TANDBERG

Implementing Secure Management

Configuring Secure HTTPS between TANDBERG Products

D50520 Rev 2

TABLE OF CONTENTS

DOCUMENT REVISION HISTORY

Revision 2	Document overhaul to add more step-by-step guides Added Level 3 Implementation Guides Added Readiness Checklists Added VCS support information Reworked Appendixes to be organized by system type
Revision 1.1	Correction Release – Added missing references for externalmanager for MPS and corrected troubleshooting step for MPS
Revision 1.0	Initial Publication for TMS v11.9, F7 and JF4.2 Releases

1 INTRODUCTION

This document is intended to provide TANDBERG customers with background information on the security of communications for management when using TANDBERG products and provide details on how to configure those systems secure management. This document is written to be current as of TMS version 12.1, TANDBERG MXP F7.1, TANDBERG VCS X4.0 and TANDBERG MPS J4.3 and should apply to future versions as well. These versions do not reflect the minimum versions required for this functionality, but rather the configurations and abilities as reflected in these versions. Minimum version requirements are elaborated on in the sections related to each device.

Document Organization

This document is broken into an technology introduction, followed by chapters based on the device to be managed. Each chapter explains the management technologies in use followed by implementation guides for that particular device.

NOTE: This document assumes an working knowledge of network technologies, X.509 certificates and TLS encryption.

- Readers not familiar with X.509, public key cryptography, or TLS should start by reviewing Appendix A – Communications, Certificate and Key Primer. Appendix A provides a primer to understand the differences in encryption vs. authentication in friendly terms as well as introducing and explaining the relevant concepts required when implementing a TLS communication setup.

- Readers needing assistance creating certificates, key pairs and converting formats may read **Appendix B – Creating and Working with Certificates for Windows Server and TMS** and **Appendix C – Creating and Working with Certificates for TANDBERG Systems** which provides step-by-step instructions for generating X.509 certificates and key pairs using commonly available tools.

Readers who do not use TMS may skip the TMS section, but users of TMS looking for secure management should read both the TMS and device-specific chapters of the devices they wish to use.

2 TECHNOLOGY INTRODUCTION - SECURITY LEVELS

The level of security an organization requires when communicating with a device will vary based on the company's policies and willingness to implement security infrastructure. Most commonly organizations are concerned about the security of information sent over a network when managing a system. Information sent that organizations may be concerned about include

- Passwords
- Video images from Conferences
- Call Detail Records
- Participant Information

When managing the system, sensitive information may be sent over a network that may or may not be secured. Many protocols send their information in a 'clear text' format which means the data can easily be read or interpreted if the information on the IP network is intercepted. Ideally, all data would be encrypted so that only the intended receivers would be able to interpret the information, but this is not always possible.

As an intermediate step, many protocols utilize methods that make it so sensitive information like passwords are not actually sent out over the network but rather use things such as

Challenge/Response systems. A Challenge/Response method is used to verify knowing a shared secret without actually sending the secret information itself over the network. This allows non-encrypted protocols such as HTTP or telnet to secure important information like passwords without the overhead or complexity of encrypting the entire conversation.

The security provided with the standard services used by systems may not be sufficient to meet an organization's needs. Most of the services are limited by the definition of the service/protocol itself. The protocols commonly used with TANDBERG systems are:

> **HTTP** – provides no encryption but can require a password for authentication. HTTP digest is supported to provide passwords without sending in the clear (Challenge/Response). The HTTP clients in TANDBERG systems only support anonymous connections (externalmanager, etc)
>
> **Telnet** – provides no encryption but can require a password for authentication. Telnet Challenge is supported to provide passwords without sending the password in the clear (Challenge/Response)
>
> **SNMP v1/v2** – provides no encryption and uses a Community Name as a password for authentication. The Community Name is sent in the clear
>
> **FTP** – provides no encryption but can require a password for authentication. The password is sent in the clear
>
> **SSH** – provides encryption for an entire conversation and uses passwords for authentication. Keys must be checked to verify identity, but X.509 Certificates are not used.
>
> **HTTPS** – provides encryption for an entire conversation and passwords for authentication. Some systems support loading your own X.509 certificate to provide identity authentication.

The above list shows that the actual level of security when managing a system will vary based on the services/protocol used when interfacing with the system. To simplify the comparison and discussion of security levels, we will simplify the behaviors into four levels. These levels do not represent industry standardized definitions but are only used for illustration and simplification.

> **Level 1 : Password Only Security** – Passwords and conversations are not protected on the network (sent in clear)
>
> **Level 2 : Password Authentication with Protected Passwords** – Password exchanges are protected using challenge/response systems but the rest of the conversation is not protected on the network (sent in clear)
>
> **Level 3 : Encryption using Simple Authentication** – Encryption is used to protect the conversation on the network and password exchange. No verification of identity through certificates or if certificates are used - errors or warnings about the certificates are ignored.
>
> **Level 4 – Encryption with Authorization through Certificates** – Encryption is used to protect the conversation and passwords on the network. Certificates are exchanged and verified before communications will proceed.

Ideally every conversation would be secure in that the conversation was encrypted and all identities are verified. However, doing so requires infrastructure and policy adherence that not all organizations are willing to invest time and resources to implement. Due to the high requirements, most organizations settle for a level of security between Level 2 and Level 3. Organizations with stricter policies on confidentiality or secrecy may demand an environment that provides Level 4 security.

With existing TANDBERG products, security up through Level 3 can be achieved when limiting yourself to using HTTPS and SSH when interacting with devices. However, for full functionality, especially when used with TMS, lower security protocols were relied upon. To address customer's needs for more strict security, TANDBERG has added functionality and configuration choices to the products to allow those who demand a higher level of security to do so.

At the time of writing, the products enhanced to support this higher level of security are the TANDBERG MXP endpoints (not including the TANDBERG 150MXP), TANDBERG MPS, TANDBERG VCS and TANDBERG TMS. Support will be expanded into other product lines in the future.

3 TANDBERG MXP ENDPOINTS

Before attempting to deploy a Secure Management solution for MXP systems, it is critical that an administrator understands the fundamentals of certificates, TLS, and the MXP platform's functionality. Appendix A of this document provides primer information on TLS and certificates for those who are not familiar with those technologies and Appendix C provides step by step instructions for creating and working with certificates.

This chapter is part reference and part implementation guide for the MXP endpoints. The first half details methods and protocols in use between an MXP system and an external management system. The material is broken into sections explaining the default methods implemented in MXP systems, followed by the functionality added beginning with F7 software to increase the security level. These sections should be read and understood before attempting to deploy MXP systems in a Secure Management solution.

The second half of this chapter provides a readiness checklist for an administrator to verify they have the information and network configuration required to configure MXP systems for Secure Management. This is followed by comprehensive step-by-step Best Practice guides to configuring an MXP system for Secure Management.

MXP Secure Management Reference Material

Default Protocols and Services in MXP Systems

All TANDBERG MXP endpoints support multiple interfaces for managing the system, including.

- HTTP
- SSH[1]
- Serial Port
- HTTPS
- SNMP
- FTP
- Telnet
- SNMP Traps

In the factory default configuration, all of the above protocols are enabled except for HTTPS and SSH. These protocols may be enabled or disabled via the XML interface, the web administrator interface under *System Configuration > Misc* , or using the dataport and xconfiguration commands.

```
xconfiguration https mode: <on/off>
xconfiguration http mode: <on/off>
xconfiguration ssh mode: <on/off>
xconfiguration telnet mode: <on/off>
xconfiguration telnetchallenge mode: <on/off>
xconfiguration snmp mode: <on/off>
xconfiguration ftp mode: <on/off>
```

Turning a service on or off requires a reboot for changes to take effect.

For services that require a server certificate, such as HTTPS, the systems come preconfigured with a X.509 certificate signed by a TANDBERG Certificate Authority. By default this certificate authority will not be trusted by a user's computer resulting in a certificate warning in their web browser unless they specify to ignore the warning or trust the TANDBERG Certificate Authority.

[1] Added in F5.0 Software

Additionally, MXP devices have internal clients used to connect to management platforms. These clients are used for External Manager, External Services, and Corporate Directory. These clients by default use HTTP to connect to the configured management addresses. These services can be configured from several locations in the onscreen menus, XML interface, web interface, or dataport. The most direct way to configure these services is through the dataport where each service has their own list of settings under them

xconfigure externalmanager
xconfigure externalservices
xconfigure corporatedirectory

All of these clients are used when the system is managed by a TANDBERG TMS system, but are optional in stand-alone implementations.

Lastly, MXP systems also implement a feedback system which can be used by external systems to monitor activity of the device. Feedback works by having the system post changes using HTTP by default to an external URL. This feedback system is configured using the XML interface or **xcommand feedbackregister** series of commands available in the dataport of the MXP or via XML

The default services/protocols can be summarized based on the security levels outlined in the Technology Introduction chapter of this document as follows:

Service/Protocol	Security Level
HTTP Server	Level 2
HTTPS Server	Level 4 (if client checks certificate)
Telnet	Level 1
Telnet Challenge	Level 2
FTP	Level 1
SNMP	Level 1
SNMP Traps	Level 1
SSH	Level 3
External services/feedback	Level 2

The overall management security of the device can be improved by disabling unneeded services, and increasing the security of the services in use. Previously, when using TMS, administrators were limited in what services they could disable. Now, when combined with Secure-Only settings in TMS, administrators can achieve a Level 3 security by using the security settings added beginning with F7 or a Level 4 security when combined with the verify certificates settings.

Secure Management Features Added Beginning with F7 Software

Starting with software release F7.0, the security of management interfaces has been increased, including the ability to achieving a security of Level 4, through the following new additions:

Uploading Root Certificates

To properly handle X.509 certificates when acting as a HTTPS client, systems now support loading a list of trusted root certificates into the system. The system allows the upload of a PEM encoded file which contains the root certificates of any CAs you wish the system to trust. The system will use these root certificates when validating X.509 certificates presented by HTTPS servers the system connects to.

The system supports uploading of a single PEM encoded file, which may contain multiple certificates. If an existing list of certificates is present, uploading a new PEM encoded file will overwrite the existing list. Certificate lists are uploaded using the web interface of the system under *System Configuration > Certificate Management*

The system supports X.509 root certificates using RSA and DSA keys that are encoded in the PEM base-64 format. Root certificates are not encrypted with passphrases. For additional help with certificate formats, please see Appendix A of this document.

Uploading Server Certificates

Uploading a server certificate to use for the system's HTTPS server was introduced in F6.0 software, but is covered here for completeness. An uploaded certificate replaces the factory default certificate and is used for the HTTPS server of the system. A certificate is uploaded through the web interface of the system under *System Configuration > Certificate Management*. The certificate to be uploaded must be a PEM encoded file. If the private key is encrypted with a passphrase, the passphrase must be supplied when uploading the certificate in the separate Passphrase field. For additional help with certificate formats, please see Appendix A of this document.

HTTPS for External Services

External Services can now be configured to use either HTTP or HTTPS. The setting is controlled with a new command **protocol** under each service's configuration.

xconfiguration externalmanager protocol: <HTTP/HTTPS>
xconfiguration externalservice protocol: <HTTP/HTTPS>
xconfiguration corporatedirectory protocol: <HTTP/HTTPS>

When protocol is set to HTTPS, only HTTPS will be used. The system intentionally will not fall back to HTTP if HTTPS fails. The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

HTTPS for Feedback URLs

Feedback URLs can now be defined to use HTTPS in the URL. If the feedback URL registered in the system starts with HTTPS, HTTPS will be used when posting events to the external server. The syntax of the **xconfiguration feedbackregister** command was not modified, only updated that using a https:// formatted URL is now valid. When the URL is set to HTTPS, only HTTPS will be used (system will not fall back to HTTP if HTTPS fails). The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

Verify Server Certificates Mode

To achieve Level 4 security, a system must not only handle certificates, it must validate them and adhere to any warnings or errors discovered. Enforcing certificate checking may not be desirable for organizations that want the encryption offered by TLS but do not wish to manage and deploy valid server certificates on all devices. Using TLS without validating certificates results in an equivalent Level 3 security. To address both needs, a new system setting is defined to control the handling of X.509 certificates

xconfiguration https verifyservercertificate: <on/off>

When the system is acting as a HTTPS client to connect to an external server, the server will provide a X.509 certificate to validate its identity. If VerifyServerCertificate is enabled, the server certificate will be validated before a connection continues. The system checks attributes of the Certificate such as:

- certificate is complete and any checksums are valid
- the current time is within the certificates valid date range

- the common name matches the hostname name used to access the server
- certificate has been signed by a certificate authority the system trusts through its root certificate

If any of the certificate checks fail - the HTTPS connection will fail and the communication is halted to prevent sharing information with an unsecure party.

If VerifyServerCertificate is disabled, the HTTPS client will accept any complete server certificate presented by the server and ignore any warnings or errors allowing the communication to continue.

Additional Settings Required for VerifyServerCertificate

When VerifyServerCertificate is enabled, the system must be able to perform its validation checks including hostname and date checks. Therefore the system's configuration must include:

- NTP must be configured and active in the system
- a DNS Server must be defined in the system
- Server Addresses must be entered as hostnames, not IP Addresses

Certificate Dataport Command

F7 introduced a new dataport command which allows the viewing or deleting the system's currently loaded server certificate or root certificate. The command's syntax is

```
certificate <list>/<del> [<cert/root>]
```

The command will list the contents of the uploaded certificate used for the system's web server or root certificate. The output can be copied and ran through OpenSSL tools if desired to validate their contents. The command can also be used to delete either certificate if needed. If replacing certificates, there is no need to delete them first, as they can be overwritten by uploading a new certificate via the web interface of the system.

Interpreting Certificate Failure Errors

For troubleshooting purposes, if an administration needs to see the specific error code that caused a certificate to be rejected when the system connects to a server, you must enable syslog in the system prior to performing a server request. The error code for the certificate will be displayed and the error codes may be referenced in the OpenSSL documentation

Example:

1. Telnet or SSH to the system
2. Enable syslog by entering **syslog on**
3. Perform a Corporate Directory Search to initiate a server connection

   xcom corpdirsearch startswith: a

4. The output will include a line starting with SSL like the following

   ```
   SSL: socklib_sslHandshake:
   VerifyResult=14=ERROR_IN_CERT_NOT_AFTER_FIELD, NameCheck: Match
   ```

5. Stop syslog by entering **syslog off**

The error code is listed as VerifyResult=14=ERROR_IN_CERT_NOT_AFTER_FIELD which means the date on the server certificate was expired. SSL Error codes can be looked up in the OpenSSL documentation

MXP Secure Management Profiles

With the additional features added beginning with F7, MXP systems can operate in a full Level 3 or Level 4 security configuration.

Level 3 is available by

- Disabling all services except for HTTPS and SSH
- Configuring feedback and client services (external manager, services, and corporate directory) to use HTTPS protocol

Level 4 is available by

- Disabling all services except for HTTPS
- Configuring feedback and client services (external manager, services, and corporate directory) to use HTTPS protocol
- Installing a valid Server Certificate on the system
- Installing root certificates to trust and enabling Verify Certificates mode

The Implementation Guides included in this chapter provide step by step best instructions for deploying systems in either of these security profiles.

MXP Secure Management Implementation Material

Readiness Checklist for implementing HTTPS Management for MXPs

This list is intended to provide a quick reference to the information that an administrator must have available before implementing HTTPS management. Failure to organize this information prior to implementation will hinder or prevent you from deploying this functionality.

If not using verify certificates

- Each endpoint must be running F7.0 software or newer
- You must have an administrative logon to the MXP endpoints
- Be familiar with the HTTPS related commands outlined in the MXP reference section of this document
- If a management tool besides TMS, your external management platform or tools must support HTTPS connections
- If using TMS to manage the endpoints, TMS must be version 11.9.1 or newer, and you must have an administrative logon to TMS

If using the verify certificates mode, the following ADDITIONAL items apply

- Each endpoint must have a fully qualified domain hostname (FQDN) that points to the system's IP Address. The FQDN is defined in your DNS Server
- You must have a NTP source that endpoints can be pointed at for time synchronization. See endpoint command xconfiguration NTP
- Each endpoint must have a DNS lookup server defined in its configuration. See endpoint command xconfiguration IP DNS
- Each endpoint will require its own Server Certificate be generated
- You must have the Private Key, Server Certificate, and passphrase (if used) for each endpoint. Certificates and Keys must be in PEM encoded files
- You must have the public certificate of the Root Certificate Authority that will be used to verify the Server Certificates used on the network
- If using TMS, TMS must have a FQDN configured, and its own Server Certificate

For information on how to generate Server Certificates or related information, please **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. Once the steps outlined in the checklist are complete and the information is gathered, you can configure your endpoints to use HTTPS. The next sections outline the steps required to configure an MXP endpoint in Level 3 or Level 4 security.

Implementation Guide for Configuring an MXP for Level 3 Security

The following provides an example of best practices for configuring a default MXP system for Level 3 security, where the system will only support communicating with management systems that use HTTPS but certificates are not verified, reducing the complexity and requirements of the deployment. If preparing a system for use with TMS, these steps should be performed either **before** adding the system to TMS or before enabling Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see Appendix C – Creating and Working with Certificates for TANDBERG Systems in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

These steps assume you already have the device configured on the IP network via DHCP or Static IP and you know the IP Address of the system. Replace <systemIP> in the examples below with the actual IP address of the system

1. Open a telnet session to the system using a telnet utility of your choice. In Windows you can use *Start Menu* > *Run* and enter **telnet <systemIP>** and hit OK
2. When prompted for a password, the default password is TANDBERG
3. After logging in, enter the following commands individually

   xconfig http mode: off

   xconfig https mode: on

   xconfig telnet mode: off

   xconfig telnetchallenge mode: off

   xconfig snmp mode: off

   xconfig ftp mode: off

   xconfig ssh mode: on **(<- Optional but recommended)**

   xconfig https vertifyservercertificate: off

4. Configure the external services to be disabled. These will be enabled automatically by TMS if needed.

   xconfig externalmanager address: 127.0.0.1

   xconfig externalmanager protocol: https

   xconfig externalservices mode: off

   xconfig corporatedirectory mode: off

5. Configure the IP Password you wish to use with the system

   xconfig strictpassword: on

   xconfig systemunit password: superstrongpasswordhere

6. Reboot the codec by entering **xcom boot** in the telnet session. Wait until the codec has fully restarted.
7. Verify the changes. Open a new web browser window and enter the address https://<systemIP> and the web page of the codec will start to open.

