Global Information Assurance Certification Paper

Failover of VPN and NAT using ADSL links with Cisco routers

GIAC Security Essentials
Practical Assignment

Version 1.4c

Option 1 – Research on Topics in Information Security

Jean-Maurice Mérel
Security Essentials Track
London
21-26 July 2004

Table of Contents

List of Figures

Abstract

The organization I work with has multiple remote sites that are located in different countries in Asia and are connected to the Headquarter via a site-to-site VPN. The choice of ADSL connectivity has been made because site has a limited number of users and technology was available in all of the capital of the countries. ADSL also combines a permanent connectivity at a fair price with adequate bandwidth.

However in some of the countries, the connectivity provided by the local ISP has demonstrated to be unstable resulting in loss of connectivity several times a month when not a week.

The organization asked if we could improve the continuity of service, installing a second ADSL connection that will be subscribed to other Internet provider.
To provide a repartition of the traffic, one ISP will be used for VPN traffic while the others will provide Internet access.
In case one ISP will fail, the VPN connectivity and the Internet access will be provided via the other ISP. In other words: one ADSL connection will be the backup of the other.

Configuring a connection to a single ISP via ADSL is a “piece of cake”, connecting to two ISP via ADSL at the same time and setting up a true failover without human intervention reveals itself trickier than initially thought. Why?

The first issue was related to the detection of the status of the ADSL connection. When connecting to an ADSL link using Ethernet, sensing the status of the Ethernet interface of the router does not reflect reliably all failure that may occur along the ADSL connection from ISP router to customer router. If we had to manually test the connection, we would ping a remote IP address through the link to make sure it is working right. It’s what “Object tracking”, a recently introduced feature, has helped us to do automatically.

While we use static routes to direct traffic either to Internet or to Headquarter via VPN, static routes should become somehow dynamic to reflect the status of the ISP connectivity. “Object tracking” not only tracks the status of a remote IP, but it allows binding of the state of static routes to the state of a tracking object.

Also note that traffic that leaves the ADSL connection uses the public IP address provided by the ISP: Internet traffic is translated (NATed) using this public IP address, and, VPN traffic is encapsulated using this IP address.
Then connecting with two ISP at a time implies that traffic that flows through one ISP has to use the IP address provided by this ISP.
When failover from an ISP to the other occurs, either Internet traffic or VPN

traffic has to use a different IP address and as a consequence connection oriented traffic such as TCP sessions and VPN has to be reinitiated.

Timeouts implied in translated traffic with NAT and Encryption process used by VPN have an important impact in our failover process.

Structure of the document

In the introduction, we look at the issues we had to face, what feature or trick helped us to solve them.

In the "Initial configuration" part, we present the basic and main configuration of an ADSL router connected to a single ISP.

In "Connecting with two ISP via ADSL" we explain the changes we've made to the initial configuration to support two connections and what where the limits of our failover mechanism.

In "Enhancing static routing with object tracking", we explain how we've improved our failover process with the "object tracking" feature.

In the fourth part, we show failover in action on our router.

What is not addressed in this document

This document focuses on topics that will help to create the failover conditions and increase availability. The topics are:

- IP interfaces and IP route,
- Object tracking configuration
- NAT overload configuration
- VPN configuration.

There are other topics that are critical to the security of a router connected directly to Internet. They are very well covered in other SANS papers and documents available on the Internet:

- CBAC or IOS Firewall Feature set. It provides a basic stateful Firewall functionality to the router. [1]
- Hardening Cisco router configuration. [2]
- Detailed configuration of an IPSEC VPN on a Cisco router. [3]

---

[1] - Cisco IOS Firewall Feature Set Foundations " by Evan Davies, http://www.sans.org/rr/whitepapers/firewalls/806.php (01 Nov 2004)
[2] USA National Security Agency, "Cisco router security configuration" , nsa2.www.conxion.com/cisco/guides/cis-2.pdf (20 sept 2002)
[3] Ryan Ettl, "Understanding and Configuring IPSec between Cisco Routers", (12 Nov 2003)

To someone willing to reproduce our configuration, we will strongly recommend to refer to the above papers.

Introduction

Detecting failover of an ADSL link and making static routes somehow "dynamic"

When a router is connected to a leased line via a serial connection, if a problem occurs along the point-to-point connection such as a modem defection, or a loss of synchronization signal or the absence of keepalive frames, it is usually reflected on the interface status going "down". In other words, the status of the interface shows reliably connectivity has been lost. On the router, the consequence is the deletion of the connected route in the routing table and of the static routes bound to the serial interface.

With an ADSL connection the WAN interface of the router is either an ATM interface directly connected to the phone line, or, an Ethernet interface connected to an ADSL modem. Because ADSL connections and parameters vary from one country or one ISP to the other, the adoption of an Ethernet interface for ADSL connection avoids dealing with those differences.
Those differences are managed by the ADSL modem, which interfaces to PC or to router through an Ethernet interface.

Components between routers and ISP include: ADSL modem, cabling (wiring and fibers), DSLAM (Digital Subscriber Line Access Multiplexer ends the ADSL connections), and the router of the ISP. Then, if one of these components is defective, connectivity will certainly be lost but the status of the Ethernet interface of the router will not reflect it.

