Technical Guide

OpenVPN

Feature Overview and Configuration Guide

Introduction

This guide describes AlliedWare Plus™ OpenVPN and its configuration.

AlliedWare Plus OpenVPN provides a seamless, secure and easy means for employees to have access to the same resources whether they are inside or outside their company premises. Staff members have the ability to work securely from remote locations such as from home or when on business trips.

Products and software version that apply to this guide

This guide applies to AlliedWare Plus products that support OpenVPN, running version **5.4.5** or later.

To see whether a product supports OpenVPN, see the following documents:

- The product's Datasheet
- The AlliedWare Plus Datasheet
- The product's Command Reference

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

Feature support may change in later software versions. For the latest information, see the above documents.

Contents

What is OpenVPN?

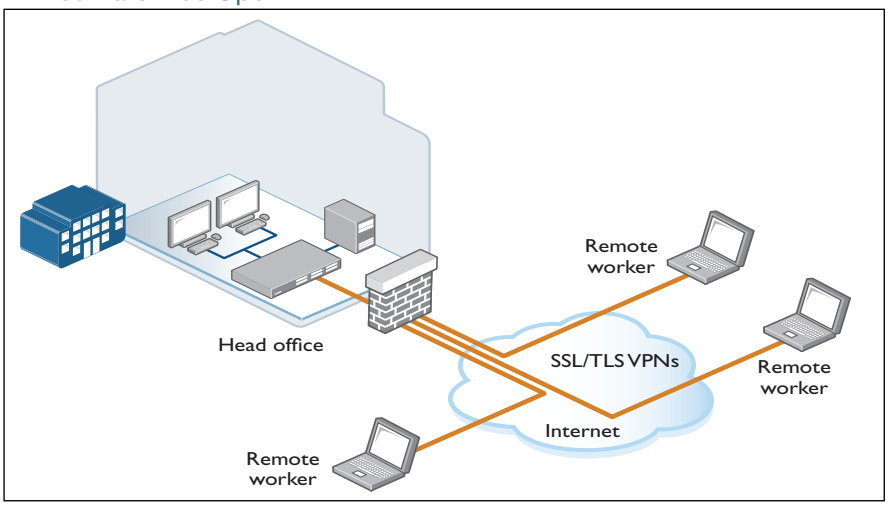
AlliedWare Plus OpenVPN is an SSL/TLS-based application used for creating a secure connection from a remote client to a central site. It establishes an encrypted and authenticated tunnel between the client and server and uses that tunnel for transporting traffic across intervening networks. AlliedWare Plus OpenVPN provides a full Data Link Layer access, proven standards-based SSL/TLS authentication and encryption, and implicit firewall/NAT traversal.

AlliedWare Plus OpenVPN is built on a solid and industry-tested security foundation and is very easy to use. It offers you the flexibility to work in a variety of modes that are easy to understand and hard to make insecure.

AlliedWare Plus OpenVPN provides the following key features:

- Protection of IPv4 and IPv6 traffic over TLS tunnel
- Configurable listening UDP port
- A maximum of 100 to 200 OpenVPN clients can concurrently connect, depending on the AR-series firewall VPN router model. See your product's Datasheet for more information.
- Group network access control based on 802.1Q tagged interfaces allows one or more clients to be associated with up to 64 VLANs
- Server authentication using certificates, client authentication via RADIUS over IPv4/IPv6
- Virtual Tunnel Interface for OpenVPN tunnels
- Single OpenVPN tunnel interface
- IPv4 and IPv6 as a delivery protocol
- Support for:
  - TAP mode (Layer 2 based Ethernet TAP) and
  - TUN mode (Layer 3 based IP tunnel)

Figure 1: AlliedWare Plus OpenVPN



About OpenVPN TAP mode

The purpose of TAP mode is to enable the remote client to operate as though it were directly connected to the LAN that lies behind the OpenVPN server.

Effectively the OpenVPN connection operates as though it were a virtual NIC card in the client, connected to the LAN in behind the OpenVPN server. So, the OpenVPN connection operates like a Network Tap that sits on that central LAN. It transfers packets from that LAN over a tunnel to the remote client (and transports packets back from the remote client).

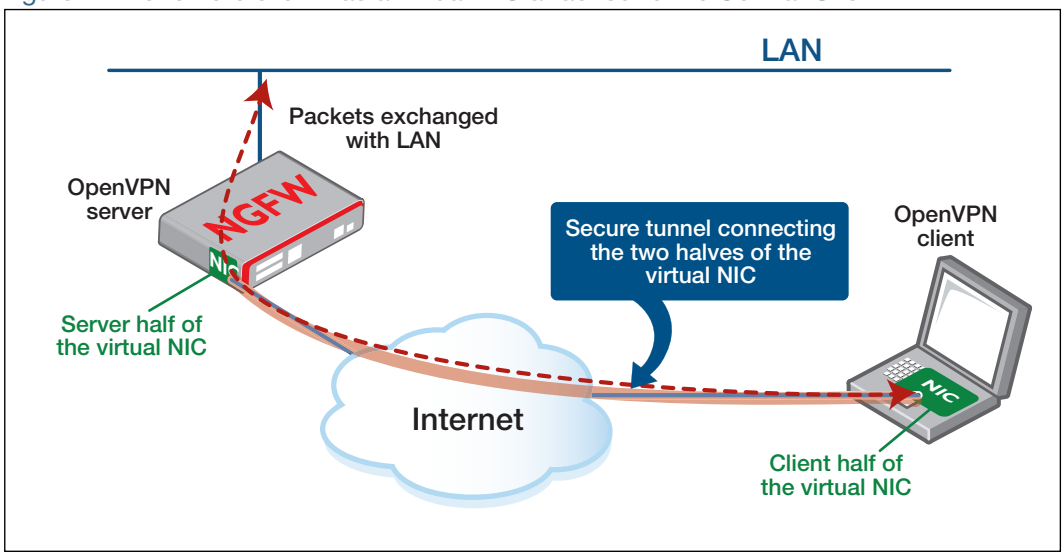
The full content of the Ethernet frames to/from the client are encapsulated in the tunnel and transported between the client and the server halves of the virtual NIC. Therefore, it appears to the client that it has received the frames directly off the central-site LAN, and appears to the central-site LAN that the client is directly connected to that LAN.

