strongSwan - Issue #3529

Strongswan is allowing IKEv2 authentication using CRL that is signed by an Intermediate CA that does not have the crlSign keyUsage extension but has the CA basic constraint

29.07.2020 17:55 - Paulo Babau

Status:	Closed		
Priority:	Normal		
Assignee:	Tobias Brunner		
Category:	libstrongswan		
Affected version:	5.6.2	**Resolution:**	No change required

Description

Strongswan is allowing IKEv2 authentication using CRL that is signed by an Intermediate CA that does not have the crlSign keyUsage extension but has the CA basic constraint using only the openssl plugin for all crypto symmetric/asymmetric and x509v3/CRL related operations.

Test setup is as follows using 2 Linux machines:

Strongswan(5.3.5) VPN Peer1 <--> Strongswan(5.6.2) VPN Peer2

My Certificate Chain is as follows for both peers:

- RootCA => SubCA => SubSubCA => Peer 1 End Entity Cert
- RootCA => Peer 2 End Entity Cert

All CRLs are inside X509 Certs using X509v3 CRL Distribution Points available through http.

The Bad CA is SubCA which has CA basic constraint but no CRLSign extension bit set.

When Peer2 receives all certs and starts to verify the trust chain(including CRLs) it succeeds with IKEv2 authentication instead of failing!
All CRLs don't have any revoked certificates but they must be present and good because i'm using strict crl checking.

