AT&T VYATTA 5600 vROUTER SITE-TO-SITE VPNS USING IPSEC

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.

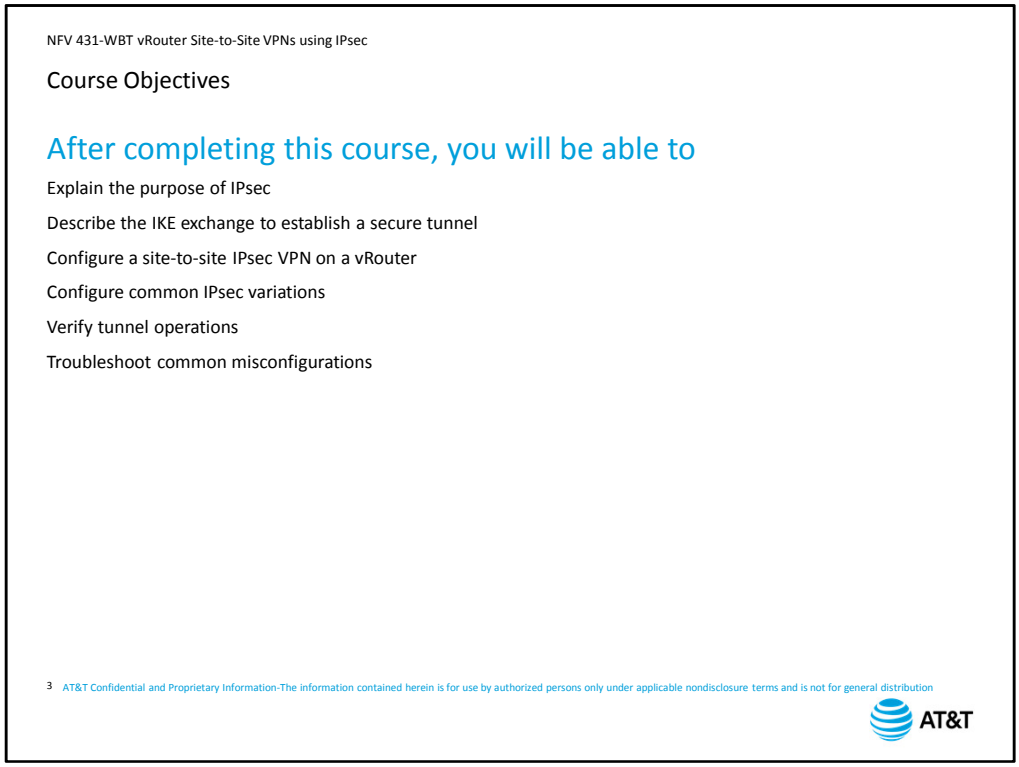
1

Welcome to the AT&T Vyatta 5600 vRouter Site-to-site VPNs Using IPsec course.

Legal Disclaimer

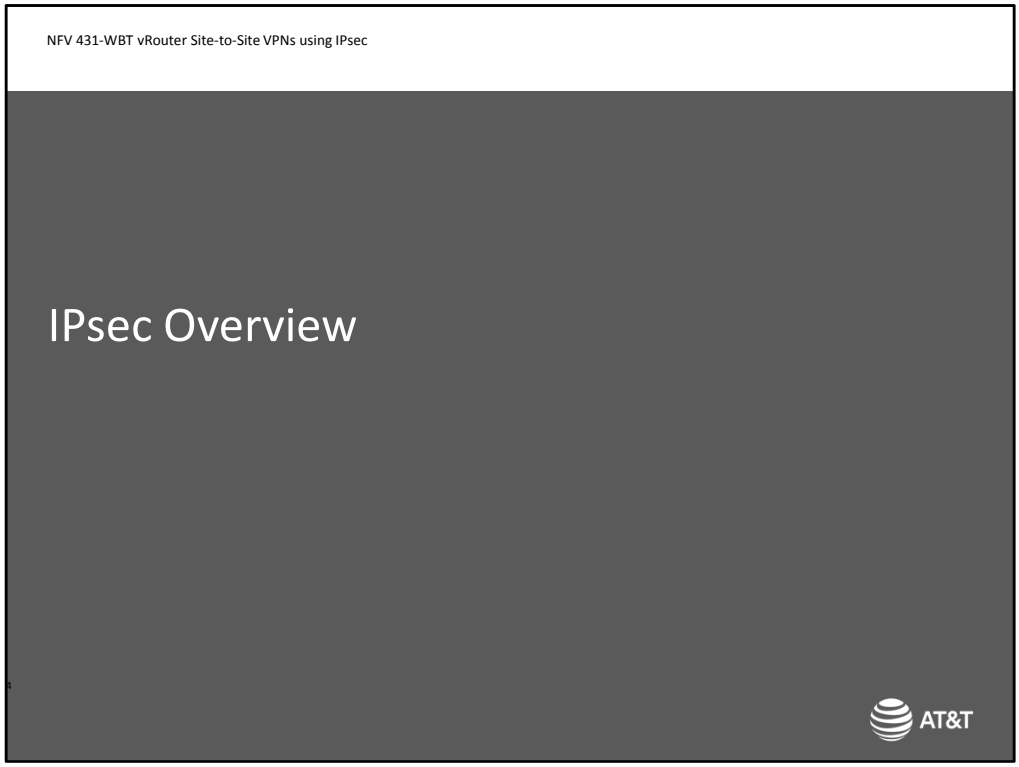
All or some of the products detailed in this presentation may still be under development and certain specifications may be subject to change. Nothing in this presentation shall be deemed to create a warranty of any kind.

Before we begin the course, please take a moment to read our legal disclaimer.

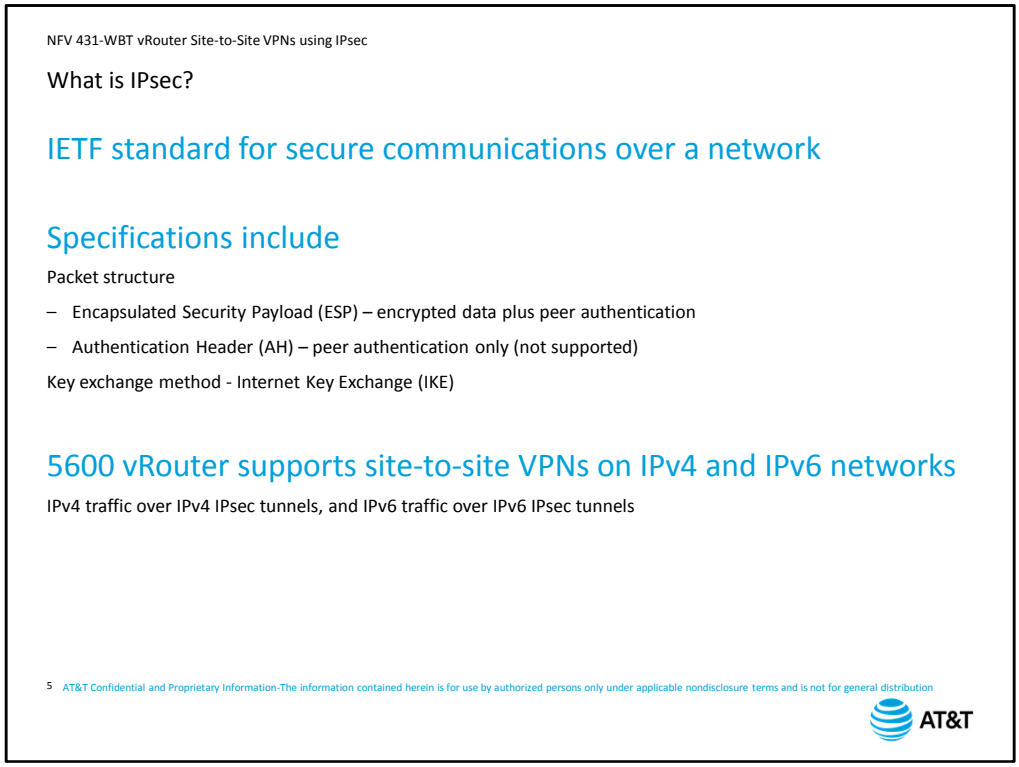


After completing this course, you will be able to:

- Explain the purpose of IPsec
- Describe the IKE exchange to establish a secure tunnel
- Configure a site-to-site IPsec VPN on a vRouter
- Configure common IPsec variations
- Verify tunnel operations and
- Troubleshoot common misconfigurations



We begin with an overview of IPsec, how it works, and what common problems can occur.



IPsec is a suite of protocols designed to provide end-to-end security at the network layer (Layer 3), using encryption and authentication techniques. The only devices that require an IPsec implementation are the IPsec endpoints.
The IPsec specifications include two packet structures:
Encapsulated Security Payload, which provides for data encryption and tunnel peer authentication.
Authentication Header, which only includes peer authentication. The vRouter IPsec implementation only supports ESP.
The IPsec specification also includes a key exchange method called Internet Key Exchange, or IKE.
The 5600 vRouter currently supports site-to-site IPsec VPN connectivity on both IPv4 and IPv6 networks (IPv4 traffic over IPv4 IPsec tunnels, and IPv6 traffic over IPv6 IPsec tunnels).

NFV 431-WBT vRouter Site-to-Site VPNs using IPsec

Internet Key Exchange (IKE)

Phase 1: Establish secure communications site to site

Verify peer identity

Negotiate encryption/authentication for Phase 2 exchanges

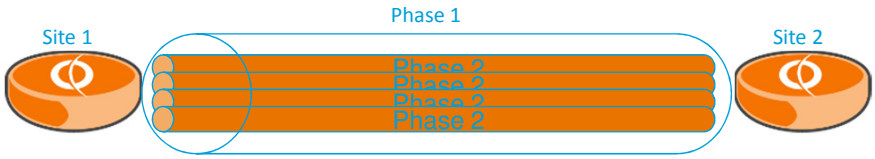
Two modes: Main mode and aggressive mode

Phase 2: Negotiate Security Association (SA) for data exchange

SA specifies which encryption and authentication methods to use for specific source/destination address pairs

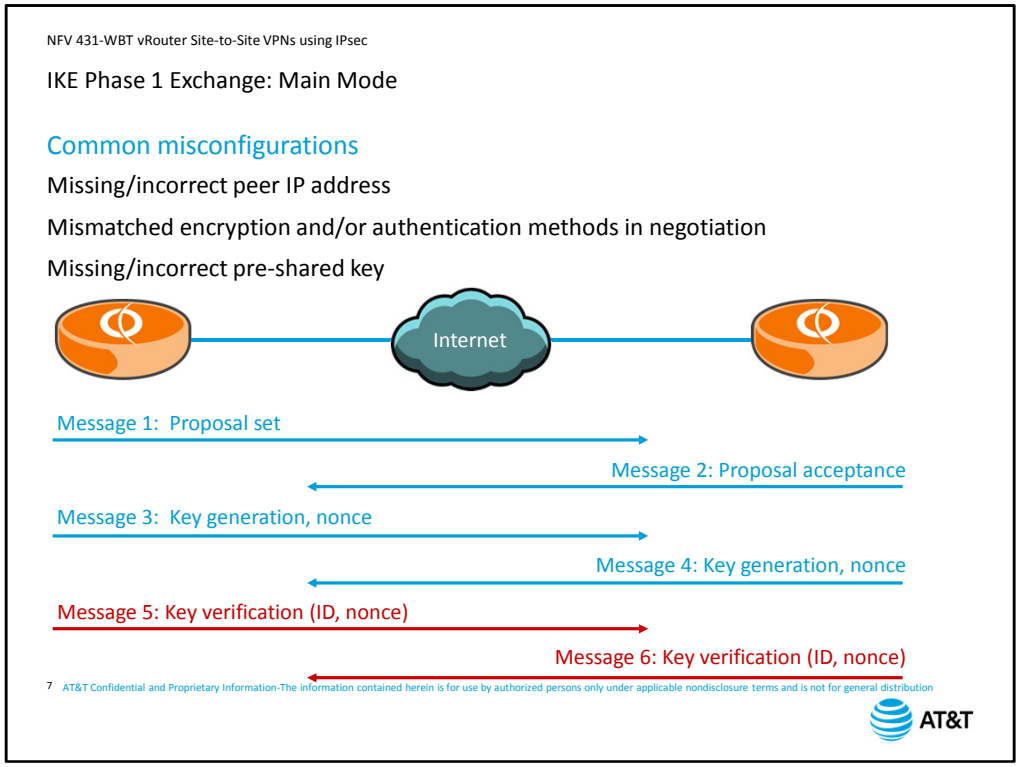
SA is unidirectional; must have SA for each direction

Can have multiple Phase 2 tunnels within one Phase 1 connection



AT&T

The Internet Key Exchange has two phases.
Phase 1 establishes a secured, encrypted connection between two tunnel peers. The purpose of Phase 1 is two-fold:
To verify the tunnel peers are supposed to communicate with each other, and to establish an encrypted connection for Phase 2 exchanges.
IKE Phase 1 has two modes, main mode and aggressive mode. The vRouter only supports main mode, as it is more secure.
IKE Phase 2 is used to negotiate Security Associations.
A Security Association (SA) specifies which encryption and authentication methods to use for a given set of source and destination addresses.
An SA is unidirectional, so in order for two sites to exchange data, you need two SAs – one for each direction.
You can have multiple Phase 2 tunnels operating between a single pair of VPN peers.
For example, you may have four discontiguous subnets at Site 1 being tunneled to Site 2.
Each subnet would have it's own Phase 2 definition, and own pair of SAs.



As we stated on the previous slide, the purpose of IKE Phase 1 is to create an encrypted connection between the two tunnel peers, and to verify peer identity. Main mode consists of six messages.

When a peer begins tunnel negotiations, it sends a message that contains the combinations of encryption and authentication methods it can support. These combinations are called proposals. If one of the proposals matches what the responding peer expects, it responds with the proposal it can support.

Two common misconfigurations can cause IKE Phase 1 to fail during the first exchange.