Running a dynamic routing protocol with its periodic update or hello would be a way to manage it. However, it is rare to find ISP willing to do it, and customer may as well not want to run routing protocols with ISP.

We then need other mechanism to reflect the real status of the ADSL connectivity. A Network technician will "naturally" do a ping to a remote IP address to test the health of the connection. It's what Cisco has introduced with the "Object tracking" option.
In April 2004, in the Tech Tips and Training section of its Packet Magazine, Cisco has given an example of configuration illustrating "static and policy routing enhancement"[4] based on the *"object tracking"* technology. This recent feature in IOS (OS of Cisco routers) allows monitoring the status of a given resource such

[4] *Cisco Systems Shyan Wignarajah and Asad Faruqui, « PACKET VOL. 16, NO. 2, SECOND QUARTER 2004: Tech Tips and Training: Static and Policy Routing Enhancements* "

as a ping-able IP address and binding the status of a static route to the availability of the tracked object.

With Object tracking, availability of the ADSL link can be truly tested and gives a much more reliable way to trigger the failover from one ISP to the other.

Synchronizing NAT overload and failover

In an ADSL connection, Internet access is realized with a PAT (also named NAT overload)[5] mechanism that translates Private IP addresses used by clients on LAN to the public IP address provided by the Internet Provider. This IP address is either delivered dynamically or provided statically by the ISP.
In our case because we want to setup a site to site VPN, we choose an ADSL subscription with a static IP address.

When clients of the LAN of the remote site want to access Internet, their private IP address is translated ("NATed") using the public IP of the Ethernet-ADSL link via a "NAT overload (or PAT)" dynamic session. When traffic is received and sent by the ADSL interface, it is with the IP address of that specific interface: because it is NATed or because it is encapsulated by VPN.

The following command is the way NAT overload is configured on Cisco classically:
ip nat inside source list 1 interface Ethernet0 overload

When entering two commands as:
ip nat inside source list 1 interface Ethernet0 overload
ip nat inside source list 1 interface Ethernet1 overload
The last one overrides the previous one because the source address is the same in both commands. Even if both commands were accepted, how the router will choose between both?
We need a mechanism that "NATs" Internet traffic based on what is the current outbound interface.

Cisco solved this issue introducing the "*route-map*" keyword in the NAT command. We're now able to define a "conditional" NAT based on the current IP routing table. In other words, if the primary ISP for Internet access fails, failover will update the routing table and NAT will translate IP source address to the public IP address of the backup ISP.

[5] *NAT: Network Address Translation. One IP address is translated to an other. PAT: Port Address Translation. A "socket"= IP address + port (TCP or UDP) is translated to an socket.*

Synchronizing IPSEC VPN and failover

When setting up our VPN, end of the tunnel is one of the public IP addresses of the ADSL links. If that link fails, failover will trigger changes in static routing, that will direct VPN traffic to the other ADSL link, that will in turn force creation of a new VPN using the new IP address at end of the VPN.

New IKE negotiation will have to start and new Security Associations (SA) to be created. However because of the Security Association's lifetime, the previous SA will remain alive for some time on the Headquarter router and will slow down the failover process. Use of proprietary Cisco *"ISAKMP Keepalive*" helps to speed the end of life of SA and in turn to accelerate the general failover.

Conventions

In the rest of the document we'll name our Internet Service Providers: "ISP1" and "ISP2".
We'll also describe two sites: the "remote site" which will be connected to Internet with ADSL links and the "Headquarter site".

Initial configuration

In this part, we show the basic configuration of an ADSL router connected to a single ISP, focusing on the following topics:

- IP addressing and static routes
- NAT configuration
- VPN Configuration

We'll make the configuration evolve in the other parts to the final configuration.

IP addressing and static routes

The interface Ethernet0 is connected to the ADSL Modem and the interface FastEthernet 0 connected to the LAN

```
hostname Test_2isp

interface Ethernet0
 description to ISP1
 ip address 192.168.1.1 255.255.255.252

interface FastEthernet0
description LAN
 ip address 192.168.200.1 255.255.255.0
```

Note: many ISPs require setting up PPOE (Point to Point Over Ethernet) for ADSL connection. PPOE permits parameters negotiation of the connection, authentication of the customer with the provider, dynamic assignment of an IP address.

