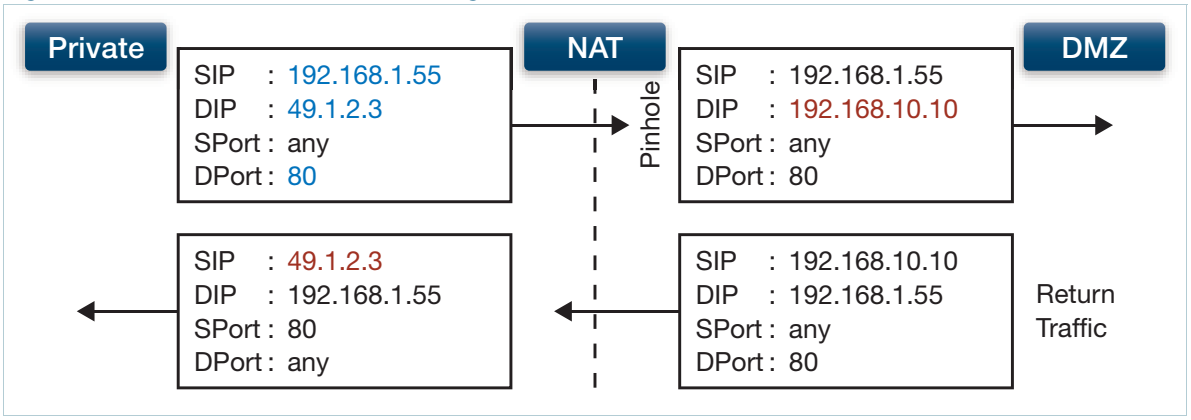
Configuring a Static ENAT rule

Also, 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 matches and 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



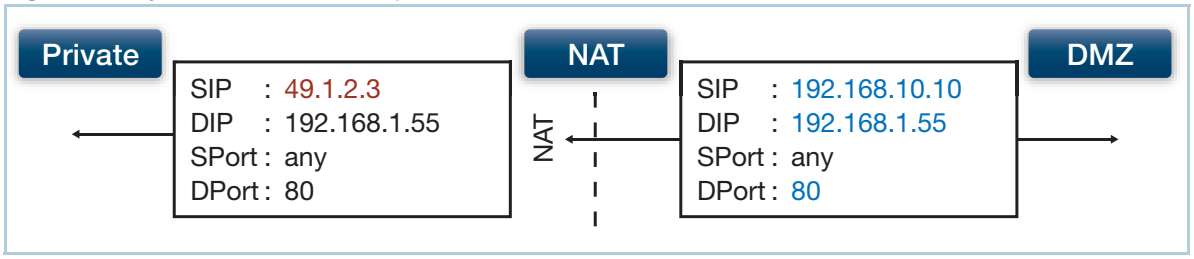
Configuring a Dynamic ENAT rule

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. This Dynamic ENAT rule is only required if traffic is initiated from the server. Server 'reply' traffic is matched by the preceding Static ENAT rule, making the Dynamic ENAT rule unnecessary for most situations.

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 AlliedWare Plus 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 (10.1.1.1) located on the Internet is routed to a different IP address (172.22.0.3) in order to reach the web server. The eth1 WAN interface itself does not need to be configured with the public IP address (172.22.0.3) allocated to 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 port-forwarding rule 1:1 maps the external public IP address (172.22.0.3) to the actual private IP address (172.22.200.3) configured on the web server.

Since the public eth1 interface itself is not configured with the public IP address allocated for the server, the firewall is also configured to send proxy-ARP responses to ARP requests to the public web server IP address (172.22.0.3). And 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