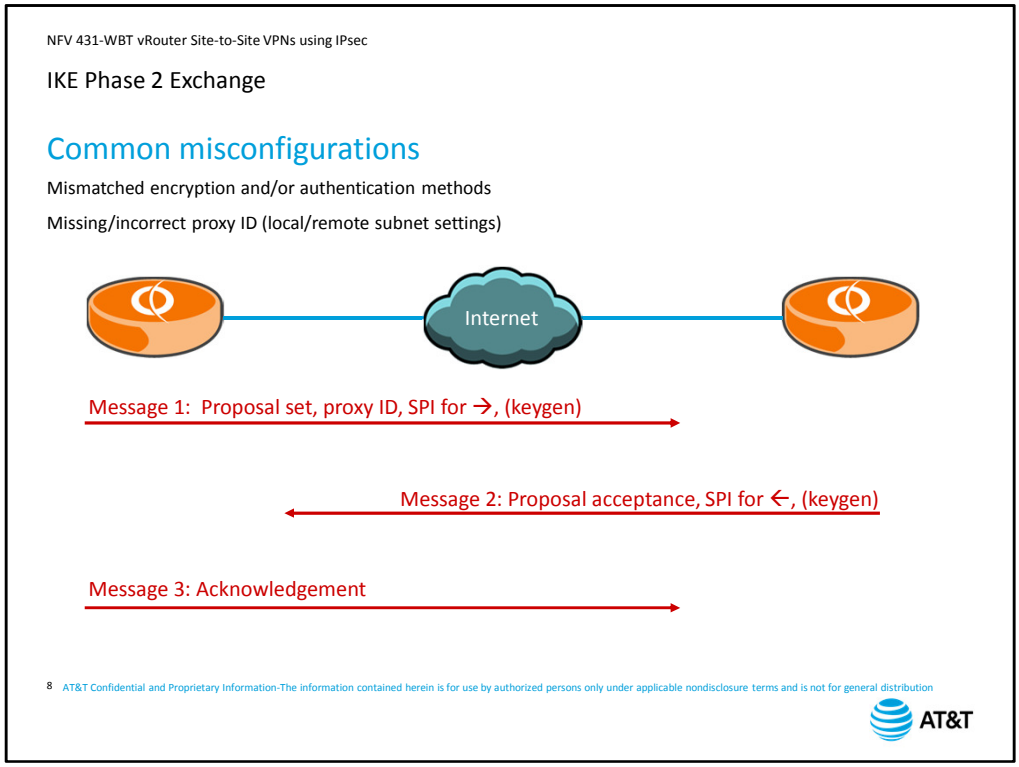
If the responder does not recognize the initiator's IP address, it will reject the connection.

If the responder cannot support any of the proposals sent by the initiator, it will reject the connection.

Assuming that the responder does accept the connection, the next two messages are the actual Diffie-Hellman key exchange. Each peer uses the exchanged data to calculate its own key. If the key exchange is successful, each peer will calculate the same key. The “nonce” value is a random number that a peer will use to verify the calculated keys in the next message exchange.

In the next two messages, the peers encrypt their IDs and the nonce value using their calculated keys. Each peer decrypts and compares the received data with what it has stored to verify that keys were computed correctly. We show these messages in red because they are encrypted, and therefore secure.

This is the second place where Phase 1 will fail. If you are using pre-shared keys, and the keys do not match on each side, the calculated keys will not match, and Phase 1 negotiations fail.



Once Phase 1 is complete, the peers have a secured connection and can begin Phase 2 negotiations.

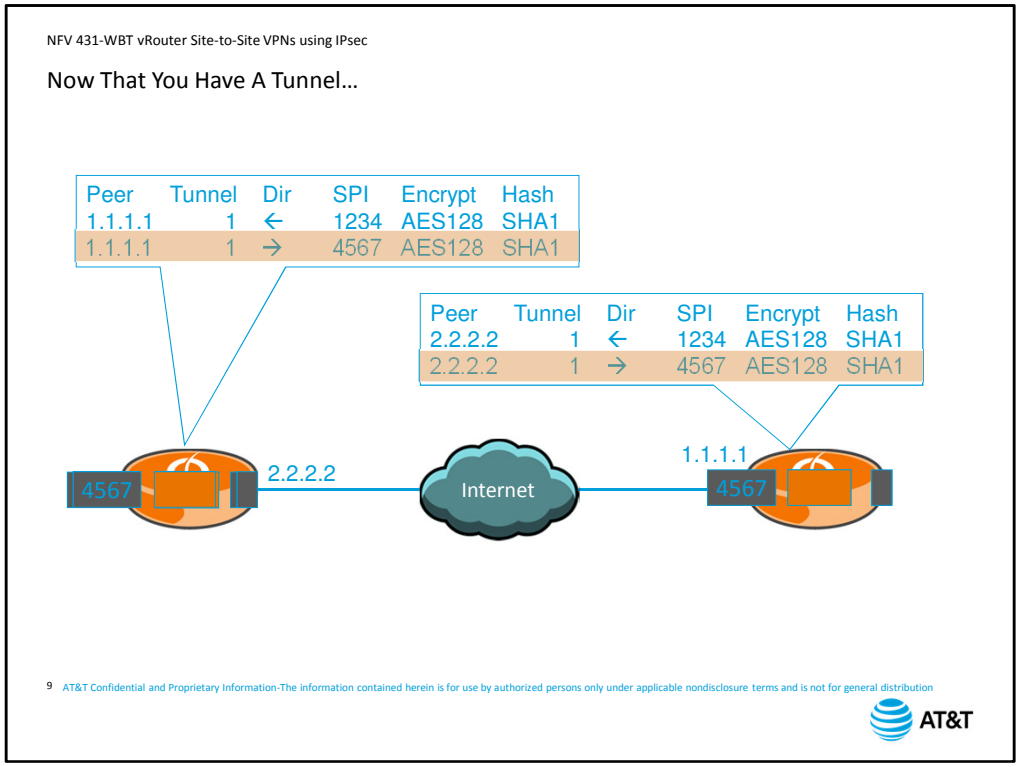
The initiator sends a message containing the Phase 2 proposals it can run; the proxy ID, which defines the source and destination subnets being tunneled, the Security Parameter Index to use for this direction of traffic, and optionally a second key generation. This optional rekeying is called Perfect Forward Secrecy or PFS.

The responder replies with the accepted proposal, the Security Association for traffic in this direction, and optionally the second half of the PFS rekeying.

The third message is simply an acknowledgement that all information has been properly exchanged and the two peers can now forward end user data.

Phase 2 typically fails for one of two reasons:

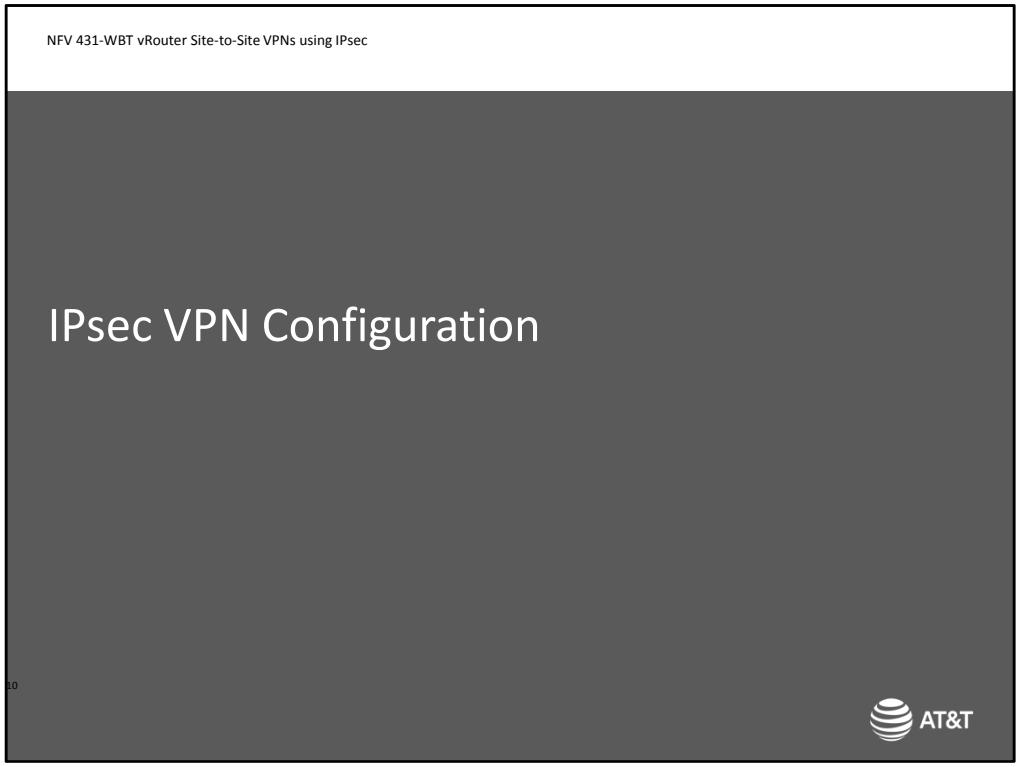
The responder cannot accept any of the proposals presented by the initiator, or the proxy ID values do not match between sites. In the vRouter configuration, the proxy ID value is set by the local-subnet and remote-subnet configuration settings.



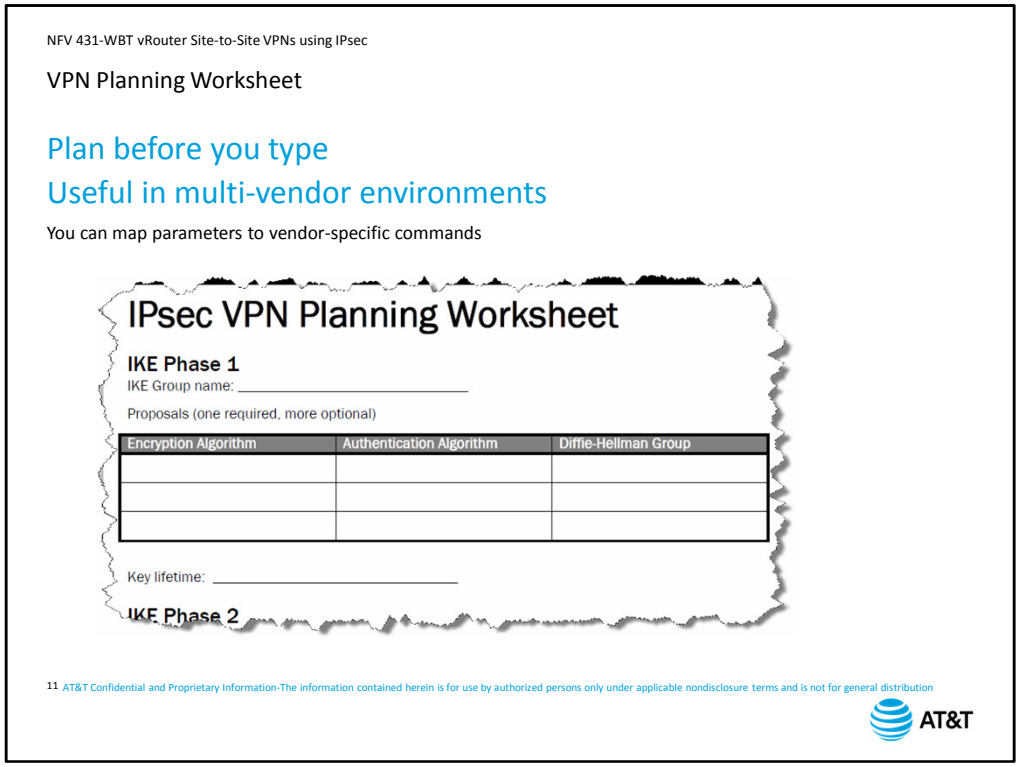
Once Phase 2 has successfully completed, each device has information in its Security Association table. The devices then use the SPI values to correctly encrypt and decrypt packets they exchange.

When a device receives a packet to be tunneled, it looks up the tunnel ID for the packet. It then uses the information to encrypt the packet, calculate the hash, adds the SPI to the encapsulation header, then sends the packet to the tunnel peer.

The receiver looks up the SPI, then uses the information to check the hash and decrypt the packet.



Now we look at the commands you need to set up a basic site-to-site VPN using IPsec.

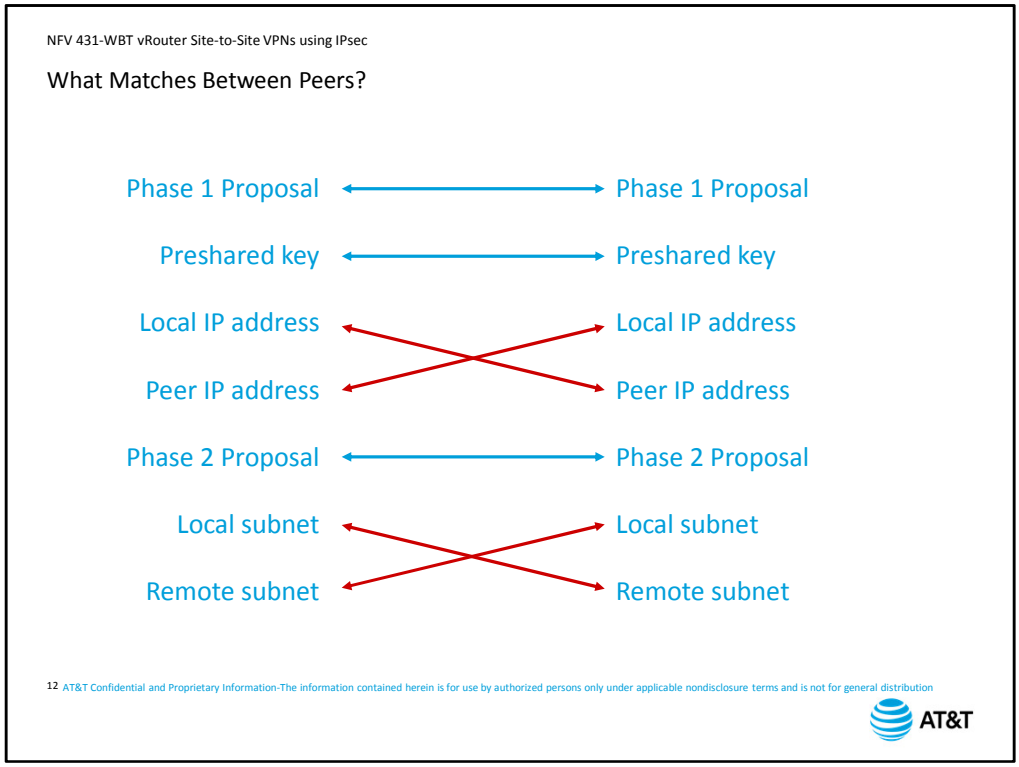


One of the most common issues with setting up VPNs, especially in a multi-vendor environment, is making sure that parameters match on either side of the connection. Any mismatch means tunnel negotiations fail, so it is worth taking the time to complete some kind of planning worksheet before you start configuring your device.

