



Orbit MCR IPsec VPN

Device to Device Tunnel Using X.509 RSA Certificates

Introduction

This document describes how to setup a device-to-device IPsec VPN using two Orbit MCR devices over the cellular network. This setup allows an end-device connected to the first Orbit (say via LAN) to securely talk to the end-device connected to the second Orbit.

The Orbit devices can be managed from the back-office using SSH, NETCONF via the static IP address assigned to their cellular interface. The end-devices can be accessed from back-office using port forwarding (destination NAT) configuration on each Orbit.

Scope

This bulletin is intended for network administrators and end users who want to enable secure communication between end-devices. The bulletin describes how to use the Orbit command line interface (CLI) to set up a device-to-device IPsec VPN tunnel using X.509 certificates between two Orbits. Please refer to Orbit MCR technical manual for additional details on how to access the Orbit CLI.

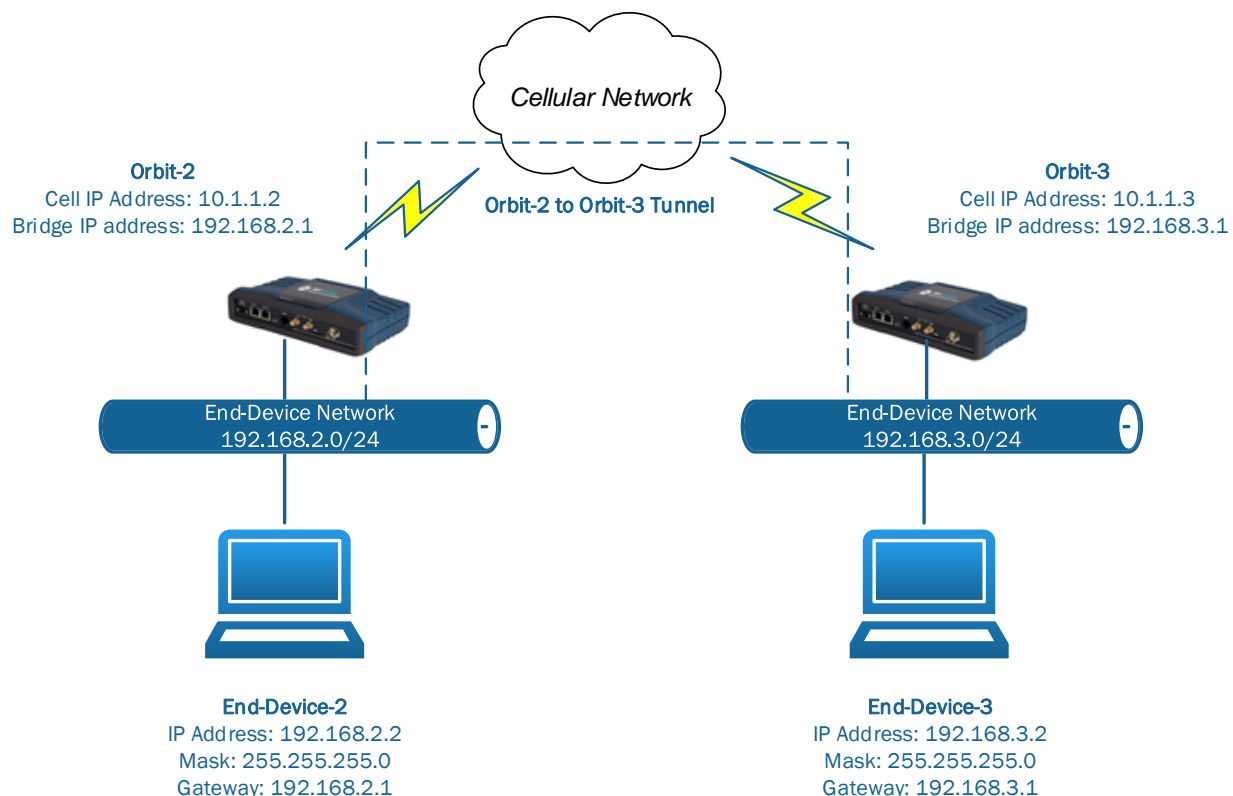
Firmware Compatibility

This bulletin is applicable to Orbit MCR devices running firmware version 1.5.1 or greater.

