GRE Over IPsec for Secure Tunneling

Yuriy Andamasov - 2018-10-13 - 0 Comments - in Interfaces

Introduction

In addition to supporting OpenVPN site-to-site and plain IPsec site-to-site, you may also wish to run a tunneling protocol over an IPsec site-to-site connection, such as GRE, IP in IP, etc. The advantage of this over just using a plain site-to-site IPsec tunnel is that the tunnel gets associated with a dedicated interface (usually `tun0` or similar), whereas "plain" IPsec does not create a dedicated interface for the tunnel and instead peers with a specified address.

Note: for the remainder of this article, unless otherwise specified, we will be using GRE as our encapsulation protocol.

There are effectively three ways of structuring such a secure tunnel:

- Setup a simple GRE tunnel and tell IPsec to encrypt it
- Make a loopback device on both hosts and source the tunnel from the loopback, assigning an address to the loopback that is used as the source address instead of the local address on the hosts (useful if it has a dynamic IP)
- Use a Virtual Tunnel Interface (VTI)

The configuration differences between the three are minimal, and the bulk of the setup is for IPsec itself.

The first solution is simple, but comes with a couple of caveats:

- if the IPsec connection breaks, the GRE tunnel will be unencrypted and sent in the clear.
- if one end uses dynamic addressing (i.e., not a static IP address), or is behind NAT, IPsec isn't able to function

The second solution requires a little bit more setup, but only allows for communication over IPsec. The use of a dummy address as the source IP means this solution solves the dynamic address/NAT problems of the simple solution, but does come with a few downfalls itself:

- if the IPsec connection isn't up, communication over GRE between the routers is impossible
- If *both* ends are NAT-ed, pre-shared secret IPsec authentication is impossible, requiring the use of either RSA keys or X.509 certificates for authentication (which is

a blessing-in-disguise due to the insecurities of pre-shared keys in general)

The third solution, using a VTI, is slightly outside of the scope of this article, and will be covered in a future article. However, VTIs are essentially IPIP tunnel interfaces that get bound to IPsec connections, solving the first solution's issue of potentially sending unencrypted traffic should the IPsec connection fail. They are (marginally) easier to set up than the loopback device, but don't allow the choice of the encapsulation protocol.

For further reading on the difference between these methods, check out this entry on the VyOS blog: On security of GRE/IPsec scenarios.

Configuration Using Just GRE/IPsec

Assuming we know both routers have a static IP, are not NATed, the configuration for GRE over IPsec (or, again, any other encapsulation protocol) is relatively simple, only requiring a few more configuration entries over the plain IPsec site-to-site configuration.

Topologies
This configuration is based off of the one detailed in the above linked IPsec site-to-site VPN article, and assumes the same names and addresses:

For `central-office-net`:

- Public address of `203.0.113.2` on interface eth1
- Local private network of `10.1.1.0/24`
- A VyOS router called `central-office-rtr`

For `remote-office-net`:

- Public address of `192.51.100.2` on interface `eth1`
- Local private network of `10.2.2.0/24`
- A VyOS router called `remote-office-rtr`

Like the other config, we will be using the pre-shared secret method for authentication; the key is `not-so-secret`.

Configuration of central-office-rtr:
Firstly, we need to set up a GRE tunnel:

```
set interfaces tunnel tun0 encapsulation gre
set interfaces tunnel tun0 local-ip 203.0.113.2
set interfaces tunnel tun0 remote-ip 192.51.100.2
set interfaces tunnel tun0 address 10.1.1.5/32
```

IPsec configuration follows the other example exactly until specifying the tunnel. Instead of running the `set vpn ipsec peer <name> tunnel` commands in the plain IPsec example article, run this instead:

```
set vpn ipsec site-to-site peer 192.51.100.2 tunnel 1 protocol gre
```

This tells IPsec to encrypt the GRE traffic between the two networks.