   NOTE: You will see SSL security warnings in your browser because the certificate authority is not trusted and the name of the server does not match that of the certificate

8. When prompted for login, enter anything as the username and your new password for the password.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using ‘Enforce Management Settings’ when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

Implementation Guide for Configuring an MXP for Level 4 Security

The following provides an example of best practices for configuring a default MXP system for Level 4 security, where the system will only support communicating with management systems that use HTTPS and have valid X.509 certificates. If preparing a system for use with TMS, these steps should be performed either **before** adding the system to TMS or before enabling Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

1. Decide the hostname that will be used to manage the system. This hostname must be defined in your network's DNS servers.

   Example: **rm204-bld5.company.com**

   NOTE: If DNS records are to be created by the DHCP server, the hostname portion of the FQDN is taken from the system's systemname setting.

2. Decide which certificate authority you will use to sign your certificates and obtain the public root certificate for that CA. Convert that file if necessary to PEM file format and save as **rootcerts.pem**

   If multiple certificate authorities need to be trusted by the system, obtain the root certificate for each and catonate each into a single PEM file. See **Appendix C – Creating and Working with Certificates for TANDBERG Systems** for additional help on PEM files

3. Using the tool of your choice, create a key pair, and generate a certificate request to submit to your CA. Your private key may be encrypted with a passphrase if desired. Remember to keep your private key secure. Unless your key was encrypted with a passphrase when you created the key you should never distribute it, store it, nor transfer it via a non-encrypted method (such as copying via a file share or internet email). Create a PEM encoded file of your private key named **privatekey.pem** . Create a backup of your private key and store it somewhere safe, such as on a CD-ROM stored in locked location. Submit your certificate request to your CA using the methods specified by your CA administrator.

4. When you receive your signed certificate back from the CA, convert it to PEM format if necessary and rename to **cert.pem** . This file is safe to transfer in a non-secure method.

5. Copy the **cert.pem** and **privatekey.pem** files to storage medium you can securely transfer (such as CD-ROM or USB Key) to a computer that you can create a private network between it and the system to be configured. Delete any copies of **privatekey.pem** files left on the computer to not leave any unsecure copies of your private key.

6. Copy the **rootcerts.pem** file to the same media where you saved the other PEM files

The remaining steps should be performed on a computer you can create a private network between it and the system to be configured. Since the default certificate installed on the system cannot be truly verified, it should not be trusted to encrypt the transmission of sending your new private key to the system over a non-secured network, even using HTTPS. For future updates, if a trusted certificate is already installed on the system, the loading of the certificates can be done over a non-secure network rather than building a private network by substituting HTTPS for HTTP and SSH for telnet in the remaining steps below.

7. Create a private network between a codec and a PC. The simplest way to do this is to use an Ethernet Switch and connect only the system to be configured and the computer used to configure it.

   a. Configure the codec using the on-screen menus to use a statically assigned IP address of 192.168.1.2 and a subnet mask of 255.255.255.0. Reboot the codec to have the changes take effect.

   b. Configure your computer with a statically assigned IP address of 192.168.1.3 and a subnet mask of 255.255.255.0. DNS servers and gateway addresses are not needed at this time in the computer or system to be configured.

   c. Verify you can communicate to the system from the computer by opening a web browser and entering http://192.168.1.2 - the webpage of the system should start to load and you will be prompted for a username and password.

8. Open the web interface of the system by opening a web browser on the computer and entering http://192.168.1.2 . When prompted for a login, enter anything for the username and the default password is TANDBERG

9. Navigate to the Certificate Management page *System Configuration* > *Certificate Management*

10. Click the *Browse* button next to Root Certificate to specify the Root Certificate file to upload to the system. Insert your secure media where the **rootcerts.pem** file is located and select that file and hit OK.

11. Click the *Browse* button next to HTTPS Certificate to specify the HTTPS certificate file to upload to the system. Insert your secure media where the **cert.pem** file is located and then select that file and hit OK

12. Click the *Browse* button next to Private Key to specify the PEM file containing your private key to upload to the system. Insert your secure media where the **privatekey.pem** file is located and then select that file and hit OK
    a. If your private key was encrypted with a passphrase, enter it in the passphrase field

13. Click *Upload* to transfer files to the system

14. Open a telnet session to the system using a telnet utility of your choice. In Windows you can use *Start Menu* > *Run* and enter **telnet 192.168.1.2** and hit OK

15. When prompted for a password, the default password is TANDBERG

16. After logging in, enter the following commands individually

```
xconfig http mode: off

xconfig https mode: on

xconfig telnet mode: off

xconfig telnetchallenge mode: off

xconfig snmp mode: off

xconfig ftp mode: off

xconfig ssh mode: on          (<- Optional but recommended)

xconfig https vertifyservercertificate: on
```

17. If the system will be deployed on a static IP address or NTP settings are not provided by the DHCP server, configure the NTP server properties for the network the system will reside on (replace example IP below with actual NTP server address). This is required when using verify certificates.

```
xconfig ntp mode: on

xconfig ntp address: 1.pool.ntp.org
```

18. If the system will be deployed on a static IP address or DNS settings are not provided by the DHCP Server, configure the DNS server properties for the network the system will reside on (replace example IP below with actual DNS server address). This is required when using verify certificates.

xconfig ip dns server 1 address: 10.10.10.2

xconfig ip dns domain name: company.com

19. Configure the systemname of the system

    xconfig systemunit name: rm204-bld5

20. Configure the external services to be disabled. These will be enabled automatically by TMS if needed.

    xconfig externalmanager address: 127.0.0.1

    xconfig externalmanager protocol: https

    xconfig externalservices mode: off

    xconfig corporatedirectory mode: off

21. Configure the IP Password you wish to use with the system

    xconfig strictpassword: on

    xconfig systemunit password: superstrongpasswordhere

22. Reboot the codec by entering **xcom boot** in the telnet session. Wait until the codec has fully restarted.

23. Open a new web browser window and enter the address https://192.168.1.2 and the web page of the codec will start to open.

    NOTE: You will see SSL security warnings in your browser because the name of the server does not match that of the certificate because we did not access the system by its hostname

24. When prompted for login, enter anything as the username and your new password for the password.

25. Verify the current server certificate by viewing the certificate details page in your browser. In Internet Explorer 7, click on the Lock Icon and select View Certificate. In Firefox, double-click on the Lock icon.

The system is now locked down and is ready to be configured with the IP properties of the network where it is intended to be deployed. Use the system's onscreen menus, dataport, or web interface to configure the IP Configuration including IP Assignment, Address, Subnet Mask, and Gateway. The IP properties can be found in the following places:

Onscreen Menus - *Control Panel > Network > IP Settings*

Web - *System Configuration > IP*

Dataport – *under xconfiguration ip*

Once the system has been moved to the network where it will be deployed, verify the certificate and FQDN by opening your web browser to the address **http://rm204-bld5.company.com** and checking for any certificate errors. Your client must have the trusted root certificate installed for the client to automatically trust the system's certificate.

For ongoing upkeep, administrators must remember that root certificates and server certificates have a fixed valid date range. Certificates must be replaced before they expire to prevent communication failures, so an administrator should be aware of when certificates will expire and check periodically that no systems are using expired certificates.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using 'Enforce Management Settings' when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

4 TANDBERG MPS

Before attempting to deploy a Secure Management solution for MPS systems, it is critical that an administrator understands the fundamentals of certificates, TLS, and the MPS platform's functionality. Appendix A of this document provides primer information on TLS and certificates for those who are not familiar with those technologies and Appendix C provides step by step instructions for creating and working with certificates.

This chapter is part reference, and part implementation guide for the MPS system. The first half details methods and protocols in use between an MPS system and an external management system. The material is broken into sections explaining the default methods implemented in MPS systems, followed by the functionality added beginning with J4.2 software to increase the security level. These sections should be read and understood before attempting to deploy MPS systems in a Secure Management solution.

The second half of this chapter provides a readiness checklist for an administrator to verify they have the information and network configuration required to configure MPS systems for Secure Management. This is followed by comprehensive step-by-step Best Practice guides to configuring an MPS system for Secure Management.

MPS Secure Management Reference Material

Default Protocols and Services in MPS Systems

The TANDBERG MPS system supports many of the same protocols and services that the MXP systems support. The description and behavior of the MPS will be very similar to the MXP with only minor changes due to components the MPS system does not require.

TANDBERG MPS systems support multiple interfaces for managing the system, including

- HTTP
- SSH
- Serial Port
- HTTPS
- SNMP
- Telnet
- SCP

In the factory default configuration, all of the above protocols are enabled except for HTTPS and SSH. These protocols may be enabled or disabled via the XML interface, the web administrator interface under *System Configuration > Misc* , or using the dataport and xconfiguration commands

```
xconfiguration https mode: <on/off>
xconfiguration http mode: <on/off>
xconfiguration ssh mode: <on/off>
xconfiguration telnet mode: <on/off>
xconfiguration snmp mode: <on/off>
```

Turning a service on or off requires a reboot for changes to take effect

For services that require a server certificate, such as HTTPS, the systems come preconfigured with a X.509 certificate signed by a TANDBERG Certificate Authority. By default this certificate authority will not be trusted by a user's computer resulting in a certificate warning in their web browser unless they specify to ignore it or trust the TANDBERG Certificate Authority.

Additionally, MPS devices have an internal client used to connect to management platforms. This client is used for External Manager. The client by default use HTTP to connect to the configured management address. This service can be configured from several locations in the XML interface, web interface, or dataport. The most direct way to configure this service is through the dataport where each service has their own list of settings under them

xconfigure externalmanager

This client is used when the system is managed by a TANDBERG TMS system, but is optional in stand-alone implementations.

MPS systems also implement a feedback system which can be used by external systems to monitor activity of the device by having the MPS system post changes using HTTP to an external URL. This feedback system is configured using the XML interface or **xcommand feedbackregister** series of commands available in the dataport of the MPS or via XML.

The existing services/protocols can be summarized based on the security levels outlined in the Introduction section of this document as follows:

Service/Protocol	Security Level
HTTP Server	Level 2
HTTPS Server	Level 4 (if client checks certificate)
Telnet	Level 1
SNMP	Level 1
SNMP Traps	Level 1
SSH	Level 3
SCP	Level 3
External Services/feedback	Level 2

The overall management security of the device can be improved by disabling unneeded services, and increasing the security of the services in use. Previously, when using TMS, administrators were limited in what services they could disable. Now, when combined with Secure-Only settings in TMS, administrators can achieve a Level 3 security by using the security settings added beginning with J4.2 or a Level 4 security when combined with the verify certificates settings.

Secure Management Features Added Beginning with J4.2 Software

Starting with software release J4.2, the security of management interfaces has been increased, including the ability to achieving a security of Level 4, through the following new additions:

Uploading Root Certificates

To properly handle X.509 certificates when acting as a HTTPS client, systems now support loading a list of trusted root certificates into the system. The system allows the upload of a PEM encoded file which contains the root certificates of any CAs you wish the system to trust. The system will use these root certificates when validating X.509 certificates presented by HTTPS servers the system connects to.

The system supports uploading of a single PEM encoded file, which may contain multiple certificates. If an existing list of certificates is present, uploading a new PEM encoded file will overwrite the existing list. Certificate lists are uploaded using the web interface of the system under *System Configuration > Certificate Management*

The system supports X.509 root certificates using RSA and DSA keys that are encoded in the PEM base-64 format. Root certificates are not encrypted with passphrases. For additional help with certificate formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document.

Uploading Server Certificates

Uploading a server certificate to use for the HTTPS server was introduced in J4.1 software, but is covered here for completeness. An uploaded certificate replaces the factory default certificate and is used for the HTTPS server of the system. A certificate is uploaded through the web interface of the system under *System Configuration > Certificate Management*. The certificate to be uploaded must be a PEM encoded file. If the private key is encrypted with a passphrase, the passphrase must be supplied when uploading the certificate in the separate Passphrase field. For additional help with certificate formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document.

HTTPS for External Services

ExternalManager service can now be configured to use either HTTP or HTTPS. The setting is controlled with a new command **protocol** under the service's configuration.

xconfiguration externalmanager protocol: <HTTP/HTTPS>

When protocol is set to HTTPS, only HTTPS will be used. The system intentionally will not fall back to HTTP if HTTPS fails. The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

HTTPS for Feedback URLs

Feedback URLs can now be defined to use HTTPS in the URL. If the feedback URL registered in the system starts with HTTPS, HTTPS will be used when posting events to the external server. The syntax of the **xconfiguration feedbackregister** command was not modified, only updated that using a https:// formatted URL is now valid. When the URL is set to HTTPS, only HTTPS will be used (system will not fall back to HTTP if HTTPS fails). The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

Verify Server Certificates Mode

To achieve Level 4 security, a system must not only handle certificates, it must validate them and adhere to any warnings or errors discovered. Enforcing certificate checking may not be desirable for organizations that want the encryption offered by TLS but do not wish to manage and deploy valid server certificates on all devices. Using TLS without validating certificates results in an equivalent Level 3 security. To address both needs, a new system setting is defined to control the handling of X.509 certificates

xconfiguration https verifyservercertificate: <on/off>

When the system is acting as a HTTPS client to connect to an external server, the server will provide a X.509 certificate to validate its identity. If VerifyServerCertificate is enabled, the server certificate will be validated before a connection continues. The system checks attributes of the Certificate such as:

- certificate is complete and any checksums are valid
- the current time is within the certificates valid date range
- the common name matches the hostname name used to access the server
- certificate has been signed by a certificate authority the system trusts through its root certificate

If any of the certificate checks fail - the HTTPS connection will fail and the communication is halted to prevent sharing information with an unsecure party.

If VerifyServerCertificate is disabled, the HTTPS client will accept any complete server certificate presented by the server and ignore any warnings or errors allowing the communication to continue.

Additional Settings Required for VerifyServerCertificate

When VerifyServerCertificate is enabled, the system must be able to perform its validation checks including hostname and date checks. Therefore the system's configuration must include:

- NTP must be configured and active in the system
- a DNS Server must be defined in the system
- Server Addresses must be entered as hostnames, not IP Addresses

Certificate Dataport Command

A new command has been added to the system's dataport which allows the viewing or deleting the system's currently loaded server certificate or root certificate. The command's syntax is

```
certificate <list>/<del> [<cert/root>]
```

The command will list the contents of the uploaded certificate used for the system's web server or root certificate. The output can be copied and ran through OpenSSL tools if desired to validate their contents. The command can also be used to delete either certificate if needed. If replacing certificates, there is no need to delete them first, as they can be overwritten by uploading a new certificate via the web interface of the system.

Interpreting Certificate Failure Errors

For troubleshooting purposes, if an administration needs to see the specific error code that caused a certificate to be rejected when the system connects to a server, you must enable syslog in the system prior to performing a server request. The error code for the certificate will be displayed and the error codes may be referenced in the OpenSSL documentation

Example:

1. Telnet or SSH to the system
2. Enable syslog by entering **syslog on**
3. Initiate a call from the web interface to another system to generate feedback and initiate a server connection
4. In the dataport, the logging output will include a line starting with SSL like the following

```
SSL: socklib_sslHandshake:
VerifyResult=14=ERROR_IN_CERT_NOT_AFTER_FIELD, NameCheck: Match
```

5. Stop syslog by entering **syslog off**

The error code is listed as VerifyResult=14=ERROR_IN_CERT_NOT_AFTER_FIELD which means the date on the server certificate was expired. SSL Error codes can be looked up in the OpenSSL documentation

MPS Secure Management Profiles

With the additional features added beginning with J4.2, MPS systems can operate in a full Level 3 or Level 4 security configuration.

Level 3 is available by

- Disabling all services except for HTTPS and SSH
- Configuring feedback and client services (external manager) to use HTTPS protocol

Level 4 is available by

- Disabling all services except for HTTPS
- Configuring feedback and client services (external manager) to use HTTPS protocol
- Installing a valid Server Certificate on the system
- Installing root certificates to trust and enabling Verify Certificates mode

The Implementation Guides included in this chapter provide step by step best instructions for deploying systems in either of these security profiles.

MPS Secure Management Implementation Material

Readiness Checklist for implementing HTTPS Management for MPSs

This list is intended to provide a quick reference to the information that an administrator must have available before implementing HTTPS management. Failure to organize this information prior to implementation will hinder or prevent you from deploying this functionality.

If not using verify certificates

- Each system must be running J4.2 software or newer
- You must have an administrative logon to the MPS systems
- Be familiar with the HTTPS related commands outlined in the MPS reference section of this document
- If a management tool besides TMS, your external management platform or tools must support HTTPS connections
- If using TMS to manage the endpoints, TMS must be version 11.9.1 or newer, and you must have an administrative logon to TMS

If using the verify certificates mode, the following ADDITIONAL items apply

- Each system must have a fully qualified domain hostname (FQDN) that points to the system's IP Address. The FQDN is defined in your DNS Server
- You must have a NTP source that systems can be pointed at for time synchronization. See system command xconfiguration NTP
- Each endpoint must have a DNS lookup server defined in its configuration. See system command xconfiguration IP DNS
- Each system will require its own Server Certificate be generated
- You must have the Private Key, Server Certificate, and passphrase (if used) for each system. Certificates and Keys must be in PEM encoded files
- You must have the public certificate of the Root Certificate Authority that will be used to verify the Server Certificates used on the network
- If using TMS, TMS must have a FQDN configured, and its own Server Certificate

For information on how to generate Server Certificates or related information, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. Once the steps outlined in the checklist are complete and the information is gathered, you can configure your systems to use HTTPS. The next sections outline the steps required to configure an MPS system in Level 3 or Level 4 security.

Implementation Guide for Configuring an MPS for Level 3 Security

The following provides an example of best practices for configuring a default MPS system for Level 3 security, where the system will only support communicating with management systems that use HTTPS but certificates are not verified, reducing the complexity and requirements of the deployment. If preparing a system for use with TMS, these steps should be performed either **before** adding the system to TMS or before enabling Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

These steps assume you already have the device configured on the IP network and you know the IP Address of the system. Replace <systemIP> in the examples below with the actual IP address of the system

1. Open a telnet session to the system using a telnet utility of your choice. In Windows you can use *Start Menu* > *Run* and enter **telnet <systemIP>** and hit OK
2. When prompted for login, use admin as the username and the default password is TANDBERG
3. Enter the following commands individually

   xconfig http mode: off

   xconfig https mode: on

   xconfig telnet mode: off

   xconfig snmp mode: off

   xconfig ssh mode: on (<- Optional but recommended)

   xconfig https verifyservercertificates: off

4. Configure the external services to be disabled. These will be enabled automatically by TMS if needed.

   xconfig externalmanager address: 127.0.0.1

   xconfig externalmanager protocol: https

5. Configure the IP Password you wish to use with the system

   xconfig systemunit password: superstrongpasswordhere

6. Reboot the system by entering **xcom boot** in the telnet session. Wait until the system has fully restarted.
7. Verify the changes. Open a new web browser window and enter the address https://<systemIP> and the web page of the system will start to open.