Terms

CLI Command Line Interface
VPN Virtual Private Network

Network Setup



For Orbit-2 to Orbit-3 tunnel, orbit-2 is the “initiator” and Orbit-3 is the responder.

General Notes

We assume that we are starting with factory default configuration on each Orbit and that the Cell interface has already been configured (Please refer to Orbit technical manual on how to configure Cell interface).

You can copy the commands listed in this bulletin, paste them into a text file and change any details necessary to match the your network configuration (**highlighted** below), and then copy and paste the commands into the CLI (after you have entered “configuration” mode as shown below) for quick and convenient configuration.

NOTE: Do not forget to turn off CLI auto-wizard as shown below. Otherwise, copy-paste of commands might not work properly.

(none) login: admin
Password:

admin connected from 127.0.0.1 using console on (none)

admin@(none) 21:12:32> **set autowizard false**

admin@(none) 21:12:32> config
Entering configuration mode private
[ok][2014-05-06 21:12:33]

[edit]
admin@(none) 21:12:33%

Orbit-2 Configuration – Loading certificates, setting up NTP

Load client key, client certificate and root CA certificate

NOTE: Please refer to another Orbit application bulletin on how to generate keys and certificates.

We assume that a PC (192.168.1.2) running TFTP server is connected to Orbit's LAN port.

1. Load private key from file "id2.key" giving it an identity of "ID2".

```
request pki get-priv-key key-identity ID2 filename id2.key manual-file-server { tftp { address 192.168.1.2 } }
```

2. Load certificate from file "id2.cert" giving it an identity of "ID2".

```
request pki get-client-cert clientcert-identity ID2 file { filename id2.cert manual-file-server { tftp { address 192.168.1.2 } } }
```

3. Load root CA certificate from file "ca.cert" giving it an identity of "TEST-CA".

```
request pki get-ca-cert cacert-identity TEST-CA file { filename ca.cert manual-file-server { tftp { address 192.168.1.2 } } }
```

4. Validate that the client certificate has been loaded on the unit.

```
run show pki client-certs
```

```
CERT  
IDENTITY
```

```
-----  
ID2
```

5. Validate that the client private key has been loaded on the unit.

```
run show pki private-keys
```

```
KEY    KEY  
IDENTITY LENGTH KEY DATE TIME  
-----  
ID2    4096  2014-05-06T15:45:22Z
```

6. Validate that the CA certificate has been loaded on the unit.

```
run show pki ca-certs
```

```
CACERT  
IDENTITY  
-----  
TEST-CA
```

Configure system to obtain time via NTP

1. Enable NTP to obtain system time from NTP server (10.1.1.10).

NOTE1: We assume that a NTP server with IP address 10.1.1.10 is available and accessible on the cellular network. The certificate based IPsec tunnel setup requires system time on Orbit to be synchronized with an NTP server, otherwise certificate validation and hence tunnel setup will fail.

```
set system ntp use-ntp true  
set system ntp ntp-server 10.1.1.10
```

NOTE2: In case the cellular plan allows internet access, you can use following configuration:

```
set system ntp use-ntp true  
set system ntp ntp-server 0.pool.ntp.org
```

```
set system ntp ntp-server 1.pool.ntp.org
set system ntp ntp-server 2.pool.ntp.org
set system ntp ntp-server 3.pool.ntp.org
```