Within the OpenVPN server, the TAP appears as a Virtual Tunnel Interface (VTI) that carries Layer 2 frames.

Note: Because the TAP encapsulates the full Ethernet frames, it can be used for transporting protocols other then IP, for example IPX or AppleTalk. It also means that any communication that relies on the exchange of broadcast or multicast packets will work seamlessly. In particular, the remote client will be able to obtain an IP address by DHCP from a DHCP server on the central LAN.

As well as enabling a remote client to appear to be connected to the central LAN, TAP mode can also be used to create a bridge connection to unite two LANs that are at separate locations.

Figure 2: The remote client has a virtual NIC attached to the Central-Site LAN



Because the TAP acts as a NIC attached to the central LAN, it will transport all that LAN's broadcast frames across the tunnel. This potentially adds extra traffic to the tunnel, transporting broadcasts that the remote client may not even be interested in. The TAP also adds the overhead of Ethernet headers on all packets transported over the VPN tunnel.

About OpenVPN TUN mode

TUN is also a virtual network device. TUN creates a Virtual Tunnel Interface (VTI) that carries Layer 3 packets. So, rather than encapsulating the full Ethernet frames, it takes the IP content of the frames, and routes that content via the tunnel. You can also use TUN to:

- transport traffic that is destined for the VPN client
- transport only Layer 3 packets
- support VPN on mobile devices.

Note: TUN cannot be used in bridges, and broadcast traffic is not transported in TUN mode.

So, with TUN mode, the VPN connection appears as an IP interface on the remote client, and the AR-Series Firewall acts as the Gateway for routing the VPN traffic to other networks.

RADIUS attributes supported by OpenVPN

When RADIUS is used for client authentication, there are several attributes that can be configured on the RADIUS server for each user. These attributes provide a mechanism for shaping the remote user's network configuration when accessing the network via VPN so that they have a similar connectivity experience as they would have if directly connected to the central site LAN.

The following attributes are supported by OpenVPN:

ID	ATTRIBUTE	TYPE	SPECIFICATION	EXAMPLE	USAGE
1	User-Name	string	RFC2865	"foo"	Client username
2	Password	string	RFC2865	"bar"	Client password
6	Service-Type	integer	RFC2865	8 = Authenticate Only	OpenVPN requests login only to the RADIUS server
8	Framed-IP-Address	ipaddr	RFC2865	10.10.10.50	IP address to be pushed to the client
9	Framed-IP-Netmask	ipaddr	RFC2865	255.255.255.0	IP netmask to be pushed to the client
22	Framed-Route	string	RFC2865	"10.10.11.0/8 10.10.10.1 1"	Route to be pushed to the client
MS-28	Microsoft-Primary-DNS-Server	ipaddr	RFC2548	10.10.10.1	Primary DNS to push to client (if multiple primary DNS servers are provided, only the first one will be used.)
MS-29	Microsoft-Secondary-DNS-Server	ipaddr	RFC2548	10.10.10.2	Secondary DNS to push to client (if no primary address provided, this will be ignored.)
97	Framed-IPv6-Prefix	ipv6prefix	RFC3162	"fc00:2::2/64"	IPv6 prefix to be pushed to the client
169	DNS-Server-IPv6-Address	ipv6addr	RFC6911	"fc00:2::1“	IPv6 DNS address to be pushed to the client (without NH)

ID	ATTRIBUTE	TYPE	SPECIFICATION	EXAMPLE	USAGE
170	Route-IPv6-Information	ipv6prefix	RFC6911	"fc00:3::/64“	IPv6 route to be pushed to the client
64	Tunnel-Type	integer	RFC3580	13 = VLAN	Client VLAN assignment. Tag the client traffic if 802.1Q tagging is configured (TAP mode only).
65	Tunnel-Medium-Type	integer	RFC3580	6 = 802	Client VLAN assignment. Tag the client traffic if 802.1Q tagging is configured (TAP mode only).
81	Tunnel-Private-Group-Id	string	RFC3580	"20" = VLANID 20	Client VLAN assignment. Tag the client traffic if 802.1Q tagging is configured (TAP mode only).

