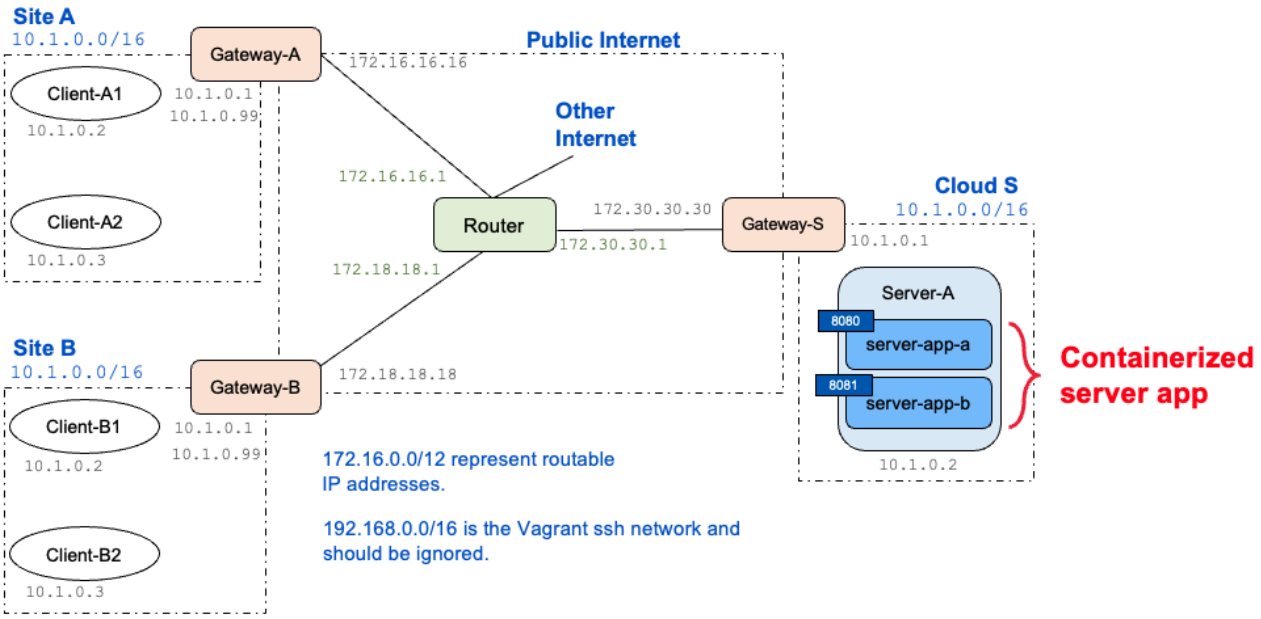
Network Security IPsec VPN Project

Group 36: YANG Je-Ruei, XU Zehui
GitHub Repository: https://github.com/jerry871002/ipsec-vpn

Table of Content

1 Network Setup



- The server on each site is moved to the cloud as a containerized application.
- From the clients' perspective, everything is unchanged. The clients are sending requests to the same IP address as it was.
- Routing decisions are made in the gateways.
    - For example, when Gateway-A or Gateway-B receives client requests, they forward them to gateway-s.
    - Gateway-S then determine which server app to forward to requests based on their source addresses.
- IPsec VPN between customer sites (Gateway-A or Gateway-B) and the cloud site (Gateway-S).

2 Server Scalability Solution

Our solution to address scalability is containerization. The servers are now containerized applications. Whenever a new customer site is added, we create a container and add routing rules in Gateway-S. The customer-side setup is mostly identical for all customers (see `scripts/site_a_gateway.sh` and `scripts/site_b_gateway.sh`), which reduces the need for manual configuration and enables better scalability.

