strongSwan - Bug #2315

FreeBSD server stops routing new connections after 16k connects/disconnects

03.05.2017 23:55 - Mike E

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	freebsd		
Target version:	5.8.3		
Affected version:	5.5.1	**Resolution:**	Fixed

Description

After cycling through 16k 'ipsec up' and 'ipsec down' commands from the client, the server will stop establishing complete connections. Logs included are from 5.5.1, but is reproducible in 5.5.2.

This message appears in the log after 16k connections are made:

```
Thu, 2017-05-04 05:30 13[CFG] trap not found, unable to acquire reqid 16384

Status of IKE charon daemon (strongSwan 5.5.1, FreeBSD 11.0-RELEASE-p9, amd64):
  uptime: 2 hours, since May 04 03:22:24 2017
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 2562
  loaded plugins: charon aes des blowfish rc2 sha2 sha1 md4 md5 random nonce x509 revocation const
raints pubkey pkcs1 pkcs7 pkcs8 pkcs12 pgp dnskey sshkey pem openssl fips-prf xcbc cmac hmac attr
kernel-pfkey kernel-pfroute resolve socket-default stroke vici updown eap-identity eap-md5 eap-msc
hapv2 eap-tls eap-ttls eap-peap xauth-generic whitelist addrblock
Virtual IP pools (size/online/offline):
  192.168.0.0/24: 254/0/1
Listening IP addresses:
  10.111.52.16
  192.168.1.1
Connections:
          rw:  10.111.52.16...%any  IKEv2
          rw:   local:  [10.111.52.16] uses pre-shared key authentication
          rw:   remote: uses pre-shared key authentication
          rw:   child:  0.0.0.0/0 === dynamic TUNNEL
Security Associations (0 up, 0 connecting):
  none
```

Related issues:

Has duplicate Issue #3293: The VPN server stops routing traffic to new tunnel...	**Closed**

Associated revisions

Revision 4958acc0 - 09.03.2020 15:27 - Tobias Brunner

kernel-interface: Reallocate previously used reqids

This is mainly an issue on FreeBSD where the current kernel still only
allows the daemon to use reqids < IPSEC_MANUAL_REQID_MAX (0x3fff = 16383).

Fixes #2315.

History

#1 - 04.05.2017 02:08 - Noel Kuntze

- Status changed from New to Feedback

I took a look at the log and the reqid 16384 isn't logged anywhere else. Can we have the status of the SPD of the kernel when that error appears? Taking a look at the kernel logs is a good idea, I think. The kernel might log something. I wonder why an acquire is detected, though, because you're using auto=add in the rw conn.

#2 - 04.05.2017 09:19 - Mike E

This is what it looks like when a failed connection is up:

```
# setkey -DP
192.168.0.1[any] 0.0.0.0/0[any] any
    in ipsec
    esp/tunnel/10.111.52.17-10.111.52.16/unique#21942
    created: May  4 15:38:55 2017  lastused: May  4 15:39:02 2017
    lifetime: 9223372036854775807(s) validtime: 0(s)
    spid=521092 seq=1 pid=6903
    refcnt=1
0.0.0.0/0[any] 192.168.0.1[any] any
    out ipsec
    esp/tunnel/10.111.52.16-10.111.52.17/unique#21941
    created: May  4 15:38:55 2017  lastused: May  4 15:39:02 2017
    lifetime: 9223372036854775807(s) validtime: 0(s)
    spid=521091 seq=0 pid=6903
    refcnt=1
root@strongswan1:~/log # setkey -D
10.111.52.16 10.111.52.17
    esp mode=tunnel spi=3375329924(0xc92f7284) reqid=16385(0x00004001)
    E: rijndael-cbc  a3bbcd56 73379332 5f720bff 4b4b5313
    A: hmac-sha2-256  ec0f85aa 78b5ea0b 699072f7 cadbbf04 7981a52c 010cae82 7b6de1c0 6b7167ae
    seq=0x00000000 replay=0 flags=0x00000000 state=mature
    created: May  4 15:38:55 2017    current: May  4 15:39:25 2017
    diff: 30(s)    hard: 1200(s)    soft: 883(s)
    last:                          hard: 0(s)    soft: 0(s)
    current: 0(bytes)    hard: 0(bytes)    soft: 0(bytes)
    allocated: 0    hard: 0    soft: 0
    sadb_seq=1 pid=6904 refcnt=1
10.111.52.17 10.111.52.16
    esp mode=tunnel spi=3329903520(0xc67a4ba0) reqid=16385(0x00004001)
    E: rijndael-cbc  ac7d5ccb 2024e5e4 e8dd1668 74a98426
    A: hmac-sha2-256  fda19954 fa15fdb6 0956d491 4994be37 c3be7164 26b872eb 94f869e8 ccda78c5
    seq=0x00000003 replay=4 flags=0x00000000 state=mature
    created: May  4 15:38:55 2017    current: May  4 15:39:25 2017
    diff: 30(s)    hard: 1200(s)    soft: 920(s)
    last: May  4 15:39:02 2017    hard: 0(s)    soft: 0(s)
    current: 252(bytes)    hard: 0(bytes)    soft: 0(bytes)
    allocated: 3    hard: 0    soft: 0
    sadb_seq=0 pid=6904 refcnt=1
```

