strongSwan - Issue #2726

Strongswan selects wrong source IP

22.08.2018 11:37 - Ralf O

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:			
Affected version:	5.5.1	**Resolution:**	

Description

Using the parameter left=%any works well until there are alias IPs at the interface.
With Linux you can use more than one IP at one interface by using the commands:
ifconfig eth0:2 10.10.10.10/24
ifconfig eth0 10.0.0.1/24

The gateway has the 10.0.0.254. The IPsec tunnel works with the parameter left=10.0.0.1.
But I don't know this IP 10.0.0.1, because it will be set by DHCP. I have to use left=%any or left=%defaultroute. The defaultroute is successfully set to 10.0.0.254 by the DHCP client.
But with left=%any, strongswan will choose the 10.10.10.10 as the source IP, which is wrong. This is not the correct source IP for using the defaultroute.
As you can see from the status page, it tries to connect from 10.10.10.10 to the IPsec server at 3.3.3.3 (which is located behind the gateway).

Status of IKE charon daemon (strongSwan 5.5.1, Linux 3.16.51, armv5tejl):
uptime: 10 seconds, since Aug 15 12:41:01 2018
malloc: sbrk 540672, mmap 0, used 164800, free 375872
worker threads: 7 of 16 idle, 5/0/4/0 working, job queue: 0/0/0/0, scheduled: 1
loaded plugins: charon aes des rc2 sha2 sha1 md5 random nonce x509 revocation constraints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf xcbc cmac hmac attr kernel-libipsec kernel-netlink resolve socket-default stroke vici updown xauth-generic
Listening IP addresses:
10.10.10.10
10.0.0.1
Connections:
ipsec0: %any,0.0.0.0/0,::/0...3.3.3.3 IKEv1 Aggressive, dpddelay=30s
ipsec0:   local: [Remote-VPN-Test] uses pre-shared key authentication
ipsec0:   local: [Remote-VPN-Test] uses XAuth authentication: any with XAuth identity 'vpntest'
ipsec0:   remote: [3.3.3.3] uses pre-shared key authentication
ipsec0:   child: dynamic === 10.2.0.0/16 TUNNEL, dpdaction=restart
Security Associations (0 up, 1 connecting):
ipsec0[1]: CONNECTING, 10.10.10.10[Remote-VPN-Test]...3.3.3.3[%any]
ipsec0[1]: IKEv1 SPIs: 68a3660f1f925fd3_i* 0000000000000000_r
ipsec0[1]: Tasks queued: QUICK_MODE
ipsec0[1]: Tasks active: ISAKMP_VENDOR ISAKMP_CERT_PRE AGGRESSIVE_MODE ISAKMP_CERT_POST ISAKMP_NATD

I'm wondering why strongswan does not use the TCP-Stack/Kernel (in my case 3.16.51), like any other program that uses sockets to determine its source IP. Why does it need to determine its source IP on its (faulty) own.
It seems that strongswan uses the IP, that is first set at the interface. If I set ifconfig eth0 10.0.0.1 first, it will choose this.

The weird thing is, that it works by using left=abc, which leads to a DNS request of abc. And after this, strongswan chooses the correct source IP, the 10.0.0.1.
But there is a disadvantage of this solution: If the DNS server has a wildcard match, it answers the request for abc and ipsec fails to connect because it sets this DNS answer to its own source IP.

History

#1 - 22.08.2018 12:32 - Tobias Brunner

- Tracker changed from Bug to Issue

- Status changed from New to Feedback

- Start date deleted (22.08.2018)

- Affected version changed from 5.6.3 to 5.5.1

> But with left=%any, strongswan will choose the 10.10.10.10 as the source IP, which is wrong. This is not the correct source IP for using the defaultroute.

Check your routes (and don't use ifconfig or route use ip ...).

Also check other issues and wiki pages related to this (e.g. in regards to strongSwan source address lookup and how to use the kernel's native implementation).

#2 - 22.08.2018 15:58 - Ralf O

ip route did not show me any misleading routes.
But adding these 2 parameters to strongswan.conf fixes my problem:

```
charon {
  plugins {
    kernel-netlink {
      fwmark = !0x42
    }
    socket-default {
      fwmark = 0x42
    }
  }
}
```

Thanks!

#3 - 22.08.2018 16:00 - Tobias Brunner

> ip route did not show me any misleading routes.

Why don't you show us what it actually shows. Sigh.

> But adding these 2 parameters to strongswan.conf fixes my problem:

OK.

#4 - 22.08.2018 16:42 - Ralf O

My device has a bit different IP addresses than the ones I used to create this issue.
But here you are, the gateway is the 172.16.0.1, the device got the 172.16.4.145 per DHCP.

```
#ip route show table all
default via 172.16.0.1 dev eth0
10.10.10.0/24 dev eth0  proto kernel  scope link  src 10.10.10.10
172.16.0.0/20 dev eth0  proto kernel  scope link  src 172.16.4.145
broadcast 10.10.10.0 dev eth0  table local  proto kernel  scope link  src 10.10.10.10
local 10.10.10.10 dev eth0  table local  proto kernel  scope host  src 10.10.10.10
broadcast 10.10.10.255 dev eth0  table local  proto kernel  scope link  src 10.10.10.10
broadcast 127.0.0.0 dev lo  table local  proto kernel  scope link  src 127.0.0.1
local 127.0.0.0/8 dev lo  table local  proto kernel  scope host  src 127.0.0.1
local 127.0.0.1 dev lo  table local  proto kernel  scope host  src 127.0.0.1
broadcast 127.255.255.255 dev lo  table local  proto kernel  scope link  src 127.0.0.1
broadcast 172.16.0.0 dev eth0  table local  proto kernel  scope link  src 172.16.4.145
local 172.16.4.145 dev eth0  table local  proto kernel  scope host  src 172.16.4.145
broadcast 172.16.15.255 dev eth0  table local  proto kernel  scope link  src 172.16.4.145

#ip route get 3.3.3.3
3.3.3.3 via 172.16.0.1 dev eth0  src 172.16.4.145
    cache
```

(logs done without fwmark parameter an while strongswan tries to connect from 10.10.10.10)

#5 - 23.08.2018 13:38 - Tobias Brunner

> My device has a bit different IP addresses than the ones I used to create this issue.
> But here you are, the gateway is the 172.16.0.1, the device got the 172.16.4.145 per DHCP.
>
> [...]
>
> (logs done without fwmark parameter an while strongswan tries to connect from 10.10.10.10)

OK, thanks. Assuming the default route is the only one that matches the destination address, and because it does not list a source IP the route lookup will first fallback to using an IP address on the interface assigned to the route (i.e. basically the first one that's found there, which you already noted). However, there is another fallback to use the gateway for a lookup (currently requires a recursive route lookup that dumps all the routes), which would have found the correct source IP via the *172.16.0.0/20* route. Actually, there is a patch in the *kernel-netlink-prefer-gw* branch that changes the order of these lookups (439637bf1b5e), the problem is that it's not very efficient to do this lookup in the way it is implemented right now.