Configure ASR1000 Encryption over OTV Unicast

Contents

[Introduction](#)
[Prerequisites](#)
[Requirements](#)
[Components Used](#)
[Background Information](#)
[Configure](#)
[Network Diagram](#)
[Configurations](#)
[Verify](#)
[Troubleshoot](#)

Introduction

This document describes the basic set of configurations that are used to bring up Overlay Transport Virtualization (OTV) with IPSec encryption. Encryption over OTV does not require any additional configurations from the OTV end. You just need to understand how OTV and IPSEC co-exists.

In order to add encryption over OTV, you need to add an Encapsulating Security Payload (ESP) header on top of OTV PDU. You can achieve encryption on the ASR1000 Edge Devices (ED) through two ways: (i) IPSec (ii) GETVPN.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- ASR1000 routers for Edge Devices (ED)
- Core (ISP Cloud)
- Catalyst 2960 switches as the access switch on either site

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

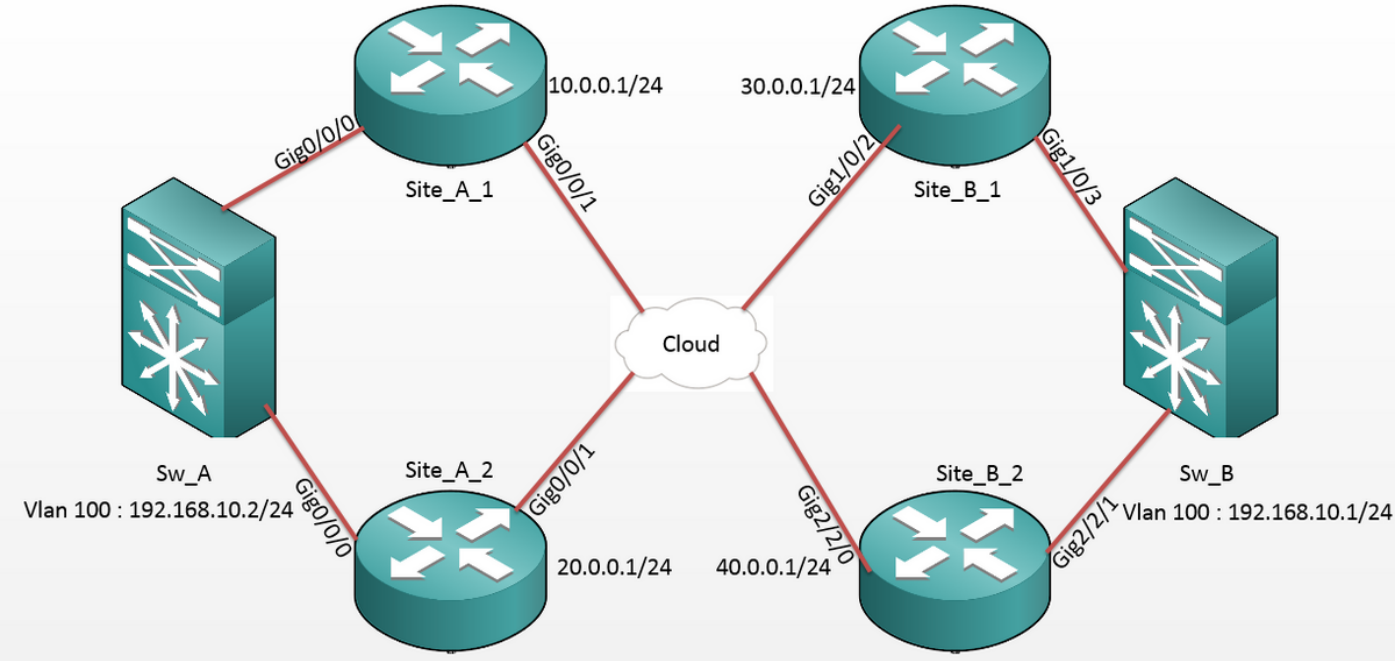
Basic functionality and configurations of OTV are presumed to be known by the users of this document.