For Internet access we simply need a default route pointing to the interface of ISP1's router

```
ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

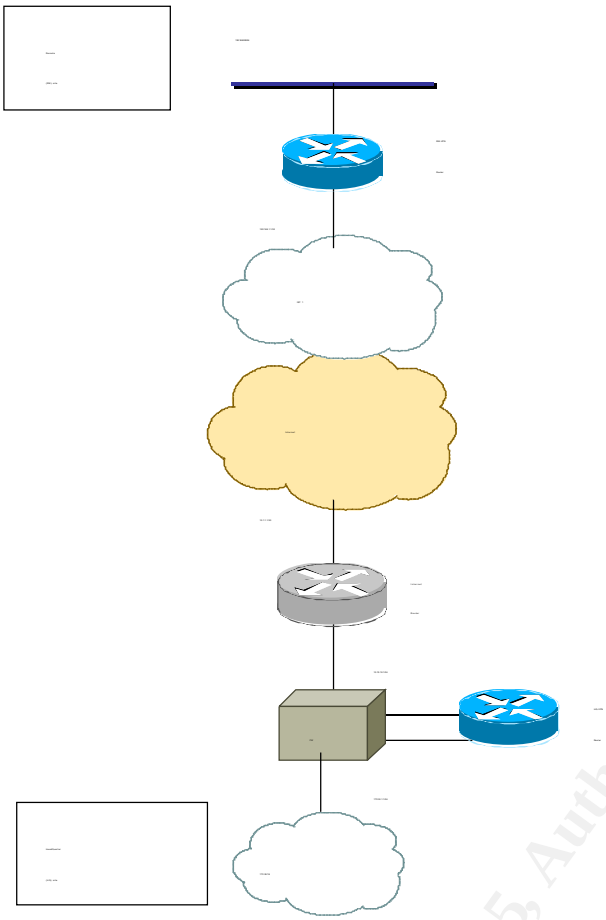

Figure 1: Connection with one ISP

NAT configuration

The traffic to be NATed is defined with an access control list (ACL).

```
ip access-list extended inet-traffic
(1) deny ip 192.168.200.0 0.0.0.255 172.30.0.0 0.0.255.255 log
(2) permit ip 192.168.200.0 0.0.0.255 any
```

Line (1) defines traffic to HQ Network that flows though the VPN and is not NATed, Line (2) defines Internet traffic that is NATed

The following defines the dynamic translation of the IP source addresses of the traffic from LAN to Internet.

```
ip nat inside source inet-traffic interface Ethernet0 overload
```

Translation is initiated from “Inside NAT” interface to “outside NAT” interface.

```
interface Ethernet0
 description to ISP1
 ip nat outside

interface FastEthernet0
 description LAN
 ip nat inside
```

VPN configuration

The following creates an IKE policy with 3DES as the encryption algorithm for data, MD5 as the hashing or digital signature technique and a pre-shared key as the mutual authentication between both VPN peers:

```
crypto isakmp policy 10
 encryption 3des
 hash md5
 authentication pre-share
```

The following defines the shared key used with the given remote peer:

```
crypto isakmp key Super-lonG-kEy address 10.10.10.1
```

The following access list defines the traffic between local IP subnet and IP subnet of the headquarter network to be encrypted:

```
ip access-list extended vpn-traffic
 permit ip 192.168.200.0 0.0.0.255 172.30.0.0 0.0.255.255
 deny ip any any log
```

The transform set defines two options on how traffic will be encrypted and digitally signed. VPN peers initially negotiate these options in the first phase of VPN creation:

```
crypto IPSec transform-set with-HQ esp-3des esp-md5-hmac
```

The crypto map binds the IPSEC access list, the transform set and the remote IPSEC peer:

```
crypto map to-HQ 10 IPSec-isakmp
 set peer 10.10.10.1
 set transform-set with-HQ
 match address vpn-traffic
```

The crypto map is then applied to the outbound interface (ADSL interface):

```
interface Ethernet0
 crypto map to-HQ
```

HQ site Configuration

Following is the configuration of the HQ VPN router:

IP addressing and IP routes:

```
hostname HQ-VPN

interface Ethernet0
 description to Internet via Firewall
 ip address 10.10.10.1 255.255.255.0
 crypto map to-RM

interface FastEthernet0
description to HQ LAN via Firewall
```

```
 ip address 172.30.1.1 255.255.255.0

ip route 0.0.0.0 0.0.0.0 10.10.10.2
```

VPN configuration.

ACL that defines the VPN traffic

```
ip access-list extended vpn-traffic
 permit ip 172.30.0.0 0.0.255.255 192.168.200.0 0.0.0.255 log
```

VPN definition:

```
crypto isakmp policy 10
 encryption 3des
 hash md5
 authentication pre-share
crypto isakmp key Super-lonG-kEy address 192.168.1.1

crypto IPSec transform-set with-RM esp-3des esp-md5-hmac

crypto map to-RM 10 IPSec-isakmp
 set peer 192.168.1.1
 set transform-set with-RM
 match address vpn-traffic

