strongSwan - Issue #2203

Protecting symetric traffic using high availability in gateway to gateway setup (both active)

03.01.2017 10:26 - Ido Ramati

Status:	Feedback		
Priority:	Normal		
Assignee:			
Category:			
Affected version:	5.4.0	**Resolution:**	

Description

Hi,

I tried to build an active-active setup similar to the one you published on
https://www.strongswan.org/testing/testresults/ha/both-active/

However, my setup includes 2-way traffic meaning that carol for instance is used as gateway and also have a right subnet which gets ipsec traffic from alice's subnet (10.1.0.0/16). I would like that virtual gateway mars will act in high avialability mode both when traffic flows from carol's subnet to alice's sbunet and vice-versa.

I have configured my setup similar to your setup but the failure detection protected only the traffic from carol to virtual gateway mars but it didn't work when traffic flowed from alice's subnet to carol's subnet which is also used as a gateway.

Is is possible to protect such a symetric traffic in which state takeover will occur and will detect traffic failures on both server sides? If yes, what shall be the setup?

Thanks,
Ido

History

#1 - 16.01.2017 16:04 - Tobias Brunner

- Status changed from New to Feedback

> I have configured my setup similar to your setup but the failure detection protected only the traffic from carol to virtual gateway mars but it didn't work when traffic flowed from alice's subnet to carol's subnet which is also used as a gateway.

What does that mean exactly?

Could you please post your config and other relevant information?

#2 - 16.01.2017 17:05 - Ido Ramati

- File HA_SSwan_setup.tif added

I attached my setup.
The setup includes:
1) 2 VMs (VM1, VM2) working as virtual gateway 1 in HA mode. Port 1 on each VM is connected to Layer 2 switch which is connected from there to traffic generator to simulate LAN traffic (clear traffic).
Port 3 on each VM isconnected to another layer 2 switch which simulates WAN traffic to another symetric setup (encrypted traffic).
2. A similar and symetric setup cosists of VM3 & VM4 to build another virtual gateway 2.
3. On virtual gateway 1 I use virtual IP: 10.10.80.10 towards the LAN and virtual IP: 10.10.10.5 towards the WAN (both are defined as cluster IP)
4. On virtual gateway 2 I use virtual IP: 10.10.70.10 towards the LAN and virtual IP: 10.10.10.10 towards the WAN (both are defined as cluster IP)
5. Both virtual gateways working in HA mode between them with port 2 on each VM used as a synchronize port for HA traffic.
6. The traffic genrator runs LAN traffic from 10.10.80.0/24 network on port 2 to network 10.10.70.0/24 on port 1 and also vice versa from port 1 to port 2.

The problems I encountered in this setup are:
1. When the sync traffic runs on separate ports as described (port 2 on the VMs), failover doesn't occur at all. Only when I setup the sync traffic to run over the WAN ports (port3 to port 3 on each VM), failover occur. Is this normal? Why the failover doesn't occur when the sync traffic runs over separate ports?
2. Even if I put the sync traffic over port 3, the failover and reintegration sometimes doesn't happen at all and the traffic simply fails. Does the StrongSwan HA solution supports these kind of setup (gateway to gateway)? If yes, why doesn't it work?

3. When the traffic runs asymmetric, meanaing that on virtual gateway 1 for instance, VM1 is active for traffic running from 10.10.80.0 to 10.10.70.0 and VM2 is active for traffic from 10.10.70.0 to 10.10.80.0. In these cases, when failover occurs, sometimes only 1 VM makes the failover while the other VM simply drop the traffic and faiover doesn't happen at all. Is this normal? What can I do to fix it?

