xylem
Let's Solve Water

USER GUIDE
AUG-10024-26

Unified Base Station Security (North American (NA))

FOR BASE STATION VERSION 2 RUNNING ON MODELS:
M400, R100NA, S100 / S50 (NORTH AMERICAN)

SENSUS
a xylem brand

Revision history

Rev No.	RNI Version	Date	Description
01	3.1 SP2	18-DEC-12	Updated for RNI 3.1 SP2.
05	3.2	13-JAN-15	Updated section on M400.
06	N/A	17-MAR-15	Misc. updates.
07	4.1	5-OCT-16	Updated for RNI 4.1.
08	4.2	31-JUL-17	Updated for RNI 4.2.
09	4.2.1	31-OCT-17	Updated for RNI 4.2 with Unified Base Station information.
10	4.3	12-FEB-18	Updated for RNI 4.3.
11	4.3.1	20-APR-18	Updated for RNI 4.3.1.
12	4.4.0	28-AUG-18	Updated for RNI 4.4.0.
13	4.4.1	26-OCT-18	Updated for RNI 4.4.1.
14	4.5.0	25-FEB-19	Updated for RNI 4.5.0.
15	4.5.1	29-APR-19	Updated for RNI 4.5.1.
16	4.6.0	27-AUG-19	Updated for RNI 4.6.0.
17	4.6.1	29-OCT-19	Updated for RNI 4.6.1.
18	4.7.0	26-FEB-20	Updated for RNI 4.7.0.
19	4.7.1	30-APR-20	Updated for RNI 4.7.1.
20	4.8	28-AUG-20	Updated for RNI 4.8.
21	4.9	10-JUN-21	Updated for RNI 4.9.
22	4.10	25-OCT-21	Updated for RNI 4.10.
23	4.11	06-JUN-22	Updated for RNI 4.11.
24	4.12	02-DEC-22	Updated for RNI 4.12.
25	4.13	16-JUN-23	Updated for RNI 4.13. Added information to differentiate between Unified 2.0 and 3.0. This document is for Unified 2.0 only.
26	4.14	15-NOV-23	Updated for RNI 4.14.

Copyright

Sensus
637 Davis Drive
Morrisville, NC 27560
1-800-638-3748
www.sensus.com

Document: RNI Base Station Security User Guide
Document Number: AUG-10024-26

Contents

Introduction 1

Scope 1

Background 1

Authentication 2

Initial access 2

Change the *tmadmin* account default password 2

Create a new sudo-enabled account 2

Add limited sudo privileges to non-root account 3

LDAP configuration 4

Firewall support 5

Firewall ssh_subnets (control of inbound SSH sessions) 5

Starting the firewall 6

OpenVPN 6

PKI to support OpenVPN 6

On the RNI NC server 7

RNI generates a self-signed cert/key for Base Station 7

Automatically starting the OpenVPN server 11

On the Unified Base Station 11

Logging and monitoring 15

SYSLOG configuration 15

SNMP configuration (v2c or v3) 16

SNMP v2c 17

SNMP v3 19

SNMP v2c and v3: Apply the configuration changes and start/restart the SNMP service 20

Appendix A – Selecting permitted SSL ciphers 21

Recommended server-side changes (server.conf) 21

Introduction

This document describes technical controls that can be used to increase the security posture of Sensus Base Stations/Transceivers and the FlexNet AMI system. This document covers commands available in the 2.x version of the base station software.

The most basic security requirement is unique and strong **root** and **user** passwords. The primary method for securing messages between a Regional Network Interface (RNI) **NC progs/tgblistener** process and the Base Stations/transceivers is to establish a secure traffic tunnel or VPN. Implementation of this secure path makes use of the OpenVPN package using RSA certificates for authentication, and the on-board host firewall (Linux IPTABLES).

In the following discussions, assume the convention Base Stations = Base Stations/Transceivers for brevity.

Scope

This document is intended for Base Station administrator who installs, configures, or operationally manages the Sensus Base Stations within an AMI system within North America.

Background

The RNI and Base Stations communicate through two TCP/IP ports. Each Base Station establishes a socket connection to both ports on the Network Controller (NC):

- 9600 (TCP): This port is used by the NC to receive Andorian (meter) messages and to receive Status Reports *from* the Base Stations.
- 9800 (TCP): This port is used to send commands *to* the Base Station from the NC (even though the Base Station technically assumes the role of a client in establishing the connection, it functions as a server to receive and process the two-way requests from the NC).