interface Ethernet0
 crypto map to-RM
```

Connecting with two ISPs and ADSL links

In this part, we focus on configuring the router to take advantage of a connection with a second ISP. We review the configuration of static routes, NAT and VPN.

Definition

When connecting our site to two ISPs, we look for more availability: one ISP must replace the other one in case it fails. This should work automatically for both VPN and Internet accesses.

To take advantage of the two IPS, we choose to use:

- ISP1 as Internet primary connection and backup connection for VPN
- ISP2 as primary connection for VPN and Internet backup connection.

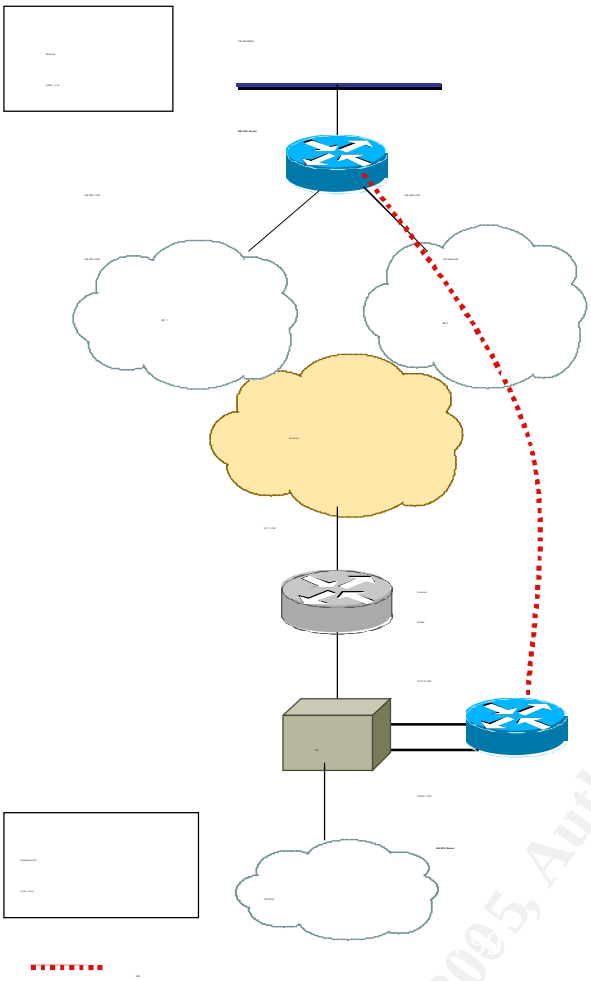

Figure 2: Connection with two ISPs

IP address and IP Routes

Let's define a new IP address for the new interface leading to ISP2

```
interface Ethernet1
 description to ISP2
 ip address 192.168.2.1 255.255.255.252
```

The primary default route is via ISP1

```
ip route 0.0.0.0 0.0.0.0 192.168.1.2
```

The backup default Route points to ISP2 and is defined with an administrative distance (weight) of 200. It will appear in Routing Table when the primary default route is removed.

```
ip route 0.0.0.0 0.0.0.0 192.168.2.2 200
```

For VPN traffic to use ISP2, we have to define a static route for VPN traffic.

```
ip route 172.30.0.0 255.255.0.0 192.168.2.2
```

A backup route is defined via ISP1

```
ip route 172.30.0.0 255.255.0.0 192.168.1.2 200
```

Once the traffic for the Headquarter is encapsulated into an IPSEC packet, the destination IP address of the IPSEC packet is the IP of the remote peer (10.10.10.1). We have then to define a static route to this remote peer that is consistent with the static route for the VPN traffic and points to ISP2. We would otherwise experience a conflict and VPN will not establish.

```
ip route 10.10.10.1 255.255.255.255 192.168.2.2
```

This is static route is also backed up by:

```
ip route 10.10.10.1 255.255.255.255 192.168.1.2 200
```

In résumé:

- traffic to Internet is directed by default route to ISP1. If ISP1 fails, it will be directed to ISP2.
- two static routes to ISP2 direct VPN traffic to Headquarter (172.30.0.0/16). If ISP2 fails, it will be directed to ISP1.

NAT

The new interface to ISP2 is defined as an “Outside NAT“ interface.

```
interface Ethernet1
 description to ISP2
 ip Nat outside
```

To be able to NAT traffic via the second ISP, it would make sense to type the following:

```
ip nat inside source inet-traffic interface Ethernet1 overload
```

However when applied, this command erases the previous command we typed:

```
ip nat inside source inet-traffic interface Ethernet0 overload
```

So this does not work! Let’s think at what we’re looking for: if ISP1 is up, we would like Internet traffic to be NATed with the IP address provided by ISP1. If ISP1 is down, we would then like the traffic to be routed to ISP2 and be NATed using the IP address delivered by ISP2.

The other aspect we’ve to take in consideration, is order of operations when packet are moved from one interface of a router to the other.
NAT come before routing.[6] Then translation of IP address occurs before routing. But NATing depends on the routing because it is going to use the IP address of the outbound interface!
What does this means? Our router has to know what interface the packet is going to be forwarded to, to be able to define to what IP address source address is going to be translated.
Hopefully, Cisco came with an enhancement to combine the NAT command with a “route-map”[7]. A Route-map allows specifying conditions on which

[6] *Cisco Systems, “NAT Order of operation”, http://www.cisco.com/warp/public/556/5.html*
[7] *Cisco Systems, « NAT Support for Multiple Pools Using Route Map »*

translation will occur. In our case, the conditions we want to match are: the outbound interface to which the traffic is going to be routed and the access-list that defines the traffic to be NATed.

We specify the conditions to match in two different Route-map isp1 and isp2. With route-map isp1, we say: if traffic is for Internet (defined by ACL inet-traffic) and outbound interface is Ethernet0 (in other words, default-route points to ISP1) we get a match.

```
route-map isp1 permit 10
 match ip address inet-traffic
 match interface Ethernet0
```

In route-map isp2, we check if we match two conditions: traffic is for Internet (defined by ACL inet-traffic) and outbound interface is Ethernet1 (default-route point to ISP2)

```
route-map isp2 permit 10
 match ip address inet-traffic
 match interface Ethernet1
```

Then the following NAT command instructs the router to translate source address of IP packet to the IP address of interface Ethernet0 (ISP1) when conditions in route-map isp1 are matched or in others words if traffic is routed to ISP1

```
ip nat inside source route-map isp1 interface Ethernet0 overload
```

Then the following NAT command instructs the router to translate source address of IP packet to the IP address of interface Ethernet1 (ISP2) when conditions of route-map isp2 are encountered or in others words if traffic is to be routed to ISP2.

