Routing issue

27.02.2020 11:27 - Brad Griffiths

Status: Feedback
Priority: Normal
Assignee: network / firewall
Category: network / firewall
Affected version: 5.4.0
Resolution:

Description

I hope someone can help, I'm stuck and have spent a few days trying to resolve the issue without success but I feel like I'm close...

This is 2x site-to-site VPN tunnel between Cisco 800 series routers and a Strongswan server that is behind a NAT firewall with UDP 500 and 4500 forwarded to it. The tunnels are up but there's no routing through the Strongswan server. I can ping from Strongswan's console and hit every subnet behind the Cisco routers, I can also ping from behind the Cisco routers to the Strongswan server's IP but no further.

I can confirm that forwarding is enabled in sysctl.conf:

```bash
net.ipv4.ip_forward = 1
```

Ifconfig

```bash
eth0    Link encap:Ethernet    HWaddr 00:16:3E:00:16:63
inet addr:10.10.0.38 Bcast:10.10.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:4954 errors:0 dropped:0 overruns:0 frame:0
TX packets:5536 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:835134 (815.5 KiB) TX bytes:1977918 (1.8 MiB)

lo     Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)

tun0    Link encap:UNSPEC    HWaddr 00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00
inet addr:10.8.0.1 P-t-P:10.8.0.2 Mask:255.255.255.255
UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

IPsec.conf