2. Commit configuration

```
commit
```

Orbit-2 Configuration - IPsec tunnel to Orbit-3

Configure IPsec tunnel

1. Enable VPN service

```
set services vpn enabled true
```

2. Create IKE policy named "IKE-POLICY-CERT"

```
set services vpn ike policy IKE-POLICY-CERT auth-method pub-key
set services vpn ike policy IKE-POLICY-CERT pki cert-type rsa
set services vpn ike policy IKE-POLICY-CERT pki cert-id ID2
set services vpn ike policy IKE-POLICY-CERT pki key-id ID2
set services vpn ike policy IKE-POLICY-CERT pki ca-cert-id TEST-CA
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 encryption-
algo aes128-cbc
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 mac-algo
sha256-hmac
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 dh-group
dh14
```

3. Create IKE peer named "ORBIT3"

```
set services vpn ike peer ORBIT3 ike-policy IKE-POLICY-CERT
set services vpn ike peer ORBIT3 peer-endpoint address 10.1.1.3
set services vpn ike peer ORBIT3 role initiator
```

4. Create IPsec policy named "IPSEC-POLICY"

```
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 encryption-
algo aes128-cbc
```

```
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 mac-algo sha256-hmac
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 dh-group dh14
```

5. Create IPsec connection named “ORBIT3”

```
set services vpn ipsec connection ORBIT3 ike-peer ORBIT3
set services vpn ipsec connection ORBIT3 ipsec-policy IPSEC-POLICY
set services vpn ipsec connection ORBIT3 local-ip-subnet 192.168.2.0/24
set services vpn ipsec connection ORBIT3 remote-ip-subnets [ 192.168.3.0/24 ]
set services vpn ipsec connection ORBIT3 failure-retry-interval 1
```

Update firewall configuration from defaults

6. Prevent LAN network traffic from going out outside of the tunnel.

```
delete services firewall address-set LOCAL-NETS addresses
```

Update firewall configuration to allow IKE/IPsec traffic

7. Add rules to allow IKE, DNS, NTP traffic and IPsec (ESP) traffic into the Cell interface.

NOTE: The IN_UNTRUSTED filter has been applied by factory defaults on Cell interface in incoming direction.

```
set services firewall filter IN_UNTRUSTED rule 2 match protocol udp
set services firewall filter IN_UNTRUSTED rule 2 match src-port services [ dns ike ntp ]
set services firewall filter IN_UNTRUSTED rule 2 actions action accept
set services firewall filter IN_UNTRUSTED rule 3 match protocol esp
set services firewall filter IN_UNTRUSTED rule 3 actions action accept
```

Update firewall configuration to allow end-device traffic through the tunnel

8. Add rules to allow all traffic from end-device-3 subnet (192.168.3.0/24) to end-device-2 subnet (192.168.2.0/24) through the tunnel.

```
set services firewall filter IN_UNTRUSTED rule 4 match ipsec direction in
set services firewall filter IN_UNTRUSTED rule 4 match ipsec tunnel-src-address 10.1.1.3/32
set services firewall filter IN_UNTRUSTED rule 4 match ipsec tunnel-dst-address 10.1.1.2/32
set services firewall filter IN_UNTRUSTED rule 4 actions action accept
```

9. Add rules to allow all traffic from end-device-2 subnet (192.168.2.0/24) to end-device-3 subnet (192.168.3.0/24) through the tunnel.

```
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec direction out
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec tunnel-src-address 10.1.1.2/32
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec tunnel-dst-address 10.1.1.3/32
set services firewall filter OUT_UNTRUSTED rule 2 actions action accept
```

Update network interface configuration from defaults

10. Disable DHCP server

```
set services dhcp enabled false
```

11. Update Bridge interface IP address to 192.168.2.1/24

```
delete interfaces interface Bridge ipv4
set interfaces interface Bridge ipv4 address 192.168.2.1 prefix-length 24
```

12. Remove NAT from Cell interface.

```
delete interfaces interface Cell nat
```

Commit Configuration

13. Commit configuration

```
commit
```

Orbit-3 Configuration - Loading certificates, setting up NTP

NOTE: From here on, we simply provide the configuration commands. Please refer to earlier sections in this bulletin to understand the intent of these commands.