```
ip nat inside source route-map isp2 interface Ethernet1 overload
```

IPSEC VPN

There is nothing specific in the VPN configuration for the remote site compare with the basic configuration. The only change that impacts directly the VPN is related to the static routes.
On the Headquarter peer, we have to define the two potential IP addresses that can be used to setup the VPN tunnel.

HQ site configuration

IKE configuration requires definition of a second peer. Connection of Remote site via ISP1 is put in first position and connection via ISP2 in the second position. What does this means? If Headquarter initiate the VPN connection it will try to connect to the first address first and if not getting any result will try the second one. If Security Association is already defined with the first address and it does not respond anymore, headquarter VPN router, will try to set the VPN

http://www.cisco.com/warp/public/105/nat_routemap.pdf (04 may 2004)

with the second address. This will happens at the end of the lifetime of the SA.

```
crypto isakmp key Super-lonG-kEy address 192.168.1.1
crypto isakmp key Super-lonG-kEy address 192.168.2.1

crypto map to-RM 10 IPSec-isakmp
 set peer 192.168.1.1
 set peer 192.168.2.1
```

Testing redundancy

For redundancy to occurs we have to disconnect one of the Ethernet interface otherwise change in the static routes will not occur.
When disconnecting Ethernet0 (our primary ISP for Internet traffic), we had a failover for Internet traffic after several minutes.
When disconnecting Ethernet1 (our primary ISP for VPN traffic), failover on VPN was happening in a rand of 5 to 10 sec if initiated from remote site, but in a much longer time if initiated from Headquarter.

Conclusion

We've improved availability providing two links. However failover is based on the status of the Ethernet interfaces connected to ADSL modem. If something is wrong with ADSL Modem or if ISP looses connectivity with the rest of Internet, interface status will not change and failover will not happen.

So we need a tool that will trigger the failover not on interface status but on true connectivity of ISP.
Also we need to improve failover conditions so it speeds up.

See last part named "Testing failover" for more details.

"Object tracking" to enhance static routing

Enhance static routing with Object tracking

What would be a simple way to test the connectivity manually via ISP1?
We could simply do a contiguous ping to a remote IP address of our choice.
Cisco has introduced "Object tracking" in the version 12.3 of their IOS® to do the same thing from the router.

Let's see what Cisco says about it[8]:

[8] *Cisco Systems, « Reliable Static Routing Backup Using Object Tracking », http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00801d862d.*

> *"Tracked objects is a generic mechanism in Cisco IOS® Software used to monitor items of interest, and notify applications if the item changes state. Tracked objects provide a loosely coupled set of building blocks that applications such as static routing or policy routing can use to build on**. In this case, a tracked object is created to monitor the state of the SAA probe. Then a static route is configured and associated with the tracked object. Static routing only refers to the tracked object and the tracked object refers to the SAA probe.***
>
> If the tracked object is UP (meaning the SAA probe succeeded), the route is installed in the routing table. Traffic to the Internet will go via the primary ISP. If the tracked object is DOWN (meaning the SAA probe failed), then the route is removed from the routing table, and a floating backup route is installed into the routing table that allows traffic to reach the Internet via the secondary ISP.

Support for this feature was integrated into Cisco IOS release 12.3(8)T.

Step to configure "Object tracking"

First **step:**
Create a probe, named a *Service Assurance Agent* (SAA. It will monitor the state of a connection using ICMP ping packets. The probe specifies the source IP address of the ICMP probe packets, allowing the selection of the link through which connectivity will be tested.

Second step:
A *track object* is created to monitor the status of the SA Agent.

You may wander: why do we need to create a tracked objects AND a SAA? Cisco provides an explanation[9];

> *"Instead of the static route directly monitoring the SAA probe, it monitors the probe via the tracked object. This might seem complex from a configuration standpoint, but it's more efficient from a code development standpoint. If ten applications were all interested in monitoring two types of items, each application would have to create new functions to do it (10 applications x 2 items = 20 new functions). Using track objects, the same scenario would require a new function for each of the two tracked objects, and 10 new functions to monitor the tracked objects (10 new functions to*

html (23 Nov 2004)

[9] *Cisco Systems, « Reliable Static Routing Backup Using Object Tracking », http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00801d862d.html (23 Nov 2004)*

monitor the tracked objects + 2 new functions for the tracked objects to monitor the items = 12 new functions).”

Third **step**:
We associate an application, here a static route, to the tracked object. The state of the static route is based on the status of the track object.

What IP address should we probe?

When defining a probe, an important question is: what remote IP address should we probe?

Let’s discuss 3 possible choices:

1- The IP address of the router of the ISP
   +: we test the ADSL connection up to the first hop of the ISP.
   +: usually the ISP allows us to ping the router we’re connected to
   -: it does not prove the full connectivity of the ISP to Internet

   Conclusion: this IP address does not test fully Internet connectivity of ISP however it is available all the time.

2- The IP address of a node on Internet
   +: It allows testing of connectivity to Internet
   -: We strictly probe the reachability of this specific address on Internet.
   -: If this IP address is not ours, are we allowed to ping it? Could an Intrusion Detection System blacklists us? Will the IP address still be available in 6 months, 1 year and more from now?

   Conclusion: are there public IP addresses that can be ping on the Internet without agreement? I guess so but was unable to find a list on Internet. I would then assume as a safe and respectful attitude, to previously look for an agreement with the owner of the probed IP address. In the long term it is a better guarantee for the lifetime and reliability of our probe. Why not asking to the Internet provider of your corporate HQ site, if one of its IP can be ping?

3- IP address in HQ corporate
   +: Test connectivity all along the way to corporate network
   +: This IP address is ours, we can setup our rules on the Firewall to make it “ping able”.
   -: Connectivity maybe lost with Headquarter but Internet connectivity still is up for remote site.

   Conclusion: this IP address is ours, and can be used with internal

agreement. It is a perfect choice for static routes used with VPN. Without any other choice, why not use it also for probing our Internet access.

Enhancing static routing for Internet traffic

First step: we define a probe that will ping the IP address of the ISP1 router on the other end of the ADSL link every 10 sec (frequency) using the source IP address of the primary link (192.168.1.1).
Probe is scheduled to run "forever".

