strongSwan - Bug #737

Split tunneling works in Android but not in iOS, OSX and Windows

16.10.2014 08:14 - Eric Lim

Status:	Closed	**Start date:**	16.10.2014
Priority:	Normal	**Due date:**	
Assignee:	Tobias Brunner	**Estimated time:**	0.00 hour
Category:	configuration		
Target version:	5.2.2		
Affected version:	5.1.2	**Resolution:**	Fixed

Description

Hi!

I have a VPN server with strongSwan U5.1.2/K3.13.0-32-generic which I want to enable split tunneling.
All platforms can access the VPN server and can browse the internet when split tunneling is disabled (leftsubnet=0.0.0.0/0).
But when I try to set several subnets in the leftsubnet to enable the split tunneling some issues occurred.
I also added leftfirewall=yes in my config and issued this iptables rules (iptables -t nat -A POSTROUTING -s 172.16.16.0/24 -o eth0 -m policy --dir out --pol ipsec -j ACCEPT &&
iptables -t nat -A POSTROUTING -s 172.16.16.0/24 -o eth0 -j MASQUERADE) when I enabled the split tunneling.

- **In Windows, I can connect to the VPN but can't browse the internet.**
- **In iOS, I can connect to the VPN and can also browse the web but not the sites included in the leftsubnet.**
- **In OSX, VPN connection is established but destroyed immediately.**

I don't have any issues for Android though and split tunnel works as expected.

Tested platforms:
- Android 4.0+
- iOS 8.0.2
- Windows 7 SP1 x64
- OS X 10.9.5

Here's how my current config looks:

```
# ipsec.conf - strongSwan IPsec configuration file

config setup
        uniqueids=never
        # strictcrlpolicy=yes
        charondebug="cfg 2, dmn 2, ike 2, net 2" 

conn %default
        ike=aes128-sha256-ecp256,aes256-sha384-ecp384,aes128-sha256-modp2048,aes128-sha1-modp2048,aes256-sha384-modp4096,aes256-sha256-modp4096,aes256-sha1-modp4096,aes128-sha256-modp1536,aes128-sha1-modp1536,aes256-sha384-modp2048,aes256-sha256-modp2048,aes256-sha1-modp2048,aes128-sha256-modp1024,aes128-sha1-modp1024,aes256-sha384-modp1536,aes256-sha256-modp1536,aes256-sha1-modp1536,aes256-sha384-modp1024,aes256-sha256-modp1024,aes256-sha1-modp1024!
        esp=aes128gcm16-ecp256,aes256gcm16-ecp384,aes128-sha256-ecp256,aes256-sha384-ecp384,aes128-sha256-modp2048,aes128-sha1-modp2048,aes256-sha384-modp4096,aes256-sha256-modp4096,aes256-sha1-modp4096,aes128-sha256-modp1536,aes128-sha1-modp1536,aes256-sha384-modp2048,aes256-sha256-modp2048,aes256-sha1-modp2048,aes128-sha256-modp1024,aes128-sha1-modp1024,aes256-sha384-modp1536,aes256-sha256-modp1536,aes256-sha1-modp1536,aes256-sha384-modp1024,aes256-sha256-modp1024,aes256-sha1-modp1024,aes128gcm16,aes256gcm16,aes128-sha256,aes128-sha1,aes256-sha384,aes256-sha256,aes256-sha1!
        dpdaction=clear
        dpddelay=300s
        rekey=yes
        left=%any
        # I only want to tunnel Google sites. So, the list are some of Google's subnet
        leftsubnet=208.64.38.55/32,66.7.213.62/32,66.171.248.172/32,64.233.160.0/19,64.102.0.0/20,64.249.64.0/19,72.14.192.0/18,74.125.0.0/16,209.85.128.0/17,216.239.32.0/19,173.194.0.0/16
        leftcert=caCert.pem
        leftfirewall=yes
        right=%any
        rightdns=8.8.8.8,8.8.4.4
        rightsourceip=172.16.16.0/24

conn IPSec-IKEv2
        keyexchange=ikev2
        auto=add

# Currently used by Windows and Android
conn IKEv2-EAP
        also="IPSec-IKEv2"
        rightauth=eap-mschapv2
        rightsendcert=never
        eap_identity=%any

conn IPSec-IKEv1
        keyexchange=ikev1
        auto=add

# Currently used by OSX and iOS
conn IKEv1-XAUTHPSK
        also="IPSec-IKEv1"
        xauth=server
        authby=xauthpsk

# Optional conn for OSX and iOS
conn CiscoIPSec
        also="IPSec-IKEv1"
        rightauth=pubkey
        rightauth2=xauth
```

