Digital Energy
MDS

APPLICATION BULLETIN

NUMBER: AB14004_A
June 2014

MDS Orbit MCR Series

GE MDS, LLC. 175 Science Parkway, Rochester, NY 14620 USA
Phone +1 (585) 242-9600, FAX +1 (585) 242-9620 Web: www.gemds.com

Orbit MCR IPSec VPN

Multiple Tunnels Using X.509 RSA Certificates

Introduction

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

In addition, we describe how to setup a second IPsec tunnel from each orbit towards a VPN appliance in the back-office. This allows back-office network to access end-devices directly via their private IP addresses. Here, we use Orbit as a VPN appliance in the back-office network. Any other commercial appliance that supports X.509 certificated-based IPsec VPN can be used as well.

This bulletin builds on the configuration described in bulletin AB14003 revision A, “ORBIT-IPSEC-VPN-Device-to-Device-Using-Certificates”.

Scope

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

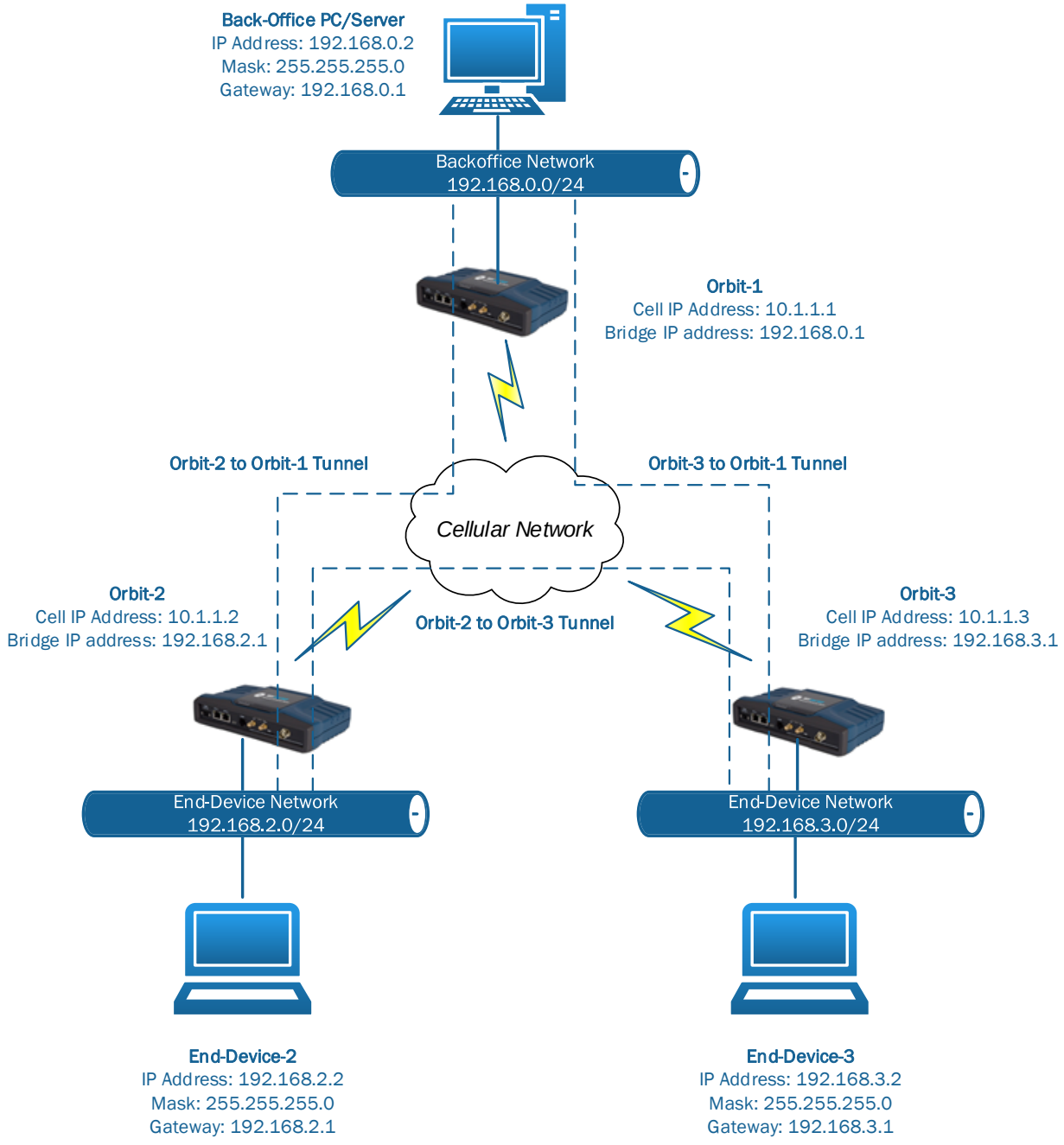
Firmware Compatibility

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

Terms

CLI Command Line Interface

Network Setup



For Orbit-2 to Orbit-3 tunnel, orbit-2 is the “initiator” and Orbit-3 is the responder.
For Orbit-2 to Orbit-1 tunnel, orbit-2 is the “initiator” and Orbit-1 is the responder.
For Orbit-3 to Orbit-1 tunnel, orbit-3 is the “initiator” and Orbit-1 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 - IPsec tunnel to Orbit-3

Please refer to application bulletin titled "IPsec-VPN-Device-to-Device-Using-Certificates" for information on how to load key, certificates on Orbit-2 and configure IPsec tunnel to Orbit3.

Oribit-3 Configuration - IPsec tunnel from Orbit-2

Please refer to application bulletin titled "IPsec-VPN-Device-to-Device-Using-Certificates" for information on how to load key, certificates on Orbit-3 and configure IPsec tunnel from Orbit2.

Oribit-2 Configuration - IPsec tunnel to Orbit-1

NOTE: We here assume that Orbit-2 has been previously configured as described earlier in this document.

Configure IPsec tunnel

```
set services vpn ike peer ORBIT1 ike-policy IKE-POLICY-CERT
set services vpn ike peer ORBIT1 peer-endpoint address 10.1.1.1
set services vpn ike peer ORBIT1 role initiator

set services vpn ipsec connection ORBIT1 ike-peer ORBIT1
set services vpn ipsec connection ORBIT1 ipsec-policy IPSEC-POLICY
set services vpn ipsec connection ORBIT1 local-ip-subnet 192.168.2.0/24
set services vpn ipsec connection ORBIT1 remote-ip-subnets [ 192.168.0.0/24 ]