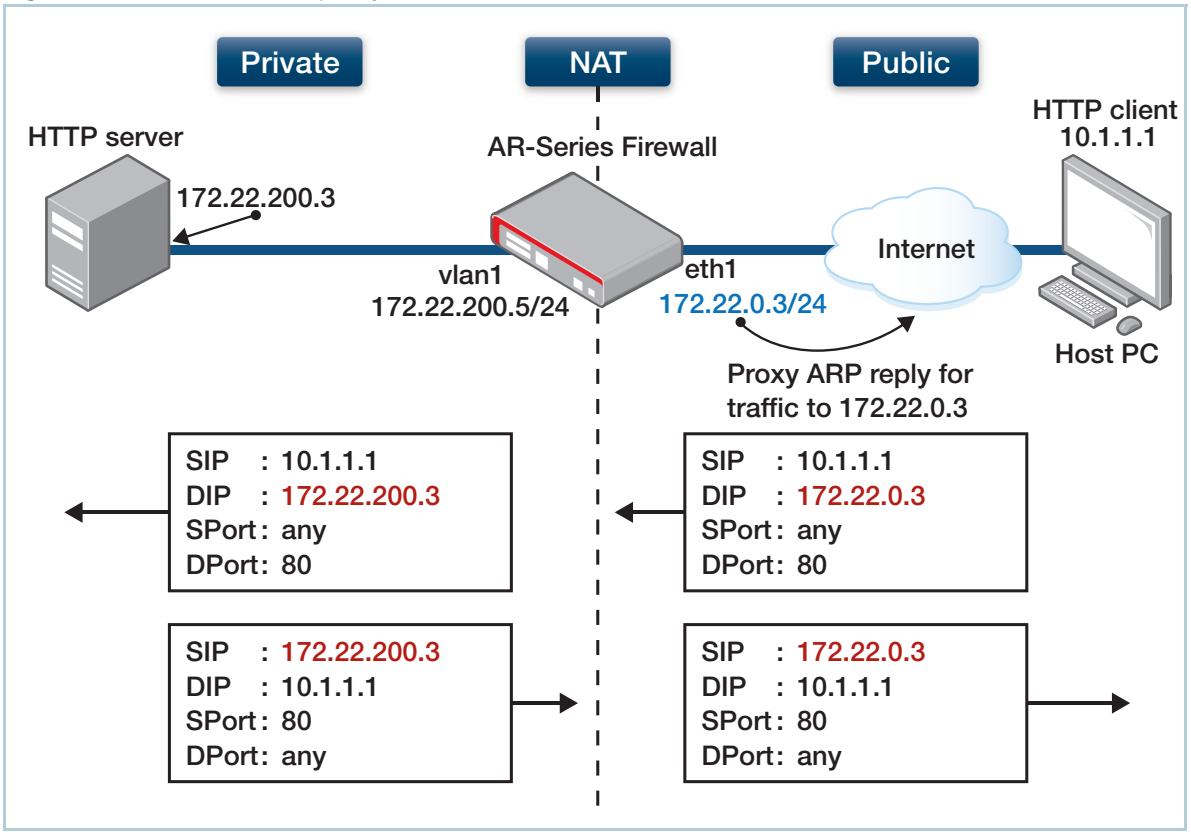


Figure 12: Configuration for static NAT with proxy-ARP

```
! 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
    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
```

Configuring source-based NAT with secondary IP addresses

In the example below, the link between the AlliedWare Plus 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 AlliedWare Plus 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 AlliedWare Plus firewall.

All traffic originating from the 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 AlliedWare Plus 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 AlliedWare Plus 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

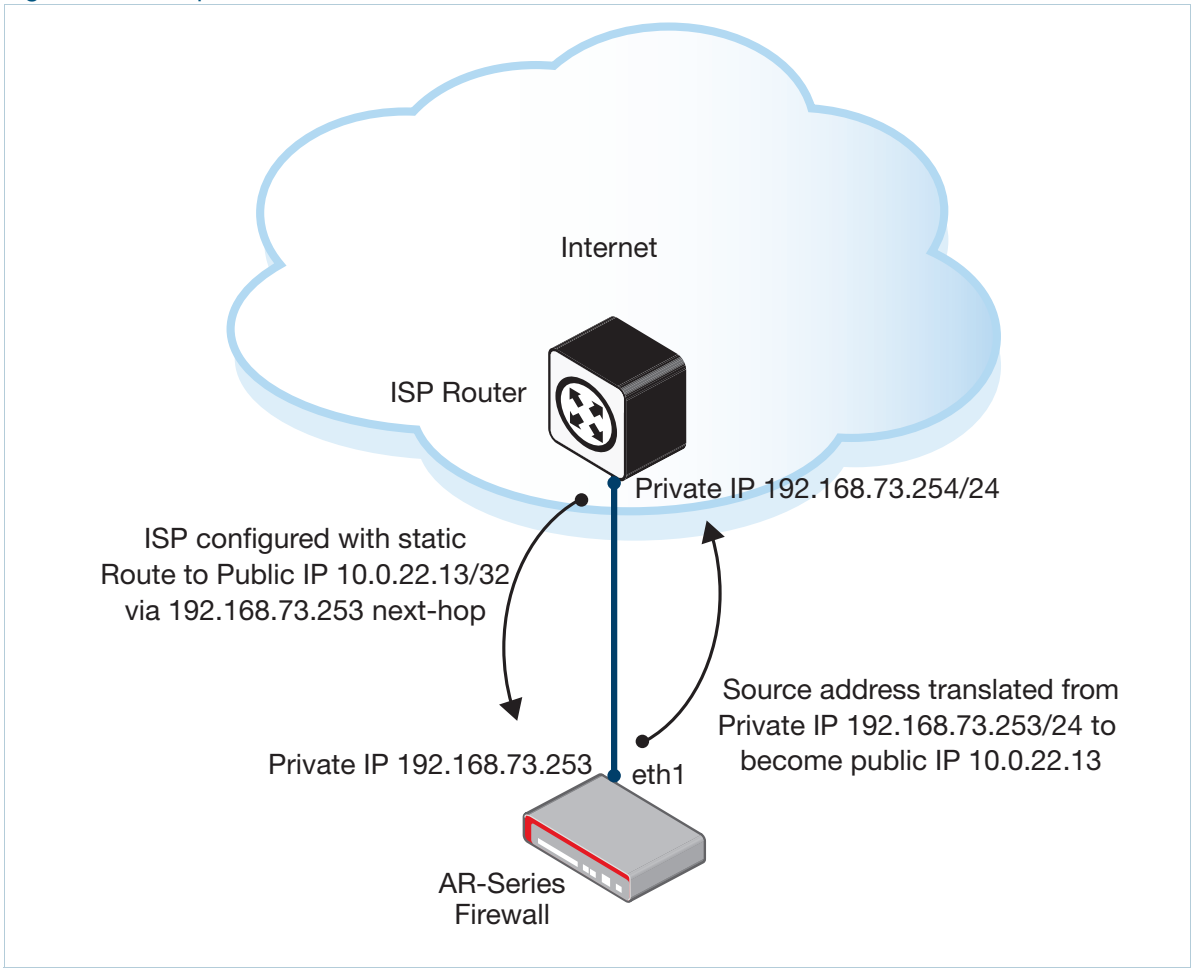


Figure 14: Configuration for source-based NAT

```
!
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 10.0.22.13/32 secondary
!
ip route 0.0.0.0/0 192.168.73.254
!
```

Configuring access to multiple internal servers via PPPoE WAN

This section provides two examples showing how to configure access to multiple internal application servers via PPPoE WAN, protected by firewall with NAT. The topology uses an AlliedWare Plus UTM Firewall or Secure VPN Router providing Internet access via a PPPoE WAN link.