The kernel didn't log anything, so I've changed syslogd to log kern.* and rerunning test now.

#3 - 04.05.2017 10:41 - Mike E

Nothing at all was logged by the kernel.

#4 - 04.05.2017 11:00 - Tobias Brunner

That because of IPSEC_MANUAL_REQID_MAX (0x3fff == 16383). Which is a strangely low limit (at least for keying daemons like strongSwan that manage reqids themselves) since reqids are 32-bit numbers.

reqids are currently allocated sequentially using a sttic counter (source:src/libcharon/kernel/kernel_interface.c#L328). The code that allocates them does not know anything about the limit above (it doesn't even know or care that it runs on a FreeBSD kernel).

> Nothing at all was logged by the kernel.

Logging in the IPsec stack of the FreeBSD kernel has to be enabled explicitly (IPSEC_DEBUG kernel option and sysctl net.inet.ipsec.debug=1).

#5 - 09.05.2017 12:54 - Mike E

I'm looking at implementing some code to re-use freed reqids. Are there any known pitfalls that need to be considered for reqid re-use?

#6 - 13.12.2019 16:22 - Tobias Brunner

- Has duplicate Issue #3293: The VPN server stops routing traffic to new tunnels when the variable "reqid" reaches "16383" added

#7 - 13.12.2019 17:15 - Tobias Brunner

I pushed a workaround for this issue to the *2315-realloc-reqids* branch.

#8 - 16.12.2019 14:18 - Geovane Gonçalves

Hi,

These are the latest considerations from the Freebsd dev team about the issue:

"IPSEC_MANUAL_REQID_MAX is not FreeBSD-specific; it is also 0x3fff on Linux.

Anyway, the comment in the header is clear enough: REQIDs over 0x3fff are reserved for the kernel. Linux uses this range for the kernel as well (see net/key/af_key.c#L1915, gen_reqid()). They simply ignore bogus user requests for higher numbers:

https://github.com/torvalds/linux/blob/master/net/key/af_key.c#L1959

```
if (t->reqid > IPSEC_MANUAL_REQID_MAX)
            t->reqid = 0;
```

In fact, FreeBSD does something similar, but produces a warning first (ipseclog LOG_DEBUG, "reqid=%d range violation, updated by kernel"). That code is present since 2002. I can't tell if libcharon is broken on Linux and merely doesn't observe it there, or if it's just poorly designed. I don't know if pfsense has any modifications to FreeBSD in this area that might be relevant. Can you reproduce the problem on FreeBSD, or just pfsense?"

By Conrad Meyer

I put this information here because I thought it might be relevant.

https://bugs.freebsd.org/bugzilla/show_bug.cgi?id=242606

Thanks,

Geovane

#9 - 16.12.2019 15:06 - Tobias Brunner

> "IPSEC_MANUAL_REQID_MAX is not FreeBSD-specific; it is also 0x3fff on Linux.

