



Dynamic routing protocols over IPSec tunnels between Palo Alto Networks and Cisco routers

Tech Note

PAN-OS 4.1

Contents

Overview	3
Summary.....	3
Network architecture.....	4
Hardware and Software versions used on this lab.....	4
Setting up IPsec tunnels	5
Setting up Cisco IPsec VTI tunnel	5
Setting up Palo Alto Networks IPsec tunnel.....	5
Verifying the IPsec configuration	9
Setting up OSPF routing	12
Setting up Cisco OSPF configuration	12
Setting up Palo Alto Networks OSPF configuration	13
Verifying the OSPF configuration	16
Conclusions	19
Revision History	19

Overview

The main goal for this paper is to show how to configure dynamic routing protocols (DRP from now on) between Palo Alto Networks next generation firewalls and Cisco routers, when they are connected via IPsec tunnels. We will focus on OSPF, but other DRPs such as RIP or BGP should be configured in a similar manner.

The content of the paper is mostly technical, and therefore the intended audience is system engineers. We understand that the reader has already a basic knowledge on how to configure a Palo Alto Networks firewall; therefore all the steps to set up a basic configuration are not covered here. More specifically we will cover only the following steps:

- Setting up the IPsec tunnel on Cisco router using VTI
- Setting up the IPsec tunnel on Palo Alto Networks firewall
- Setting up OSPF on Cisco router
- Setting up OSPF on Palo Alto Networks
- Verifying the configurations

The documentation included in this paper is not intended, by any means, to substitute any official document from Palo Alto Networks. The official documentation can be found in the public website and also in the corporate Intranet for the employees.

Summary

Some Cisco VPN configurations rely on the utilization of GRE or L2TP tunnels for encapsulation and crypto maps with IPsec. This means that the utilization of DRPs is encapsulated in Cisco via GRE or L2TP, and therefore the other endpoint of the IPsec tunnel needs to support these tunneling protocols in order to establish successfully a DRP adjacency. Palo Alto Networks, as of PAN-OS version 4.1.1, doesn't support the decapsulation of GRE or L2TP and therefore DRPs over IPsec cannot be configured based upon these protocols.

But Cisco also supports setting up IPsec tunnels based upon VTI (Virtual Tunnel Interface), which are fully compatible with Palo Alto Networks firewalls. Detailed information on setting up Cisco IPsec VTI, can be found on the following link: http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gtIPSecm.html.

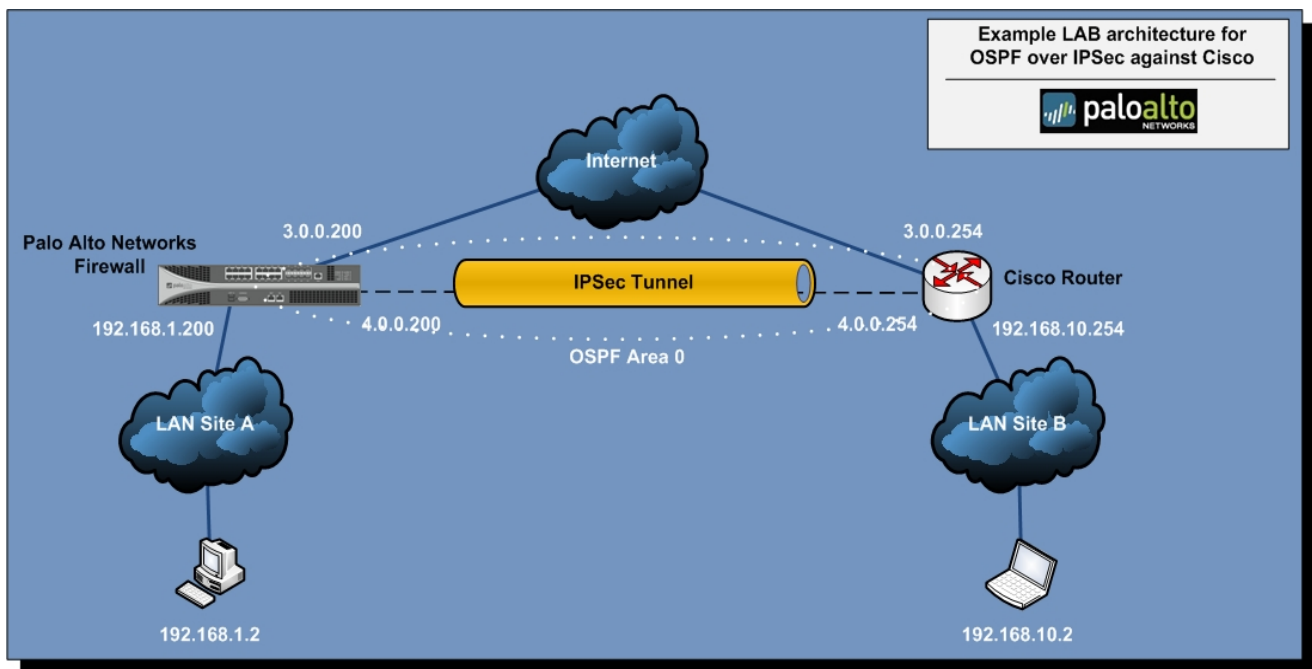
Following there's a short definition on Cisco IPsec VTIs and its benefits, coming from the information in the previous link:

“IP security (IPsec) virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing <...>

<...> The use of IPsec VTIs both greatly simplifies the configuration process when you need to provide protection for remote access and provides a simpler alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation and crypto maps with IPsec.”

Network architecture

The following diagram, Picture 1, shows the logical network diagram that we will use in our lab:



Picture 1.- Lab logical diagram

Mnemonic rules:

- ✓ All the addresses that end on .254 are related to the Cisco router.
- ✓ All the addresses that end on .200 are related to the Palo Alto Networks firewall.
- ✓ All the addresses that end on .2 are related to the end workstations.