3 Authentication Credentials Configuration

The sites authenticate themselves with **public-key authentication**. Thus, this setup involves a public-key infrastructure (PKI). Currently, the keys and certificates are generated on the host machine and directly copy them into the VMs. However, this process is not applicable in real-world scenarios. The proper process should be having some separate CA machines,

and then each site requests a certificate from a CA (CA can authenticate the site with a pre-shared key in this step to ensure this is a legitimate customer and not an attacker).

4 Deployment Process

To set up this testbed, please follow the following steps:

1. Install the dependency on the host machine
   `sudo apt install strongswan-pki`
2. Run `scripts/gen_cert.sh` at the **root directory** of the project. Make sure the `pki` folder is generated.
3. Create the VMs, the VMs that have a certificate are `gateway-a`, `gateway-b`, and `gateway-s`.
   `vagrant up [vm]`
4. We're done!

5 Commands/Scripts Used for the Configuration

5.1 Gateway-A

```
#!/usr/bin/env bash

##### Install swanctl #####

apt-get update -y
apt-get install -y strongswan-swanctl charon-systemd

##### Setup iptables rules #####

# NAT traffic going to the internet
route add default gw 172.16.16.1
iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE

# Redirect traffic that was going to the local server to the cloud
server
# DNAT stands for Destination NAT
ip addr add 10.1.0.99/16 dev enp0s9
iptables -t nat -A PREROUTING -p tcp -d 10.1.0.99 --dport 8080 -j DNAT
--to-destination 172.30.30.30:8080

# Save the iptables rules
iptables-save > /etc/iptables/rules.v4
ip6tables-save > /etc/iptables/rules.v6

##### Setup strongSwan site-to-site VPN #####
```

```
# Root CA certificate
cp /vagrant/pki/cert/rootCACert.pem /etc/swanctl/x509ca/

# Site A certificate
cp /vagrant/pki/cert/siteACert.pem /etc/swanctl/x509/

# Site A private key
cp /vagrant/pki/key/siteAKey.pem /etc/swanctl/private/

# swanctl configurations
cp /vagrant/config/swanctl-site-a.conf /etc/swanctl/swanctl.conf

# load the certificates and private keys into the charon daemon
swanctl --load-creds

# load the connections defined in swanctl.conf
swanctl --load-conns
```

5.2 Gateway-B

Mostly the same as Gateway-A except for the certificate and private key files. The swanctl configuration line is also slightly different, but as you will see in section 6, the contents are again mostly identical.

5.3 Gateway-S

```
#!/usr/bin/env bash

##### Install swanctl #####

apt-get update -y
apt-get install -y strongswan-swanctl charon-systemd

## Traffic going to the internet
route add default gw 172.30.30.1

# Redirect traffic to the dedicated server
iptables -t nat -A PREROUTING -p tcp -s 172.16.16.16 --dport 8080 -j
DNAT --to-destination 10.1.0.2:8080
iptables -t nat -A PREROUTING -p tcp -s 172.16.18.18 --dport 8080 -j
DNAT --to-destination 10.1.0.2:8081

## Save the iptables rules
iptables-save > /etc/iptables/rules.v4
```

```
ip6tables-save > /etc/iptables/rules.v6

# Root CA certificate
cp /vagrant/pki/cert/rootCACert.pem /etc/swanctl/x509ca/

# Cloud S certificate
cp /vagrant/pki/cert/cloudSCert.pem /etc/swanctl/x509/

# Cloud S private key
cp /vagrant/pki/key/cloudSKey.pem /etc/swanctl/private/

# swanctl configurations
cp /vagrant/config/swanctl-cloud-s.conf /etc/swanctl/swanctl.conf

# load the certificates and private keys into the charon daemon
swanctl --load-creds

# load the connections defined in swanctl.conf
swanctl --load-conns
```