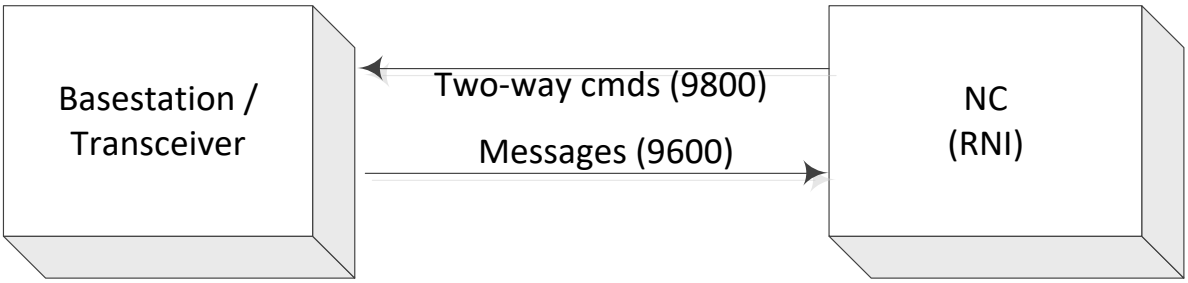

Figure 1: Base Station to NC TCP/IP connections

Andorian uses TCP ports 9600 and 9800 as a raw transit protocol.

The communications path (network and transport layer – IP) between a Base Station and the Network Controller is unencrypted by default. However, when FlexNet devices are enabled for encryption, most messages between the RNI and those devices are secured by AES-CCM at the FlexNet application layer.

Authentication

Authentication is provided by local passwords or network authentication techniques for Base Station administrators. This can involve Active Directory, as applied though LDAP. For OpenVPN, Sensus has documented the use of RSA X.509 digital certificates for TLS mutual Authentication.

Initial access

Once the Base Station has been commissioned, a default account named *tmadmin* is enabled for access.

Using a local or remote connection, users can connect to the Base Station using this account and the default password. The initial default login to the *tmadmin* account can be acquired on a new base station by calling the phone number supplied in the packing materials of the shipment.

This tmadmin account's default password should **ONLY** be used for the initial login, and then MUST BE changed and made unique on first login. This is detailed in the next few sections.

Following good security practices, the factory-default password MUST BE changed BEFORE the Base Station is ever allowed access to the internet.

Change the *tmadmin* account default password

Using the tmadmin login, issue the following command:

```
$ passwd
Enter existing password <Enter existing tmadmin password>
Enter new password <Enter a new complex password>
Enter new password again (to validate) <Re-Enter new password>
```

The default tmadmin password is now changed.

Warning: The root account does not have a password. Do not change/set the password of the root account as this weakens the overall security of the Base Station. Do not change the password using the **sudo** command.

The **tmadmin** account is local account and serves a special purpose. It is recommended that you do not use this account for remote SSH access (see the following section). It should be used for local access to the Base Station for field technicians either by local SSH on site, or the USB console connection. The **tmadmin** account should NOT be removed from the Base Station.

Create a new sudo-enabled account

It is not best practice to access the application servers with the root account. There is a big security posture improvement with the addition of non-root accounts for remote access, for audit purposes, to prevent human error, etc. Therefore, you should create a new sudo-enabled account to use as a remote network administrator of the Base Station.

Alternatively, you can use LDAP for network administrator accounts, instead of creating additional local accounts.

The recommendation is to create at least one other privileged account to be used for remote network administration of the Base Station over SSH. This keeps the built-in tmadmin account as a local ‘rescue account’ for recovery via the USB console port or locally on the LAN (Ethernet) on site.

Using the tmadmin login, then you can create a new admin account (for remote SSH).

>sudo useradd -U -G tmadmin,sudo,adm,systemd-journal -m -s /bin/bash **<username>**

>sudo passwd <**username**>

New password: **<Enter a new complex password>**

Retype new password: <**Re-Enter new password**>
passwd: password updated successfully

Add limited sudo privileges to non-root account

If operations and maintenance commands require root privileges, steps need to be taken to authorize other local accounts. Accounts listed in a file under /etc/sudoers.d/ can issue commands to elevate their privilege in order to execute commands at levels normally executed by a root user. This file can be customized to comply with local IT security policy, etc. One implementation in Linux to accomplish this is to add users to this file. Once a member of this file, the user simply enters “sudo” before the command statement requiring root privilege level, and they are prompted for their own password in order to minimize risk of passers-by executing privileged commands.

Note: Do not modify the Sensus configuration files in the /etc/sudoers.d/ directory. Please create a new file for customer-specific customizations.

