strongSwan - Issue #3307

Probable non compliance with RFC 7296 wrt traffic selector narrowing?

10.01.2020 23:55 - Rohan Shethia

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:	configuration		
Affected version:	5.6.3	**Resolution:**	

Description

Scenario 1

A client within Host A (192.168.111.1) initiates the IPsec connection by pinging a client in Host B (192.168.131.1)
This is results in the following traffic selectors proposed by the initiator (Host A):
TSi - 192.168.111.1 and range 192.168.111.0/24
TSr - 192.168.131.1 and range 192.168.131.0/24
TSi - Range 192.168.112.0/24 (since Host A is configured with two local subnets it proposes it)
TSr - Range 192.168.131.0/24

The responder (Host B) responds with:
TSi - Range 192.168.131.0/24
TSr - Range 192.168.111.0/24

And in this case strongSwan on Host A installs the CHILD_SA for the narrowed set, local - 192.168.111.0/24 remote - 192.168.131.0/24 which is expected according to the RFC.

Scenario 2

A client within Host B (192.168.131.1) initiates the IPsec connection by pinging a client in Host A (192.168.111.1)
This is results in the following traffic selectors proposed by the initiator (Host B):
TSi - 192.168.131.1 and range 192.168.131.0/24
TSr - 192.168.111.1 and range 192.168.111.0/24
(Since that's all that is configured)

In this case the responder (Host A) responds with a **TS_UNACCEPTABLE**

This confuses me because this doesn't seem to follow any of the traffic selector negotiation guidelines from the RFC (section 2.9)

Posting the specific guidelines wrt to narrowing from the RFC:

The responder performs the narrowing as follows:

- If the responder's policy does not allow it to accept any part of the proposed Traffic Selectors, it responds with a TS_UNACCEPTABLE Notify message.
- If the responder's policy allows the entire set of traffic covered by TSi and TSr, no narrowing is necessary, and the responder can return the same TSi and TSr values.
- If the responder's policy allows it to accept the first selector of TSi and TSr, then the responder MUST narrow the Traffic Selectors to a subset that includes the initiator's first choices. In this example above, the responder might respond with TSi being with all ports and IP protocols.
- If the responder's policy does not allow it to accept the first selector of TSi and TSr, the responder narrows to an acceptable subset of TSi and TSr.

Posting the swanctl config files for both boxes running strongSwan.

History

#1 - 13.01.2020 15:01 - Tobias Brunner

- Status changed from New to Feedback

Read the log of the responder (set *cfg* to 2 to see more details regarding the TS negotiation).

#2 - 13.01.2020 18:48 - Rohan Shethia

Tobias Brunner wrote:

> Read the log of the responder (set *cfg* to 2 to see more details regarding the TS negotiation).

The logs for the responder are:

```
Jan 13 17:40:10 10[CFG] <8> looking for an ike config for 172.29.0.252...172.29.0.215
Jan 13 17:40:10 10[CFG] <8> ike config match: 3100 (172.29.0.252 172.29.0.215 IKEv2)
Jan 13 17:40:10 10[CFG] <8>   candidate: 172.29.0.252...172.29.0.215, prio 3100
Jan 13 17:40:10 10[CFG] <8> found matching ike config: 172.29.0.252...172.29.0.215 with prio 3100
Jan 13 17:40:10 10[CFG] <8> selecting proposal:
Jan 13 17:40:10 10[CFG] <8>   proposal matches
Jan 13 17:40:10 10[CFG] <8> received proposals: IKE:3DES_CBC/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
Jan 13 17:40:10 10[CFG] <8> configured proposals: IKE:3DES_CBC/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
Jan 13 17:40:10 10[CFG] <8> selected proposal: IKE:3DES_CBC/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
Jan 13 17:40:10 10[CFG] <8> received supported signature hash algorithms: sha256 sha384 sha512
Jan 13 17:40:10 10[CFG] <8> sending supported signature hash algorithms: sha256 sha384 sha512
Jan 13 17:40:10 13[CFG] <8> looking for peer configs matching 172.29.0.252[172.29.0.252]...172.29.0.215[172.29
.0.215]
Jan 13 17:40:10 13[CFG] <8> peer config match local: 20 (ID_IPV4_ADDR -> ac:1d:00:fc)
Jan 13 17:40:10 13[CFG] <8> peer config match remote: 20 (ID_IPV4_ADDR -> ac:1d:00:d7)
Jan 13 17:40:10 13[CFG] <8> ike config match: 3100 (172.29.0.252 172.29.0.215 IKEv2)
Jan 13 17:40:10 13[CFG] <8>   candidate "remote-peer-2", match: 20/20/3100 (me/other/ike)
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> selected peer config 'remote-peer-2'
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> looking for a child config for 192.168.111.1/32[icmp/8] 192.168.111.
0/24 === 192.168.131.1/32[icmp/8] 192.168.131.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> proposing traffic selectors for us:
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  192.168.111.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  192.168.112.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> proposing traffic selectors for other:
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  192.168.131.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>   candidate "net-2" with prio 7+7
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> found matching child config "net-2" with prio 14
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> selecting proposal:
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>   proposal matches
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> received proposals: ESP:3DES_CBC/HMAC_SHA1_96/NO_EXT_SEQ
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> configured proposals: ESP:3DES_CBC/HMAC_SHA1_96/NO_EXT_SEQ
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> selected proposal: ESP:3DES_CBC/HMAC_SHA1_96/NO_EXT_SEQ
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> selecting traffic selectors for us:
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.111.0/24, received: 192.168.111.1/32[icmp/8] => mat
ch: 192.168.111.1/32[icmp/8]
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.111.0/24, received: 192.168.111.0/24 => match: 192.
168.111.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.112.0/24, received: 192.168.111.1/32[icmp/8] => no
match
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.112.0/24, received: 192.168.111.0/24 => no match
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> selecting traffic selectors for other:
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.131.0/24, received: 192.168.131.1/32[icmp/8] => mat
ch: 192.168.131.1/32[icmp/8]
Jan 13 17:40:10 13[CFG] <remote-peer-2|8>  config: 192.168.131.0/24, received: 192.168.131.0/24 => match: 192.
168.131.0/24
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> unable to install policy 192.168.131.0/24 === 192.168.111.0/24 in fo
r reqid 12, the same policy for reqid 9 exists
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> unable to install policy 192.168.131.0/24 === 192.168.111.0/24 fwd f
or reqid 12, the same policy for reqid 9 exists
Jan 13 17:40:10 13[CFG] <remote-peer-2|8> unable to install policy 192.168.111.0/24 === 192.168.131.0/24 out f
or reqid 12, the same policy for reqid 9 exists
```