5.4 Cloud Server

The Dockerfile in this script defines a simple nodejs image that installs all the npm dependencies and starts the application.

```
#!/usr/bin/env bash

# Install docker
apt-get update -y
apt-get install -y docker.io

## Build app image
cp /vagrant/scripts/Dockerfile /home/vagrant/server_app
cd /home/vagrant/server_app
docker build -t server-app:v1 .

## Traffic going to the internet
route add default gw 10.1.0.1

## Save the iptables rules
iptables-save > /etc/iptables/rules.v4
ip6tables-save > /etc/iptables/rules.v6

# Run app
docker run -p 8080:8080 -d --name site-a-server server-app:v1
docker run -p 8081:8080 -d --name site-b-server server-app:v1
```

6 Modified Configuration Files

6.1 Gateway-A

6.1.1 /etc/swanctl/swanctl.conf

```
connections {
    site-a-to-cloud {
        local_addrs = 172.16.16.16
        remote_addrs = 172.30.30.30
        local {
            auth = pubkey
            certs = siteACert.pem
            id = "C=FI, O=Network Security, CN=Site A"
        }
        remote {
            auth = pubkey
            id = "C=FI, O=Network Security, CN=Cloud S"
        }
        children {
            net-net {
                local_ts = 172.16.16.16/32
                remote_ts = 172.30.30.30/32

                esp_proposals = default

                # installs a trap policy which triggers the tunnel
                # as soon as matching traffic has been detected
                start_action = trap
            }
        }
        # IKEv2
        version = 2
        proposals = default
    }
}
```

6.1.2 /etc/iptables/rules.v4

```
# Generated by iptables-save v1.6.1 on Tue May 16 13:23:05 2023
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
```

```
-A PREROUTING -d 10.1.0.99/32 -p tcp -m tcp --dport 8080 -j DNAT
--to-destination 172.30.30.30:8080
-A POSTROUTING -o enp0s8 -j MASQUERADE
COMMIT
# Completed on Tue May 16 13:23:05 2023
# Generated by iptables-save v1.6.1 on Tue May 16 13:23:05 2023
*filter
:INPUT ACCEPT [6074:14434268]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [3801:261128]
COMMIT
# Completed on Tue May 16 13:23:05 2023
```

6.2 Gateway-B

The contents of `/etc/iptables/rules.v4` and `/etc/swanctl/swanctl.conf` are the same as Gateway-A except for the local credentials and the local address.

6.3 Gateway-S

6.3.1 /etc/swanctl/swanctl.conf

```
connections {
    cloud-to-site-a {
        local_addrs = 172.30.30.30
        remote_addrs = 172.16.16.16
        local {
            auth = pubkey
            certs = cloudSCert.pem
            id = "C=FI, O=Network Security, CN=Cloud S"
        }
        remote {
            auth = pubkey
            id = "C=FI, O=Network Security, CN=Site A"
        }
        children {
            net-net {
                local_ts = 172.30.30.30/32
                remote_ts = 172.16.16.16/32

                esp_proposals = default

                # installs a trap policy which triggers the tunnel
                # as soon as matching traffic has been detected
                start_action = trap
```

```
            }
        }
        # IKEv2
        version = 2
        proposals = default
    }

    cloud-to-site-b {
        local_addrs = 172.30.30.30
        remote_addrs = 172.18.18.18
        local {
            auth = pubkey
            certs = cloudSCert.pem
            id = "C=FI, O=Network Security, CN=Cloud S"
        }
        remote {
            auth = pubkey
            id = "C=FI, O=Network Security, CN=Site B"
        }
        children {
            net-net {
                local_ts = 172.30.30.30/32
                remote_ts = 172.18.18.18/32

                esp_proposals = default

                # installs a trap policy which triggers the tunnel
                # as soon as matching traffic has been detected
                start_action = trap
            }
        }
        # IKEv2
        version = 2
        proposals = default
    }
}
```