```bash
config setup
    uniquelds=yes

conn %default
    ike1lifetime=60h
    keylife=20h
    rekeymargin=3h
    keyingtries=1
    keyexchange=ikev2
    mobike=no
    type=tunnel
```
conn qld
leftid=$publicIP
left=0.0.0.0
leftsubnet=10.10.0.0/24
leftfirewall=yes
rightid=$qldpublicIP
right=$qldpublicIP
rightsubnet=192.168.4.0/24
ike=aes128-shal-modp1536
esp=aes128-shal
authby=secret
auto=start
aggressive=no
dpdaction=none
keyexchange=ikev2
forceencaps=yes

conn nsw
leftid=$publicIP
left=0.0.0.0
leftsubnet=10.10.0.0/24
leftfirewall=yes
rightid=$nswpublicIP
right=$nswpublicIP
rightsubnet=192.168.50.0/24
ike=aes128-shal-modp1536
esp=aes128-shal
authby=secret
auto=start
aggressive=no
dpdaction=none
keyexchange=ikev2
forceencaps=yes

# work around for multiple rightsubnet's not coming up
conn net-192.1.1.0
also=nsw
rightsubnet=192.1.1.0/24 # subnet to be removed after project.
auto=start

conn net-192.168.2.0
also=nsw
rightsubnet=192.168.2.0/24
auto=start

conn net-192.168.10.0
also=nsw
rightsubnet=192.168.10.0/24
auto=start

conn net-192.168.11.0
also=nsw
rightsubnet=192.168.11.0/24
auto=start

conn net-192.168.12.0
also=nsw
rightsubnet=192.168.12.0/24
auto=start

iptables -L

Chain INPUT (policy ACCEPT)
target prot opt source destination

Chain FORWARD (policy ACCEPT)
ACCEPT all -- 192.168.12.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 7
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.12.0/24 policy match dir out pol ipsec reqid 7
proto esp
ACCEPT all -- 192.168.11.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 6
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.11.0/24 policy match dir out pol ipsec reqid 6
proto esp
ACCEPT all -- 192.168.10.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 5
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.10.0/24 policy match dir out pol ipsec reqid 5
proto esp
ACCEPT all -- 192.168.2.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 4
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.2.0/24 policy match dir out pol ipsec reqid 4
proto esp
ACCEPT all -- 192.1.1.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 3
proto esp
ACCEPT all -- 10.10.0.0/24 192.1.1.0/24 policy match dir out pol ipsec reqid 3
proto esp
ACCEPT all -- 192.168.50.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 2
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.50.0/24 policy match dir out pol ipsec reqid 2
proto esp
ACCEPT all -- 192.168.4.0/24 10.10.0.0/24 policy match dir in pol ipsec reqid 1
proto esp
ACCEPT all -- 10.10.0.0/24 192.168.4.0/24 policy match dir out pol ipsec reqid 1
proto esp

Chain OUTPUT (policy ACCEPT)
target prot opt source destination

ip route list table 220

192.168.4.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.50.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.2.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.1.1.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.12.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.11.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.10.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38
192.168.10.0/24 via 10.10.0.253 dev eth0 proto static src 10.10.0.38

strongswan statusall

Status of IKE charon daemon (strongSwan 5.4.0, Linux 2.6.32-573.22.1.el6.x86_64, x86_64):
uptime: 2 hours, since Feb 27 15:53:01 2020
malloc: sbrk 536576, mmap 0, used 395568, free 141008
worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 6
loaded plugins: charon aes des rc2 sha2 shal md4 md5 random nonce x509 revocation constraints acert pubkey pkcs1 pkcs8 ppg dnskey sshkey pem gencrypt fips-prf gmp xcbc cmac hmac ctr ccm gcm curl a ttr kernel-netlink resolve socket-default farp stroke vici updown eap-identity eap-md5 eap-gtc eap -mschapv2 eap-tls eap-ttls eap-peap xauth-generic xauth-eap xauth-pam xauth-noauth
Listening IP addresses:
10.10.0.38
10.8.0.1
Connections:
qld: 0.0.0.0...$qldpublicIP IKEv2
qld: local: [{$publicIP}] uses pre-shared key authentication
qld: remote: [{$qldpublicIP}] uses pre-shared key authentication
qld: child: 10.10.0.0/24 === 192.168.4.0/24 TUNNEL
nsw: 0.0.0.0...$nswpublicIP IKEv2
nsw: local: [{$publicIP}] uses pre-shared key authentication
nsw: remote: [{$nswpublicIP}] uses pre-shared key authentication
nsw: child: 10.10.0.0/24 === 192.168.50.0/24 TUNNEL

19.03.2020
net-192.1.1.0: child: 10.10.0.0/24 === 192.1.1.0/24 TUNNEL
net-192.168.2.0: child: 10.10.0.0/24 === 192.168.2.0/24 TUNNEL
net-192.168.10.0: child: 10.10.0.0/24 === 192.168.10.0/24 TUNNEL
net-192.168.11.0: child: 10.10.0.0/24 === 192.168.11.0/24 TUNNEL
net-192.168.12.0: child: 10.10.0.0/24 === 192.168.12.0/24 TUNNEL

Security Associations (2 up, 0 connecting):
  nsw[2]: ESTABLISHED 2 hours ago, 10.10.0.38[$publicIP]...$nswpublicIP[$nswpublicIP]
  nsw[2]: IKEv2 SPIs: 46c0b19f93bfa9a8_i* 8a01cb01c438e167_r, pre-shared key reauthentication
    on in 21 minutes
  nsw[2]: IKE proposal: AES_CBC_128/IEEE_SHA1_96/PRF_HMAC_SHA1/MODP_1536

  net-192.1.1.0(22): INSTALLED, TUNNEL, reqid 3, ESP in UDP SPIs: cad2bd6d_i 46c51c0_0
  net-192.1.1.0(22): AES_CBC_128/IEEE_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 28 minutes

  net-192.1.1.0(22): 10.10.0.0/24 === 192.1.1.0/24

  net-192.168.2.0(28): INSTALLED, TUNNEL, reqid 4, ESP in UDP SPIs: c39937e1_i 8b0618b8_o
  net-192.168.2.0(28): AES_CBC_128/IEEE_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 40 minutes

  qld[1]: ESTABLISHED 2 hours ago, 10.10.0.38[$publicIP]...$qldpublicIP[$qldpublicIP]
  qld[1]: IKEv2 SPIs: 3abad3dd49feb352_i* 8dd4b1b7dd1a61b_r, pre-shared key reauthentication
    on in 20 minutes
  qld[1]: IKE proposal: AES_CBC_128/IEEE_SHA1_96/PRF_HMAC_SHA1/MODP_1536

  net-192.168.10.0(25): INSTALLED, TUNNEL, reqid 5, ESP in UDP SPIs: c98592f8_i fd2bad4b_o

  net-192.168.2.0(28): 10.10.0.0/24 === 192.168.10.0/24

  net-192.168.2.0(28): AES_CBC_128/IEEE_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 40 minutes

  net-192.168.2.0(28): 10.10.0.0/24 === 192.168.2.0/24

  qld[1]: 10.10.0.0/24 === 192.168.4.0/24

  qld[2]: INSTALLED, TUNNEL, reqid 1, ESP in UDP SPIs: c597f73f_i 07697ac4_o
  qld[2]: AES_CBC_128/IEEE_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 34 minutes
  qld[2]: 10.10.0.0/24 === 192.168.4.0/24

History

#1 - 27.02.2020 15:53 - Tobias Brunner
- Status changed from New to Feedback

See ForwardingAndSplitTunneling.

#2 - 27.02.2020 23:44 - Brad Griffiths

Thank you for the reference, I'd looked over this document previously during the troubleshooting which did help with the config (leftfirewall=yes was a huge help with the routes) and have had another read through it again. It's possible I've missed something that's glaringly obvious to someone else.

Looking through the link I can confirm the following have been applied to this server:

- Forwarding is enabled with (net.ipv4.ip_forward=1)
- There's no NAT being performed at the moment by this server (iptables -t nat -L shows no entries)
- Forwarding rules are in iptables (provided by leftfirewall=yes)

My understanding of MTU/MSS might be wrong but if I can ping eth0 of Strongswan from the other side of the VPN tunnel this shouldn't need to be? I believe by this stage the packet is unpacked and ready to send onto the LAN?

I have set forceencaps=yes in ipsec.conf assuming GRE isn't making it through to Strongswan.

#3 - 28.02.2020 09:22 - Tobias Brunner

- Forwarding is enabled with (net.ipv4.ip_forward=1)

There is more to it than that (see blow).

- There's no NAT being performed at the moment by this server (iptables -t nat -L shows no entries)
Then how about the routing inside the network(s) behind the VPN server? Is the VPN server the default gateway? If not, do hosts know to route packets addressed to the remote networks back to the VPN server?

- Forwarding rules are in `iptables` (provided by `leftfirewall=yes`)

These are only necessary if you use a DROP policy for the FORWARD chain (or you have other rules that would otherwise drop the VPN traffic). That isn't the case here.

    My understanding of MTU/MSS might be wrong but if I can ping eth0 of Strongswan from the other side of the VPN tunnel this shouldn't need to be?

Could depend on the MTU beyond your VPN server. But pings are not affected by this anyway, unless you manually increase their size.

    I have set `forceencaps=yes` in `ipsec.conf` assuming GRE isn't making it through to Strongswan.

Forcing UDP encapsulation is only necessary if there is no NAT (i.e. in your case NAT traversal will be used anyway) and if plain ESP packets don't get through for some reason. But why GRE? How do you figure that comes into play?