We have provided a planning worksheet with this course. Select the *Attachments* tab in the upper right corner.

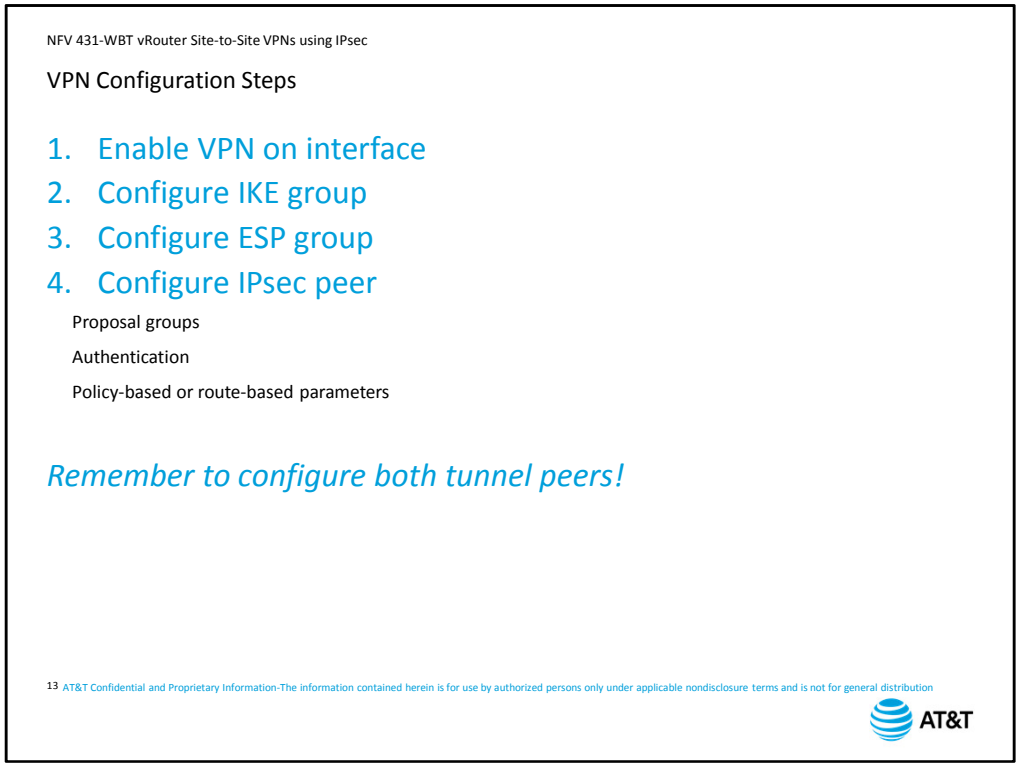
A planning worksheet helps in a multi-vendor environment, as it gives you a place to write down all the parameters you will need for your configuration

Because vendor-specific terminology may differ, you can take the worksheet and vendor documentation and match parameter for parameter to ensure that you have a complete configuration on both ends of your connection.

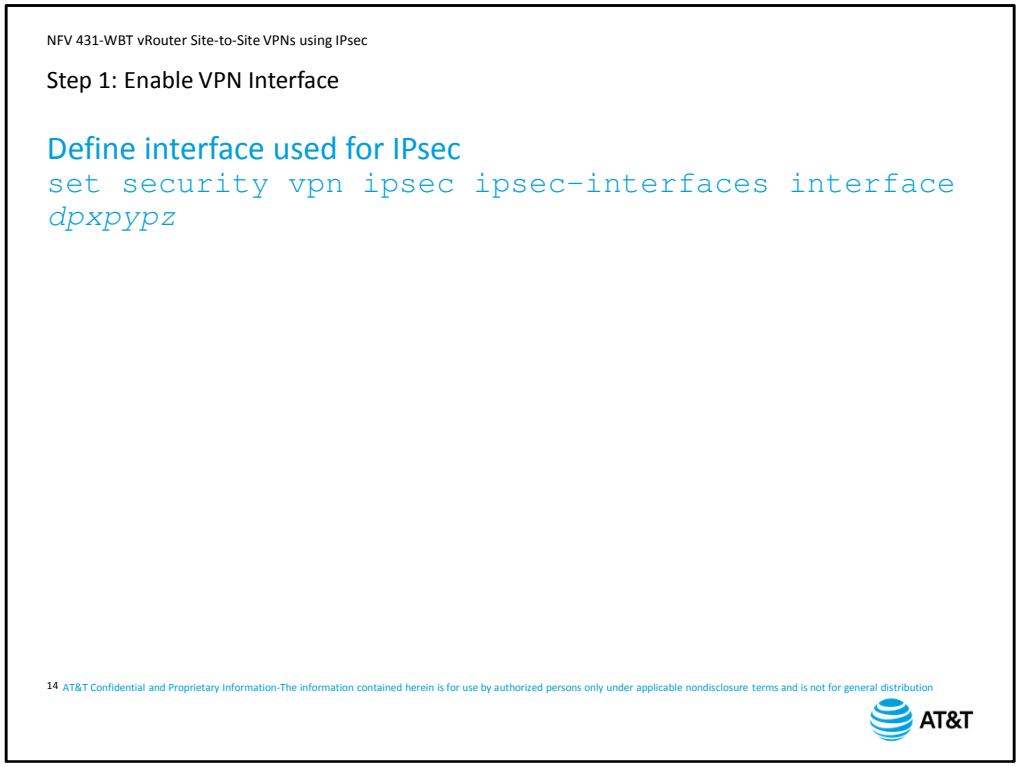


When filling out your configuration worksheet for each tunnel peer, you should check to make sure that the parameters match between peers as needed:

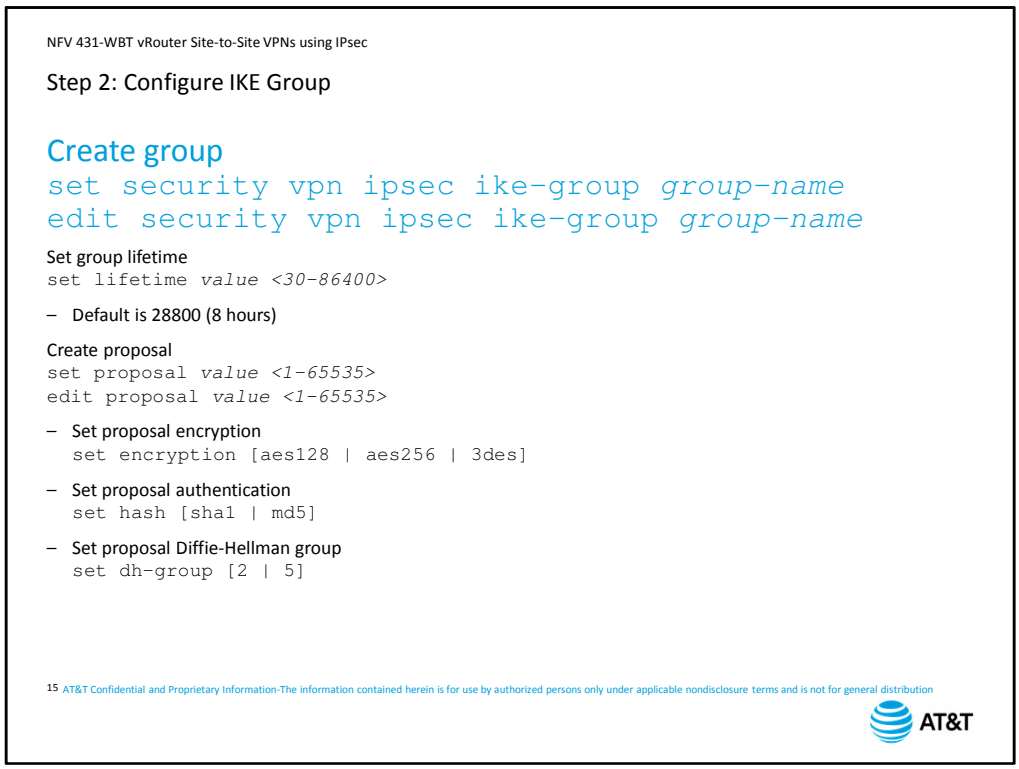
- At least one Phase 1 proposal must match.
- The preshared key must match.
- Local and Peer IP addresses must be transposed: that is, the local address for one side of the connection is the peer address for the other side of the connection, and vice-versa.
- At least one Phase 2 proposal must match
- If configuring policy-based VPNs, the local and remote subnet settings must be transposed; that is, the local subnet for one side of the tunnel is the remote subnet for the other side, and vice-versa. Also, the subnet addresses and masks must match exactly.



Configuring an IPsec VPN on the vRouter consists of four major steps:
The first step is to enable VPN support on the interface that is the tunnel termination point. Usually this is the interface connected to the Internet.
The second step is to configure a set of IKE Phase 1 proposal parameters, called an IKE group.
The third step is to configure a set of IKE Phase 2 proposal parameters, called an ESP group.
Finally, configure the IPsec peer, referencing the proposal groups, configuring authentication, and setting up either the policy-based or route-based parameters.
Remember, a VPN consists of two peers, so you must configure both sides of the connection with matching parameters.



Step 1 has a single command. You need to enable IPsec support on any interface that must perform encapsulation and decapsulation of IPsec packets.



The next step is to create an IKE group and configure the proposals included in the group. AT&T recommends the use of the `edit` command to save you some typing and to ensure that all subsequent parameters are associated with the correct IKE group.

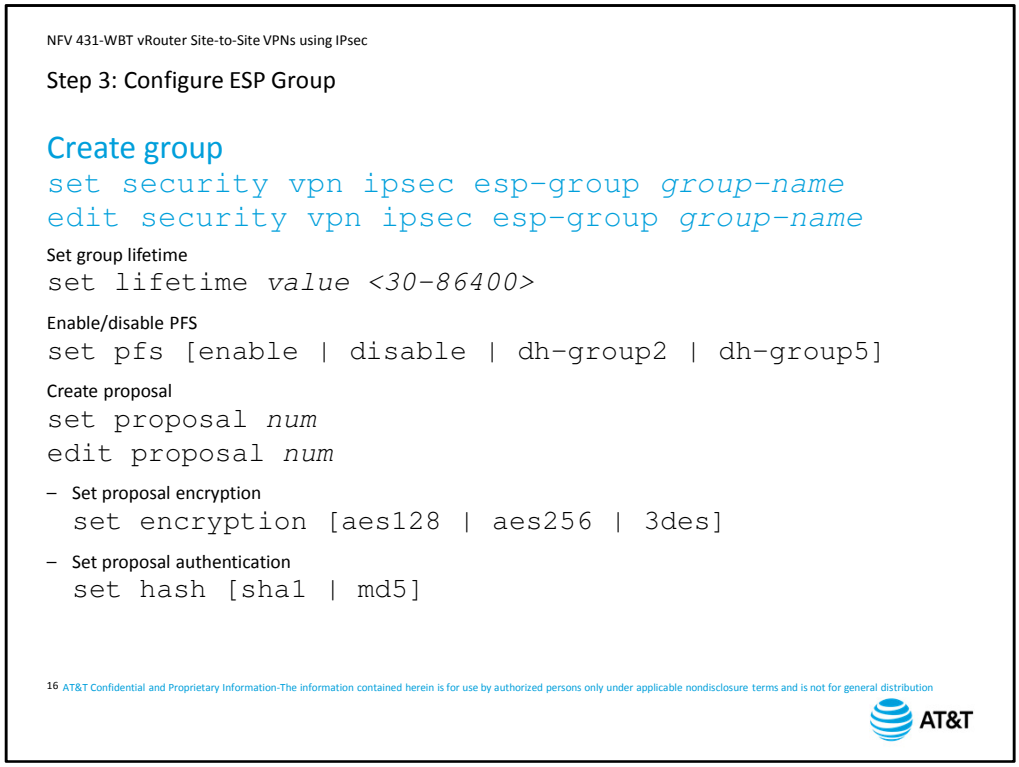
First, set the key lifetime in seconds for the group. This is a negotiated parameter; the peers will agree to use the shortest key lifetime proposed. The default setting is 28800 seconds – that is, 8 hours.

Next, create a proposal. You can include up to 10 proposals in a single IKE group. As long as your peer configuration includes a proposal that matches yours, this stage of the IKE Phase 1 exchange will complete successfully. Again, we recommend using the `edit` command.

Set the encryption for the specified proposal. The default is *aes128*.

Set the authentication hash method for the proposal. The default is *sha1*.

Finally, select the Diffie-Hellman group (dh-group) for the proposal. There is no default setting; you must include a Diffie-Hellman selection.



The next step is to create an ESP group and configure the proposals included in the group. Again, we suggest using the `edit` command.