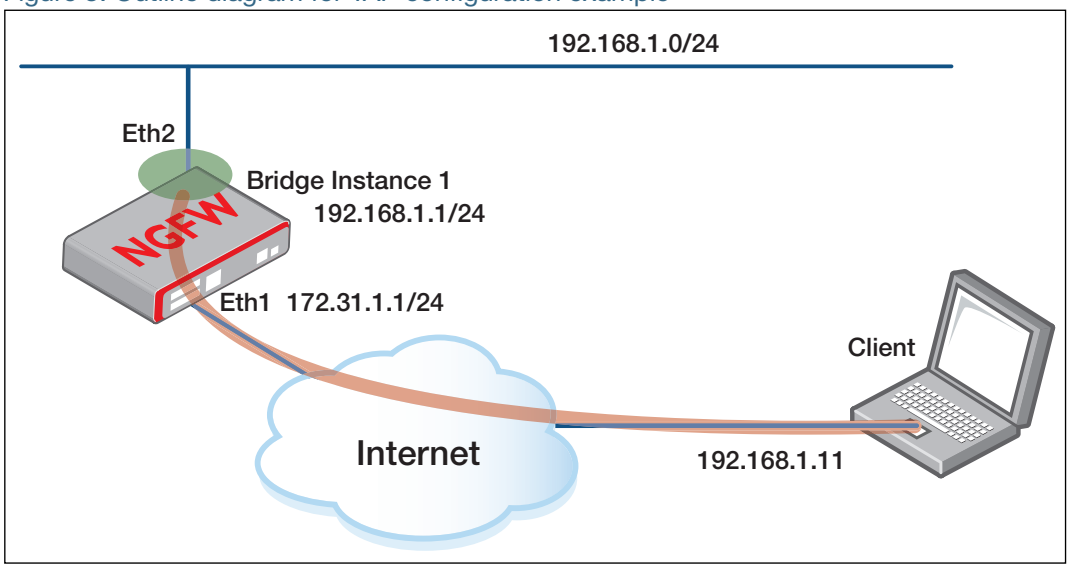
Configuration Examples

OpenVPN supports remote access from multiple operating systems and mobile devices, which means you can have remote access to the company internal network. For more information about how to configure OpenVPN on the client device, visit OpenVPN.

The following examples show how to configure both OpenVPN TAP service and TUN service.

Example 1: Configuring OpenVPN TAP service

Figure 3: Outline diagram for TAP configuration example



Step 1. Configure local RADIUS server for OpenVPN TAP mode

```
awplus#configure terminal
```

- Specify a local RADIUS server host (IP address 127.0.0.1 indicates that the device itself is operating as the RADIUS server) and set parameters for the server.

```
awplus(config)#radius-server host 127.0.0.1 key awplus-local-radius-server
awplus(config)#aaa authentication openvpn default group radius
```

- Enter the local RADIUS server configuration mode.

```
awplus(config)#radius-server local
```

- Configure the client user group and configure the client IP address. Note that this step is optional for configuring OpenVPN TAP mode.

```
awplus(config-radsrv)#group client
```

- Configure the client user IP address. If you want to support more client users, you need to create a group for each client user. Note that if you want to configure the client IP address/mask with the RADIUS server, then this step is required. If you don't want to configure the client IP address/mask with the RADIUS server, then this step is not required and you can configure the client IP address via DHCP.

```
awplus(config-radsrv-group)#attribute Framed-IP-Address
192.168.1.11
```

- Configure the IP subnet mask of the tunnel interface. Note that if you want to configure the client IP address/mask with the RADIUS server, then this step is required. If you don't want to configure the client IP address/mask with the RADIUS server, then this step is not required and you can configure the client IP address via DHCP.

```
awplus(config-radsrv-group)#attribute Framed-IP-Netmask
255.255.255.0
```

- **Optional:** Configure the route for packets routing from network 192.168.0.0/16 to the remote network through the tunnel with 192.168.1.1 being the IP address of the remote tunnel interface.

```
awplus(config-radsrv-group)#attribute Framed-Route
"192.168.0.0/16 192.168.1.1"
```

- Return to the local RADIUS server configuration mode.

```
awplus(config-radsrv-group)#exit
```

- Add the NAS with an IP address to the list of clients that may send authentication requests to the local RADIUS server. In this case, the NAS is the device itself, so the NAS address is 127.0.0.1.

```
awplus(config-radsrv)#nas 127.0.0.1 key awplus-local-radius-
server
```

- Add a user to the RADIUS server database and specify the user name and password.

```
awplus(config-radsrv)#user remote password very_secret group
client
```

- Enable the local RADIUS server.

```
awplus(config-radsrv)#server enable
```

Step 2. **Configure server authentication**

- Declare local CA (Certificate Authority) as the trust point that the system uses.

```
awplus(config)#crypto pki trustpoint local
```

- Obtain a system certificate from local CA.

```
awplus(config)#crypto pki enroll local
```

- Export this CA public certificate, so the VPN client can use it to verify the Computer Certificate of the VPN router. This generates a file named **cacert.pem** on the flash file system (this file will be used in this example). This command is entered from Privilege Exec mode, and so does not appear in the device running configuration. This certificate

can also optionally be exported to the CLI terminal window directly, or to an external TFTP server directory.

```
awplus#crypto pki export local pem cacert.pem
```

Step 3. Configure the interface connecting the device to the Internet

```
awplus(config)#interface eth1
awplus(config-if)#ip address 172.31.1.1/24
```

Step 4. Enable OpenVPN TAP service

- Create a virtual tunnel interface (VTI) for the OpenVPN router to be accessed by the client.

```
awplus(config-if)#interface tunnel1
awplus(config-if)#tunnel mode openvpn tap
awplus(config-if)#exit
```

Step 5. Connect OpenVPN clients to the LAN

- Create a virtual Ethernet bridge to connect the VPN clients to the LAN.

```
awplus(config)#bridge 1
```

This newly created bridge will have two ports. One is the physical port eth2 that is connected to the LAN network. The other is the tunnel interface where the virtual OpenVPN TAP NIC will connect to.