The unity plugin is enabled in strongswan.conf

```
charon {
        load_modular = yes
        plugins {
                include strongswan.d/charon/*.conf
        }
        cisco_unity = yes
}

include strongswan.d/*.conf
```

ipsec statusall result:

```
Status of IKE charon daemon (strongSwan 5.1.2, Linux 3.13.0-32-generic, x86_64):
  uptime: 5 seconds, since Oct 16 01:22:04 2014
  malloc: sbrk 1617920, mmap 0, used 564480, free 1053440
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 0
  loaded plugins: charon test-vectors sqlite aes rc2 sha1 sha2 md4 md5 rdrand random nonce x509 re
vocation constraints pkcs1 pkcs7 pkcs8 pkcs12 pem openssl fips-prf xcbc cmac hmac ctr ccm gcm attr
 kernel-netlink resolve socket-default stroke updown eap-identity eap-sim eap-sim-pcsc eap-aka eap
-aka-3gpp2 eap-simaka-pseudonym eap-simaka-reauth eap-md5 eap-gtc eap-mschapv2 eap-dynamic eap-rad
ius eap-tls eap-ttls eap-peap eap-tnc xauth-generic xauth-eap xauth-noauth tnc-tnccs tnccs-20 tncc
s-11 tnccs-dynamic dhcp lookip error-notify led addrblock unity
Virtual IP pools (size/online/offline):
  172.16.16.0/24: 254/0/0
Listening IP addresses:
  104.131.111.153
Connections:
 IPSec-IKEv2:  %any...%any  IKEv2, dpddelay=300s
 IPSec-IKEv2:   local:  [C=US, O=strongSwan, CN=104.131.111.153] uses public key authentication
 IPSec-IKEv2:    cert:  "C=US, O=strongSwan, CN=104.131.111.153"
 IPSec-IKEv2:   remote: uses public key authentication
 IPSec-IKEv2:   child:  208.64.38.55/32 66.7.213.62/32 66.171.248.172/32 64.233.160.0/19 64.102.0.
```

```
0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/19 173.194.0.0/16 ==
= dynamic TUNNEL, dpdaction=clear
   IKEv2-EAP:  %any...%any  IKEv2, dpddelay=300s
   IKEv2-EAP:   local:  [C=US, O=strongSwan, CN=104.131.111.153] uses public key authentication
   IKEv2-EAP:    cert:  "C=US, O=strongSwan, CN=104.131.111.153"
   IKEv2-EAP:   remote: uses EAP_MSCHAPV2 authentication with EAP identity '%any'
   IKEv2-EAP:   child:  208.64.38.55/32 66.7.213.62/32 66.171.248.172/32 64.233.160.0/19 64.102.0.
0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/19 173.194.0.0/16 ==
= dynamic TUNNEL, dpdaction=clear
 IPSec-IKEv1:  %any...%any  IKEv1, dpddelay=300s
 IPSec-IKEv1:   local:  [C=US, O=strongSwan, CN=104.131.111.153] uses public key authentication
 IPSec-IKEv1:    cert:  "C=US, O=strongSwan, CN=104.131.111.153"
 IPSec-IKEv1:   remote: uses public key authentication
 IPSec-IKEv1:   child:  208.64.38.55/32 66.7.213.62/32 66.171.248.172/32 64.233.160.0/19 64.102.0.
0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/19 173.194.0.0/16 ==
= dynamic TUNNEL, dpdaction=clear
IKEv1-XAUTHPSK:  %any...%any  IKEv1, dpddelay=300s
IKEv1-XAUTHPSK:   local:  [C=US, O=strongSwan, CN=104.131.111.153] uses pre-shared key authenticat
ion
IKEv1-XAUTHPSK:    cert:  "C=US, O=strongSwan, CN=104.131.111.153"
IKEv1-XAUTHPSK:   remote: uses pre-shared key authentication
IKEv1-XAUTHPSK:   remote: uses XAuth authentication: any
IKEv1-XAUTHPSK:   child:  208.64.38.55/32 66.7.213.62/32 66.171.248.172/32 64.233.160.0/19 64.102.
0.0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/19 173.194.0.0/16
=== dynamic TUNNEL, dpdaction=clear
  CiscoIPSec:  %any...%any  IKEv1, dpddelay=300s
  CiscoIPSec:   local:  [C=US, O=strongSwan, CN=104.131.111.153] uses public key authentication
  CiscoIPSec:    cert:  "C=US, O=strongSwan, CN=104.131.111.153"
  CiscoIPSec:   remote: uses public key authentication
  CiscoIPSec:   remote: uses XAuth authentication: any
  CiscoIPSec:   child:  208.64.38.55/32 66.7.213.62/32 66.171.248.172/32 64.233.160.0/19 64.102.0.
0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/19 173.194.0.0/16 ==
= dynamic TUNNEL, dpdaction=clear
Security Associations (0 up, 0 connecting):
  none
```