Example 1:

This set of commands would allow that **specific account** to sudo run **ALL commands**.

cd **/etc/sudoers.d/**

vi <custom filename>

#Add the following line:

<account name> ALL=(ALL) ALL

Ctrl-x exits

*NOTE: Do not edit/modify the existing /etc/sudoers file

Example 2: Limited which commands (reboot and poweroff) that can be sudo executed by a specific user.

```
# This set of commands would restrict the specific account myadmin to only sudo run the
reboot and poweroff commands only.

cd /etc/sudoers.d/

vi <custom filename>

#Add the following line:

myadmin        ALL=(ALL)       /sbin/reboot, /sbin/poweroff

Ctrl-x exits

*NOTE: Do not edit/modify the existing /etc/sudoers file
```

LDAP configuration

You will need to have details for the following LDAP values:

- LDAP Bind: Organization Unit (OU), Distinguished Name (DN), Common Name (CN)
- LDAP Bind password
- URL of the LDAP server (including IP/FQDN) and protocol/port

On the Base Station:

```
>sudo tgbconfig ldap.base "ou=People,dc=example,dc=com"
# Set the Organizational Unit (OU) and Distinguished Name (DN)

>sudo tgbconfig ldap.binddn "cn=admin,dc=example,dc=com"
# set the Common Names (CN)

>sudo tgbconfig ldap.bindpw "<LDAP bind password>"
# set the ldap password

>sudo tgbconfig ldap.enabled 1
# enable the ldap config

>sudo tgbconfig ldap.uri "ldap://<FQDN/IP of your ldap server>"
# set the URI of the ldap server (includes the FQDN/IP and port/protocol)
```

To view the LDAP configuration and apply the configuration changes:

```
>sudo tgbconfig ldap
# To Verify successful entry of all parameters

>sudo tgbconfig-apply.sh
# reloads the configuration parameters previously updated
```

An example of tgbconfig LDAP:

```
"ldap": {
    "base": "ou=People,dc=example,dc=com",
    "binddn": "cn=admin,dc=example,dc=com",
```

```
    "bindpw": "ABCDEF01234567890",
    "enabled": 1,
    "uri": "ldap://192.168.1.1"
}
```

Note: This process auto-configures several files for nslcd/nsswitch/ldap.config and for the /etc/pam.d service.

Note: The LDAP-enabled sudo command may need to be install on some versions of the Unified Base Station (`>sudo apt-get install sudo-ldap`) if needed.

Firewall support

The firewall is enabled on new Base Stations by default.

The Base Station firewall implements the following rules when enabled:

- Anti-Spoofing: All packets which do not specify the Local IP as the destination are dropped.
- All internal loopback connections are allowed.
- Incoming ICMP types 11/0, 11/1, 0/0, 3, and 8/0 are accepted, but rate limited to 2 per second globally. All other ICMP is dropped.
- Incoming UDP 161/snmp is accepted at a global rate of 10 per second. All other UDP connections are dropped.
- Incoming TCP 22/ssh is accepted but only from the Management Console / SSH subnets (see the following section) at a global rate of 5 per minute. All other incoming TCP connections are dropped.
- Incoming TCP 443/https is accepted for FWMA upgrade management when enabled (tgbconfig webservice.enabled) *(in a non-North American model base station)*.

Note: IPv6 has been disabled.

Firewall ssh_subnets (control of inbound SSH sessions)

The firewall uses the newer **firewall.ssh_subnets** array to hold **one or more sets of subnet/mask** that will be permitted to SSH inbound to the Base Station. This replaces the **management_console** of older Unified Base Station Software.

To be most secure, Base Station access via SSH, should contain the IP address/subnet of the RNI's Network Controller (NC) server at minimum.

It is recommended to include an IP address/subnet that is not over the OpenVPN connect to allow remote access in the event of a VPN outage to administration the base station or debug the VPN.

The **ssh_subets** array can be set with the following commands:

```
>sudo tgbconfig network.firewall.ssh_subnets [ ]
# this creates a new array of subnets/masks elements, (or deletes and recreates an existing array)
>sudo tgbconfig network.firewall.ssh_subnets.0 "<ip v4 subnet>/<ip v4 subnet
   mask –short form>"
# Repeat the command above for the next elements in your array (as needed).
# i.e. "10.0.0.0/8" or "172.24.1.4/32", etc.
# in CIDR notation
>sudo tgbconfig network.firewall.ssh_subnets
# to review your config
```

An example of tgbconfig firewall.ssh_subnets:

"firewall": {

"ssh_subnets": ["10.0.0.0/8", "172.24.1.4/32"]

}

 # Not Recommended: To disable this access control of SSH in the host firewall, enter 0.0.0.0/0 subnet/mask = any IP can connect to the Base Station's SSH service.

Starting the firewall

When the firewall is enabled in tgbconfig, it starts automatically at boot time.

On all Base Station Models (North American and non-North American)

Enable the firewall and start/restart the firewall service.