Load client key, client certificate and root CA certificate

```
request pki get-priv-key key-identity ID3 filename id3.key manual-file-server { tftp { address 192.168.1.2 } }
```

```
request pki get-client-cert clientcert-identity ID3 file { filename id3.cert manual-file-server { tftp { address 192.168.1.2 } } }
```

```
request pki get-ca-cert cacert-identity TEST-CA file { filename ca.cert manual-file-server { tftp {
address 192.168.1.2 } } }
```

Configure system to obtain time via NTP

NOTE1: We assume that a NTP server with IP address 10.1.1.10 is available and accessible on the cellular network. The certificate based IPsec tunnel setup requires system time on Orbit to be synchronized with an NTP server, otherwise certificate validation and hence tunnel setup will fail.

```
set system ntp use-ntp true
set system ntp ntp-server 10.1.1.10
```

NOTE2: In case the cellular plan allows internet access, you can use following configuration:

```
set system ntp use-ntp true
set system ntp ntp-server 0.pool.ntp.org
set system ntp ntp-server 1.pool.ntp.org
set system ntp ntp-server 2.pool.ntp.org
set system ntp ntp-server 3.pool.ntp.org
```

```
commit
```

Orbit-3 Configuration – IPsec tunnel from Orbit-2

Configure IPsec tunnel

```
set services vpn enabled true

set services vpn ike policy IKE-POLICY-CERT auth-method pub-key
set services vpn ike policy IKE-POLICY-CERT pki cert-type rsa
set services vpn ike policy IKE-POLICY-CERT pki cert-id ID3
set services vpn ike policy IKE-POLICY-CERT pki key-id ID3
set services vpn ike policy IKE-POLICY-CERT pki ca-cert-id TEST-CA
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 encryption-
algo aes128-cbc
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 mac-algo
sha256-hmac
set services vpn ike policy IKE-POLICY-CERT ciphersuite AES128-CBC-SHA256-DH14 dh-group
dh14
```



```
set services vpn ike peer ORBIT2 ike-policy IKE-POLICY-CERT
set services vpn ike peer ORBIT2 peer-endpoint address 10.1.1.2
set services vpn ike peer ORBIT2 role responder
```

```
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 encryption-
algo aes128-cbc
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 mac-algo
sha256-hmac
set services vpn ipsec policy IPSEC-POLICY ciphersuite AES128-CBC-SHA256-DH14 dh-group
dh14
```

```
set services vpn ipsec connection ORBIT2 ike-peer ORBIT2
set services vpn ipsec connection ORBIT2 ipsec-policy IPSEC-POLICY
set services vpn ipsec connection ORBIT2 local-ip-subnet 192.168.3.0/24
set services vpn ipsec connection ORBIT2 remote-ip-subnets [ 192.168.2.0/24 ]
```

Update firewall configuration from defaults

```
delete services firewall address-set LOCAL-NETS addresses
```

Update firewall configuration to allow IKE/IPsec traffic

NOTE: The IN_UNTRUSTED filter has been applied by factory defaults on Cell interface in incoming direction.

```
set services firewall filter IN_UNTRUSTED rule 2 match protocol udp
set services firewall filter IN_UNTRUSTED rule 2 match src-port services [ dns ike ntp ]
set services firewall filter IN_UNTRUSTED rule 2 actions action accept
set services firewall filter IN_UNTRUSTED rule 3 match protocol esp
set services firewall filter IN_UNTRUSTED rule 3 actions action accept
```

Update firewall configuration to allow end-device traffic through the tunnel

```
set services firewall filter IN_UNTRUSTED rule 4 match ipsec direction in
set services firewall filter IN_UNTRUSTED rule 4 match ipsec tunnel-src-address 10.1.1.2/32
set services firewall filter IN_UNTRUSTED rule 4 match ipsec tunnel-dst-address 10.1.1.3/32
set services firewall filter IN_UNTRUSTED rule 4 actions action accept
```

```
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec direction out
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec tunnel-src-address 10.1.1.3/32
set services firewall filter OUT_UNTRUSTED rule 2 match ipsec tunnel-dst-address 10.1.1.2/32
```

```
set services firewall filter OUT_UNTRUSTED rule 2 actions action accept
```