   NOTE: You will see SSL security warnings in your browser because the certificate authority is not trusted and the name of the server does not match that of the certificate

8. When prompted for login, enter anything as the username and your new password for the password.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using 'Enforce Management Settings' when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

Implementation Guide for Configuring an MPS for Level 4 Security

The following provides an example of best practices for configuring a default MPS system for Level 4 security, where the system will only support communicating with management systems that use HTTPS and have valid X.509 certificates. If preparing a system for use with TMS, these steps should be performed **before** enabling the Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

1. Decide the hostname that will be used to manage the system.

   Example: **mps-204.company.com**

2. Decide which certificate authority you will use to sign your certificates and obtain the public root certificate for that CA. Convert that file if necessary to PEM file format and save as **rootcerts.pem**

   If multiple certificate authorities need to be trusted by the system, obtain the root certificate for each and catonate each into a single PEM file. See **Appendix C – Creating and Working with Certificates for TANDBERG Systems** for additional help on PEM files

3. Using the tool of your choice, create a key pair, and generate a certificate request to submit to your CA. Your private key may be encrypted with a passphrase if desired. Remember to keep your private key secure. Unless your key was encrypted with a passphrase when you created the key you should never distribute it, store it, nor transfer it via a non-encrypted method (such as copying via a file share or internet email). Create a PEM encoded file of your private key named **privatekey.pem** . Create a backup of your private key and store it somewhere safe, such as on a CD-ROM stored in locked location. Submit your certificate request to your CA using the methods specified by your CA administrator.

4. When you receive your signed certificate back from the CA, convert it to PEM format if necessary and rename to **cert.pem** . This file is safe to transfer in a non-secure method.

5. Copy the **cert.pem** and **privatekey.pem** files to storage medium you can securely transfer (such as CD-ROM or USB Key) to a computer that you can create a private network between it and the system to be configured. Delete any copies of **privatekey.pem** files left on the computer to not leave any unsecure copies of your private key.

6. Copy the **rootcerts.pem** file to the same media where you saved the other PEM files

The remaining steps should be performed on a computer you can create a private network between it and the system to be configured. Since the default certificate installed on the system cannot be truly verified, it should not be trusted to encrypt the transmission of sending your new private key to the system over a non-secured network, even using HTTPS. For future updates, if a trusted certificate is already installed on the system, the loading of the certificates can be done over a non-secure network rather than building a private network by substituting HTTPS for HTTP and SSH for telnet in the remaining steps below.

7. Create a private network between the MPS and a PC. The simplest way to do this is to use an Ethernet Switch and connect only MPS to be configured and the computer used to configure it. Connect the MPS using the Ethernet Port on the front of the system controller card.
   a. Configure the MPS using the LCD panel on the front of the system by entering the *Parameter Config* > *SC Config* and configure the IP Address to 192.168.1.2 and the subnet mask of 255.255.255.0. Reboot the system to have the changes take effect.
   b. Configure your computer with a statically assigned IP address of 192.168.1.3 and a subnet mask of 255.255.255.0. DNS servers and gateway addresses are not needed at this time in the computer or system to be configured.

   c. Verify you can communicate to the system from the computer by opening a web browser and entering http://192.168.1.2 - the webpage of the system should start to load and you will be prompted for a username and password

8. Open the web interface of the system by opening a web browser on the computer and entering http://192.168.1.2 . When prompted for a login, enter anything for the username and the default password is TANDBERG

9. Navigate to the Certificate Management page *System Configuration* > *Certificate Management*

10. Click the *Browse* button next to Root Certificate to specify the Root Certificate file to upload to the system. Insert your secure media where the **rootcerts.pem** file is located and select that file and hit OK.

11. Click the *Browse* button next to HTTPS Certificate to specify the HTTPS certificate file to upload to the system. Insert your secure media where the **cert.pem** file is located and then select that file and hit OK

12. Click the *Browse* button next to Private Key to specify the PEM file containing your private key to upload to the system. Insert your secure media where the **privatekey.pem** file is located and then select that file and hit OK

    a. If your private key was encrypted with a passphrase, enter it in the passphrase field

13. Click *Upload* to transfer files to the system

14. Open a telnet session to the system using a telnet utility of your choice. In Windows you can use *Start Menu* > *Run* and enter **telnet 192.168.1.2** and hit OK

15. When prompted for login, use admin as the username and the default password is TANDBERG

16. Enter the following commands individually

```
xconfig http mode: off

xconfig https mode: on

xconfig telnet mode: off

xconfig snmp mode: off

xconfig ssh mode: on          (<- Optional but recommended)

xconfig https verifyservercertificates: on
```

17. Configure the external services to be disabled. These will be enabled automatically by TMS if needed.

```
xconfig externalmanager address: 127.0.0.1

xconfig externalmanager protocol: https
```

18. Configure the NTP server properties for the network the system will reside on (replace example IP below with actual NTP server address)

```
xconfig ntp address: 1.pool.ntp.org
```

19. Configure the DNS server properties for the network the system will reside on (replace example IP below with actual DNS server address). This is required when using verify certificates.

```
xconfig ip dns server 1 address: 10.10.10.2

xconfig ip dns domain name: company.com
```

20. Configure the systemname of the system

```
xconfig systemunit name: mps-204
```

21. Configure the IP Password you wish to use with the system

    xconfig systemunit password: superstrongpasswordhere

22. Reboot the system by entering **xcom boot** in the telnet session. Wait until the system has fully restarted.

23. Verify the changes. Open a new web browser window and enter the address https://192.168.1.2 and the web page of the system will start to open.

    NOTE: You will see SSL security warnings in your browser because the name of the server does not match that of the certificate because we did not access the system by its hostname

24. When prompted for login, enter anything as the username and your new password for the password.

25. Verify the current server certificate by viewing the certificate details page in your browser. In Internet Explorer 7, click on the Lock Icon and select View Certificate. In Firefox, double-click on the Lock icon.

The system is now locked down and is ready to be configured with the IP properties of the network where it is intended to be deployed. Use the system's LCD Panel, dataport or web interface to configure the IP Configuration of the System Controller including IP Assignment, Address, Subnet Mask, and Gateway. Media blade or additional Ethernet port configurations can be done now or when the device is deployed on its intended network. The IP properties can be found in the following places:

LCD Panel – *Parameter Config > SC Config*

Web – *System Configuration > IP*

Dataport – *under xconfiguration ip*

Once the system has been moved to the network where it will be deployed, verify the certificate and FQDN by opening your web browser to the address **http:// mps-204.company.com** and checking for any certificate errors. Your client must have the trusted root certificate installed for the client to automatically trust the system's certificate.

For ongoing upkeep, administrators must remember that root certificates and server certificates have a fixed valid date range. Certificates must be replaced before they expire to prevent communication failures, so an administrator should be aware of when certificates will expire and check periodically that no systems are using expired certificates.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using 'Enforce Management Settings' when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

5 TANDBERG VCS

Before attempting to deploy a Secure Management solution for VCS systems, it is critical that an administrator understands the fundamentals of certificates, TLS, and the VCS platform's functionality. Appendix A of this document provides primer information on TLS and certificates for those who are not familiar with those technologies and Appendix C provides step by step instructions for creating and working with certificates.

This chapter is part reference, and part implementation guide for the VCS system. The first half details methods and protocols in use between an VCS system and an external management system. The material is broken into sections explaining the default methods implemented in VCS systems, followed by the functionality added beginning with X4.0 software to increase the security level. These sections should be read and understood before attempting to deploy VCS systems in a Secure Management solution.

The second half of this chapter provides a readiness checklist for an administrator to verify they have the information and network configuration required to configure VCS systems for Secure Management. This is followed by comprehensive step-by-step Best Practice guides to configuring a VCS system for Secure Management.

VCS Secure Management Reference Material

Default Protocols and Services in VCS Systems

The TANDBERG VCS system supports many of the same protocols and services that the MXP systems support. The description and behavior of the VCS will be very similar to the MXP with only minor changes due to components the VCS system does not require.

TANDBERG VCS systems support multiple interfaces for managing the system, including

- HTTP
- SSH
- Serial Port
- HTTPS
- SCP
- Telnet
- SNMP

In the factory default configuration, all of the above protocols are enabled except for Telnet. These protocols may be enabled or disabled via the XML interface, the web administrator interface under *System Configuration* > *System* and *System Configuration* > *SNMP*, or using the dataport and xconfiguration commands

```
xconfiguration administration https mode: <on/off>
xconfiguration administration http mode: <on/off>
xconfiguration administration ssh mode: <on/off>
xconfiguration administration telnet mode: <on/off>
xconfiguration snmp mode: <on/off>
```

Turning a service on or off requires a reboot for changes to take effect

For services that require a server certificate, such as HTTPS, the systems come preconfigured with a X.509 certificate signed by a TANDBERG Certificate Authority. By default this certificate authority will not be trusted by a user's computer resulting in a certificate warning in their web browser unless they specify to ignore it or trust the TANDBERG Certificate Authority.

Additionally, VCS devices have an internal client used to connect to management platforms. This client is used for External Manager. The client by default uses HTTP to connect to the configured management address. This service can be configured from several locations in the XML interface, web interface, or dataport. The most direct way to configure this service is via the web interface under *System Configuration* > *External Manager*. This client is used when the system is managed by a TANDBERG TMS system, but is optional in stand-alone implementations.

VCS systems also implement a feedback system which can be used by external systems to monitor activity of the device by having the VCS system post changes using HTTP to an external URL. This feedback system is configured using the XML interface or **xcommand feedbackregister** series of commands available in the dataport of the MPS or via XML.

The existing services/protocols can be summarized based on the security levels outlined in the Introduction section of this document as follows:

Service/Protocol	Security Level
HTTP Server	Level 2
HTTPS Server	Level 4 (if client checks certificate)
Telnet	Level 1
SNMP	Level 1
SSH	Level 3
SCP	Level 3
External Service/feedback	Level 2

The overall management security of the device can be improved by disabling unneeded services, and increasing the security of the services in use. Previously, when using TMS, administrators were limited in what services they could disable. Now, when combined with Secure-Only settings in TMS, administrators can achieve a Level 3 security by using the security settings added beginning with X4.0 or a Level 4 security when combined with the verify certificates settings.

Secure Management Features Added Beginning with X4.0 Software

Starting with software release X4.0, the security of management interfaces has been increased, including the ability to achieving a security of Level 4, through the following new additions:

HTTPS for External Services

ExternalManager service can now be configured to use either HTTP or HTTPS. The setting is controlled via the web interface under *System Configuration* > *External Manager* or with a new command **protocol** under the service's dataport configuration command:

```
xconfiguration externalmanager protocol: <HTTP/HTTPS>
```

When protocol is set to HTTPS, only HTTPS will be used. The system intentionally will not fall back to HTTP if HTTPS fails. The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

HTTPS for Feedback URLs

Feedback URLs can now be defined to use HTTPS in the URL. If the feedback URL registered in the system starts with HTTPS, HTTPS will be used when posting events to the external server. The syntax of the **xconfiguration feedbackregister** command was not modified, only updated that using a https:// formatted URL is now valid. When the URL is set to HTTPS, only HTTPS will be used (system will not fall back to HTTP if HTTPS fails). The management server must also be configured to support HTTPS. See the TANDBERG TMS section of this document for information on how TMS interacts with HTTPS and these settings.

Verify Server Certificates Mode

To achieve Level 4 security, a system must not only handle certificates, it must validate them and adhere to any warnings or errors discovered. Enforcing certificate checking may not be desirable for organizations that want the encryption offered by TLS but do not wish to manage and deploy valid server certificates on all devices. Using TLS without validating certificates results in an equivalent Level 3 security. To address both needs, a new system setting is defined to control the handling of X.509 certificates

xConfiguration ExternalManager Server Certificate Verify Mode: <off/on>

This setting is also configurable via the web interface under *System Configuration* > *External Manager*

When the system is acting as a HTTPS client to connect to an external server, the server will provide a X.509 certificate to validate its identity. If Certificate Verify Mode is enabled, the server certificate will be validated before a connection continues. The system checks attributes of the Certificate such as:

- certificate is complete and any checksums are valid
- the current time is within the certificates valid date range
- the common name matches the hostname name used to access the server
- certificate has been signed by a certificate authority the system trusts through its root certificate

If any of the certificate checks fail - the HTTPS connection will fail and the communication is halted to prevent sharing information with an unsecure party.

If Certificate Verify Mode is disabled, the HTTPS client will accept any complete server certificate presented by the server and ignore any warnings or errors allowing the communication to continue.

Additional Settings Required for Certificate Verify Mode

When Server Certificate Verify is enabled, the system must be able to perform its validation checks including hostname and date checks. Therefore the system's configuration must include:

- NTP must be configured and active in the system
- a DNS Server must be defined in the system
- Server Addresses must be entered as hostnames, not IP Addresses

Related Existing Commands

Previous VCS software releases included functionality that is also relevant to Secure Management. These features include uploading of Server Certificates, Private Keys, and the uploading of Root Certificates to trust. These functionalities are on the *Maintenance* > *Security* page within the VCS's web interface.

Root Certificates

Uploading a list of trusted root certificates into the system allows the upload of a PEM encoded file which contains the root certificates of any CAs you wish the system to trust. The system will use these root certificates when validating X.509 certificates presented by HTTPS servers the system connects to.

The system supports uploading of a single PEM encoded file, which may contain multiple certificates. If an existing list of certificates is present, uploading a new PEM encoded file will overwrite the existing list. Certificate lists are uploaded using the web interface of the system under *Maintenance* > *Security* . These certificates are also used by the SIP functionality of the VCS.

The system supports X.509 root certificates using RSA and DSA keys that are encoded in the PEM base-64 format. Root certificates are not encrypted with passphrases. For additional help with certificate formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document.

Server Certificates and Private Key

Uploading a Server Certificate replaces the factory default certificate and is used for the HTTPS server of the system. A certificate is uploaded through the web interface of the system under *Maintenance* > *Security*. The certificate to be uploaded must be a PEM encoded file. The VCS does not currently support Private Keys that are encrypted using passphrases. For additional help with certificate formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document.

VCS Secure Management Profiles

With the additional features added beginning with X4, VCS systems can operate in a full Level 3 or Level 4 security configuration.

Level 3 is available by

- Disabling all services except for HTTPS and SSH
- Configuring feedback and client services (external manager) to use HTTPS protocol

Level 4 is available by

- Disabling all services except for HTTPS
- Configuring feedback and client services (external manager) to use HTTPS protocol
- Installing a valid Server Certificate on the system
- Installing root certificates to trust and enabling Verify Certificates mode

The Implementation Guides included in this chapter provide step by step best instructions for deploying systems in either of these security profiles.

VCS Secure Management Implementation Material

Readiness Checklist for implementing HTTPS Management for VCSs

This list is intended to provide a quick reference to the information that an administrator must have available before implementing HTTPS management. Failure to organize this information prior to implementation will hinder or prevent you from deploying this functionality.

If not using verify certificates

- Each system must be running X4.0 software or newer
- You must have an administrative logon to the VCS systems
- Be familiar with the HTTPS related commands outlined in the VCS reference section of this document
- If a management tool besides TMS, your external management platform or tools must support HTTPS connections
- If using TMS to manage the endpoints, TMS must be version 11.9.1 or newer, and you must have an administrative logon to TMS

If using the verify certificates mode, the following ADDITIONAL items apply

- Each system must have a fully qualified domain hostname (FQDN) that points to the system's IP Address. The FQDN is defined in your DNS Server
- You must have a NTP source that systems can be pointed at for time synchronization. See system command **xconfiguration NTP**
- Each endpoint must have a DNS lookup server defined in its configuration. See system commands **xconfiguration IP DNS**
- Each system will require its own Server Certificate be generated
- You must have the Private Key, Server Certificate, and passphrase (if used) for each system. Certificates and Keys must be in PEM encoded files
- You must have the public certificate of the Root Certificate Authority that will be used to verify the Server Certificates used on the network
- If using TMS, TMS must have a FQDN configured, and its own Server Certificate

For information on how to generate Server Certificates or related information, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. Once the steps outlined in the checklist are complete and the information is gathered, you can configure your systems to use HTTPS. The next sections outline the steps required to configure an VCS system in Level 3 or Level 4 security.

Implementation Guide for Configuring a VCS for Level 3 Security

The following provides an example of best practices for configuring a default VCS system for Level 3 security, where the system will only support communicating with management systems that use HTTPS but certificates are not verified, reducing the complexity and requirements of the deployment. If preparing a system for use with TMS, these steps should be performed either **before** adding the system to TMS or before enabling Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

These steps assume you already have the device configured on the IP network and you know the IP Address of the system. Replace <systemIP> in the examples below with the actual IP address of the system

1. Using your web browser, open a connection to the VCS by entering the URL https://<systemIP>

   NOTE: You will see SSL security warnings in your browser because the certificate authority is not trusted and the name of the server does not match that of the certificate

2. On the Logon Page, click on the Administrative Logon Button. When prompted for login, use admin as the username and the default password is TANDBERG

3. Disable unnecessary services. Browse to *System Configuration* > *System*. Using the drop down boxes, set Enabled to OFFTelnet and HTTP to OFF, and HTTPS and SSH to ON (Optional, but recommended). Click the Save button to save the changes.

4. Disable SNMP. Browse to *System Configuration* > *SNMP*. Using the drop down boxes, set Enabled to OFF . Click the Save button to save the changes

5. Configure the external services to be disabled. These will be enabled automatically by TMS if needed. Browse to *System Configuration* > *External Manager.* Configure

   - Address field to 127.0.0.1
   - Protocol to HTTPS
   - Certificate Verify Mode to Off

   Click the Save button to save the changes.

6. Configure the administrator password you wish to use with the system. Browse to *Maintenance* > *Administration* > *Administration Accounts.* Click on the *View/Edit* link for the admin account. Enter a strong complex password, confirm it, and click the Save button.

7. Reboot the system by Browsing to *Maintenance* > *Restart* . Wait until the system has fully restarted.

8. Verify the changes. Open a new web browser window and enter the address https://<systemIP> and the web page of the system will start to open.