This topology uses firewall zones including:

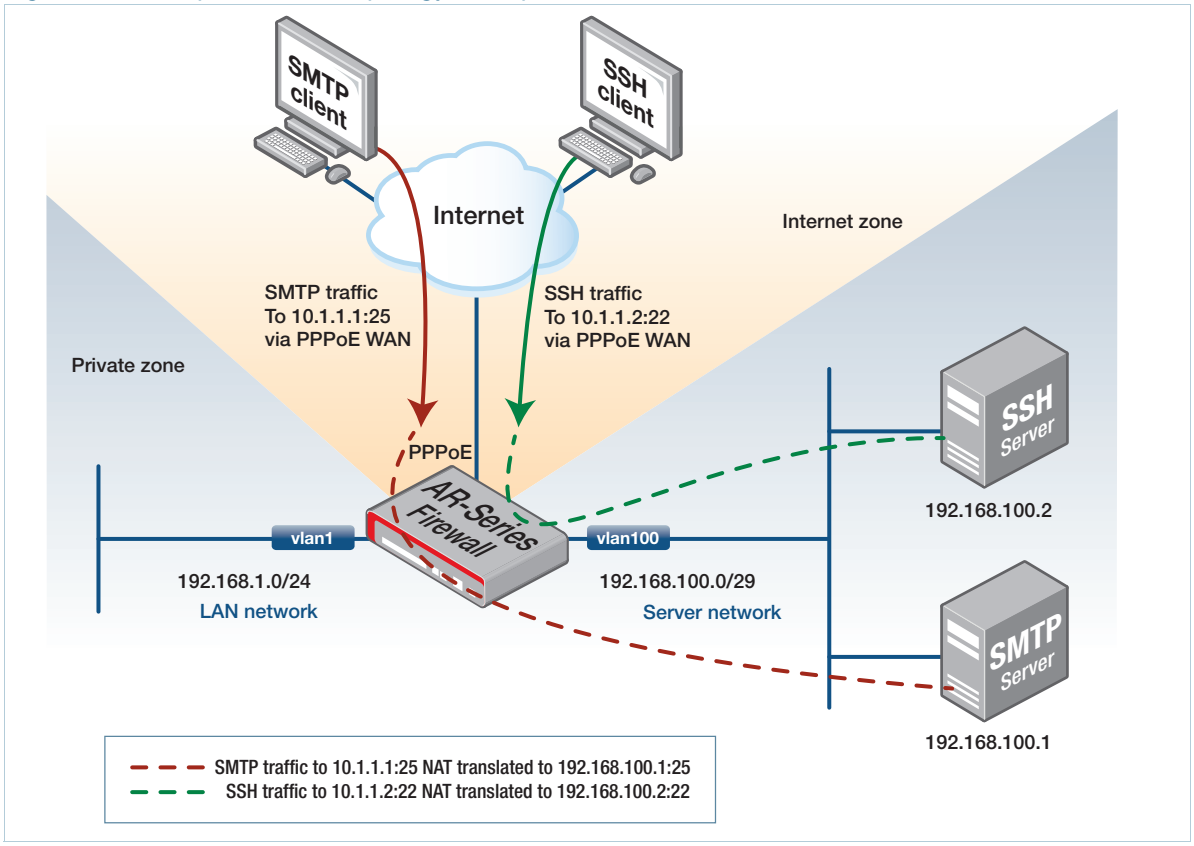
- The public 'internet' zone, with the PPP interface in it.
- The ‘private’ zone, containing a LAN with host computers and a separate server network containing application servers.

The configuration provides access to the servers from clients located in either the private zone or in the Internet. The two examples are:

- "Configuring server access with external DNS" on page 56—No internal DNS server is used within the private customer network.
- "Configuring server access with internal DNS" on page 61—There is an internal DNS server in the private zone that private clients can access directly without having to perform DNS requests to external DNS servers in the Internet.

The network topology shown below is the same for both examples.

Figure 15: Example network topology: multiple internal servers via PPPoE WAN



Configuring server access with external DNS

This section contains:

- An explanation of the example, describing each aspect of the network, followed by
- An annotated configuration file, showing the commands to configure this example.

This example has a connection to the Internet via a PPPoE client WAN interface in the AlliedWare Plus firewall. It includes configuring a static IPv4 default route to the Internet via the firewall's PPP WAN link.

The ISP router (providing Internet connectivity) must also be configured to route traffic via its PPP interface to an entire public IPv4 subnet (10.1.1.0 mask 255.255.255.248) towards the AlliedWare Plus firewall PPP interface. This allows access from any clients to the range of public IP addresses (within the publicly allocated subnet) that are allocated for use by various application servers hosted at the user site.

External DNS servers (located in the Internet) resolve FQDN to the public IPv4 addresses allocated for use by the servers located at the customer site. No internal DNS server is used in this example.

The ISP router dynamically allocates a single /32 IPv4 host address to the AlliedWare Plus firewall PPPoE WAN interface via standard PPP IPCP negotiation. When the ISP router allocates this address via its PPP interface, the ISP router automatically creates a /32 IPv4 host route for this dynamically allocated address, unless configured otherwise. You need to configure the PPP WAN interface of the firewall to be in a public 'internet' zone.

Note that:

- You do not need to explicitly configure the PPP WAN interface of the AlliedWare Plus firewall itself for addresses within the public IPv4 subnet 10.1.1.0/29 that the ISP routes to.
- The /32 IPv4 host address that is dynamically allocated to the PPP WAN interface of the firewall can be within the same subnet that is allocated for use by the servers, or it can be within a completely different subnet.
- On the AlliedWare Plus firewall, you do **not** need to configure the public IP subnet or public IP addresses allocated to each application server on the PPP WAN.

