strongSwan - Feature #2793

Remote identity with certificate

09.10.2018 14:34 - Wojciech Sikora

Status:	Feedback	**Start date:**	09.10.2018
Priority:	Normal	**Due date:**	
Assignee:		**Estimated time:**	0.00 hour
Category:			
Target version:			
Resolution:			

Description

hello everyone

I have a question, do you have any support plan, match the remote ID which does not depend on the certificate
for example difference remote id them subject name?

I put belowe part of RFC which talk aboute remote ID

part of RFC :
The Identification payloads, denoted IDi and IDr in this document,
allow peers to assert an identity to one another. This identity may
be used for policy lookup, but does not necessarily have to match
anything in the CERT payload; both fields may be used by an
implementation to perform access control decisions.

BR
Wojtek

History

#1 - 11.10.2018 17:13 - Tobias Brunner

- Status changed from New to Feedback

> I have a question, do you have any support plan, match the remote ID which does not depend on the certificate for example difference remote id them subject name?

No, that binding to identities in the certificate is on purpose. You can use *subjectAlternativeName* extensions to have different identities than the full subject DN (e.g. a FQDN or an email address). And it's possible to not send an IDr payload during IKE_AUTH and accept a returned IDr different than the one configured as long as both identities are confirmed by the certificate (either match one of the SANs or the subject DN).

There is an old patch in the *cert-id-binding-option* branch that adds an option to disable that strong binding between IKE identities and certificates, but it's not recommended and unmaintained.

#2 - 12.10.2018 10:44 - Szymon Lenarczyk

Hi Tobias,

Also saw a similar comment from Martin in the mailing list (https://lists.strongswan.org/pipermail/users/2014-April/006021.html)
"Due to the security implications, we have not planned to mainstream these changes."

Would you kindly elaborate on what security implications this has and why it's not recommended?

I interpret the quoted RFC 7296 piece (p. 3.5.) as giving freedom of choice to the implementation (or configuration options)
how the matching is performed and specifically describes one use of matching the identity with the certificate DN/SAN for authorization / access control.
What would you say the security hindrance is for peers that don't make use of authorization?

For context, the motivation behind being able to have a StrongSwan client authenticate the SEGW successfully
expecting a certain pre-configured IDr that's not confirmed by the certificate is a case where
an existing network is already deployed that allows this "mismatch" and re-configuring the SEGW is costly and,
frankly, not deemed necessary, due to (currently) unknown security benefits of this required matching.
Certain SEGWs (like the Juniper SRX series) allow for custom local-identity configuration

(https://www.juniper.net/documentation/en_US/junos/topics/reference/configuration-statement/security-edit-local-identity.html)

Thanks
Szymon

#3 - 12.10.2018 13:22 - Tobias Brunner

> I interpret the quoted RFC 7296 piece (p. 3.5.) as giving freedom of choice to the implementation (or configuration options)
> how the matching is performed and specifically describes one use of matching the identity with the certificate DN/SAN for authorization / access control.
> What would you say the security hindrance is for peers that don't make use of authorization?

If you don't use the identities for authorization (or derive other things from the identity, e.g. particular virtual IPs etc., which then might be misused) I don't really see any security implications. But is that common?

> For context, the motivation behind being able to have a StrongSwan client authenticate the SEGW successfully
> expecting a certain pre-configured IDr that's not confirmed by the certificate is a case where
> an existing network is already deployed that allows this "mismatch" and re-configuring the SEGW is costly and,
> frankly, not deemed necessary, due to (currently) unknown security benefits of this required matching.

It basically allows anybody with an accepted certificate to authenticate with an arbitrary accepted identity. If that's not a problem for you, the mentioned branch might provide a solution.

> Certain SEGWs (like the Juniper SRX series) allow for custom local-identity configuration
> (https://www.juniper.net/documentation/en_US/junos/topics/reference/configuration-statement/security-edit-local-identity.html)

That doesn't mean it's a good idea :)

#4 - 12.10.2018 15:40 - Szymon Lenarczyk

Tobias Brunner wrote:

>> I interpret the quoted RFC 7296 piece (p. 3.5.) as giving freedom of choice to the implementation (or configuration options)
>> how the matching is performed and specifically describes one use of matching the identity with the certificate DN/SAN for authorization / access control.
>> What would you say the security hindrance is for peers that don't make use of authorization?
>
> If you don't use the identities for authorization (or derive other things from the identity, e.g. particular virtual IPs etc., which then might be misused) I don't really see any security implications. But is that common?

Happens in certain circumstances for simple implementation, when access control is "binary" and is based solely on selectors.

>> For context, the motivation behind being able to have a StrongSwan client authenticate the SEGW successfully
>> expecting a certain pre-configured IDr that's not confirmed by the certificate is a case where
>> an existing network is already deployed that allows this "mismatch" and re-configuring the SEGW is costly and,
>> frankly, not deemed necessary, due to (currently) unknown security benefits of this required matching.
>
> It basically allows anybody with an accepted certificate to authenticate with an arbitrary accepted identity. If that's not a problem for you, the mentioned branch might provide a solution.

Alternatively, instead of accepting arbitrary identities, one may want to match the identity to what is configured with

```
remote_auth_cfg->add(cfg, AUTH_RULE_IDENTITY, remote_identity);
```

which still requires the peer to identify themselves in a specific way.

>> Certain SEGWs (like the Juniper SRX series) allow for custom local-identity configuration
>> (https://www.juniper.net/documentation/en_US/junos/topics/reference/configuration-statement/security-edit-local-identity.html)
>
> That doesn't mean it's a good idea :)

True, which is why this could only be offered as a temporal workaround for networks that aren't designed with good ideas in mind :-)

#5 - 15.10.2018 10:20 - Tobias Brunner

> > > I interpret the quoted RFC 7296 piece (p. 3.5.) as giving freedom of choice to the implementation (or configuration options)
> > > how the matching is performed and specifically describes one use of matching the identity with the certificate DN/SAN for authorization / access control.
> > > What would you say the security hindrance is for peers that don't make use of authorization?
> >
> > If you don't use the identities for authorization (or derive other things from the identity, e.g. particular virtual IPs etc., which then might be misused) I don't really see any security implications. But is that common?
>
> Happens in certain circumstances for simple implementation, when access control is "binary" and is based solely on selectors.

What selectors?

> > > For context, the motivation behind being able to have a StrongSwan client authenticate the SEGW successfully
> > > expecting a certain pre-configured IDr that's not confirmed by the certificate is a case where
> > > an existing network is already deployed that allows this "mismatch" and re-configuring the SEGW is costly and,
> > > frankly, not deemed necessary, due to (currently) unknown security benefits of this required matching.
> >
> > It basically allows anybody with an accepted certificate to authenticate with an arbitrary accepted identity. If that's not a problem for you, the mentioned branch might provide a solution.
>
> Alternatively, instead of accepting arbitrary identities, one may want to match the identity to what is configured with
>
> > remote_auth_cfg->add(cfg, AUTH_RULE_IDENTITY, remote_identity);
>
> which still requires the peer to identify themselves in a specific way.

Sure, that's why I wrote "accepted identity", but anybody with an accepted certificate can send the configured identity, so it really doesn't help much other than to e.g. select different configs (which any accepted client can then do by changing the sent identity).