strongSwan - Bug #1051

Passive node in HA configuration receive "UNDEFINED" as integrity algorithm if AEAD algorithm is used

03.08.2015 09:06 - Alexander Sukhomlinov

Status:	Closed	**Start date:**	03.08.2015
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	libcharon		
Target version:	5.3.3		
Affected version:	5.2.2	**Resolution:**	Fixed

Description

Hello!

We try to set up HA (active\active) plugin on our test bench using aes256gcm16 as encryption algorithm for esp and ike. We have 3 linux machines (using as vpn gateways, 2 in HA and 1 - as initiator to HA gateway) which are used for testing.

When connection is initiated to the HA gateway, one of nodes in HA, which becomes a mode - PASSIVE, cannot establish passive CHILD_SA. We found that why passive node can't establish CHILD_SA and it is revealed in logs on passive node:

```
Aug  3 11:20:53 7208 charon: 04[IKE] IKE_SA cvpncln1 state change: CONNECTING => PASSIVE
Aug  3 11:20:53 7208 charon: 04[MGR] checkin IKE_SA cvpncln1
Aug  3 11:20:53 7208 charon: 04[MGR] check-in of IKE_SA successful.
Aug  3 11:20:53 7208 charon: 04[CFG] received HA CHILD_ADD message
Aug  3 11:20:53 7208 charon: 04[MGR] checkout IKE_SA
Aug  3 11:20:53 7208 charon: 04[MGR] IKE_SA cvpncln1 successfully checked out
Aug  3 11:20:53 7208 charon: 04[CHD]   using AES_GCM_16 for encryption
Aug  3 11:20:53 7208 charon: 04[CHD]   using UNDEFINED for integrity
Aug  3 11:20:53 7208 charon: 04[CHD] no keylength defined for UNDEFINED
Aug  3 11:20:53 7208 charon: 04[CHD] HA CHILD_SA key derivation failed
```

So, passive node somehow receive "UNDEFINED" as integrity algorithm, but active node just ignores this and doesn't use integrity algorithm at all:

```
Aug  3 11:20:52 7206 charon: 16[IKE] IKE_SA cvpncln1 state change: CONNECTING => ESTABLISHED
Aug  3 11:20:52 7206 charon: 16[CFG] selecting proposal:
Aug  3 11:20:52 7206 charon: 16[CFG]   proposal matches
Aug  3 11:20:52 7206 charon: 16[CFG] received proposals: ESP:AES_GCM_16_256/NO_EXT_SEQ
Aug  3 11:20:52 7206 charon: 16[CFG] configured proposals: ESP:AES_GCM_16_256/MODP_4096/NO_EXT_SEQ
Aug  3 11:20:52 7206 charon: 16[CFG] selected proposal: ESP:AES_GCM_16_256/NO_EXT_SEQ
Aug  3 11:20:52 7206 charon: 16[CFG] selecting traffic selectors for us:
Aug  3 11:20:52 7206 charon: 16[CFG]  config: 192.168.25.0/24, received: 192.168.25.0/24 => match:
 192.168.25.0/24
Aug  3 11:20:52 7206 charon: 16[CFG] selecting traffic selectors for other:
Aug  3 11:20:52 7206 charon: 16[CFG]  config: 192.168.26.0/24, received: 192.168.26.0/24 => match:
 192.168.26.0/24
Aug  3 11:20:52 7206 charon: 16[CHD]   using AES_GCM_16 for encryption
Aug  3 11:20:52 7206 charon: 16[CHD] adding inbound ESP SA
Aug  3 11:20:52 7206 charon: 16[CHD]   SPI 0xc6acd01d, src 192.168.100.207 dst 192.168.100.197
Aug  3 11:20:52 7206 charon: 16[KNL] adding SAD entry with SPI c6acd01d and reqid {1}  (mark 0/0x0
0000000)
Aug  3 11:20:52 7206 charon: 16[KNL]   using encryption algorithm AES_GCM_16 with key size 288
Aug  3 11:20:52 7206 charon: 16[KNL]   using replay window of 32 packets
Aug  3 11:20:52 7206 charon: 16[KNL] sending XFRM_MSG_UPDSA 205: => 352 bytes @ 0x7f82d67fb500
```

We researched that any encryption algorithms in GCM or CCM mode cannot use integrity algorithms and strongswan just ignores this in config. But if we use aes256-sha2_512-modp4096! for ike and esp options, then Active and Passive nodes using integrity algorithms fine and HA works perfectly! But we must use only GCM algorithms in tests...

