Securing Cisco Group Encrypted Transport VPN Group Members Using the Fail-Close Feature

Introduction

The fail-close feature in Cisco® Group Encrypted Transport VPN on Cisco IOS® Software Release 12.4(22)T provides additional security by keeping unregistered Group Encrypted Transport VPN group member routers from forwarding unencrypted data packets. It sets up an implicit **permit ip any any** statement at the end of the security policy during the pre-registration phase. After the group member successfully registers with the Group Encrypted Transport VPN key server, the **permit ip any any** statement is removed from the security policy on the group member.

You can specify exceptions to this rule for the traffic that needs to be forwarded in the clear through a deny entry in the access control list (ACL), which allows routing and management packets from a particular host to flow in the clear.

After the group member is successfully registered for all its groups, the policies downloaded from the key server take over, governing the group member's behavior, and the fail-close ACL and implicit **permit ip any any** command are removed. Group members keep the policies downloaded from the key server even if the **re**-registration fails and the IP Security (IPsec) security associations expire.

It is imperative that you review the exact policy applied in the fail-close ACL before "activating" the policy. You should be able to access the group member at all times—even when the fail-close policy is activated. If you are not careful when designing the fail-close ACL, you could lock yourself out of the device.

Deploying the Fail-Close Feature

When the fail-close feature is activated, all unencrypted traffic that flows to, from, and through this device is prevented prior to and during registration. As mentioned previously, after the group member is successfully registered to all its groups, the policies downloaded from the key server take over, governing the group member's behavior, and the fail-close ACL and implicit **permit ip any any** command are removed from the group member's security policy. Group members keep the policies downloaded from the key server even if the re-registration fails and the IPsec security associations expire.

Note: In prior releases of IOS that do not support the fail-close feature, it is possible to implement similar functionality using an interface ACL. Fail-close functionality using an interface ACL might still be useful to customers looking to enforce a policy where certain packets must always be encrypted, regardless of the downloaded key server policy.

When the fail-close feature is configured and activated on the group member, the Group Encrypted Transport VPN traffic flow rules are as follows:

- Before the group member registers with the key server or during the registration process, all unencrypted traffic is dropped unless an explicit policy is configured on the group member to allow certain traffic in the clear.

- If the group member successfully completes the registration process, the group member applies the group policy that is downloaded from the key server in conjunction with the local group-member policy if it exists.
- If the group member does **not** successfully complete the registration process, the fail-close policy drops all traffic unless there is an explicit **deny** statement configured in the fail-close ACL. Any **deny** statements in the local group-member policy will not take effect. This local **deny** policy must be duplicated in the fail-close policy because of the deny-jump action (see more on the **deny-jump** action described later in this document).
- Fail-close mode is valid only for the first-time registration attempt or the registration attempt caused by the execution of the **clear crypto gdoi** command on the group member.
- After a successful registration, if the group member fails to register in subsequent registrations, the fail-close policy does not apply. The GM group member still applies the IPsec policy that was downloaded from the key server until it expires. After the policy expires, all traffic is dropped unless there is a specific local policy on the group member for the traffic.
- An explicit **access-list xxx deny udp any eq 848 any eq 848** in the fail-close ACL is not required unless the key server is behind another group member.
- If a group member router is participating in multiple Group Encrypted Transport VPN groups, the fail-close policy is active until all the groups have successfully registered to the key server.

These rules are discussed in depth as follows.

Note: **Permit** implies encrypted traffic; nonencrypted traffic is dropped if there is no IPsec security association for this traffic. The **deny** statement implies traffic will be allowed to traverse in the clear.

The fail-close command follows:

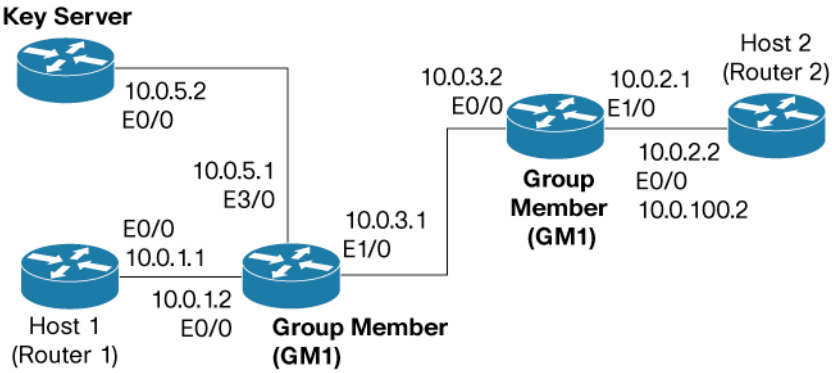
```
crypto map <map name> gdoi fail-close
   match address <ACL number>
   activate
```

If the fail-close crypto map is configured and activated on the group member, an implicit **permit ip any any** statement is inserted at the end of the group member's policy. Any and all traffic to, from, and through this device is blocked.