Basic zones and rules

Physically, the application servers (SMTP and SSH servers used in this example) are located in a private zone of the firewall, and you need to configure them with private IP addresses. Those private IP addresses cannot therefore be directly accessed from clients located on the Internet.

Instead, you need to configure NAT port-forwarding rules so that the AlliedWare Plus firewall 1:1 statically maps specific application traffic to specific public server IP addresses to the actual internal private IP addresses that the network administrator has allocated to each server.

Via NAT port-forwarding rules, inbound application traffic flows from a client to each public server IP address/application destination port has its destination IP translated to the private IP address of the application server. When the TCP connection is established through to the internal application

server, the internal firewall connection tracker automatically tracks the associated state and details of the TCP session (protocol, IP addresses, ports, NAT translations).

Associated reply traffic from the application server (back to the client) matches the incoming flow originating from the client. And via connection flow association, the reply traffic automatically has its source IP address translated back to the server's public IP address via the same NAT port-forwarding rules handling the inbound traffic flows.

You need to configure a VLAN within the 'private' zone, where the private client host computers are. You may optionally (as in this example) configure the PPPoE WAN interface to obtain DNS information via PPP IPCP negotiation from the ISP router. You need to configure DNS forwarding to allow DNS lookups from private hosts to be proxied, cached (stored) and forwarded through the firewall to allow client DNS resolution.

PPP options & DNS

Optionally (as in this example), you may configure the AlliedWare Plus firewall with static DNS primary and secondary IP name-server addresses, if known. However, if **ppp ipcp dns request** is also configured on the PPP interface, the firewall will automatically use the DNS server address information (when learned via PPP) in preference to any static DNS address entries. The firewall will keep using the dynamically learned DNS server information as long as the PPP connection remains up.

To automatically detect if the PPP WAN connection to the ISP router fails, you can configure the PPP interface **keepalive** option (which enables regular PPP LCP echo request messaging via the PPP interface). This allows the PPP link to automatically re-establish after the firewall detects a link-down event due to a failure to receive keepalive responses (PPP LCP echo responses) from the ISP router.

You can configure TCP MSS clamping on the PPPoE connection (**ip tcp adjust-mss <value>** command), to avoid unnecessary TCP fragmentation issues occurring due to the additional PPP/PPPoE header encapsulations that are applied before transmission out the physical eth WAN interface.

Note that DNS requests to external DNS servers from private hosts in private VLANs resolve to the public IP addresses allocated for each of the servers. Traffic from private hosts are therefore sent to application server public IP addresses, not to internal private IP addresses configured on the servers. Also, traffic from those private hosts in vlan1 to application server IP addresses do not ingress the PPP WAN interface, as they are instead received via the private VLAN interface. Therefore, you need to configure a firewall zone labeled 'any', which is not associated with any interface, but which matches all traffic from all networks, including any traffic from the Internet as well as from clients in the private zone.

Zone and rules for external DNS

```
!
zone any
network all
 ip subnet 0.0.0.0/0
!
```

Both the firewall allow rules and NAT port-forwarding rules use this zone named 'any', ensuring that client traffic from the Internet to the public application server IP addresses are NATed and forwarded to the internal application server IP addresses. Similarly, the same firewall and NAT port-forwarding rules ensure that traffic from private clients to the public application server IP addresses are automatically 'looped back', NATed and forwarded to the internal application server private IP addresses (NAT loopback).

In this example, note that the AlliedWare Plus firewall applies the rules (or more specifically, the rule actions) in the following order:

1. NAT port-forwarding rule actions, before any other firewall rule actions
2. Any other firewall rule actions
3. NAT masquerade rule actions, after any firewall rule actions

NAT port-forwarding rules will have already translated destination IP address(es) to become private IP address(es) based on the order of NAT and firewall rule processing. The firewall rules (allowing application traffic), are therefore configured to allow application traffic to reach destination internal private server IP addresses. This is instead of allowing access to the destination public server IP addresses as one might otherwise assume.

Configuration

Figure 16: Complete device configuration for server access with external DNS example

```
!
no service ssh
!
zone any      ←All traffic from all public or private networks matches this zone labeled ‘any’, as it‘s not interface specific.
 network all
  ip subnet 0.0.0.0/0
!

zone internet
 network server_public      ←Server traffic is sent to specific server host IP addresses within this public server subnet.
  ip subnet 10.1.1.0/29
  host smtp
   ip address 10.1.1.1
  host ssh
   ip address 10.1.1.2
network wan      ←This network entity matches all traffic ingressing/egressing via the PPP link only, from the entire Internet.
  ip subnet 0.0.0.0/0 interface ppp0
  host ppp
   ip address dynamic interface ppp0      ←This command accounts for the PPP WAN IP address, which is dynamically assigned by the ISP router.
!
```