swanctl -x for Peer2:

```
List of X.509 CA Certificates

  subject:  "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
  issuer:   "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
  validity:  not before Jul 02 19:12:16 2020, ok
             not after  Jun 30 19:12:16 2030, ok (expires in 3622 days)
  serial:    37
  flags:     CA CRLSign
  CRL URIs:  http://192.168.144.244/subca-no-crl-key-usage-rsa.p7
  OCSP URIs: http://192.168.144.244:7791
  authkeyId: e1:7f:10:99:16:47:4f:97:a7:e8:f3:42:c9:b2:29:bf:60:03:8d:de
  subjkeyId: a7:6d:50:8f:f6:09:f0:3c:e2:8e:b1:59:41:40:0f:06:7c:c5:9b:39
  pubkey:    RSA 2048 bits
  keyid:     b6:e6:f3:00:6f:a7:8d:15:87:e4:2f:1a:17:92:ce:b4:f5:91:88:12
  subjkey:   a7:6d:50:8f:f6:09:f0:3c:e2:8e:b1:59:41:40:0f:06:7c:c5:9b:39

  subject:  "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
  issuer:   "XXXXXXXXXXX, CN=rootca-rsa"
  validity:  not before Jul 02 19:12:11 2020, ok
             not after  Jun 30 19:12:11 2030, ok (expires in 3622 days)
  serial:    17
  flags:     CA
  CRL URIs:  http://192.168.144.244/rootca-rsa.p7
  OCSP URIs: http://192.168.144.244:7777
  authkeyId: e6:8b:e8:d4:bc:1d:d9:61:83:69:76:c0:27:ae:2a:2d:88:e7:5a:58
  subjkeyId: e1:7f:10:99:16:47:4f:97:a7:e8:f3:42:c9:b2:29:bf:60:03:8d:de
  pubkey:    RSA 2048 bits
```

```
  keyid:     58:9c:e5:c0:70:d9:b7:d6:5b:f5:9c:a2:0a:bf:3d:5b:23:9c:6b:fd
  subjkey:   e1:7f:10:99:16:47:4f:97:a7:e8:f3:42:c9:b2:29:bf:60:03:8d:de

  subject:  "XXXXXXXXXXX, CN=rootca-rsa"
  issuer:   "XXXXXXXXXXX, CN=rootca-rsa"
  validity:  not before Jan 10 20:34:09 2020, ok
             not after  Jan 07 20:34:09 2030, ok (expires in 3449 days)
  serial:    01:00:01
  flags:     CA CRLSign self-signed
  authkeyId: e6:8b:e8:d4:bc:1d:d9:61:83:69:76:c0:27:ae:2a:2d:88:e7:5a:58
  subjkeyId: e6:8b:e8:d4:bc:1d:d9:61:83:69:76:c0:27:ae:2a:2d:88:e7:5a:58
  pubkey:    RSA 2048 bits
  keyid:     7a:d2:0a:fe:34:95:7c:19:c1:12:6b:25:08:b8:af:56:c4:c5:9f:7f
  subjkey:   e6:8b:e8:d4:bc:1d:d9:61:83:69:76:c0:27:ae:2a:2d:88:e7:5a:58

#swanctl -T

07[ENC] parsed IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG)
N(REDIR_SUP) ]
07[IKE] 10.46.82.31 is initiating an IKE_SA
07[IKE] sending cert request for "XXXXXXXXXXX, CN=rootca-rsa"
07[ENC] generating IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP)
 N(HASH_ALG) ]
06[ENC] parsed IKE_AUTH request 1 [ EF(1/4) ]
06[ENC] received fragment #1 of 4, waiting for complete IKE message
07[ENC] parsed IKE_AUTH request 1 [ EF(2/4) ]
07[ENC] received fragment #2 of 4, waiting for complete IKE message
08[ENC] parsed IKE_AUTH request 1 [ EF(3/4) ]
08[ENC] received fragment #3 of 4, waiting for complete IKE message
05[ENC] parsed IKE_AUTH request 1 [ EF(4/4) ]
05[ENC] received fragment #4 of 4, reassembling fragmented IKE message
05[ENC] parsed IKE_AUTH request 1 [ IDi CERT CERT CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi TSr
 N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
05[IKE] received cert request for "XXXXXXXXXXX, CN=rootca-rsa"
05[IKE] received 2 cert requests for an unknown ca
05[IKE] received end entity cert "XXXXXXXXXXX, CN=tlm3-16x.example.com"
05[IKE] received issuer cert "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
05[IKE] received issuer cert "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
05[CFG] looking for peer configs matching 10.46.44.120[CN=alpha]...10.46.82.31[XXXXXXXXXXX, CN=tlm
3-16x.example.com]
05[CFG] selected peer config 'SPD-1-1-9999'
05[CFG]   using certificate "XXXXXXXXXXX, CN=tlm3-16x.example.com"
05[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-
usage-rsa"
05[CFG] checking certificate status of "XXXXXXXXXXX, CN=tlm3-16x.example.com"
05[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
05[CFG] generating ocsp request failed
05[CFG] ocsp check failed, fallback to crl
05[CFG]   fetching crl from 'http://192.168.144.244/subsubca-issued-by-no-crl-key-usage-rsa.p7' ..
.
05[CFG]   using certificate "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
05[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
05[CFG]   reached self-signed root ca with a path length of 1
05[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
05[CFG]   crl correctly signed by "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
05[CFG]   crl is valid: until Aug 15 18:53:34 2020
05[CFG] certificate status is good
05[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX,  CN=subca-no-crl-key-usage-rsa"
05[CFG] checking certificate status of "XXXXXXXXXXX,  CN=subsubca-issued-by-no-crl-key-usage-rsa"
05[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
05[CFG] generating ocsp request failed
05[CFG] ocsp check failed, fallback to crl
05[CFG]   fetching crl from 'http://192.168.144.244/subca-no-crl-key-usage-rsa.p7' ...
05[CFG]   using certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
05[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
05[CFG]   reached self-signed root ca with a path length of 0
05[CFG]   crl correctly signed by "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
```

```
05[CFG]   crl is valid: until Aug 15 18:53:33 2020
05[CFG] certificate status is good
05[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
05[CFG] checking certificate status of "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
05[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
05[CFG] generating ocsp request failed
05[CFG] ocsp check failed, fallback to crl
05[CFG]   fetching crl from 'http://192.168.144.244/rootca-rsa.p7' ...
05[CFG]   using trusted certificate "XXXXXXXXXXX, CN=rootca-rsa"
05[CFG]   crl correctly signed by "XXXXXXXXXXX, CN=rootca-rsa"
05[CFG]   crl is valid: until Aug 15 18:53:33 2020
05[CFG] certificate status is good
05[CFG]   reached self-signed root ca with a path length of 2
05[IKE] authentication of 'XXXXXXXXXXX, CN=tlm3-16x.example.com' with RSA_EMSA_PKCS1_SHA2_256 succ
essful
05[IKE] authentication of 'CN=alpha' (myself) with RSA_EMSA_PKCS1_SHA2_256 successful
05[IKE] scheduling rekeying in 3446s
05[IKE] IKE_SA SPD-1-1-9999[3] established between 10.46.44.120[CN=alpha]...10.46.82.31[XXXXXXXXXX
X, CN=tlm3-16x.example.com]
05[IKE] scheduling reauthentication in 3369s
05[IKE] maximum IKE_SA lifetime 3729s
05[IKE] sending end entity cert "CN=alpha"
05[IKE] CHILD_SA SPD-1-1-1{3} established with SPIs c8e67137_i c7566a3c_o and TS 10.46.44.120/32[t
cp] === 10.46.82.31/32[tcp/3083]
05[ENC] generating IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) ]
05[ENC] splitting IKE message with length of 1692 bytes into 2 fragments
05[ENC] generating IKE_AUTH response 1 [ EF(1/2) ]
05[ENC] generating IKE_AUTH response 1 [ EF(2/2) ]
```