These commands use the Base Station's configuration to dynamically make host firewall rules.

```
>sudo tgbconfig network.firewall_enabled 1
# enable the host firewall within the base station
>sudo tgbconfig-apply.sh
# applies config changes to the system
>sudo systemctl [start|stop|restart|status] sensus-firewall.service
# Manual start/stop/restart of the Firewall Service
```

OpenVPN

Use the following instructions to install and configure the OpenVPN software on the Base Station and RNI. **As configured, only traffic destined for the tunnel endpoint will flow through the tunnel. You must alter the configuration to have all traffic flow through the tunnel**. The AES key size is set to 128 bits (by default) but can be set to 256 bits if desired, to be on par with FlexNet messaging/security.

PKI to support OpenVPN

The first step in building an OpenVPN configuration is to establish a PKI (public key infrastructure). The PKI consists of:

- A Root Certificate Authority–the third party that authenticates the certificate of the issuing Certificate Authority, aka subordinate.
- Note: If you don't have an issuing CA or are okay with having self-signed certificates in use within your AMI network, you can follow all steps as documented.
- An issuing CA certificate. In this case, the Root CA will serve as the issuer and sign (authenticate) the server and client certificates.
- A separate certificate (also known as a public key) and private key for the server.
- A separate certificate (also known as a public key) and private key for the client (TGB).
- Any pertinent policies used to establish X.509/500 parameters for your PKI elements. Where your policies deviate from this guide, you are encouraged to use and test them before finalizing the operational state of the VPN tunnels.

On the RNI NC server

Note: This is the same process for both North American and non-North American Base Stations.

SSH to the RNI's NC node, and su to root.

```
rpm -qa | grep openvpn
# check to see if the openvpn package is installed

yum install openvpn.x86_64
# if not present; the install that openvpn package for Linux
```

There are two ways to create the certificates/keys needed for the mutual TLS authentication of the RNI Network Controllers and the Base Station for the OpenVPN TLS tunnel:

- Use the Certificate Management program built into the RNI to generate certificates/keys (using an RNI self-signed certificate).
- Use the Certificate Management program built into the RNI to make a certificate signing request to a Certificate Authority (CA) of your choosing (like Verisign, goDaddy, Entrust, etc.) and import that certificate/key. Not shown in this document.

Note: Do this for each Base Station uniquely.

RNI generates a self-signed cert/key for Base Station

This section shows the use of the built-in Cert Authority (CA) to generate certificates/keys in the RNI's CA itself using self-signed certificates.

```
cd /opt/flexnet/httpd/bin/
# As root change directory to this directory

./certMgmt.sh
# run the built in certMgmt.sh script in the Network controller node of the RNI
```

```
************************************************************
 ********** Certificate Management System **********
 ************************************************************
 *
 *
 *
 * Please select the Certificate Management option:
 *
 * 1) Generate a new Certificate Request (Commercial Cert)
 * 2) Import new Certificate from an existing request
 * 3) Generate a new Local CA Signed Certificate
 * 4) Import Certificate into FlexNet Truststore
 * 5) Print Certificate Information
 * q) Quit
 ************************************************************

> 3
# Choose #3 in menu and press return.
```

```
The current hostname is yourRNI.yourdomain.com

 Do you wish to change it? (y/n) : y
 Please Enter the Fully Qualified Hostname : m400.yourdomain.com

# Enter the Fully Qualified Hostname of the Base Station i.e. m400.yourdomain.com in this example


 ************************************************
 * Generating Certificate for Host:
 * m400.yourdomain.com

 ************************************************

Generating a 4096 bit RSA private key
.....................................................................................................++
.....................++
writing new private key to '/opt/flexnet/security/private/m400.yourdomain.com.key'
-----
Using configuration from /opt/flexnet/security/conf/m400.yourdomain.com.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 20487 (0x5007)
        Validity
            Not Before: Apr 20 15:11:21 2018 GMT
            Not After : Apr 19 15:11:21 2028 GMT
        Subject:
            countryName               = US
            stateOrProvinceName       = NC
            organizationName          = Customer
            commonName                = m400.yourdomain.com
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                1F:66:D2:E0:68:88:67:0E:D6:9D:40:E1:9E:8C:9C:3B:FA:A6:93:02
            X509v3 Authority Key Identifier:
                keyid:38:A9:FC:B0:B8:CB:C8:5D:E9:85:6D:4B:89:28:B0:4C:FD:5E:DC:BA

Certificate is to be certified until Apr 19 15:11:21 2028 GMT (3652 days)

Write out database with 1 new entries
Data Base Updated
 ************************************************
 * Successfully created Certificate for:
 * m400.yourdomain.com
 ************************************************
```

```
Press [Enter] to continue... ENTER


 ************************************************************
 ********** Certificate Management System ***************
 ************************************************************
 *
 *
 *
 * Please select the Certificate Management option:
 *
 * 1) Generate a new Certificate Request (Commercial Cert)
 * 2) Import new Certificate from an existing request
 * 3) Generate a new Local CA Signed Certificate
 * 4) Import Certificate into FlexNet Truststore
 * 5) Print Certificate Information
 * q) Quit
 ************************************************************


  > q

# Type q and return to quit this script.
```