If you want to allow certain traffic to traverse in the clear, you must configure an explicit match address ACL under the fail-close GDOI crypto map, as shown in the fail-close command configuration.

A **deny** statement configured under the fail-close crypto map ACL is different from a **deny** statement that is locally configured on the group member ACL. The local group-member ACL will always exist and be applied after successful registration of the group member, In constrast, the fail-close ACL is removed after the group member successfully registers to the key server.

Figure 1. Group Encrypted Transport VPN Fail-Close Feature Discussion Topology



Note: The configurations provided in the following examples are not recommended configurations. They are shown here only to illustrate the behavior of the **permit** and **deny** statements in the fail-close ACL.

In addition, the fail-close policy is usually a subset of the group policy that is defined on the key server for the group. The idea here is that the traffic that needs to flow in the clear before registration would most likely be the same after registration. Any local group-member policy that is defined on a particular group member is usually applicable for that particular group member and not for the entire group.

Prior to Group Member Registration or First-Time Group Member Registration Failure

The interface on the key server is in shutdown state to show what happens to the traffic on GM1 and GM2 with fail-close configured on GM1. On GM1, the fail-close crypto map is configured with the keyword “activate”, which causes all traffic to be dropped:

```
crypto map diffint gdoi fail-close
  match address 133
  activate
```

Note: You should execute the command **sh crypto map gdoi fail-close diffint** to check the fail-close ACL before applying the **activate** command to prevent being locked out of the device because of incorrect or incomplete configurations.

```
GM1#sh crypto map gdoi fail-close diffint
Crypto Map: "diffint"
        Activate: yes
        Fail-Close Access-List: (Deny = Forward In Clear, Permit = Drop)
            access-list 133 deny eigrp any any
            access-list 133 permit ip host 10.0.1.1 host 10.0.2.2
            access-list 133 deny ip host 10.0.1.1 host 10.0.2.1
            access-list 133 deny ip host 10.0.1.1 host 10.0.3.2
            access-list 133 deny udp any port = 848 any port = 848
```

The fail-close ACL is also visible through the **show crypto ruleset** command:

```
GM1#sh crypto ruleset
Ethernet1/0:
 IP 10.0.3.1 10.0.3.2 IPSec SA
 IP 10.0.3.1 10.0.3.2 IPSec Cryptomap
  58 ANY ANY Go in clear
 IP 10.0.1.1 10.0.2.2 IPSec Fail Close
 IP 10.0.1.1 10.0.2.1 Go in clear
 IP 10.0.1.1 10.0.3.2 Go in clear
  11 ANY/848 ANY/848 Go in clear
 IP ANY ANY DENY
```

Following are a few explicit **deny** statements for access list 133; only traffic matching these deny commands is allowed to pass in the clear:

```
access-list 133 deny   eigrp any any
access-list 133 permit ip host 10.0.1.1 host 10.0.2.2
access-list 133 deny   ip host 10.0.1.1 host 10.0.2.1
access-list 133 deny   ip host 10.0.1.1 host 10.0.3.2
access-list 133 deny   udp any eq 848 any eq 848
```

For example, the command **access-list 133 deny eigrp any any** allows **eigrp**; you can verify this by executing a **show ip route** command on GM1:

```
GM1#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static
route
       o - ODR, P - periodic downloaded static route
```

The gateway of last resort is not set:

```
     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
D       10.0.2.0/24 [90/307200] via 10.0.3.2, 00:00:13, Ethernet1/0
C       10.0.3.0/24 is directly connected, Ethernet1/0
C       10.0.1.0/24 is directly connected, Ethernet0/0
C       10.0.4.0/24 is directly connected, Ethernet2/0
C       10.0.5.0/24 is directly connected, Ethernet3/0
D       10.65.9.2/32 [90/409600] via 10.0.3.2, 00:00:13, Ethernet1/0
D       10.0.100.0/24 [90/307200] via 10.0.3.2, 00:00:13, Ethernet1/0
```

A ping from host 1 at 10.0.1.1 to 10.0.2.1 on GM2 should go through because the **access list command access-list 133 deny ip host 10.0.1.1 host 10.0.2.1** is in ACL 133 on GM1:

```
host1#ping 10.0.2.1 source 10.0.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.2.1, timeout is 2 seconds:

Packet sent with a source address of 10.0.1.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1012 ms

A ping from host1 10.0.1.1 to host2 10.0.2.2 should fail because we have the statement **access-list 133 permit ip host 10.0.1.1 host 10.0.2.2** in ACL 133 on GM1:

```
host1#ping 10.0.2.2 source 10.0.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:

Packet sent with a source address of 10.0.1.1

.....

