strongSwan - Bug #3541

"unable to install policy" if Windows client reconnects and virtual IPv4 and IPv6 addresses are assigned

17.08.2020 06:49 - Richard Laager

Status:	Closed	**Start date:**	17.08.2020
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	kernel-interface		
Target version:	5.9.2		
Affected version:	5.8.2	**Resolution:**	Fixed

Description

We have a "road warrior" VPN setup. The clients are typically Windows, but we do have Linux, Android, and iOS used a bit too. Twice now we have seen an issue where a client's connection fails. Both times, it was a Windows client, but that may be a coincidence since that is the common case.

This time, the log messages were like this (with IPV6_SUBNET being substituted here for the actual prefix):

```
ipsec[4657]: 06[CFG] unable to install policy IPV6_SUBNET::1f/128 === ::/0 in for reqid 1366, the same policy for reqid 1364 exists
 ipsec[4657]: 06[CFG] unable to install policy IPV6_SUBNET::1f/128 === ::/0 fwd for reqid 1366, the same policy for reqid 1364 exists
 ipsec[4657]: 06[CFG] unable to install policy ::/0 === IPV6_SUBNET::1f/128 out for reqid 1366, the same policy for reqid 1364 exists
```

From what I recall, last time it was IPv4 where the routes were stuck.

We were previously running on Ubuntu 18.04 with strongswan 5.6.2. After the first time, we upgraded to Ubuntu 20.04 with strongswan 5.8.2, but it has just recurred.

Last time, I tried removing the "ip xfrm rules". That didn't fix anything. I restarted the strongswan service and that fixed it. Likewise, restarting the strongswan-starter service fixed it this time too. So it seems that the problem is the in-memory state in strongswan (broadly defined, possibly literally the charon? daemon).

Is there anything we should look for now? More importantly, is there anything we should look for next time before restarting the service?

Associated revisions

Revision 414f2c37 - 18.01.2021 13:58 - Tobias Brunner

mem-pool: Be less strict when reassigning existing online leases

Also assign online leases to a peer connecting from the same endpoint
when it requests any virtual IP. This is mainly a workaround for
Windows clients that remember the virtual IPv6 address and re-request it
the next time the connection is initiated (even if it is not a
reauthentication) but don't do the same for virtual IPv4 addresses.
This can result in duplicate policies with different reqids because
these are allocated for unique sets of traffic selectors.

Fixes #3541.

History

#1 - 17.08.2020 06:59 - Richard Laager

- File ipsec.conf added

I have attached our configuration file, slightly sanitized.

#2 - 17.08.2020 15:48 - Tobias Brunner

- Category changed from charon to kernel-interface

- Status changed from New to Feedback

If there really is an active CHILD_SA with a duplicate policy, the same reqid should get assigned. There might be a weird race condition (previous CHILD_SA gone but policies not yet fully uninstalled - although that should not actually happen as the reqid is released after removing the policies). You'd have to check the log before and around the time when this happens to see what's going on (preferably with log levels for *chd* and *knl* set to 2).

#3 - 17.08.2020 23:00 - Richard Laager

I have raised the logging levels for chd and knl to 2. At this point, we are just waiting for it to recur. If the current pattern holds, it should happen again within a couple of weeks, but who knows; it was stable for a long time before this.

#4 - 18.08.2020 09:14 - Tobias Brunner

You mentioned that you had to restart the daemon to fix this. Does that mean that (re-)connecting with this client/identity (assuming the virtual IP is reassigned based on the identity) resulted in this error repeatedly?

#5 - 18.08.2020 10:04 - Richard Laager

Correct, reconnecting the client is not sufficient. As you expected, the client is reassigned the same IP, so it just keeps hitting the same issue each time. On the most recent failure, the user tried connecting 4 times, got the same address 4 times, and failed 4 times.

The desired reqid increases each time and the reqid that exists stays the same.

In grepping the logs, looks like we other instances of this that I didn't hear about. It looks like this happened on the 29th (twice), 2nd, 5th, 6th, and 13th. Some were IPv4 and some were IPv6.

#6 - 18.08.2020 12:51 - Tobias Brunner

Hm, then a race condition seems unlikely. Sounds more like the kernel interface's tracking of these policies got out of sync for some reason. Let's see if the logs can shed some light on it.