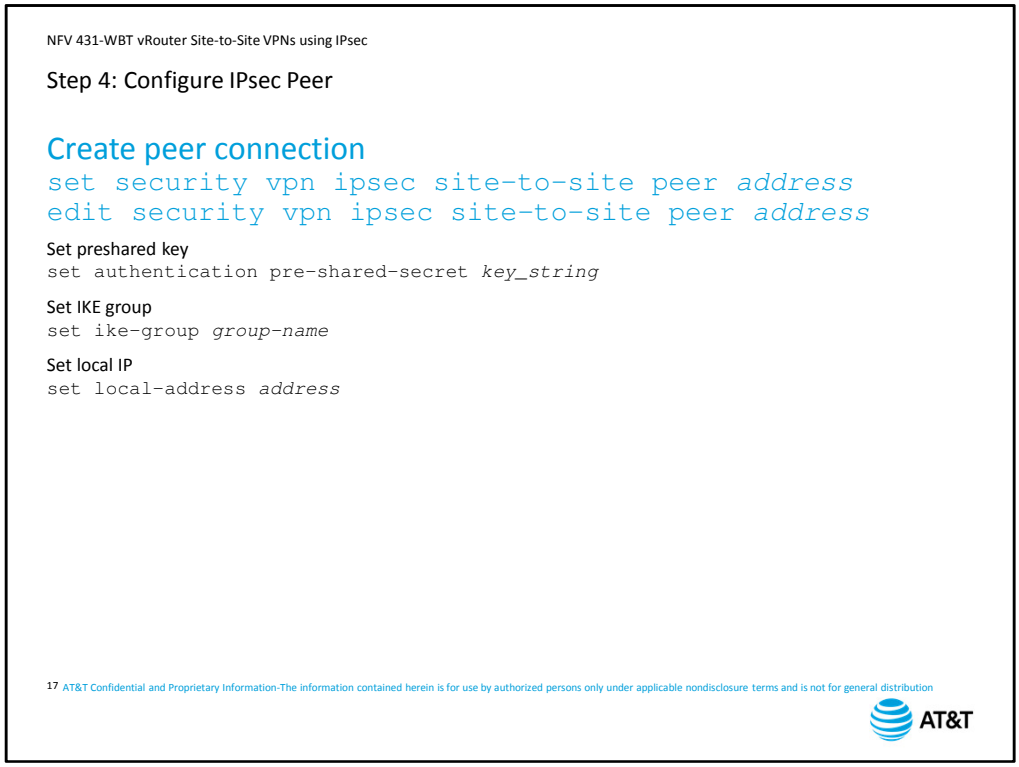
First, set the key lifetime for the group. This is a negotiated parameter; the peers will agree to use the shortest key lifetime proposed. The default setting is 8 hours.

Next, you can choose to disable perfect forward secrecy. By default, PFS is enabled.

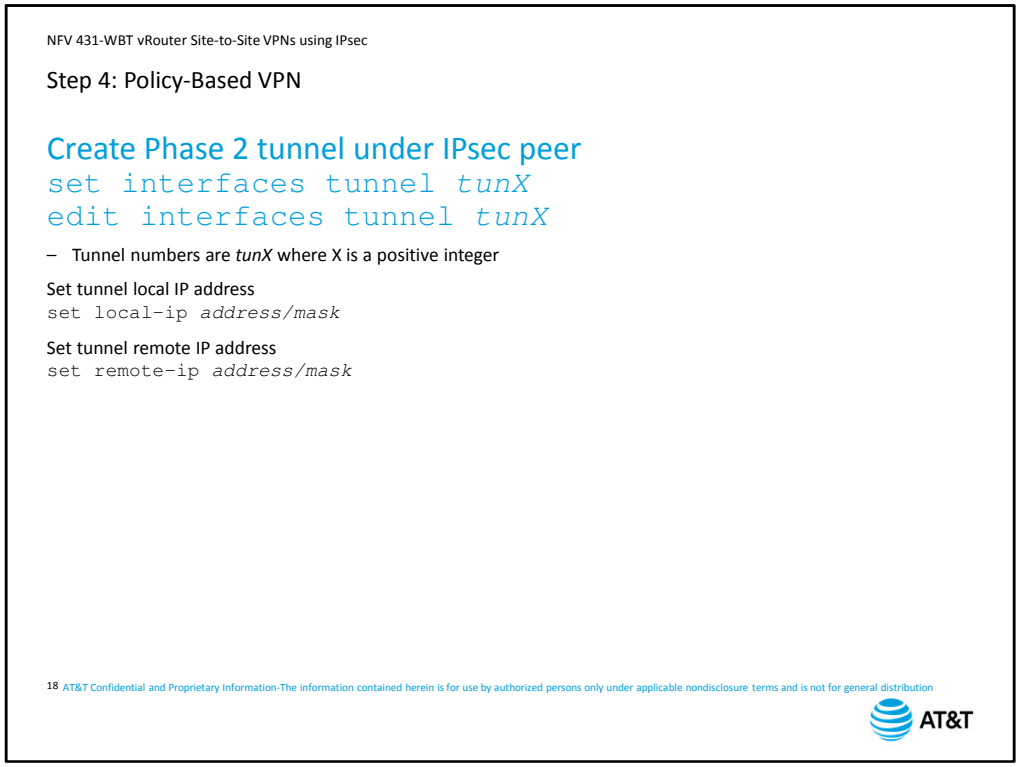
Next, create a proposal. You can include up to 10 proposals in a single ESP group. As long as the peer configuration includes a proposal that matches yours, this stage of the IKE Phase 2 negotiation will complete successfully.

Set the encryption for the proposal. The default is *aes128*.

Finally, set the authentication hash method for the proposal. The default is *sha1*.



Finally, create the connection to the tunnel peer. First, specify the site-to-site peer address. This address must match the local IP address configured on the remote peer.
Again, because there are several parameters for each VPN peer, using the edit command will ensure that your parameters all map to the same peer.
Next, if using preshared keys, set the authentication to pre-shared-secret and enter the preshared key string in plain text. This key will be stored in plain text so it can be recovered by looking at the configuration.
Next, specify which IKE proposal group you want to use with this peer.
Next, set the local IP address. This is the IP address of the interface you enabled VPN support on in step 1.

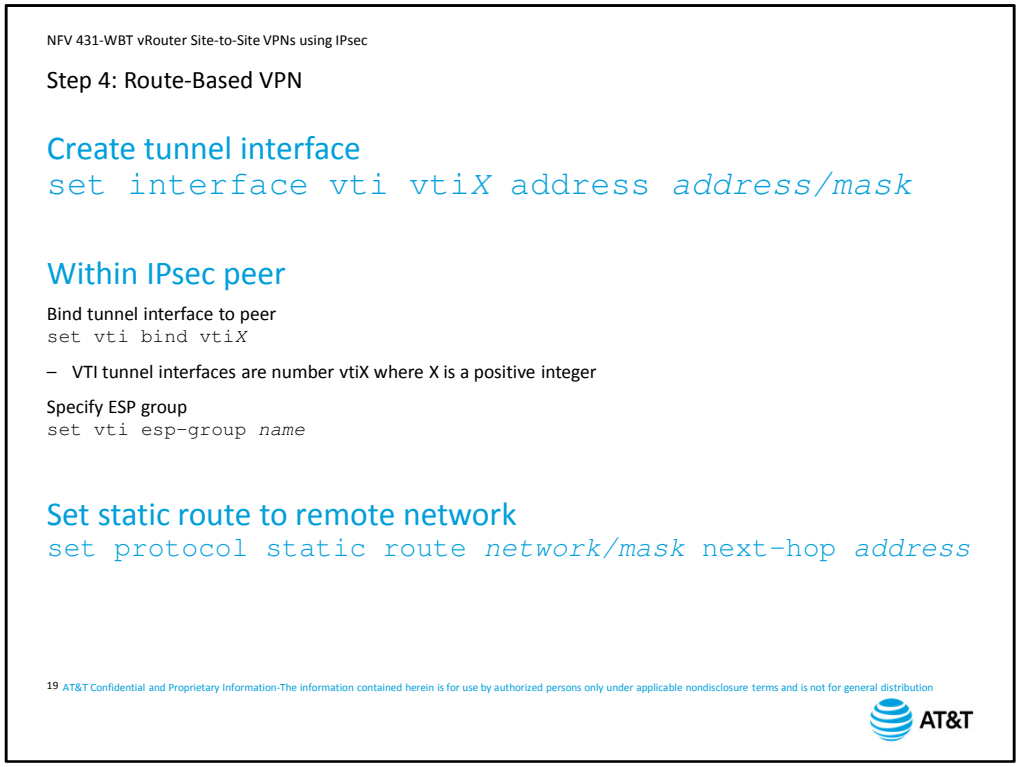


If you are configuring policy-based VPNs, your next step is to create the Phase 2 tunnel. Remember, you may have multiple Phase 2 tunnels if you are configuring for multiple source and destination private networks over the same VPN.

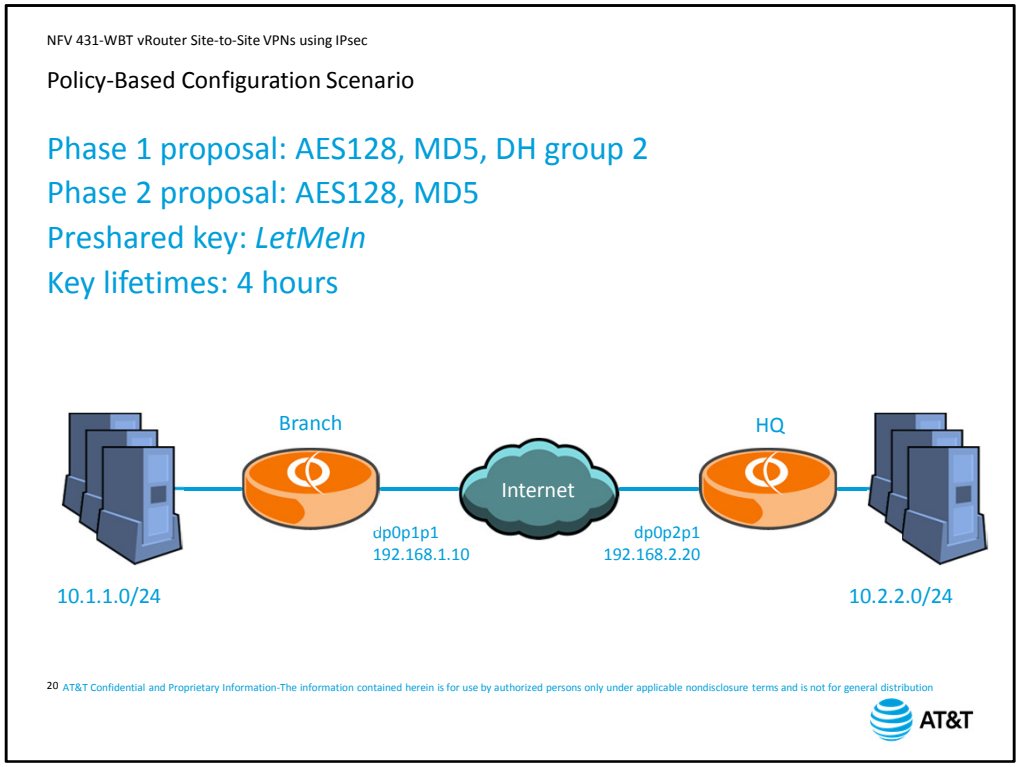
Again, because the tunnel has several additional parameters, we recommend using the `edit` command to create the tunnel and move within the configuration hierarchy.

For the Phase 2 tunnel, we specify the local IP address. This is the subnet where traffic to the tunnel originates.

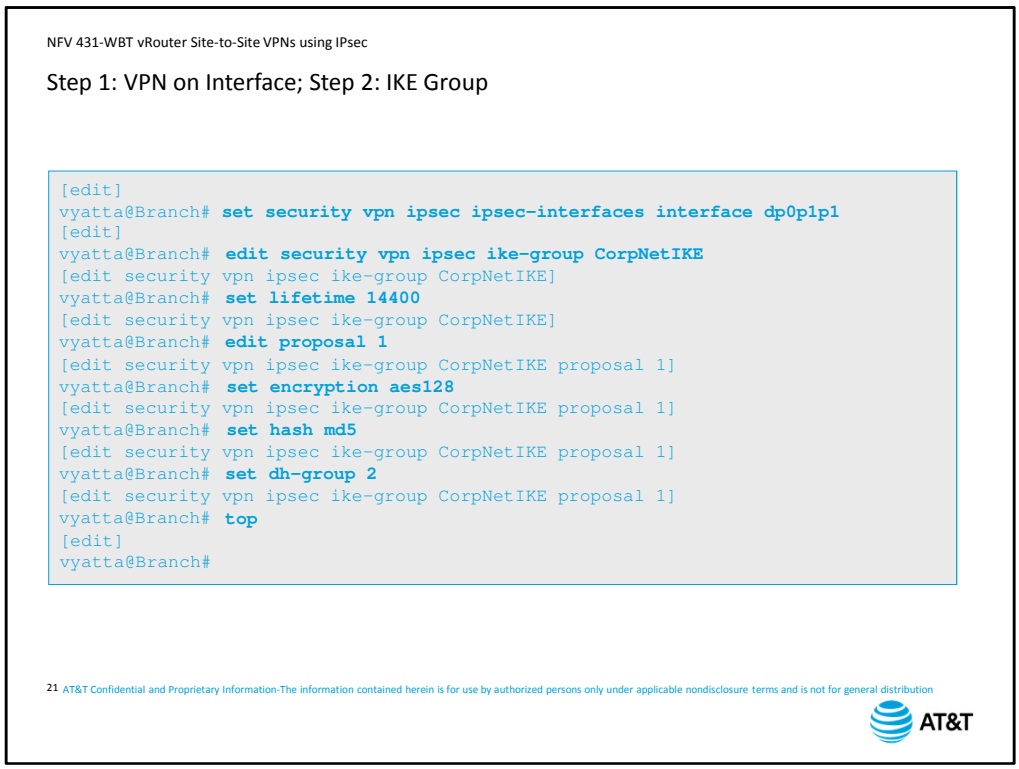
Then specify the remote IP address. This is the destination for traffic to be tunneled, and corresponds to a subnet configured on the far end of the tunnel.



If configuring route-based VPNs, you have to configure some elements under the VPN peer, and others at different levels in the configuration hierarchy.
Outside the IPsec peer hierarchy, you need to create a tunnel interface and assign it an IP address. Both peers must use the same subnet for the tunnel interface.
Under the IPsec peer, you specify which virtual tunnel interface is bound to the peer.
You also specify the Phase 2 ESP group used for IKE.
Finally, set a static route to the remote network, using the peer virtual tunnel IP address as the next-hop address.