According to a mail thread, back in Aug/2017 in users mailing list, David Keane add the same issue as me(basically David's second question):

```
David Keane dkeane3000 at gmail.com
Fri Aug 4 00:54:19 CEST 2017
Previous message: [strongSwan] Revoking of own/local certificates and crlsign question
Next message: [strongSwan] Strongswan VPN Profile for Android.
Messages sorted by: [ date ] [ thread ] [ subject ] [ author ]
Hi Noel,

Thank you for the reply. Regarding the crlSign bit, in our environment we
can only accept if it has been signed by a CA AND has the crlSIgn bit set.
I changed the line in the code to "if (!(x509->get_flags(x509) &
X509_CRL_SIGN))" but it just seem to get ignored even when the CA cert on
the peer doesnt have the crlSIgn set. Is there something else in the code
that I am missing?

Thanks,

David

[strongSwan] Revoking of own/local certificates and crlsign question
Noel Kuntze noel.kuntze+strongswan-users-ml at thermi.consulting
Tue Aug 1 16:57:29 CEST 2017
Next message: [strongSwan] Revoking of own/local certificates and crlsign question
Messages sorted by: [ date ] [ thread ] [ subject ] [ author ]
Hello David,

strongSwan does not check its own certificate for revocation, because it does not improve security
.
Users/Administrators are expected to rotate the certificates, if they are revoked or run out.
If the remote peer checks for revocation, it will reject the certificate anyway.

Regarding the second query, charon checks if CRLSign OR the CA bit is set[1], if the x509 plugin i
s used.
It seems it was forgotten to check the X509_CRLSign bit in the openssl plugin's issued_by method[2
].
In the x509 plugin's, it is checked[1].
```

Going to "3.3. RevocatioN" in RFC5280[3], the following is stated:
" A CRL is a time-stamped list identifying revoked certificates that is signed by a CA or CRL issuer and made freely available in a public repository. "

As it explicitely mentions "or", I interpret this here as that either or both bits can be set.

Kind regards

Noel

[1] https://github.com/strongswan/strongswan/blob/master/src/libstrongswan/plugins/x509/x509_crl.c#L473
[2] https://github.com/strongswan/strongswan/blob/master/src/libstrongswan/plugins/openssl/openssl_crl.c#L262
[3] https://tools.ietf.org/html/rfc5280#section-3.3