#7 - 02.10.2020 10:36 - Tobias Brunner

Any update on this?

#8 - 02.10.2020 11:25 - Richard Laager

It has not recurred since we raised the debug level, so we're still in a holding pattern.

#9 - 02.10.2020 12:48 - Tobias Brunner

> It has not recurred since we raised the debug level, so we're still in a holding pattern.

I see. If it was some kind of race condition, more logging could definitely change the timing and subsequently prevent the issue.

#10 - 18.11.2020 05:30 - Richard Laager

We got a verified instance of this happening again. The log levels were turned up, as requested. What should I look for in the logs? Is there a way I can send you logs privately?

#11 - 18.11.2020 11:56 - Tobias Brunner

> What should I look for in the logs?

An explanation for the issue, of course.

> Is there a way I can send you logs privately?

Sure, fire away.

#12 - 23.11.2020 15:11 - Tobias Brunner

> I've attached the unredacted logs for the day in question.

Thanks. Note that you have duplicate log messages (once under "ipsec" and once under "charon"), you might want to look into the logger and/or

syslog daemon configuration.

> The first failure is:
> 2020-11-17T18:43:43.160556-06:00 swan charon: 09[CFG] unable to install
> policy ...::a/128 === ::/0 in for reqid 2831, the same
> policy for reqid 2830 exists
>
> I think the interesting bits start at 2020-11-17T18:42:37.

Yes, that's when the duplicate policy was initially installed.

The problem is that the client requests a specific virtual IPv6 address, but not a specific virtual IPv4 address. So when the client connects at 18:42:37 we see this:

```
07[IKE] peer requested virtual IP %any
07[CFG] reassigning offline lease to '...'
07[IKE] assigning virtual IP 10.1.8.13 to peer '...'
07[IKE] peer requested virtual IP ...::a
07[CFG] reassigning offline lease to '...'
07[IKE] assigning virtual IP ...::a to peer '...'
```

So this results in the traffic selectors 0.0.0.0/0 ::/0 === 10.1.8.13/32 ...::a/128. It's important to note that reqids are allocated per CHILD_SA (i.e. IPsec SA pair) and for the complete set of unique local and remote traffic selectors, not for individual policies derived from them. So for these traffic selectors the reqid 2830 is allocated and used when the IPsec SAs and policies are installed.

Now when the client connects again at 18:43:43 we see this:

```
09[IKE] peer requested virtual IP %any
09[CFG] assigning new lease to '...'
09[IKE] assigning virtual IP 10.1.8.52 to peer '...'
09[IKE] peer requested virtual IP ...::a
09[CFG] reassigning online lease to '...'
09[IKE] assigning virtual IP ...::a to peer '...'
```

As you can see, a new IPv4 address is assigned as there is no requested address, while the IPv6 address is reassigned because the client connects from the same address/port so it's assumed to be a reauthentication and assignment of online leases is allowed. The problem is that this results in different traffic selectors: 0.0.0.0/0 ::/0 === 10.1.8.52/32 ...::a/128. And for these a new reqid (2831) is allocated. Now when installing the policies, the IPv4 policies are fine as they are different. However, the IPv6 policies conflict and result in the seen error message.

> Again, this is strongswan 5.6.2-1ubuntu2.5 on Ubuntu 20.04.

What's running on the other end? From the requested config attributes I'd guess some kind of Windows implementation. What kind/version?

The question is, why does the peer only send a specific virtual IPv6 address but not IPv4 address. Is there an address configured that it also requests the very first time (or does it request :: then? If so, I'd avoid such a configuration and maybe assign static addresses based on the client's identity. If not, this seems like a weird implementation quirk/bug.

It's currently not possible to prevent reassigning previous addresses, in particular during reauthentication. As mentioned above, a possible solution is assigning virtual IPs statically based on client identity. That way, both IP addresses will always be the same for the same client identity (no matter what the client requests as the server is allowed to assign a different IP at any time) and the same traffic selectors will result, thus avoiding the conflict. This can be done via database, RADIUS or DHCP (although the latter currently not for IPv6), see VirtualIp for details.

As far as code changes go. A possible workaround for this issue might be to check that either all requested virtual addresses are wildcards, or none are and that all are successfully reallocated, otherwise allocate new leases for *all* of them.