Hardware and Software versions used on this lab

- ✓ Cisco: 7200 Router emulated with Dynamips* and running IOS version 12.4(2)T.
- ✓ Palo Alto Networks: PA-2050 firewall running PAN-OS version 4.1.1.

**Note: "Dynamips is a Cisco router emulator written by Christophe Fillot. It emulates 1700, 2600, 3600, 3700, and 7200 hardware platforms, and runs standard IOS images. Of course, this emulator cannot replace a real router, it is simply a complementary tool to real labs for administrators of Cisco networks or people wanting to pass their CCNA/CCNP/CCIE exams."*

More information can be found on the following link, for those readers interested on this tool:
<http://dynagen.org/tutorial.htm>

Setting up IPsec tunnels

Setting up Cisco IPsec VTI tunnel

For this lab we will configure static IPsec VTIs on Cisco side. In the documentation that can be found on http://www.cisco.com/en/US/docs/ios/12_3t/12_3t14/feature/guide/gtIPScrm.html, there are also examples on how you can work with Dynamic VTIs.

Summary steps for setting up the static IPsec VTI in Cisco router are as follow:

```
enable  
configure terminal  
crypto isakmp policy Number  
crypto isakmp key Key address 0.0.0.0 0.0.0.0  
crypto ipsec transform-set transform-set-name transform-set-type  
crypto IPsec profile profile-name  
set transform-set transform-set-name  
interface type Number  
ip address address mask  
tunnel mode ipsec ipv4  
tunnel source interface  
tunnel destination ip-address  
tunnel protection IPsec profile profile-name [shared]
```

Following you will find the detailed Cisco config for this step as configured in our lab:

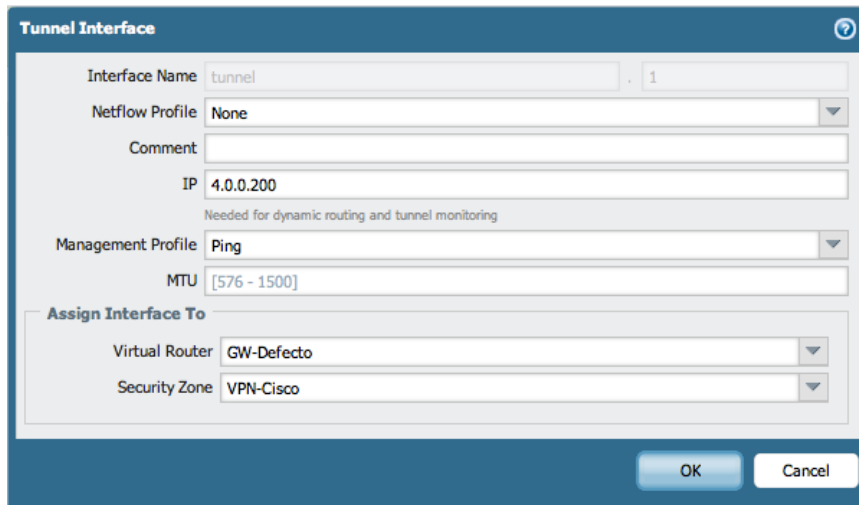
```
crypto isakmp policy 1  
  encr aes  
  authentication pre-share  
  group 2  
crypto isakmp key paloalto address 0.0.0.0 0.0.0.0  
!  
!  
crypto ipsec transform-set T1 esp-aes esp-sha-hmac  
!  
crypto ipsec profile P1  
  set transform-set T1  
!  
!  
interface Tunnel0  
  ip address 4.0.0.254 255.255.255.0  
  ip ospf mtu-ignore  
  load-interval 30  
  tunnel source 3.0.0.254  
  tunnel destination 3.0.0.200  
  tunnel mode ipsec ipv4  
  tunnel protection ipsec profile P1  
!
```

Note: The command “ip ospf mtu-ignore” is important for the OSPF configuration to work, and it will be discussed later on in the OSPF part.

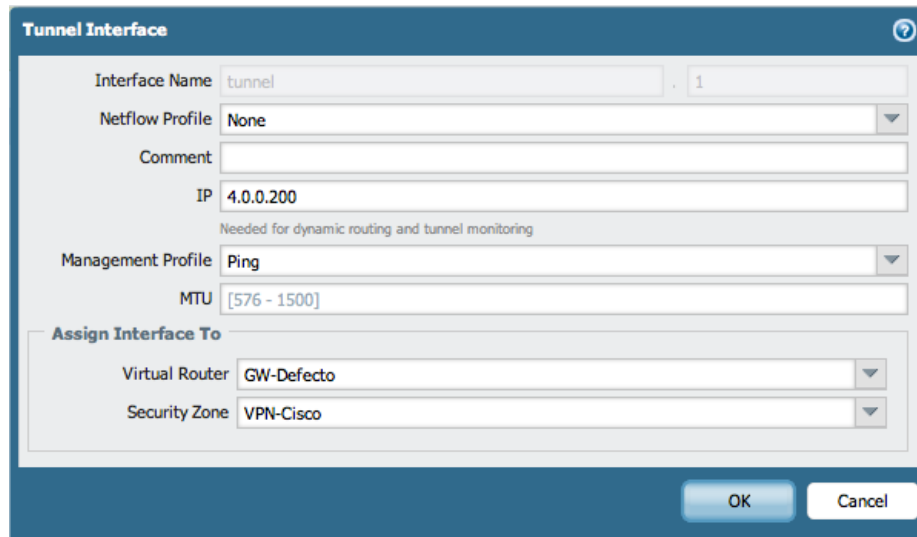
Setting up Palo Alto Networks IPsec tunnel

For the configuration in Palo Alto Networks, we will show both the configuration via GUI and also in the xml config file once is done. Configuration steps, together with the GUI screenshots, are as follow:

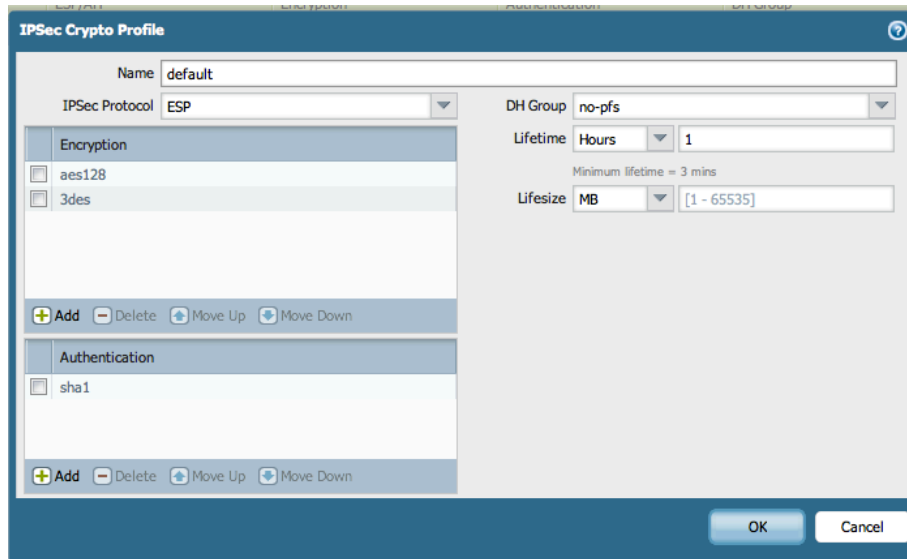
1. Go to the Network tab.
2. Create a Tunnel interface and assign it an IP (required for DRP to work; in our lab is 4.0.0.200). We have decided to place the tunnel on a dedicated and intermediate Security Zone called “VPN-Cisco”. This config will force specific security policies to allow the traffic flowing from Security Zone “VPN-Cisco” to the final internal Security Zone (“LAN in our lab). If you don’t want to create these rules, you can place the tunnel in the “LAN” Zone, as long as you don’t have an explicit “any-any-deny” rule (if this is the case, you still need to set up explicit security rules).



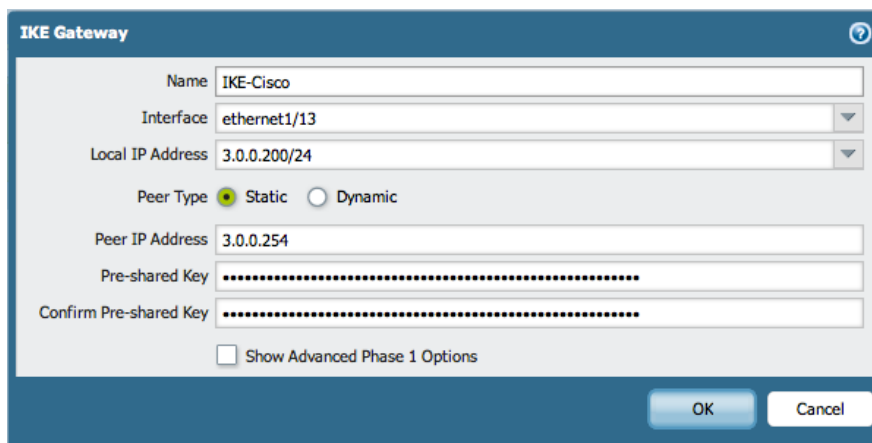
3. Review IKE Crypto default profile settings. We don’t need to change anything here for this lab. These are the crypto settings used for phase 1.



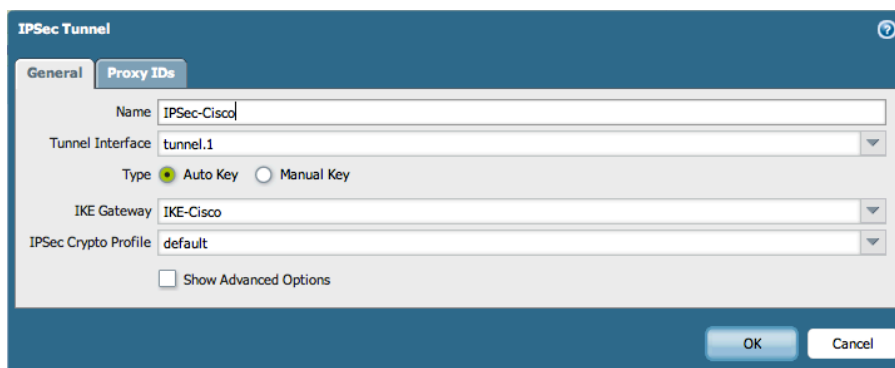
4. Review IPSec Crypto default profile settings. We disable PFS -Perfect Forward Secrecy- since we are not using it in Cisco config (DH Group = “no-pfs”). If we don’t disable it we will receive a proposal mismatch error while trying to set up phase 2.

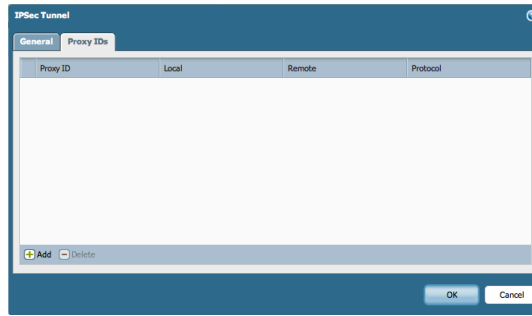


5. **Configure the IKE Gateway.** We select the physical interface (Ethernet 1/13 in our lab), the physical local IP address (3.0.0.200), the peer's physical IP address (3.0.0.254) and the preshared key (*paloalto* in our lab).