```
rtr 1
 type echo protocol ipIcmpEcho 192.168.1.2 source-ipaddr
192.168.1.1
 frequency 10
rtr schedule 1 life forever start-time now
```

Second step: we create the tracked object that will be associated with our static default route.

```
track 111 rtr 1 reachability
```

Third step: our primary default-route via ISP1 is associated with the tracked object #111.

```
ip route 0.0.0.0 0.0.0.0 192.168.1.2 track 111
```

If the probe (rtr 1) times out, the status of the tracked object (track 111) will go down, then the static route (0.0.0.0/0) will be removed from the routing table and will be replace by the floating/backup default route:

```
ip route 0.0.0.0 0.0.0.0 192.168.2.2 200
```

NAT timeouts

When failover condition occurs for Internet traffic, failover for existing traffic is slow down because the router caches the NAT translations. If you clear the NAT cache this speed the failover. To make it occurring automatically we can change the timeout values for NAT.
Default NAT timeout values are:

- *timeout*: 86,400 seconds (24 hours): except for overload translations
- *udp-timeout*: 300 seconds (5 minutes): for UDP packets
- *dns-timeout*: 60 seconds (1 minute): for DNS packets
- *tcp-timeout*: 86,400 seconds (24 hours): for TCP packets
- *finrst-timeout*: 60 seconds (1 minute): after RST (reset) or FIN (finish) has been received.
- *icmp-timeout*: 60 seconds (1 minute): for ICMP packets

To reduce downtime during failover we change the timers to 30 sec:

```
ip nat translation timeout 30
ip nat translation tcp-timeout 30
ip nat translation udp-timeout 30
ip nat translation icmp-timeout 30
```

Enhancing static routing for VPN traffic

VPN traffic uses ISP2 as its primary link so we'll track the status of connectivity via ISP2.
Then we'll associate the static routes used by the VPN with the new tracked object.

First step: A probe ping the IP address of the router of ISP2 on the other end of the ADSL link. Frequency is every 10 sec.

```
rtr 2
 type echo protocol ipIcmpEcho 192.168.2.2 source-ipaddr
192.168.2.1
 frequency 10
rtr schedule 2 life forever start-time now
```

Second step: we create the tracked object associated with the static routes used by VPN traffic:

```
track 222 rtr 2 reachability
```

Third step: let's associate the static routes used by VPN to the tracked object

```
ip route 172.30.0.0 255.255.0.0 192.168.2.2 track 222
ip route 10.10.10.1 255.255.255.255 192.168.2.2 track 222
```

If the track object is down, the static routes will be removed from the routing table and will let appear the floating static routes defined as following:

```
ip route 172.30.0.0 255.255.0.0 192.168.1.2 200
ip route 10.10.10.1 255.255.255.255 192.168.1.2 200
```

VPN Security Association lifetime

When we loose connectivity via ISP2, all routes for VPN are directed to ISP1. Because of the change in the routing table, a new VPN is created using ISP1 IP address. Failover occurs is some seconds.

If the down time of ISP2 is short, when routing will come back to the initial communication, the Security Association (SA) build initially via ISP2 will still be alive on remote peer because the default lifetime of SA is equal to 3600 sec or one hour. This will cause following type of error messages:

```
%CRYPTO-4-RECVD_PKT_INV_SPI (x1): decaps: rec'd IPSEC packet has
invalid spi for destaddr=[IP_address], prot=[dec], spi=[hex]([dec])
```

```
Explanation: An IPSec packet was received that specified an SPI that
does not exist in the SADB (Security Associations DataBase). This may
be a temporary condition due to slight differences in aging of SAs
between the IPSec peers, or it may be because the local SAs have been
cleared.
```

To speed clearing of SA and failover conditions, Cisco has introduced a proprietary feature named “ISAKMP keepalives” or “Dead peer detection periodic message”.
Here is its definition by Cisco:[10]

> *“The IPSec Dead Peer Detection Periodic Message Option feature allows you to configure your router to query the liveliness of its Internet Key Exchange (IKE) peer at regular intervals. The benefit of this approach over the default approach (on-demand dead peer detection) is earlier detection of dead peers.”*

The following command define that keepalive messages will be sent every 30 sec, after 5 retries without any message or data from remote peer, it will be considered as “dead” and SA will be terminated.