#13 - 23.11.2020 21:26 - Richard Laager

> What's running on the other end? From the requested config attributes I'd guess some kind of Windows implementation. What kind/version?

This is the built-in client in Windows. Based on the ciphers negotiated, I'd say Windows 10. We do still allow Windows 7 (mostly on a "just in case" basis) but nothing older, and my notes show that Windows 7 uses SHA1 for ESP.

> The question is, why does the peer only send a specific virtual IPv6 address but not IPv4 address. Is there an address configured that it also requests the very first time (or does it request :: then?

We do not hard-code any addresses (IPv4 or IPv6) for any VPN clients. The clients add the VPN using the following PowerShell snippet:

```
Add-VpnConnection "Wiktel VPN" -ServerAddress vpn.wiktel.com -TunnelType Ikev2 -EncryptionLevel Maximum -Authe
```

```
nticationMethod Eap -RememberCredential
Set-VpnConnectionIPsecConfiguration "Wiktel VPN" -Force -AuthenticationTransformConstants GCMAES256 -CipherTra
nsformConstants GCMAES256 -DHGroup ECP384 -EncryptionMethod GCMAES256 -IntegrityCheckMethod SHA384 -PfsGroup E
CP384
Add-VpnConnectionRoute "Wiktel VPN" -DestinationPrefix ::/1
Add-VpnConnectionRoute "Wiktel VPN" -DestinationPrefix 8000::/1
```

The two routes are a work-around for the annoying fact that it won't let you add an IPv6 default route. So we split the IPv6 address space in half and use two routes.

I booted a Windows VM of mine to test. This has not connected to the VPN in many days at least. It initially requested any IPv4 and a specific IPv6:

```
2020-11-23T14:10:32.489035-06:00 swan charon: 01[IKE] peer requested virtual IP %any
2020-11-23T14:10:32.489076-06:00 swan charon: 01[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:10:32.489117-06:00 swan charon: 01[IKE] assigning virtual IP 10.1.8.3 to peer 'rlaager'
2020-11-23T14:10:32.489164-06:00 swan charon: 01[IKE] peer requested virtual IP ...::5
2020-11-23T14:10:32.489206-06:00 swan charon: 01[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:10:32.489247-06:00 swan charon: 01[IKE] assigning virtual IP ...::1f to peer 'rlaager'
```

I see it was assigned a different address. As far as I can see (from e.g. "ipsec statusall" and grepping the logs), the requested ...::5 address was not in use.

I then disconnect, deleted the VPN configuration, re-added the VPN configuration using the PowerShell script, and re-connected. For this, the first connection on a new VPN profile, it requested %any and %any6:

```
2020-11-23T14:13:25.909014-06:00 swan charon: 12[IKE] peer requested virtual IP %any
2020-11-23T14:13:25.909056-06:00 swan charon: 12[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:13:25.909098-06:00 swan charon: 12[IKE] assigning virtual IP 10.1.8.32 to peer 'rlaager'
2020-11-23T14:13:25.909146-06:00 swan charon: 12[IKE] peer requested virtual IP %any6
2020-11-23T14:13:25.909188-06:00 swan charon: 12[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:13:25.909230-06:00 swan charon: 12[IKE] assigning virtual IP ...::1f to peer 'rlaager'
```

I then disconnected and re-connected and it's back to requesting %any and a particular IPv6 address:

```
2020-11-23T14:17:29.204596-06:00 swan charon: 14[IKE] peer requested virtual IP %any
2020-11-23T14:17:29.204672-06:00 swan charon: 14[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:17:29.204716-06:00 swan charon: 14[IKE] assigning virtual IP 10.1.8.3 to peer 'rlaager'
2020-11-23T14:17:29.204759-06:00 swan charon: 14[IKE] peer requested virtual IP ...::1f
2020-11-23T14:17:29.204808-06:00 swan charon: 14[CFG] reassigning offline lease to 'rlaager'
2020-11-23T14:17:29.204851-06:00 swan charon: 14[IKE] assigning virtual IP ...::1f to peer 'rlaager'
```

> a possible solution is assigning virtual IPs statically based on client identity.

The client identity is their username, right? So this would prohibit a particular user from connecting twice (as they'd get the same IP address assigned and/or the second connection would not fail because the IP is in use elsewhere). That's not ideal. It might be something we have to accept if there are no other options.