   NOTE: You will see SSL security warnings in your browser because the certificate authority is not trusted and the name of the server does not match that of the certificate

9. On the Logon Page, click on the Administrative Logon Button. When prompted for login, use admin as the username and the password you created.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using ‘Enforce Management Settings’ when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

Implementation Guide for Configuring a VCS for Level 4 Security

The following provides an example of best practices for configuring a default VCS system for Level 4 security, where the system will only support communicating with management systems that use HTTPS and have valid X.509 certificates. If preparing a system for use with TMS, these steps should be performed **before** enabling the Secure-Only mode in TMS. For any help creating certificates or converting file formats, please see Appendix C – Creating and Working with Certificates for TANDBERG Systems in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

1. Decide the hostname that will be used to manage the system.

   Example: **vcs-ext.company.com**

2. Decide which certificate authority you will use to sign your certificates and obtain the public root certificate for that CA. Convert that file if necessary to PEM file format and save as **rootcerts.pem**

   If multiple certificate authorities need to be trusted by the system, obtain the root certificate for each and catonate each into a single PEM file. See Appendix C – Creating and Working with Certificates for TANDBERG Systems for additional help on PEM files

3. Using the tool of your choice, create a key pair, and generate a certificate request to submit to your CA. Your private key must NOT be encrypted with a passphrase to be compatible with the VCS. Remember to keep your private key secure. Since your key was created without a passphrase you should never distribute it, store it, nor transfer it via a non-encrypted method (such as copying via a file share or internet email). Create a PEM encoded file of your private key named **privatekey.pem** . Create a backup of your private key and store it somewhere safe, such as on a CD-ROM stored in locked location. Submit your certificate request to your CA using the methods specified by your CA administrator.

4. When you receive your signed certificate back from the CA, convert it to PEM format if necessary and rename to **cert.pem** . This file is safe to transfer in a non-secure method.

5. Copy the **cert.pem** and **privatekey.pem** files to storage medium you can securely transfer (such as CD-ROM or USB Key) to a computer that you can create a private network between it and the system to be configured. Delete any copies of **privatekey.pem** files left on the computer to not leave any unsecure copies of your private key.

6. Copy the **rootcerts.pem** file to the same media where you saved the other PEM files

The remaining steps should be performed on a computer you can create a private network between it and the system to be configured. Since the default certificate installed on the system cannot be truly verified, it should not be trusted to encrypt the transmission of sending your new private key to the system over a non-secured network, even using HTTPS. For future updates, if a trusted certificate is already installed on the system, the loading of the certificates can be done over a non-secure network rather than building a private network by substituting HTTPS for HTTP and SSH for telnet in the remaining steps below.

7. Create a private network between the VCS and a PC. The simplest way to do this is to use an Ethernet Switch and connect only VCS to be configured and the computer used to configure it. Connect the VCS using the LAN Port 1 on the front of the appliance.
   a. If the VCS is new or defaulted, it will have a default IP Address of 192.168.0.100. If the unit is already configured with another IP address, complete the initial configuration wizard described on the VCS Installation Sheet to configure the VCS to be IP Address 192.168.0.100 with a subnet mask of 255.255.255.0 and default gateway of 192.168.0.1. Reboot the system at the end of the wizard to have the changes take effect.
   b. Configure your computer with a statically assigned IP address of 192.168.0.101 and a subnet mask of 255.255.255.0. DNS servers and gateway addresses are not needed at this time in the computer or system to be configured.

c. Verify you can communicate to the system from the computer by opening a web browser and entering https://192.168.0.100 - the webpage of the system should start to load and you should see the Logon page

8. Open the web interface of the system by opening a web browser on the computer and entering https://192.168.0.100. When the Login Page opens, click the Administrator Login button, and then enter admin as the username and the default password is TANDBERG

9. Navigate to the Certificate Management page *Maintenance* > *Security*

10. Under *Trusted CA Certificate*, Click the *Browse* button to specify the Root Certificate file to upload to the system. Insert your secure media where the **rootcerts.pem** file is located and select that file and hit OK. Click the *Upload CA certificate* button to upload the file to the system.

11. Under *Server Certificate Data*, Click the *Browse* button next to Select server private key. Insert your secure media where the **privatekey.pem** file is located and then select that file and hit OK

12. Under Server Certificate Data, Click the *Browse* button next to server certificate file Insert your secure media where the **cert.pem** file is located and then select that file and hit OK

13. Click *Upload* server certificate data to transfer files to the system. The certificate changes will take effect immediately on the system.

14. Disable unnecessary services. Browse to *System Configuration* > *System*. Using the drop down boxes, set Enabled to OFF for Telnet and HTTP. Set Enabled to ON for HTTPS and SSH (Optional, but recommended). Enter the name for the system in the System Name field. Example: **vcs-ext** .Click the Save button to save the changes.

15. Disable SNMP. Browse to *System Configuration* > *SNMP*. Using the drop down boxes, set Enabled to OFF . Click the Save button to save the changes

16. Configure the external services to be disabled. These will be enabled automatically by TMS if needed. Browse to *System Configuration* > *External Manager.* Configure

    - Address field to 127.0.0.1
    - Protocol to HTTPS
    - Certificate Verify Mode to Off

    Click the Save button to save the changes.

17. Configure the administrator password you wish to use with the system. Browse to *Maintenance* > *Administration* > *Administration Accounts.* Click on the *View/Edit* link for the admin account. Enter a strong complex password, confirm it, and click the Save button.

18. Configure the NTP server for the network the system will reside on by browsing to *System* > *Time* and configuring the NTP Server address with a valid NTP Server for your network. Click the Save button to save the changes.

19. Configure the DNS server properties for the network the system will reside on by Browsing to *System* > *DNS*. Configure at least the first DNS Server address field with a valid DNS server for your network. Click the Save button to save the changes. DNS is required when using verify certificates.

20. Reboot the system by entering browsing to *Maintenance* > *Restart* and chose to restart the system. Wait until the system has fully restarted.

21. Verify the changes. Open a new web browser window and enter the address https://192.168.0.100 and the web page of the system will start to open.

    NOTE: You will see SSL security warnings in your browser because the name of the server does not match that of the certificate because we did not access the system by its hostname

22. On the Login Page, click on the Administrative Logon Button. When prompted for login, use admin as the username and the password you created.

23. Verify the current server certificate by viewing the certificate details page in your browser. In Internet Explorer 7, click on the Lock Icon and select View Certificate. In Firefox, double-click on the Lock icon.

The system is now locked down and is ready to be configured with the IP properties of the network where it is intended to be deployed. Use the system's serial port, dataport or web interface to configure the IP Configuration of the device including IP Assignment, Address, Subnet Mask, and Gateway. The IP properties can be found in the following places:

Serial Port – *Refer to VCS Installation Sheet*

Web – *System Configuration > IP*

Dataport – *under xconfiguration ip*

Once the system has been moved to the network where it will be deployed, verify the certificate and FQDN by opening your web browser to the address https://vcs-ext.company.com and checking for any certificate errors. Your client must have the trusted root certificate installed for the client to automatically trust the system's certificate.

For ongoing upkeep, administrators must remember that root certificates and server certificates have a fixed valid date range. Certificates must be replaced before they expire to prevent communication failures, so an administrator should be aware of when certificates will expire and check periodically that no systems are using expired certificates.

The system is now locked down and is ready to be added to TMS if needed. TMS will automatically configure the external services and feedback addresses using 'Enforce Management Settings' when the system is added to TMS or when Secure-Only is enabled in TMS. If TMS is to be used in the Secure Management solution, continue on to the TMS chapter of this document to complete the configuration of the total solution.

TANDBERG TMS

Before attempting to deploy a Secure Management solution for using TANDBERG Management Suite (TMS), it is critical that an administrator understands the fundamentals of certificates, TLS, how TMS communicates with devices and the related TMS security functionalities. Appendix A of this document provides primer information on TLS and certificates for those who are not familiar with those technologies and Appendix B provides step by step instructions for creating and working with certificates for TMS.

This chapter is part reference, and part implementation guide for a TMS server. The first half details methods and protocols in use between TMS and external systems. The material is broken into sections explaining the default methods implemented in TMS, followed by the functionality added beginning with TMS version 11.9.1 software to increase the security level. These sections should be read and understood before attempting to deploy TMS and systems in a Secure Management solution.

The second half of this chapter provides a readiness checklist for an administrator to verify they have the information and network configuration required to configure TMS for Secure Management. This is followed by comprehensive step-by-step Best Practice guides to configuring TMS for Secure Management.

TMS Secure Management Reference Material

Default Protocols and Services and Communication used with TMS

The following sections will describe the methods and security available by default when using TMS with different external systems, including devices, databases, software integrations, and users.

TMS Interaction with Devices

The TANDBERG Management Suite (TMS) uses multiple protocols to communicate with the devices and software packages it interacts with. In general TMS communicates with systems and external integrations using the following protocols

- HTTP
- FTP
- Vendor Specific Protocols
- HTTPS
- SNMP
- Telnet
- SNMP Traps

Which protocol is used for a specific device or task is generally dictated by what is supported by the device. Example, TMS would prefer to use HTTPS to communicate with an endpoint, but the endpoint may only support telnet and SNMP for management interfaces. When multiple choices are available, TMS will generally use the more secure of the methods available. Because TMS interoperates with existing multiple devices types and vendors, TMS cannot dictate the protocols to use when communicating with systems or require protocols or methods to achieve the highest level of security.

Connections from TMS will use secure protocols when possible. Example, if a TANDBERG MXP codec has HTTPS enabled, TMS will use HTTPS when connecting to the codec. Other systems that may not support HTTPS, may support challenge response on their login methods. Example, a TANDBERG Classic codec does not support HTTPS, but does support telnet challenge which uses a MD5 challenge response system so that the system's password is not sent over the network. Other HTTP systems may support digest as a method to exchange passwords.

Additionally systems must also open their own connections to TMS, such as when sending feedback to TMS about activity and may only use HTTP or SNMP Traps. Not all functions may be available via the secure interface, such as only being able to transfer files via FTP. Devices themselves do not have a set of user credentials to connect to TMS, but still must be able to open connections to TMS autonomously – this requires some interfaces always be open for anonymous connections.

Because of these restrictions TMS traditionally cannot be locked down and require only authenticated, encrypted connections into TMS to achieve an equivalent Level 4 security. TMS by default can deliver up to Level 2 security depending on the device types being managed and it is important to recognize that even though encrypted channels are not used all the time or in all cases, sensitive information such as passwords are protected when allowed by the device.

For additional details on which protocols are used with a specific piece of equipment, please see the TMS Product Support document available from the TANDBERG website.

TMS Interaction with Users

TMS by default only uses HTTP when communicating with users accessing TMS. All browsing and file transfers are done via HTTP. Note that TMS does offer links to systems that allows a user to interact with a system's management interfaces directly, but those links open new connections that do not go through TMS and are between the user and the system directly.

TMS is not setup for HTTPS when installed, however the IIS web server does support secure login methods such as NTLM, Kerberos, and digest authentication, so user logins to TMS will be secure as long as the client accessing TMS supports one of these methods.

TMS Interaction with the Database

TMS by default uses a locally install SQL 2005 database server which it interacts with using ADO.NET SQL Client. The SQL Server by default does not allow any remote connections and the communications between TMS and the database happen locally -- they do not go out over the network. The connection settings that TMS uses to connect to the database are encrypted on the local server.

When using a customer supplied external SQL server, TMS relies upon the security methods of the SQL Server and client to protect the SQL conversation over the network. SQL Server provides password encryption for the SQL Client connection if the SQL Server has a server certificate installed. Additionally, TMS supports using delegated SQL Users to minimize the exposure the remote user has to the SQL Server. Lastly, an administrator can enable SSL encryption for the entire connection if desired (**Note**: Requiring SSL will put increased CPU load on the SQL client and server). For additional information on password encryption and SSL within SQL Server, see the following Microsoft articles as a start point:

How SQL Server uses a certificate when the Force Protocol Encryption option is turned on - http://support.microsoft.com/kb/318605

How to enable SSL encryption for an instance of SQL Server 2005 by using Microsoft Management Console - http://support.microsoft.com/default.aspx?scid=kb;en-us;316898

How To: Connect to SQL Server Using SQL Authentication in ASP.NET 2.0 - http://msdn2.microsoft.com/en-us/library/ms998300.aspx#paght000010_step5

How to enable SSL for SQL Server 2000

http://support.microsoft.com/default.aspx?scid=kb;EN-US;Q276553

TMS Interaction with External Integrations

External Integrations such as the Microsoft Exchange Integration, typically communicate back to TMS via a centralized server, rather than having clients communicate directly with TMS. These integrations all support using HTTP or HTTPS to communicate to TMS. However, since TMS by default is not installed with HTTPS, the integrations will use HTTP and password encryption support will vary depending on the integration product being used.

TMS does not initiate outbound connections to external integrations with the exception of See&Share, which uses TLS on its connections, even when using the HTTP port 80.

Default TMS Communications Security Summary

The following table summarizes TMS's communications security with devices, users, and the database. The table reflects the maximum level with the existing TMS product. Actual levels can vary based on the systems being supported and SQL server configuration.

Service/Protocol	Security Level
User Interaction	Level 2
Device Interaction – HTTP	Level 2
Device Interaction – HTTPS	Level 3
Device Interaction – SNMP	Level 1
Device Interaction – SNMP Traps	Level 1
Device Interaction – Telnet	Level 1
Device Interaction – Telnet Challenge	Level 2
Device Interaction – FTP	Level 1
Device Interaction – Feedback	Level 1
Database Interaction	Level 4
External Integrations	Level 2

Secure Management Features Added Beginning with TMS v11.9.1 Software

To help strengthen the communications security available in TMS, TMS version 11.9.1 introduced HTTPS compatibility, removed the use of FTP for some MXP/MPS functions, and added a new Secure-Only mode to allow Level 3 or Level 4 security in TMS.

HTTPS Compatibility for Web Server

TANDBERG will now document and support configurations that will enable HTTPS on the TMS web server which allows fully encrypted connections to TMS by both users and devices. When enabled, TMS will support both HTTP and HTTPS connections for compatibility.

Removal of FTP use for MXP/MPS

Starting with TMS version 11.9.1, TMS no longer depends on FTP for TANDBERG MXP and MPS systems by migrating functionality that previously used FTP to HTTP/HTTPS for these system types.

Secure-Only Option in TMS Administrative Tools

Starting with TMS version 11.9.1, TMS added a new mode for customers looking to use only secure communication protocols between TMS and devices. When this mode is enabled, TMS will **only** use secure protocols (such as HTTPS) when communicating to supported devices, and those devices will only use HTTPS when opening connections to TMS. When enabled and used with compatible systems, Level 3 or Level 4 security can be achieved between TMS and devices.

This mode only changes behavior for devices that can support both HTTPS in to the device and from the device. These devices will be labeled as 'Secure-only supported devices' in this document. Devices that do not support Secure-only communications with TMS will continue to operate as they did previously. As of this document's writing, the supported devices are TANDBERG MXP endpoints, TANDBERG VCS and the TANDBERG MPS. Support does require minimum software versions on the devices as the behavior communicating **from** the device to TMS has changed as well. The minimum required versions of software for each device type are covered in the chapter corresponding to that device in this document.

Note: While previous versions of TANDBERG systems supported a HTTPS server for web management of the device, they did not support a HTTPS client for connecting **to** TMS and therefore are not supported in Secure-Only mode.

The setting has two states and an optional parameter.

Off - TMS will continue to use HTTPS to communicate to devices when possible, and instruct the devices to use HTTP when communicating to TMS. This is an equivalent Level 2 security when communicating with supported devices.

On - TMS will only use HTTPS to communicate to supported systems and instruct devices to use HTTPS when communicating to TMS. This is an equivalent Level 3 security when communicating with supported devices.

Verify Certificates – When enabled, TMS will validate certificates presented by devices and will not communicate if the certificates are not valid. Additionally, TMS will configure supported devices to do the same, by configuring the Verify Server Certificates setting in the device which requires they validate certificates presented before allowing communication. This is an equivalent Level 4 security when communicating with supported devices.

Security Level possibilities with TMS

With the additional features and changes available starting with TMS version 11.9.1, communications security when using TMS can be greatly enhanced for those organizations looking to enforce policy. By enabling HTTPS and Secure-Only on the TMS server

- Interaction with Users can be elevated to Level 3 or Level 4 security depending on their browser's certificate settings
- Interaction with devices that support the Secure-only mode can achieve Level 3 security, and if Verify Certificates mode is enabled, Level 4.
- Interaction with External Integrations can achieve Level 3 security or Level 4 if the remote system validates certificates
- Interaction with the Database can be as high as Level 4 security as it was previously

The implementation sections of this chapter provide specific instructions on how to configure TMS to achieve these security levels. It is important to note that for compatibility reasons, communications with devices that do not support the Secure-Only configurations are not changed. Therefore you total effective security level will depend on the device types managed, and may differ to different systems. See the implementation guides of this chapter for additional optional changes that can be made to restrict TMS to Secure-Only systems.

How does TMS change when Secure-Only Mode is enabled?

The following describes how the normal operation of TMS will change when the Secure-Only mode is enabled.

- The customer installs a server certificate on the IIS server which enables HTTPS in addition to HTTP. HTTPS becomes an optional method for connection to the TMS web server
- Systems types that support Secure-Only and are running at least the minimum version for Secure-Only are all treated as Secure-Only systems and must be configured to meet the requirements of Secure-Only
- TMS will instruct Secure-Only supported devices to use HTTPS when connecting to TMS and configure their management settings appropriately, including:
    - Externalmanager protocol

  - Externalservices protocol
  - Corporate Directory protocol
  - Feedback URL

- TMS will not fallback to HTTP if a HTTPS connection attempt to a Secure-Only supported device fails.
- The System Alive-Status scanner which uses a short interval SNMP scan to improve response time of the detection of systems going offline is disabled for Secure-Only supported systems (due to the use of SNMP). The scanner will still operate for systems that do not support Secure-Only communications and Secure-Only systems will still update based on normal feedback and watchdog scan intervals.
- TMS will not rely on SNMP queries or SNMP traps to Secure-Only supported devices
- TMS will not use FTP or telnet to Secure-Only supported devices
- TMS will not open tickets on Secure-Only supported devices that have their non-secure services disabled
- TMS will open a HTTPS Connection failure ticket if TMS cannot communicate to a Secure-Only supported system
- Required services enabled on Secure-Only supported devices are reduced so that only HTTPS must be enabled on the device (Telnet, SNMP, and FTP all may be disabled on device)
- Optionally users may connect to TMS using HTTPS rather than HTTP by using https://<servername> when connecting to TMS

If the Verify Certificates checkbox is enabled, the following additional changes also apply

- TMS sets the verify server certificates setting to on for Secure-Only supported devices when setting management settings on the system
- TMS will validate certificates supplied by HTTPS devices when communicating to them, and if the validation fails, it will open a new Certificate Validation Error ticket on the system and will not continue the communication
- By having their verify server certificate setting enabled, Secure-only supported devices communicating to TMS expect to receive a valid server certificate when connecting to TMS and will stop their communication if the server certificate provided does not pass their validation check
- TMS requires Secure-Only supported systems be tracked by hostname. TMS will automatically change supported systems to this method if a hostname is available in the system's Connection Settings.