All settings are made correctly.

Self-compiled Linux kernel - 3.8.15 with ha patches x86_64
Strongswan 5.2.2 (./configure --sysconfdir=/etc --prefix=/usr/local --libexecdir=/usr/lib -enable-xauth-generic --enable-openssl --enable-curl --enable-ha --with-linux-headers=/usr/src/linux-3.18.5)

iptables -vnL | grep CLUSTERIP (on both nodes)

```
 170M  188G CLUSTERIP  all  --  eth0   *       0.0.0.0/0            192.168.100.197      CLUSTERIP
 hashmode=sourceip clustermac=01:00:5E:00:00:20 total_nodes=2 local_node=0 hash_init=0
    0     0 CLUSTERIP  all  --  eth1   *       0.0.0.0/0            192.168.25.10        CLUSTERIP
 hashmode=sourceip clustermac=01:00:5E:00:00:30 total_nodes=2 local_node=0 hash_init=0
```

ipsec.conf (same on both HA nodes):

```
config setup
        strictcrlpolicy=no
        uniqueids = yes

conn %default
    ikelifetime=24h
    keylife=5m
    rekeymargin=1m
    keyingtries=%forever
    mobike=yes
    rekey=no
    dpdaction=clear
    dpddelay=5s
    dpdtimeout=15s

conn cvpncln1
    keyexchange=ikev2
    ike=aes256gcm16-sha2_512-modp4096!
    esp=aes256gcm16-modp4096!
    left=192.168.100.197
    leftcert=cert1
    leftid="C=KZ, O=gamma, CN=7206"
    leftsubnet=192.168.25.0/24
    leftfirewall=yes
    right=192.168.100.207
    rightid="C=KZ, O=gamma, CN=7207"
    rightsubnet=192.168.26.0/24
    rightfirewall=yes
    auto=route
```

strongswan.conf (First HA node):

```
charon {
  load = curl pem pkcs1 x509 revocation openssl random nonce hmac aes des sha1 sha2 md5 xcbc ctr c
cm gcm stroke kernel-netlink socket-default updown xauth-generic ha
  plugins {
       ha {
          local = 192.168.25.2
          remote = 192.168.25.3
          segment_count = 2
          fifo_interface = yes
          monitor = yes
          resync = yes
          autobalance = 10
       }
  }

retransmit_base = 1
retransmit_timeout = 5
retransmit_tries = 5

    syslog {
        daemon {
```

```
            default = 3
            asn = -1
            net = -1
            esp = -1
            job = -1
            enc = -1
        }
        auth {
            ike_name = yes
            default = -1
            ike = 1
        }
    }
}
```

strongswan.conf (Second HA node):

```
charon {
  load = curl pem pkcs1 x509 revocation openssl random nonce hmac aes des sha1 sha2 md5 xcbc ctr c
cm gcm stroke kernel-netlink socket-default updown xauth-generic ha
  plugins {
       ha {
          local = 192.168.25.3
          remote = 192.168.25.2
          segment_count = 2
          fifo_interface = yes
          monitor = yes
          resync = yes
          autobalance = 10
       }
  }

retransmit_base = 1
retransmit_timeout = 5
retransmit_tries = 5

    syslog {
        daemon {
            default = 3
            asn = -1
            net = -1
            esp = -1
            job = -1
            enc = -1
        }
        auth {
            ike_name = yes
            default = -1
            ike = 1
        }
    }
}
```

ipsec.conf (initiator gateway to HA):

```
config setup
        strictcrlpolicy=no
        uniqueids = yes

conn %default
    ikelifetime=24h
    keylife=5m
    rekeymargin=1m
    keyingtries=%forever
    mobike=yes
    rekey=no
```

```
    dpdaction=clear
    dpddelay=5s
    dpdtimeout=15s

conn cvpncln1
    keyexchange=ikev2
    ike=aes256gcm16-sha2_512-modp4096!
    esp=aes256gcm16-modp4096!
    left=192.168.100.207
    leftcert=cert1
    leftid="C=KZ, O=gamma, CN=7207"
    leftsubnet=192.168.26.0/24
    leftfirewall=yes
    right=192.168.100.197
    rightid="C=KZ, O=gamma, CN=7206"
    rightsubnet=192.168.25.0/24
    rightfirewall=yes
    auto=route
```

strongswan.conf (initiator gateway to HA):

```
charon {
  load = curl pem pkcs1 x509 revocation openssl random nonce hmac aes des sha1 sha2 md5 xcbc ctr c
cm gcm stroke kernel-netlink socket-default updown xauth-generic

retransmit_base = 1
retransmit_timeout = 5
retransmit_tries = 5

    syslog {

        daemon {
            default = 3
            ans = -1
            enc = -1
            net = -1
            job = -1
            esp = -1
        }
        auth {
            ike_name = yes
            default = -1
            ike = 1
        }
    }
}
```