This creates a new set of cert/key files, on the RNI in these locations: (using the m400.yourdomain.com example):

- The CA's cert (pem): at location: /opt/flexnet/security/CA/cacert.pem
- The Base Station's cert (crt) at location: /opt/flexnet/security/certs/ **m400.yourdomain.com**.crt
- The Base Station's private key (key) at location: /opt/flexnet/security/private/**m400.yourdomain.com**.key

You need to transfer a copy of these three files to the Base Station (usually over SCP/SSH) from the RNI's NC locations. Copy them to the Base Station's /opt/tgbprogs/openvpn directory.

From the RNI NC node:

```
>scp /opt/flexnet/security/CA/cacert.pem tmadmin@ m400.yourdomain.com:~
>scp /opt/flexnet/security/certs/m400.yourdomain.com.crt tmadmin@
    m400.yourdomain.com:~
>scp /opt/flexnet/security/private/m400.yourdomain.com.key tmadmin@
    m400.yourdomain.com:~

# Note: you will need to type the password for the account you scp with on the base station (in this example it is
tmadmin) and file will be stored in that users "home directory"
```

Automatically starting the OpenVPN server

Sensus recommends starting the OpenVPN server automatically. To configure it to start up upon boot, issue the following command as root:

	On RHEL <=6 (in NA RNI 3.x)	On RHEL 7 or higher (in NA RNI 4.x)
Make the OpenVPN service start automatically at boot	chkconfig **openvpn** on	systemctl enable **openvpn**
To manually start/stop/restart the OpenVPN services	service **openvpn** start	systemctl [**start**|stop|restart|status] **openvpn**

If you see *Failed*, use log entries in /var/log/messages and go back through the previous steps to ensure you followed them correctly.

Because all connections originate in the Base Station, verify that the NC firewall allows incoming connections on UDP port 1194, and the Base Station firewall allows outgoing connections on UDP 1194. Also note that any external firewalls between the NC and Base Station must allow UDP port 1194 connections from Base Station to NC, and UDP packet flows on that port in both directions. Possible commands to use include **iptables –L –n -v**.

On the Unified Base Station

The Unified FlexNet Base Station ships with the OpenVPN package as part of the built product. You can start configuring it with the root user account as soon as you have access into it.

Note: This is same process for North American and non-North American Base Stations.

Log in the Base Station with a privileged account.

You will need to move the 3 files from the NC (that were placed in the “home” directory into the proper directory for OpenVPN service and change ownership to root:tmadmin

```
>cd ~
```

this cd to the users home directory

```
>sudo mv cacert.pem  /opt/tgbprogs/openvpn/
>sudo mv m400.yourdomain.com.crt /opt/tgbprogs/openvpn/
>sudo mv m400.yourdomain.com.key /opt/tgbprogs/openvpn/
```

this three commands mv the files from the users home directory into the /opt/tgbprogs/openvpn directory

```
>sudo chown root:tmadmin /opt/tgbprogs/openvpn/*
```

this changes the file ownership of all files /opt/tgbprogs/openvpn/ directory to “owner root” and “group tmadmin”

Check that 3 files are correctly transferred and stored in the base station file system

```
>ls -l /opt/tgbprogs/openvpn
```

total 32
-rw-r----- 1 root tmadmin 1960 Nov 12 16:28 **cacert.pem**
-rw-r--r-- 1 root tmadmin 441 Nov 15 16:18 client.conf
-rw-r--r-- 1 root tmadmin 3428 Sep 24 13:19 client.conf.template
-rw-r--r-- 1 root tmadmi 441 May 18 2016 default-client.conf
-rw-r----- 1 root tmadmin 7142 Nov 12 16:28 **m400.yourdomain.com.crt**
-rw-r----- 1 root tmadmin 3272 Nov 12 16:28 **m400.yourdomain.com.key**

You should see a cacert.pem, a specific file for this base station crt and key files respectively, m400.yourdomain.com.crt & m400.yourdomain.com.key in this example