External Integrations when Secure-Only is enabled

When Secure-Only is enabled, both HTTP and HTTPS are still available on the TMS server. Any external integration products pointed at the TMS server will continue to operate as normal since HTTP is still available. With HTTPS available, external integrations may use HTTPS if desired and their TMS Connection Settings are updated. The following TANDBERG integration products support HTTPS connections to TMS if configured to do so:

- Microsoft Exchange Integration
- 3rd Party Booking API
- Microsoft Live Communications Server Integration
- Microsoft SIP-CX Gateway

- IBM Lotus Notes Integration[2]
- IBM Lotus Sametime Integration
- TANDBERG See & Share

It is recommended to configure these products to use HTTPS to TMS if HTTPS is available on the TMS server as these integrations do use login credentials to access TMS.

Additional TMS Server Requirements when Secure-Only is enabled

When Secure-Only is enabled, the additional rules should be adhered to when operating TMS

- Secure-Only supported systems must be entered in TMS by hostname (rather than IP address) and should be tracked by hostname. Use the Connection Settings page for each device to update these settings if required. TMS will enforce this tracking method on supported system types as long as a hostname is available in the Connection Settings page for that device
- The TMS host server needs a valid DNS server configured
- The TMS host server should have a valid time server to keep its date accurate
- Override DNS under *Administrative Tools* > *Network Settings* must be set to No (default)
- TMS must have a valid DNS hostname associated to its IP address and must be entered in the *TMS Server Fully Qualified Hostname* fields under *Administrative Tools* > *Configuration* > *Network Settings*
- Because a website may only have one Server Certificate, all systems must use the same hostname to reach TMS. So the local and public TMS Server DNS Addresses must be the same and devices on both the public and local networks must be able to reach TMS via that hostname.

Turning Secure-Only Mode Off

Turning Secure-Only off will reverse TMS's behavior without impact, and systems will have their connection settings automatically reverted to use HTTP methods the next time Enforce Management Settings are applied to the system. Connectivity will not be disrupted while waiting for the Enforce Management Settings update unless HTTPS has been disabled on IIS server of TMS by disabling or removing the server certificate before all systems have been updated.

Systems may continue to use HTTPS for their own web servers as TMS supports HTTPS or HTTP connectivity **to** devices even with Secure-Only disabled in TMS.

[2] Requires Notes Integration version 11.1

TMS Secure Management Implementation Material

Readiness Checklist for implementing HTTPS Management for TMS

This list is intended to provide a quick reference to the information that an administrator must have available before implementing HTTPS management. Failure to organize this information prior to implementation will hinder or prevent you from deploying this functionality.

If not using verify certificates

- TMS must be running software version 11.9.1 or newer
- It is recommended that each managed system that meets the minimum software requirements for Secure-Only mode is configured in their secure-only mode prior to proceeding with TMS changes. Please see the corresponding chapter of this document for each device type
- You must have an login with site administrator rights to TMS
- You must have a Windows login with administrator rights to the Windows Server running TMS
- Be familiar with the Secure-Only commands and changes outlined in the TMS reference section of this document

If using the verify certificates mode, the following ADDITIONAL items apply

- Each managed system that meets the minimum software requirements for Secure-Only mode must have a fully qualified domain hostname (FQDN) that points to the system's IP Address. The FQDN is defined in your DNS Server.
- Each managed system that meets the minimum software requirements for Secure-Only mode must be entered in TMS by hostname (rather than IP address) and should be tracked by hostname. Use the Connection Settings page for each device to update these settings if required. TMS will enforce this tracking method on supported system types as long as a hostname is available in the Connection Settings page for that device
- Each managed system that meets the minimum software requirements for Secure-Only mode should already have its Server Certificate, Private Key, and Trusted Root Certificates installed and operational. Please see the corresponding chapter of this document for each device type
- You must have the Server Certificate and Private Key for the certificate that will be used in TMS, or it must already be installed on the server
- You must have the public certificate of the Root Certificate Authority that will be used to verify the Server Certificates used on the network or have it already installed on the server
- The Windows Server TMS is hosted on should have its time synchronized via NTP or domain membership
- The Windows Server TMS is hosted on should have its DNS name servers configured properly
- TMS must have a FQDN assigned to it, and configured as the TMS Server Fully Qualified Hostname under *Administrative Tools* > *Configuration* > *Network Settings* in TMS
- You must have the Server Certificate, and Private Key ready, or already installed on the Windows Server Hosting TMS

For information on how to generate Server Certificates or related information, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS** in this document. Once the steps outlined in the checklist are complete and the information is gathered, you can configure your systems to use HTTPS. The next sections outline the steps required to configure TMS system in Level 3 or Level 4 security.

Implementation Guide for Configuring TMS for Level 3 Security

The following provides an example of best practices for configuring a default TMS installation for Level 3 security, where the system will only support communicating with compatible external devices using HTTPS. In this Level 3 Security environment, certificates are not verified which greatly reduces the complexity and requirements of the deployment.

These steps assume you are configuring TMS with a locally installed SQL Server. Steps to increase the security level of the SQL connection if desired can be done independently of this implementation guide and will not be covered here. For more information on security of the SQL connection, please see the TMS Reference Material section of this chapter.

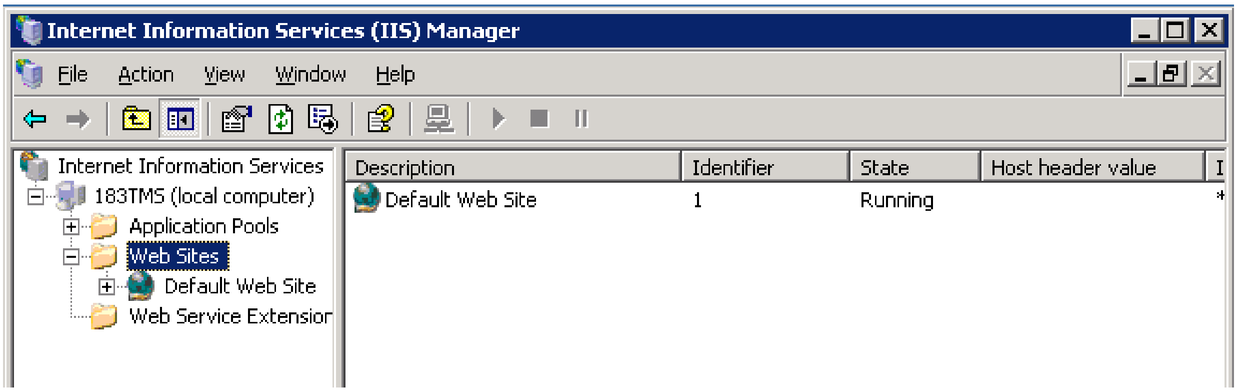
For any help creating certificates or converting file formats, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

NOTE: Secure-Only changes will apply to all systems that are **capable** of running in Secure-Only mode based on their type and software version, not just systems **configured** for Secure-Only. So when checking or configuring systems in the steps below, be sure to include all systems that meet the Secure-Only minimum software version requirement.

1. Set the TMS website to use a Server Certificate. To enable HTTPS in IIS, you must have a X.509 certificate for the server. Administrators have several alternatives for creating and installing a server certificate for Windows. If you do not have an existing certificate installed, or are unsure, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS** before proceeding for help creating and installing X.509 certificates for Windows Servers,

   - Open **Internet Information Services (IIS) Manager** from the Administrative Tools folder (*Start Menu > Administrative Tools*)

   - Expand the *Web Sites* folder under local computer


Internet Information Services (IIS) Manager
File Action View Window Help
Internet Information Services
183TMS (local computer)
Application Pools
Web Sites
Default Web Site
Web Service Extensior

Description	Identifier	State	Host header value	I
Default Web Site	1	Running		*

   - Right-Click on *Default Web Site* (or the web site TMS is installed into if another was selected during installation) and select *Properties*

   - Select the *Directory Security* tab. The *View Certificate* button should be available in the Secure Communications section. Click *View Certificate* to see the details of the installed certificate. If the button is not available, the certificate has not been selected. Click the *Server Certificate* button and in the wizard, select *Assign an existing certificate* to select a certificate installed on the server.

   - Verify you can access TMS via HTTPS by opening a web browser and entering https://<servername>/tms The page should open normally and appear the same as when using HTTP http://<servername>/tms

The following steps assume the systems to be managed are already added to TMS. This is not a requirement as systems may be added to TMS before or after Secure-Only is enabled. See the section at the end of this guide for notes on adding systems to TMS after Secure-Only has been enabled. Secure-Only systems that are already in TMS, should be configured for Secure-Only communications per the Implementation Guide of this document for that device type before proceeding for the smoothest implementation.

2. Go to the *Administrative Tools* > *Configuration* > *Network* in TMS. Set *Secure-Only mode* to be on, ensure the *Verify Certificates* checkbox is unchecked, and click the *Save* button.

3. TMS will immediately initiate an Enforce Management Setting background task where all Secure-Only compatible systems will be updated with the correct Secure Management settings. This update will start immediately after enabling Secure-Only mode, but may take some time to update all systems depending on the number of systems in TMS

   If TMS cannot connect to the system via HTTPS, the settings will not be updated until the Connection Error is resolved and the automated background task updating Enforce Management Settings has a chance has run again.

4. After giving the Enforce Management Setting task some time to complete, you should open the TMS Ticketing Service page under *Systems* > *Ticketing Service* and look for any new tickets for *No HTTPS Response* or *Certificate Validation Errors*. Review the configuration and connectivity to any system which has opened new tickets system since enabling Secure-Only mode.

TMS will now use HTTPS only to supported Secure-Only systems. HTTP is still available on the TMS Server for compatibility reasons with other device types. Users should be instructed to use HTTPS when connecting to TMS and any external software integrations can now be updated to use HTTPS when connecting to TMS.

Adding Systems to TMS after Secure-Only has been enabled

TMS will continue to operate as normal for systems that do not support Secure-Only connectivity and therefore the Add System procedures will not change for those systems. Any device that does meet the Secure-Only software version requirements should be configured per the system's chapter of this document before attempting to add it to TMS. Failure to do so will result in Connectivity Error messages and partial configuration of the system until the system is properly configured; the Connection Errors are resolved in TMS; and the Enforce Management settings have been applied to the system.

Optional Level 3 Change - Restricting TMS to Secure-Only Operation

For compatibility reasons, when Secure-Only is enabled in TMS, non-secure protocols and devices in the TMS server are not disabled. This is to ensure compatibility with devices that do not support Secure-Only and to ease discovery and deployment of systems. If an organization wishes to lock TMS down to using **only** devices that support Secure-Only mode and prevent all non-secure access to the TMS server, this can be done through a manual process.

> **WARNING**: This lock-down model should only be considered by advanced administrators who are willing to accommodate the additional limitations when TMS is locked down. This is not recommended for most TMS installations. The limitations this configuration imposes on TMS and devices are outlined later in this section.

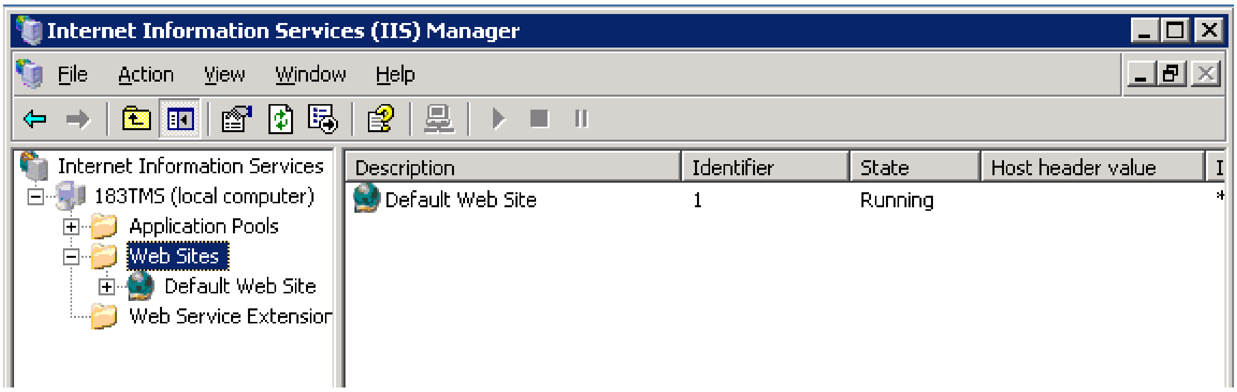
Configuration

1. Complete the steps required to enable Secure-Only mode outlined in the previous section

2. Any external integrations (Exchange Integration, IBM Lotus Notes Integration, etc) must be configured to use HTTPS in their TMS Connection Settings

3. Open **Internet Information Services (IIS) Manager** from the Administrative Tools folder (*Start Menu* > *Administrative Tools*)

4. Expand the *Web Sites* folder under local computer


Internet Information Services (IIS) Manager
File Action View Window Help
Internet Information Services
183TMS (local computer)
Application Pools
Web Sites
Default Web Site
Web Service Extension

Description	Identifier	State	Host header value	I
Default Web Site	1	Running		*

5. Right-Click on *Default Web Site* (or the web site TMS is installed into if another was selected during installation) and select *Properties*

6. Select the *Directory Security* tab. Click *Edit* under the Secure Communications Section

7. Mark the checkboxes for *Require secure channel (SSL)* and *Require 128bit Encryption* and click Ok close the window, and Ok again to close the Properties window and save the changes. If prompted that your change affects other child directories, accept these changes. Close Internet Information Services (IIS) Manager.

8. Under *Administrative Tools* > *Configuration* > *Network settings* in TMS, make the following changes

    - Set *Scan SNMP Capable Systems To Allow…* to OFF
    - Set *SNMP Broadcast/Multicast Addresses* to 127.0.0.1

9. Users must now access TMS using https://servername/tms Requests to http:// will result in a 403 error from IIS. IIS does not offer a built-in auto-redirect to https:// for web sites, but there are several options for setting this up manually. This non-TANDBERG web page discusses some possibilities. TANDBERG Support will not be able to assist with customized options to automate redirections in IIS.

Limitations of Restricting TMS to Secure-Only

- The only devices that support Secure-Only communications will be usable in TMS. See the other chapters of this document for specifics on which devices are supported

- Some device types that use non-secure protocols (such as telnet or SNMP only) will appear usable in TMS, but would not be secure and may not operate properly so they should not be used or the lock-down would be compromised.

- Users who access the old URL to TMS will only get an error page unless the administrator adds other methods to IIS to redirect them to the https URL. Users should be trained to use the new https URL

- Discovery of rogue systems functionality is hindered and should not be relied upon

- Pre-registration of codecs with factory default settings will not work. Pre-registration will work if HTTPS is enabled for externalmanager in the codec prior to activation – **xconfigure externalmanager protocol: https**

- Secure-only supported devices must have HTTPS mode set to on before TMS can add them successfully to TMS or communicate with them (HTTPS is disabled by default on systems)

- All external integrations to TMS must be updated to use HTTPS in their TMS Connection Settings

- If the Require Secure Connection properties are enabled on the specific virtual directories instead of the Parent Web Site properties, they will be reset when TMS is upgraded or uninstalled or reinstalled. This means the changes must be re-applied after each TMS upgrade.

- The SNMP Services used by TMS are left running for compatibility reasons. SNMP will not be relied upon for communicating with secure-only systems and may be blocked at the network level (Firewall) if desired

Implementation Guide for Configuring TMS for Level 4 Security

The following provides an example of best practices for configuring a default TMS installation for Level 4 security when communicating with supported systems. In this scenario TMS communications with supported systems will only use HTTPS and identities will be verified by validating X.509 certificates.

These steps assume you are configuring TMS with a locally installed SQL Server. Steps to increase the security level of the SQL connection if desired can be done independently of this implementation guide and will not be covered here. For more information on security of the SQL connection, please see the TMS Reference Material section of this chapter.

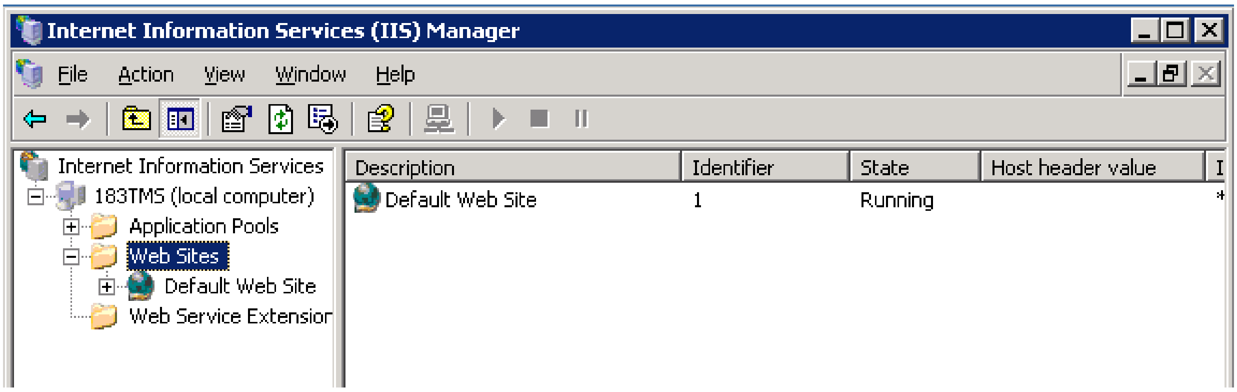
For any help creating certificates or converting file formats, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS** in this document. The filenames used in this example are only examples used for clarity. The actual filenames used may be any you chose.

NOTE: Secure-Only changes will apply to all systems that are **capable** of running in Secure-Only mode based on their type and software version, not just systems **configured** for Secure-Only. So when checking or configuring systems in the steps below, be sure to include all systems that meet the Secure-Only minimum software version requirement.