> As far as code changes go. A possible workaround for this issue might be to check that either all requested virtual addresses are wildcards, or none are and that all are successfully reallocated, otherwise allocate new leases for all of them.

Something along those lines might work.

Above, you wrote:

> As you can see, a new IPv4 address is assigned as there is no requested address, while the IPv6 address is reassigned because the client connects from the same address/port so it's assumed to be a reauthentication and assignment of online leases is allowed.

If the client is connecting from the same address/port and it's assumed to be a reauthentication, why is a *new* IPv4 address being assigned? If it the *same* IPv4 was to be assigned, it seems like that would also avoid the issue.

#14 - 24.11.2020 11:23 - Tobias Brunner

> I booted a Windows VM of mine to test. This has not connected to the VPN in many days at least. It initially requested any IPv4 and a specific IPv6:
> [...]
>
> I see it was assigned a different address. As far as I can see (from e.g. "ipsec statusall" and grepping the logs), the requested ...::5 address was not in use.

> I then disconnect, deleted the VPN configuration, re-added the VPN configuration using the PowerShell script, and re-connected. For this, the first connection on a new VPN profile, it requested %any and %any6:
> [...]
>
> I then disconnected and re-connected and it's back to requesting %any and a particular IPv6 address:
>
> [...]

Thanks for these tests. So it looks like Windows remembers the virtual IPv6 address quite permanently for some reason and tries to reuse it later.

> > a possible solution is assigning virtual IPs statically based on client identity.
>
> The client identity is their username, right? So this would prohibit a particular user from connecting twice (as they'd get the same IP address assigned and/or the second connection would not fail because the IP is in use elsewhere). That's not ideal. It might be something we have to accept if there are no other options.

Yes, if you want users to have multiple concurrent connections from different devices that won't work.

> Above, you wrote:
>
> > As you can see, a new IPv4 address is assigned as there is no requested address, while the IPv6 address is reassigned because the client connects from the same address/port so it's assumed to be a reauthentication and assignment of online leases is allowed.
>
> If the client is connecting from the same address/port and it's assumed to be a reauthentication, why is a *new* IPv4 address being assigned? If it the *same* IPv4 was to be assigned, it seems like that would also avoid the issue.

Good question. There is currently a check that only reassigns online leases if the client explicitly requests the same address (source:src/libcharon/attributes/mem_pool.c#L310), which obviously fails if the client does not request a specific address at all. But I guess we could be less strict there and either just reassign any online leases if the peer connects from the same endpoint, or only if it requests %any or the same address. I've pushed a commit that does the latter to the *3541-reassign-online* branch. Probably an easier fix than the check I mentioned above.

#15 - 25.11.2020 08:16 - Richard Laager

Thanks for the quick patch!

I locally rebuilt the Ubuntu package plus your commit from the 3541-reassign-online branch. It works generally, including from Windows. Unfortunately, the issue here is intermittent and I have no way to reproduce it. So it's really hard for me to explicitly confirm that this fixed it.

I'm certainly not an expert in this code, but your explanation seems solid and the code change looks good to my casual read, for whatever that is worth.

#16 - 25.11.2020 12:02 - Tobias Brunner

> Unfortunately, the issue here is intermittent and I have no way to reproduce it. So it's really hard for me to explicitly confirm that this fixed it.

Yeah, no problem. I suppose you could check the log after a while. Because instead of this

```
09[IKE] peer requested virtual IP %any
09[CFG] assigning new lease to '...'
09[IKE] assigning virtual IP 10.1.8.52 to peer '...'
```

you might now see this

```
09[IKE] peer requested virtual IP %any
09[CFG] assigning online lease to '...'
09[IKE] assigning virtual IP 10.1.8.52 to peer '...'
```

if a client reauthenticates (or just reconnects from the same endpoint while the server still has some state).

#17 - 18.01.2021 14:03 - Tobias Brunner

- Subject changed from "unable to install policy" to "unable to install policy" if Windows client reconnects and virtual IPv4 and IPv6 addresses are assigned

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Target version set to 5.9.2

- Resolution set to Fixed

Files

ipsec.conf	2.91 KB	17.08.2020	Richard Laager