

- Automatically inject packets, which destination address is within corporate network addressing, to ipsec0.
- Keep the other packets going through eth1.
- That is: ping www.yahoo.com (Internet access through eth1) must respond, ping 192.168.0.33 (intranet access going through ipsec0) must answer.

One can ask why being so complicated. It's enough to configure default gateway (or a list a subnet in the routing table) on the stations to point towards the br0 address and the router will do its job routing toward intranet or towards Internet.

Perfectly exact.

Yes but imagine you have several dozens of such local stations without DHCP to manage their network configuration

and you are 1000 thousands miles away. You cans send the machine, ask someone to plug it in the router and in the hub with the correct cables you've provided.

That's all.

A solution

So

```
ebtables -t nat -A PREROUTING -i eth0 -p ipv4 --ip-dst 192.168.0.0/19 -j dnat --to-dst $MAC_ADDR_OF_IPSEC0 --dnat-target ACCEPT
```

actually works and is enough for the challenge.

A notice on \$MAC_ADDR_Of_IPSEC0.

Since the rule is set before ipsec is launched mac address of ipsec0 is not set at this time.

Never mind: since ipsec0's mac address will be the same as that of the outbound interface that is equal to eth1's mac address.

And eth1 exists as soon as kernel recognizes it so before firewall (and ebtables) are launched.

Hence this rule:

```
ebtables -t nat -A PREROUTING -i eth0 -p ipv4 --ip-dst 192.168.0.0/19 -j dnat --to-dst $MAC_ADDR_OF_ETH1 --dnat-target ACCEPT
```

Of course that's the first step.

One can then take into account a few other things:

- protecting eth1 at the ip level (via iptables) or at the mac level (anti spoofing via ebtables or iptables),

- protecting intranet from devices attempting to access the tunnel from eth1,
- be sure such mac address is associated with such ip address without using arp -s, via ebtables,
- etc, etc.