Why does it try to install a policy again when it has already installed the policies?

The policies on the responder before any attempt at a connection are as follows:

```
src 192.168.131.0/24 dst 192.168.111.0/24
    dir fwd priority 375422
    tmpl src 172.29.0.215 dst 172.29.0.252
        proto esp reqid 9 mode tunnel
src 192.168.131.0/24 dst 192.168.111.0/24
    dir in priority 375424
```

```
    tmpl src 172.29.0.215 dst 172.29.0.252
        proto esp reqid 9 mode tunnel
src 192.168.111.0/24 dst 192.168.131.0/24
    dir out priority 375422
    tmpl src 172.29.0.252 dst 172.29.0.215
        proto esp reqid 9 mode tunnel
src 192.168.131.0/24 dst 192.168.112.0/24
    dir fwd priority 375422
    tmpl src 172.29.0.215 dst 172.29.0.252
        proto esp reqid 9 mode tunnel
src 192.168.131.0/24 dst 192.168.112.0/24
    dir in priority 375424
    tmpl src 172.29.0.215 dst 172.29.0.252
        proto esp reqid 9 mode tunnel
src 192.168.112.0/24 dst 192.168.131.0/24
    dir out priority 375422
    tmpl src 172.29.0.252 dst 172.29.0.215
        proto esp reqid 9 mode tunnel
src 0.0.0.0/0 dst 0.0.0.0/0
    socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
    socket out priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
    socket in priority 0
src 0.0.0.0/0 dst 0.0.0.0/0
    socket out priority 0
src ::/0 dst ::/0
    socket in priority 0
src ::/0 dst ::/0
    socket out priority 0
src ::/0 dst ::/0
    socket in priority 0
src ::/0 dst ::/0
    socket out priority 0
```

Update: When I choose to not install the policies for that child using *policies=no*, the CHILD_SA is established (since it does not hit the problem of having to install a duplicate policy) but obviously traffic doesn't go through since there are no policies now. I seem to be missing something.

#3 - 13.01.2020 19:01 - Tobias Brunner

Trap policies -> reqid gets allocated for **configured** TS -> policies derived from TS are installed accordingly.

Narrowing as responder -> **subset** of configured TS is negotiated -> new reqid is allocated (there is currently no subset matching, same exact TS = same reqid) -> policies are installed -> conflict with previous policies.

As initiator, the reqid of trap policies is maintained if the responder narrows the TS (to avoid this exact issue). However, as responder there is currently no solution, only workarounds: don't use traps, use matching configs so no narrowing occurs, or use static reqids.

#4 - 13.01.2020 19:34 - Rohan Shethia

Oh okay, that makes sense to me. Thank you so much.

Edit: I'm trying to understand how it is different to do subset matching for the responder? Because it works correctly for the initiator (as observed and as you mentioned). If it's just not been implemented I could maybe try working on it and submit the change.

#5 - 14.01.2020 16:19 - Tobias Brunner

> If it's just not been implemented I could maybe try working on it and submit the change.

Unfortunately, it's not so simple. What's interesting is that there actually was code up until about two years ago (removed with 5.6.3) that tried to reuse reqids of trapped configs even as responder. However, it was problematic if multiple CHILD_SA configs shared the same name (which is possible e.g. with swanctl.conf and other more dynamic backends) and also if narrowing resulted in multiple policies (in which case the SAs have no selectors on them) as no new acquires would be generated for the larger/other policies with the same reqid (their traffic would even get tunneled inadvertently due to the matching reqid). The latter actually also applies to initiators where the reqid is kept, so narrowing that results in duplicate policies is not ideal in any case. This is not an issue if the narrowing results in subsets of **all** configured TS (i.e. if it results in "smaller" and different policies getting installed e.g. if the initiator proposes only a single IP address). It's only an issue in situations like these where narrowing occurs but some of the same policies still are derived from it.

Files

HostA.rtf	1.22 KB	10.01.2020	Rohan Shethia
Host B.rtf	1.2 KB	10.01.2020	Rohan Shethia