- Assign eth2 and tunnel2 to the bridge.

```
awplus(config)#interface eth2
awplus(config-if)#bridge-group 1
awplus(config)#interface tunnel1
awplus(config-if)#bridge-group 1
```

Step 6. Enable other traffic to be routed to the Internet

Not all the traffic that enters the eth2 interface of the AR-series firewall is destined to go to the OpenVPN clients. Much of the traffic is destined to simply be routed to the Internet. So, we need a method to route traffic out of the bridge instance and deliver it through eth1 to the Internet.

- To do this, we need to configure an IP address on the bridge instance.

```
awplus(config)#interface br1
awplus(config-if)# ip address 192.168.1.1/24
```

The logic for routing packets from the LAN to the Internet is as follows:

- When packets enter the AR-Series Firewall via interface ETH2, they are deemed to have entered Bridge Instance 1.
- If the destination IP address of the packet is not within the subnet of Bridge Instance 1 (192.168.1.0/24), then the packet needs to be routed out of the bridge instance.

Configuring OpenVPN client for TAP service

Several OpenVPN clients are available for many platforms. Most have in common that they rely on a **.ovpn file**. Once the .ovpn file is created, client configuration is typically a matter of loading the file. This file was tested with OpenVPN 2.3 but should work with OpenVPN 2.1 or newer clients.

OpenVPN TAP mode client **.ovpn** config file

```
#Configure for client mode
client
#The server requires the client to provide a username/password for
#authentication.
auth-user-pass
#Require encryption
cipher AES-128-CBC
#Configure for TAP mode
dev tap
proto udp
#The address of the OpenVPN router to connect to
remote 172.31.1.1
```

Then the content of the **cacert.pem** flash file generated previously at step 2 is pasted into the **.ovpn** config file, with a header and footer around it, as shown below.

Samples of **.ovpn** file templates can be found on the OpenVPN website. These sample templates typically include explanations of the various **.ovpn** file configuration options, advice on default settings, and also show locations of where to paste the **cacert.pem** content.

Contents of the **cacert.pem** file

```
-----BEGIN CERTIFICATE-----
MIICXDCCAcWgAwIBAgIBADANBgkqhkiG9w0BAQUFADA0MRcwFQYDVQQKEw5BbGxp
ZWQtVGVsZXNpczEZMBcGA1UEAxMQQWxsaWVkd2FyZVBsdXNDQTAeFw0xNTA2MTkx
MDQxMzZaFw0zNTA2MTQxMDQxMzZaMDQxFzAVBgNVBAoTDkFsbGllZC1UZWxlc2lz
MRkwFwYDVQQDExBBbGxpZWR3YXJlUGx1c0NBMIGfMA0GCSqGSIb3DQEBAQUAA4GN
ADCBiQKBgQCywR9mBFrxnIVw577vtmvui5RnTBjLMJWkqF6eOhvt1Rzw9H8SZWPI
zjLs+Z21b5Nsc8YjB0kLcISu3lfZrxtIPlkEm81U8mImHJnTBAmkHUZi5fBJbH12
KG7rUZ/Zxq591+vatwabyRiDIPEeis/aa1wEFm03uBc21NMsQYENiwIDAQABo34w
fDAMBgNVHRMEBTADAQH/MCwGCWCGSAGG+EIBDQQfFh1PcGVuU1NMIEdlbmVyYXRl
ZCBDZXJ0aWZpY2F0ZTAdBgNVHQ4EFgQUXJbOiME3bd2KhOLH8D25nt7xQ6UwHwYD
VR0jBBgwFoAUXJbOiME3bd2KhOLH8D25nt7xQ6UwDQYJKoZIhvcNAQEFBQADgYEA
YZG8DWKYj8rYkkvbZ1/ZhsLzubAH1ggUueS1eCzanh9zi5o92+pwOTPgi/g3JnDe
cpw06L6kvgTItZxwa32whO2RjUzzhCLegc9DKyEixmAZ6bUd29Idsn1DpGHEgPXT
QgSTQrb4HcOAlbrG7Eh/IlRN1ByE4QAUNaP6N84nZwo=
-----END CERTIFICATE-----
```

This **.ovpn** file can be used by all clients. The individual clients use the username and password to authenticate themselves. OpenVPN client applications typically prompt the user to enter the username and password when connecting, and store this for subsequent connections.