On 31.07.2017 13:32, David Keane wrote:
> Hi all,
>
> My setup is as follows:
>
> Host 1 --> MyVPNGW --> PeerVPNGW --> Host 2
>
> MyVPNGW is also connected to a CA server that contains a CRL on /var/www/html/. I am using a ROOT CA and an Intermediate CA certificate on each side with their relevant client certs
>
> I have a few questions that I need answers in relation to strongswan's revocation procedures that maybe you can help with. My 1st issue is that if I add MyVPNGW's certificate to the CRL marked as revoked and then initiate the tunnel from MyVPNGW also, I find that the tunnel will establish, with or without strict-crl-policy and the CRL embedded in the certificate appears to be ignored. The CRLDP is embedded in the certificate itself as follows:
>
> X509v3 CRL Distribution Points:
>
> Full Name:
>   URI:http://192.168.1.1/crl.der
>
> I can see in the logs that the CRL was fetched correctly:
>
> received end entity cert "XXXXXXXXX"
>   using certificate "XXXXXX"
>   using trusted intermediate ca certificate "XXXXXX"
> *checking certificate status of "CN=PeerVPNGW"*
>   fetching crl from 'http://192.168.1.1/crl.der' ...
>   using trusted ca certificate "XXXXXX"
>   reached self-signed root ca with a path length of 0
>   using trusted certificate "XXXXXXXXXX"
>   crl correctly signed by "XXXXXXXXXX"
>   crl is valid: until Aug 11 23:27:53 2017
> certificate status is good
>   using trusted ca certificate "XXXXXXXXXXXX"
> *checking certificate status of "CN=intermediate_ca.test.com <http://intermediate_ca.test.com>"*
>   fetching crl from 'http://192.168.1.1/rootcrl.der' ...
>   using trusted certificate "XXXXXXXXXX"
>   crl correctly signed by "XXXXXXXXXXXX"
>   crl is valid: until Aug 11 14:47:48 2017
> certificate status is good
>   reached self-signed root ca with a path length of 1
> authentication of 'XXXXXXXXX' with ECDSA_WITH_SHA384_DER successful
> IKE_SA testsa[1] established between X.X.X.X[XXXXX]...X.X.X.X[XXXXXX]
> scheduling reauthentication in 86370s
> maximum IKE_SA lifetime 86400s
> connection 'testsa' established successfully
>
> If I mark the PeerVPNGW cert as revoked, the connection fails and I can see in the IPSec logs that the certificate was revoked. I notice in the logs (see above) that it only seems to check the c

```
ertificate status of the peer certs and not the local side, is that correct? Is there any way of g
etting strongswan to validate its local certificate against the CRL?
>
> My 2nd question is in relation CRLsign. My understanding of the standards is that the CRL should
 be ignored unless it was signed by a CA certificate that has the CRLsign bit set. I am finding th
at strongswan seems to ignore this. If I create a CRL from a certificate that doesnt have the CRLs
ign bit set and then revoke the PeerVPNGW cert, I find that that the connection fails as its seein
g the PeerVPNGW cert as being revoked. I would have expected it to ignore the CRL as the CRLsign b
it wasnt set and it shouldnt be recognised as a valid CRL. Just wondering what your opinions are o
n this?
>
> Thank you,
>
> David
```

I've also tested the hint by Noel on ..../src/libstrongswan/plugins/openssl/openssl_crl.c for the public "issued_by" method but it didn't work. :-(

Applied the following patch:

```
diff --git a/src/libstrongswan/plugins/openssl/openssl_crl.c b/src/libstrongswan/plugins/openssl/o
penssl_crl.c
index 88f7a67c2..1eab6d782 100644
--- a/src/libstrongswan/plugins/openssl/openssl_crl.c
+++ b/src/libstrongswan/plugins/openssl/openssl_crl.c
@@ -296,7 +296,8 @@ METHOD(certificate_t, issued_by, bool,
                return FALSE;
        }
        x509 = (x509_t*)issuer;
-       if (!(x509->get_flags(x509) & X509_CA))
+
+       if (!(x509->get_flags(x509) & (X509_CA | X509_CRL_SIGN)))
        {
                return FALSE;
        }
```

Is this an already known issue?

Can you help me?

History

#1 - 29.07.2020 18:09 - Paulo Babau

Forgot to add the End Entity Certficates for Peer2:

```
root@stpm8_20_14:~# swanctl -x

List of X.509 End Entity Certificates

  subject:  "XXXXXXXXX, CN=tlm3-16x.example.com"
  issuer:   "XXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
  validity:  not before Jul 02 19:12:24 2020, ok
             not after  Jun 30 19:12:24 2030, ok (expires in 3622 days)
  serial:    69
  altNames:  ldap:///cn=tlm3-16x.example.com, XXXXXXXXX
  flags:     serverAuth clientAuth
  CRL URIs:  http://192.168.144.244/subsubca-issued-by-no-crl-key-usage-rsa.p7
  OCSP URIs: http://192.168.144.244:7807
  authkeyId: a7:6d:50:8f:f6:09:f0:3c:e2:8e:b1:59:41:40:0f:06:7c:c5:9b:39
  subjkeyId: fc:c2:a8:f2:38:8c:7b:77:04:20:6b:87:41:3c:64:2c:54:fe:ce:86
  pubkey:    RSA 2048 bits
  keyid:     67:6b:0d:1f:24:d1:f4:bf:bd:2f:78:bf:85:f1:a0:f0:15:ec:6d:5a
  subjkey:   fc:c2:a8:f2:38:8c:7b:77:04:20:6b:87:41:3c:64:2c:54:fe:ce:86

  subject:  "CN=alpha"
  issuer:   "XXXXXXXXX, CN=rootca-rsa"
  validity:  not before Jul 21 13:11:00 2020, ok
```

```
            not after  Jul 21 13:11:00 2022, ok (expires in 721 days)
  serial:    70:f8:7d:d9:7f:46:21:68:4b:0d:49:1d:d9:3e:97:fd:01:85:1c:22
  altNames:  ldap:///cn=alpha
  flags:     serverAuth
  authkeyId: e6:8b:e8:d4:bc:1d:d9:61:83:69:76:c0:27:ae:2a:2d:88:e7:5a:58
  subjkeyId: b8:d8:5d:4f:12:b9:6e:9c:e8:4b:e4:3f:75:d2:a4:5e:94:22:14:65
  pubkey:    RSA 3072 bits, has private key
  keyid:     6a:9c:49:f9:96:3d:5b:97:8b:f9:32:b4:06:7d:f3:03:49:4e:d6:8f
  subjkey:   b8:d8:5d:4f:12:b9:6e:9c:e8:4b:e4:3f:75:d2:a4:5e:94:22:14:65
```