And here's my firewall rules:

nat

```
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT      !esp  --  anywhere             anywhere             to:104.131.111.153
ACCEPT     all  --  172.16.16.0/24       anywhere             policy match dir out pol ipsec
MASQUERADE  all  --  172.16.16.0/24       anywhere
```

filter

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     udp  --  anywhere             anywhere             udp dpt:isakmp
ACCEPT     udp  --  anywhere             anywhere             udp dpt:ipsec-nat-t
ACCEPT     esp  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
```

```
target     prot opt source               destination
ACCEPT     esp  --  anywhere             anywhere
```

Also, one weird thing in iOS split tunneling is it tunnels only the first subnet assigned in the leftsubnet, but others following it does not.
Am I missing something here? Do you have any ideas why this issues are occurring when I try to enable the split tunneling?
I had attached some log files for each platform when I tried to connect to them.
I hope all the details I had provided will help solving this issue.
Looking forward for your reply.

My Best,

Eric

Associated revisions

Revision 02df52fd - 05.12.2014 10:12 - Tobias Brunner

unity: Only do narrowing of responder's TS if we received 0.0.0.0/0

iOS and Mac OS X clients establish individual IPsec SAs for the traffic
selectors received in Split-Include attributes (might have been different
in earlier releases). If we return 0.0.0.0/0 as TSr that either results
in a bunch of Quick Mode exchanges (for each TS), or with the latest
client releases an error notify (ATTRIBUTES_NOT_SUPPORTED).
We also can't install the IPsec SA with all configured subnets as that
would cause conflicts if the client later negotiates SAs for other subnets,
which iOS 8 does based on traffic to such subnets.

For Shrew and the Cisco client, which propose 0.0.0.0/0, we still need to
override the narrowed TS with 0.0.0.0/0, as they otherwise won't accept
the Quick Mode response. Likewise, we also have to narrow the TS before
installing the IPsec SAs and policies.

So we basically have to follow the client's proposal and only modify TSr
if we received 0.0.0.0/0. Since we don't get the original TS in the
narrow hook we handle the inbound QM messages and make note of IKE_SAs on
which we received a TSr of 0.0.0.0/0.

Fixes #737.

History

#1 - 16.10.2014 16:58 - Tobias Brunner

- Tracker changed from Issue to Bug

- Status changed from New to Feedback

- Priority changed from High to Normal

- **In Windows, I can connect to the VPN but can't browse the internet.**

Windows 7 supports split tunneling only in very specific circumstances. Please have a look at ForwardingAndSplitTunneling.

- **In iOS, I can connect to the VPN and can also browse the web but not the sites included in the leftsubnet.**