Figure 16: Complete device configuration for server access with external DNS example (continued)

```
zone private
network lan      ←This is the private LAN where network PCs are connected.
  ip subnet 192.168.1.0/24
network server    ←This is the private server network where servers with private IP addresses are located.
  ip subnet 192.168.100.0/29
  host smtp_server
   ip address 192.168.100.1
  host ssh_server
   ip address 192.168.100.2
!

application smtp_app      ←Statically configure an application to match SMTP server traffic, if not already available in application list.
protocol tcp
 # sport any     ←Command commented out because it is not required. If a specific source or destination port or port range is not explicitly configured, then by default port range any is automatically used.
dport 25
!
application ssh_app   ← Statically configure an application to match SSH server traffic, if not already available in application list.
protocol tcp
 # sport any     ←Command commented out because it is not required. If a specific source or destination port or port range is not explicitly configured, then by default port range any is automatically used.
dport 22
!

firewall
 rule 10 permit any from private to private ←Allow all private to private traffic flows.
 rule 20 permit any from private to internet ←Allow all private to internet traffic flows.
 rule 30 permit any from internet.wan.ppp to internet ←Allow traffic from PPP source IP address to access internet, e.g., for ping test to Internet or DNS lookups from the firewall.
 rule 100 permit smtp_app from any to private.server.smtp_server ←Allow traffic from private vlan1 or from the Internet to reach the internal private IP address of the server.
 rule 200 permit ssh_app from any to private.server.ssh_server ←Allow traffic from private vlan1 or from the Internet to reach the internal private IP address of the server.
# rule 300 permit smtp_app from any to internet.server_public.smtp ←Rule commented out because it's not required. AlliedWare Plus firewalls apply firewall rules after NAT port-forwarding address translation. (So destination IP address is already translated to private IP address, so rule 100 is configured instead.)
# rule 400 permit ssh_app from any to internet.server_public.ssh ←Rule commented out because it's not required. AlliedWare Plus firewalls apply firewall rules after NAT port-forwarding address translation. (So destination IP address is already translated to private IP address, so rule 200 is configured instead.)
protect     ←Enable firewall protection.
!
```

Figure 16: Complete device configuration for server access with external DNS example (continued)

```
nat
 rule 10 masq any from private to internet    ←Any traffic to internet originating from private zone has its source IP translated to become public IP allocated to PPP WAN. (Server reply traffic does not match this rule 10. Server reply traffic is matched by NAT rules 100/200 instead.)
 rule 100 portfwd smtp_app from any to internet.server_public.smtp with dst private.server.smtp_server    ←Any traffic to SMTP server wan host IP has its dest IP translated to become internal server IP.
 rule 200 portfwd ssh_app from any to internet.server_public.ssh with dst private.server.ssh_server    ←Any traffic to SSH server wan host IP has its dest IP translated to become internal server IP.
 # rule 300 masq smtp_app from private.server.smtp_server to public with src internet.server_public.smtp    ←Rule commented out because it's not required. SMTP server 'reply traffic' automatically matches rule 100 above, by automatic inbound/outbound connection flow association via the internal firewall connection flow tracker. You only need this masquerade rule if there is a server initiating an outbound connection, and you need to modify the source IP address on egress to the Internet to use a different IP address than the one allocated on the WAN interface.
# rule 400 masq ssh_app from private.server.ssh_server to public with src internet.server_public.ssh    ←Rule commented out because it's not required. SSH server 'reply traffic' automatically matches rule 200 above, by automatic inbound/outbound connection flow association via the internal firewall connection flow tracker. You only need this masquerade rule if there is a server initiating an outbound connection, and you need to modify the source IP address on egress to the Internet to use a different IP address than the one allocated on the PPPoE WAN interface.

enable  ←Enable NAT.
!
# ip name-server <address>    ←You can configure Primary and Secondary DNS server IP addresses if known.
ip domain-lookup
!
vlan database
 vlan 100 state enable
!

interface eth1
 encapsulation ppp 0
!
interface vlan1
 description "private lan for network host computers"
 ip address 192.168.1.254/24
!
interface vlan100
 description "server lan"
 ip address 192.168.100.6/29
!
```

Figure 16: Complete device configuration for server access with external DNS example (continued)

```
interface ppp0
 description "internet WAN PPP connection, with dynamic IP allocated from ISP"
 ppp service-name <service_name>    ←Service name 'any' is default if service name not explicitly configured.
 ppp ipcp dns request
 keepalive
 ip address negotiated
 ppp username <username>
 ppp password <password>
 mtu 1492
 ip tcp adjust-mss 1452   ←Configure TCP MSS clamping on WAN (IPv4 MSS value calculated as interface MTU less 40).
!
ip route 0.0.0.0/0 ppp0    ←Static default route to the Internet via PPP WAN.
!
ip dns forwarding    ←Configure DNS requests from LAN clients to be proxied through AlliedWare Plus firewall and results stored in DNS cache.
ip dns forwarding cache size 10 timeout 3600
!
```

Configuring server access with internal DNS

This section describes the differences in the firewall configuration necessary to support server access with internal DNS. Apart from changes to the firewall configuration the rest of the device configuration remains unchanged.

In this second example, there is an internal DNS server that is attached to an interface within the private zone of the firewall. DNS requests from private clients to the DNS server will typically resolve to the private internal application server IP address, not the public IP addresses of the application servers.

If this is the case, you can simplify the configuration of the firewall and NAT components of the device configuration contained in Table 16 as follows, with the **zone any** and associated configuration commented out.