We will use this configuration scenario to provide a complete configuration example.
The vRouters are connected to the Internet using data plane interface 1. We will need to activate VPN support on these interfaces. These are also the local and remote peer addresses.
The subnets we want to tunnel are 10.1.1.0 at the branch site and 10.2.2.0 at the head quarters site. These are the local and remote subnets for the tunnel configuration.
For the Phase 1 proposal, we want to use AES 128 encryption, MD5 hashing, and Diffie-Hellman group 2
For the Phase 2 proposal, we want to use AES 128 encryption and MD5 hashing.
For the preshared key, we will use the string *LetMeIn*.
We will set the key lifetimes to 4 hours for both Phase 1 and Phase 2.
You can open this scenario in a separate window by clicking the *Attachments* tab and selecting the *Policy-based Configuration Scenario* document. This will allow you to refer to the diagram as we go through the configuration commands.

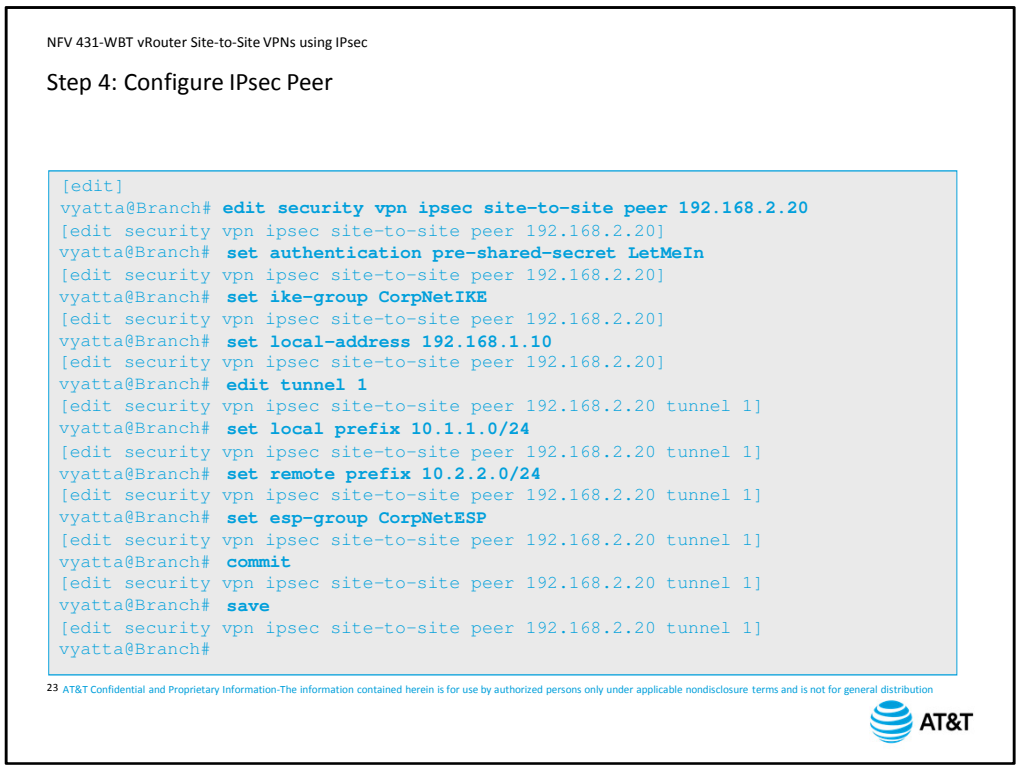


First we configure the branch office.
We begin with step 1, enabling IPsec VPNs on the interface connected to the Internet.
Next, we create the IKE proposal group.
We set the lifetime to 14 thousand 4 hundred 40 seconds, or four hours.
Next, we create and edit proposal 1 within the group.
We set the encryption to AES128 and the hash algorithm to MD5.
Next set the dh-group to 2. If we wanted to create a second proposal for this group, we could use the `up` command twice to go up two levels, then create a second proposal. However, we only have one proposal, so we use the `top` command to go back to the top of the hierarchy.

Step 3: Configure ESP Group

```
[edit]
vyatta@Branch# edit security vpn ipsec esp-group CorpNetESP
[edit security vpn ipsec esp-group CorpNetESP]
vyatta@Branch#  set lifetime 14400
[edit security vpn ipsec ike-group CorpNetESP]
vyatta@Branch#  edit proposal 1
[edit security vpn ipsec ike-group CorpNetESP proposal 1]
vyatta@Branch#  set encryption aes128
[edit security vpn ipsec ike-group CorpNetESP proposal 1]
vyatta@Branch#  set hash md5
[edit security vpn ipsec ike-group CorpNetESP proposal 1]
vyatta@Branch#  top
[edit]
vyatta@Branch#
```

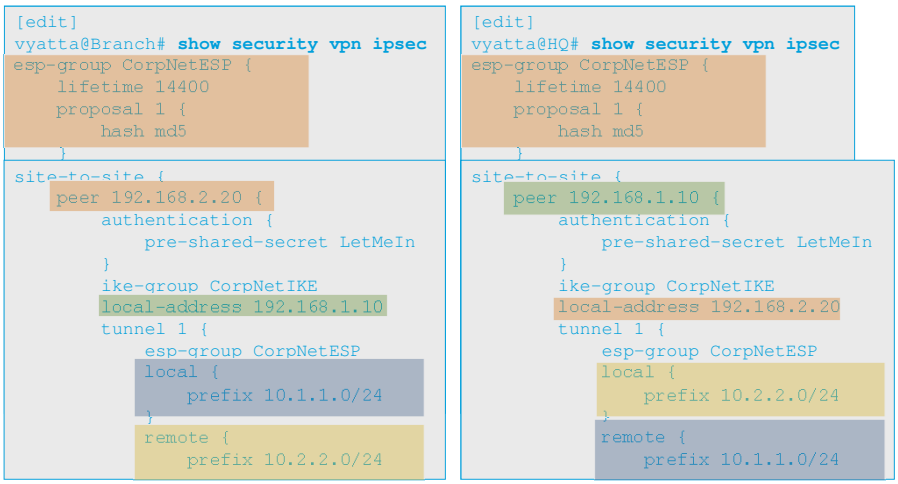

Next, we create the ESP proposal group. Again, we use the `edit` command to create the group and move us within the hierarchy.
We set the group key lifetime, then create the first proposal in the group.
For the proposal, we set the encryption algorithm, and the hash algorithm.
Then return to the top of the hierarchy.



We are now ready to configure the peer.
First, we use the `edit` command to create the VPN peer.
Next, we set the authentication method and preshared key
Next, we specify which IKE group to use for this VPN. We use the IKE group we created in step 2.
Next, we specify our local IP address. The vRouter will match this local IP address with the interface configured with this address – in this case, data plane interface 1.
Next, we create the Phase 2 tunnel, set the local prefix, or the source of traffic to be tunneled, set the remote prefix, or the destination of traffic to be tunneled and specify which ESP group to use for this tunnel. This is the ESP group we created in step 3.
Finally, we commit and save our configuration.

NFV 431-WBT vRouter Site-to-Site VPNs using IPsec

Peer Configurations Side-by-Side



AT&T

When we compare the configurations for the two sites, we see that the proposals are the same on both devices, and that both devices have enabled IPsec on the correct interface. When we look at the VPN configuration itself, we can see the correspondences between local and remote settings, color-coded here for clarity. Settings that are local to the branch office are remote at headquarters, and settings that are local at headquarters are remote at the branch office.

NFV 431-WBT vRouter Site-to-Site VPNs using IPsec

Route-Based Configuration Scenario

Tunnel interface is *vti1*

Uses subnet 10.100.100.0/30

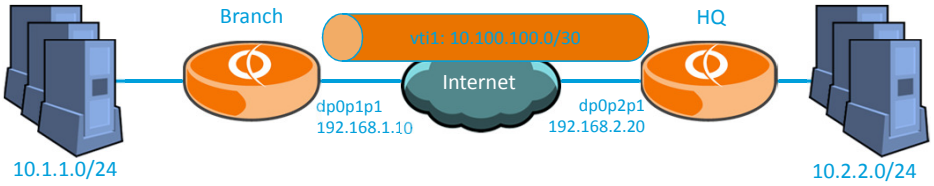
Branch is .10, HQ is .20

Phase 1 proposal: AES128, MD5, DH group 2

Phase 2 proposal: AES128, MD5

Preshared key: *LetMeIn*

Key lifetimes: 4 hours



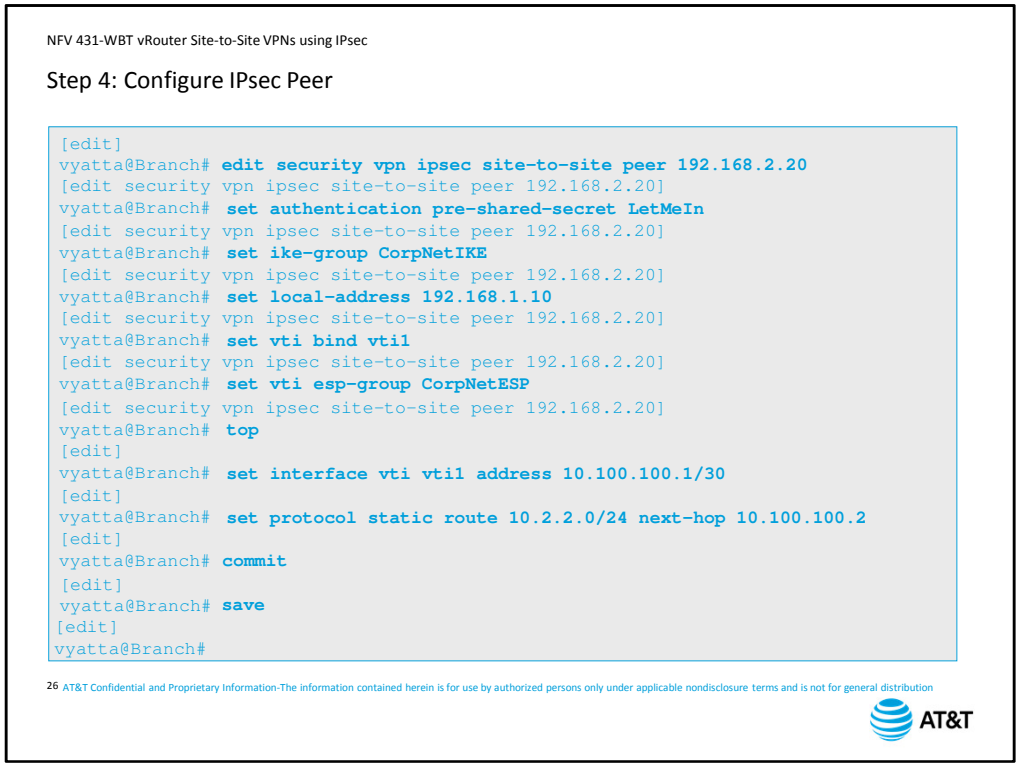
AT&T

Now let's look at the configuration for the same network, using a route-based configuration.

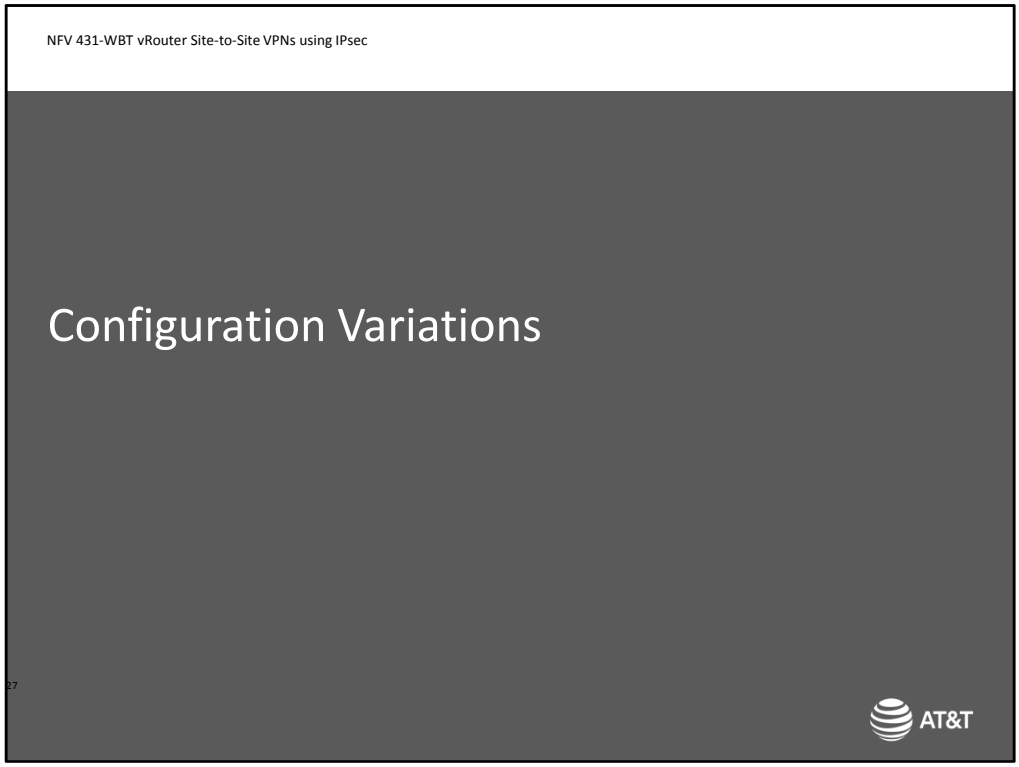
We need to add the tunnel interface to the configuration. We use tunnel interface vti1, and subnet 10.100.100.0 with a 30-bit mask. The branch office will use the first address, and HQ will use the second address.

The rest of our configuration parameters are the same – same tunnel peers, same private networks, same proposals and keys.