Update network interface configuration from defaults

```
set services dhcp enabled false
delete interfaces interface Bridge ipv4
set interfaces interface Bridge ipv4 address 192.168.3.1 prefix-length 24
delete interfaces interface Cell nat
```

Commit Configuration

```
commit
```

Testing tunnel (to Orbit-3) on Orbit2

NOTE: “monitor” command and “ping” command with src-address option is available in firmware version 1.5.8 or greater. To test connectivity in prior firmware versions, please ping end-devices from each other instead.

1. Validate that tunnel is connected.

```
admin@(none) 13:53:48% run show services vpn
services vpn ipsec ipsec-status connection ORBIT3
state          connected
failure-reason  none
last-timestamp 2014-05-06T19:28:49+00:00
ima-evaluation  none
ima-recommendation none
```

2. Validate that you can pass traffic by pinging Orbit-3 bridge IP address (192.168.3.1) from Orbit-2 (using its Bridge IP address i.e. 192.168.2.1 as source address)

```
admin@(none) 13:47:58% run ping 192.168.3.1 src-address 192.168.2.1
PING 192.168.3.1 (192.168.3.1) from 192.168.2.1 : 56(84) bytes of data.
64 bytes from 192.168.3.1: icmp_req=1 ttl=64 time=5570 ms
64 bytes from 192.168.3.1: icmp_req=2 ttl=64 time=4573 ms
64 bytes from 192.168.3.1: icmp_req=3 ttl=64 time=3583 ms
```

Testing tunnel (from Orbit-2) on Orbit3

1. Validate that tunnel is connected.

```
admin@(none) 13:53:48% run show services vpn
services vpn ipsec ipsec-status connection ORBIT2
state          connected
failure-reason none
last-timestamp 2014-05-06T19:28:49+00:00
ima-evaluation none
ima-recommendation none
```

2. Validate that you can pass traffic by pinging Orbit-2 bridge IP address (192.168.2.1) from Orbit-3 (using its Bridge IP address i.e. 192.168.3.1 as source address)

```
admin@(none) 13:59:06% run ping 192.168.2.1 src-address 192.168.3.1
PING 192.168.2.1 (192.168.2.1) from 192.168.3.1 : 56(84) bytes of data.
64 bytes from 192.168.2.1: icmp_req=1 ttl=64 time=5048 ms
64 bytes from 192.168.2.1: icmp_req=2 ttl=64 time=4058 ms
64 bytes from 192.168.2.1: icmp_req=3 ttl=64 time=3058 ms
64 bytes from 192.168.2.1: icmp_req=4 ttl=64 time=2079 ms
64 bytes from 192.168.2.1: icmp_req=5 ttl=64 time=1079 ms
64 bytes from 192.168.2.1: icmp_req=6 ttl=64 time=250 ms
```

You should also be able to ping end-device-3 from end-device-2 and vice versa.

Troubleshooting

The following are common reasons for VPN connection failure:

1. Invalid certificate or keys loaded on the device.
2. Time not synchronized on the device. Note that after cell connection is established, device can take few minutes to sync time from NTP server. VPN connection will not succeed until time is synchronized.
3. Mismatch in ciphersuites configured for IKE policy on device and the peer.
4. Mismatch in ciphersuites configured for IPsec policy on device and the peer.
5. Mismatch in remote and local IP subnets configured for IPsec connection on device and the peer.

You can monitor internal IPsec logs on orbit using following commands:

```
admin@(none) 17:13:26% run monitor start charon.log
[ok][2014-05-07 17:13:31]
```

...internal IPsec logs appear here...

```
admin@(none) 17:13:26% run monitor stop charon.log  
[ok][2014-05-07 17:13:31]
```

End of application bulletin.