Success rate is 0 percent (0/5)

GM1 expects these packets to be encrypted because of the permit statement in the policy. However, because these packets are arriving in the clear (unencrypted), they will be dropped. You can verify this drop on GM1 by looking at the access-list matches:

```
GM1#sh ip access-lists
Extended IP access list 120
    10 permit ip host 10.0.3.1 host 10.0.3.2
Extended IP access list 133
    10 deny eigrp any any (12 matches)
    20 permit ip host 10.0.1.1 host 10.0.2.2 (5 matches) <-----access list matches
    30 deny ip host 10.0.1.1 host 10.0.2.1
    40 deny ip host 10.0.1.1 host 10.0.3.2
    50 deny udp any eq 848 any eq 848
```

Similarly, a ping from host 1 10.0.1.1 to host 2 10.0.100.2 should fail because there is no explicit deny statement on GM1:

```
host1#ping 10.0.100.2 source 10.0.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.100.2, timeout is 2 seconds:

Packet sent with a source address of 10.0.1.1

.....

Success rate is 0 percent (0/5)

A ping from GM1 10.0.3.1 to GM2 10.0.3.2 should be encrypted because GM1 uses crypto map diffint 10 ipsec-isakmp to set up a point-to-point IPsec tunnel with peer 10.0.3.2, which is on GM2. This tunnel is not a Group Encrypted Transport VPN tunnel, although it is in the same map set:

```
crypto map diffint 10 ipsec-isakmp
 set peer 10.0.3.2
 set transform-set man
 match address 120
```

Checking the Internet Key Exchange (IKE) security-association status for this IKE/IPsec tunnel shows the following:

```
GM1#sh crypto isa sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
10.0.3.1        10.0.3.2        QM_IDLE           1001 ACTIVE
10.0.3.2        10.0.3.1        QM_IDLE           1002 ACTIVE

GM1#sh crypto ipsec sa
     PFS (Y/N): N, DH group: none

interface: Ethernet1/0
    Crypto map tag: diffint, local addr 10.0.3.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.3.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.3.2/255.255.255.255/0/0)
   current_peer 10.0.3.2 port 500 <----------------IKE Port
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 6, #pkts encrypt: 6, #pkts digest: 6
    #pkts decaps: 5, #pkts decrypt: 5, #pkts verify: 5
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 7, #recv errors 0

     local crypto endpt.: 10.0.3.1, remote crypto endpt.: 10.0.3.2
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet1/0
```

```
     current outbound spi: 0x54D712FE(1423381246)

GM1#ping 10.0.3.2 source 10.0.3.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.3.2, timeout is 2 seconds:

Packet sent with a source address of 10.0.3.1

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/20 ms

```
GM1#sh ip access-lists
Extended IP access list 120
    10 permit ip host 10.0.3.1 host 10.0.3.2 (10 matches)
```

The statement **access-list 133 deny udp any eq 848 any eq 848** in the fail-close ACL is there for a reason. In this topology, the key server is behind GM1. If GM1 fails to register to the key server for some reason and fail-close is activated on GM1, then GM2 will be unable to register to the key server because GM1 will drop all packets in the absence of access-list 133.

After Group Encrypted Transport VPN Registration

Now, execute the **no shut** command on the interface e0/0 on the key server and allow the group members, GM1 and GM2, to register to the key server. (Executing the command **clear crypto gdoi group diffint** on the group members forces them to start to register right away.)

With the registration of GM1 for group “diffint”, the following is the policy downloaded from the key server:

```
GM1#sh crypto gdoi
GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 3333
    Rekeys received          : 0
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.2
    Group Server list        : 10.0.5.2
<snip……>

 ACL Downloaded From KS 10.0.5.2:
   access-list  permit ip host 10.0.1.1 host 10.0.2.2
   access-list  permit ip host 10.0.2.2 host 10.0.1.1
```

```
GM2#sh crypto gdoi
GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 3333
    Rekeys received          : 5
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.2
    Group Server list        : 10.0.5.2
  <snip……..>

 ACL Downloaded From KS 10.0.5.2:
   access-list  permit ip host 10.0.1.1 host 10.0.2.2
   access-list  permit ip host 10.0.2.2 host 10.0.1.1
```