```
...
Oct 16 01:59:41 server charon: 16[CFG] sending UNITY_SPLIT_INCLUDE: 208.64.38.55/32 66.7.213.62/32 66.171.248.
172/32 64.233.160.0/19 64.102.0.0/20 64.249.64.0/19 72.14.192.0/18 74.125.0.0/16 209.85.128.0/17 216.239.32.0/
19 173.194.0.0/16
...
Oct 16 01:59:42 server charon: 04[CFG] selecting traffic selectors for us:
Oct 16 01:59:42 server charon: 04[CFG]  config: 208.64.38.55/32, received: 208.64.38.55/32 => match: 208.64.38
.55/32
Oct 16 01:59:42 server charon: 04[CFG]  config: 66.7.213.62/32, received: 208.64.38.55/32 => no match
...
```

While a list of all subnets is sent in a Unity Split-Include payload (courtesy of the unity plugin), the iOS client seems to return the first one instead of proposing 0.0.0.0/0 in its Quick Mode request. Interestingly though, it then continues to initiate Quick Mode exchanges, apparently for each individual remote subnet. That's not really how strongSwan expects Cisco Unity to work, it expects a single Quick Mode exchange with a remote subnet of 0.0.0.0/0 (but since Unity is not standardized or even documented...).

Which version of iOS do you use? If you use iOS 8 (and the upcoming Mac OS X Yosemite) you might want to consider using IKEv2, see AppleIKEv2Profile.

- **In OSX, VPN connection is established but destroyed immediately.**

Like the iOS client it sends the first subnet instead of 0.0.0.0/0 during Quick Mode. But then it sends back an ATTRIBUTES_NOT_SUPPORTED notify and deletes the IKE_SA. Could be because strongSwan returns 0.0.0.0/0 as its traffic selector since 5.1.2.

I actually did some tests now with iOS 8, and I see it return ATTRIBUTES_NOT_SUPPORTED like Mac OS X did in your tests. As far as I could determine it is, in fact, caused by f8262aa1a624, which is required for interoperability with Shrew and the original Cisco client (and maybe others). If you don't have any such clients you may revert that commit to fix the issue.

Otherwise, it looks like we have to return the original traffic selector the client proposed (the first subnet for iOS/Mac, and 0.0.0.0/0 for the others). Unfortunately, we don't have that information available when we narrow/modify the traffic selector in the unity plugin. In the *unity-ios* branch I've implemented a workaround though. It's kind of a hack, but it works fine for iOS 8, Shrew and the Cisco client in my tests.

#2 - 23.10.2014 11:59 - Alexander Kalashnikov

Hello,

Cisco devices are using 0.0.0.0/0 TS for one shared SA negotiation, when Unity is used.
The list of IPs placed in Unity payloads are used for routing purpose only.

ShrewSoft VPN has option to use such configuration. Actually it is so by default [1].

Whoever Blackberry or Cisco VPN client has not such option, so it fails to negotiate phase 2 because of invalid configuration.

So, it seems like you should consider refactoring of overall logic. Like split TS for SA and TS in Cisco Unity by adding some additional **left(right)subnet** parameter for this separate entities.

____

[1] https://www.shrew.net/static/help-2.1.x/vpnhelp.htm?PolicySettings.html

#3 - 23.10.2014 12:21 - Tobias Brunner

> Whoever Blackberry or Cisco VPN client has not such option, so it fails to negotiate phase 2 because of invalid configuration.

I tested with the Cisco client (and Shrew and iOS 8) and it worked fine. I've no Blackberry so I can't test that. But as far as I know Blackberries support IKEv2, so this is not really an issue. Did you try the code in the *unity-ios* branch?

> So, it seems like you should consider refactoring of overall logic. Like split TS for SA and TS in Cisco Unity by adding some additional **left(right)subnet** parameter for this separate entities.

Yes, a unity plugin specific option to define these subnets, while configuring *leftsubnet=0.0.0.0/0* would probably help, but until it's easy to define such options in ipsec.conf/swanctl.conf this won't happen.

#4 - 05.12.2014 10:35 - Tobias Brunner

- Status changed from Feedback to Closed

- Target version set to 5.2.2

- Resolution set to Fixed

I did some additional tests (with actual traffic to different subnets). It seems that iOS does negotiate separate IPsec SAs for each subnet passed via Unity. And at least iOS 8.1.1 does so only based on traffic, only the first IPsec SA is established automatically. Therefore, similar to the traffic selector returned to the client, the *unity* plugin should only replace the TS to be installed if the client proposed 0.0.0.0/0 and otherwise just accept the client's proposal of one of these subnets.

The associated commit does so.

#5 - 26.12.2014 09:19 - Len Relsson

This patch from revision 02df52fd fixes MacOSX client but it breaks native Cisco VPN client.

It can be reproduced easily every time.

I have applied only this single patch to strongswan-5.2.1.

After applying the patch, I can't connect anymore using Cisco VPN client, neither when using the same strongSwan as a client.

After successful authentication (RSA+XAUTH) and after sending two subnets (UNITY_SPLIT_INCLUDE) I'm getting this error on server side every time:

```
Dec 26 01:36:53 vpnserver charon: 13[IKE] no matching CHILD_SA config found
Dec 26 01:36:53 vpnserver charon: 13[ENC] generating INFORMATIONAL_V1 request 942675489 [ HASH N(INVAL_ID) ]
Dec 26 01:36:53 vpnserver charon: 13[NET] sending packet: from *ipserver*[500] to *ipremote*[500] (76 bytes)
```

After removing the patch Cisco VPN Client and strongSwan as a client work like before.

#6 - 29.12.2014 09:31 - Alexander Ostapchuk

I was build ver 5.2.2rc1 and Cisco VPN client work as expected.

This patch also req this patch https://git.strongswan.org/?p=strongswan.git;a=commit;h=bf5d0693efe8aca8c1b87457ed2da322d72a23fa and may be more.

Len Relsson wrote:

> This patch from revision 02df52fd fixes MacOSX client but it breaks native Cisco VPN client.
>
> It can be reproduced easily every time.
>
> I have applied only this single patch to strongswan-5.2.1.

#7 - 30.12.2014 02:44 - Len Relsson

Alexander Ostapchuk wrote:

> I was build ver 5.2.2rc1 and Cisco VPN client work as expected.

I can confirm that 5.2.2rc1 works with Cisco VPN client. Thanks.

Still I can't get Shrew client to work with 5.2.2rc1.
I'm getting the same message as before:

```
no matching CHILD_SA config found
```

Files

android-logs.txt	41.4 KB	16.10.2014	Eric Lim
ios-logs.txt	176 KB	16.10.2014	Eric Lim
osx-logs.txt	54.6 KB	16.10.2014	Eric Lim
windows-logs.txt	45 KB	16.10.2014	Eric Lim