You can also follow these documents for the same:

- OTV Unicast Configuration
- OTV Multicast Configuration

Configure

Network Diagram



Configurations

Site A: ED Configurations:

```
Site_A_1#show run
Building configuration...
otv site bridge-domain 99
!
otv site-identifier 0000.0000.0001
crypto isakmp policy 10
 hash md5
 authentication pre-share
```

```
Site_A_2#show run
Building configuration...
otv site bridge-domain 99
!
otv site-identifier 0000.0000.0001
crypto isakmp policy 10
 hash md5
 authentication pre-share
```

```
crypto isakmp key cisco address 30.0.0.1
crypto isakmp key cisco address 40.0.0.1
!
crypto ipsec transform-set tset esp-aes
esp-md5-hmac
 mode tunnel
!
crypto map cmap 1 ipsec-isakmp
 set peer 30.0.0.1
 set transform-set tset
 match address cryptoacl
crypto map cmap 3 ipsec-isakmp
 set peer 40.0.0.1
 set transform-set tset
 match address cryptoacl3
!
interface Overlay99
 no ip address
 otv join-interface GigabitEthernet0/0/1
 otv adjacency-server unicast-only
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet0/0/0
 no ip address
service instance 99 ethernet
 encapsulation dot1q 99
```

```
crypto isakmp key cisco address 30.0.0.1
crypto isakmp key cisco address 40.0.0.1
!
crypto ipsec transform-set tset esp-aes
esp-md5-hmac
 mode tunnel
!
crypto map cmap 2 ipsec-isakmp
 set peer 30.0.0.1
 set transform-set tset
 match address cryptoacl2
crypto map cmap 3 ipsec-isakmp
 set peer 40.0.0.1
 set transform-set tset
 match address cryptoacl3
!
interface Overlay99
 no ip address
 otv join-interface GigabitEthernet0/0/1
 otv use-adjacency-server 10.0.0.1 30.0.0.1
unicast-only
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet0/0/0
 no ip address
service instance 99 ethernet
```

```
 bridge-domain 99
 !
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet0/0/1
 ip address 10.0.0.1 255.255.255.0
 crypto map cmap
!
ip access-list extended cryptoacl
 permit gre host 10.0.0.1 host 30.0.0.1
ip access-list extended cryptoacl3
 permit gre host 10.0.0.1 host 40.0.0.1
```

```
 encapsulation dot1q 99
 bridge-domain 99
 !
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet0/0/1
 ip address 20.0.0.1 255.255.255.0
 crypto map cmap
!
ip access-list extended cryptoacl2
 permit gre host 20.0.0.1 host 30.0.0.1
ip access-list extended cryptoacl3
 permit gre host 20.0.0.1 host 40.0.0.1
```

Site B: ED Configurations:

```
Site_B_1#sh run
Building configuration...
otv site bridge-domain 99
!
otv site-identifier 0000.0000.0002
crypto isakmp policy 10
 hash md5
 authentication pre-share
crypto isakmp key cisco address 10.0.0.1
crypto isakmp key cisco address 20.0.0.1
```

```
Site_B_2#sh run
Building configuration...
otv site bridge-domain 99
!
otv site-identifier 0000.0000.0002
crypto isakmp policy 10
 hash md5
 authentication pre-share
crypto isakmp key cisco address 10.0.0.1
crypto isakmp key cisco address 20.0.0.1
```

```
!
crypto ipsec transform-set tset esp-aes esp-md5-hmac
 mode tunnel
!
crypto map cmap 1 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set tset
 match address cryptoacl
crypto map cmap 2 ipsec-isakmp
 set peer 20.0.0.1
 set transform-set tset
 match address cryptoacl2
!
interface Overlay99
 no ip address
 otv join-interface GigabitEthernet1/0/2
 otv use-adjacency-server 10.0.0.1 unicast-only
 otv adjacency-server unicast-only
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet1/0/3
 no ip address
service instance 99 ethernet
 encapsulation dot1q 99
```

```
!
crypto ipsec transform-set tset esp-aes esp-md5-hmac
 mode tunnel
!
crypto map cmap 1 ipsec-isakmp
 set peer 10.0.0.1
 set transform-set tset
 match address cryptoacl
crypto map cmap 2 ipsec-isakmp
 set peer 20.0.0.1
 set transform-set tset
 match address cryptoacl2
!
interface Overlay99
 no ip address
 otv join-interface GigabitEthernet2/2/0
 otv use-adjacency-server 10.0.0.1 30.0.0.1 unicast-only
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet2/2/1
 no ip address
 service instance 99 ethernet
 encapsulation dot1q 99
 bridge-domain 99
```

```
 bridge-domain 99
 !
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet1/0/2
 ip address 30.0.0.1 255.255.255.0
crypto map cmap
!
ip access-list extended cryptoacl
 permit gre host 30.0.0.1 host 10.0.0.1
ip access-list extended cryptoacl2
 permit gre host 30.0.0.1 host 20.0.0.1
```

```
 !
 service instance 100 ethernet
 encapsulation dot1q 100
 bridge-domain 100
 !
 service instance 101 ethernet
 encapsulation dot1q 101
 bridge-domain 101
 !
!
interface GigabitEthernet2/2/0
 ip address 40.0.0.1 255.255.255.0
 crypto map cmap
!
ip access-list extended cryptoacl
 permit gre host 40.0.0.1 host 10.0.0.1
ip access-list extended cryptoacl2
 permit gre host 40.0.0.1 host 20.0.0.1
```