The full IPsec example configuration looks like this (from the `vpn ipsec` configuration tree:

```
edit vpn ipsec
#[edit vpn ipsec]
set esp-group central-rtr-esp compression 'disable'
set esp-group central-rtr-esp lifetime '1800'
set esp-group central-rtr-esp mode 'tunnel'
set esp-group central-rtr-esp pfs 'enable'
set esp-group central-rtr-esp proposal 1 encryption 'aes256'
set esp-group central-rtr-esp proposal 1 hash 'sha256'
set ike-group central-rtr-ike ikev2-reauth 'no'
set ike-group central-rtr-ike key-exchange 'ikev1'
set ike-group central-rtr-ike lifetime '3600'
set ike-group central-rtr-ike proposal 1 encryption 'aes256'
set ike-group central-rtr-ike proposal 1 hash 'sha256'
set ipsec-interfaces interface 'eth1'
set site-to-site peer 192.51.100.2 authentication mode 'pre-shared-
secret'
set site-to-site peer 192.51.100.2 authentication pre-shared-secret
'not-so-secret'
set site-to-site peer 192.51.100.2 ike-group 'central-rtr-ike'
set site-to-site peer 192.51.100.2 default-esp-group 'central-rtr-
esp'
set site-to-site peer 192.51.100.2 local-address '203.0.113.2'
set site-to-site peer 192.51.100.2 tunnel 0 protocol gre
```

Configuration of remote-office-rtr:
Essentially the same as `central-office-rtr` with the appropriate addresses changed:

```
set interfaces tunnel tun0 encapsulation gre
set interfaces tunnel tun0 local-ip 192.51.100.2
set interfaces tunnel tun0 remote-ip 203.0.113.2
set interfaces tunnel tun0 address 10.2.2.5/32
```

Similarly for IPsec:

```
edit vpn ipsec
#[edit vpn ipsec]
set esp-group remote-rtr-esp compression 'disable'
set esp-group remote-rtr-esp lifetime '1800'
```

```
set esp-group remote-rtr-esp mode 'tunnel'
set esp-group remote-rtr-esp pfs 'enable'
set esp-group remote-rtr-esp proposal 1 encryption 'aes256'
set esp-group remote-rtr-esp proposal 1 hash 'sha256'
set ike-group remote-rtr-ike ikev2-reauth 'no'
set ike-group remote-rtr-ike key-exchange 'ikev1'
set ike-group remote-rtr-ike lifetime '3600'
set ike-group remote-rtr-ike proposal 1 encryption 'aes256'
set ike-group remote-rtr-ike proposal 1 hash 'sha256'
set ipsec-interfaces interface 'eth1'
set site-to-site peer 203.0.113.2 authentication mode 'pre-shared-
secret'
set site-to-site peer 203.0.113.2 authentication pre-shared-secret
'not-so-secret'
set site-to-site peer 203.0.113.2 ike-group 'remote-rtr-ike'
set site-to-site peer 203.0.113.2 default-esp-group 'remote-rtr-esp'
set site-to-site peer 203.0.113.2 local-address '192.51.100.2'
set site-to-site peer 203.0.113.2 tunnel 0 protocol gre
```

There should now be a functioning GRE tunnel between the two routers that is encrypted using IPsec.

Configuring GRE/IPsec Using a Loopback Interface

We will be using the same network setup from above, except that we no longer know the public IP address of `remote-office-net`. As a result, we have to use VyOS' feature of IDs in lieu of remote addresses, which also requires us to use RSA or X.509 authentication. We will be using RSA for this example.

Additionally, we will be using the dummy interface `dum0` on both routers as our loopback interface, and `central-office-rtr` will have the address `192.168.1.100` for its dummy interface, and `remote-office-rtr` will use `192.168.1.200`.

This VyOS blog entry may also be of interest: [How to setup an IPsec connection between two NATed peers: using id's and RSA keys.]

Generate RSA Keys on Both Routers:

The IPsec setup is a little bit more complicated. Because we have to use IDs instead of IP addresses for the remote address, we'll also have to set up our preferred authentication method (either RSA or X.509 certificates; we'll be using RSA) instead of using a pre-shared secret. For RSA, we have to generate a key pair on both routers. This is done from operational mode with the command:

```
vyos@central-office-rtr:~$ generate vpn rsa-key bits 4096 random
```

`/dev/urandom`

We also need a key from the remote office:

```
vyos@remote-office-rtr:~$ generate vpn rsa-key bits 4096 random
/dev/urandom
```

(You may also use a 2048 bit key, or use `/dev/random` as your source of entropy, although it's worth noting `/dev/urandom` is not less secure than `/dev/random` and is generally preferred, as it is a bit quicker.)
This should result in the public key being printed to the screen (typically starting with `0sAQ...`). This key will need to be added on `remote-office-rtr`. Similarly, the public key generated by `remote-office-rtr` will need to be added to `central-office-rtr`.