6.3.2 /etc/iptables/rules.v4

```
# Generated by iptables-save v1.6.1 on Tue May 16 13:47:29 2023
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -s 172.16.16.16/32 -p tcp -m tcp --dport 8080 -j DNAT
```

```
--to-destination 10.1.0.2:8080
-A PREROUTING -s 172.16.18.18/32 -p tcp -m tcp --dport 8080 -j DNAT
--to-destination 10.1.0.2:8081
COMMIT
# Completed on Tue May 16 13:47:29 2023
# Generated by iptables-save v1.6.1 on Tue May 16 13:47:29 2023
*filter
:INPUT ACCEPT [4023:14347003]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [2730:219015]
COMMIT
# Completed on Tue May 16 13:47:29 2023
```

7 Output of Debugging Commands

7.1 Gateway-A

The enp0s9 interface connects to the local network, and the enp0s8 interface connects to the Internet.

```
vagrant@gateway-a:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:59:f1:88:50:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.112.15/24 brd 192.168.112.255 scope global dynamic enp0s3
       valid_lft 55176sec preferred_lft 55176sec
    inet6 fe80::59:f1ff:fe88:50de/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c9:ed:20 brd ff:ff:ff:ff:ff:ff
    inet 172.16.16.16/24 brd 172.16.16.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec9:ed20/64 scope link
       valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1e:87:69 brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.1/16 brd 10.1.255.255 scope global enp0s9
       valid_lft forever preferred_lft forever
    inet 10.1.0.99/16 scope global secondary enp0s9
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe1e:8769/64 scope link
       valid_lft forever preferred_lft forever
```

```
vagrant@gateway-a:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.16.1     0.0.0.0         UG    0      0        0 enp0s8
0.0.0.0         192.168.112.2   0.0.0.0         UG    100    0        0 enp0s3
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0        0 enp0s9
172.16.16.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.112.0   0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
192.168.112.2   0.0.0.0         255.255.255.255 UH    100    0        0 enp0s3
```

`swanctl --list-sas` display the current Security Associations (SAs) established by the swanctl daemon, which is quite similar to what `ipsec statusall` does.

```
vagrant@gateway-a:~$ sudo swanctl --list-sas
site-a-to-cloud: #6, ESTABLISHED, IKEv2, 656383a0f6bbcd7c_i 34388214a33acb96_r*
  local  'C=FI, O=Network Security, CN=Site A' @ 172.16.16.16[4500]
  remote 'C=FI, O=Network Security, CN=Cloud S' @ 172.30.30.30[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
  established 7250s ago, rekeying in 6945s
  net-net: #24, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 446s ago, rekeying in 2822s, expires in 3514s
    in  c73c628b,      0 bytes,     0 packets
    out c1289f38,      0 bytes,     0 packets
    local  172.16.16.16/32
    remote 172.30.30.30/32
```

IPsec policies on the system.

```
vagrant@gateway-a:~$ sudo ip xfrm policy
src 172.16.16.16/32 dst 172.30.30.30/32
        dir out priority 367231
        tmpl src 172.16.16.16 dst 172.30.30.30
                proto esp spi 0xc1289f38 reqid 1 mode tunnel
src 172.30.30.30/32 dst 172.16.16.16/32
        dir fwd priority 367231
        tmpl src 172.30.30.30 dst 172.16.16.16
                proto esp reqid 1 mode tunnel
src 172.30.30.30/32 dst 172.16.16.16/32
        dir in priority 367231
        tmpl src 172.30.30.30 dst 172.16.16.16
                proto esp reqid 1 mode tunnel
```