1. Set the TMS website to use a Server Certificate. To enable HTTPS in IIS, you must have a X.509 certificate for the server. Administrators have several alternatives for creating and installing a server certificate for Windows. If you do not have an existing certificate installed, or are unsure, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS** before proceeding for help creating and installing X.509 certificates for Windows Servers,

    - Open **Internet Information Services (IIS) Manager** from the Administrative Tools folder (*Start Menu > Administrative Tools*)

    - Expand the *Web Sites* folder under local computer



Description	Identifier	State	Host header value	I
Default Web Site	1	Running		*

- Right-Click on *Default Web Site* (or the web site TMS is installed into if another was selected during installation) and select *Properties*

- Select the *Directory Security* tab. The *View Certificate* button should be available in the Secure Communications section. Click *View Certificate* to see the details of the installed certificate. If the button is not available, the certificate has not been selected. Click the *Server Certificate* button and in the wizard, select *Assign an existing certificate* to select a certificate installed on the server.

- Verify you can access TMS via HTTPS by opening a web browser and entering https://<servername>/tms The page should open normally and appear the same as when using HTTP http://<servername>/tms

2. Ensure the Windows Server has the public certificate of the root CA used to trust certificates on the network installed in the Local Computer Certificate Store. Check this by opening the Certificate Manager for the Computer Account
    a. Goto *Start Menu* > *Run*.. enter **mmc.exe** and click OK
    b. From the File Menu, select *Add/Remove Snap-In*
    c. Click the *Add* Button, and from the list select *Certificates* and click Add
    d. When prompted, select *Computer Accoun*t and click *Next*. On the next screen, ensure *Local Computer* is selected and click Finish
    e. Click *Close* to close the Select Snap-in Window, and Click *Ok* to close the Add window
    f. Expand the folders to Certificates (Local Computer) > Trusted Root Certification Authorities > Certificates
    g. Browse the list to find the name of your Certificate Authority.
    h. If your CA's certificate is not present, Add it by Right Clicking the folder, and selecting *All Tasks* > *Import..* The certificate wizard will guide you through importing the certificate
    i. Browse to the certificate file, and then click *Next*
    j. When prompted to specify where to import the certificate, ensure import to *Trusted Root Certification Authorities* is specified and click *Next*. On the Summary Page click *Finish*.

    The certificate will now be available to all users of the computer

3. Verify the Windows Server has the correct date and time. The server should be setup to synchronize it's time by being a member of a Windows Domain. If not a member of a domain, use the Internet Time tab in the Date & Time Control Panel of Windows to configure an NTP source to synchronize to.

4. Verify TMS is configured with the proper Fully Qualified Domain Name that clients and devices will use to reach the server. In TMS, under *Administrative Tools* > *Configuration* > *Network Settings*, the *TMS Server Fully Qualified Hostname* settings must be configured with the FQDN of the TMS server.

The following steps assume the systems to be managed are already added to TMS. This is not a requirement as systems may be added to TMS before or after Secure-Only is enabled. See the section at the end of this guide for notes on adding systems to TMS after Secure-Only has been enabled. Secure-Only systems that are already in TMS, should be configured for Secure-Only communications per the Implementation Guide of this document for that device type before proceeding for the smoothest implementation.

5. Ensure all Secure-Only systems have their hostname (not just IP Address) configured in TMS and that TMS has no Connection Errors to the system. Each system must have its FQDN entered in the **hostname** field of the Connection Settings tab of System Navigator for that system. Administrators can quickly browse for systems that do not have their hostname defined by browsing through System Navigator. If a system has its hostname defined, the Network Address column in System Navigator will show the name entered and not an IP Address. For any Secure-Only system that does not have its hostname entered in TMS, add it on the Connection Settings tab for that system and check that for any Connection Errors reported.

6. Go to the *Administrative Tools* > *Configuration* > *Network* in TMS. Set *Secure-Only mode* to be on, ensure the *Verify Certificates* checkbox is checked, and click the *Save* button.

7. TMS will immediately initiate an Enforce Management Setting background task where all Secure-Only compatible systems will be updated with the correct Secure Management settings. This update will start immediately after enabling Secure-Only mode, but may take some time to update all systems depending on the number of systems in TMS

    If TMS cannot connect to the system via HTTPS or verify the system's certificate, the settings will not be updated until the Connection Error is resolved and the automated background task updating Enforce Management Settings has a chance to update all systems.

    The Enforce Management Settings will also change all Secure-Only systems to *Track By Hostname* in their Connection Settings tab in System Navigator. If for some reason TMS is not able to communicate with a system, this and other settings will not be changed until the connection errors are resolved.

8. After giving the Enforce Management Setting task some time to complete, you should open the TMS Ticketing Service page under *Systems > Ticketing Service* and look for any new tickets regarding *No HTTPS Response* or *Certificate Validation Errors*. Review the configuration and connectivity to any system which has opened new tickets system since enabling Secure-Only mode.

If there are any systems in TMS that are capable of Secure-Only operation, but have not been properly configured, their communication with TMS will be broken and will have a *No HTTPS Response* or *Certificate Validation Error* ticket open.

TMS will now use HTTPS only and verify the certificate when connecting to supported Secure-Only systems. HTTP is still available on the TMS Server for compatibility reasons with other device types. Users should be instructed to use HTTPS when connecting to TMS and any external software integrations can now be updated to use HTTPS when connecting to TMS. Administrators looking to further restrict the security of TMS should read the XYZ section for additional optional configuration.

Adding Systems to TMS after Secure-Only has been enabled

TMS will continue to operate as normal for systems that do not support Secure-Only connectivity and therefore the Add System procedures will not change for those systems. Any device that does meet the Secure-Only software version requirements should be configured per the system's chapter of this document before attempting to add it to TMS. Failure to do so will result in Connectivity Error messages and partial configuration of the system until the system is properly configured; the Connection Errors are resolved in TMS; and the Enforce Management settings have been applied to the system.

Optional Level 4 Change - Restricting TMS to Secure-Only Operation

For compatibility reasons, when Secure-Only is enabled in TMS, non-secure protocols and devices in the TMS server are not disabled. This is to ensure compatibility with devices that do not support Secure-Only and to ease discovery and deployment of systems. If an organization wishes to lock TMS down to using **only** devices that support Secure-Only mode and prevent all non-secure access to the TMS server, this can be done through a manual process.

> **WARNING**: This lock-down model should only be considered by advanced administrators who are willing to accommodate the additional limitations when TMS is locked down. This is not recommended for most TMS installations. The limitations this configuration imposes on TMS and devices are outlined later in this section.

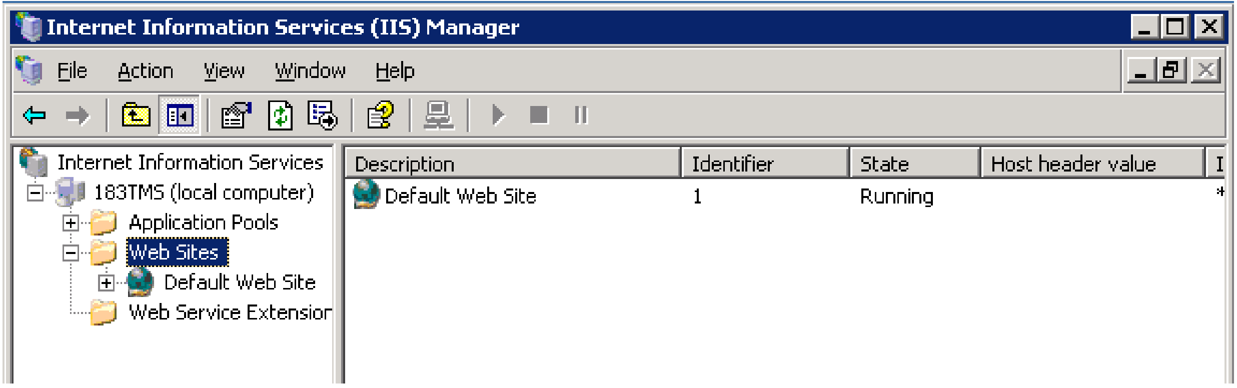
Configuration

1. Complete the steps required to enable Secure-Only mode outlined in the previous section

2. Any external integrations (Exchange Integration, IBM Lotus Notes Integration, etc) must be configured to use HTTPS in their TMS Connection Settings

3. Open **Internet Information Services (IIS) Manager** from the Administrative Tools folder (*Start Menu > Administrative Tools*)

4. Expand the *Web Sites* folder under local computer



Internet Information Services (IIS) Manager
File Action View Window Help
Internet Information Services
183TMS (local computer)
Application Pools
Web Sites
Default Web Site
Web Service Extensior

Description	Identifier	State	Host header value	I
Default Web Site	1	Running		*

5. Right-Click on *Default Web Site* (or the web site TMS is installed into if another was selected during installation) and select *Properties*

6. Select the *Directory Security* tab. Click *Edit* under the Secure Communications Section

7. Mark the checkboxes for *Require secure channel (SSL)* and *Require 128bit Encryption* and click Ok close the window, and Ok again to close the Properties window and save the changes. If prompted that your change affects other child directories, accept these changes. Close Internet Information Services (IIS) Manager.

8. Under *Administrative Tools* > *Configuration* > *Network settings* in TMS, make the following changes

   - Set *Scan SNMP Capable Systems To Allow…* to OFF
   - Set *SNMP Broadcast/Multicast Addresses* to 127.0.0.1

9. Users must now access TMS using https://servername/tms Requests to http:// will result in a 403 error from IIS. IIS does not offer a built-in auto-redirect to https:// for web sites, but there are several options for setting this up manually. This non-TANDBERG web page discusses some possibilities. TANDBERG Support will not be able to assist with customized options to automate redirections in IIS.

Limitations of Restricting TMS to Secure-Only

- The only devices that support Secure-Only communications will be usable in TMS. See the other chapters of this document for specifics on which devices are supported
- Some device types that use non-secure protocols (such as telnet or SNMP only) will appear usable in TMS, but would not be secure and may not operate properly so they should not be used or the lock-down would be compromised.
- Users who access the old URL to TMS will only get an error page unless the administrator adds other methods to IIS to redirect them to the https URL. Users should be trained to use the new https URL
- Discovery of rogue systems functionality is hindered and should not be relied upon
- Pre-registration of codecs with factory default settings will not work. Pre-registration will work if HTTPS is enabled for externalmanager in the codec prior to activation – xconfigure externalmanager protocol: https
- Secure-only supported devices must have HTTPS mode set to on before TMS can add them successfully to TMS or communicate with them (HTTPS is disabled by default on systems)
- All external integrations to TMS must be updated to use HTTPS in their TMS Connection Settings
- If the Require Secure Connection properties are enabled on the specific virtual directories instead of the Parent Web Site properties, they will be reset when TMS is upgraded or

uninstalled or reinstalled. This means the changes must be re-applied after each TMS upgrade.

- The SNMP Services used by TMS are left running for compatibility reasons. SNMP will not be relied upon for communicating with secure-only systems and may be blocked at the network level (Firewall) if desired

APPENDIX A – COMMUNICATIONS, CERTIFICATE AND KEY PRIMER

COMMUNICATIONS SECURITY PRIMER

This appendix will focus on the management of devices over an IP network and the communications between the devices and tool interacting with them, be it an administrator from their computer, or software acting as a management tool.

TANDBERG devices support most of the common communications protocols used in management, providing interfaces to interact with the device over these protocols. Examples include

- HTTP
- FTP
- SSH
- HTTPS
- SNMP
-
- Telnet
- SNMP Traps
-

TANDBERG provides this range of interfaces to provide the best flexibility to meet customer demands. However many of these protocols are not secure in a strict sense. In most networks today, information security is important to prevent interception or manipulation of information. Most organizations want to be confident that information or access that is intended to be restricted, remains that way. An important aspect of maintaining this confidence is ensuring that as information is exchanged, it is not done in a way that reduces the confidence or security of the information.

Communication Security

When managing devices remotely, the information exchanged between the tool and a device it is managing is sent over an IP network that may or may not be secured. Many protocols send their information in a ‘clear text’ format which means the data can easily be read or interpreted if the information on the IP network is intercepted. Ideally, all data would be encrypted so that even if the information is intercepted in transit, only the intended receivers would be able to interpret the information. Due to implementations, technical limitations, and complexity, full encryption is not always possible.

Examples of protocols with no built-in protection

- Telnet
- HTTP
- FTP
- SNMP v1/v2

Secure the Important Stuff

As an intermediate step, many protocols utilize methods that make it so sensitive information like passwords are not actually sent out over the network but rather use things such as Challenge/Response systems. A Challenge/Response method is used to verify knowing a shared secret between systems (such as a password) without actually sending the secret information itself over the network. This allows non-encrypted protocols such as HTTP or Telnet to secure important information like passwords without the overhead or complexity of encrypting the entire conversation. TANDBERG implements HTTP-digest authentication and Telnet Challenge (a MD5 challenge/response system) in the TANDBERG MXP and TANDBERG MPS product lines to provide a secure way of exchanging passwords over the network without encrypting the entire conversation.

Examples of protocols with improved protection using challenge/response systems

1. Telnet using Telnet Challenge
2. HTTP using HTTP-digest authentication

Encrypt the Conversation

The next level of protection is to encrypt not only sensitive information, but the entire communication so that only those who can decrypt the data can get to the enclosed information. Encryption can protect the entire contents of the conversation, but comes at a complexity in implementation and complicates use. Encryption requires the receiver to know how to decrypt the information – but not allow just anyone to decrypt the information. How decryption information is shared with the intended receiver varies between methods and is where most of the complexity arises. Encryption systems are constantly trying to balance security with how much burden the system puts on the use of the system. An example of a protocol that uses encryption to protect its communications is SSH

Ensuring Identity

The last step is to ensure identity or authentication. The security of information over the IP network is irrelevant if the person you told how to view/decrypt the information is not the intended recipient. If you are giving your bank account number to someone who claims to be from your bank, you want to ensure the person is who they claim to be before handing them your account number.

There are many levels and ways to achieve authentication of an identity. A simple example is a password – the idea is that only the intended person and the secured resource know the password. If the person supplies the right password, they are assumed to be the correct person. However, anyone who knows the password can now claim to be that person, so these systems can still be improved upon.

More advanced systems combine multiple checks to ensure identity. 'something you have and something you know' is a common phrase describing a two element system. Example: having a key (something you have) and having a PIN code (something you know) used together are more secure then either method used alone. Additionally, secure systems should have a means to ensure the information they are given, is authentic. Example: A person may present to you an ID card to prove their identity to you, but you need to be able to validate that the ID card itself is real, not a fake.

TLS

In IP communications, the most common way to address these concerns today is by using encryption and authentication using TLS (Transport Layer Security). TLS is the successor to SSL (Secure Sockets Layer) which most people are familiar with from using their Internet Browser. While the two are defined separate, modern TLS is often commonly referred to as SSL.

TLS is a standardized method that includes privacy through full encryption of a conversation and identity verification through authentication. Because TLS addresses the points of both privacy and authentication in sufficiently safe methods, it is commonly accepted as a true 'secure' way of exchanging information between parties. TLS operates as connection method, rather then as an application itself. It creates a secure connection that other applications will use – such as a web browser session, a mail transfer session, etc. It is used as the connection *pipe* built between two systems that other applications will flow through.

TLS implements authentication through digitally signed Certificates and privacy through symmetric encryption where the encryption keys have been exchanged in real-time using trusted key pairs that have been shared securely using an existing public/private key pair. The next sections will elaborate on these concepts.

Examples of protocols that implement TLS

- HTTPS
- SMTP over TLS

Certificate Primer

Certificates are electronic credentials used to certify the identity of something or someone. Certificates as certified by a *Certificate Authority* (CA). A simple example of certificates and certificate authorities is a government ID card or driver's license.

In daily life, you often need to prove to others you are who you claim to be. To do this, you need something to show to others they sufficiently trust to prove your identity. Governments usually fill this need. You provide information to the government to validate who you are. If your information is accepted, the government certifies your identity by issuing an ID card (a certificate) so that you can prove your identity to others. Here, the government is acting as a *Certificate Authority*, issuing *certificates* (IDs) to those it will vouch for their identity. For someone to accept your driver's license or ID as a valid way of proving your identity, they must trust the certificate itself is valid, is not altered, and trust the issuing authority (the CA) itself. The trust of the issuing authority is a conscious choice, but verifying the credential is valid and not altered can be more complex. For secure communications on the internet, this is achieved by the creation of X.509 certificates and digitally signing of them with *key pairs*.

Key Pairs

Key pairs are a form of encryption commonly known as *public key cryptology* where a matching set of keys are used – a public and private key. The keys are unique in that what is encoded by one key, can only be decoded by the other key of the pair, and vise versa. One key is held private (the private key), while the other key is distributed to others publicly (the public key). When something can be decoded by the public key, one is ensured the data was encrypted by the holder of the private key – ensuring the originator of the information and that it has not been modified in transit. The inverse works as well, where data encrypted with a public key can only be decoded by the holder of the private key. This relationship allows secure exchange of information and verifying the originator of information presented as long as the private key remains secret and secured.

Validating Certificates through Signing

Key pairs are used with X.509 certificates in both the granting and validating of certificates. When someone wants a certificate, they generate a *certificate signing request* or often just called a *certificate request*. The certificate requester generates a key pair and submits their public key as part of the request to a Certificate Authority (CA). The CA has their own private and public key pair. The CA uses their private key to sign the certificate request and return the request to the requester as a *signed certificate.*

When the certificate holder presents their signed certificate (which includes their public key) to validate their identity to a 3rd party, anyone with the public key of the CA that signed the certificate can validate the certificate as being valid and unmodified since the CA signed it. From this point, the certificate holder is trusted by the 3rd party to be whom they claim to be and the public key sent by the certificate holder is trusted. With the public key of the certificate holder, information can now be sent securely to the certificate holder by encrypting it with their public key. This process can be repeated in the other direction as well, so the other party's identity can be verified as well.

Who are Certificate Authorities?

For digital communications and X.509 certificates, there are many Certificate Authorities in the world – you can even become your own. The issue is that the Certificate Authority itself must be trusted, as by trusting it, you trust the identity of anyone they certify. This means Certificate Authorities must be very secure, reliable organizations. To trust a Certificate Authority, you import their certificate, which includes their public key, into your device or computer. Most major operating systems come pre-installed with a list of CAs the vendor believes to be trustworthy. Users or administrators may also add/remove to this list. Due to the cost of obtaining certificates for every device from public CAs, certificates are often reserved for critical systems where security is important. Alternatively, many organizations setup their own Certificate Authority, which is setup to be automatically or manually trusted by the users of their organization.