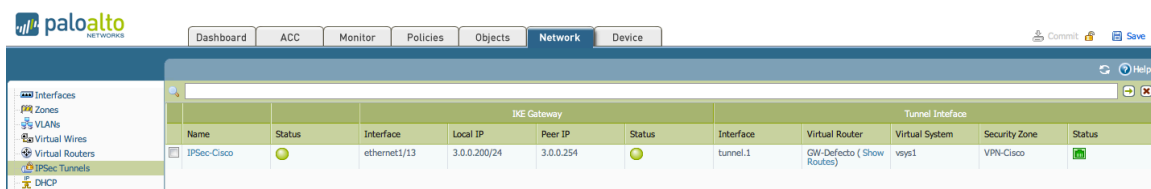


6. **Configure the IPsec Gateway.** We select the tunnel interface created in step 2 (tunnel.1 in our lab), the IKE Gateway from step 5 (IKE-Cisco) and the IPsec Crypto Profile (*default* in our lab, without pfs as explained before). Note that we don't need to configure anything on the Proxy IDs part, because Cisco proposal will be 0.0.0.0/0 and for this one you don't need to configure any Proxy ID.





7. **Commit your policy.** Once is committed, and if everything is ok, you should see two green spots on the IPsec Tunnel indicating that both phases have been completed successfully:



Following we show also the XML config for the IKE and IPsec parts already reviewed:

```
ike {
  crypto-profiles {
    ike-crypto-profiles {
      default {
        encryption [ aes128 3des ];
        hash [ sha1 ];
        dh-group [ group2 ];
        lifetime {
          hours 8;
        }
      }
    }
  }
  ipsec-crypto-profiles {
    default {
      esp {
        encryption [ aes128 3des ];
        authentication [ sha1 ];
      }
      dh-group no-pfs;
      lifetime {
        hours 1;
      }
    }
  }
}
gateway {
  IKE-Cisco {
    protocol {
      ikev1 {
        dpd {
          enable yes;
          interval 5;
          retry 5;
        }
        ike-crypto-profile default;
        exchange-mode auto;
      }
    }
  }
  authentication {
```


You can also review the status of the tunnel, via the following command:

```
admin@PA-2050> show vpn flow name IPSec-Cisco
```

```
tunnel  IPSec-Cisco
id:          4
type:        IPSec
gateway id:  2
local ip:    3.0.0.200      peer ip:    3.0.0.254
inner interface: tunnel.1  outer interface: ethernet1/13
state:       active
session:     7494
tunnel mtu:  1428
lifetime remain: 1481 sec
latest rekey: 2119 seconds ago
monitor:     off
en/decap context: 10
local spi:   FD2C4879
remote spi:  7029E18A
key type:    auto key
protocol:    ESP
auth algorithm: SHA1
enc algorithm: AES128
proxy-id local ip: 0.0.0.0/0
proxy-id remote ip: 0.0.0.0/0
proxy-id protocol: 0
proxy-id local port: 0
proxy-id remote port: 0
anti replay check: yes
copy tos:    no
authentication errors: 0
decryption errors: 0
inner packet warnings: 0
replay packets: 0
packets received when lifetime expired: 0
packets received when lifeseize expired: 0
sending sequence: 428
receive sequence: 1025
encap packets:    1149
decap packets:    2724
encap bytes:      138024
decap bytes:      282752
key acquire requests: 2
```

On Cisco router you can, among other things, verify the status of the tunnel interface:

```
R-jesud#show interfaces tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 4.0.0.254/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 3.0.0.254, destination 3.0.0.200
  Tunnel protocol/transport IPSEC/IP
  Tunnel TTL 255
  Fast tunneling enabled
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Tunnel protection via IPSec (profile "P1")
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
    2806 packets input, 248962 bytes, 0 no buffer
```

```
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
6584 packets output, 889928 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
```

You can also verify the status of the SA (Security Association):

```
R-jesusd#show crypto ipsec sa
```

```
interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3.0.0.254

protected vrf: (none)
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
current_peer 3.0.0.200 port 500
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 6629, #pkts encrypt: 6629, #pkts digest: 6629
#pkts decaps: 2826, #pkts decrypt: 2826, #pkts verify: 2826
#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.: 3.0.0.254, remote crypto endpt.: 3.0.0.200
path mtu 1514, ip mtu 1514
current outbound spi: 0xFD2C4879(4247537785)

inbound esp sas:
  spi: 0x7029E18A(1881792906)
  transform: esp-aes esp-sha-hmac ,
  in use settings ={Tunnel, }
  conn id: 11, flow_id: SW:11, crypto map: Tunnel0-head-0
  sa timing: remaining key lifetime (k/sec): (4425711/1813)
  IV size: 16 bytes
  replay detection support: Y
  Status: ACTIVE
inbound ah sas:

inbound pcp sas:

outbound esp sas:
  spi: 0xFD2C4879(4247537785)
  transform: esp-aes esp-sha-hmac ,
  in use settings ={Tunnel, }
  conn id: 12, flow_id: SW:12, crypto map: Tunnel0-head-0
  sa timing: remaining key lifetime (k/sec): (4425633/1809)
  IV size: 16 bytes
  replay detection support: Y
  Status: ACTIVE

outbound ah sas:

outbound pcp sas:
```

Finally you can also create a static route, both on Cisco and Palo Alto Networks, to point through the tunnel to the private networks on each remote side. Once that this route has been set up, you should be able to ping from 192.168.1.2 to 192.168.10.2 and the opposite. Remember to review the security policy to be sure that the traffic is allowed; as stated in the Introduction chapter we won't cover that part of the configuration on this document. The routes for each device should be:

- ✓ Palo Alto Networks firewall: 192.168.10.0/24 → 4.0.0.254
- ✓ Cisco router: 192.168.1.0/24 → 4.0.0.200

Setting up OSPF routing

Once that the tunnel configuration has been done, the OSPF configuration is pretty standard both in Cisco and also in Palo Alto Networks. As shown in diagram 1, and for sake of simplicity, we will configure only OSPF and using only one area (backbone area 0).

Once that the configuration has been completed, both devices should learn dynamically via OSPF the private networks on each remote site, through the tunnel interfaces.