Verify

Use this section in order to confirm that your configuration works properly.

1. Check if the MAC address of the internal VLAN host (in this case the SVI on the 2960 catalyst switch’s) have been learnt on the OTV route tables.
2. Check if the crypto encap’s and decap’s are performed for the Overlay (OTV traffic) traffic.

Once the OTV comes up after you configure the crypto map on the join interface, check the active forwarder for the local VLAN’s (in this case VLAN 100 and 101). This shows that Site_A_1 and Site_B_2 are the active forwarders for the even VLAN’s since you will test the traffic encryption for pings initiated from VLAN 100 on Site A to VLAN 100 on Site B:

```
Site_A_1#show otv vlan

Key:  SI - Service Instance, NA - Non AED, NFC - Not Forward Capable.

Overlay 99 VLAN Configuration Information
```

```
 Inst VLAN BD   Auth ED             State               Site If(s)
 0    100  100  *Site_A_1           active              Gi0/0/0:SI100
 0    101  101   Site_A_2           inactive(NA)        Gi0/0/0:SI101
 0    200  200  *Site_A_1           active              Gi0/0/0:SI200
 0    201  201   Site_A_2           inactive(NA)        Gi0/0/0:SI201
 Total VLAN(s): 4

Site_B_2#show otv vlan

Key:  SI - Service Instance, NA - Non AED, NFC - Not Forward Capable.


Overlay 99 VLAN Configuration Information

 Inst VLAN BD   Auth ED             State               Site If(s)
0    100  100  *Site_B_2           active              Gi2/2/1:SI100
 0    101  101   Site_B_1           inactive(NA)        Gi2/2/1:SI101
 0    200  200  *Site_B_2           active              Gi2/2/1:SI200
 0    201  201   Site_B_1           inactive(NA)        Gi2/2/1:SI201
 Total VLAN(s): 4
```

In order to check if the packets indeed get encapsulated and decapsulated on either ED, you should check if the IPSec session is active and the counter values in the crypto sessions in order to confirm that the packets are indeed getting encrypted and decrypted. In order to check if the IPSec session is active, since it becomes active only if any traffic flows through, check the output of **show crypto isakmp sa**. Here, only the outputs for the active forwarders are checked, but this should show the active status on all the ED's for OTV over encryption to work.

```
Site_B_2#show otv vlan

Key:  SI - Service Instance, NA - Non AED, NFC - Not Forward Capable.


Overlay 99 VLAN Configuration Information

 Inst VLAN BD   Auth ED             State               Site If(s)
0    100  100  *Site_B_2           active              Gi2/2/1:SI100
 0    101  101   Site_B_1           inactive(NA)        Gi2/2/1:SI101
 0    200  200  *Site_B_2           active              Gi2/2/1:SI200
 0    201  201   Site_B_1           inactive(NA)        Gi2/2/1:SI201
 Total VLAN(s): 4
```