IPsec SAs on the system.

```
vagrant@gateway-a:~$ sudo ip xfrm state
src 172.16.16.16 dst 172.30.30.30
        proto esp spi 0xc1289f38 reqid 1 mode tunnel
        replay-window 0 flag af-unspec
        auth-trunc hmac(sha256) 0xe0764e2314770fcd427e500c92f9fd1ec18ca893226be7dec292b4a1d84ccc1e 128
        enc cbc(aes) 0x64f43f022b8fe296a8be8b3fc72ddc81
        anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
src 172.30.30.30 dst 172.16.16.16
        proto esp spi 0xc73c628b reqid 1 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha256) 0x587d50f159b4b3d2eb30a15f00c397a9a5afd84460d9313a7d504bd451aa250b 128
        enc cbc(aes) 0xae6527c26f1eeaea3f7a7b832c4c2922
        anti-replay context: seq 0x0, oseq 0x0, bitmap 0x00000000
```

```
vagrant@gateway-a:~$ sudo iptables-save
# Generated by iptables-save v1.6.1 on Wed May 17 10:25:27 2023
*nat
:PREROUTING ACCEPT [132:10900]
:INPUT ACCEPT [11:3640]
:OUTPUT ACCEPT [70:13535]
:POSTROUTING ACCEPT [38:3379]
-A PREROUTING -d 10.1.0.99/32 -p tcp -m tcp --dport 8080 -j DNAT --to-destination 172.30.30.30:8080
-A POSTROUTING -o enp0s8 -j MASQUERADE
COMMIT
# Completed on Wed May 17 10:25:27 2023
# Generated by iptables-save v1.6.1 on Wed May 17 10:25:27 2023
*filter
:INPUT ACCEPT [27580:68220575]
:FORWARD ACCEPT [89073:172031397]
:OUTPUT ACCEPT [17451:904552]
COMMIT
# Completed on Wed May 17 10:25:27 2023
```

7.3 Gateway-B

In reference to the previous sections, Gateway-B is very similar to Gateway-A, therefore I will not include repetitive screenshots.

7.3 Gateway-S

The enp0s9 interface connects to the local network, and the enp0s8 interface connects to the Internet.

```
vagrant@gateway-s:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:59:f1:88:50:de brd ff:ff:ff:ff:ff:ff
    inet 192.168.120.15/24 brd 192.168.120.255 scope global dynamic enp0s3
       valid_lft 86065sec preferred_lft 86065sec
    inet6 fe80::59:f1ff:fe88:50de/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:95:fe:47 brd ff:ff:ff:ff:ff:ff
    inet 172.30.30.30/24 brd 172.30.30.255 scope global enp0s8
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe95:fe47/64 scope link
       valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c2:f1:4a brd ff:ff:ff:ff:ff:ff
    inet 10.1.0.1/16 brd 10.1.255.255 scope global enp0s9
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec2:f14a/64 scope link
       valid_lft forever preferred_lft forever
```

```
vagrant@gateway-s:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.30.30.1     0.0.0.0         UG    0      0        0 enp0s8
0.0.0.0         192.168.120.2   0.0.0.0         UG    100    0        0 enp0s3
10.1.0.0        0.0.0.0         255.255.0.0     U     0      0        0 enp0s9
172.30.30.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.120.0   0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
192.168.120.2   0.0.0.0         255.255.255.255 UH    100    0        0 enp0s3
```

`swanctl --list-sas` display the current Security Associations (SAs) established by the swanctl daemon

```
vagrant@gateway-s:~$ sudo swanctl --list-sas
cloud-to-site-b: #2, ESTABLISHED, IKEv2, 88376b0a8f656536_i 13235cdecb23c46a_r*
  local  'C=FI, O=Network Security, CN=Cloud S' @ 172.30.30.30[4500]
  remote 'C=FI, O=Network Security, CN=Site B' @ 172.18.18.18[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
  established 72s ago, rekeying in 12889s
  net-net: #5, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 72s ago, rekeying in 3380s, expires in 3888s
    in  ca0373db,    420 bytes,     5 packets,    35s ago
    out cd5ac2ad,    420 bytes,     5 packets,    35s ago
    local  172.30.30.30/32
    remote 172.18.18.18/32
cloud-to-site-a: #1, ESTABLISHED, IKEv2, b0f6fc2ceab72620_i 2dabda5aa29631dc_r*
  local  'C=FI, O=Network Security, CN=Cloud S' @ 172.30.30.30[4500]
  remote 'C=FI, O=Network Security, CN=Site A' @ 172.16.16.16[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_AES128_XCBC/ECP_256
  established 1527s ago, rekeying in 12534s
  net-net: #3, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA2_256_128
    installed 1527s ago, rekeying in 1828s, expires in 2434s
    in  c2ae2db1, 335980 bytes,  3961 packets,   105s ago
    out c12b8fe6, 429662 bytes,  3831 packets,   105s ago
    local  172.30.30.30/32
    remote 172.16.16.16/32
```