The setup files are:

VM1

ipsec.conf

```
config setup
        strictcrlpolicy=no
        charondebug="ike 4, knl 2, cfg 2, chd 2, dmn 2, lib 2, net 2"

conn %default
        ikelifetime=60m
        keylife=20m
        rekeymargin=3m
        keyingtries=1
        keyexchange=ikev2
        ike=aes256-sha-modp2048!
        esp=aes256-sha1!
        mobike=no
        dpdaction=restart

conn conn_1
        keylife=20m
        right=10.10.10.10
        leftsubnet=10.10.80.0/24
        esp=aes256-sha1-modp4096!
        leftfirewall=yes
        auto=route
        authby=secret
        rightfirewall=no
        rekeymargin=3m
        rightsubnet=10.10.70.0/24
        ike=aes256-sha1-ecp384!
        ikelifetime=60m
        keyingtries=10
        left=10.10.10.5
```

strongswan.conf

```
# strongswan.conf - strongSwan configuration file
#
# Refer to the strongswan.conf(5) manpage for details
#
# Configuration changes should be made in the included files

charon {
        #load = curl aes des sha1 sha2 md5 pem pkcs1 openssl random socket-default nonce x509 revocation hmac
xcbc stroke kernel-netlink socket-raw updown vici
        # number of worker threads in charon
        threads = 16
    # two defined file loggers
        plugins {
                vici {
                        socket = unix:///var/run/charon.vici
                }
    ha {
      local = 10.10.10.1
      remote = 10.10.10.3
      segment_count = 2
      autobalance = 10
      fifo_interface = yes
      monitor = yes
    }
    filelog {
        /var/log/charon.log {
            # add a timestamp prefix
            time_format = %b %e %T
            # prepend connection name, simplifies grepping
            ike_name = yes
            # overwrite existing files
```

```
            append = no
            # increase default loglevel for all daemon subsystems
            default = 2
            # flush each line to disk
            flush_line = yes
        }
        stderr {
            # more detailed loglevel for a specific subsystem, overriding the
            # default loglevel.
            ike = 2
            knl = 3
        }
    }
    # and two loggers using syslog
    syslog {
        # prefix for each log message
        identifier = charon-custom
        # use default settings to log to the LOG_DAEMON facility
        daemon {
        }
        # very minimalistic IKE auditing logs to LOG_AUTHPRIV
        auth {
            default = 2
            ike = 2
        }
        }

        }
}
include strongswan.d/*.conf
```

iptables

```
# Generated by iptables-save v1.4.21 on Wed Dec 21 13:34:59 2016
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -d 10.10.10.5/32 -i eth3 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:30 --tota
l-nodes 2 --local-node 0 --hash-init 0
-A INPUT -d 10.10.80.10/32 -i eth1 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:35 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -p esp -j ACCEPT
-A INPUT -p udp -m udp --dport 500 --sport 500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A INPUT -s 10.10.10.3/32 -d 10.10.10.1/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -s 10.10.10.4/32 -d 10.10.10.1/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -d 224.0.0.1/32 -p igmp -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.91/32 -i eth0 -m policy --dir in --pol ipsec --reqid 2 --proto esp
 -j ACCEPT
-A INPUT -d 10.112.88.91/32 -p tcp -m tcp --dport 5000 -j ACCEPT
-A INPUT -s 147.234.132.156/32 -p tcp -m tcp --sport 8080 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.0/24 -p tcp -m tcp --sport 510:65535 --dport 22 -m state --state N
EW,ESTABLISHED -j ACCEPT
-A INPUT -i eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
-A FORWARD -s 10.10.80.0/24 -d 10.10.70.0/24 -j ACCEPT
-A FORWARD -s 10.10.70.0/24 -d 10.10.80.0/24 -j ACCEPT
-A FORWARD -o eth3 -m policy --dir out --pol ipsec --proto esp -j ACCEPT
-A OUTPUT -s 10.112.88.91/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 2 --proto e
sp -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A OUTPUT -s 10.112.88.91/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 3 --proto e
sp -j ACCEPT
-A OUTPUT -d 10.10.70.0/24 -p esp -j ACCEPT
-A OUTPUT -d 10.10.10.0/24 -p esp -j ACCEPT
-A OUTPUT -s 10.112.88.91/32 -d 147.234.132.156/32 -j ACCEPT
-A OUTPUT -s 10.112.88.91/32 -p tcp -m tcp --sport 5000 -j ACCEPT
-A OUTPUT -d 147.234.132.156/32 -p tcp -m tcp --dport 8080 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 --dport 500 -j ACCEPT
-A OUTPUT -d 10.10.10.3/32 -s 10.10.10.1/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -d 10.10.10.4/32 -s 10.10.10.1/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -s 224.0.0.1/32 -p igmp -j ACCEPT
```

```
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -s 10.112.88.0/24 -d 147.234.132.156/32 -p tcp -m tcp --sport 22 --dport 510:65535 -m state --state
ESTABLISHED -j ACCEPT
-A OUTPUT -o eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
COMMIT
# Completed on Wed Dec 21 13:34:59 2016
```