#2 - 29.07.2020 18:23 - Paulo Babau

- File strongswan_ipsec_statusall.log added

#3 - 29.07.2020 18:48 - Tobias Brunner

- Status changed from New to Feedback

> Strongswan is allowing IKEv2 authentication using CRL that is signed by an Intermediate CA that does not have the crlSign keyUsage extension but has the CA basic constraint using only the openssl plugin for all crypto symmetric/asymmetric and x509v3/CRL related operations.

As Noel already mentioned in the email you quoted, accepting a CRL that's issued by a CA without cRLSign keyUsage is in compliance with RFC 5280. The keyUsage is basically to allow non-CA certificates to sign CRLs. The latter is what the *openssl* plugin currently doesn't allow for some reason (I guess Noel is right that it was simply overlooked).

> I've also tested the hint by Noel on ..../src/libstrongswan/plugins/openssl/openssl_crl.c for the public "issued_by" method but it didn't work. :-(
>
> Applied the following patch:
> [...]

The patch fixes the mentioned behavior above (i.e. so that both the *x509* and *openssl* plugins accept cRLSign issuers that are not CAs) and I now pushed it to the *3529-openssl-crlsign* branch. But it does not change the code to what you want. If you **only** want to accept issuer certificates with cRLSign keyUsage (which violates the RFC, I guess) you obviously have to remove the X509_CA flag from that if statement.

#4 - 29.07.2020 19:51 - Paulo Babau

The behavior that i'm expecting(whether is RFC compliant or not) is that by providing an Intermediate CA("XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa" which has signed a valid CRL => http://192.168.144.244/subca-no-crl-key-usage-rsa.p7) with flags "CA" set but not "CRLSign" should be detected when verifying the trust chain(in this case at peer 2) and fail the IKEv2 authentication. Is this possible?

#5 - 30.07.2020 08:26 - Tobias Brunner

> The behavior that i'm expecting(whether is RFC compliant or not) is that by providing an Intermediate CA("XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa" which has signed a valid CRL => http://192.168.144.244/subca-no-crl-key-usage-rsa.p7) with flags "CA" set but not "CRLSign" should be detected when verifying the trust chain(in this case at peer 2) and fail the IKEv2 authentication.

I really don't see the point in that.

> Is this possible?

Please read what I wrote in my previous comment more closely.

#6 - 30.07.2020 17:35 - Paulo Babau

Forgive me for my laziness. Here is the final patch that i applied to openssl_crl.c@issued_by(...) but without any success:

```
diff --git a/src/libstrongswan/plugins/openssl/openssl_crl.c b/src/libstrongswan/plugins/openssl/openssl_crl.c
index 88f7a67c2..5b92ca3d9 100644
--- a/src/libstrongswan/plugins/openssl/openssl_crl.c
+++ b/src/libstrongswan/plugins/openssl/openssl_crl.c
@@ -290,17 +290,32 @@ METHOD(certificate_t, issued_by, bool,
        x509_t *x509;
        ASN1_BIT_STRING *sig;
        bool valid;
+       char serial_str[512] = {0};
+       chunk_t serial_chunk, serial_hex_chunk;
```

```
        if (issuer->get_type(issuer) != CERT_X509)
        {
                return FALSE;
        }
-       x509 = (x509_t*)issuer;
-       if (!(x509->get_flags(x509) & X509_CA))
-       {
-               return FALSE;
-       }
-       key = issuer->get_public_key(issuer);
+
+    x509 = (x509_t*) issuer;
+       serial_chunk = x509->get_serial(x509);
+       serial_hex_chunk = chunk_to_hex(serial_chunk, serial_str, TRUE);
+       DBG1(DBG_LIB, "issuer(%s) certificate flags=%x", serial_str, x509->get_flags(x509));
+
+    if(!(x509->get_flags(x509) & X509_CA))
+    {
+        DBG1(DBG_LIB, "CRL issuer doesn't have the X.509v3 CA extension bit on!");
+        return FALSE;
+    }
+
+    if(!(x509->get_flags(x509) & X509_CRL_SIGN))
+    {
+               DBG1(DBG_LIB, "CRL issuer doesn't have the X.509v3 CRLSign extension bit on!");
+        return FALSE;
+    }
+
+    key = issuer->get_public_key(issuer);
        if (!key)
        {
                return FALSE;
```