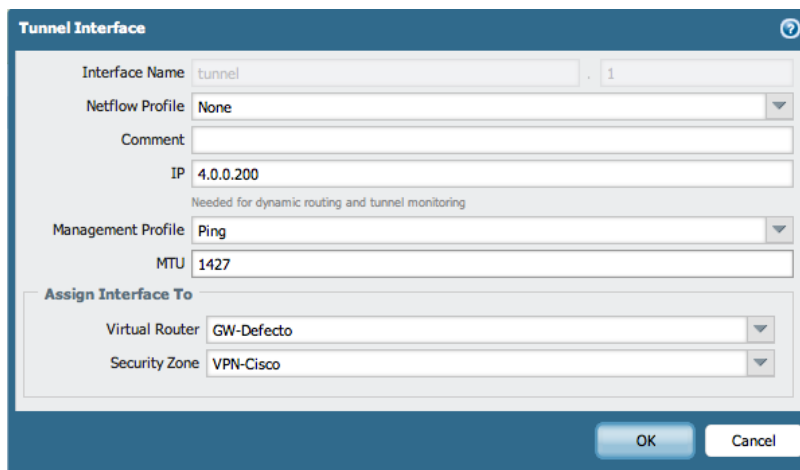
Setting up Cisco OSPF configuration

Following you have the OSPF configuration created for our lab on the Cisco router. As you can see first we create the instance for our OSPF process (instance 1). After that, we add in area 0 the router interfaces holding the tunnel network (4.0.0.0/24) and also the private network (192.168.10.0/24), where the workstation is placed; this is done via the “network” command:

```
router ospf 1
 log-adjacency-changes
 network 4.0.0.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
```

Note 1: It’s important to remember that in the tunnel interface definition we added the command “ip ospf mtu-ignore”. This is because Cisco OSPF checks whether neighbors are using the same MTU on a common interface. If the receiving MTU in the DBD packet is higher than the IP MTU configured on the incoming interface, OSPF adjacency will not be established. You will notice this problem because OSPF adjacency will not happen and you will receive error messages about MTU, while doing “debug ip ospf events” in the router.

Another alternative is to adjust the MTU on the firewall side. We did it also in our lab decreasing the value from 1500 (default) to a value of 1427, under the definition of the tunnel.1 interface. At this point you can remove the “ip ospf mtu-ignore” from Cisco configuration. The following screenshot show this setting configured in Palo Alto Networks tunnel interface definition:



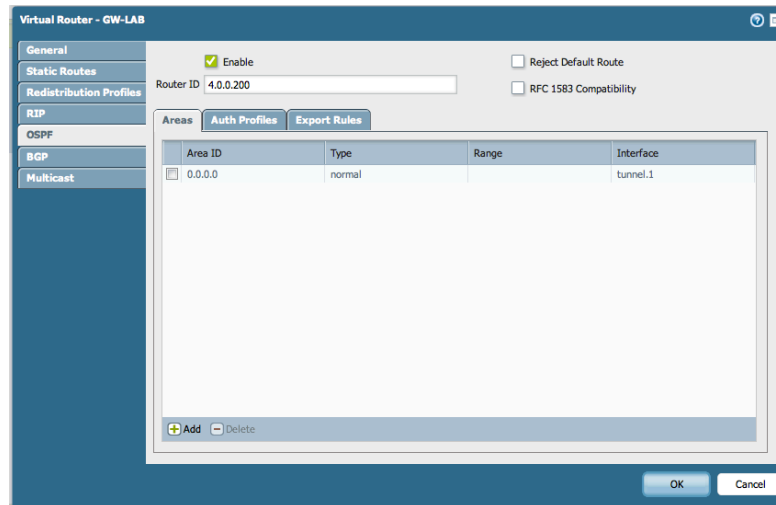
The screenshot shows the configuration for a Tunnel Interface in Palo Alto Networks. The interface name is 'tunnel.1'. The IP address is set to 4.0.0.200, which is noted as being needed for dynamic routing and tunnel monitoring. The MTU is configured to 1427. The interface is assigned to the 'GW-Defecto' virtual router and the 'VPN-Cisco' security zone. The management profile is set to 'Ping' and the netflow profile is 'None'.

Note 2: Remember also to remove the static routing entry pointing to 192.168.1.0/24, if you have one. This was the entry that we created to check the connectivity through the IPSec tunnel before, but now we want to reach our destination via OSPF learnt entries.

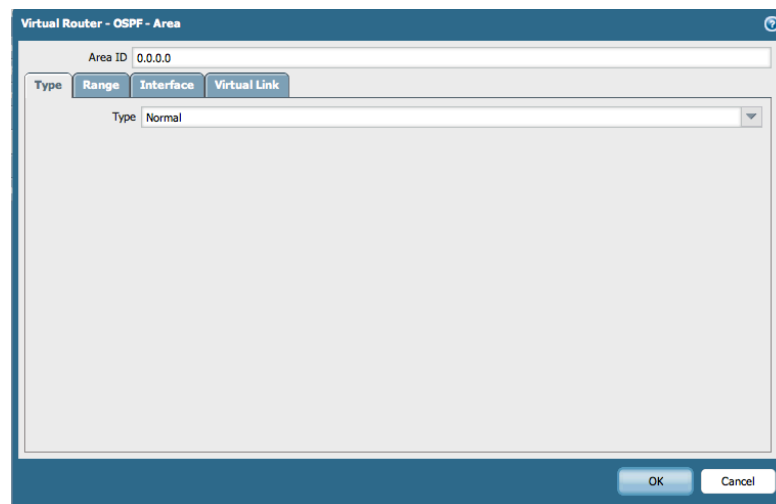
Setting up Palo Alto Networks OSPF configuration

The configuration is also pretty simple. We will cover how to configure it via GUI and also the xml output file, like we did with the IPSec tunnel:

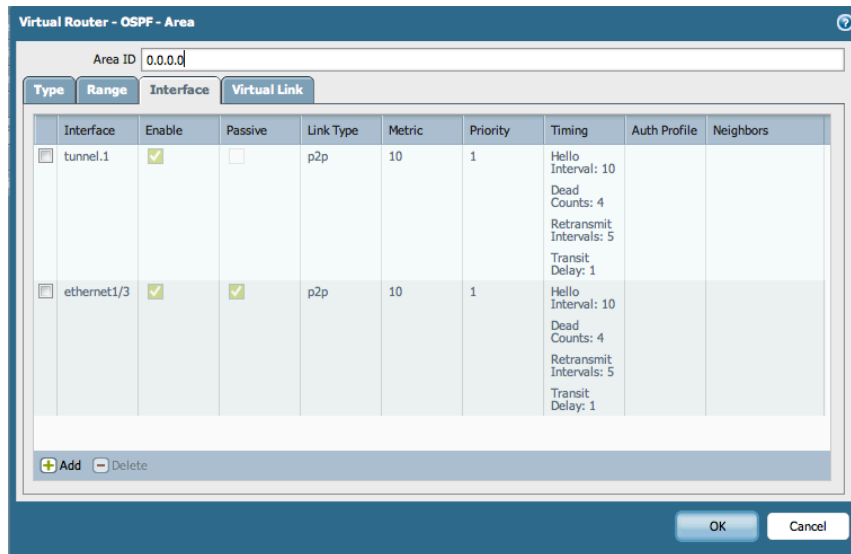
1. Go to the **Network** tab and then to the **Virtual Routers** area.
2. Edit your existing virtual router (GW-LAB in our example).
3. Remove the static route to 192.168.10.0/24, if you have one.
4. Configure basic OSPF settings. You need to enable the protocol and assign a Router ID (4.0.0.200 in our example).



5. Create area 0. You add area ID 0 (defined as 0.0.0.0), configured as Type “Normal”.



6. **Configure the interfaces in area 0.** You add the tunnel interface (tunnel.1), configured as p2p (point to point) type. You add also the rest of interfaces that you wish will participate in the OSPF configuration, so that these networks will be learnt by the peer via OSPF as internal LSA's – Type 1 (you can add them as “normal” or “passive” interfaces to the area 0 configuration). In our lab we have added interface Ethernet 1/3, which holds network 192.168.1.0/24, where the Workstation of this side is placed. You don't need to change any other parameter.



7. **Commit your config.**

Following we show also the XML config for the virtual router configuration, including the OSPF definition:

```

virtual-router {
  GW-LAB {
    interface [ ethernet1/1 ethernet1/13 ethernet1/2 ethernet1/3 tunnel tunnel.1 ];
    routing-table {
      ip {
        static-route {
          Defecto {
            nexthop {
              ip-address 192.168.57.1;
            }
            metric 10;
            destination 0.0.0.0/0;
          }
        }
      }
    }
  }
  protocol {
    rip {
      enable no;
      reject-default-route no;
      allow-redirect-default-route no;
    }
    ospf {
      enable yes;
      reject-default-route no;
      allow-redirect-default-route no;
      rfc1583 no;
      area {
        0.0.0.0 {

```

```

interface {
  tunnel.1 {
    enable yes;
    passive no;
    metric 10;
    priority 1;
    hello-interval 10;
    dead-counts 4;
    retransmit-interval 5;
    transit-delay 1;
    link-type {
      p2p ;
    }
  }
  ethernet1/3 {
    enable yes;
    passive yes;
    metric 10;
    priority 1;
    hello-interval 10;
    dead-counts 4;
    retransmit-interval 5;
    transit-delay 1;
    link-type {
      p2p ;
    }
  }
}
type {
  normal ;
}
}
}
router-id 4.0.0.200;
}
bgp {
  enable no;
  reject-default-route no;
  routing-options {
    as-format 2-byte;
    med {
      deterministic-med-comparison no;
      always-compare-med no;
    }
    graceful-restart {
      enable no;
      stale-route-time 120;
      local-restart-time 120;
      max-peer-restart-time 120;
    }
    aggregate {
      aggregate-med no;
    }
    default-local-preference 100;
  }
  dampening-profile {
    default {
      cutoff 1.25;
      reuse 0.5;
      max-hold-time 900;
      decay-half-life-reachable 300;
      decay-half-life-unreachable 900;
      enable yes;
    }
  }
  allow-redist-default-route no;
  install-route no;
}
}

```

```

admin-dists {
  static 10;
  ospf-int 30;
  ospf-ext 110;
  ibgp 200;
  ebgp 20;
  rip 120;
}
}
}

```

Verifying the OSPF configuration

To finish with our lab we will show how to check that the config is correct, both on Palo Alto Networks and Cisco. As with the IPsec chapter we will provide only some tips or ideas.

In the firewall you can review the routing logs under the system tabs. You can do it via GUI or via CLI. Following we show an example output for CLI:

```

admin@PA-2050> show log system subtype equal routing direction equal backward
Time          Severity Subtype Object EventID ID Description
=====
2012/01/04 16:03:41 info          routing GW-LAB routed- 0 OSPF full adjacency established with
neighbor. interface tunnel.1, neighbor router ID 4.0.0.254, neighbor IP address 4.0.0.254.