Now, in order to confirm if the packets get encrypted and decrypted, you first need to know what to expect in the outputs of **show crypto session detail**. So, when you initiate the ICMP echo packet from the Sw_A switch towards the Sw_B, this is expected:

- While the ICMP echo leaves from the Site_A_1 ED which is the active forwarder for the VLAN 100, it will have to encapsulate the OTV payload (ICMP Echo + MPLS + GRE)
- Then once the ICMP echo reaches the Site_B_2 ED which is the active forwarder for VLAN 100, it would have to decapsulate the OTV payload (ICMP Echo + MPLS + GRE)
- Now, once the Site_B_2 ED receives the ICMP Echo Reply from Sw_B, it would have to again encapsulate the OTV payload (ICMP Echo + MPLS + GRE)
- And once the ICMP Echo Reply reaches the Site_A_1 ED, I would again have to **again decapsulate** the OTV payload (ICMP Echo + MPLS + GRE)

After the successful pings from Sw_A to Sw_B, expect to see an increment of 5 counters under "enc" and “dec” section of the **show crypto session detail** output on both the active forwarder ED’s.

Now, check the same from the ED's:

```
Site_A_1(config-if)#do show crypto session detail | section enc

K - Keepalives, N - NAT-traversal, T - cTCP encapsulation

       Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 4608000/3345

       Outbound: #pkts enc'ed 10 drop 0 life (KB/Sec) 4607998/3291 <<<< 10 counter before ping

Site_A_1(config-if)#do show crypto session detail | section dec

       Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 4608000/3343

       Inbound:  #pkts dec'ed 18 drop 0 life (KB/Sec) 4607997/3289 <<<< 18 counter before ping

Site_B_2(config-if)#do show crypto session detail | section enc

K - Keepalives, N - NAT-traversal, T - cTCP encapsulation

       Outbound: #pkts enc'ed 18 drop 0 life (KB/Sec) 4607997/3295 <<<< 18 counter before ping

       Outbound: #pkts enc'ed 9 drop 0 life (KB/Sec) 4607999/3295

Site_B_2(config-if)#do show crypto session detail | section dec

       Inbound:  #pkts dec'ed 10 drop 0 life (KB/Sec) 4607998/3293 <<<< 10 counter before ping

       Inbound:  #pkts dec'ed 1 drop 0 life (KB/Sec) 4607999/3293

Site_B_2(config-if)#do show crypto session detail | section dec

       Inbound:  #pkts dec'ed 10 drop 0 life (KB/Sec) 4607998/3293 <<<< 10 counter before ping

       Inbound:  #pkts dec'ed 1 drop 0 life (KB/Sec) 4607999/3293

Site_A_1(config-if)#do show crypto session detail | section enc

K - Keepalives, N - NAT-traversal, T - cTCP encapsulation

       Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) 4608000/3339

       Outbound: #pkts enc'ed 15 drop 0 life (KB/Sec) 4607997/3284 <<<< 15 counter after ping
(After ICMP Echo)
```

```
Site_A_1(config-if)#do show crypto session detail | section dec

       Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) 4608000/3338

       Inbound:  #pkts dec'ed 23 drop 0 life (KB/Sec) 4607997/3283 <<<< 23 counter after ping
(After ICMP Echo Reply)

Site_B_2(config-if)#do show crypto session detail | section enc

K - Keepalives, N - NAT-traversal, T - cTCP encapsulation

       Outbound: #pkts enc'ed 23 drop 0 life (KB/Sec) 4607997/3282 <<<< 23 counter after ping
(After ICMP Echo Reply)

       Outbound: #pkts enc'ed 9 drop 0 life (KB/Sec) 4607999/3282

Site_B_2(config-if)#do show crypto session detail | section dec

       Inbound:  #pkts dec'ed 15 drop 0 life (KB/Sec) 4607997/3281 <<<< 15 counter after ping
(After ICMP Echo)

       Inbound:  #pkts dec'ed 1 drop 0 life (KB/Sec) 4607999/3281
```

This configuration guide is able to convey the required configuration details with the use of IPSec for the Unicast core dual homed setup.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.