Which CA to use is a preference by an organization and is not something TANDBERG can advise on which to use.

Important Aspects of Certificates and Keys

When validating certificates, devices not only check who signed the certificate, but that the information in it is still validate and matches the person/device using it. A certificate includes descriptive information about the certificate holder but also includes important information such as validate date ranges and the name of the device the certificate was granted to. Most software will present an error or warning if this information is not valid. The commonly checked properties are

- date ranges – a certificate has both a start and end date on its validity
- common name – the name the certificate is assigned to. For computers, this usually represents the hostname or fully qualified hostname of the server – example www.tandberg.com
- purpose – a certificate may be assigned a specific use, such as for a mail server, or web server

Common mistakes include changing the names of servers without getting a new certificate so that the name check will fail, letting certificates expire, or not accessing the service using the name registered on the certificate (using an IP address rather then a hostname). Using invalid certificates will lead to services failing to connect or users seeing warning messages.

Passphrases

Because the private key in a key pair is so valuable in the security of the system, best practice dictates that the key itself is also encrypted. This encryption uses a strong passphrase to ensure only trusted entities can read the key. However, the key must be read by services/software using the key to decode things so the passphrase must entered each time the software starts (when it reads the private key). Because of this, encryption is often not used on private keys to make it easier to use with server software.

Self-signed Certificates

Instead of being signed by a Certificate Authority, it is possible to sign a certificate using its own key. These certificates are not secure because their identity can not be validated against a CA. These certificates do not provide the security of a signed certificate and should not be used in production-ready networks concerned about security. Self-signed certificates are often used for testing purposes or by organizations who do not demand the full security that signed certificates offer.

Revoking Certificates

If a certificate becomes compromised, a CA may revoke the previously granted certificate – marking it as no longer valid. This is done by the CA by publishing a certificate revocation list (CRL). The CRL is distributed to clients who check it when checking if a certificate is valid.

Secure Communications using TLS/SSL

When communicating over the internet securely, the challenge is to ensure

- Who you are talking to is who they claim to be
- That no one can intercept your information and interpret it

Transport Layer Security (TLS) addresses both these requirements and is the most common form of encryption used between clients and servers on the internet. TLS addresses who you are talking to by means of X.509 certificate exchanges and verifying the certificates.

When a client connects to a server, the server provides its certificate to prove who it is, and provide its public key. The certificate has been signed using the private key of a Certificate Authority. The client looks at the certificate to see who signed it and then uses the public key provided by the signing Certificate Authority to validate the certificate and its contents. If this check passes, the client trusts the public key provided by the server, and provides its own public key back to the server. Optionally the server may require a certificate from the client to validate the client's identity and public key (this is known as Mutual Authentication). Using these public keys and private keys, the client and server can encrypt and exchange a new random shared secret which they use to encrypt the rest of their conversation. The new shared secret is used to encrypt to the rest of the communication rather then the public and private keys because it is more efficient and less intensive then using the key pairs to encrypt the rest of the conversation. Now, the client and server can have a secure conversation free of worry of eavesdroppers and imposters.

Glossary of Terms

CA – A Certificate Authority. A organization that validates and signs certificate requests

Certificate Response – another term often used to label the signed certificate a CA returns in response to a certificate request

CRL – Certificate Revocation List. A list from a CA of previously signed certificates that it marks as no longer valid

csr – filename extension normally associated with certificate signing request. Request sent to a Certificate Authority to obtain a certificate

Passphrase – a secret phrase of text used to secure or encrypt something. Similar to a password, but typically more secure through greater length or complexity

PEM – a text file format and filename extension used when storing certificates and keys. There are several formats for storing keys and certificate. A PEM file includes descriptive headers and a text representation of keys and certificates

PKCS#12 – another file format used to save or distribute keys and certificates. Commonly used by Microsoft Internet Explorer for easy import of certificates and by Microsoft Windows for importing/exporting keys and certificates. Can be converted to PEM format using openssl tools

Private key – the portion of a matching key pair that is held secret. Used to validate and decode any information encrypted with the matching public key. Loss or compromise of a private key will invalidate the security of any information encrypted with that key.

Public key – the portion of a matching key pair that is publicly distributed and shared. Used to validate and decode information encoded with the matching private key.

Self-Signed Certificate – a certificate digitally signed using its own key rather then by a CA. Does not provide the same level of security as a CA signed certificate

SSL – Secure Sockets Layer – a protocol used to ensure secure communications using identities and digital signatures

X.509 – An international standard from the ITU-T that defines aspects of public key infrastructure, including digital certificates and validation methods. X.509 certificates are used with SSL communications

Additional References

A description of **SSL**, how it works, and what security it provides. Written for the layman. - http://luxsci.com/info/about_ssl.html

SSL FAQ from Verisign - http://www.verisign.com/ssl/ssl-information-center/faq/index.html

SSL Introduction from Apache - http://httpd.apache.org/docs/2.0/ssl/ssl_intro.html

APPENDIX B – CREATING AND WORKING WITH CERTIFICATES FOR WINDOWS SERVER AND TMS

This appendix will provide external references and assistance with generating and requesting X.509 certificates to enable HTTPS on Windows Servers, and TMS. For help generating and working with certificates for other devices such as endpoints, MCUs, etc, please see **Appendix C – Creating and Working with Certificates for TANDBERG Systems**

The two most common methods to generating certificate requests are using Microsoft supplied Certificate wizards or using utilities provided by the OpenSSL open source project. Either can be used to prepare certificates for use with Windows. Please note these guides do not represent the only way to provision Windows Servers, but are presented only for assistance. This document will provide examples using the Microsoft tools, and then the same tasks using the OpenSSL tools. Administrators may chose whichever toolset they prefer to use. The OpenSSL tool kit has an advantage that it can work with multiple formats and convert between them easier then the Microsoft Tools.

The format and steps required to create requests may vary between Certificate Authorities. Most Certificate Authorities have detailed FAQs on how to generate certificate requests for their services. These pages offer detailed step-by-step guides. Some Examples are below

Thawte – By Product Listing

Verisign – By Product Listing

Microsoft Certificate Tools

Windows Server provides a Certificate Wizard which can be used to generate certificate requests, import existing certificates, or install a certificate response. The wizard is automatically loaded when starting related tasks. Additionally, Windows provides a Certificate Services functionality that operates as a Certificate Authority that can be integrated with Active Directory. The Enterprise CA will not be discussed here, but further information about requesting certificate using an integrated Enterprise CA can be found at Microsoft's website

Managing Microsoft Certificate Services and SSL for Windows 2000

Public Key Infrastructure for Windows Server 2003

The most common Tool you must interface with on Windows is the Certificates Snap-In Tool for Microsoft Management Console. To access certificates that have been loaded into a Windows Server, you must use the Certificates Snap-In for Microsoft Management Console (MMC). Certificates may be stored in a user's profile, service account's profile, or the local computer account's profile. If a certificate is to be available to all users, the certificates must be loaded into the Local Computer Account.

To open the Certificate Snap-In to the Local Computer Account

1. Goto *Start Menu* > *Run*.. enter **mmc.exe** and click OK
2. From the File Menu, select *Add/Remove Snap-In*
3. Click the *Add* Button, and from the list select *Certificates* and click Add
4. When prompted, select *Computer Accoun*t and click *Next*. On the next screen, ensure *Local Computer* is selected and click Finish
5. Click *Close* to close the Select Snap-in Window, and Click *Ok* to close the Add window
6. The Certificates Item can be expanded to view the various containers. Containers of interest include:

Personal > Certificates – Default location for user loaded Server Certificates

Trusted Root Certification Authorities > Certificates – Default location for certificates of Root CAs to trust

Certificate Enrollment Requests > Certificates – Default location for pending certificates and keys generated through the Certificate Wizard

Common TMS Certificate Tasks using Microsoft Tools

Generate a Server Certificate for the TMS Server using the Microsoft Certificate Wizard

1. Under *Administrative Tools*, open *Internet Information Services (IIS) Manager*.
2. Expand the *Web Sites* Folder and right click on the *Default Web Site* item and select *Properties*. If TMS was installed to another web site during installation, please select that web site.
3. Click the *Directory Security* tab.
4. Click *Server Certificate* in the Secure communications section
5. Select *Create a new certificate*
6. Select *Prepare the request now, but send it later*.
7. The Wizard will prompt for a Name for the certificate. This is just a friendly name that will be used when browsing through Certificate listings and can be anything. You should use a name that makes it easy to recognize the certificate as for this server. Select the key length (1024 is appropriate) and do not mark the CSP checkbox
8. The Wizard will prompt for the X.509 attributes of the certificate.
    a. The Organization Name and Unit are labels that will be stored in the certificate and should help identify the valid use of the certificate.
    b. The Common Name is critical and must match the fully qualified host name that users use to access the web server – Example: tms.na.corporation.com
    c. The Geographical Information are labels that will appear in the certificate and should help identify the valid use of the certificate.
    d. Specify the filename to use for the certificate request and click *Next* and Finish the Certificate Wizard. The resultant .csr file is the certificate request including your public key that you will submit to a Certificate Authority to sign.

Your resulting Certificate Request file must now be sent to the Certificate Authority you wish to sign your Certificate (Verisign, Thawte, your internal IT Group, etc). Instructions on how to submit the certificate request will be provided by the Certificate Authority.

Once your Certificate Authority has signed your Certificate Request, they will return you a signed certificate, usually in PEM format. If the certificate is on an email or webpage and not in its own file, copy the contents starting with the -----BEGIN CERTIFICATE----- line and through the -----END CERTIFICATE----- line. Save the contents to a text file and name the file **cert.pem**. This is your signed certificate file.

9. Under Administrative Tools, open *Internet Information Services (IIS) Manager*.
10. Expand the Web Sites Folder and right click on the Default Web Site item and select Properties. If TMS was installed to another web site during installation, please select that web site.
11. Click the *Directory Security* tab.
12. Click *Server Certificate* in the Secure communications section
13. Choose *Process the Pending Request and Install the Certificate*, then click *Next*.

14. Select the location of the certificate response file, and then click *Next*.

15. The Wizard will confirm the information contained in the certificate, acknowledge the information and continue clicking Next to finish the installation.

You can review installed certificates and keys on a server using the Certificates Snap-in in Microsoft Management Console (MMC). Certificates will be stored in the Computer Account, under Personal>Certificates.

Renewing a Server Certificate for the TMS Server using the Microsoft Certificate Wizard

Renewing a server certificate is similar to requesting a new certificate. The IIS Certificate Wizard is used to generate a new certificate signing request and install the resulting certificate.

1. Under *Administrative Tools*, open *Internet Information Services (IIS) Manager*.
2. Expand the *Web Sites* Folder and right click on the *Default Web Site* item and select *Properties*. If TMS was installed to another web site during installation, please select that web site.
3. Click the *Directory Security* tab.
4. Click *Server Certificate* in the Secure communications section
5. Select *Renew the current certificate*
6. Select *Prepare the request now, but send it later*.
7. You will be prompted for the filename to save the request in. Click *Next*, verify the information, and click Finish to complete the wizard.

Your resulting Certificate Request file must now be sent to the Certificate Authority you wish to sign your Certificate (Verisign, Thawte, your internal IT Group, etc). Instructions on how to submit the certificate request will be provided by the Certificate Authority.

Once your Certificate Authority has signed your Certificate Request, they will return you a signed certificate, usually in PEM format. Save the resulting certificate in a separate text file and copy it to your server. This is your certificate request file.

1. Under Administrative Tools, open *Internet Information Services (IIS) Manager*.
2. Expand the Web Sites Folder and right click on the Default Web Site item and select Properties. If TMS was installed to another web site during installation, please select that web site.
3. Click the *Directory Security* tab.
4. Click *Server Certificate* in the Secure communications section
5. Choose *Process the Pending Request and Install the Certificate*, then click *Next*.
6. Select the location of the certificate response file, and then click *Next*.
7. The Wizard will confirm the information contained in the certificate, acknowledge the information and continue clicking *Next* to finish the installation.

Generating a Self-Signed Certificates using the Microsoft Self-SSL tool

Microsoft provides a utility in their IIS 6.0 Resource Kit to allow easy creation of self-signed certificates for IIS named SelfSSL. The use of self-signed certificates is not recommended for production use, but may be used if an organization accepts the security limitations. This tool can be used for IIS6 or II5. These instructions refer to installing a self-signed certificate on the Default Web Site of the IIS Server

To install a self-signed certificate using SelfSSL, perform the following

1. Download the IIS 6.0 Resource Kit from Microsoft's Support Site
2. Install the IIS 6.0 Resource Kit on your TMS Server. You may use the default options, or choose custom installation and only select SelfSSL to install. Once the installation completes, there will be a new Folder added to your Start Menu under *All Programs* > *IIS Resources* > *SelfSSL*
3. Open SelfSSL from the Start Menu *All Programs* > *IIS Resources* > *SelfSSL* > *SelfSSL* . A command prompt will open showing the help for the command
4. Creating and installing a certificate is done by simply entering the selfssl command with the appropriate options. The command line options you should specify are

   /T – this installs the newly created certificate on the local server as trusted so you will not get any errors when viewing the website locally on the computer

   /N:commonname – commonname should be the hostname users use to connect to your server. Example: tms.na.company.com

   /V:days – the length of time in days you wish the certificate to be valid

   Example: Currently users reach your TMS server at http://tms.na.company.com/tms and you want to create and install a self-signed certificate for this server that will be good for 2 years, you would enter the command as

   ```
   selfssl.exe /T /N:tms.na.company.com /V:730
   ```

5. After entering the command, the program will prompt you

   Do you want to replace the SSL settings for site 1 (Y/N)?

   Site 1 is the Default Web Site – Enter Y and press enter. The certificate will now be installed on the server.

Self-signed certificates will still generate certificate warnings when clients connect to the server because the signing authority is not trusted. Users can avoid future warnings by adding the certificate to their local computer's trusted list.

> Firefox users are prompted what to do with the certificate when they connect to the HTTPS site – Either accept the certificate permanently, temporarily, or not at all.
>
> Internet Explorer users can install the certificate by clicking View Certificate in the Warning dialog when connect to the HTTPS site and installing the certificate following the options provided.

Find if your certificate authority is trusted by the server

1. Open the Certificates Snap-In to the Local Computer Account
2. Expand the folders for Trusted Root Certification Authorities > Certificates
3. Browse the list for the name of your Certificate Authority. Double-click a certificate to see additional details.

   Note: A certificate may be loaded in a user's profile, but would only be available to that user. The certificate should be loaded in the Local Computer account if it is to be available to all users, including the service accounts of the server

Adding a Certificate to the Trusted Root CA List for all users of the Server

1. Open the Certificates Snap-In to the Local Computer Account
2. Expand the folders for Trusted Root Certification Authorities > Certificates

3. If your CA's certificate is not present, Add it by Right Clicking the folder, and selecting *All Tasks > Import..* The certificate wizard will guide you through importing the certificate

4. Browse to the certificate file, and then click *Next*

5. When prompted to specify where to import the certificate, ensure import to *Trusted Root Certification Authorities* is specified and click *Next*. On the Summary Page click *Finish*.

Viewing the contents of a Certificate File

Windows has built in viewing of both .PEM encoded certificates, and PKCS #12 encoded certificates. Viewing the content of a certificate is useful to ensure a certificate is the particular certificate you are looking for, and for viewing the details of the certificate itself.

Viewing PEM encoded files

1. To view a text encoded PEM certificate, rename the file extension to be .cer in Windows. If the certificate is on an email or webpage, copy the contents starting with the -----BEGIN CERTIFICATE----- line and through the -----END CERTIFICATE----- line. Save the contents to a text file and name the file with the extension .cer

2. Browse to the .cer file on your Desktop in Windows. Double-click the .cer file and a details Window will open showing you the details of the certificate, including if it is trusted by the currently installed Certificate Authorities.

NOTE: Installing Certificates by clicking the Install Certificate button in the details Window will only install the Certificate in the current user's profile – not the Local Computer Account which is needed for TMS installs.

Viewing PKCS#12 encoded files

1. PKCS#12 encoded files are binary files, so unlike PEM files they cannot be copied and pasted between messages or web pages. A PKCS#12 file should have a file extension of .pfx on Windows. If not, rename the rename the file's extension to be .pfx in Windows.

2. Browse to the .pfx file on your Desktop in Windows. Double-click the .cer file and a details Window will open showing you the details of the certificate, including if it is trusted by the currently installed Certificate Authorities. If the file included the private key for the certificate, it will be noted at the bottom of the Details Window.

NOTE: Installing Certificates by clicking the Install Certificate button in the details Window will only install the Certificate in the current user's profile – not the Local Computer Account which is needed for TMS installs.

Additional Tips for the Microsoft Certificates MMC Snap-In

Adding an Existing Server Certificate to the Local Computer Account of the Server

1. Open the Certificates Snap-In to the Local Computer Account

2. Expand the folders for Personal > Certificates

3. Right Click the Certificates folder, and select *All Tasks > Import..* The certificate wizard will guide you through importing the certificate

4. Browse to the certificate file, and then click *Next*

5. When prompted to specify where to import the certificate, ensure import to *Personal* is specified and click *Next*. On the Summary Page click *Finish*.

To find the private key of a Certificate Generated by the Certificate Wizard

1. Open the Certificates Snap-In to the Local Computer Account
2. Expand the folders and find the certificate in question. User installed certificates generally are in the Personal > Certificates folder, and pending Certificate Requests will be in the Certificate Enrollment Requests > Certificates folder.
3. Right-Click on the Certificate and select *All Tasks > Export…* . The Certificate Export Wizard will be opened
4. Click Next to start the Wizard. Select toYes, Export the Private Key and click Next
5. Personal Information Exchange will be selected, along with Enable Strong Encryption by default. Click Next.
6. You can choose a password to protect the private key. This is optional and can be skipped if the resulting file will be handled in a secure manner. Click Next
7. Name the file to export the certificate to. Click Next and Finish to complete the export. The resulting file will be a PKCS #12 formatted certificate with the private key included. To extract the private key, the resulting pfx file can be converted using OpenSSL tools.

OpenSSL Certificate Tools

OpenSSL is an open source project designed to run on unix/linux systems and is commonly installed in most distributions. A 32bit Windows port of OpenSSL is also available from Shining Light Productions and will behave the same for the tasks described here. . In Windows you will perform these steps from the command prompt. To use the tools, open a command prompt and Navigate to the openssl\bin installation directory and enter the commands as shown.