```

You can also review the routing table and check that indeed the firewall is learning the expected routes via OSPF through tunnel.1 interface:

```

admin@PA-2050> show routing route type ospf

```

```

flags: A:active, ?:loose, C:connect, H:host, S:static, ~:internal, R:rip, O:ospf, B:bgp,
Oi:ospf intra-area, Oo:ospf inter-area, Ol:ospf ext-type-1, O2:ospf ext-type-2

```

```

VIRTUAL ROUTER: GW-LAB (id 3)

```

```

=====
destination    nexthop          metric flags      age  interface      next-AS
4.0.0.0/24     4.0.0.254       11121 A Oi           3591 tunnel.1
4.0.0.254/32   4.0.0.254       10    A Oi           3591 tunnel.1
192.168.1.0/24 0.0.0.0         10     Oi           3596 ethernet1/3
192.168.10.0/24 4.0.0.254      11     A Oi           3591 tunnel.1
total routes shown: 4

```

If debugging is needed, you can enable OSPF pcap, through the following command:

```

admin@PA-2050> debug routing pcap ospf on

```

And then you can review the OSPF messages using:

```

admin@PA-2050> debug routing pcap ospf view

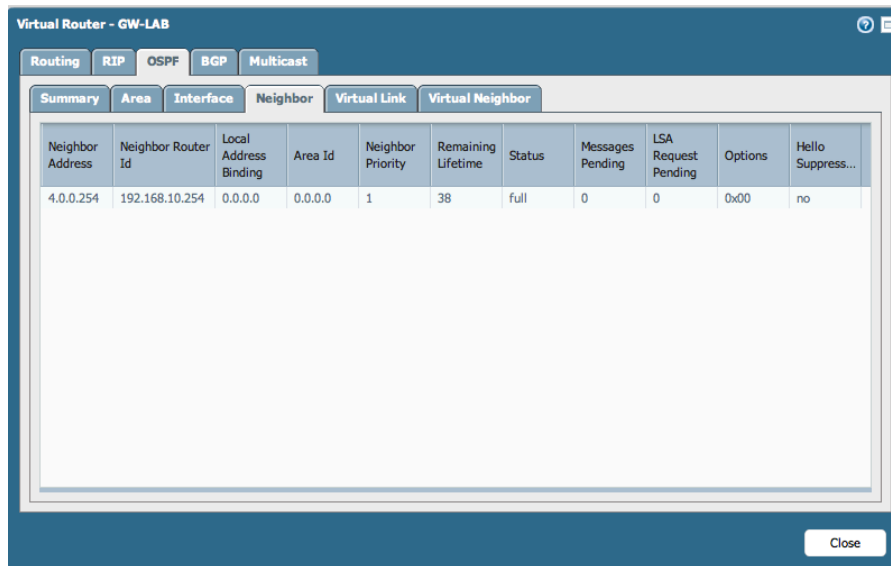
```

```

21:54:17.901832 IP 4.0.0.254 > 224.0.0.5: OSPFv2, Database Description, length: 44
21:54:17.910759 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Database Description, length: 32
21:54:19.087632 IP 4.0.0.254 > 224.0.0.5: OSPFv2, Hello, length: 60
21:54:22.378517 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Hello, length: 48
21:54:22.918691 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Database Description, length: 32
21:54:24.481365 IP 4.0.0.254 > 224.0.0.5: OSPFv2, Database Description, length: 44
21:54:24.483937 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Database Description, length: 52
21:54:30.899167 IP 4.0.0.254 > 224.0.0.5: OSPFv2, Database Description, length: 44
21:54:30.901459 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Database Description, length: 52
21:54:32.139662 IP 4.0.0.254 > 224.0.0.5: OSPFv2, Hello, length: 60
21:54:32.388558 IP 4.0.0.200 > 224.0.0.5: OSPFv2, Hello, length: 48

```


Through the GUI you can also go to the virtual router definition and go over “More Runtime Stats”. You will find there the full routing table; you can also check specific information on OSPF, like shows the following screenshot:



On the Cisco router you can also review the routing table. The entries that appear marked with “O” are learnt via OSPF through VTI Tunnel0:

```
R-jesusd#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

Gateway of last resort is not set

   3.0.0.0/24 is subnetted, 1 subnets
C       3.0.0.0 is directly connected, FastEthernet1/0
   4.0.0.0/24 is subnetted, 1 subnets
C       4.0.0.0 is directly connected, Tunnel0
C       192.168.10.0/24 is directly connected, FastEthernet0/0
O       192.168.1.0/24 [110/1121] via 4.0.0.200, 00:18:37, Tunnel0
R-jesusd#
```

If you wish you can also check the status of the OSPF instance, the neighbors, ... Following we show a couple of options of the “show ip ospf” command in Cisco:

```
R-jesusd#sh ip ospf database

        OSPF Router with ID (192.168.10.254) (Process ID 1)

        Router Link States (Area 0)

Link ID        ADV Router      Age          Seq#          Checksum Link count
4.0.0.200      4.0.0.200      1182        0x80000006   0x00F40C  3
192.168.10.254 192.168.10.254 1695        0x80000004   0x00CAF4  3
R-jesusd#
```

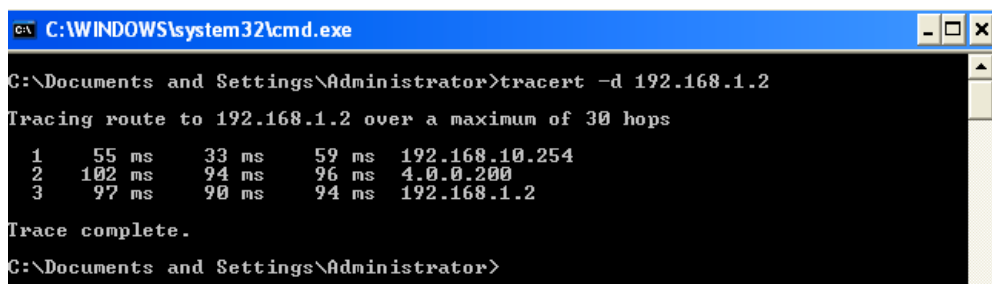
```
R-jesusd#sh ip ospf neighbor
```

```
Neighbor ID      Pri   State           Dead Time   Address      Interface
4.0.0.200        0    FULL/ -         00:00:35   4.0.0.200   Tunnel0
R-jesusd#
```