The command show crypto ruleset shows the active rules:

```
GM1#sh crypto ruleset
Ethernet1/0:
 IP 10.0.1.1 10.0.2.2 IPSec SA
 IP 10.0.1.1 10.0.2.2 IPSec Cryptomap
 IP 10.0.2.2 10.0.1.1 IPSec SA
 IP 10.0.2.2 10.0.1.1 IPSec Cryptomap
 IP 10.0.3.1 10.0.3.2 IPSec Cryptomap
```

Following are the IPsec security associations that are installed on GM1:

```
GM1#sh crypto ipsec sa
 <snip…….>

interface: Ethernet1/0
    Crypto map tag: diffint, local addr 10.0.3.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.1.1/255.255.255.255/0/0)

current_peer 0.0.0.0 port 848 <-------------GDOI crypto map
     PERMIT, flags={origin_is_acl,}

<snip…….>
```

```
     local crypto endpt.: 10.0.3.1, remote crypto endpt.: 0.0.0.0
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet1/0
     current outbound spi: 0xC96F3E79(3379510905)

<snip…….>
protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   current_peer 0.0.0.0 port 848 <----------GDOI crypto map
     PERMIT, flags={origin_is_acl,}
<snip…….>

     local crypto endpt.: 10.0.3.1, remote crypto endpt.: 0.0.0.0
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet1/0
     current outbound spi: 0xC96F3E79(3379510905)
```

The IPsec security associations installed on GM2 follow:

```
GM2#sh crypto ipsec sa
     PFS (Y/N): N, DH group: none
     PFS (Y/N): N, DH group: none
     PFS (Y/N): N, DH group: none

interface: Ethernet0/0
    Crypto map tag: diffint, local addr 10.0.3.2

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.1.1/255.255.255.255/0/0)
   current_peer 0.0.0.0 port 848 <--------------GDOI crypto map
     PERMIT, flags={origin_is_acl,}
<snip……>

     local crypto endpt.: 10.0.3.2, remote crypto endpt.: 0.0.0.0
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet0/0
     current outbound spi: 0x57D2FF13(1473445651)
```

A ping from host1 10.0.1.1 to host2 10.0.2.2 should go through with the encrypted packet. The command **show crypto session detail** will show the number of packets encrypted and decrypted, as shown here on the group members:

```
GM1#sh crypto sess detail
Crypto session current status
<snip…….>

Interface: Ethernet1/0
Uptime: 02:49:28
Session status: UP-ACTIVE
Peer: 0.0.0.0 port 848 fvrf: (none) ivrf: (none)
      Phase1_id: 10.0.5.2
      Desc: (none)
  IKE SA: local 10.0.3.1/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1006 lifetime:7w0d
  IKE SA: local 10.0.3.1/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1003 lifetime:21:10:31
  IKE SA: local 10.0.3.1/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1005 lifetime:7w0d
  IPSEC FLOW: permit ip host 10.0.2.2 host 10.0.1.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/3076
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/3076
  IPSEC FLOW: permit ip host 10.0.1.1 host 10.0.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 15 drop 0 life (KB/Sec) 0/3076 <--------
        Packets encrypted and decrypted
        Outbound: #pkts enc'ed 15 drop 0 life (KB/Sec) 0/3076 <--------
        Packets encrypted and decrypted

GM2#sh crypto sess detail
Crypto session current status

<Snip…….>

Interface: Ethernet0/0
Uptime: 02:47:36
Session status: UP-ACTIVE
```

```
Peer: 0.0.0.0 port 848 fvrf: (none) ivrf: (none)
      Phase1_id: 10.0.5.2
      Desc: (none)
  IKE SA: local 10.0.3.2/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1006 lifetime:7w0d
  IKE SA: local 10.0.3.2/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1003 lifetime:21:12:23
  IKE SA: local 10.0.3.2/848 remote 10.0.5.2/848 Active
          Capabilities:(none) connid:1005 lifetime:7w0d
  IPSEC FLOW: permit ip host 10.0.2.2 host 10.0.1.1
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 15 drop 0 life (KB/Sec) 0/3181 <-------
        Packets encrypted and decrypted
        Outbound: #pkts enc'ed 15 drop 0 life (KB/Sec) 0/3181 <-------
        Packets encrypted and decrypted
  IPSEC FLOW: permit ip host 10.0.1.1 host 10.0.2.2
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/3181
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/3181
```