You may import a key on either host through the command:

```
set vpn rsa-keys rsa-key-name <name> rsa-key <key>
```

where <name> is the name of the key (typically, and in our case, the name of the system the key is from), and <key> is the public key string from that machine that was printed when it was made. Importing the keys will be shown in the configuration example for both routers below, which assumes we have generated a key on both routers.

Configuration for central-office-rtr:
Firstly, we have to make our loopback device and configure our tunnel appropriately:

```
set interfaces dummy dum0 address 192.168.1.100/32

set interfaces tunnel tun0 encapsulation gre
set interfaces tunnel tun0 local-ip 192.168.1.100/32
set interfaces tunnel tun0 remote-ip 192.168.1.200/32
```

Now, we need to import the key we generated earlier on the Remote office's router:

```
vyos@central-office-rtr# set vpn rsa-keys rsa-key-name REMOTE-KEY
rsa-key 0sAQ...
```

(RSA key truncated for brevity)

And now to set up IPsec itself:
(**Note:** we'll be using the same `ike-group` and `esp-group` names and settings as the previous configuration example above.) Because the remote office has a dynamic or unknown public IP, we'll need to use an ID for the peer name. The ID can be any string, as long as it's prefixed with an @ symbol. It's essentially a stand-in for an IP address that's useful for when we do not know the exact address of a particular host.

```
set vpn ipsec ipsec-interfaces interface 'eth1'
set vpn ipsec site-to-site peer @REMOTE-OFFICE authentication mode
rsa
set vpn ipsec site-to-site peer @REMOTE-OFFICE authentication rsa-
key-name REMOTE-KEY
set vpn ipsec site-to-site peer @REMOTE-OFFICE default-esp-group
'central-rtr-esp'
set vpn ipsec site-to-site peer @REMOTE-OFFICE ike-group 'central-
rtr-ike'
set vpn ipsec site-to-site peer @REMOTE-OFFICE local-address
203.0.113.2
set vpn ipsec site-to-site peer @REMOTE-OFFICE connection-type
respond
set vpn ipsec site-to-site peer @REMOTE-OFFICE tunnel 1 local prefix
192.168.1.100/32
set vpn ipsec site-to-site peer @REMOTE-OFFICE tunnel 1 remote prefix
192.168.1.200/32
```

The last two lines are to use the dummy interface addresses for the tunnel.

Configuration for remote-office-rtr:
Setting up the dummy interface and tunnel on the remote router:

```
set interfaces dummy dum0 address 192.168.1.200/32

set interfaces tunnel tun0 encapsulation gre
set interfaces tunnel tun0 local-ip 192.168.1.200/32
set interfaces tunnel tun0 remote-ip 192.168.1.100/32
```

Import the key from `central-office-rtr`:

```
vyos@remote-office-rtr# set vpn rsa-keys rsa-key-name CENTRAL-KEY
rsa-key 0sAQ...
```

And now setup IPsec:

```
set vpn ipsec ipsec-interfaces interface 'eth1'
set vpn ipsec site-to-site peer 203.0.113.2 authentication id
@REMOTE-OFFICE
set vpn ipsec site-to-site peer 203.0.113.2 authentication mode rsa
set vpn ipsec site-to-site peer 203.0.113.2 authentication rsa-key-
name CENTRAL-KEY
```

```
set vpn ipsec site-to-site peer 203.0.113.2 authentication remote-id
@CENTRAL-OFFICE
set vpn ipsec site-to-site peer 203.0.113.2 connection-type initiate
set vpn ipsec site-to-site peer 203.0.113.2 default-esp-group
'remote-rtr-esp'
set vpn ipsec site-to-site peer 203.0.113.2 ike-group 'remote-rtr-
ike'
set vpn ipsec site-to-site peer 203.0.113.2 local-address any
set vpn ipsec site-to-site peer 203.0.113.2 tunnel 1 local prefix
192.168.1.200/32
set vpn ipsec site-to-site peer 203.0.113.2 tunnel 1 remote prefix
192.168.1.100/32
```

Your tunnel should now be setup properly.

The key difference between this and the previous setup is that, due to lack of known address on one end, we have to use the additional configuration option `connection-type`. This way, the router setup with the `initiate` connection type will start the connection (since it knows the static IP of the other router) and the other router can then `respond` to it. We also must specify the ID used for the remote office on `central-office-rtr` in the authentication parameters on the `remote-office-rtr` itself.