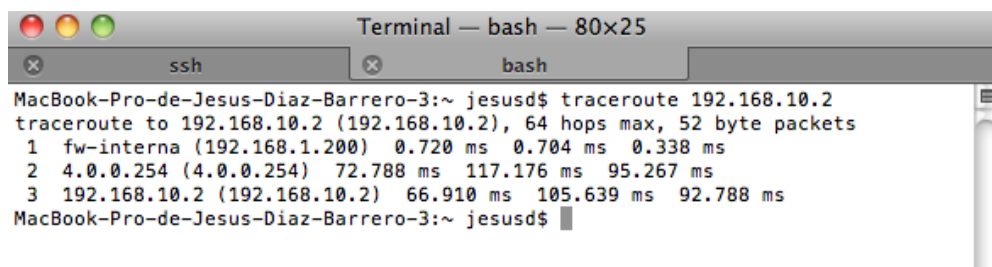
If you need to debug the OSPF messages, you can use the “debug ip ospf events command”. Following you have an example output for this debugging option:

```
R-jesusd#debug ip ospf events
OSPF events debugging is on
R-jesusd#
*Jan 4 16:25:14.411: OSPF: Send hello to 224.0.0.5 area 0 on Tunnel0 from 4.0.0.254
*Jan 4 16:25:21.763: OSPF: Rcv hello from 4.0.0.200 area 0 from Tunnel0 4.0.0.200
*Jan 4 16:25:21.763: OSPF: End of hello processing
*Jan 4 16:25:22.683: OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 192.168.10.254
*Jan 4 16:25:23.435: OSPF: Rcv hello from 4.0.0.200 area 0 from Tunnel0 4.0.0.200
*Jan 4 16:25:23.435: OSPF: End of hello processing
*Jan 4 16:25:24.411: OSPF: Send hello to 224.0.0.5 area 0 on Tunnel0 from 4.0.0.254
*Jan 4 16:25:24.575: OSPF: Rcv DBD from 4.0.0.200 on Tunnel0 seq 0x1A9FF50 opt 0x42 flag 0x7 len 32 mtu 1500 state INIT
*Jan 4 16:25:24.575: OSPF: 2 Way Communication to 4.0.0.200 on Tunnel0, state 2WAY
*Jan 4 16:25:24.575: OSPF: Send DBD to 4.0.0.200 on Tunnel0 seq 0xE25 opt 0x52 flag 0x7 len 32
*Jan 4 16:25:24.579: OSPF: First DBD and we are not SLAVE
*Jan 4 16:25:24.699: OSPF: Rcv DBD from 4.0.0.200 on Tunnel0 seq 0xE25 opt 0x42 flag 0x0 len 112 mtu 1500 state EXSTART
*Jan 4 16:25:24.699: OSPF: NBR Negotiation Done. We are the MASTER
*Jan 4 16:25:24.699: OSPF: Send DBD to 4.0.0.200 on Tunnel0 seq 0xE26 opt 0x52 flag 0x3 len 132
*Jan 4 16:25:24.895: OSPF: Rcv DBD from 4.0.0.200 on Tunnel0 seq 0xE26 opt 0x42 flag 0x0 len 32 mtu 1500 state EXCHANGE
*Jan 4 16:25:24.895: OSPF: Send DBD to 4.0.0.200 on Tunnel0 seq 0xE27 opt 0x52 flag 0x1 len 32
*Jan 4 16:25:25.199: OSPF: Rcv DBD from 4.0.0.200 on Tunnel0 seq 0xE27 opt 0x42 flag 0x0 len 32 mtu 1500 state EXCHANGE
*Jan 4 16:25:25.199: OSPF: Exchange Done with 4.0.0.200 on Tunnel0
*Jan 4 16:25:25.199: OSPF: Synchronized with 4.0.0.200 on Tunnel0, state FULL
*Jan 4 16:25:25.203: %OSPF-5-ADJCHG: Process 1, Nbr 4.0.0.200 on Tunnel0 from LOADING to FULL, Loading Done
```

Finally, and obviously, you should be able to reach the private networks on each side from the workstations, without any need to set up static routes nor in the firewall nor in the router. We provide example screenshots from a *traceroute* on both end workstations, which demonstrates that the traffic is indeed going through the expected hops, inside the IPSec tunnel:



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>tracert -d 192.168.1.2
Tracing route to 192.168.1.2 over a maximum of 30 hops
  0  55 ms  33 ms  59 ms  192.168.10.254
  1  102 ms  94 ms  96 ms  4.0.0.200
  2  97 ms  90 ms  94 ms  192.168.1.2
Trace complete.
C:\Documents and Settings\Administrator>
```



```
Terminal — bash — 80x25
MacBook-Pro-de-Jesus-Diaz-Barrero-3:~ jesusd$ traceroute 192.168.10.2
traceroute to 192.168.10.2 (192.168.10.2), 64 hops max, 52 byte packets
 0  fw-interna (192.168.1.200)  0.720 ms  0.704 ms  0.338 ms
 1  4.0.0.254 (4.0.0.254)  72.788 ms  117.176 ms  95.267 ms
 2  192.168.10.2 (192.168.10.2)  66.910 ms  105.639 ms  92.788 ms
MacBook-Pro-de-Jesus-Diaz-Barrero-3:~ jesusd$
```

Conclusions

We have shown the flexibility and simplicity in the approach of using routed IPSec VPNs with virtual tunnel interfaces, which Palo Alto Networks next generation firewalls offer, as well as Cisco routers (through the utilization of IPSec VTIs). We have set up a lab to demonstrate the interoperability between both vendors and also that this network architecture allows running dynamic routing protocols, like OSPF, inside the IPSec tunnel. This set up should be valid also for other dynamic routing protocols, such as RIP or BGP, allowing the customers to use their required routing protocols in a safe manner (inside an IPSec tunnel).

Revision History

Date	Revision	Comment
2012/01/04	A	First version of the document
2012/01/06	B	Change in the OSPF configuration to make congruent announcements (LSA's Type 1 in both sides, instead of Type1 vs Type5)
2012/01/09	C	Change format of the document