Configuration

Figure 17: Firewall (partial) configuration for server access with internal DNS

```
!
# zone any    ←Zone and associated network entity configuration commented out because it's not required. Client traffic no longer resolves to public IP address.
# network all
  # ip subnet 0.0.0.0/0
!
```

Figure 17: Firewall (partial) configuration for server access with internal DNS (continued)

```
zone internet
 network server_public ←Server traffic from Internet clients only is sent to specific server host IP addresses within this public server subnet.
   ip subnet 10.1.1.0/29
   host smtp
    ip address 10.1.1.1
   host ssh
    ip address 10.1.1.2
network wan ←This network entity matches all traffic ingressing/egressing via the PPP link only.
   ip subnet 0.0.0.0/0 interface ppp0
   host ppp
    ip address dynamic interface ppp0 ←This entry accounts for the PPP WAN IP, which is dynamically assigned by the ISP router.
!

zone private
network lan ←This is the private LAN where network PCs are located.
   ip subnet 192.168.1.0/24
network server ←This is the private LAN where servers with private IP addresses are located.
   ip subnet 192.168.100.0/29
   host smtp_server
    ip address 192.168.100.1
   host ssh_server
    ip address 192.168.100.2
!

firewall
 rule 10 permit any from private to private  ←Allow all private to private traffic flows. (This includes access from private client IP addresses to private server IP addresses.)
 rule 20 permit any from private to internet ←Allow all private to Internet traffic flows.
 rule 30 permit any from internet.wan.ppp to internet ←Allow traffic from PPP source IP to access internet, e.g., for ping test to the Internet or DNS lookups from the firewall.
 rule 100 permit smtp_app from internet.wan to private.server.smtp_server ←Allow traffic to reach internal private IP of server, via PPP WAN from Internet
 rule 200 permit ssh_app from internet.wan to private.server.ssh_server ←Allow traffic to reach internal private IP of server, via PPP WAN from Internet
protect
!

nat
You do not need any NAT rules from private LAN to server LAN. In this example, because of the internal DNS server, private clients access private server IP addresses directly (without NAT), instead of via public server IP addresses.
 rule 10 masq any from private to internet  ←Any traffic from the private zone to internet has its source IP address translated to become the public IP address allocated to the PPP WAN. (Server reply traffic does not match this NAT rule 10.
 rule 100 portfwd smtp_app from internet.wan to internet.server_public.smtp with dst private.server.smtp_server  ←Any traffic from Internet clients to the SMTP server WAN host IP address has its destination IP address translated to become the internal server IP address. SMTP server reply traffic matches this rule due to automatic internal connection tracker flow association.
rule 200 portfwd ssh_app from internet.wan to internet.server_public.ssh with dst private.server.ssh_server  ←Any traffic from Internet clients to the SSH server WAN host IP address has its destination IP address translated to become the internal server IP address. SSH server reply traffic matches this rule due to automatic internal connection tracker flow association.
enable
```

Diagnostics

You can use the following series of diagnostics commands to verify the solution.

To check if the traffic flows match the rules, use these commands to see rule hit counters:

```
awplus# show firewall rule
awplus# show nat rule
```

To see active session flows that are being connection-tracked (including NAT translations being applied), use the command:

```
awplus# show firewall connections
```

To check that firewall and NAT rules are correctly configured, use the commands:

```
awplus# show firewall rule config-check
awplus# show nat rule config-check
```

Other commands

Other useful commands include:

- To see a list of available applications:

  ```
  awplus# show application [detail]
  ```

- To check the status of interfaces and associated interface counters and addresses:

  ```
  awplus# show interface [brief]
  ```

- With **term mon** enabled, to capture and investigate any PPP/PPPoE negotiation issues:

  ```
  awplus# debug ppp <options>
  ```

  Use this very carefully, as it can generate lots of data onto the CLI command shell screen.

- To see log messages, useful for detecting errors, such as PPP authentication failure due to incorrect PPP username/password:

  ```
  awplus# show log
  ```

- To see status of DNS and cache entries:

  ```
  awplus# show ip dns <options>
  ```

- For a running capture of IP packets traversing an interface:

  ```
  awplus# tcp dump <interface|options>
  ```

  Use this very carefully, as it can generate lots of data onto the CLI command shell screen.

IPoE vs PPPoE

Note that if the WAN link were IP over Ethernet (IPoE) instead of PPPoE, then you would also need to configure Proxy ARP to ensure that the AlliedWare Plus firewall responds to ARP requests for the public server IP addresses. This would be necessary because the server public IP addresses or associated subnet are not physically configured on the WAN interface. (See "Configuring static NAT with proxy ARP" on page 52.)

As the WAN link in these examples is point-to-point, Proxy ARP does not apply and is not required. However, because the link is PPP, the ISP router **must** instead be configured with a route to reach the subnet containing public server IP addresses via its PPP interface.

Configuring Network Address and Port Translation (NAPT)

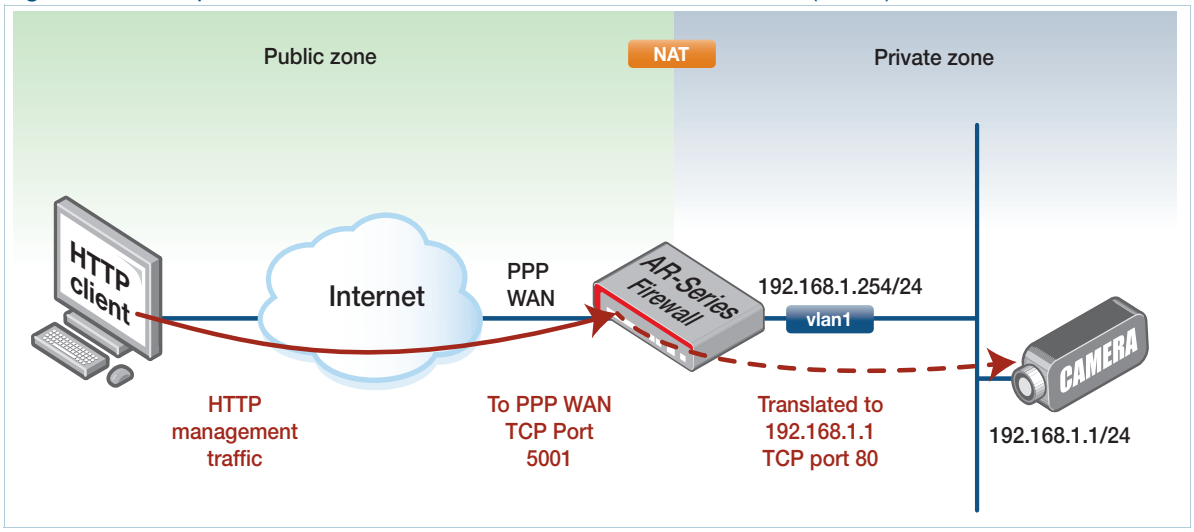
This example shows how to configure an AlliedWare Plus firewall to perform Network Address and Port Translation (NAPT).