Subsequent Registration Failures After One Successful Registration

After one such successful registration, shut down the key server interface. If the IPsec security associations expire on both group members, the fail-close ACL should not be active. In this case, the downloaded key server policy and the local group-member policy take effect:

```
GM1#sh crypto gdoi
GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 3333
    Rekeys received          : 0
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.2
    Group Server list        : 10.0.5.2

    GM Reregisters in        : 226 secs
    Rekey Received           : never

    Rekeys received
         Cumulative          : 0
```

```
        After registration  : 0
    Rekey Acks sent         : 0

 ACL Downloaded From KS 10.0.5.2:
   access-list  permit ip host 10.0.1.1 host 10.0.2.2
   access-list  permit ip host 10.0.2.2 host 10.0.1.1
```

To show that that the fail-close ACL is no longer valid, allow the SPIs (Security Parameter Index) to expire :

```
GM1#sh crypto ipsec sa | incl spi
     current outbound spi: 0x0(0)
     current outbound spi: 0x0(0)
     current outbound spi: 0x0(0)

GM1#sh crypto ipsec sa
   ...……..
   ...…...…...……..

   interface: Ethernet1/0
    Crypto map tag: diffint, local addr 10.0.3.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.1.1/255.255.255.255/0/0)
   current_peer 0.0.0.0 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0

     local crypto endpt.: 10.0.3.1, remote crypto endpt.: 0.0.0.0
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet1/0
     current outbound spi: 0x0(0)
inbound esp sas:
```

```
     inbound ah sas:

     inbound pcp sas:
outbound esp sas:

     outbound ah sas:

     outbound pcp sas:

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.1/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (10.0.2.2/255.255.255.255/0/0)
   current_peer 0.0.0.0 port 848
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 15, #pkts encrypt: 15, #pkts digest: 15
    #pkts decaps: 15, #pkts decrypt: 15, #pkts verify: 15
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 25, #recv errors 0
local crypto endpt.: 10.0.3.1, remote crypto endpt.: 0.0.0.0
     path mtu 1000, ip mtu 1000, ip mtu idb Ethernet1/0
     current outbound spi: 0x0(0)

     inbound esp sas:

     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:

     outbound ah sas:

     outbound pcp sas:
```

Ping from host 1 to host 2:

```
host1#ping 10.0.2.2 source 10.0.1.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 10.0.2.2, timeout is 2 seconds:

Packet sent with a source address of 10.0.1.1 .....

Success rate is 0 percent (0/5)

The command **show crypto session detail** shows the packet drops:

```
GM1#sh crypto sess detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Ethernet1/0
Session status: DOWN
Peer: 0.0.0.0 port 500 fvrf: (none) ivrf: (none)
      Desc: (none)
      Phase1_id: (none)
  IPSEC FLOW: permit ip host 10.0.2.2 host 10.0.1.1
        Active SAs: 0, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 0/0
  IPSEC FLOW: permit ip host 10.0.1.1 host 10.0.2.2
        Active SAs: 0, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 0/0
        Outbound: #pkts enc'ed 0 drop 5 life (KB/Sec) 0/0 <-------Packet
                                            drops due to expired SPIs
```

The following configuration implies that the fail-close ACL is no longer active after one successful registration, even though the IPsec security associations have expired:

```
GM1#sh ip access-lists
Extended IP access list 120
    10 permit ip host 10.0.3.1 host 10.0.3.2 (29 matches)
Extended IP access list 133
    10 deny eigrp any any (157 matches)
```

```
    20 permit ip host 10.0.1.1 host 10.0.2.2 <--(no matches in the fail-close policy)
    30 deny ip host 10.0.1.1 host 10.0.2.1
    40 deny ip host 10.0.1.1 host 10.0.3.2
    50 deny udp any eq 848 any eq 848 (18 matches)
```

```
GM1#sh crypto ruleset
Ethernet1/0:
 IP 10.0.1.1 10.0.2.2 IPSec Cryptomap
 IP 10.0.2.2 10.0.1.1 IPSec Cryptomap
 IP 10.0.3.1 10.0.3.2 IPSec SA
 IP 10.0.3.1 10.0.3.2 IPSec Cryptomap
```

Fail-Close Behavior with Two Groups

If two or more GDOI groups are configured on the group member, the fail-close ACL will still be active until all GDOI groups successfully complete the registration process to the respective key server for that group. In the following example, two groups—diffint and test—are configured on the group member. The group diffint has registered to key server 99, but the group test has not been able to complete the registration process:

Group member, GM1 has registered for group diffint:

```
GM1#sh crypto gdoi
GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 3333
    Rekeys received          : 0
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.2
    Group Server list        : 10.0.5.2

    GM Reregisters in        : 2958 secs
    Rekey Received           : never

    Rekeys received
         Cumulative          : 0
         After registration  : 0
    Rekey Acks sent          : 0
```

ACL Downloaded From KS 10.0.5.2:

