strongSwan - Issue #3469

Duplicate Child SA after rekey with AWS Transit Gateway(TG)

02.06.2020 12:39 - Krishnamurthy Daulatabad

Status:	Closed		
Priority:	Normal		
Assignee:	Tobias Brunner		
Category:	interoperability		
Affected version:	5.7.1	**Resolution:**	No change required

Description

We are configuring PSK mode tunnels with Amazon TG and tunnels come up fine. After every rekey, there is a duplicate CHILD_SA created and old CHILD_SA not deleted. Here is the sequence that we have understood from the logs:

1. After ~55 mins of establishment, First rekey is initiated by TG - creates a new CHILD_SA
2. Does not delete old CHILD_SA or IKE_SA
3. Strongswan client initiates new IKE_SA at ~60mins after establishment. Creates a new CHILD_SA and deletes the old IKE_SA
4. Now there is one IKE_SA and 2 CHILD_SA after first rekey.

This sequence keeps repeating and duplicate or stale CHILD_SAs don't go way. Couple of issues we see here:

1. We are suspecting TG is not deleting the IKE_SA and old CHILD_SA
2. strongswan is initiating another rekey again. Is it due to TG not deleting old sas?

Do you suspect any configuration issue here? If we configure rekey time of 50min on strongswan side, there is no issue at all and rekey works fine (without duplicate CHILD_SAs)
I have attached the initial logs (ike_init.log) and rekey logs (ike_rekey.log) here. And ike_summary.log that shows issue we are observing. Please let me know if you need any other information.

Brief logs of swanctl --list-sas at start and after first rekey:

```
18.188.222.191: #1, ESTABLISHED, IKEv2, c977eea3eb906dc3_i* 42fad70f4728be57_r
  local  '192.168.201.122' @ 192.168.201.122[4500]
  remote '18.188.222.191' @ 18.188.222.191[4500]
  AES_CBC-256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
  established 584s ago, rekeying in 2854s
  18.188.222.191: #1, reqid 65538, INSTALLED, TUNNEL-in-UDP, ESP:AES_CBC-256/HMAC_SHA1_96
    installed 585s ago, rekeying in 2677s, expires in 3376s
    in  54084b13,      0 bytes,     0 packets
    out e58fdfec,      0 bytes,     0 packets
    local  192.168.201.122/32
    remote 0.0.0.0/0

2020-06-02T04:34:25.0+0000 13[IKE] <18.188.222.191|1> CHILD_SA 18.188.222.191{1} established with
SPIs 54084b13_i e58fdfec_o and TS 192.168.201.122/32 === 0.0.0.0/0
```

After first rekey

```
18.188.222.191: #8, ESTABLISHED, IKEv2, 52851766943439d5_i* ac1ec71c0db0e869_r
  local  '192.168.201.122' @ 192.168.201.122[4500]
  remote '18.188.222.191' @ 18.188.222.191[4500]
  AES_CBC-256/HMAC_SHA1_96/PRF_HMAC_SHA1/MODP_1024
  established 497s ago, rekeying in 2811s
  18.188.222.191: #8, reqid 65538, INSTALLED, TUNNEL-in-UDP, ESP:AES_CBC-256/HMAC_SHA1_96/MODP_102
4
    installed 605s ago, rekeying in 2786s, expires in 3355s
    in  4e6dce1c,      0 bytes,     0 packets
    out 693b53bd,      0 bytes,     0 packets
    local  192.168.201.122/32
    remote 0.0.0.0/0
  18.188.222.191: #10, reqid 65538, INSTALLED, TUNNEL-in-UDP, ESP:AES_CBC-256/HMAC_SHA1_96/MODP_10
24
```

```
    installed 335s ago, rekeying in 3056s, expires in 3625s
    in  113eb365,      0 bytes,     0 packets
    out 6830a1f7,      0 bytes,     0 packets
    local  192.168.201.122/32
    remote 0.0.0.0/0
```

Here is a gist of the configuration we are using in swanctl.conf

rekey_time=60m
auth=psk
ike proposals (configured via vici): aes256-sha-prfsha1-modp1024
esp_proposals: aes256-sha-sha256-sha384-modp1024

History

#1 - 02.06.2020 13:54 - Tobias Brunner

- Description updated

- Status changed from New to Feedback

1. After ~55 mins of establishment, First rekey is initiated by TG - creates a new CHILD_SA
2. Does not delete old CHILD_SA or IKE_SA

That sounds wrong. The host initiating the rekeying should delete the old CHILD_SA afterwards. However, according to the log, the peer is not actually initiating a rekeying (CREATE_CHILD_SA exchanges to rekey a CHILD_SA contain a REKEY_SA notify with the SPI of the old SA). Instead, it seems to just create a duplicate CHILD_SA for some reason. Maybe check the log of that peer if available.

2. strongswan is initiating another rekey again. Is it due to TG not deleting old sas?

Lifetimes are managed by each peer individually. And the IKE and CHILD_SA lifetimes are independent too.

If we configure rekey time of 50min on strongswan side, there is no issue at all and rekey works fine (without duplicate CHILD_SAs)

Sounds like a bug in the other implementation. So if that's a workaround, maybe go with that.

#2 - 02.06.2020 15:13 - Krishnamurthy Daulatabad

Not getting any logs on TG side. Will try anyway.

Regarding the rekey, if the rekey is done by one of the peer's, will the other cancel the rekey for that iteration (for IKE SA or CHILD SA)? This is what happens in case of rekey collision right?

#3 - 02.06.2020 15:29 - Tobias Brunner

Regarding the rekey, if the rekey is done by one of the peer's, will the other cancel the rekey for that iteration (for IKE SA or CHILD SA)? This is what happens in case of rekey collision right?

Yeah, kinda. But if there is no rekeying (as in this case), just a new CHILD_SA, which is perfectly legal, there is no collision/nothing to cancel.

#4 - 30.09.2020 13:53 - Tobias Brunner

- Category set to interoperability

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to No change required

Files

ike_init.log	4.2 MB	02.06.2020	Krishnamurthy Daulatabad
ike_rekey.log	1.69 MB	02.06.2020	Krishnamurthy Daulatabad
ike_summary.log	14.7 KB	02.06.2020	Krishnamurthy Daulatabad