The following command can be used to display the start and end dates of the certificates on the device:

Certificate expiry information

```
awplus#show crypto pki certificates local

--------------------
Trustpoint "local" Certificate Chain
--------------------
Self-signed root certificate
   Subject     : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAA05050G152000036
   Issuer      : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAA05050G152000036
   Valid From  : Jun 11 10:41:50 2016 GMT
   Valid To    : Jun  9 10:41:50 2026 GMT
   Fingerprint : 5C6A616A 3A28699A D24156E5 505E4CE7 2B109D0D
```

Example 2: Configuring OpenVPN TUN service

Configuring the router for OpenVPN TUN service

Step 1. Configure the local RADIUS server for OpenVPN TUN mode

```
awplus#configure terminal
```

- Declare local CA (Certificate Authority) as the trust point that the system uses.

```
awplus(config)#crypto pki trustpoint local
```

- Obtain a system certificate from local CA.

```
awplus(config)#crypto pki enroll local
```

- Enter the local RADIUS server configuration mode.

```
awplus(config)#radius-server local
```

- Configure client user group and configure client IP address.

```
awplus(config-radsrv)#group client
```

- Configure client user IP address. If you want to support more client users, you need to create a group for each client user.

```
awplus(config-radsrv-group)#attribute Framed-IP-Address
192.168.2.11
```

- Configure IP subnet mask of the tunnel interface.

```
awplus(config-radsrv-group)#attribute Framed-IP-Netmask
255.255.255.0
```

- **Optional**: Configure the route for packets routing from network 192.168.0.0/16 to the remote network through the tunnel with 192.168.2.1 being the IP address of the remote tunnel interface.

```
awplus(config-radsrv-group)#attribute Framed-Route
"192.168.0.0/16 192.168.2.1"
```

- Return to the local RADIUS server configuration mode.

```
awplus(config-radsrv-group)#exit
```

- Add the NAS with an IP address to the list of clients that may send authentication requests to the local RADIUS server. In this case, the NAS is the switch itself, so the NAS address is 127.0.0.1.

```
awplus(config-radsrv)#nas 127.0.0.1 key awplus-local-radius-
server
```

- Add a user to the RADIUS server database and specify the user name and password.

```
awplus(config-radsrv)#user remote password very_secret group
client
```

- Enable local RADIUS server.

```
awplus(config-radsrv)#server enable
awplus(config-radsrv)#exit
```

Step 2. Configure the OpenVPN to authenticate using RADIUS

- Specify a local RADIUS server host (IP address 127.0.0.1 indicates that the switch itself is operating as the RADIUS server) and set parameters for the server.

```
awplus(config)#radius-server host 127.0.0.1 key awplus-local-radius-server
awplus(config)#aaa authentication openvpn default group radius
```

Step 3. Configure tunnel interface

- Create a tunnel interface.

```
awplus(config-if)#interface tunnel20
awplus(config-if)#tunnel mode openvpn tun
```

- Configure an IP address for the tunnel interface.

```
awplus(config-if)#ip address 192.168.2.1/24
```

Step 4. Configure other interfaces

```
awplus(config)#interface eth1
awplus(config-if)#ip address 172.31.1.1/24
awplus(config)#interface eth2
awplus(config-if)#ip address 192.168.1.1/24
```

Configuring OpenVPN client for TUN service

Several OpenVPN clients are available for many platforms. Most have in common that they rely on a **.ovpn-file**. Once the .ovpn file is created client configuration is typically a matter of loading the file.

- Each OpenVPN client logs in using a unique user name and password, and an Open VPN client can be configured to use the following .ovpn config file and associated CA certificate.
- Some OpenVPN clients are able to reference the CA certificate file directly without having to paste in the certificate information into the .ovpn file template as below.

This file tested with OpenVPN 2.3 but should work with OpenVPN 2.1 or newer clients.

OpenVPN TUN mode client **.ovpn** config file

```
remote  172.31.1.1 1194 udp
pull
tls-client
cipher AES-128-CBC
auth SHA1
tls-cipher TLS-DHE-RSA-WITH-AES-256-CBC-SHA
auth-user-pass
ca cacert.pem
dev-type tun
topology subnet
port 1194
verb 7
```

- The file **cacert.pem** referred to in the **.ovpn** file is created on the AR-series firewall by the command **crypto pki export local pem cacert.pem**.