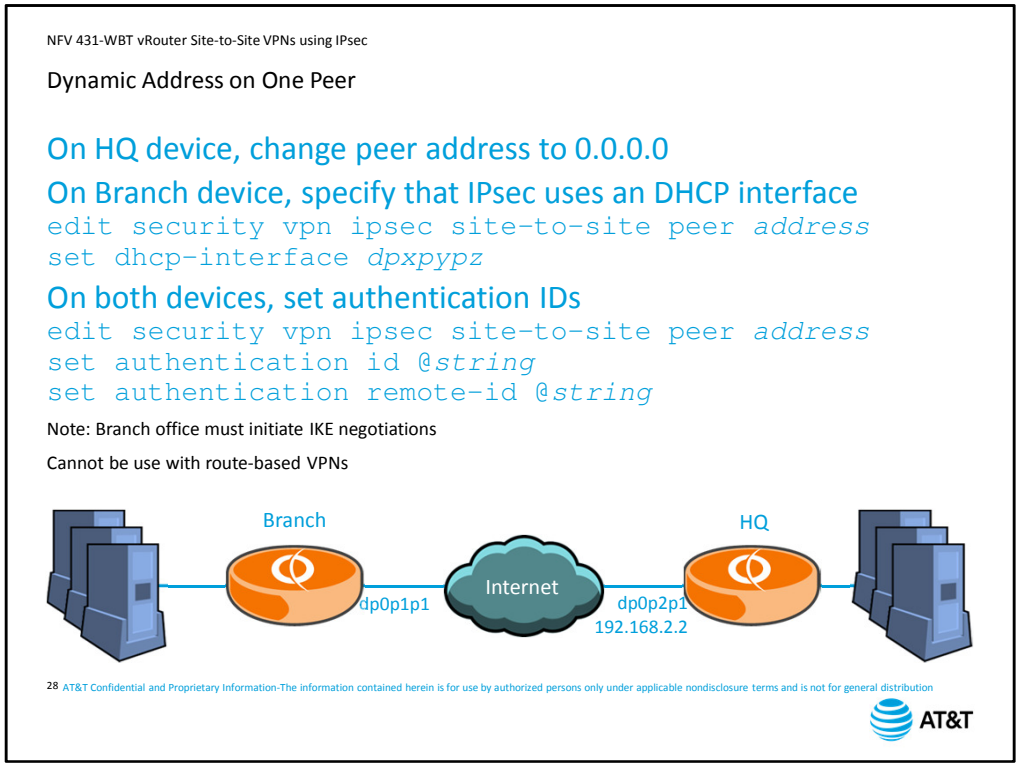
You can open this scenario in a separate window by clicking the *Attachments* tab and selecting the *Route-based Configuration Scenario* document. This will allow you to refer to the diagram as we go through the configuration commands.



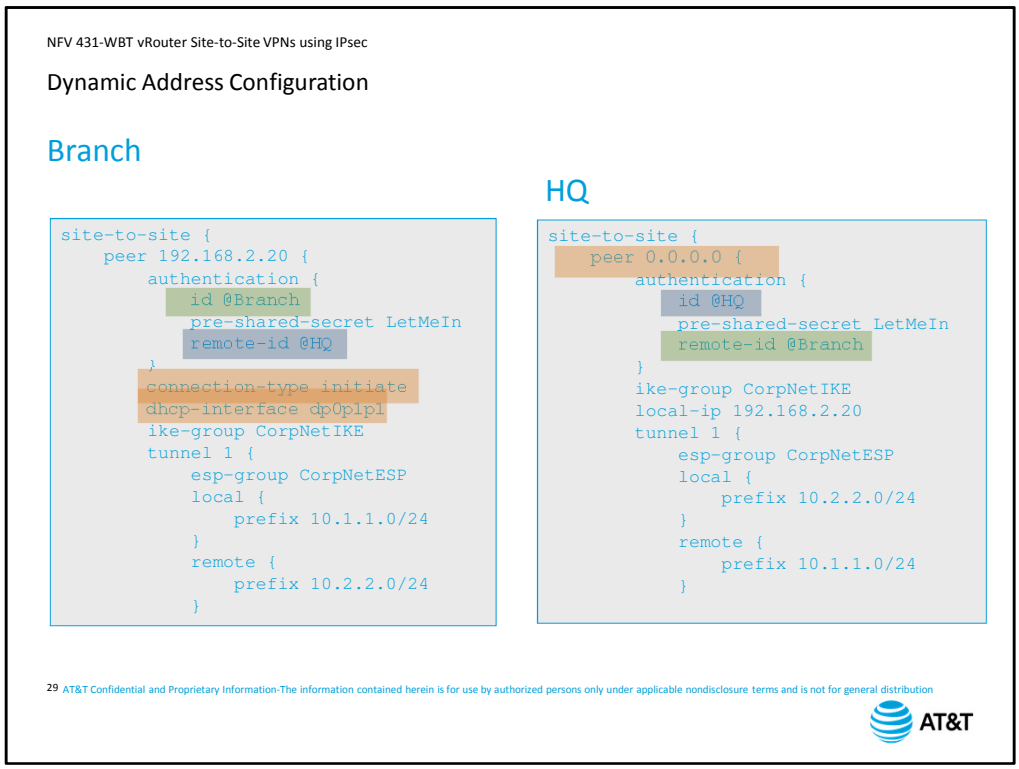
We pick up with our configuration after we have completed all the Phase 1 settings for our IPsec peer.
The next step is to bind the VTI interface to the peer. Don't worry about not having this interface defined yet; we will define it before we commit the changes.
Next, specify the ESP group for this Phase 2 connection.
To create the VTI interface, move to the top of the configuration hierarchy,
Then define the interface and assign it an IP address.
Add the static route to the remote private network, using the VTI peer as the next-hop address.
Commit and save your changes.



Let’s look at a couple of additional variations to the IPsec VPN configuration.



One common variation with VPNs is to have a peer with a dynamic IP address.
In this case, the branch office acquires its address from the ISP using DHCP.
Modifying the configuration to support a dynamically-addressed peer involves a few modifications to the standard configuration. First, on the device with the fixed address – in this case, the HQ device – change the peer IP address to 0.0.0.0, then configure as before.
On the branch device, specify that the outbound interface connecting to the HQ peer is a DHCP interface. This setting is done under the site-to-site peer. You will do this instead of specifying a local address.
Because you can no longer use IP addresses to verify the identify of the peers, you should set an additional authentication check in the form of an ID string. This information will be exchanged during the key authentication part of IKE Phase 1, and if it does not match, Phase 1 will fail. You must include the at sign at the beginning of the ID string.
Note that because the branch office does not have a fixed identity, HQ cannot initiate the connection – it does not have any information to establish a connection. Only the branch office can be the initiator of the VPN.
Finally, a dynamically-addressed device cannot use route-based VPNs.
You can open this scenario in a separate window by clicking the *Attachments* tab and selecting the *Dynamic Address Configuration Scenario* document. This will allow you to refer to the diagram as we go through the configuration commands.

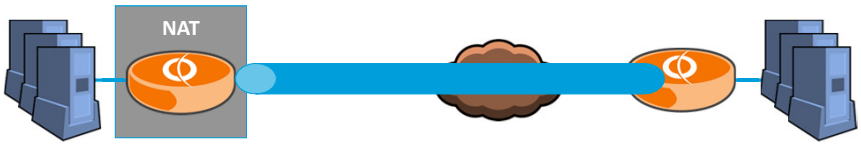


Looking at the configuration side-by-side, we see that the HQ peer is now defined as 0.0.0.0.
The branch is configured to use a DHCP interface for the VPN.
And we now have local and remote ID information to match during Phase 1.
The vRouter automatically added the parameter `connection type initiate` to the Branch office, indicating that this device must be the one to initiate the connection.

VPNs and NAT

NAT for the original packet

Define NAT rules to match private network traffic with correct outbound interface



NAT for the IPsec packet

Either enable NAT-traversal on VPN peer or IPsec-passthrough on NAT device

```
set security vpn ipsec nat-traversal [enable | disable]
```

Modify configuration for dynamic addressing; device being translated is dynamic peer

AT&T

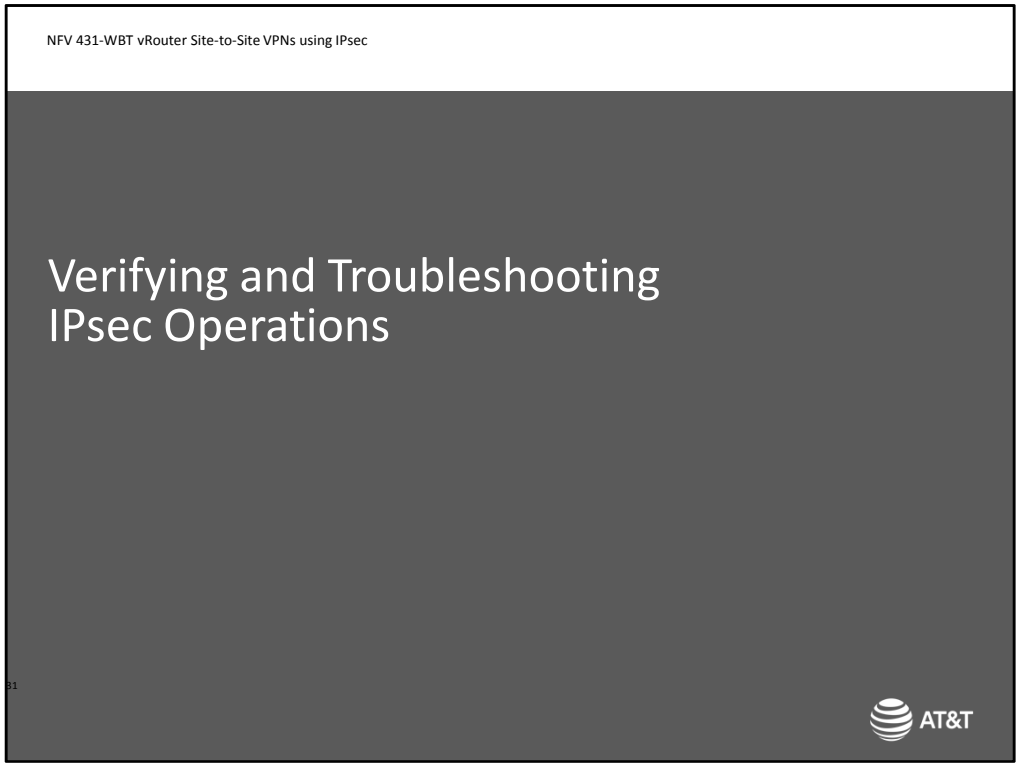
If you are using NAT with your VPNs, you need to first consider what is being translated and where the translation is taking place.

If you need to translate the original packet, make sure that the NAT policy is assigned to the appropriate outbound interface as determined by the routing decision for the ORIGINAL packet. If you are using route-based VPNs, the NAT policy should be applied to the tunnel interface. If you are using policy-based VPNs, you will need to look at your routing table to determine where the NAT policy should apply.

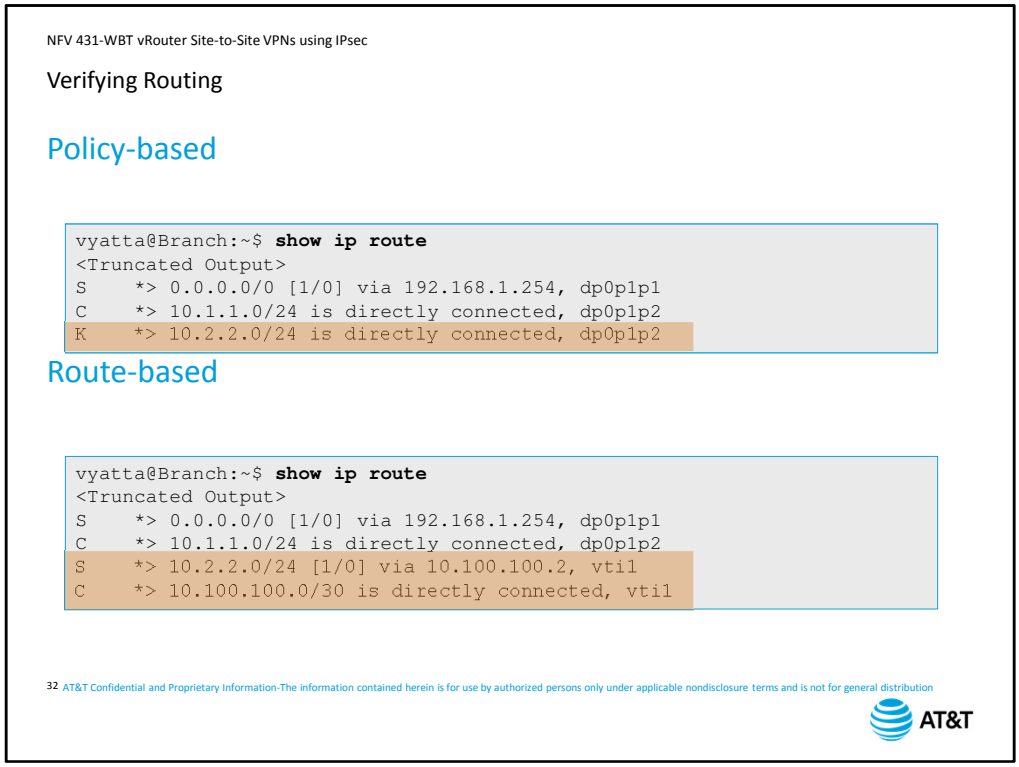
If you need to support NAT on the IPsec packets sent between tunnel peers, you need to either enable NAT-traversal on the tunnel peers, or IPsec-passthrough on the NAT device. NAT-traversal is a standard that re-encapsulates the IPsec packet inside a UDP packet to support NAT. Note that NAT-traversal is a global setting for IPsec and is not tunnel-specific. Do not enable both NAT-traversal on the tunnel peers and IPsec-passthrough on the NAT device; they cannot co-exist.

You also need to modify the configuration as you would for dynamic addressing, as we discussed on the previous two screens.

You can open this scenario in a separate window by clicking the *Attachments* tab and selecting the *VPNs and NAT Configuration Scenario* document. This will allow you to refer to the diagram as we go through the configuration commands.



Now let’s look at the commands you can use to verify your VPN is up and operational, as well as what to look for when it is not.



As we discussed earlier, a policy-based VPN will create a kernel route in your routing table for the destination network. Note that this only includes the defined destination network; at this level, you cannot view the VPN policy.
A route-based VPN will add two entries to your routing table: the static route to the destination network that you manually configured, and the directly-connected subnet of the tunnel interface.
Note that we have used static routes in our examples, but you could also configure a dynamic routing protocol such as OSPF instead of a static route.