Serial numbers for Root and Intermediate CA:
"XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa" => 37
"XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa" => 17
"XXXXXXXXXXX, CN=rootca-rsa" => 01:00:01

Here is the output for "swanctl -T" at peer2:

```
08[ENC] parsed IKE_SA_INIT request 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG) N(REDIR_SUP)
 ]
08[IKE] 10.46.82.31 is initiating an IKE_SA
08[IKE] sending cert request for "C=US, ST=MD, L=Catonsville, O=GSS, E=rootca-rsa@gossamersec.com, CN=rootca-r
sa"
08[ENC] generating IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(HASH_ALG)
 ]
06[ENC] parsed IKE_AUTH request 1 [ EF(1/4) ]
06[ENC] received fragment #1 of 4, waiting for complete IKE message
08[ENC] parsed IKE_AUTH request 1 [ EF(2/4) ]
08[ENC] received fragment #2 of 4, waiting for complete IKE message
05[ENC] parsed IKE_AUTH request 1 [ EF(3/4) ]
05[ENC] received fragment #3 of 4, waiting for complete IKE message
07[ENC] parsed IKE_AUTH request 1 [ EF(4/4) ]
07[ENC] received fragment #4 of 4, reassembling fragmented IKE message
07[ENC] parsed IKE_AUTH request 1 [ IDi CERT CERT CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi TSr N(EAP_ONLY)
 N(MSG_ID_SYN_SUP) ]
07[IKE] received cert request for "XXXXXXXXXXX, CN=rootca-rsa"
07[IKE] received end entity cert "XXXXXXXXXXX, CN=tlm3-16x.example.com"
07[IKE] received 2 cert requests for an unknown ca
07[IKE] received issuer cert "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
07[IKE] received issuer cert "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
07[CFG] looking for peer configs matching 10.46.44.120[CN=alpha]...10.46.82.31[XXXXXXXXXXX, CN=tlm3-16x.exampl
e.com]
07[CFG] selected peer config 'SPD-1-1-9999'
07[CFG]   using certificate "XXXXXXXXXXX, CN=tlm3-16x.example.com"
07[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
07[CFG] checking certificate status of "XXXXXXXXXXX, CN=tlm3-16x.example.com"
07[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
07[CFG] generating ocsp request failed
07[CFG] ocsp check failed, fallback to crl
07[CFG]   fetching crl from 'http://192.168.144.244/subsubca-issued-by-no-crl-key-usage-rsa.p7' ...
07[CFG]   using certificate "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
07[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
```

```
07[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
07[CFG]   reached self-signed root ca with a path length of 1

07[LIB] issuer(37) certificate flags=81 <============= my own debug please disregard

07[CFG]   crl correctly signed by "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
07[CFG]   crl is valid: until Aug 15 18:53:34 2020
07[CFG] certificate status is good
07[CFG]   using untrusted intermediate certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
07[CFG] checking certificate status of "XXXXXXXXXXX, CN=subsubca-issued-by-no-crl-key-usage-rsa"
07[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
07[CFG] generating ocsp request failed
07[CFG] ocsp check failed, fallback to crl
07[CFG]   fetching crl from 'http://192.168.144.244/subca-no-crl-key-usage-rsa.p7' ...
07[CFG]   using certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
07[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
07[CFG]   reached self-signed root ca with a path length of 0

07[LIB] issuer(17) certificate flags=1  <============= my own debug please disregard
07[LIB] CRL issuer doesn't have the X.509v3 CRLSign extension bit on!  <============= my own debug please disr
egard

07[CFG]   using certificate "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
07[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
07[CFG]   reached self-signed root ca with a path length of 0

07[LIB] issuer(17) certificate flags=1  <============= my own debug please disregard
07[LIB] CRL issuer doesn't have the X.509v3 CRLSign extension bit on!  <============= my own debug please disr
egard

07[CFG] crl response verification failed          <=========== CRL of malformed CA(serial number 17 with subje
ctName = "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa") verification as failed! Shouldn't this be enough to fai
l the authentication process?

07[CFG] certificate status is not available
07[CFG]   using trusted ca certificate "XXXXXXXXXXX, CN=rootca-rsa"
07[CFG] checking certificate status of "XXXXXXXXXXX, CN=subca-no-crl-key-usage-rsa"
07[LIB] building CRED_CERTIFICATE - OCSP_REQUEST failed, tried 0 builders
07[CFG] generating ocsp request failed
07[CFG] ocsp check failed, fallback to crl
07[CFG]   fetching crl from 'http://192.168.144.244/rootca-rsa.p7' ...
07[CFG]   using trusted certificate "XXXXXXXXXXX, CN=rootca-rsa"

07[LIB] issuer(010001) certificate flags=a1  <============= my own debug please disregard

07[CFG]   crl correctly signed by "XXXXXXXXXXX, CN=rootca-rsa"
07[CFG]   crl is valid: until Aug 15 18:53:33 2020
07[CFG] certificate status is good
07[CFG]   reached self-signed root ca with a path length of 2
07[IKE] authentication of 'XXXXXXXXXXX, CN=tlm3-16x.example.com' with RSA_EMSA_PKCS1_SHA2_256 successful
07[IKE] authentication of 'CN=alpha' (myself) with RSA_EMSA_PKCS1_SHA2_256 successful
07[IKE] IKE_SA SPD-1-1-9999[2] established between 10.46.44.120[CN=alpha]...10.46.82.31[XXXXXXXXXXX, CN=tlm3-1
6x.example.com]
07[IKE] scheduling rekeying in 3300s
07[IKE] scheduling reauthentication in 3280s
07[IKE] maximum IKE_SA lifetime 3640s
07[IKE] sending end entity cert "CN=alpha"
07[IKE] CHILD_SA SPD-1-1-1{2} established with SPIs c005332e_i c2d3b8ba_o and TS 10.46.44.120/32[tcp] === 10.4
6.82.31/32[tcp/3083]
07[ENC] generating IKE_AUTH response 1 [ IDr CERT AUTH SA TSi TSr N(AUTH_LFT) ]
07[ENC] splitting IKE message with length of 1692 bytes into 2 fragments
07[ENC] generating IKE_AUTH response 1 [ EF(1/2) ]
07[ENC] generating IKE_AUTH response 1 [ EF(2/2) ]
```