Example export command

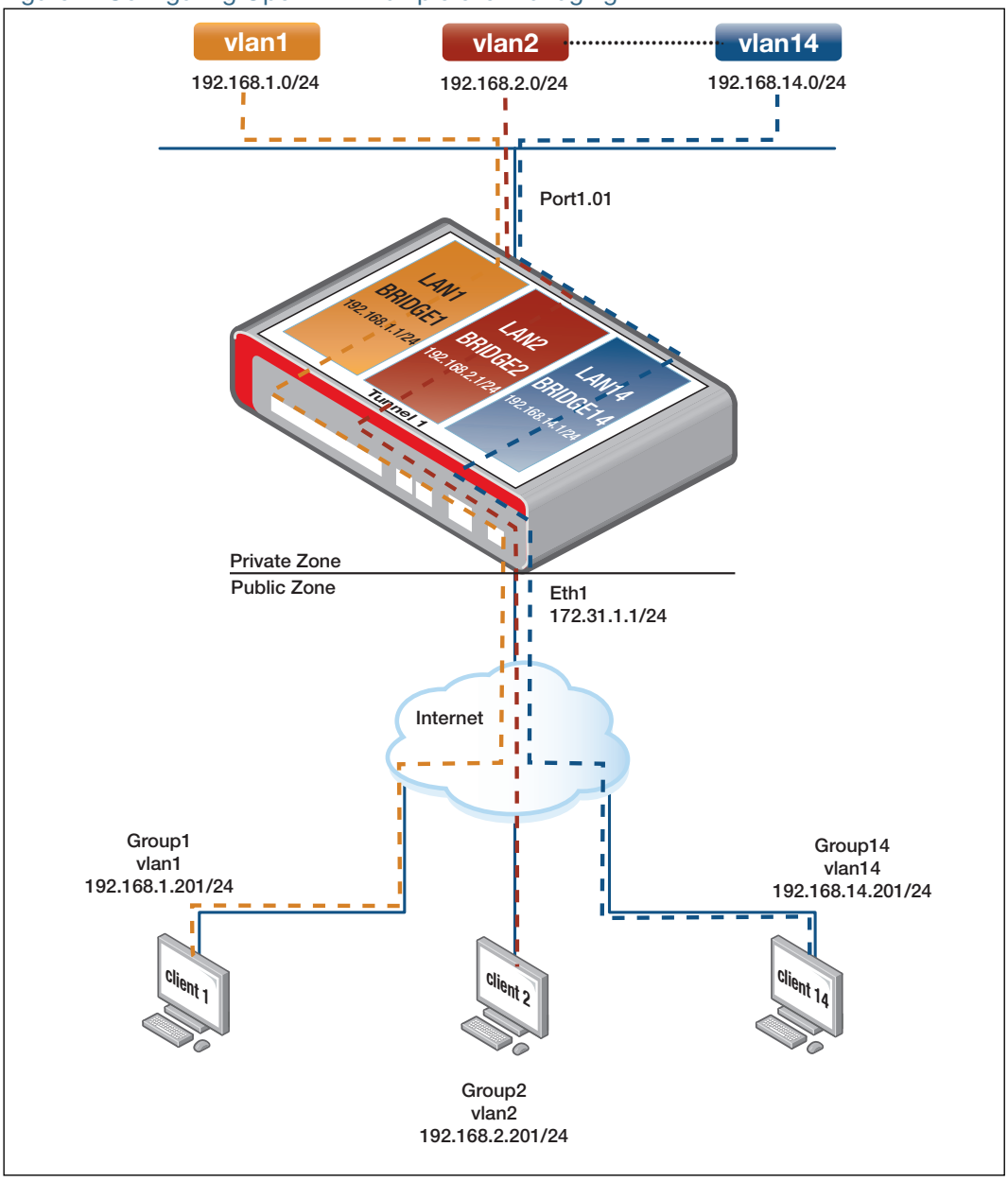
```
awplus#crypto pki export local pem cacert.pem
Copying...
Successful operation
```

Then, this file is copied off the AR-Series Firewall, and copied onto the client device.

Example 3: Configuring OpenVPN multiple client bridging

The following example consists of multiple (14) remote OpenVPN clients connecting into a central AR-Series Firewall. In this example, the AR-Series Firewall is configured to bridge each incoming client VPN connection to a specific VLAN. Each bridged network is configured to be in its own unique firewall network entity inside the private zone. Traffic between each firewall network entity is blocked. Firewall rules are configured to allow each client to access only the specific VLAN that they are a member of.

Figure 4: Configuring OpenVPN multiple client bridging



AR-Series Firewall configuration

Step 1. Local radius server and CA configuration

- Configure the local radius server.
- Declare local CA (Certificate Authority) as the trust point that the system uses, and obtain a system certificate from local CA.

Local radius server and CA configuration

```
radius-server host 127.0.0.1 key radius
!
aaa authentication openvpn default group radius
!
crypto pki trustpoint local
!
crypto pki enroll local
radius-server local
 server enable
 nas 127.0.0.1 key radius
```

Step 2. Client user group database configuration

- Configure a client user group database, with each group entry consisting of an IP address and subnet mask. An 802.1q VLAN tag is associated with each group, and each client is allocated an IP address and subnet mask for the specific network that they need to access.

Client user group database configuration

```
group group1
 vlan 1
 attribute Framed-IP-Address 192.168.1.201
 attribute Framed-IP-Netmask 255.255.255.0
group group2
 vlan 2
 attribute Framed-IP-Address 192.168.2.201
 attribute Framed-IP-Netmask 255.255.255.0
group group3
 vlan 3
 attribute Framed-IP-Address 192.168.3.201
 attribute Framed-IP-Netmask 255.255.255.0
group group4
 vlan 4
 attribute Framed-IP-Address 192.168.4.201
 attribute Framed-IP-Netmask 255.255.255.0
group group5
 vlan 5
 attribute Framed-IP-Address 192.168.5.201
 attribute Framed-IP-Netmask 255.255.255.0
group group6
 vlan 6
 attribute Framed-IP-Address 192.168.6.201
 attribute Framed-IP-Netmask 255.255.255.0
```

```
group group7
 vlan 7
 attribute Framed-IP-Address 192.168.7.201
 attribute Framed-IP-Netmask 255.255.255.0
group group8
 vlan 8
 attribute Framed-IP-Address 192.168.8.201
 attribute Framed-IP-Netmask 255.255.255.0
group group9
 vlan 9
attribute Framed-IP-Address 192.168.9.201
 attribute Framed-IP-Netmask 255.255.255.0
group group10
 vlan 10
 attribute Framed-IP-Address 192.168.10.201
 attribute Framed-IP-Netmask 255.255.255.0
group group11
 vlan 11
 attribute Framed-IP-Address 192.168.11.201
 attribute Framed-IP-Netmask 255.255.255.0
group group12
 vlan 12
 attribute Framed-IP-Address 192.168.12.201
 attribute Framed-IP-Netmask 255.255.255.0
group group13
 vlan 13
 attribute Framed-IP-Address 192.168.13.201
 attribute Framed-IP-Netmask 255.255.255.0
group group14
 vlan 14
 attribute Framed-IP-Address 192.168.14.201
 attribute Framed-IP-Netmask 255.255.255.0
```

Step 3. **Radius server user database authentication configuration**

- Configure the Radius Server database to authenticate each incoming OpenVPN client connection, and associate each client to their appropriate client user group.