That's not relevant because no keying daemon (that is still in development) is using PF_KEY on Linux and the XFRM interface (or the XFRM/IPsec core in general) doesn't impose any limits on the use of reqids by the userland.

> REQIDs over 0x3fff are reserved for the kernel.

I'd say, if anything, it's the other way around. The lower ones are "reserved" for manual use (i.e. the kernel does not allocate reqids there if none is specified in the request, see below).

> Linux uses this range for the kernel as well (see net/key/af_key.c#L1915, gen_reqid()).

No, it doesn't, that code is specific to the PF_KEY interface. I guess keying daemons using that interface were expected to not allocate reqids manually, but let the kernel do that for them (like SPIs, however, there is no SADB_GETREQID like there is SADB_GETSPI, and allocating a separate reqid for each policy and direction breaks several scenarios and seems unnecessarily wasteful). The range below 0x4000 was excluded for manual use (like the range for SPIs < 256, which, however, is backed by RFC 4303), but I guess that really meant manual (i.e. via setkey) and not via a keying daemon.

I don't see why the kernel couldn't accept requests with reqids > 0x3fff while still allocating them for requests that don't specify one. For instance, on Linux the code in gen_reqid() simply searches for an unused reqid starting with 0x4000, which works whether such reqids were allocated by the kernel or the keying daemon (but again, that interface is not relevant on Linux except for compatibility with legacy applications like racoon or setkey).

#10 - 21.01.2020 10:58 - Andrey Elsukov

Tobias Brunner wrote:

> I don't see why the kernel couldn't accept requests with reqids > 0x3fff while still allocating them for requests that don't specify one. For instance, on Linux the code in gen_reqid() simply searches for an unused reqid starting with 0x4000, which works whether such reqids were allocated by the kernel or the keying daemon (but again, that interface is not relevant on Linux except for compatibility with legacy applications like racoon or setkey).

So, Tobias, what do you suggest? Will you fix this problem in strongswan or you think it should be fixed in kernel?

#11 - 21.01.2020 14:56 - Tobias Brunner

> So, Tobias, what do you suggest? Will you fix this problem in strongswan or you think it should be fixed in kernel?

As I wrote, I think it doesn't make sense for the kernel to impose restrictions on the reqids used by the userland (especially nowadays). So I'm all in

favor of a kernel change.

But since that would probably take a while and may be not what the kernel devs want to do, we can also patch strongSwan to reuse the reqids. The patch in the *2315-realloc-reqids* branch shouldn't really have much of a negative impact, so we can probably just go with that.

#12 - 21.01.2020 15:58 - Geovane Gonçalves

Thanks Tobias. Do you have any estimate of when the patch will be incorporated into the production version?

#13 - 06.03.2020 15:30 - Lars Pedersen

Geovane Gonçalves wrote:

> Thanks Tobias. Do you have any estimate of when the patch will be incorporated into the production version?

Tobias Brunner wrote:

> > So, Tobias, what do you suggest? Will you fix this problem in strongswan or you think it should be fixed in kernel?
>
> As I wrote, I think it doesn't make sense for the kernel to impose restrictions on the reqids used by the userland (especially nowadays). So I'm all in favor of a kernel change.
>
> But since that would probably take a while and may be not what the kernel devs want to do, we can also patch strongSwan to reuse the reqids. The patch in the *2315-realloc-reqids* branch shouldn't really have much of a negative impact, so we can probably just go with that.

Can you give an estimate if/when this fix hits master?

#14 - 09.03.2020 15:32 - Tobias Brunner

- Tracker changed from Issue to Bug

- Subject changed from freebsd server stops routing new connections after 16k connects/disconnects to FreeBSD server stops routing new connections after 16k connects/disconnects

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Target version set to 5.8.3

- Resolution set to Fixed

The workaround (reusing released reqids) is now in master.

Files

netstat.txt	1.04 KB	03.05.2017	Mike E
ifconfig.txt	796 Bytes	03.05.2017	Mike E
ipsec_server.conf	402 Bytes	03.05.2017	Mike E
charon_debug.log.bz2	4.47 MB	03.05.2017	Mike E