```
   access-list  permit ip host 10.0.1.1 host 10.0.2.2
   access-list  permit ip host 10.0.2.2 host 10.0.1.1
……..
……..
```

Group member, GM1 has not completed registration for group test:

```
    Group Name               : test
    Group Identity           : 4444
    Rekeys received          : 0
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.3
    Group Server list        : 10.0.5.3

    GM Reregisters in        : 0 secs
    Rekey Received           : never


    Rekeys received
         Cumulative          : 0
         After registration  : 0
```

ACL Downloaded From KS 10.0.5.3:

```
TEK POLICY:
  Ethernet1/0:

GM1#
00:09:05: %GDOI-4-GM_RE_REGISTER: The IPSec SA created for group test may
have expired/been cleared, or didn't go through. Re-register to KS.
enc#
00:09:05: %CRYPTO-5-GM_REGSTER: Start registration to KS 10.0.5.3 for
group test using address 10.0.3.1
enc#
```

Executing the **show crypto ruleset** command on the group member (GM1) shows that the fail-close ACL is still being used:

```
GM1#sh crypto ruleset
```

```
Ethernet1/0:
 IP 10.0.1.1 10.0.2.2 IPSec SA
 IP 10.0.1.1 10.0.2.2 IPSec Cryptomap
 IP 10.0.2.2 10.0.1.1 IPSec SA
 IP 10.0.2.2 10.0.1.1 IPSec Cryptomap
 IP 10.0.3.1 10.0.3.2 IPSec SA
 IP 10.0.3.1 10.0.3.2 IPSec Cryptomap
  58 ANY ANY Go in clear
 IP 10.0.1.1 10.0.2.2 IPSec Fail Close
 IP 10.0.1.1 10.0.2.1 Go in clear
 IP 10.0.1.1 10.0.3.2 Go in clear
  11 ANY/848 ANY/848 Go in clear
 IP ANY ANY DENY
```

When the group member GM1 completes the registration for group test, the fail-close ACL is removed:

```
GM1#sh crypto gdoi group test
    Group Name               : test
    Group Identity           : 4444
    Rekeys received          : 0
    IPSec SA Direction       : Both
    Active Group Server      : 10.0.5.3
    Group Server list        : 10.0.5.3

    GM Reregisters in        : 3408 secs
    Rekey Received           : never

    Rekeys received
         Cumulative          : 0
         After registration  : 0
    Rekey Acks sent          : 0
```

ACL Downloaded From KS 10.0.5.3:

```
   access-list  permit ip host 10.0.1.1 host 10.0.100.2
   access-list  permit ip host 10.0.100.2 host 10.0.1.1
……..<snip>……
```

```
GM1#sh crypto ruleset
Ethernet1/0:
 IP 10.0.1.1 10.0.2.2 IPSec SA
 IP 10.0.1.1 10.0.2.2 IPSec Cryptomap
 IP 10.0.2.2 10.0.1.1 IPSec SA
 IP 10.0.2.2 10.0.1.1 IPSec Cryptomap
 IP 10.0.1.1 10.0.100.2 IPSec SA
 IP 10.0.1.1 10.0.100.2 IPSec Cryptomap
 IP 10.0.100.2 10.0.1.1 IPSec SA
 IP 10.0.100.2 10.0.1.1 IPSec Cryptomap
 IP 10.0.3.1 10.0.3.2 IPSec SA
 IP 10.0.3.1 10.0.3.2 IPSec Cryptomap
```

Deny-Jump Behavior with Fail-Close

The **deny jump** behavior occurs if you configure crypto maps with different priorities where the deny entries on the lower priority map form a subset of the high-priority map that causes the jump. The deny-jump behavior is a core IPsec behavior and works based on the crypto map priorities. With Group Encrypted Transport VPN and fail-close, this behavior can cause some confusion.

Following is an example showing how fail-close works and how a packet is processed based on this configuration:

```
crypto map test gdoi fail-close
   match address 102
   activate
crypto map  test 10 isakmp
   match address 101
crypto map test  20 gdoi
   match address 103
   group ks1_group


access-list 101 deny   ip 10.0.1.0 0.0.0.255 10.0.1.0 0.0.0.255
access-list 101 deny   ip 12.0.1.0 0.0.0.255 12.0.1.0 0.0.0.255
access-list 101 permit ip 11.0.0.0 0.255.255.255 11.0.0.0 0.255.255.255
access-list 102 deny   ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 102 deny   tcp any eq telnet any
access-list 102 deny   ospf any any
access-list 103 deny   ospf any any
access-list 103 deny   any host 10.0.5.1
```