```

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

```
set services firewall filter IN_UNTRUSTED rule 5 match ipsec direction in
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-src-address 10.1.1.1/32
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-dst-address 10.1.1.2/32
set services firewall filter IN_UNTRUSTED rule 5 actions action accept

set services firewall filter OUT_UNTRUSTED rule 3 match ipsec direction out
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-src-address 10.1.1.2/32
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-dst-address 10.1.1.1/32
set services firewall filter OUT_UNTRUSTED rule 3 actions action accept
```

Commit Configuration

```
commit
```

Oribit-3 Configuration- IPsec tunnel to Orbit-1

NOTE: We here assume that orbit-3 has been previously configured as described earlier in this document.

Configure IPsec tunnel

```
set services vpn ike peer ORBIT1 ike-policy IKE-POLICY-CERT
set services vpn ike peer ORBIT1 peer-endpoint address 10.1.1.1
```

set services vpn ike peer ORBIT1 role initiator

set services vpn ipsec connection ORBIT1 ike-peer ORBIT1
set services vpn ipsec connection ORBIT1 ipsec-policy IPSEC-POLICY
set services vpn ipsec connection ORBIT1 local-ip-subnet 192.168.3.0/24
set services vpn ipsec connection ORBIT1 remote-ip-subnets [192.168.0.0/24]

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

set services firewall filter IN_UNTRUSTED rule 5 match ipsec direction in
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-src-address 10.1.1.1/32
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-dst-address 10.1.1.3/32
set services firewall filter IN_UNTRUSTED rule 5 actions action accept

set services firewall filter OUT_UNTRUSTED rule 3 match ipsec direction out
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-src-address 10.1.1.3/32
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-dst-address 10.1.1.1/32
set services firewall filter OUT_UNTRUSTED rule 3 actions action accept

Commit Configuration

commit

Orbit-1 Configuration – Loading certificates, setting up NTP

Load client key, client certificate and root CA certificate

request pki get-priv-key key-identity ID1 filename id1.key manual-file-server { tftp { address 192.168.1.2 } }

request pki get-client-cert clientcert-identity ID1 file { filename id1.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-1 Configuration - IPsec tunnel from Orbit-2