```
crypto isakmp keepalive 30 5
```

On the Headquarter site, it will speed the end of life of the Security Association with one IPSEC peer and the move to the other IPSEC peer.
We use also to clean up SA after a failover condition.

Testing failover for Internet and VPN traffics

For our testing, we’ve disconnected cable between the ADSL modem and the phone line, so the status of our Ethernet interfaces was unchanged.

Both ISPs are up

IP routing table:
Default route points to ISP1 (192.168.1.2).
172.30.0.0/16 and 10.10.10.1/32 are used for VPN and points to ISP2 (192.168.2.2)

```
test-2isp#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
…
Gateway of last resort is 192.168.1.2 to network 0.0.0.0

 192.168.2.0/30 is subnetted, 1 subnets
```

[10] *Cisco Systems, “IPSec Dead Peer Detection Periodic Message Option”,http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801ee19a.html*

```
C 192.168.2.0 is directly connected, Ethernet1/0
C 192.168.200.0/24 is directly connected, FastEthernet0/0
 10.0.0.0/32 is subnetted, 1 subnets
S 10.10.10.1 [1/0] via 192.168.2.2
 192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Ethernet0/0
S* 0.0.0.0/0 [1/0] via 192.168.1.2
S 172.30.0.0/16 [1/0] via 192.168.2.2
```

Track objects are both up.

```
test-2isp#show track brief
Track Object Parameter Value
111 rtr 1 reachability Up
222 rtr 2 reachability Up
```

Internet traffic is NATed using IP address given by ISP1: 192.168.1.1

```
test-2isp#sh ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 192.168.1.1:24065 192.168.200.10:24065 10.199.199.199:23 10.199.199.199:23
```

VPN is established via ISP2: see IP address is 192.168.2.1

```
test-2isp#sh crypto isakmp sa
dst src state conn-id slot status
192.168.2.1 10.10.10.1 QM_IDLE 1 0 ACTIVE
```

ISP1 is down

ISP1 is down. We expect Internet and VPN traffic to flow through ISP2.

Track object 111 is down.

```
debug track
*Mar 3 05:44:38.745: Track: 111 Change #40 rtr 1, reachability Up->Down
test-2isp#sh track brief
Track Object Parameter Value
111 rtr 1 reachability Down
222 rtr 2 reachability Up
```

Main default route has been deleted and replace by the backup (Admin distance = 200), all routes are via ISP2 (192.168.2.2)

```
test-2isp#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

Gateway of last resort is 192.168.2.2 to network 0.0.0.0

 192.168.2.0/30 is subnetted, 1 subnets
C 192.168.2.0 is directly connected, Ethernet1/0
C 192.168.200.0/24 is directly connected, FastEthernet0/0
 10.0.0.0/32 is subnetted, 1 subnets
S 10.10.10.1 [1/0] via 192.168.2.2
S* 0.0.0.0/0 [200/0] via 192.168.2.2
S 172.30.0.0/16 [1/0] via 192.168.2.2
```

VPN traffic flows via ISP2 (no change)

```
test-2isp#sh crypto isakmp sa
dst src state conn-id slot status
192.168.2.1 10.10.10.1 QM_IDLE 1 0 ACTIVE
```

Now, Internet traffic is NATed using ISP2's IP address (192.168.2.1)

```
test-2isp#sh ip nat translations `
Pro Inside global Inside local Outside local Outside global
tcp 192.168.2.1:23553 192.168.200.10:23553 10.199.199.199:23 10.199.199.199:23
```

Experienced failover for a continuous ping is some seconds for the VPN traffic and about 30 sec for Internet traffic.

ISP2 is down

ISP2 is down. We expect Internet and VPN traffic to flow through ISP1.

Track object 222 is down.

```
test-2isp#sh track brief
Track Object Parameter Value
111 rtr 1 reachability Up
222 rtr 2 reachability Down
```

Static route for VPN (172.30/16 and 10.10.10.1/32) has been deleted and replaced by their backup (Admin distance = 200), all routes are via ISP1 (192.168.1.2)

```
test-2isp#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP

Gateway of last resort is 192.168.1.2 to network 0.0.0.0

C 192.168.200.0/24 is directly connected, FastEthernet0/0
 10.0.0.0/32 is subnetted, 1 subnets
S 10.10.10.1 [200/0] via 192.168.1.2
 192.168.1.0/30 is subnetted, 1 subnets
C 192.168.1.0 is directly connected, Ethernet0/0
S* 0.0.0.0/0 [1/0] via 192.168.1.2
S 172.30.0.0/16 [200/0] via 192.168.1.2
```

VPN is flowing via ISP1 now.