VM2

ipsec.conf

```
config setup
        strictcrlpolicy=no
        charondebug="ike 4, knl 2, cfg 2, chd 2, dmn 2, lib 2, net 2"

conn %default
        ikelifetime=60m
        keylife=20m
        rekeymargin=3m
        keyingtries=1
        keyexchange=ikev2
        ike=aes256-sha-modp2048!
        esp=aes256-sha1!
        mobike=no
        dpdaction=restart

conn conn_1
        keylife=20m
        right=10.10.10.10
        leftsubnet=10.10.80.0/24
        esp=aes256-sha1-modp4096!
        leftfirewall=yes
        auto=route
        authby=secret
        rightfirewall=no
        rekeymargin=3m
        rightsubnet=10.10.70.0/24
        ike=aes256-sha1-ecp384!
        ikelifetime=60m
        keyingtries=10
        left=10.10.10.5
```

strongswan.conf

```
# strongswan.conf - strongSwan configuration file
#
# Refer to the strongswan.conf(5) manpage for details
#
# Configuration changes should be made in the included files

charon {
        #load = curl aes des sha1 sha2 md5 pem pkcs1 openssl random socket-default nonce x509 revocation hmac
xcbc stroke kernel-netlink socket-raw updown vici
        # number of worker threads in charon
        threads = 16
    # two defined file loggers
        plugins {
                vici {
                        socket = unix:///var/run/charon.vici
                }
    ha {
      local = 10.10.10.3
      remote = 10.10.10.1
      segment_count = 2
      autobalance = 10
      fifo_interface = yes
      monitor = yes
    }
    filelog {
        /var/log/charon.log {
            # add a timestamp prefix
            time_format = %b %e %T
            # prepend connection name, simplifies grepping
            ike_name = yes
```

```
            # overwrite existing files
            append = no
            # increase default loglevel for all daemon subsystems
            default = 2
            # flush each line to disk
            flush_line = yes
        }
        stderr {
            # more detailed loglevel for a specific subsystem, overriding the
            # default loglevel.
            ike = 2
            knl = 3
        }
    }
    # and two loggers using syslog
    syslog {
        # prefix for each log message
        identifier = charon-custom
        # use default settings to log to the LOG_DAEMON facility
        daemon {
        }
        # very minimalistic IKE auditing logs to LOG_AUTHPRIV
        auth {
            default = 2
            ike = 2
        }
        }

        }
}
include strongswan.d/*.conf

iptables

# Generated by iptables-save v1.4.21 on Wed Dec 21 13:34:59 2016
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -d 10.10.10.5/32 -i eth3 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:30 --tota
l-nodes 2 --local-node 0 --hash-init 0
-A INPUT -d 10.10.80.10/32 -i eth1 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:35 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -p esp -j ACCEPT
-A INPUT -p udp -m udp --dport 500 --sport 500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A INPUT -s 10.10.10.1/32 -d 10.10.10.3/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -s 10.10.10.4/32 -d 10.10.10.3/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -d 224.0.0.1/32 -p igmp -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.92/32 -i eth0 -m policy --dir in --pol ipsec --reqid 2 --proto esp
 -j ACCEPT
-A INPUT -d 10.112.88.92/32 -p tcp -m tcp --dport 5000 -j ACCEPT
-A INPUT -s 147.234.132.156/32 -p tcp -m tcp --sport 8080 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.0/24 -p tcp -m tcp --sport 510:65535 --dport 22 -m state --state N
EW,ESTABLISHED -j ACCEPT
-A INPUT -i eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
-A FORWARD -s 10.10.80.0/24 -d 10.10.70.0/24 -j ACCEPT
-A FORWARD -s 10.10.70.0/24 -d 10.10.80.0/24 -j ACCEPT
-A FORWARD -o eth3 -m policy --dir out --pol ipsec --proto esp -j ACCEPT
-A OUTPUT -s 10.112.88.92/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 2 --proto e
sp -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A OUTPUT -s 10.112.88.92/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 3 --proto e
sp -j ACCEPT
-A OUTPUT -d 10.10.70.0/24 -p esp -j ACCEPT
-A OUTPUT -d 10.10.10.0/24 -p esp -j ACCEPT
-A OUTPUT -s 10.112.88.92/32 -d 147.234.132.156/32 -j ACCEPT
-A OUTPUT -s 10.112.88.92/32 -p tcp -m tcp --sport 5000 -j ACCEPT
-A OUTPUT -d 147.234.132.156/32 -p tcp -m tcp --dport 8080 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 --dport 500 -j ACCEPT
-A OUTPUT -d 10.10.10.1/32 -s 10.10.10.3/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -d 10.10.10.4/32 -s 10.10.10.3/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
```

```
-A OUTPUT -s 224.0.0.1/32 -p igmp -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -s 10.112.88.0/24 -d 147.234.132.156/32 -p tcp -m tcp --sport 22 --dport 510:65535 -m state --state
ESTABLISHED -j ACCEPT
-A OUTPUT -o eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
COMMIT
# Completed on Wed Dec 21 13:34:59 2016
```