Configure the OpenVPN client parameters on the Base Station:

```
>sudo tgbconfig network.vpn.clients [ ]
# the M400 model and have 1 or MORE backhaul OpenVPN tunnels so declare an
json array (we will be using just one on array element 0

>sudo tgbconfig network.vpn.clients.0
# set the 0th element of the array (1st)

>sudo tgbconfig network.vpn.clients.0.enabled 1
# enables the 0th openvpn client

>sudo tgbconfig network.vpn.clients.0.host <outside (routeable IP) your-RNI-
NC-IP-address>
# this is usually the Internet addressable IP of the NC of the RNI.

>sudo tgbconfig network.vpn.clients.0.port 1194
# This set the protocol/port to udp/1194

>sudo tgbconfig network.vpn.clients.0.ca /opt/tgbprogs/openvpn/ca.pem

# point to the ca's cert file you on the local tgb filesystem (can be either a ca.crt or a ca.pem format)


>sudo tgbconfig network.vpn.clients.0.cert /opt/tgbprogs/openvpn/< m400.yourdomain.com >.crt

# point to the tgb's specific .crt file you on the local tgb filesystem , i.e. m400.yourdomain.com.crt


>sudo tgbconfig network.vpn.clients.0.key /opt/tgbprogs/openvpn/< m400.yourdomain.com >.key

# point to the tgb's specific .key file you on the local tgb filesystem , i.e. m400.yourdomain.com.crt


>sudo tgbconfig network.vpn.clients.0.cipher AES-128-CBC

#  (AES-128-CBC is default or you can "set on both side AES-256-CBC", as needed)

>sudo tgbconfig network.vpn.clients.0.name "<some nickname for this OPENVPN connection>"

# this will be used as the filename for your generated OPENVPN config file for this specific OpenVPN
connection (as you can have multiple). File will be stored in: /opt/tgbprogs/openvpn/
```

Set the RNI section of the Base Station configuration to call the RNI through the OpenVPN tunnel.

IMPORTANT: If you do not route your RNI traffic into the tunnel, it attempts to reach the RNI outside of the tunnel (insecure). Ensure the IP address that you assign to the Network Controller is inside the tunnel (tun0 interface of the Network controller within the RNI).

```
>sudo tgbconfig rni.rnis.data_ip <NC_inside_tunnel_endpoint_IP>
# (e.g. 10.11.12.1 or whatever the address of the NC Tun0 interface is).

>sudo tgbconfig rni.rnis.connection_ip <NC_inside_tunnel_endpoint_IP>
# (e.g. 10.11.12.1 or whatever the address of the NC Tun0 interface is).
```

To validate the configuration:

```
>sudo tgbconfig
# the result will be a json view of your entire config (Specifically check the network.vpn and rni.rnis sections.
```

To apply your config changes and restart services:

```
>sudo tgbconfig-apply.sh
# the result will be a json view of your entire config (Specifically check the network.vpn and rni.rnis sections.

>sudo systemctl restart openvpn
>sudo tgbprog restart

# OR

>sudo reboot
#Reboot or restart services to take affect
```

To check the status of your OpenVPN tunnel from the Base Station to the RNI NC:

```
> ifconfig
# if openVPN connection was successful you will see a Tun0 interface up/up.
# (it is good to make note of your side IP address of the point-to-point link)

>ping <the NC side inside the OpenVPN tunnel headend>
# (e.g. 10.11.12.1 in this example e.g. the Tun0 address of the NC in the RNI)

# ALSO more detailed debugging of the OpenVPN can be done with:

>netstat -rn
# you should see routes to tun0

>tail -f /var/log/syslog
# you will see the OpenVPN attempting to authentication (TLS mutual auth) and then ultimately bringing up the tunnel
```

Note: You do **not** need to configure the OpenVPN configuration files directly. The tgbconfig commands you have entered are used to automatically generate the specific OpenVPN configuration file.

Logging and monitoring

The Base Station supports standard Linux logging and monitoring with SYSLOG and SNMP. These mechanisms provide standards-based logging and monitoring of Base Station equipment in the field.

SYSLOG is enabled to log events locally by default. In order to send events to a remote SYSLOG server, complete the steps in the SYSLOG and SNMP configuration sections.

SYSLOG configuration

SYSLOG is enabled to log all events locally by default. In order to send all events to a remote SYSLOG server, the following configurations must be completed:

SSH into the Base Station and issue the following commands:

```
>sudo tgbconfig remote_logging.host <ip address of remote syslog server>

>sudo tgbconfig remote_logging.enabled  1
```

Following is an example of a tgbconfig remote_logging:

```
  "remote_logging": {
     "protocol": "udp",
     "level": "*",
     "facility": "*",
     "enabled": 1,
     "host": "192.168.1.1",
     "port": 514
  }
```

Apply the configuration changes and restart the rsyslog service:

```
>sudo tgbconfig-apply.sh
# to apply the current configuration

>sudo systemctl restart rsyslog
# to restart the SYSLOG service
```

This configuration sends all events to the remote SYSLOG server. After restarting the SYSLOG server, verify that events are being received from the Base Station. In addition, please ensure that the IP Address entered in the SYSLOG configuration file is reachable over the network from the Base Station.

When OpenVPN is enabled on the Base Station, verify that SYSLOG traffic is going over the VPN tunnel.

Since SYSLOG messages are in the clear, running them over the OpenVPN connection is recommended. The default port for SYSLOG connections is 514.