Radius server user database authentication configuration

```
user user1 encrypted password <password1> group group1
user user2 encrypted password <password2> group group2
user user3 encrypted password <password3> group group3
user user4 encrypted password <password4> group group4
user user5 encrypted password <password5> group group5
user user6 encrypted password <password6> group group6
user user7 encrypted password <password7> group group7
user user8 encrypted password <password8> group group8
user user9 encrypted password <password9> group group9
user user10 encrypted password <password10> group group10
user user11 encrypted password <password11> group group11
user user12 encrypted password <password12> group group12
user user13 encrypted password <password13> group group13
user user14 encrypted password <password14> group group14
```

Step 4. **Firewall zone and network entity configuration**

- Configure a named private firewall zone entity and an associated network entity for each subnet, and configure a public zone entity for the WAN connection to the Internet via eth1.

Firewall zone and network entity configuration

```
zone private
 network lan1
  ip subnet 192.168.1.0/24!
 network lan2
  ip subnet 192.168.2.0/24
 network lan3
  ip subnet 192.168.3.0/24
 network lan4
  ip subnet 192.168.4.0/24
 network lan5
  ip subnet 192.168.5.0/24
 network lan6
  ip subnet 192.168.6.0/24
 network lan7
  ip subnet 192.168.7.0/24
 network lan8
  ip subnet 192.168.8.0/24
 network lan9
  ip subnet 192.168.9.0/24
 network lan10
  ip subnet 192.168.10.0/24
 network lan11
  ip subnet 192.168.11.0/24
 network lan12
  ip subnet 192.168.12.0/24
 network lan13
  ip subnet 192.168.13.0/24
 network lan14
  ip subnet 192.168.14.0/24
!
zone public
 network all
  ip subnet 0.0.0.0/0 interface eth1
 network interface
  ip subnet 172.31.1.0/24
 host router
  ip address 172.31.1.1
```

Step 5. **Firewall OpenVPN application configuration**

- Configure application for OpenVPN to be passed by the firewall.

Firewall OpenVPN application configuration

```
application openvpn
 protocol udp
 dport 1194
```

Step 6. **Firewall rules configuration**

When the firewall is enabled, all traffic is then blocked by default, so firewall rules need to configured to allow specific application traffic to pass through the firewall.

- Configure a firewall rule to allow traffic from the private firewall zone to access the public internet.
- Configure firewall rules to allow traffic from each OpenVPN client to access their specific named LAN network entity.
- Configure a firewall rule to allow incoming OpenVPN traffic to pass through the public interface.

Firewall rules configuration

```
firewall
 rule 100 permit any from private to public
 rule 110 permit any from private.lan1 to private.lan1
 rule 210 permit any from private.lan2 to private.lan2
 rule 310 permit any from private.lan3 to private.lan3
 rule 410 permit any from private.lan4 to private.lan4
 rule 510 permit any from private.lan5 to private.lan5
 rule 610 permit any from private.lan6 to private.lan6
 rule 710 permit any from private.lan7 to private.lan7
 rule 810 permit any from private.lan8 to private.lan8
 rule 910 permit any from private.lan9 to private.lan9
 rule 1010 permit any from private.lan10 to private.lan10
 rule 1110 permit any from private.lan11 to private.lan11
 rule 1210 permit any from private.lan12 to private.lan12
 rule 1310 permit any from private.lan13 to private.lan13
 rule 1410 permit any from private.lan14 to private.lan14
 rule 1500 permit openvpn from public to public.interface.router
protect
```

Step 7. **Firewall NAT rules configuration**

- Configure a firewall NAT masquerade rule to translate the source IP address of all traffic originating from the private zone destined to the internet using the public IP address of eth1 WAN.

Firewall NAT rules configuration

```
nat
 rule 100 masq any from private to public
enable
```

Step 8. **VLAN database configuration**

- Configure the VLAN database.

VLAN database configuration

```
vlan database
 vlan 2-14 state enable
```

Step 9. **Switchport configuration**

- In this example, switchport 1.0.1 is configured to be an 802.1q trunked member of the VLANs that OpenVPN clients will access. This port could be connected to a separate Layer2 access switch. In this example, the native VLAN has been removed from the switchport, so that only 802.1q VLAN tagged frames are accepted.

Switchport configuration

```
interface port1.0.1
 switchport mode trunk
 switchport trunk allowed vlan add 1-14
 switchport trunk native vlan none
```

Step 10. **WAN interface configuration**

- Configure public interface ethernet1 with the static ip address allocated by the Internet Service Provider.