In this example:

- A public zone and a private zone are configured.
- The PPP WAN interface (whose public IP address is dynamically allocated) is located in the public zone.
- The vlan1 interface (statically configured with a private IP address) is located in the private zone.
- A webcam, located in the private zone, can be managed via HTTP from a host located in the Internet.

However, HTTP management traffic from a host sourced from the Internet arrives at the WAN interface of the AlliedWare Plus firewall, destined to a non-standard TCP listen port (5001).

Figure 18: Example network: Network Address and Port Translation (NAPT)



To allow HTTP management traffic destined to the PPP WAN IP/non-default TCP port 5001 to reach the internal webcam, you need to:

- Configure a static application to match the TCP management traffic incoming from the Internet destined to the non-standard port number.
- Configure private and public zones.
- Then configure a NAT port-forwarding rule (rule 20 in Figure 19), to ensure the TCP application traffic destination IP address is translated to become the internal IP address of the webcam. Also, via the same rule, the destination port is translated to the internal HTTP port 80.

  Note that firewall rule actions are applied **after** any NAT port-forwarding is applied. Therefore, since the firewall is also used, the firewall rule (rule 20) is configured to permit the post-NAPT

translated HTTP application traffic sourced from the Internet to reach the internal webcam private IP address.

Figure 19: Configuration for NAPT example

```
!
application webcam_management
 protocol tcp
 dport 5001
!
zone public
 network internet
  ip subnet 0.0.0.0/0 interface ppp1
  host router_wan
    ip address dynamic interface ppp1
!
zone private
 network lan
  ip subnet 192.168.1.0/24
  host webcam
   ip address 192.168.1.1
!
firewall
 rule 10 permit any from private to public
 rule 20 permit http from public to dst private.lan.webcam
protect

!
nat
 rule 10 masq any from private to public
 rule 20 portfwd webcam_management from public.internet to
public.internet.router_wan with dst private.lan.webcam dport 80
 enable
!
interface vlan1
 ip address 192.168.1.254/24
!
interface ppp1
 ip address negotiated
!
ip route 0.0.0.0/0 ppp1
!
```

Configuring subnet-based NAT

Subnet-based NAT is supported from 5.4.7-0.1.

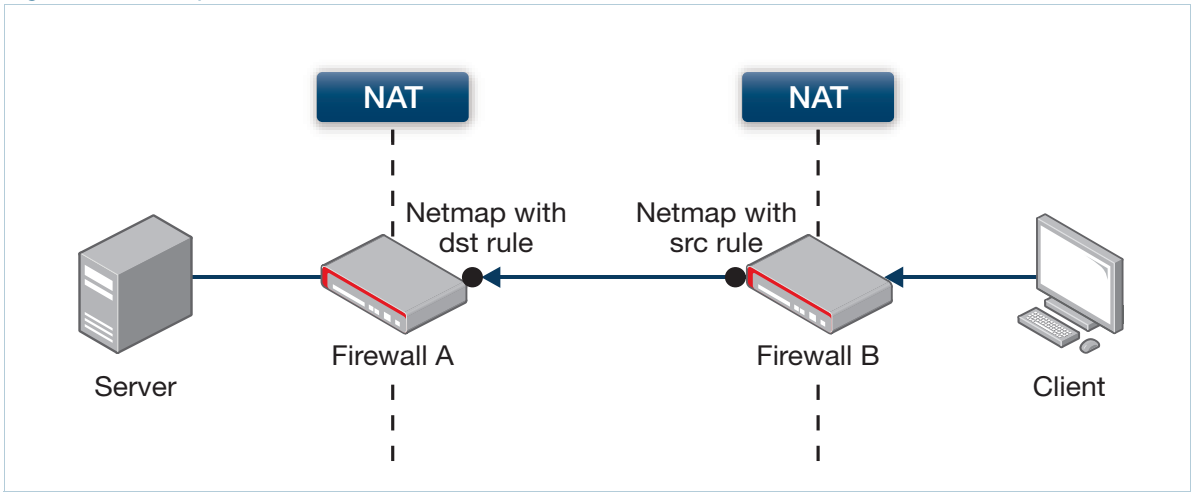
Subnet-based NAT translates just the network portion of a packet's source or destination IP address to a different network address—the host portion of the address is unchanged. There is a one-to-one mapping from addresses in one subnet to the other. Subnet-based NAT allows a user to perform NAT translation 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 20: 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 68).

Figure 21: Example: subnet NAT configuration for 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 public.wan2 to public.wan1 with dst private.lan
 enable
```

Figure 22: Example: subnet-based NAT configuration for 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.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"]
------------------------------------------------------------------------
  ID    Action      From              With (dst/src) Entity  Hits
        App         To                With dport
------------------------------------------------------------------------
  10    netmap-src  private.lan       public.wan2            1
        any         public.wan1
```

```
Firewall-A#show nat rule
[* = Rule is not valid - see "show nat rule config-check"]
-----------------------------------------------------------------------
  ID    Action     From                With (dst/src) Entity   Hits
        App        To                  With dport
-----------------------------------------------------------------------
  10   netmap-dst  public.wan2         private.lan             1
       any         public.wan1         -
```

```
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
```

Bi-directional configuration for subnet NAT

The following two configurations include a second rule to allow bi-directional translation, so that traffic can be initiated from either end.

