I'm trying to use strongswan with kernel-libipsec on linux (3.9.4) but unfortunately it's not working. Here is the relevant part of "ipsec restart --nofork" output:

```
05[IKE] authentication of 'XxXxX' with RSA successful
05[IKE] IKE_SA XxXxX[1] established between FOO...BAR
05[IKE] scheduling reauthentication in 10002s
05[IKE] maximum IKE_SA lifetime 10542s
05[ENC] generating QUICK_MODE request 3912695104 [ HASH SA No KE ID ID ]
02[NET] received packet: from X.W.Y.Z[4500] to A.B.C.D[4500] (324 bytes)
02[ENC] parsed QUICK_MODE response 3912695104 [ HASH SA No KE ID ID N((24576)) ]
02[ESP] failed to create ESP context: unsupported encryption algorithm
02[ESP] failed to create SAD entry
02[ESP] failed to create ESP context: unsupported encryption algorithm
02[ESP] failed to create SAD entry
02[IKE] unable to install inbound and outbound IPsec SA (SAD) in kernel
02[ENC] generating QUICK_MODE request 1737399506 [ HASH SA No KE ID ID ]
15[ENC] parsed QUICK_MODE response 1737399506 [ HASH SA No KE ID ID N((24576)) ]
15[ESP] failed to create ESP context: unsupported encryption algorithm
15[ESP] failed to create SAD entry
15[ESP] failed to create ESP context: unsupported encryption algorithm
15[ESP] failed to create SAD entry
15[ESP] failed to create ESP context: unsupported encryption algorithm
15[ESP] failed to create SAD entry
15[IKE] unable to install inbound and outbound IPsec SA (SAD) in kernel
15[ENC] generating QUICK_MODE request 2932897482 [ HASH SA No KE ID ID ]
16[NET] received packet: from X.W.Y.Z[4500] to A.B.C.D[4500] (324 bytes)
16[ENC] parsed QUICK_MODE response 2932897482 [ HASH SA No KE ID ID N((24576)) ]
16[ESP] failed to create ESP context: unsupported encryption algorithm
16[ESP] failed to create SAD entry
16[ESP] failed to create ESP context: unsupported encryption algorithm
16[ESP] failed to create SAD entry
16[ESP] failed to create ESP context: unsupported encryption algorithm
16[ESP] failed to create SAD entry
16[IKE] unable to install inbound and outbound IPsec SA (SAD) in kernel
16[ENC] generating QUICK_MODE request 502652095 [ HASH SA No KE ID ID ]
06[NET] received packet: from X.W.Y.Z[4500] to A.B.C.D[4500] (324 bytes)
06[ENC] parsed QUICK_MODE response 502652095 [ HASH SA No KE ID ID N((24576)) ]
06[ESP] failed to create ESP context: unsupported encryption algorithm
06[ESP] failed to create SAD entry
06[ESP] failed to create ESP context: unsupported encryption algorithm
06[ESP] failed to create SAD entry
06[IKE] unable to install inbound and outbound IPsec SA (SAD) in kernel
06[ENC] generating INFORMATIONAL_V1 request 3861531521 [ HASH N(NO_PROP) ]
06[ENC] generating INFORMATIONAL_V1 request 3956559828 [ HASH N(NO_PROP) ]
06[ENC] generating INFORMATIONAL_V1 request 3218326822 [ HASH N(NO_PROP) ]
06[ENC] generating INFORMATIONAL_V1 request 4192011303 [ HASH N(NO_PROP) ]
```
When ipsec is running I see that ipsec0 interface is created:

ipsec0:  flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1400
         00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
         RX packets 0  bytes 0 (0.0 B)
         RX errors 0  dropped 0  overruns 0  frame 0
         TX packets 0  bytes 0 (0.0 B)
         TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0

I have tested and strongswan is working just fine when used with kernel-netlink and same configuration.

Do you have any pointers or am I missing something obvious?

Thanks,
Luka

Associated revisions

Revision 3f29ff82 - 12.08.2013 12:21 - Tobias Brunner
libipsec: Don't limit traditional algorithms to AES and SHA1/2
Closes #377.

History

#1 - 09.08.2013 16:12 - Tobias Brunner
- Tracker changed from Issue to Feature
- Subject changed from kernel-libipsec to kernel-libipsec support more algorithms
- Description updated
- Category set to configuration
- Status changed from New to Assigned
- Assignee set to Tobias Brunner

02[ESP] failed to create ESP context: unsupported encryption algorithm
02[ESP] failed to create SAD entry
libipsec currently supports the AES and AES-GCM algorithms only. I suppose 3DES is negotiated here (as it is IKEv1).

There is not really a reason to not support other algorithms too, I guess, but there was simply no need for them so far (and it might even encouraging people to use newer algorithms). Anyway, I'll have a look at it next week.