VM3

ipsec.conf

```
config setup
        strictcrlpolicy=no
        charondebug="ike 4, knl 2, cfg 2, chd 2, dmn 2, lib 2, net 2"

conn %default
        ikelifetime=60m
        keylife=20m
        rekeymargin=3m
        keyingtries=1
        keyexchange=ikev2
        ike=aes256-sha-modp2048!
        esp=aes256-sha1!
        mobike=no
        dpdaction=restart

conn conn_1
        keylife=20m
        right=10.10.10.5
        leftsubnet=10.10.70.0/24
        esp=aes256-sha1-modp4096!
        leftfirewall=no
        auto=route
        authby=secret
        rightfirewall=no
        rekeymargin=3m
        rightsubnet=10.10.80.0/24
        ike=aes256-sha1-ecp384!
        ikelifetime=60m
        keyingtries=10
        left=10.10.10.10
```

strongswan.conf

```
# strongswan.conf - strongSwan configuration file
#
# Refer to the strongswan.conf(5) manpage for details
#
# Configuration changes should be made in the included files

charon {
        #load = curl aes des sha1 sha2 md5 pem pkcs1 openssl random socket-default nonce x509 revocation hmac
xcbc stroke kernel-netlink socket-raw updown vici
        # number of worker threads in charon
        threads = 16
    # two defined file loggers
        plugins {
                vici {
                        socket = unix:///var/run/charon.vici
                }
    ha {
      local = 10.10.10.2
      remote = 10.10.10.4
      segment_count = 2
      autobalance = 10
      fifo_interface = yes
      monitor = yes
    }
    filelog {
        /var/log/charon.log {
            # add a timestamp prefix
            time_format = %b %e %T
            # prepend connection name, simplifies grepping
```

```
            ike_name = yes
            # overwrite existing files
            append = no
            # increase default loglevel for all daemon subsystems
            default = 2
            # flush each line to disk
            flush_line = yes
        }
        stderr {
            # more detailed loglevel for a specific subsystem, overriding the
            # default loglevel.
            ike = 2
            knl = 3
        }
    }
    # and two loggers using syslog
    syslog {
        # prefix for each log message
        identifier = charon-custom
        # use default settings to log to the LOG_DAEMON facility
        daemon {
        }
        # very minimalistic IKE auditing logs to LOG_AUTHPRIV
        auth {
            default = 2
            ike = 2
        }
        }

        }
}
include strongswan.d/*.conf

iptables

# Completed on Thu Oct 20 09:21:53 2016
# Generated by iptables-save v1.4.21 on Thu Oct 20 09:21:53 2016
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -d 10.10.10.10/32 -i eth3 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:20 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -d 10.10.70.10/32 -i eth1 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:25 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -p esp -j ACCEPT
-A INPUT -p udp -m udp --dport 500 --sport 500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A INPUT -s 10.10.10.4/32 -d 10.10.10.2/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -s 10.10.10.1/32 -d 10.10.10.2/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -d 224.0.0.1/32 -p igmp -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.90/32 -i eth0 -m policy --dir in --pol ipsec --reqid 2 --proto esp
 -j ACCEPT
-A INPUT -d 10.112.88.90/32 -p tcp -m tcp --dport 5000 -j ACCEPT
-A INPUT -s 147.234.132.156/32 -p tcp -m tcp --sport 8080 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.0/24 -p tcp -m tcp --sport 510:65535 --dport 22 -m state --state N
EW,ESTABLISHED -j ACCEPT
-A INPUT -i eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
-A FORWARD -s 10.10.80.0/24 -d 10.10.70.0/24 -j ACCEPT
-A FORWARD -s 10.10.70.0/24 -d 10.10.80.0/24 -j ACCEPT
-A FORWARD -o eth3 -m policy --dir out --pol ipsec --proto esp -j ACCEPT
-A OUTPUT -s 10.112.88.90/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 2 --proto e
sp -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A OUTPUT -s 10.112.88.90/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 3 --proto e
sp -j ACCEPT
-A OUTPUT -d 10.10.70.0/24 -p esp -j ACCEPT
-A OUTPUT -d 10.10.10.0/24 -p esp -j ACCEPT
-A OUTPUT -s 10.112.88.90/32 -d 147.234.132.156/32 -j ACCEPT
-A OUTPUT -s 10.112.88.90/32 -p tcp -m tcp --sport 5000 -j ACCEPT
-A OUTPUT -d 147.234.132.156/32 -p tcp -m tcp --dport 8080 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 --dport 500 -j ACCEPT
```

```
-A OUTPUT -d 10.10.10.4/32 -s 10.10.10.2/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -d 10.10.10.1/32 -s 10.10.10.2/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -s 224.0.0.1/32 -p igmp -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -s 10.112.88.0/24 -d 147.234.132.156/32 -p tcp -m tcp --sport 22 --dport 510:65535 -m state --state
ESTABLISHED -j ACCEPT
-A OUTPUT -o eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
COMMIT
```