IPsec policies on the system.

```
vagrant@gateway-s:~$ sudo ip xfrm policy
src 172.30.30.30/32 dst 172.18.18.18/32
        dir out priority 367231
        tmpl src 172.30.30.30 dst 172.18.18.18
                proto esp spi 0xcd5ac2ad reqid 1 mode tunnel
src 172.18.18.18/32 dst 172.30.30.30/32
        dir fwd priority 367231
        tmpl src 172.18.18.18 dst 172.30.30.30
                proto esp reqid 1 mode tunnel
src 172.18.18.18/32 dst 172.30.30.30/32
        dir in priority 367231
        tmpl src 172.18.18.18 dst 172.30.30.30
                proto esp reqid 1 mode tunnel
src 172.30.30.30/32 dst 172.16.16.16/32
        dir out priority 367231
        tmpl src 172.30.30.30 dst 172.16.16.16
                proto esp spi 0xc12b8fe6 reqid 1 mode tunnel
src 172.16.16.16/32 dst 172.30.30.30/32
        dir fwd priority 367231
        tmpl src 172.16.16.16 dst 172.30.30.30
                proto esp reqid 1 mode tunnel
src 172.16.16.16/32 dst 172.30.30.30/32
        dir in priority 367231
        tmpl src 172.16.16.16 dst 172.30.30.30
                proto esp reqid 1 mode tunnel
```

IPsec SAs on the system.

```
vagrant@gateway-s:~$ sudo ip xfrm state
src 172.30.30.30 dst 172.18.18.18
        proto esp spi 0xcd5ac2ad reqid 1 mode tunnel
        replay-window 0 flag af-unspec
        auth-trunc hmac(sha256) 0x2ff7eb9cff64ddf2427635c4e867001a47c3bb238c5d3d885915002074339c84 128
        enc cbc(aes) 0x49e9eb99e6480d2eaed568a0c94ab570
        anti-replay context: seq 0x0, oseq 0x5, bitmap 0x00000000
src 172.18.18.18 dst 172.30.30.30
        proto esp spi 0xca0373db reqid 1 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha256) 0xc9455c845c87aa05f2df6f470c361f3e1f3c779458abb1356789cb6a386bb1e9 128
        enc cbc(aes) 0x92b8c23b6d34efb99e8774d4bc259e16
        anti-replay context: seq 0x5, oseq 0x0, bitmap 0x0000001f
src 172.30.30.30 dst 172.16.16.16
        proto esp spi 0xc12b8fe6 reqid 1 mode tunnel
        replay-window 0 flag af-unspec
        auth-trunc hmac(sha256) 0x85d98f74ec5d5a756c42fbe123b680a45d6217b3e3ed8f1f35e62cef670381cc 128
        enc cbc(aes) 0xed7fe6a23290e880f988ede95fbca7db
        anti-replay context: seq 0x0, oseq 0xef7, bitmap 0x00000000
src 172.16.16.16 dst 172.30.30.30
        proto esp spi 0xc2ae2db1 reqid 1 mode tunnel
        replay-window 32 flag af-unspec
        auth-trunc hmac(sha256) 0x61d13d615b558b4d0976d3c1a29841045e94f4b229f19e026901592f703f3b4a 128
        enc cbc(aes) 0x04056e55543a4f4e7ebe1c12dcc3f555
        anti-replay context: seq 0xf79, oseq 0x0, bitmap 0xffffffff
```