```
KS access-list for group ks1_group is:
access-list 110 permit ip any any
```

After activating the fail-close command and prior to successful registration of the group member to the key server, the packet is processed in the following sequence. Note that the key-server group ACL has not been downloaded yet:

ACL Number	Description	Proxy	Action
ACL 101	crypto map ACL	deny ip 10.0.1.0 0.0.0.255 10.0.1.0 0.0.0.255	Jump to next ACL
		deny ip 12.0.1.0 0.0.0.255 12.0.1.0 0.0.0.255	Jump to next ACL
		permit ip 11.0.0.0 0.0.0.255 11.0.0.0 0.0.0.255	Protect the packet
ACL 103	Local group-member ACL	deny ospf any any	Jump to next ACL
		deny any host 10.0.5.1	Jump to next ACL
ACL 102	Fail-close ACL	deny ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255	Packet goes out in the clear
		deny tcp any eq telnet any	Packet goes out in the clear
		deny ospf any any	Packet goes out in clear
Implicit		**permit ip any any**	Packet will be dropped

If there is an IP packet from 10.0.1.1 to 10.0.1.56, the packet hits **deny 10.0.1.0 10.0.1.0** on ACL 101. This entry is a deny entry under the crypto map ACL. The search stops here and jumps to ACL 103 to continue the search. Because there is no match under ACL 103, the search continues under ACL 102. Under the fail-close ACL 102, there is a match in the **deny 10.0.0.0 10.0.0.0** statement. The packet from 10.0.1.1 to 10.0.1.56 goes out in the clear.

If there is an IP packet from 12.0.1.1 to 12.0.1.56, the packet hits a match on ACL 101. This is a deny entry under the crypto map ACL. The search stops at ACL 101 and jumps to ACL 103 to continue the search. If there is no match under ACL 103, the search continues in ACL 102. If no match is found under ACL 102, the packet hits the implicit **permit ip any any** and is dropped.

Configurations

Configurations with One Group Encrypted Transport VPN Group

Key Server for Group “diffint”

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key whatmeworry address 10.0.3.1
crypto isakmp key whatmeworry address 10.0.3.2
!
crypto ipsec transform-set gdoi-p esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-p
 set transform-set gdoi-p
```

```
!
crypto gdoi group diffint
 identity number 3333
 server local
  rekey lifetime seconds 3600
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa mykeys
  rekey transport unicast
  sa ipsec 1
   profile gdoi-p
   match address ipv4 120
   replay counter window-size 64
  address ipv4 10.0.5.2
!
!
interface Ethernet0/0
 description key server interface
 ip address 10.0.5.2 255.255.255.0
!
!.............<snip>…….
!
access-list 120 permit ip host 10.0.1.1 host 10.0.2.2
access-list 120 permit ip host 10.0.2.2 host 10.0.1.1
```

Group Member GM1

```
crypto isakmp policy 1
authentication pre-share
crypto isakmp key whatmeworry address 10.0.3.2
crypto isakmp key whatmeworry address 10.0.3.4
crypto isakmp key whatmeworry address 10.0.4.2
crypto isakmp key whatmeworry address 10.0.5.2
crypto isakmp key whatmeworry address 10.0.5.3
!
!
crypto ipsec transform-set man esp-des
crypto gdoi group diffint
identity number 3333
server address ipv4 10.0.5.2
```

```
!
!
crypto map diffint gdoi fail-close
 match address 133
 activate
crypto map diffint 10 ipsec-isakmp
 set peer 10.0.3.2
 set transform-set man
 match address 120
crypto map diffint 30 gdoi
 set group diffint
!
!
interface Ethernet0/0
 ip address 10.0.1.2 255.255.255.0
!
interface Ethernet1/0
 ip address 10.0.3.1 255.255.255.0
 ip mtu 1000
 crypto map diffint
!
interface Ethernet2/0
 ip address 10.0.4.1 255.255.255.0
 no ip route-cache cef
 no ip route-cache
!
interface Ethernet3/0
 ip address 10.0.5.1 255.255.255.0
!
……..<snip>……..
!
access-list 120 permit ip host 10.0.3.1 host 10.0.3.2
access-list 133 deny   eigrp any any
access-list 133 permit ip host 10.0.1.1 host 10.0.2.2
access-list 133 deny   ip host 10.0.1.1 host 10.0.2.1
access-list 133 deny   ip host 10.0.1.1 host 10.0.3.2
access-list 133 deny   udp any eq 848 any eq 848
```