VM4

```
ipsec.conf

config setup
        strictcrlpolicy=no
        charondebug="ike 4, knl 2, cfg 2, chd 2, dmn 2, lib 2, net 2"

conn %default
        ikelifetime=60m
        keylife=20m
        rekeymargin=3m
        keyingtries=1
        keyexchange=ikev2
        ike=aes256-sha-modp2048!
        esp=aes256-sha1!
        mobike=no
        dpdaction=restart

conn conn_1
        keylife=20m
        right=10.10.10.5
        leftsubnet=10.10.70.0/24
        esp=aes256-sha1-modp4096!
        leftfirewall=no
        auto=route
        authby=secret
        rightfirewall=no
        rekeymargin=3m
        rightsubnet=10.10.80.0/24
        ike=aes256-sha1-ecp384!
        ikelifetime=60m
        keyingtries=10
        left=10.10.10.10

strongswan.conf

# strongswan.conf - strongSwan configuration file
#
# Refer to the strongswan.conf(5) manpage for details
#
# Configuration changes should be made in the included files

charon {
        #load = curl aes des sha1 sha2 md5 pem pkcs1 openssl random socket-default nonce x509 revocation hmac
xcbc stroke kernel-netlink socket-raw updown vici
        # number of worker threads in charon
        threads = 16
    # two defined file loggers
        plugins {
                vici {
                        socket = unix:///var/run/charon.vici
                }
    ha {
      local = 10.10.10.4
      remote = 10.10.10.2
      segment_count = 2
      autobalance = 10
      fifo_interface = yes
      monitor = yes
    }
    filelog {
        /var/log/charon.log {
            # add a timestamp prefix
            time_format = %b %e %T
```

```
            # prepend connection name, simplifies grepping
            ike_name = yes
            # overwrite existing files
            append = no
            # increase default loglevel for all daemon subsystems
            default = 2
            # flush each line to disk
            flush_line = yes
        }
        stderr {
            # more detailed loglevel for a specific subsystem, overriding the
            # default loglevel.
            ike = 2
            knl = 3
        }
    }
    # and two loggers using syslog
    syslog {
        # prefix for each log message
        identifier = charon-custom
        # use default settings to log to the LOG_DAEMON facility
        daemon {
        }
        # very minimalistic IKE auditing logs to LOG_AUTHPRIV
        auth {
            default = 2
            ike = 2
        }
        }

        }
}
include strongswan.d/*.conf

iptables

# Generated by iptables-save v1.4.21 on Wed Jan  4 16:44:59 2017
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -d 10.10.10.10/32 -i eth3 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:20 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -d 10.10.70.10/32 -i eth1 -j CLUSTERIP --new --hashmode sourceip --clustermac 01:00:5e:00:00:25 --tot
al-nodes 2 --local-node 0 --hash-init 0
-A INPUT -p esp -j ACCEPT