Verifying IKE Phase 1

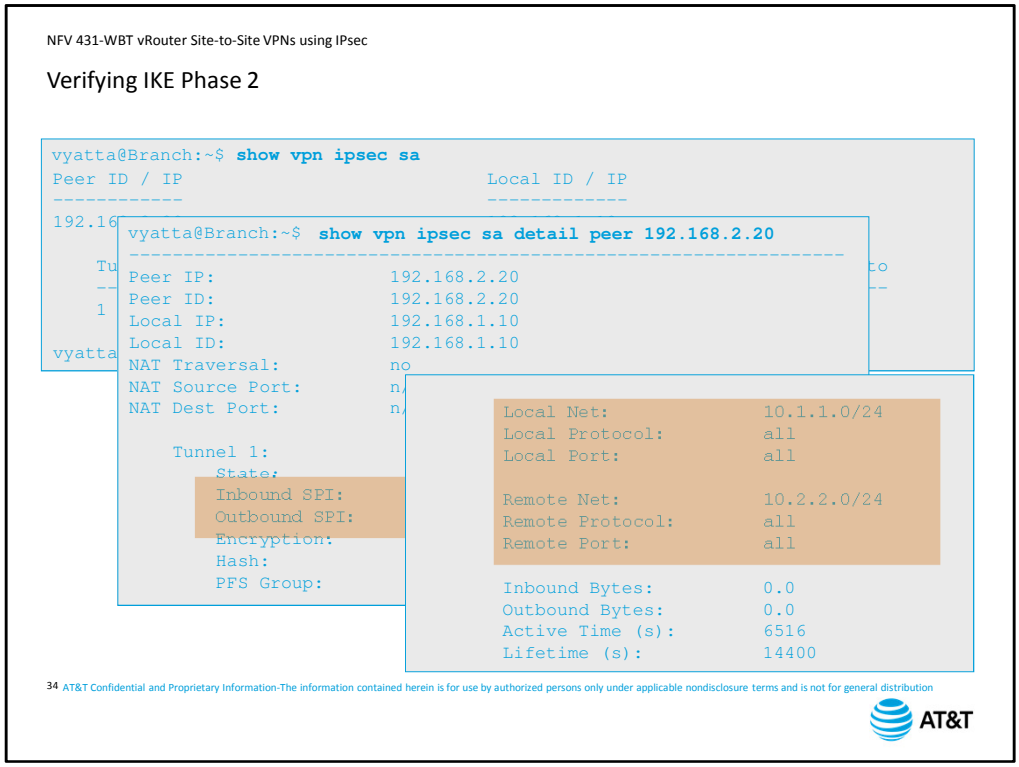
```
vyatta@Branch:~$ show vpn ike sa
Peer ID / IP                            Local ID / IP
------------                            -------------
192.168.2.20                            192.168.1.10

    State  Encrypt  Hash  D-H Grp  NAT-T  A-Time  L-Time
    -----  -------  ----  -------  -----  ------  ------
    up     aes128   md5   2        no     6285    14400


vyatta@Branch~$  show vpn ike secrets
Local IP/ID                             Peer IP/ID
-----------                             -----------
192.168.1.10                            192.168.2.20
N/A                                     N/A

    Secret: "LetMeIn"
```


AT&T

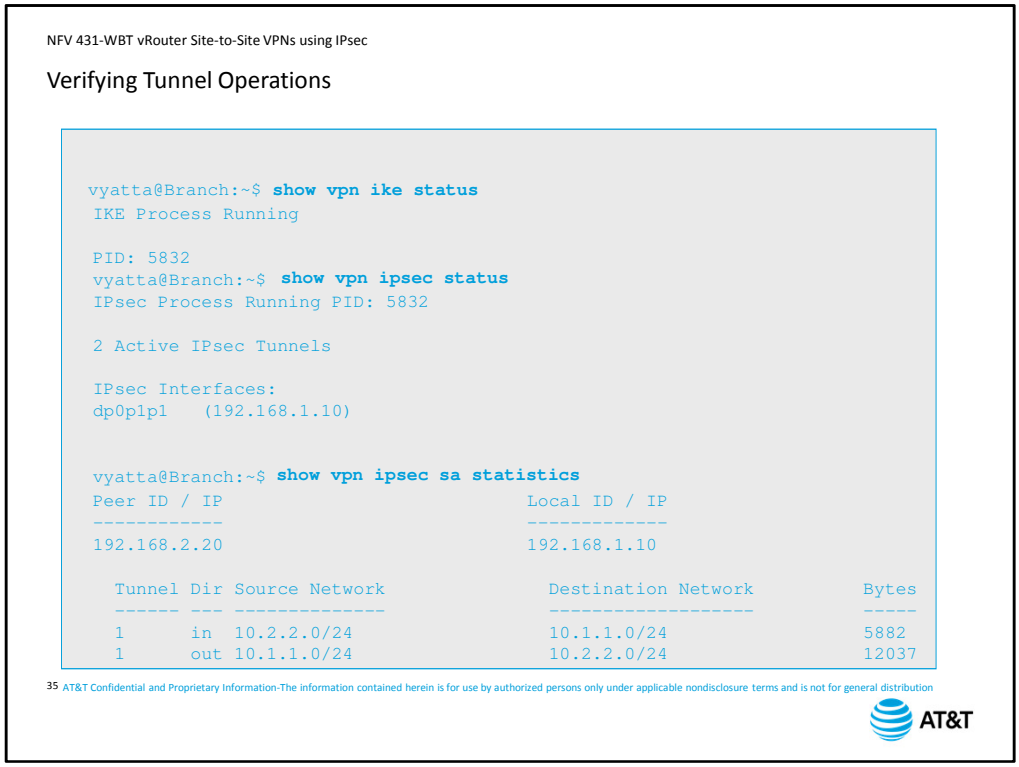
To verify that IKE Phase 1 completed successfully, you can view the negotiated Phase 1 Security Association.

The command is `show vpn ike sa`. The output shows the local and remote peer IP addresses and the agreed-upon encryption and hash algorithms. A-time is the length of time in seconds that the Phase 1 connection has been active, and L-time is the negotiated key duration in seconds.

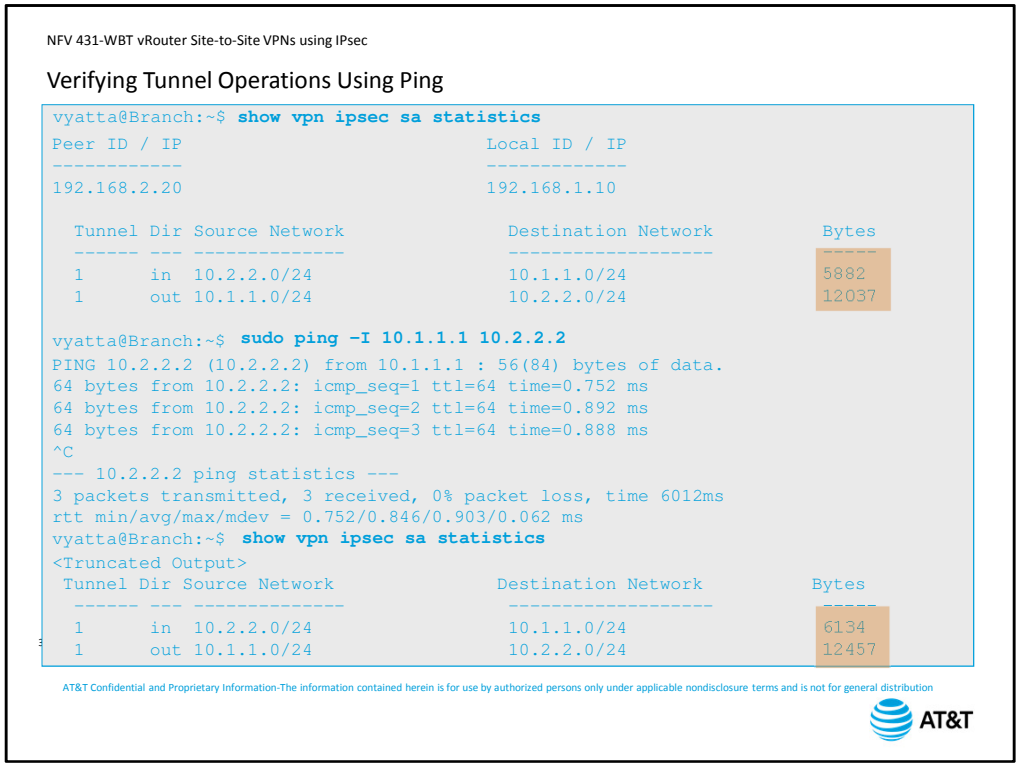
If you do not have access to the configuration, you can view the configured shared secret password string with the command `show vpn ike secrets`.



To view the negotiated Security Associations for Phase 2, use the command `show vpn ipsec sa`. For a successfully-established tunnel, you will see the agreed-upon encryption and hash algorithms, the time the tunnel has been active, and the tunnel key lifetime.
You can view additional details by adding the `detail` parameter and the peer information to the command.
The details include the inbound and outbound security parameter index numbers, and the source and destination networks being tunneled for policy-based VPNs. If you do not have access to the configuration, you can use this command to view the configured local and remote networks, protocol, and ports being tunneled between the peers.



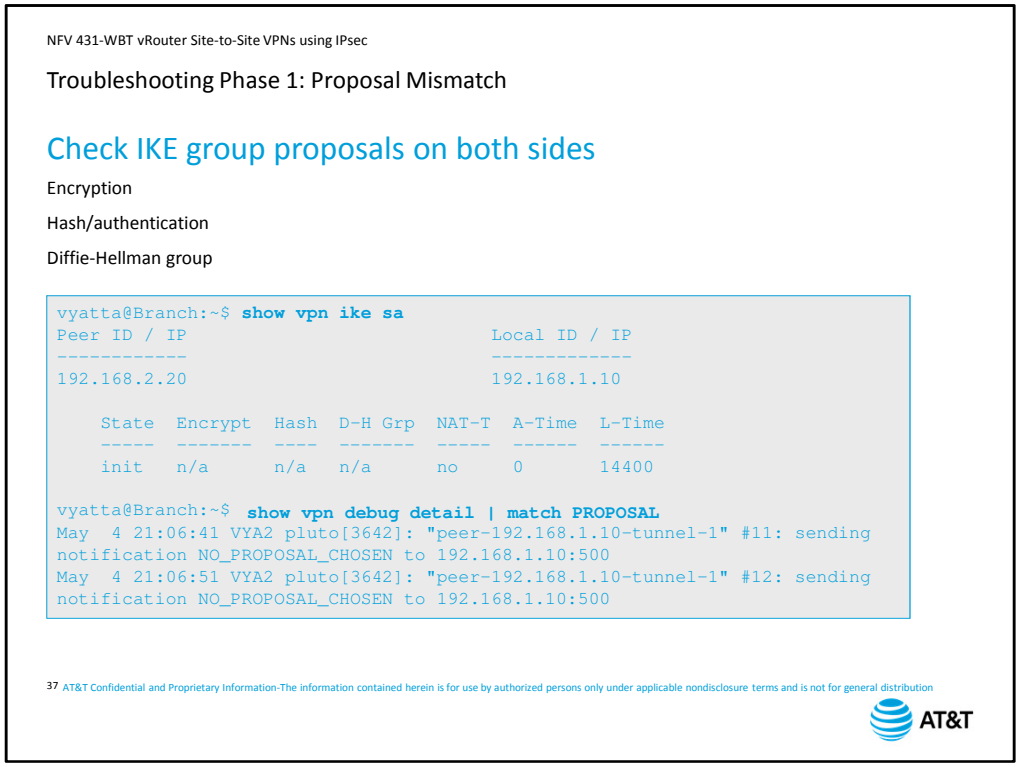
You can verify that the IKE Phase 1 process is running with `show vpn ike status`.
You can verify that the IKE Phase 2 process is running with `show vpn ipsec status`.
This command also shows how many active IPsec tunnels are running on the device.
You can view the volume of traffic traversing your tunnels with the command `show vpn ipsec statistics`.



The easiest way to tell a tunnel is fully operational is to pass traffic through it and verify that tunnel counters are incrementing. However, because the decision to tunnel a packet is based on both source and destination IP addresses, you generally cannot use a simple ping from the vRouter to verify that the tunnel is passing traffic.

If you use the `sudo` command, you can access the full Linux ping command, which allows you to specify the ping source address by using the –I option. Not only is the ping successful,

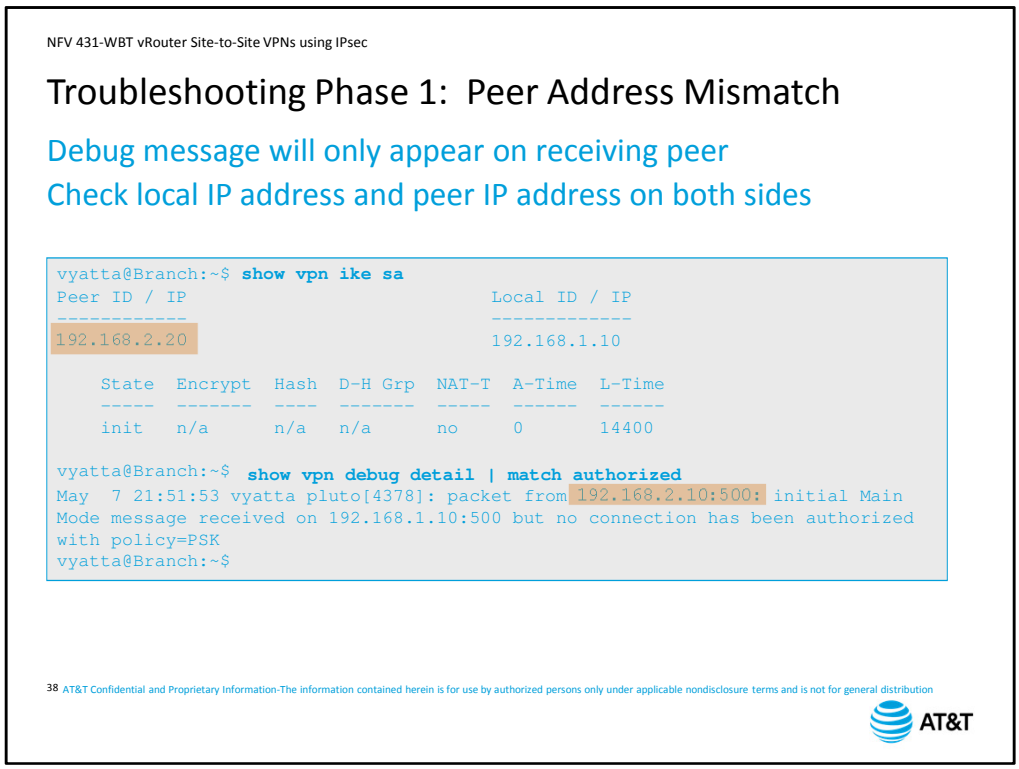
But the statistics have incremented, showing that traffic is passing through the tunnel and not through some other route.