Group Member GM2

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key whatmeworry address 10.0.3.1
crypto isakmp key whatmeworry address 10.0.5.2
!
!
crypto ipsec transform-set man esp-des
!
crypto gdoi group diffint
 identity number 3333
 server address ipv4 10.0.5.2
!
!
crypto map man 10 ipsec-isakmp
 set peer 10.0.3.1
 set transform-set man
 match address 120
crypto map diffint 30 gdoi
 set group diffint
!
!
interface Ethernet0/0
 ip address 10.0.3.2 255.255.255.0
 crypto map diffint
!
!
interface Ethernet1/0
ip address 10.0.2.1 255.255.255.0
 ip pim sparse-dense-mode
!
access-list 120 permit ip host 10.0.3.2 host 10.0.3.1
```

Configuration with Two Group Encrypted Transport VPN Groups

Key Server 1 Configuration for Group “diffint”

```
crypto ipsec transform-set gdoi-p esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-p
```

```
 set transform-set gdoi-p
!
crypto gdoi group diffint
 identity number 3333
 server local
  rekey lifetime seconds 3600
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa mykeys
  rekey transport unicast
  sa ipsec 1
   profile gdoi-p
   match address ipv4 120
   replay counter window-size 64
  address ipv4 10.0.5.2
!
!
interface Ethernet0/0
 ip address 10.0.5.2 255.255.255.0
```

Key Server 2 Configuration for Group “test”

```
crypto ipsec transform-set gdoi-p esp-3des esp-sha-hmac
crypto ipsec transform-set gdoi-p1 esp-null esp-sha-hmac
!
crypto ipsec profile gdoi-p
 set transform-set gdoi-p
!
crypto ipsec profile gdoi-p1
 set security-association lifetime seconds 19000
 set transform-set gdoi-p1
!
crypto gdoi group test
 identity number 4444
 server local
  rekey address ipv4 121
  rekey lifetime seconds 3600
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa mykeys
  rekey transport unicast
```

```
  sa ipsec 1
   profile gdoi-p
   match address ipv4 120
   replay counter window-size 64
```

Group-Member Configurations: GM1 and GM2

```
GM1#sh run
crypto gdoi group diffint
 identity number 3333
 server address ipv4 10.0.5.2
!
crypto gdoi group test
 identity number 4444
 server address ipv4 10.0.5.3
!
!
crypto map diffint gdoi fail-close
 match address 133
 activate
crypto map diffint 10 gdoi
 set group diffint
crypto map diffint 20 gdoi
 set group test
crypto map diffint 30 ipsec-isakmp
 set peer 10.0.3.2
 set transform-set man
 match address 120
!
!
crypto map test 10 gdoi
 set group test


GM2#sh run
crypto gdoi group diffint
 identity number 3333
 server address ipv4 10.0.5.2
!
```

```
crypto gdoi group test
 identity number 4444
 server address ipv4 10.0.5.3
!
!
crypto map diffint 10 gdoi
 set group diffint
crypto map diffint 20 gdoi
 set group test
crypto map diffint 30 ipsec-isakmp
 set peer 10.0.3.1
 set transform-set man
 match address 120
```

Common Host Configurations

Host1

```
interface Ethernet0/0
 ip address 10.0.1.1 255.255.255.0
 no ip route-cache
!
ip default-gateway 10.0.1.2
ip forward-protocol nd
```

Host2

```
interface Ethernet0/0
ip address 10.0.100.2 255.255.255.0 secondary
 ip address 10.0.2.2 255.255.255.0
 no ip route-cache
!
ip default-gateway 10.0.2.1
ip forward-protocol nd
!
!
access-list 101 permit ip host 10.0.1.1 host 239.0.1.1
```

Conclusion

The Group Encrypted Transport VPN fail-close feature is highly recommended for protecting the flow of data between the group members in your network. It allows you to enforce a desired

security policy for the data traffic during the migration to or installation of a new Group Encrypted Transport VPN network.

During the initial setup and registration of new group members, you may want to filter and limit the type of traffic that flows through the Group Encrypted Transport VPN network. The local fail-close policy helps ensure that the network is secure by restricting unwanted data traffic while at the same time providing critical connectivity for other types of traffic (for example, routing, Telnet, Secure Shell [SSH] Protocol, etc.) during the transition to a group policy.

You must review the exact policy applied in the fail-close ACL before activating the policy. You should be able to access the group member at all times—even when the fail-close policy is activated. Note: As mentioned previously, if you are not careful when designing the fail-close ACL, you could lock yourself out of the device.

For More Information

For more information about the fail-close feature or any other Group Encrypted Transport VPN feature, please contact your Cisco sales representative.

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV
Amsterdam, The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