-A INPUT -p udp -m udp --dport 500 --sport 500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A INPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A INPUT -s 10.10.10.2/32 -d 10.10.10.4/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -s 10.10.10.3/32 -d 10.10.10.4/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A INPUT -d 224.0.0.1/32 -p igmp -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.93/32 -i eth0 -m policy --dir in --pol ipsec --reqid 2 --proto esp
 -j ACCEPT
-A INPUT -d 10.112.88.93/32 -p tcp -m tcp --dport 5000 -j ACCEPT
-A INPUT -s 147.234.132.156/32 -p tcp -m tcp --sport 8080 -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 147.234.132.156/32 -d 10.112.88.0/24 -p tcp -m tcp --sport 510:65535 --dport 22 -m state --state N
EW,ESTABLISHED -j ACCEPT
-A INPUT -i eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
-A FORWARD -s 10.10.80.0/24 -d 10.10.70.0/24 -j ACCEPT
-A FORWARD -s 10.10.70.0/24 -d 10.10.80.0/24 -j ACCEPT
-A FORWARD -o eth3 -m policy --dir out --pol ipsec --proto esp -j ACCEPT
-A OUTPUT -s 10.112.88.93/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 2 --proto e
sp -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4510 --dport 4510 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 4500 --dport 4500 -j ACCEPT
-A OUTPUT -s 10.112.88.93/32 -d 147.234.132.156/32 -o eth0 -m policy --dir out --pol ipsec --reqid 3 --proto e
sp -j ACCEPT
-A OUTPUT -d 10.10.70.0/24 -p esp -j ACCEPT
-A OUTPUT -d 10.10.10.0/24 -p esp -j ACCEPT
-A OUTPUT -s 10.112.88.93/32 -d 147.234.132.156/32 -j ACCEPT
-A OUTPUT -s 10.112.88.93/32 -p tcp -m tcp --sport 5000 -j ACCEPT
-A OUTPUT -d 147.234.132.156/32 -p tcp -m tcp --dport 8080 -j ACCEPT
-A OUTPUT -p udp -m udp --sport 500 --dport 500 -j ACCEPT
```

```
-A OUTPUT -d 10.10.10.2/32 -s 10.10.10.4/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -d 10.10.10.3/32 -s 10.10.10.4/32 -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A OUTPUT -s 224.0.0.1/32 -p igmp -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -s 10.112.88.0/24 -d 147.234.132.156/32 -p tcp -m tcp --sport 22 --dport 510:65535 -m state --state
ESTABLISHED -j ACCEPT
-A OUTPUT -o eth3 -d 224.0.0.0/8 -p vrrp -j ACCEPT
COMMIT
```

Thanks,
Ido

#3 - 16.01.2017 17:55 - Tobias Brunner

> 1. When the sync traffic runs on separate ports as described (port 2 on the VMs), failover doesn't occur at all. Only when I setup the sync traffic to run over the WAN ports (port3 to port 3 on each VM), failover occur. Is this normal? Why the failover doesn't occur when the sync traffic runs over separate ports?

Debug it. Check the logs (look for messages with "heartbeat" or "segment" in them). Check the network traffic on these interfaces (tcpdump/Wireshark).

> 2. Even if I put the sync traffic over port 3, the failover and reintegration sometimes doesn't happen at all and the traffic simply fails. Does the StrongSwan HA solution supports these kind of setup (gateway to gateway)? If yes, why doesn't it work?

strongSwan doesn't care how the other peer is setup. It simply knows the other peer's IP and whether that's virtual or physical doesn't really matter. But there could obviously be issues with firewalls, ARP, switches etc. Check the logs for details on what's going on when reintegrating a host.