Figure 23: 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 24: 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.wan1 with src public.wan2
 rule 20 netmap any from public.wan1 to public.wan2 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 AlliedWare Plus 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 can not 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 25: Example: partial sessions through firewall

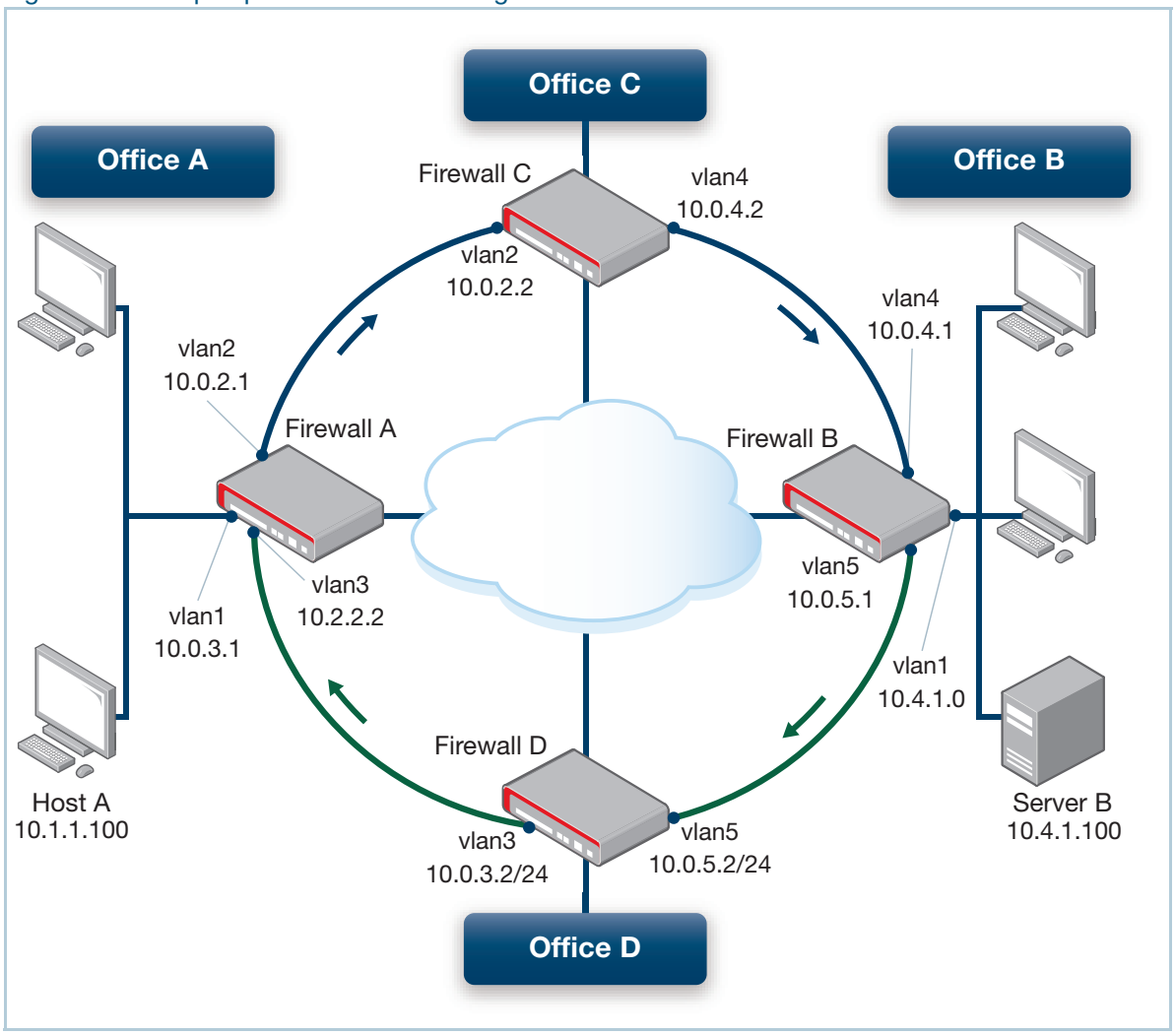


Figure 26: Example: partial sessions through firewall—configuration for Firewall C

```
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
```

Figure 27: Example: partial sessions through firewall—configuration for Firewall D

```
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
 protect
!
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
```

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.
6. Server B responds with a TCP SYN/ACK from 10.4.1.100:80 to 10.1.1.100:1024.
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.
12. Server B responds with an ACK and HTTP response.
13. Firewall D permits this as a connection match for the traffic flow that was permitted by rule 10 in step 8.

Command summary

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.

The following configuration extract illustrates these points:

```
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 and Firewall-D is as follows:

```
Firewall-C#show firewall rule

[* = Rule is not valid - see "show firewall rule config-check"]
  ID     Action  App       From            To                  Hits
-------------------------------------------------------------------
  10     permit  any       Transit         Transit             10
```

```
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, Firewall-D shows:

```
Firewall-D#show firewall rule


[* = Rule is not valid - see "show firewall rule config-check"]
  ID    Action   App       From          To                  Hits
-------------------------------------------------------------------
  10   permit    any       Transit       Transit             4
```

```
Firewall-D#show firewall connections
tcp 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
```