As we said at the beginning of the course, proposal mismatches are a common misconfiguration.

The first indication of a Phase 1 proposal mismatch is that the Phase 1 SA does not exist. We look at the `show vpn debug detail` output, using match to limit our output to search for proposal problems. You will need to run this command on both sides of the connection. In this case, the output indicates that no proposal was chosen for this connection, indicating that the two devices could not agree.

The next step is to check the configuration of both tunnel peers, verifying that the encryption, hash, and dh-group configurations are the same.



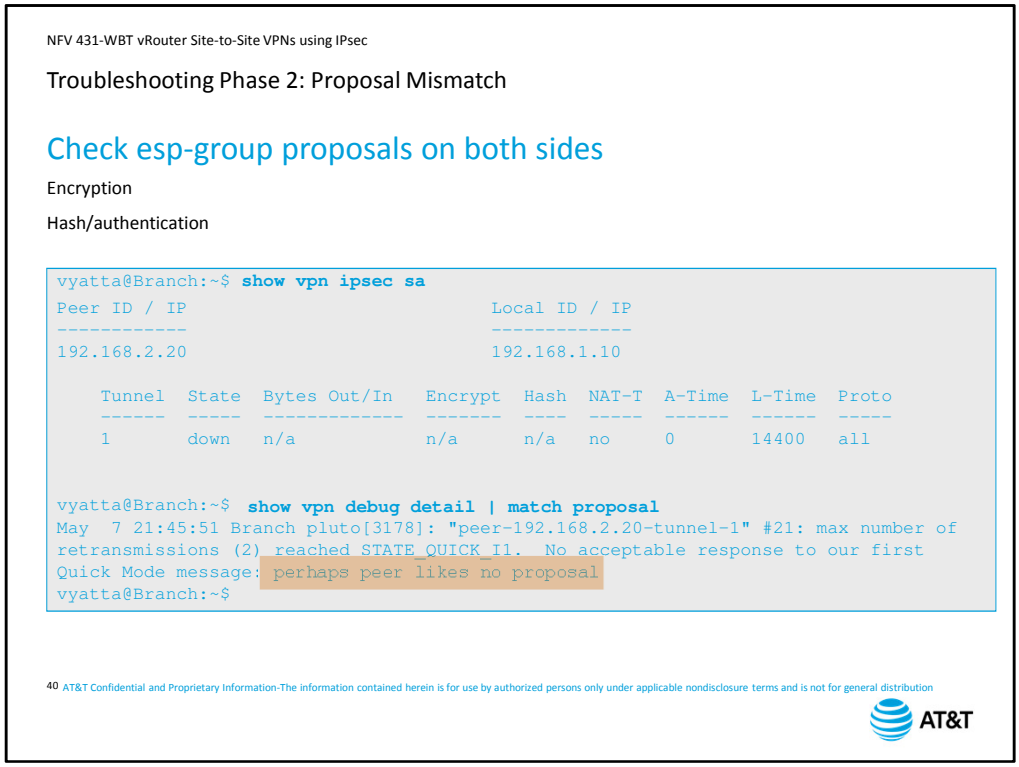
Another common Phase 1 problem is a mismatch of peer IP addresses.
Again, the first indication is that Phase 1 has not completed successfully. If we run the previous debug search and see that Phase 1 proposals are not the problem,
We search through the `vpn debug` output again, this time looking for the word *authorized*, because only recognized peer addresses are authorized.
When we look at the output, we see that the peer IP expected according to the SA does not match the peer IP we have received.
In this case, the debug message we are looking for will only appear on the receiving peer. Be sure to look at both peers, since you will not know which peer is initiating the Phase 1 negotiations.
You will need to check the local IP and peer IP address configuration on both sides of the connection.

Troubleshooting Phase 1: Preshared Key Mismatch

Key verification fails – shows up as malformed payload in debug
Check preshared keys on both peers

```
vyatta@Branch:~$ show vpn ike sa
Local IP      Peer IP       State Encrypt Hash NAT-T  A-Time L-Time
--------      -------       ----- ------- ---- ------ ------ -----
192.168.1.10  192.168.2.20  init  n/a     n/a  no          0 28800
vyatta@Branch:~$  show vpn debug detail | match preshared
May  7 21:56:10 Branch pluto[3178]: "peer-192.168.2.20-tunnel-1" #6: probable
authentication failure (mismatch of preshared secrets?): malformed payload in
packet
```

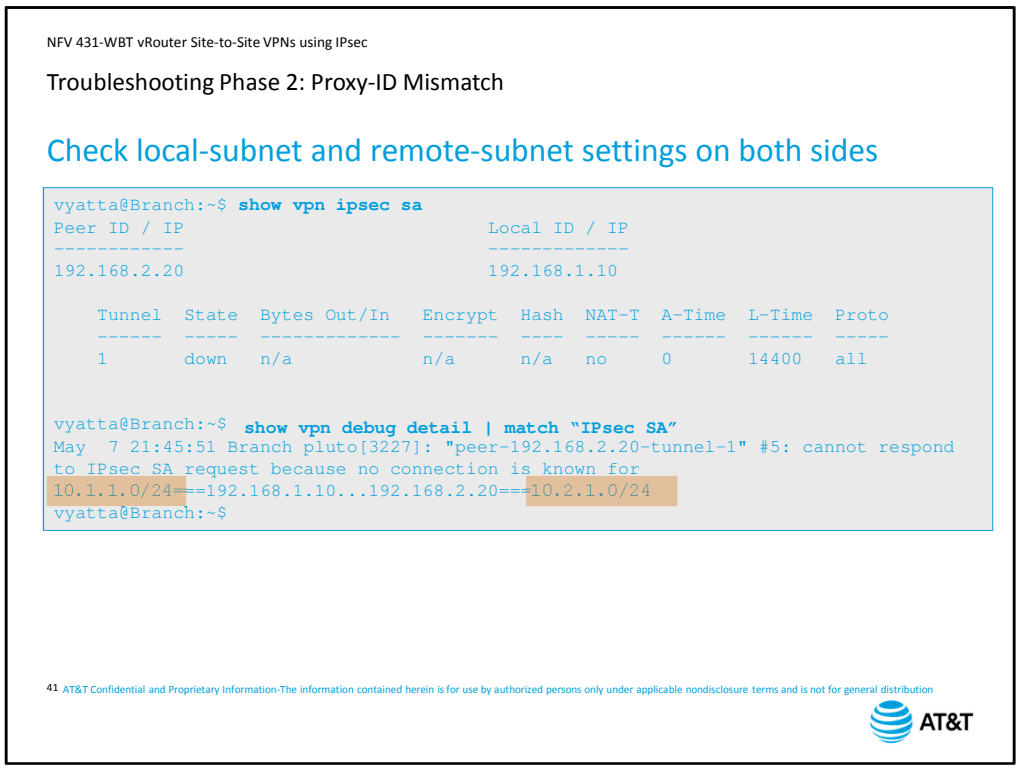

If the problem is a mismatched preshared key, once again the SA will not exist.
When we search the debug output this time, we search for the word *preshared*.
The effect of a mismatched key is that key verification fails. However, the debug provides us with a hint to look at the preshared keys.
Don't forget to check both tunnel peers.



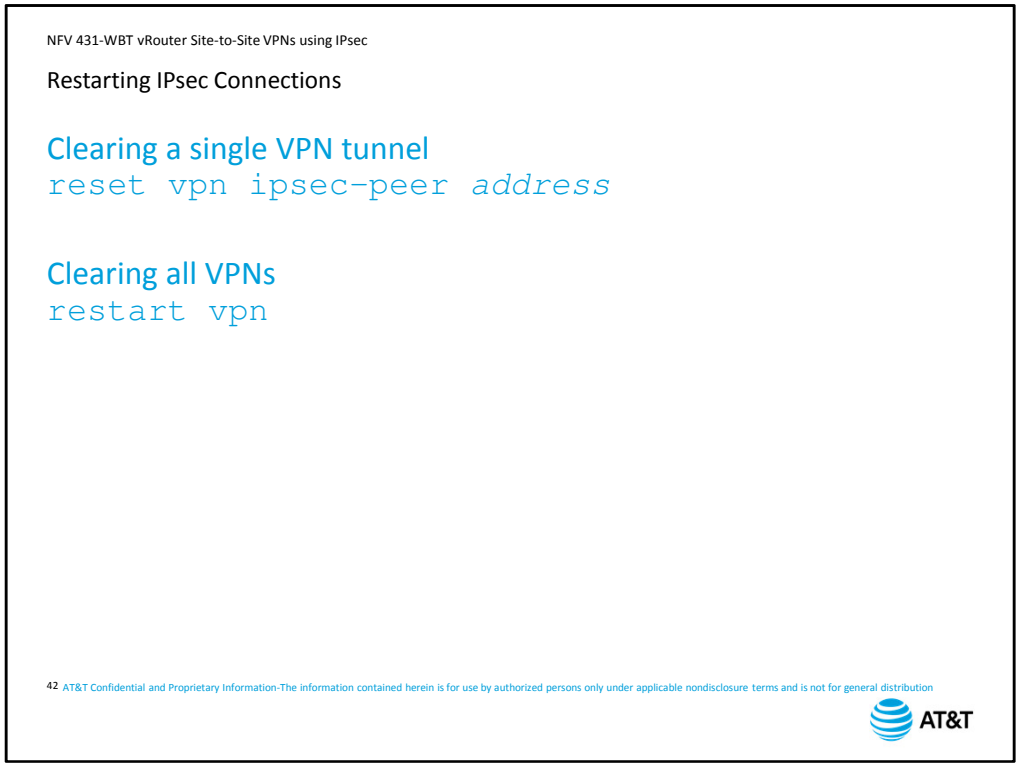
If Phase 1 is completing successfully, as indicated by a Phase 1 SA, but the tunnel still is not coming up, check the Phase 2 SAs. In this case, we have no SAs, indicating that Phase 2 has not completed successfully.

We check the debug output for the word *proposal*, this time in all lowercase. The output shows us that no Phase 2 proposal was selected.

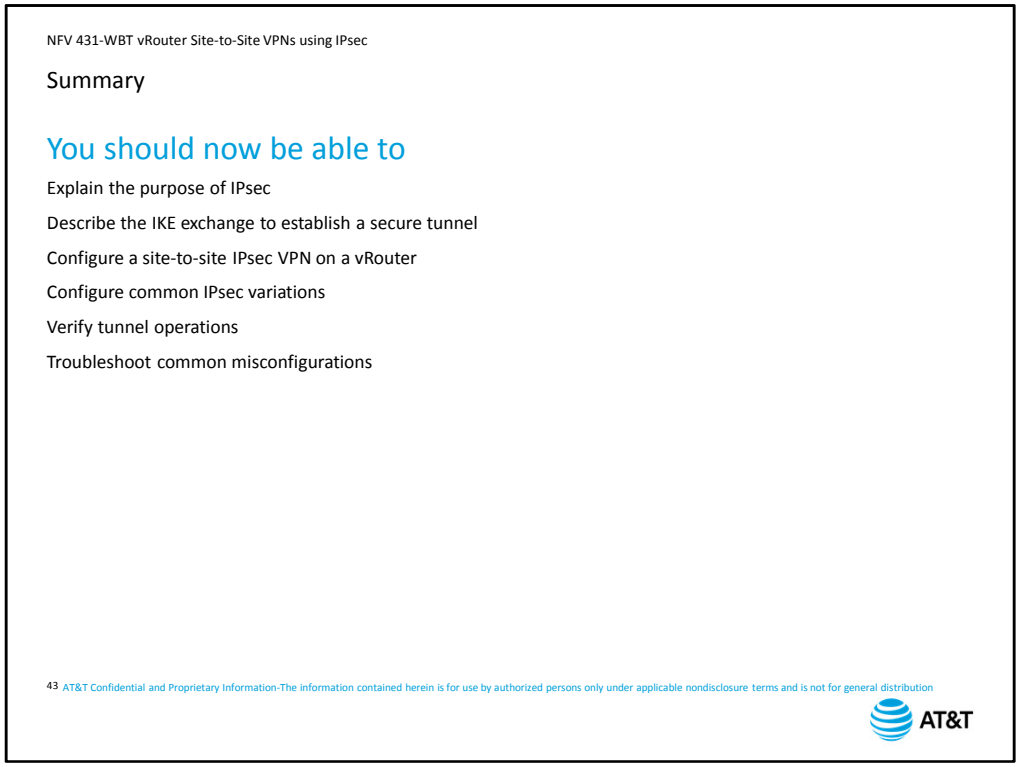
The next step is to check the Phase 2 proposals on both sides of the connection, checking both encryption and hash algorithms.



If proposals are not the problem at Phase 2, our next step is to check the proxy ID by searching for the string *IPsec SA,* and checking the message to see what local and remote subnets we have received from our peer for this tunnel.
We can then compare these values with the configured values for the tunnel, either by using the `show vpn ipsec sa detail` command, or by looking at the configuration.



Once you have made configuration changes, you may need to restart IPsec.
You can clear a single VPN tunnel with the `reset vpn ipsec-peer` command, specifying the IP address of the remote peer.
You can clear all IPsec VPNs on the device by restarting the VPN IPsec process with the `restart vpn` command.



This concludes the AT&T Vyatta 5600 vRouter Site-to-site VPNs using IPsec course.
You should now be able to:

- Explain the purpose of IPsec
- Describe the IKE exchange to establish a secure tunnel
- Configure a site-to-site IPsec VPN on a vRouter
- Configure common IPsec variations
- Verify tunnel operations
- Troubleshoot common misconfigurations

We hope that this information has been useful to you, and that you will take additional AT&T University courses in the future.
Thank you.

End of Course – Site-to-Site VPNs using IPsec