> 3. When the traffic runs asymmetric, meanaing that on virtual gateway 1 for instance, VM1 is active for traffic running from 10.10.80.0 to 10.10.70.0 and VM2 is active for traffic from 10.10.70.0 to 10.10.80.0. In these cases, when failover occurs, sometimes only 1 VM makes the failover while the other VM simply drop the traffic and faiover doesn't happen at all. Is this normal? What can I do to fix it?

Not sure what you mean. When a failover occurs all segments should be handled by the remaining HA host (check the log for details).

#4 - 12.02.2017 13:07 - Ido Ramati

- File charon_long_re-integration.log added

- File charon_no_failover.log added

In order to minimize environment problems I set the same setup over bare-metal environment using Ubuntu 16.10 with kernel 4.8.11 patched with your latest HA patch.
However, I still encounter similar problems:

When the sync traffic runs on dedicated ports, failover still doesn't occur.

> Debug it. Check the logs (look for messages with "heartbeat" or "segment" in them). Check the network traffic on these interfaces (tcpdump/Wireshark).

1) Looking at the logs, I see that IKE is going into loop of retransmissions although the ipsec link is down (due to failover). The sync link is still up and the segments remain the same as before the failover.

```
Feb 12 13:58:02 06[IKE] <conn_1|1> retransmit 1 of request with message ID 2
Feb 12 13:58:02 06[NET] <conn_1|1> sending packet: from 10.10.10.10[500] to 10.10.10.5[500] (76 bytes)
Feb 12 13:58:02 06[MGR] <conn_1|1> checkin IKE_SA conn_1[1]
```

Why does the IKE go into retransmission loop instead of changing the segment and start failover? Only a fail of the sync channel is used as a trigger for the failover process to start?

2) Even when I configure the sync channel over the IPsec channel (in-band), the failover happen very quickly (less than 2 seconds), but the re-integration occur after very long time: can be between 40 - 150 seconds. Log file is attached.
From the log file:

IPsec link failover (by putting ethernet link down):

```
Feb 12 13:12:51 16[IKE] <conn_1|4> retransmit 1 of request with message ID 0
Feb 12 13:12:51 16[NET] <conn_1|4> sending packet: from 10.10.10.5[500] to 10.10.10.10[500] (76 bytes)
```

Ethernet link raise up again after ~14 seconds but IPsec link remains down:

```
Feb 12 13:13:05 05[CFG] HA segment 1 was handled twice, dropping
Feb 12 13:13:05 05[CFG] HA segment 1 deactivated, now active: 2
Feb 12 13:13:05 05[CFG] HA segment 2 was handled twice, taking
```

IPsec link raise up only after 2.5 minutes(!):

```
Feb 12 13:15:32 04[CFG] <conn_1|10> handling HA CHILD_SA conn_1{3} 10.10.80.0/24 === 10.10.70.0/24 (segment in
: 2*, out: 1)
Feb 12 13:15:32 15[CFG] <conn_1|10> handling HA CHILD_SA conn_1{5} 10.10.80.0/24 === 10.10.70.0/24 (segment in
: 1, out: 1)
```

I checked all network elements and there are no network problems (the setup runs on an internal lab)
What can be the reason for the reintegration process to take so long?

#5 - 15.02.2017 11:56 - Tobias Brunner

Looks like the current HA solution isn't really designed to be used on initiators, only responders. So I guess what you want to do is currently not possible.

#6 - 15.02.2017 13:03 - Ido Ramati

Thanks, Tobias.

Are there any plans / roadmap to support HA based on intiators?

Thanks,
Ido

#7 - 15.02.2017 14:20 - Tobias Brunner

> Are there any plans / roadmap to support HA based on intiators?

No, currently not.

Files

HA_SSwan_setup.tif	2.13 MB	16.01.2017	Ido Ramati
charon_no_failover.log	63.1 KB	12.02.2017	Ido Ramati
charon_long_re-integration.log	368 KB	12.02.2017	Ido Ramati