Do you know why the failure, on CRL verification, of a "malformed"(i'm saying this because of the requirement to have a CA that issues CRLs to always have the CA and CRLSign bit on) CA doesn't fail the authentication process?

#7 - 31.07.2020 08:56 - Tobias Brunner

That's because your version is too old. Strict CRL verification for intermediate CA certificates has only been implemented since 5.6.3.

#8 - 31.07.2020 10:55 - Paulo Babau

Thanks for all the help, will try to backport(unfortunately due to hard requirements i'm stuck with 5.6.2 version :-( ) these fixes from 5.6.3(although the most important fix for me is the first fix):

- CRL validation results other than revocation (e.g. a skipped check because the CRL couldn't be fetched) are now stored also for intermediate CA certificates and not only for end-entity certificates, so a strict CRL policy can be enforced in such cases.
- The issuer of fetched CRLs is now compared to the issuer of the checked certificate (#2608).
- CRLs that are not yet valid are now ignored to avoid problems in scenarios where expired certificates are removed from new CRLs and the clock on the host doing the revocation check is trailing behind that of the host issuing CRLs. Not doing this could result in accepting a revoked and expired certificate, if it's still valid according to the trailing clock but not contained anymore in not yet valid CRLs.

Again thanks for all your help. :-)

#9 - 31.07.2020 13:12 - Tobias Brunner

- Status changed from Feedback to Closed

- Assignee set to Tobias Brunner

- Resolution set to No change required

Files

strongswan_ipsec_statusall.log	2.82 KB	29.07.2020	Paulo Babau