```
test-2isp#sh crypto isakmp sa
dst src state conn-id slot status
10.10.10.1 192.168.1.1 QM_IDLE 2 0 ACTIVE
```

Internet traffic is NATed via ISP1

```
test-2isp#sh ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 192.168.1.1:24577 192.168.200.10:24577 10.199.199.199:23 10.199.199.199:23
```

Conclusion

ADSL connection provides permanent and low priced bandwidth to Internet. Combined with VPN it offers a convenient way to connect small and medium sites to the corporate network. When stability of the connection is questionable a second ISP can provide more availability.
Because ADSL connectivity problems do not always result in Interface status changes, configuring an automatic failover can be somehow problematic.

Object tracking from Cisco provides a way to probe a remote IP address and to bind the state of a static route with the probe's status.
To speed the failover, for the Internet traffic we've changed the NAT timeout and for the VPN traffic we've introduced the Keepalive ISAKMP.

References

National Security Agency, "Cisco router security configuration" , nsa2.www.conxion.com/cisco/guides/cis-2.pdf (20 sept 2002)

Cisco Systems, "IPSec Dead Peer Detection Periodic Message Option", http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801ee19a.html

Ryan Ettl, "Understanding and Configuring IPSec between Cisco Routers", (12 Nov 2003)

Cisco Systems Shyan Wignarajah and Asad Faruqui, « PACKET VOL. 16, NO. 2, SECOND QUARTER 2004: Tech Tips and Training: Static and Policy Routing Enhancements », http://www.cisco.com/en/US/about/ac123/ac114/ac173/Q2-04/department_techtips.html (April 2004)

Cisco Systems, « Reliable Static Routing Backup Using Object Tracking », http://www.cisco.com/en/US/products/sw/iosswrel/ps5413/products_feature_guide09186a00801d862d.html (23 Nov 2004)

Cisco Systems, « NAT Support for Multiple Pools Using Route Map » http://www.cisco.com/warp/public/105/nat_routemap.pdf (04 may 2004)

Cisco Systems, « NAT commands» http://www.cisco.com/en/US/about/ac123/ac114/ac173/Q2-04/department_techtips.html

Evan Davies, « CBAC - Cisco IOS Firewall Feature Set Foundations » http://www.sans.org/rr/whitepapers/firewalls/806.php (01 Nov 2004)

Ian C. Rudy, « Building a Secure Enterprise Grade V3PN « http://www.sans.org/rr/whitepapers/modeling/1321.php (01 Nov 2004)

Appendice: configuration

Following configuration focuses on the topics presented into this document: static routing, object tracking, VPN configuration and NAT.

```
test-2isp#sh run
!
no ip cef
!
!
track 111 rtr 1 reachability
!
track 222 rtr 2 reachability
!
!
crypto isakmp policy 1
 authentication pre-share
 group 2
crypto isakmp key Super-lonG-kEy address 10.10.10.1
crypto isakmp keepalive 20 periodic
!
!
crypto ipsec transform-set standard esp-des esp-md5-hmac
!
crypto map adbvpn 10 ipsec-isakmp
 description map to interface using standard transform set
 set peer 10.10.10.1
 set security-association idle-time 60
 set transform-set standard
 match address vpn-traffic
!
!
!
interface Ethernet0/0
 description To IPS1
 ip address 192.168.1.1 255.255.255.252
 ip nat outside
 no ip route-cache
 half-duplex
 no cdp enable
 crypto map adbvpn
!
interface FastEthernet0/0
description To LAN
 ip address 192.168.200.1 255.255.255.0
 ip nat inside
 no ip route-cache
 no ip mroute-cache
 speed auto
!
interface Ethernet1/0
 description TO ISP2
 ip address 192.168.2.1 255.255.255.252
 ip nat outside
 no ip route-cache
 half-duplex
 no cdp enable
 crypto map adbvpn
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.2 track 111
```

```
ip route 0.0.0.0 0.0.0.0 192.168.2.2 200
ip route 172.30.0.0 255.255.0.0 192.168.2.2 track 222
ip route 172.30.0.0 255.255.0.0 192.168.1.2 200
ip route 10.10.10.1 255.255.255.255 192.168.2.2 track 222
ip route 10.10.10.1 255.255.255.255 192.168.1.2 200
!
ip nat translation timeout 30
ip nat translation tcp-timeout 30
ip nat translation udp-timeout 30
ip nat translation icmp-timeout 30
ip nat inside source route-map ISP1 interface Ethernet0/0 overload
ip nat inside source route-map ISP2 interface Ethernet1/0 overload
!
!
!
ip access-list extended inet-traffic
 deny   ip 192.168.200.0 0.0.0.255 172.30.0.0 0.0.255.255 log
 permit ip 192.168.200.0 0.0.0.255 any log
ip access-list extended vpn-traffic
 permit ip 192.168.200.0 0.0.0.255 172.30.0.0 0.0.255.255 log
 deny   ip 192.168.200.0 0.0.0.255 any log
!
route-map ISP1 permit 10
 match ip address inet-traffic
 match interface Ethernet0/0
!
route-map ISP2 permit 10
 match ip address inet-traffic
 match interface Ethernet1/0
!
!
rtr 1
 type echo protocol ipIcmpEcho 192.168.1.2 source-ipaddr 192.168.1.1
 frequency 10
 timeout 30
rtr schedule 1 life forever start-time now
rtr 2
 type echo protocol ipIcmpEcho 192.168.2.2 source-ipaddr 192.168.2.1
 frequency 10
 timeout 30
rtr schedule 2 life forever start-time now
!
!
end
```