WAN interface configuration

```
interface eth1
 ip address 172.31.1.1/24
```

Step 11. **VTI and VTI sub interfaces configuration**

- Configure Virtual Tunnel Interface (VTI) in OpenVPN Tap mode, and configure a series of sub interfaces with associated 802.1q VLAN ID encapsulation. OpenVPN is configured to use port number 1194.
- Each incoming VPN data stream is decrypted. The resulting Ethernet frames contain a source IP address and subnet mask that is matched against a specific client user group database entry. The 802.1q VLAN tag configured in the matching client user group entry is inserted into the decrypted Ethernet frames.
- This allows incoming decrypted 802.1q tagged Ethernet data streams to be forwarded to the appropriate VTI sub interface based on the matching 802.1q VLAN tags.

VTI and VTI sub interfaces configuration

```
interface tunnel1
 encapsulation dot1q 1
 encapsulation dot1q 2
 encapsulation dot1q 3
 encapsulation dot1q 4
 encapsulation dot1q 5
 encapsulation dot1q 6
 encapsulation dot1q 7
 encapsulation dot1q 8
 encapsulation dot1q 9
 encapsulation dot1q 10
 encapsulation dot1q 11
 encapsulation dot1q 12
 encapsulation dot1q 13
 encapsulation dot1q 14
 tunnel openvpn tagging 1
 tunnel openvpn port 1194
 tunnel mode openvpn tap
```

Step 12. **Bridge configuration**

- Configure bridges instances to allow each OpenVPN client to connect to their own unique bridged network.

Bridge configuration

```
bridge 1
bridge 2
bridge 3
bridge 4
bridge 5
bridge 6
bridge 7
bridge 8
bridge 9
bridge 10
bridge 11
bridge 12
bridge 13
bridge 14
```

Step 13. **Bridge IP address configuration**

- Each bridge instance is configured with an IP address within the network that each client connects to.

Bridge IP address configuration

```
interface br1
 ip address 192.168.1.1/24
!
interface br2
 ip address 192.168.2.1/24
!
interface br3
 ip address 192.168.3.1/24
!
interface br4
 ip address 192.168.4.1/24
!
interface br5
 ip address 192.168.5.1/24
!
interface br6
 ip address 192.168.6.1/24
!
interface br7
 ip address 192.168.7.1/24
!
interface br8
 ip address 192.168.8.1/24
!
interface br9
 ip address 192.168.9.1/24
!
interface br10
 ip address 192.168.10.1/24
```

```
!
interface br11
 ip address 192.168.11.1/24
!
interface br12
 ip address 192.168.12.1/24
!
interface br13
 ip address 192.168.13.1/24
!
interface br14
 ip address 192.168.14.1/24
```

Step 14. **Association of VLANs with bridges configuration**

- Associate each VLAN with a bridge instance.

Association of VLANs with bridges configuration

```
interface vlan1
 bridge-group 1
!
interface vlan2
 bridge-group 2
!
interface vlan3
 bridge-group 3
!
interface vlan4
 bridge-group 4
!
interface vlan5
 bridge-group 5
!
interface vlan6
 bridge-group 6
!
interface vlan7
 bridge-group 7
!
interface vlan8
 bridge-group 8
!
interface vlan9
 bridge-group 9
!
interface vlan10
 bridge-group 10
!
interface vlan11
 bridge-group 11
!
interface vlan12
 bridge-group 12
!
interface vlan13
 bridge-group 13
!
interface vlan14
 bridge-group 14
```

Step 15. **Association of VTI sub interfaces with bridges configuration**

- Each VTI sub interface is linked to the appropriate bridge group.
- This final step ensures each incoming OpenVPN client connection is bridged to the appropriate VLAN interface, allowing each client to access their respective networks.

Association of VTI sub interfaces with bridges Configuration

```
interface tunnel1.1
 bridge-group 1
!
interface tunnel1.2
 bridge-group 2
!
interface tunnel1.3
 bridge-group 3
!
interface tunnel1.4
 bridge-group 4
!
interface tunnel1.5
 bridge-group 5
!
interface tunnel1.6
 bridge-group 6
!
interface tunnel1.7
 bridge-group 7
!
interface tunnel1.8
 bridge-group 8
!
interface tunnel1.9
 bridge-group 9
!
interface tunnel1.10
 bridge-group 10
!
interface tunnel1.11
 bridge-group 11
!
interface tunnel1.12
 bridge-group 12
!
interface tunnel1.13
 bridge-group 13
!
interface tunnel1.14
 bridge-group 14
```

Configuring OpenVPN client for Bridge TAP service

- Each OpenVPN client logs in using a unique user name and password, and an Open VPN client can be configured to use the following **.ovpn** config file and associated CA certificate.
- Some OpenVPN clients are able to reference the CA certificate file directly without having to paste in the certificate information into the .ovpn file template as below.

Example **.ovpn** config file

```
# tun.ovpn
client
auth-user-pass
cipher AES-128-CBC
dev tap
proto udp
remote 172.31.1.1
ca c:/users/support/cacert.pem
verb 7
```

- The file **cacert.pem** referred to in the **.ovpn** file is created on the AR-series firewall by the command **crypto pki export local pem cacert.pem**.

Example **crypto pki export local pem cacert.pem** command

```
awplus#crypto pki export local pem cacert.pem
Copying...
Successful operation
```

Then, this file is copied off the AR-Series Firewall, and copied onto the client device.