strongSwan - Issue #2580

[CFG] handling xx attribute failed in Android or Ubuntu, but works in macOS

06.03.2018 20:27 - Scep CAfail

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:			
Affected version:	5.6.2	**Resolution:**	

Description

Hi,

I applied the following patches for translating RADIUS attributes to IKEv2 equivalent for split-tunnel and split-dns.
https://github.com/kthkaya/strongswan/commit/98105a8a39b53710e0ad072e037e0b69522a8055
https://github.com/kthkaya/strongswan/commit/a2049bfc44bb2a18cafed84fb8da2c2d7b9b87f3

Split-tunnel routes are stored in the form of Address1/Netmask1,Address2/Netmask2 in RADIUS. Then each Address1/Netmask1 is contained within an individual attribute payload as described in the RFC.
Same goes for split-dns. The only difference is the separator (" " instead of ",") at RADIUS.

It works with the native VPN adapter of macOS. The pushed routes correctly appear in the route table, and the pushed domains correctly appear in scutil --dns. Everything OK!
However in Android using strongSwan VPN client, and in Ubuntu using network-manager-strongswan, I get the following errors.

15[CFG] handling (25) attribute failed
15[CFG] handling INTERNAL_IP4_SUBNET attribute failed

Looks like some trivial encoding issue. Can you point me to the right direction?

Regards,
Alan

History

#1 - 07.03.2018 09:04 - Tobias Brunner

- Status changed from New to Feedback

> However in Android using strongSwan VPN client, and in Ubuntu using network-manager-strongswan, I get the following errors.
>
> 15[CFG] handling (25) attribute failed
> 15[CFG] handling INTERNAL_IP4_SUBNET attribute failed
>
> Looks like some trivial encoding issue. Can you point me to the right direction?

Nope, strongSwan currently does not support these attributes.

#2 - 07.03.2018 16:59 - Scep CAfail

> Nope, strongSwan currently does not support these attributes.

Ouch...
I believe the current approach to split-tunneling is via traffic selectors(TS). Client proposes 0.0.0.0/0 and server tailors this proposal in its response with whatever specified in leftsubnets. But I believe we don't have something like leftsubnets=%radius right? (probably it would be ugly to do so with IKEv2, best would be to just implement that IKEv2 attribute handler)

Were I to implement a handler for INTERNAL_IP4_SUBNET where the routes in the attributes are installed into the routing table, would TS negotiation phase overwrite the routes installed by the handler? Or, if I am interpreting source:src/libcharon/plugins/unity/unity_handler.c#L69-L85 correctly, would it be a better approach to adjust the client's TS proposal based on the attribute payload received?

Thanks.

#3 - 07.03.2018 17:13 - Tobias Brunner

> But I believe we don't have something like leftsubnets=%radius right?

The *eap-radius* plugin could theoretically implement the narrow() hook on listener_t to change the traffic selector that's installed and sent back to the client.

INTERNAL_IP4_SUBNET works differently (see #2185-1 regarding some of my concerns). It would certainly work with route-based VPNs, but that's a special case.

> Were I to implement a handler for INTERNAL_IP4_SUBNET where the routes in the attributes are installed into the routing table, would TS negotiation phase overwrite the routes installed by the handler?

Routes are not IPsec policies, but depending on the client that's theoretically an option (e.g. on Android or other route based VPNs).

> Or, if I am interpreting source:src/libcharon/plugins/unity/unity_handler.c#L69-L85 correctly, would it be a better approach to adjust the client's TS proposal based on the attribute payload received?

As mentioned, the *eap-radius* plugin could probably do something similar. It's also possible to do that on the client (i.e. narrow the TS right before installing them, but the server would probably have to do the same).

#4 - 09.03.2018 17:49 - Scep CAfail

Tobias Brunner wrote:

> > But I believe we don't have something like leftsubnets=%radius right?
>
> The *eap-radius* plugin could theoretically implement the narrow() hook on listener_t to change the traffic selector that's installed and sent back to the client.
>
> INTERNAL_IP4_SUBNET works differently (see #2185-1 regarding some of my concerns). It would certainly work with route-based VPNs, but that's a special case.
>
> > Were I to implement a handler for INTERNAL_IP4_SUBNET where the routes in the attributes are installed into the routing table, would TS negotiation phase overwrite the routes installed by the handler?
>
> Routes are not IPsec policies, but depending on the client that's theoretically an option (e.g. on Android or other route based VPNs).
>
> > Or, if I am interpreting source:src/libcharon/plugins/unity/unity_handler.c#L69-L85 correctly, would it be a better approach to adjust the client's TS proposal based on the attribute payload received?
>
> As mentioned, the *eap-radius* plugin could probably do something similar. It's also possible to do that on the client (i.e. narrow the TS right before installing them, but the server would probably have to do the same).

This is very helpful, thank you Tobias. The workarounds you suggest indeed would save the day.
The sentence from your earlier reply *Since the attributes are exchanged only after the client already proposed the traffic selectors* clarifies why it's implementation is a bit in the grey-area. Don't you think INTERNAL_IP4_SUBNET seems to be destined to be applicable for route-based VPNs only? Could Childless SA be an elegant approach?

Alan