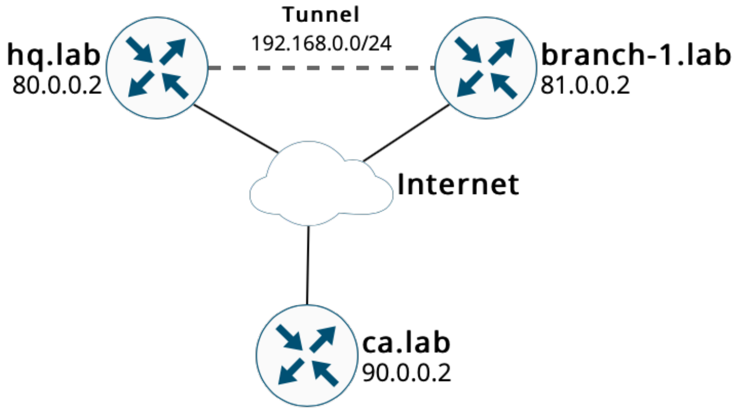
Configuring Basic Site-to-Site VPNs

Using PKI Authentication

- Using pre-shared keys becomes infeasible as VPN deployments grow.
- A more scalable solution is enrolling devices in PKI for authentication
- We use a publicly available CA to simplify the demonstration. Not a great idea in production!



Step 0. The Groundwork

The first step is configuring a CA, if you haven't already.

`ca(config)# crypto key generate rsa modulus 2048 label CA`	Create CA keypair

`ca(config)# ip http server`	Enable HTTP server for CA

`ca(config)# access-list 42 permit 80.0.0.0 0.0.0.255`
`ca(config)# access-list 42 permit 81.0.0.0 0.0.0.255`
`ca(config)# access-list 42 permit 90.0.0.0 0.0.0.255`
`ca(config)# ip http access-class 42`
	Optionally restrict traffic to CA

`ca(config)# crypto pki server CA`
`ca(cs-server)# issuer-name CN=ca.lab`
`ca(cs-server)# grant auto`
`ca(cs-server)# no shutdown`
	Configure and start a *very* basic CA server

Step 1. Enroll Devices

We can now enroll other IOS devices with the CA. The **branch-1** configuration is identical to **hq** except for name substitutions.

`hq(config)# ip domain-name lab`
`hq(config)# crypto key generate rsa modulus 2048 label PKI`
	Generate an RSA keypair to be enrolled in the PKI

`hq(config)#crypto pki trustpoint CA`
`hq(ca-trustpoint)# rsakeypair PKI`
`hq(ca-trustpoint)# enrollment url http://90.0.0.2`
`hq(ca-trustpoint)# subject-name CN=hq.lab`
`hq(ca-trustpoint)# fqdn hq.lab`
`hq(ca-trustpoint)# fingerprint HEX_STRING`
	Configure how we will enroll with the CA. The fingerprint is optional but will reduce the interactive prompts during enrollment. You can find it by running **`show crypto pki server`** on the CA.

`hq(config)# crypto pki authenticate CA`
`hq(config)# crypto pki enroll CA`
	Barring any typos, you can now authenticate and enroll with the CA server.

Step 2. The IPSec Stuff

Now that **hq** and **branch-1** have certificates issued by the same CA, we can use certificates instead of PSKs for authentication.

```
hq(config)# crypto ikev2 profile IKEV2_PROF
hq(config-ikev2-prof)# authentication local rsa-sig
hq(config-ikev2-prof)# authentication remote rsa-sig
hq(config-ikev2-prof)# identity local fqdn hq.lab
hq(config-ikev2-prof)# match identity remote fqdn branch-1.lab
hq(config-ikev2-prof)# pki trustpoint CA

branch-1(config)#crypto ikev2 profile IKEV2_PROF
branch-1(config-ikev2-prof)# authentication local rsa-sig
branch-1(config-ikev2-prof)# authentication remote rsa-sig
branch-1(config-ikev2-prof)# identity local fqdn branch-1.lab
branch-1(config-ikev2-prof)# match identity remote fqdn hq.lab
branch-1(config-ikev2-prof)# pki trustpoint CA
```

In the **IKEv2 profile**, we set the local and remote authentication methods and identifiers.

We also specify the PKI trustpoint (CA) that should be shared across devices.

```
hq(config)# crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
hq(cfg-crypto-trans) mode tunnel

branch-1(config)# crypto ipsec transform-set TS esp-aes 256 esp-sha256-hmac
branch-1(cfg-crypto-trans) mode tunnel
```

The **transform set** defines how traffic is protected.

```
hq(config)# crypto ipsec profile IPSEC_PROF
hq(ipsec-prof)# set transform-set TS
hq(ipsec-prof)# set ikev2-profile IKEV2_PROF

branch-1(config)# crypto ipsec profile IPSEC_PROF
branch-1(ipsec-prof)# set transform-set TS
branch-1(ipsec-prof)# set ikev2-profile IKEV2_PROF
```

The **IPsec profile** joins the IKEv2 profile and transform set. It is what will be applied to the tunnel interface.

```
hq(config)# interface tunnel 0
hq(config-if)# ip address 192.168.0.1 255.255.255.0
hq(config-if)# tunnel mode ipsec ipv4
hq(config-if)# tunnel protection ipsec profile IPSEC_PROF
hq(config-if)# ip mtu 1400
hq(config-if)# ip tcp adjust-mss 1360
hq(config-if)# tunnel source gigabitEthernet 0/3
hq(config-if)# tunnel destination 81.0.0.2

branch-1(config)# interface tunnel 0
branch-1(config-if)# ip address 192.168.0.2 255.255.255.0
branch-1 (config-if)# tunnel mode ipsec ipv4
branch-1 (config-if)# tunnel protection ipsec profile IPSEC_PROF
branch-1 (config-if)# ip mtu 1400
branch-1 (config-if)# ip tcp adjust-mss 1360
branch-1 (config-if)# tunnel source gigabitEthernet 0/3
branch-1 (config-if)# tunnel destination 80.0.0.2
```

We create a new tunnel interface and apply the IPsec profile to it.

Adjusting the MTU and MSS are optional but recommended.