

Public IP Address via Bonded ADSL using Zeroshell.

I wanted to bond four ADSL lines together and present a public, non natted IP address to our firewall. In doing this I wanted to aggregate the bandwidth and provide transparent failover if an ADSL connection were to become faulty.

I couldn't find any guides on how to do this and it took me nearly a week to figure it out but the finished product works REALLY well. I have 4x ADSL circuits, all of them 10mb/s downstream and 1mb/s upstream. When I run a speed test over my aggregated ADSLs I get a combined bandwidth of about 35meg down and 3.7 up. Perfect!

You will need two zeroshell boxes. I ordered two reconditioned Dell 1950 1U servers. I installed 2 additional dual head intel network cards into each server, giving me a total of 6 ETH interfaces on each machine.

I installed one zeroshell in our datacentre. I connected ETH0 to a private network which I can dial into for backup access incase I locked myself out for some reason. IP address 192.168.1.2.

On ETH1 I added four public IP addresses. On ETH2 I connected another VLAN with a different range of IP addresses, this is the range that I intended to make available to my firewall.

On site, I added a router to each of my ADSL lines, each router had a block of 4 addresses. I connected the routers to ETH1, ETH2, ETH3 and ETH4 and configured the interfaces with the third (routable) address from the router that was connected.

the next thing I did was create a series of static routes that ensured that when I tried to reach one of my 4 IPs in the datacentre, it forced the connection over each router individually. For example I needed to be sure that when I wanted to get to datacentre IP1, it used ADSL 1, Datacentre IP 2 was reached via ADSL router 2 and so on. I tested the routes were working properly by doing trace routes from the zeroshell to each of my 4 datacentre IP addresses, ensuring that each one went out via the correct router.

The next thing I did was to create the VPNs themselves, this is quite simple. Create four VPNs, one to each of your datacentre IP addresses. Make sure that the VPNs are UDP with no encryption. Make the datacentre side the 'server' and the site the 'client' side. another thing to remember is in the VPN config window you can add additional commands. I suggest adding '--local x.x.x.x'. 'x.x.x.x' is the IP address that you want the VPN to come from, forcing it to come from the correct address. The static routes that we put in earlier should automatically do

this but I found this is worth adding as well.

When you have 4 VPNs up and running, add them to a bond at each end. When you have done this you need to bridge the bond with ETH0 at the client site and bridge the bond on the datacentre box with your other routable IPs, in my case I had these on ETH2.

Make sure the ETH0 on your client site has an IP and mask that is on the routable subnet you have added in your datacentre. For example my range was 45.10.20.0/255.255.255.240, so I gave my client zeroshell ETH0 45.10.20.2/255.255.255.240. Now, when I connected a device to the ETH0 on site, I can see the 45.10.20.0/255.255.255.240 network in the datacentre. Next thing to do is give your firewall a spare IP on the range. I used 45.10.20.3 and put in the gateway as my network gateway in the datacentre - 45.10.30.1.

And that's it, true routable IP addresses over bonded VPNs.

I found this works really well, very low latency and true aggregation for a fraction of the price of a leased line. I run site to site VPNs from my firewall to other sites and they all work really well, the encapsulation on the bonded VPNs on the is completely transparent to anything that you run over the aggregated connection.

John Feeney 19/9/11