If you are using OpenVPN and wish to have this connection travel over the tunnel, contact Sensus for complete information to set up this configuration. The configuration may change based on where the SYSLOG server resides on the RNI network and may involve provisioning on the Base Station and the RNI.

SNMP configuration (v2c or v3)

SNMP provides a method to monitor the resources on a Base Station. This is provided through standard SNMP configuration performed on the Base Station. If you are using SNMP, it is important to change the factory-supplied community string to a secure, unique string for your implementation. The factory-supplied “public” string is the same for all factory-shipped systems, and it must be changed to ensure your system is only accessed by authorized personnel.

Locate the MIB files for the NMS/SNMP station to use on the Base Station

The MIB files are located on the Base Station file system in the /usr/share/snmp/mibs directory. You can use SCP/SFTP to copy the files from the Base Station for use with your NMS (SMNP) monitoring tool.

> cd **/usr/share/snmp**

>sudo **tar cvf basestation-mibs.tar mibs/**

>sudo **mv basestation-mibs.tar ~**

>sudo **chown tmadmin:tmadmin basestation-mibs.tar**

This set of commands (above)

- changes directory to the parent directory where the mibs directory is located,
- creates a tarball of the mibs subdirectory,
- then moves that tarball into a “home directory”,
- then changes the file ownership from root to privileged user (in this example tmadmin)

 # Now you can SCP or SFTP that set of MIB files tarball off the base station system into your NMS/SNMP host and expand (tar xvf basestation-mibs.tar) to load into your SNMP server.

*NOTE: the .txt files in that mibs directory is the MIB file itself for each respective

This configuration allows remote monitoring of the Base Station via SNMP. Additional options can be provisioned in the SNMP setup. Type **tgbconfig snmp** to view these parameters.

When OpenVPN is enabled on the Base Station, verify that SNMP traffic is using the VPN tunnel. SNMP messages should be run over the OpenVPN connection, as they carry sensitive information in the clear. SNMPv1 and SNMPv2 do not offer any intrinsic encryption facilities. The default ports for SNMP connections are 161 and 162.

If you are using OpenVPN and wish to have SNMP travel over the tunnel, contact Sensus for complete information on configuration settings. The configuration may change based on where the SNMP client resides on the RNI network and may involve provisioning on the Base Station and the RNI.

SNMP v2c

The following steps outline how to change the public read-only and read-write community strings and enable SNMP monitoring using SNMP v2c.

SSH into the Base Station and issue the following commands:

```
# to set the base station for inbound SNMPwalks, SNMP MIB gets and sets
>sudo tgbconfig snmp.enabled 1
>sudo tgbconfig snmp.ro_community <12 char complex RO ASCII string>
>sudo tgbconfig snmp.rw_community <12 char complex RW ASCII string>

#if you wish to set up SNMP Traps for Alarms (Highly Recommend)
>sudo tgbconfig snmp.notification.enabled 1
>sudo tgbconfig snmp.host "<IP Address of your NMS/SNMP station>"
>sudo tgbconfig snmp.type "trap"

# You may also set the snmp.contact, snmp.engine_id, and snmp.location for your environment,
rather than the default values listed below.

# for SNMPv2c you do not need to configure the snmp.authentication nor snmp.privacy sections,
nor set the snmp.ro_user nor snmp.rw_user fields in SNMP v2c
```

Following is an example of the JSON configuration of SNMP v2c:


```
{
    "authentication":    {
        "password":    "",
        "protocol":    ""
    },
    "contact":    "SYSCONTACT",
    "enabled":    1,
    "engine_id":    "SYSID",
    "location":    "SYSLOCATION",
    "notification": {
        "enabled":    1,
        "host": "10.1.208.172",
        "type": "trap"
    },
    "privacy":    {
        "password":    "",
        "protocol":    ""
    },
    "ro_community": "A1234567890bc",
    "ro_user":    "",
    "rw_community": "a1234567890eF",
    "rw_user":    "",
    "security_model":    "v2c"
}
```

SNMP v3

Note: SNMP v3 requires Unified Base Station software release 2.6 or later.

The following steps outline how to change the public string and enable SNMP monitoring using SNMP v3.

SSH into the Base Station and issue the following commands:

>sudo tgbconfig snmp.enabled **1**

>sudo tgbconfig snmp.security_model **v3**

>sudo tgbconfig snmp.ro_user "**<set a readonly UserID>**"
>sudo tgbconfig snmp.rw_user "**<set a readwrite UserID>**"
>sudo tgbconfig snmp.privacy.protocol **<AES|DES>**
your choice is between AES or DES only – this sets the SNMP v3 encryption algorithm

>sudo tgbconfig snmp.privacy.password "<a strong password to be used for encryption>"

>sudo tgbconfig snmp.authentication.protocol **<MD5|SHA>**
your choice is between MD5 or SHA (SHA1) only – this sets the authentication hash algorithms for SNMPv3

>sudo tgbconfig snmp.authentication.password **"<a strong password to be used for authentication on both the readonly and readwrite accounts>"**

 # you may optionally set the "snmp.location", snmp.engine_id, snmp.contact to fit your Network management standards. And you can also set the snmp.notifications (for traps) same as in SNMP v2c. (see above example).