Main problem is Why HA node which is set up PASSIVE state receive "UNDEFINED" as integrity algorithm (but actually must ignore like ACTIVE node) in method of keymat_v2.c:

```
METHOD(keymat_v2_t, derive_child_keys ....

...

    if (proposal->get_algorithm(proposal, INTEGRITY_ALGORITHM,
                                &int_alg, &int_size))
    {
        DBG2(DBG_CHD, "  using %N for integrity",
             integrity_algorithm_names, int_alg);

        if (!int_size)
        {
            int_size = keymat_get_keylen_integ(int_alg);
        }
        if (!int_size)
        {
```

```
            DBG1(DBG_CHD, "no keylength defined for %N",
                 integrity_algorithm_names, int_alg);
            return FALSE;
        }
        /* to bytes */
        int_size /= 8;
    }
...
```

And in ha_dispatcher.c in function static void process_child_add:

```
static void process_child_add(private_ha_dispatcher_t *this,
                              ha_message_t *message)
{
...

    if (ike_sa->get_version(ike_sa) == IKEV2)
    {
        keymat_v2_t *keymat_v2 = (keymat_v2_t*)ike_sa->get_keymat(ike_sa);

        ok = keymat_v2->derive_child_keys(keymat_v2, proposal, dh,
                        nonce_i, nonce_r, &encr_i, &integ_i, &encr_r, &integ_r);
    }

...

    if (!ok)
    {
        DBG1(DBG_CHD, "HA CHILD_SA key derivation failed");
        child_sa->destroy(child_sa);
        proposal->destroy(proposal);
        charon->ike_sa_manager->checkin(charon->ike_sa_manager, ike_sa);
        return;
    }
...
```

Logs are included in files.

Any suggestions and help would be greatly appreciated

Associated revisions

Revision a7f381ef - 04.08.2015 11:23 - Tobias Brunner

ha: Properly initialize algo variables when installing CHILD_SAs

If AEAD algorithms are used no integrity algorithm will be received from
the other HA node. But since AUTH_UNDEFINED is 1024 and not 0 this value
was incorrectly added to the proposal, resulting in a failure during key
derivation. The variables are now explicitly initialized to 0, as already
was the case for the IKE SAs.

Fixes #1051.

History

#1 - 03.08.2015 09:40 - Alexander Sukhomlinov

Forgot to say, it affected to 5.3.2-3 version too.

#2 - 03.08.2015 13:40 - Tobias Brunner

- Subject changed from Passive node in HA configuration receive "UNDEFINED" as integrity algorithm to Passive node in HA configuration receive "UNDEFINED" as integrity algorithm if AEAD algorithm is used

- Category set to libcharon

- Status changed from New to Feedback

- Target version set to 5.3.3

Thanks for the report. The reason is that a couple of checks in source:src/libcharon/plugins/ha/ha_dispatcher.c are incorrect, in this particular case

the one here on line 732:

```
if (integ)
{
    proposal->add_algorithm(proposal, INTEGRITY_ALGORITHM, integ, 0);
}
```

Because integ is initialized to AUTH_UNDEFINED, which is defined as 1024, not 0, this value was incorrectly added to the proposal. The same is true for encr (ENCR_UNDEFINED is 1024 too). For some reason this was done differently for the IKE SAs in process_ike_add where encr and integ are explicitly initialized to 0, which is also the reason why using AEAD algorithms works there. So we can either fix the checks (e.g. integ != AUTH_UNDEFINED) or initialize the variables to 0. The latter is the easier fix and will result in consistent code for IKE and CHILD SAs. I pushed a patch for this to the *ha-aead* branch.

#3 - 04.08.2015 06:19 - Alexander Sukhomlinov

Thank for reply and info. Now it works.

#4 - 04.08.2015 11:25 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to Fixed

Thanks for testing. I pushed the fix to master.

Files

syslog_first_ha_node.log	340 KB	03.08.2015	Alexander Sukhomlinov
syslog_initiator_to_ha.log	398 KB	03.08.2015	Alexander Sukhomlinov
syslog_second_ha_node.log	221 KB	03.08.2015	Alexander Sukhomlinov