There are many resources available on the web that describe the creation and signing of certificates. This document will only show the basics needed for TMS administrators. Please recognize there are also many ways to achieve the same results using different commands or sequences. This document will act only as a basic guide. Some additional resources can be found at

OpenSSL's FAQ Page - http://www.openssl.org/support/faq.html

Linux SSL HOWTO – http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html

This document will not go into setting up a CA or signing certificates as a CA

Common TMS Certificate Tasks using OpenSSL

Generating a Server Certificate for the TMS Server using OpenSSL

To generate a Server Certificate for Windows using OpenSSL, the following steps must be completed

1. Generate a Private Key
2. Generate a Certificate Signing Request
3. Have the Certificate Authority Sign the Certificate Signing Request
4. Convert the certificate and private key into a PKCS#12 formatted file
5. Import the resulting signed certificate and private key to the server

These steps differ from using the Microsoft Certificate Wizard as the Microsoft tool automates steps 1 and 4. Using the OpenSSL tools is sometimes desirable because of the greater control over options possible. The majority of these steps can also be performed on any machine, not just the TMS server.

The full steps are outlined below:

1. Generate a Private Key. Assuming you do not have a key to start, we will generate a new 1024bit RSA key without a passphrase

   ```
   openssl genrsa -out privatekey.pem 1024
   ```

   privatekey.pem is your private key in PEM format. It must be protected and remain hidden from other people!

2. Start the process of generating a new certificate signing request

   ```
   openssl req -new -key privatekey.pem -out certcsr.pem
   ```

   The program will now prompt you for the values to go into the certificate. Details on the values are below. The critical value is the commonname which must be the full hostname that users will use to reach your site. Example, if users reach your site as http://tms.na.company.com/tms your commonname should be **tms.na.company.com**

   a. The Organization Name and Unit are labels that will be stored in the certificate and should help identify the valid use of the certificate.
   b. The Common Name is critical and must match the fully qualified host name that users use to access the web server – Example: tms.na.corporation.com
   c. The Geographical Information are labels that will appear in the certificate and should help identify the valid use of the certificate.
   d. Leave the challenge password empty

   The result is certcsr.pem which is a certificate signing request in PEM format. The resulting Certificate Signing Request file must now be sent to the Certificate Authority you wish to sign your Certificate (Verisign, Thawte, your internal IT Group, etc). Instructions on how to submit the certificate request will be provided by the Certificate Authority.

3. Once your Certificate Authority has signed your Certificate Request, they will return you a signed certificate, usually in PEM format via email or a webpage. If the certificate is on an email or webpage and not in its own file, copy the contents starting with the -----BEGIN CERTIFICATE----- line and through the -----END CERTIFICATE----- line. Save the contents to a text file and name the file **cert.pem**. You can quickly verify the contents of your cert.pem file with the following command

   ```
   openssl x509 –in cert.pem –noout –text
   ```

4. To import your certificate and private key to Windows, they must be packaged together into a PKCS#12 formatted file. OpenSSL provides a method to do this for you. The example is your private key is in **privatekey.pem** and your signed certificate is in **cert.pem**. Enter the following command on a single line. 'My Server Cert' is a friendly name for the certificate; use a name that will identify the certificate. When prompted for an export password, enter a password to protect the certificate. You will be prompted for this password when importing to the certificate into Windows.

   ```
   openssl pkcs12 -export -in cert.pem –inkey privatekey.pem -out certificate.pfx -
   name "My Server Cert"
   ```

   The resulting certificate.pfx file is suitable for importing to the Server using the Certificate MMC Snap-In. Do **not** distribute this file to anyone! It contains your private key.

5. Import the certificate.pfx file to the Certificate Store of the Local Computer Account. Copy the certificate.pfx file to the Windows server in a secure fashion (such as using a CD-ROM or USB Key). Import the Certificate to be used by TMS by performing the following:

   a. Under Administrative Tools, open *Internet Information Services (IIS) Manager*.
   b. Expand the Web Sites Folder and right click on the Default Web Site item and select Properties. If TMS was installed to another web site during installation, please select that web site.
   c. Click the *Directory Security* tab.

d. Click *Server Certificate* in the Secure communications section
e. Choose *Import a Certificate from a pfx file and* click *Next*.
f. Select the location of the certificate.pfx file, and then click *Next*. It is optional to allow the certificate to be exportable, but you **must** keep the privatekey.pem file you chose not to let the certificate be exportable to use for future renewals
g. When prompted, enter the export password created when creating the certificate.pfx file and click Next
h. Leave the SSL port to be 443, and Click Next.
i. The Wizard will confirm the information contained in the certificate, acknowledge the information and continue clicking Next and then Finish.

The certificate is now installed in IIS for use, and stored in the Certificate store of the computer. You can review installed certificates and keys on a server using the Certificates Snap-in in Microsoft Management Console (MMC). Certificates will be stored in the Computer Account, under Personal. To protect your certificate, you should retain the privatekey.pem file in a safe, secure location and delete all additional copies of it, and the certificate.pfx file.

Renewing a Server Certificate for the TMS Server using OpenSSL

Since the Windows Server has the private key, the easiest way to renew a TMS certificate is to use the Windows Certificate Wizard. Please see Renewing a Server Certificate for the TMS Server using the Microsoft Certificate Wizard

Generating a Self-Signed Certificates using OpenSSL

To generate a self-signed certificate for TMS rather than having a true Certificate Authority sign the certificate signing request, you follow all the same steps used to generate a key, signing request, and install the certificate outlined in Generating a Server Certificate for the TMS Server using OpenSSL with one exception. Instead of sending the certcsr.pem file to a Certificate Authority, you process the file yourself using OpenSSL. To self-sign a Certificate Signing Request, use the following command. **certcsr.pem** is your certificate signing request in PEM format. **privatekey.pem** is your private key in PEM format. *Days* is the number of days you'd like the certificate to be valid

```
openssl x509 -req -days 360 -in certcsr.pem -signkey privatekey.pem -out cert.pem
```

*The resulting **cert.pem** is your self-signed certificate*

Using the cert.pem file, you can complete the remainder of the installation steps as if the certificate were returned from a regular Certificate Authority.

Find if your certificate authority is trusted by the server

This task must be completed using the Windows Certificate Tools. Please see Find if your certificate authority is trusted by the server

Adding a Certificate to the Trusted Root CA List for all users of the Server

This task must be completed using the Windows Certificate Tools. Please see Adding a Certificate to the Trusted Root CA List for all users of the Server

Viewing the contents of a Certificate File

If you need to verify the contents of a certificate, this can be done with the x509 commands of openssl. To display the contents of a certificate, enter the command below, substituting **cert.pem** with your specific filename.

```
openssl x509 –in cert.pem –noout –text
```

Openssl will display the contents of the certificate.

Exporting a PEM Certificate and Private Key to PKCS#12 for Windows

If you have a signed certificate in PEM format and private key you wish to import into Windows for use as a server certificate, use the following command to convert the file to a PKCS#12 file Windows will understand. In the example is your private key is in **privatekey.pem** and your signed certificate is in **cert.pem**. Enter the following command on a single line. 'My Server Cert' is a friendly name for the certificate; use a name that will identify the certificate. When prompted for an export password, enter a password to protect the certificate. You will be prompted for this password when importing to the certificate into Windows.

```
openssl pkcs12 -export -in cert.pem –inkey privatekey.pem -out certificate.pfx -
name “My Server Cert”
```

The resulting certificate.pfx file is suitable for importing to the Server using the Certificate MMC Snap-In. Do **not** distribute this file to anyone! It contains your private key.

To import the certificate and private key into Windows perform the following

1. Copy the certificate.pfx file to the Windows server in a secure fashion (such as using a CD-ROM or USB Key)

If the Certificate is to be used by any service on the Server, it should be installed into the Local Computer Account using the Certificates Snap-In.

2. Goto *Start Menu* > *Run*.. enter **mmc.exe** and click OK
3. From the File Menu, select *Add/Remove Snap-In*
4. Click the *Add* Button, and from the list select *Certificates* and click Add
5. When prompted, select *Computer Accoun*t and click *Next*. On the next screen, ensure *Local Computer* is selected and click Finish
6. Click *Close* to close the Select Snap-in Window, and Click *Ok* to close the Add window
7. Expand the Personal > Certificates folder. Right Click the Certificates folder, and select *All Tasks* > *Import*..
8. Browse to the location of the certificates.pfx file and select Next
9. When prompted, enter the password created when creating the certificate.pfx file. It is optional to allow the certificate to be exportable, but you **must** keep the privatekey.pem file you chose not to let the certificate be exportable to use for future renewals. Click Next
10. *Place all Certificates…* will be selected and the location will be Personal. Click Next
11. Click Finish to complete the import
12. If the certificate was imported successfully, go back and delete the files copied to the server and the certificate.pfx file for security purposes.

The certificate may now be selected for use by applications such as IIS through their management interfaces.

APPENDIX C – CREATING AND WORKING WITH CERTIFICATES FOR TANDBERG SYSTEMS

This appendix will provide external references and assistance with generating and requesting X.509 certificates to enable HTTPS on TANDBERG endpoints, MCUs, and gatekeepers. For help generating and working with certificates for Windows Servers and TMS, please see **Appendix B – Creating and Working with Certificates for Windows Server and TMS**

The TANDBERG systems do not have internal tools for generating or manipulating certificates, so you must use external tools such as the utilities provided by the OpenSSL open source project. Please note these guides do not represent the only way to work with certificates, but are presented only for assistance. This document will provide examples using the OpenSSL tools.

The format and steps required to create requests may vary between Certificate Authorities. Most Certificate Authorities have detailed FAQs on how to generate certificate requests for their services. These pages offer detailed step-by-step guides. Some Examples are below

Thawte – By Product Listing

Verisign – By Product Listing

OpenSSL Certificate Tools

OpenSSL is an open source project designed to run on unix/linux systems and is commonly installed in most distributions. A 32bit Windows port of OpenSSL is also available from Shining Light Productions and will behave the same for the tasks described here. . If using Windows, you will perform these steps from the command prompt. To use the tools, open a command prompt and Navigate to the openssl\bin installation directory and enter the commands as shown.

There are many resources available on the web that describes the creation and signing of certificates. This document will only show the basics needed for TANDBERG system administrators. Please recognize there are also many ways to achieve the same results using different commands or sequences. This document will act only as a basic guide. Some additional resources can be found at

OpenSSL's FAQ Page - http://www.openssl.org/support/faq.html

Linux SSL HOWTO – http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html

This document will not go into setting up a CA or signing certificates as a CA

Common TANDBERG System Certificate Tasks using OpenSSL

Generating a Server Certificate for a TANDBERG System using OpenSSL

To generate a Server Certificate for Windows using OpenSSL, the following steps must be completed

1. Generate a Private Key
2. Generate a Certificate Signing Request
3. Have the Certificate Authority Sign the Certificate Signing Request
4. Import the resulting signed certificate and private key to the system

These steps are performed on a computer separate from the TANDBERG system.

The full steps are outlined below:

1. Assuming you do not have a key to start, we will generate a new 1024bit RSA key

   ```
   openssl genrsa -out privatekey.pem 1024
   ```

   Optional: If you want the key to be encrypted with a passphrase, add **–des3** to the command and you will be prompted for a passphrase when the command executes

   ```
   openssl genrsa –out privatekey.pem –des3 1024
   ```

   privatekey.pem is your private key in PEM format. It must be protected and remain hidden from other people!

2. Start the process of generating a new certificate signing request

   ```
   openssl req -new -key privatekey.pem -out certcsr.pem
   ```

   The program will now prompt you for the values to go into the certificate. Details on the values are below. The critical value is the commonname which must be the full hostname that users will use to reach your system. Example, if users reach your site as http://codec43.na.company.com your commonname should be **codec43.na.company.com**

   a. The Organization Name and Unit are labels that will be stored in the certificate and should help identify the valid use of the certificate.
   b. The Common Name is critical and must match the fully qualified host name that users use to access the web server – Example: codec43.na.corporation.com
   c. The Geographical Information are labels that will appear in the certificate and should help identify the valid use of the certificate.
   d. Leave the challenge password empty

   The result is certcsr.pem which is a certificate signing request in PEM format. The resulting Certificate Signing Request file must now be sent to the Certificate Authority you wish to sign your Certificate (Verisign, Thawte, your internal IT Group, etc). Instructions on how to submit the certificate request will be provided by the Certificate Authority.

3. Once your Certificate Authority has signed your Certificate Request, they will return you a signed certificate, usually in PEM format via email or a webpage. If the certificate is on an email or webpage and not in its own file, copy the contents starting with the -----BEGIN CERTIFICATE----- line and through the -----END CERTIFICATE----- line. Save the contents to a text file and name the file **cert.pem**. You can quickly verify the contents of your cert.pem file with the following command

   ```
   openssl x509 –in cert.pem –noout –text
   ```

4. The signed certificate cert.pem, privatekey.pem, and your passphrase are the components needed to configure your TANDBERG system. Please find the chapter for your type of device in the main portion of this document, and follow the implementation guides for that device for specific instructions on deploying the certificate to the system.

Renewing a Server Certificate for a TANDBERG System using OpenSSL

Renewing an existing certificate is similar to creating your original certificate request. You can reuse the certificate signing request you generated when creating the original certificate, or generate a new one using your existing private key. To re-use the original certificate signing request, simply re-submit the request to your Certificate Authority. Use the steps below to generate a new request.

1. Start the process of generating a new certificate signing request

   ```
   openssl req -new -key privatekey.pem -out certcsr.pem
   ```

   The program will now prompt you for the values to go into the certificate. Details on the values are below. The critical value is the commonname which must be the full hostname that users will use to reach your system. Example, if users reach your site as http://codec43.na.company.com your commonname should be **codec43.na.company.com**

a. The Organization Name and Unit are labels that will be stored in the certificate and should help identify the valid use of the certificate.
b. The Common Name is critical and must match the fully qualified host name that users use to access the web server – Example: codec43.na.corporation.com
c. The Geographical Information are labels that will appear in the certificate and should help identify the valid use of the certificate.
d. Leave the challenge password empty

The result is certcsr.pem which is a certificate signing request in PEM format. The resulting Certificate Signing Request file must now be sent to the Certificate Authority you wish to sign your Certificate (Verisign, Thawte, your internal IT Group, etc). Instructions on how to submit the certificate request will be provided by the Certificate Authority.

Once the certificate has been signed, it is installed using the same process as a new certificate.

Generating a Self-Signed Certificates using OpenSSL

To generate a self-signed certificate for a TANDBERG system rather than having a true Certificate Authority sign the certificate signing request, you follow all the same steps used to generate a key, signing request, and install the certificate outlined in Generating a Server Certificate for a TANDBERG System using OpenSSL with one exception. Instead of sending the certcsr.pem file to a Certificate Authority, you process the file yourself using OpenSSL. To self-sign a Certificate Signing Request, use the following command. **certcsr.pem** is your certificate signing request in PEM format. **privatekey.pem** is your private key in PEM format. *Days* is the number of days you'd like the certificate to be valid

```
openssl x509 -req -days 360 -in certcsr.pem -signkey privatekey.pem -out cert.pem
```

*The resulting **cert.pem** is your self-signed certificate*

Using the cert.pem file, you can complete the remainder of the installation steps as if the certificate were returned from a regular Certificate Authority.

Find if your certificate authority is trusted by the system

The TANDBERG VCS allows displaying of the currently loaded root certificate on the Maintanence>Security webpage. TANDBERG MXP and MPS systems allow listing the currently loaded root certificate using the certificate list root dataport command. In both cases, if the text version of the certificate was not included in the PEM file, you will have to save the text output as a PEM file and feed the file into OpenSSL to view the contents of the encoded file. Example, if you saved the contents as cert.pem, the OpenSSL command to view it would be:

```
openssl x509 –in cert.pem –noout –text
```

Adding a Certificate to the Trusted Root CA List for the system

This task must be completed using the web interface of the specific system. Please see the implementation guide section of this document for the specific system you are managing.

Viewing the contents of a Certificate File

If you need to verify the contents of a certificate, this can be done with the x509 commands of OpenSSL. To display the contents of a certificate, enter the command below, substituting **cert.pem** with your specific filename.

```
openssl x509 –in cert.pem –noout –text
```

OpenSSL will display the contents of the certificate.

Merging Multiple Root Certificates into a PEM file for use on TANDBERG Systems

When loading the root certificates for certificate authorities to trust to a TANDBERG system you must ensure the certificates are in the proper format and if you have multiple CAs to trust, combine the certificates into a single PEM file.

Converting to PEM format

PEM is the base-64 (text only) encoding of the DER binary form of an X.509 certificate with a header and footer line for each certificate. Most certificate authorities will provide their root certificate in both formats, DER and PEM. If the certificate file has a .cer extension, it may already a PEM file or can be a binary version (DER). Open the file with a text editor. If the file is a uniform block of text, with a header and footer as shown below, then the file is a PEM file.

```
-----BEGIN CERTIFICATE-----

asdfjkljgfldfkjalsdkjfalsdjfasdfasdf

-----END CERTIFICATE-----
```

If the certificate is not in PEM format, you can use the OpenSSL tools to convert a certificate into a PEM format. If you have a certificate in DER format named cert.der you can use the following command to copy it to PEM format

```
openssl x509 –in cert.der –inform DER –out cert.pem –outform PEM
```

Additionally, Windows can be used to covert .cer files to PEM. Files with a CER extension can be double-clicked in Windows and a properties window opens. Click on the *Details* tab, and clicking the *Copy to File* button opens the Certificate Export Wizard. The wizard will prompt for the file format to use and a filename to save the certificate as. Select *Base-64 encoded (.CER)* for the Export File Format and specify any filename. The resulting file will be a PEM encoded version of the original certificate.

Merging multiple certificates into a single file

Assuming you have two certificates to merge together, cert1.pem and cert2.pem, the following methods demonstrate how they can be combined into a single text file, combined.pem

On UNIX/Linux

1. Navigate to the directory containing the files. Catonate the files together using

```
cat cert1.pem cert2.pem > combined.pem
```

On Windows

1. Open a command prompt using the Start Menu or *Run..* option entering **cmd.exe**
2. Navigate to the directory containing the certificate files. Catonate the files together using

```
type cert1.pem  cert2.pem > combined.pem
```

These principles can be applied to combine more than 2 files as well. The resulting **combined.pem** file is the file you will upload to the TANDBERG system as the root certificate.