#NOTE the base station does NOT set a SNMP v3 contex, it uses the global context

#NOTE: you do NOT need the ro_community nor the snmp.rw_community to be set in SNMP v3 as you using UserIDs and passwords instead.

Following is an example of the JSON configuration of SNMP v3:

```
tgbconfig snmp
{
	"ro_user":	"sensusro",
	"security_model":	"v3",
	"engine_id":	"SYSID",
	"notification": {
		"type": "trap",
		"host": "10.1.208.244",
		"enabled":	0
	},
	"rw_user":	"sensusrw",
	"privacy":	{
		"password":	"AcryptoStrongPa55word",
```

```
        "protocol":     "AES"
    },
    "enabled":      1,
    "ro_community": "",
    "authentication":       {
        "password":     "Area11yStrongPa55word",
        "protocol":     "SHA"
    },
    "contact":      "SYSCONTACT",
    "location":     "SYSLOCATION",
    "rw_community": ""
}
```

Example of a snmpwalk from an NMS station to the previous configuration:

```
# An Example of a NMS host (unix) using a snmpwalk command to walk for a specific OID (given config sample shown above)

>snmpwalk -v 3 -u sensusrw -A Area11yStrongPa55word -a SHA -X AcryptoStrongPa55word -x AES -l authPriv 10.22.60.179 .1.3.6.1.2.1.4.20.1.1

IP-MIB::ipAdEntAddr.10.22.60.179 = IpAddress: 10.22.60.179

IP-MIB::ipAdEntAddr.127.0.0.1 = IpAddress: 127.0.0.1
```

SNMP v2c and v3: Apply the configuration changes and start/restart the SNMP service

```
>sudo tgbconfig-apply.sh

# to apply the current configuration


>sudo systemctl restart snmpd

# to restart the SNMP service
```

Appendix A – Selecting permitted SSL ciphers

Use **openvpn - -show-ciphers** on the RNI NC system or the Base Station command line, to see what Crypto ciphers are available on that side. (Your settings will determine which ciphers are enabled.)

To view the ciphers that are available, issue the following command:

```
>openvpn - -show-ciphers
```

This produces a list similar to the following:

AES-128-CBC 128-bit default key (fixed)
AES-128-OFB 128-bit default key (fixed)
AES-128-CFB 128-bit default key (fixed)
AES-192-CBC 192-bit default key (fixed)
AES-192-OFB 192-bit default key (fixed)
AES-192-CFB 192-bit default key (fixed)
AES-256-CBC 256-bit default key (fixed)
AES-256-OFB 256-bit default key (fixed)
AES-256-CFB 256-bit default key (fixed)
AES-128-CFB1 128-bit default key (fixed)
AES-192-CFB1 192-bit default key (fixed)
AES-256-CFB1 256-bit default key (fixed)
AES-128-CFB8 128-bit default key (fixed)
AES-192-CFB8 192-bit default key (fixed)
AES-256-CFB8 256-bit default key (fixed)

In the client and server configuration files, the cipher option is used to specify a colon delimited list of permitted ciphers. For example:

```
> cipher AES-128-CBC:AES-256-CBC
```

This option may be used to limit the ciphers for performance reasons or export requirements.

Recommended server-side changes (server.conf)

- It is likely that the server setting will need to be changed based on the host network.
- Use **cipher AES-256-CBC** unless you have other instructions. At least one cipher must be active in the server.conf file. This needs to be the same cipher used in the client.conf file.

Xylem |'zīləm|

1) The tissue in plants that brings water upward from the roots;

2) a leading global water technology company.

We're a global team unified in a common purpose: creating advanced technology solutions to the world's water challenges. Developing new technologies that will improve the way water is used, conserved, and re-used in the future is central to our work. Our products and services move, treat, analyze, monitor and return water to the environment, in public utility, industrial, residential and commercial building services settings. Xylem also provides a leading portfolio of smart metering, network technologies and advanced analytics solutions for water, electric and gas utilities. In more than 150 countries, we have strong, long-standing relationships with customers who know us for our powerful combination of leading product brands and applications expertise with a strong focus on developing comprehensive, sustainable solutions.

For more information on how Xylem can help you, go to www.xylem.com

xylem

Sensus
637 Davis Drive
Morrisville, NC 27560
Tel +1.800.638.3748
www.sensus.com