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 ID1
set services vpn ike policy IKE-POLICY-CERT pki key-id ID1
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 192.168.0.0/24
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 10.1.1.1/32
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 10.1.1.1/32
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 192.168.0.1 prefix-length 24
delete interfaces interface Cell nat

Commit Configuration

commit

Orbit-1 Configuration - IPsec tunnel from Orbit-3

Configure IPsec tunnel

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 responder

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 192.168.0.0/24
set services vpn ipsec connection ORBIT3 remote-ip-subnets [192.168.3.0/24]

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

set services firewall filter IN_UNTRUSTED rule 5 match ipsec direction in
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-src-address 10.1.1.3/32
set services firewall filter IN_UNTRUSTED rule 5 match ipsec tunnel-dst-address 10.1.1.1/32
set services firewall filter IN_UNTRUSTED rule 5 actions action accept

set services firewall filter OUT_UNTRUSTED rule 3 match ipsec direction out
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-src-address 10.1.1.1/32
set services firewall filter OUT_UNTRUSTED rule 3 match ipsec tunnel-dst-address 10.1.1.3/32
set services firewall filter OUT_UNTRUSTED rule 3 actions action accept

Commit Configuration

commit

Testing tunnel (to Orbit-1) on Orbit-2

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) 20:54:26% run show services vpn
services vpn ipsec ipsec-status connection ORBIT3

```
 state           connected
 failure-reason     none
 last-timestamp     2014-05-07T19:14:42+00:00
 ima-evaluation     none
 ima-recommendation none
services vpn ipsec ipsec-status connection ORBIT1
 state           connected
 failure-reason     none
 last-timestamp     2014-05-07T19:14:32+00:00
 ima-evaluation     none
 ima-recommendation none
[ok][2014-05-07 20:54:36]
admin@(none) 20:54:36>
```

2. Validate that you can pass traffic by pinging Orbit-1 bridge IP address (192.168.0.1) from Orbit-2 (using its Bridge IP address i.e. 192.168.2.1 as source address)

```
admin@(none) 20:54:36% run ping 192.168.0.1 src-address 192.168.2.1
PING 192.168.0.1 (192.168.0.1) from 192.168.2.1 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_req=1 ttl=64 time=3342 ms
64 bytes from 192.168.0.1: icmp_req=2 ttl=64 time=2388 ms
64 bytes from 192.168.0.1: icmp_req=3 ttl=64 time=1448 ms
64 bytes from 192.168.0.1: icmp_req=4 ttl=64 time=648 ms
64 bytes from 192.168.0.1: icmp_req=5 ttl=64 time=558 ms
```

Testing tunnel (to Orbit-1) on Orbit-3

1. Validate that tunnel is connected.

```
admin@(none) 20:56:40% run show services vpn
services vpn ipsec ipsec-status connection ORBIT2
 state           connected
 failure-reason     none
 last-timestamp     1970-01-01T00:00:00+00:00
 ima-evaluation     none
 ima-recommendation none
services vpn ipsec ipsec-status connection ORBIT1
 state           connected
 failure-reason     none
 last-timestamp     2014-05-06T19:28:23+00:00
 ima-evaluation     none
```

```
 ima-recommendation none
[ok][2014-05-07 20:56:45]
```

3. Validate that you can pass traffic by pinging Orbit-1 bridge IP address (192.168.0.1) from Orbit-3 (using its Bridge IP address i.e. 192.168.3.1 as source address)

```
admin@(none) 20:56:45% run ping 192.168.0.1 src-address 192.168.3.1
PING 192.168.0.1 (192.168.0.1) from 192.168.3.1 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_req=1 ttl=64 time=5617 ms
64 bytes from 192.168.0.1: icmp_req=2 ttl=64 time=4654 ms
64 bytes from 192.168.0.1: icmp_req=3 ttl=64 time=3726 ms
64 bytes from 192.168.0.1: icmp_req=4 ttl=64 time=2746 ms
64 bytes from 192.168.0.1: icmp_req=5 ttl=64 time=1786 ms
64 bytes from 192.168.0.1: icmp_req=6 ttl=64 time=824 ms
```

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.