#2 - 09.08.2013 16:34 - Luka Perkov

Yes, I'm using IKEv1 with 3DES not by choice ;)

This is relavent part from my ipsec.conf file:

```plaintext
esp=3des-shal-modp1024
ike=3des-shal-modp1024
keyexchange=ikev1
```

Thanks for explanation. I can help by testing patches if you need that.

#3 - 12.08.2013 12:29 - Tobias Brunner

- Subject changed from kernel-libipsec support more algorithms to kernel-libipsec: support more algorithms
- Category changed from configuration to libipsec
- Status changed from Assigned to Closed
- Target version set to 5.1.1
- Resolution set to Fixed

The associated commit removes the algorithm limitation in libipsec.

#4 - 12.08.2013 14:11 - Luka Perkov

Hi Tobias,

I have tested your change and now it works just fine. I have one more question and I would really appreciate if you could make some clarifications.

When ipsec is established I see this new interface:

```plaintext
ipsec0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1400
unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
RX packets 3 bytes 252 (252.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3 bytes 252 (252.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

I need to manually add my left IP address on eth0 interface before I start ipsec:

```plaintext
ip addr add A.B.C.D/32 dev eth0
```

I'm dealing with IKEv1 and need to put in place some complex routing rules. Having TUN interface seemed like the easiest approach. I'm wondering if assigning left IP address to ipsec0 interface could be somehow handled automatically? Is that even possible or it does not work that way?

Thanks again.

Luka

#5 - 12.08.2013 14:59 - Tobias Brunner

I'm not sure what you are trying to do exactly. What do you mean with left? Is that the address you use to communicate with the VPN gateway (i.e. the only address on eth0) or some kind of virtual IP address used inside the tunnel. If the latter, you should check out the config options used to assign virtual IP addresses.

I'm dealing with IKEv1 and need to put in place some complex routing rules.

What complex routing rules?

#6 - 12.08.2013 23:18 - Luka Perkov

Tobias Brunner wrote:

If the latter, you should check out the config options used to assign virtual IP addresses.
Yes, that is it. But unfortunately if I put anything in leftsourceip the tunnel is not created.

I'm dealing with IKEv1 and need to put in place some complex routing rules.

What complex routing rules?

For example I need to send through the tunnel all traffic to 1.1.1.1/24 except traffic for addresses 1.1.1.51, 1.1.1.99 and 1.1.1.101. I order for that to work I "manually" configure ip xfrm, routes and iptables. Is there a better way to do it?

#7 - 13.08.2013 10:39 - Tobias Brunner

If the latter, you should check out the config options used to assign virtual IP addresses.

Yes, that is it. But unfortunately if I put anything in leftsourceip the tunnel is not created.

You have to configure matching statements for rightsourceip on the other VPN endpoint.

I'm dealing with IKEv1 and need to put in place some complex routing rules.

What complex routing rules?

For example I need to send through the tunnel all traffic to 1.1.1.1/24 except traffic for addresses 1.1.1.51, 1.1.1.99 and 1.1.1.101. I order for that to work I "manually" configure ip xfrm, routes and iptables. Is there a better way to do it?

Do you want to send traffic to these hosts unencrypted or not at all?

If you use the kernel-netlink plugin you can install passthrough or drop policies (which you can also install manually with ip xfrm).

Something like:

```bash
conn except
  leftsubnet=<local IP>/32
  rightsubnet=1.1.1.51/32,1.1.1.99/32,1.1.1.101/32
  type=passthrough (or drop)
  auto=route
```

This is not yet supported by the kernel-libipsec plugin.

#8 - 13.08.2013 14:41 - Luka Perkov

Tobias Brunner wrote:

If the latter, you should check out the config options used to assign virtual IP addresses.

Yes, that is it. But unfortunately if I put anything in leftsourceip the tunnel is not created.

You have to configure matching statements for rightsourceip on the other VPN endpoint.

On the other end is not strongswan. I'll double check if I can do anything.

I'm dealing with IKEv1 and need to put in place some complex routing rules.

What complex routing rules?

For example I need to send through the tunnel all traffic to 1.1.1.1/24 except traffic for addresses 1.1.1.51, 1.1.1.99 and 1.1.1.101. I order for that to work I "manually" configure ip xfrm, routes and iptables. Is there a better way to do it?

Do you want to send traffic to these hosts unencrypted or not at all?
If you use the `kernel-netlink` plugin you can install `passthrough` or `drop` policies (which you can also install manually with `ip xfrm`).

Something like:

[...]

This is not yet supported by the `kernel-libipsec` plugin.

I have too many of them. It's really easier to have all the `ip xfrm` rules in script. Thanks anyway.