Dialogic®

Tech Note

Decoding HTTPS Traffic Between Wireshark and Dialogic® PowerMedia™ Extended Media Server (XMS)

NETWORK FUEL™

Introduction

When developing and debugging PowerMedia XMS RESTFul applications if may be necessary to review server to application messaging over HTTP. Wireshark (http://www.wireshark.org/) is perhaps the best known tool for capturing and analyzing IP packets. However, PowerMedia XMS will often use secure HTTP. (HTTPS) HTTPS packets captured by Wireshark are encrypted and cannot be read as captured.

Wireshark has facilities for decrypting packets from known sources, but setting up Wireshark to do so can be a challenge. To help, this technote provides step-by-step guidelines for capturing and viewing PowerMedia XMS HTTPS packet traffic, as well as some suggestions for viewing unencrypted HTTP.

The technote assumes the reader is familiar with using Wireshark.

Preliminaries

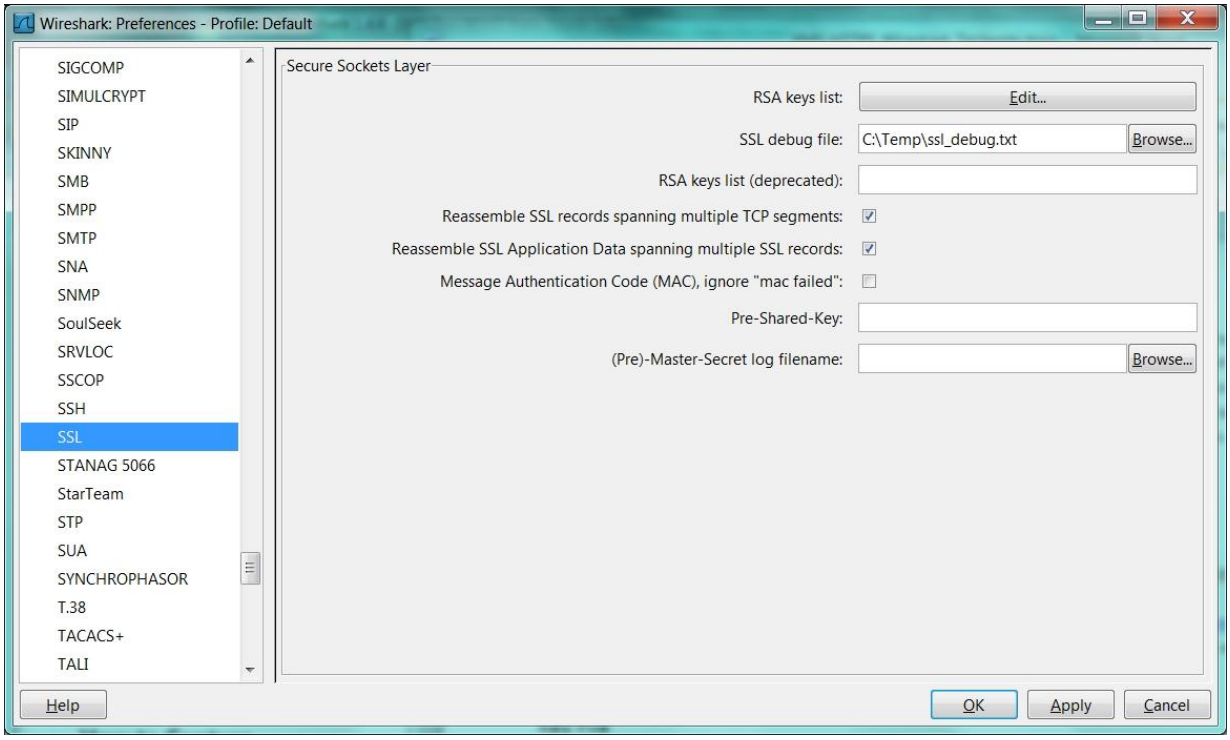
The latest stable version of Wireshark (1.8.6) for Windows, available as of April, 2013 was used for this technote. WinPCAP, the low-level packet capture utility used by Wireshark, was version 4.1.2.

Wireshark itself provides instructions for using Secure Socket Layer (SSL) components (http://wiki.wireshark.org/SSL) and there are a number of forums and blogs which cover the subject. These can be consulted for background information as what is presented here pertains directly to PowerMedia XMS.

Key File

In order to decrypt captured HTTPS packets, Wireshark must have access to the RSA private key used to encrypt it. This file is available on the PowerMedia XMS system in /etc/lighttpd/ssl/xms.key. SFTP into the system (default username/password is root/powermedia) and copy it to your Wireshark system.

On Wireshark, Edit → Preferences → Protocols. Scroll down to SSL and select it. The following screen will pop up:



An entry should now be added for associating the PowerMedia XMS IP address, HTTPS port, protocol and key file. Click on Edit → New and fill in the PowerMedia XMS IP address, port 443, http protocol and the full local path of the of the xms.key file copied from the PowerMedia XMS system. Add a second entry using port 10443 instead of 443. A valid entry for the SSL Debug File should also be set, as it is a comprehensive log of the Wireshark SSL subsystem. Selecting OK on both windows will save the entries.

Wireshark is now ready to decrypt PowerMedia XMS HTTPS traffic. Any further configuration will be done on the web browser or other client. Each case will be considered separately.

RESTful Management API Traffic

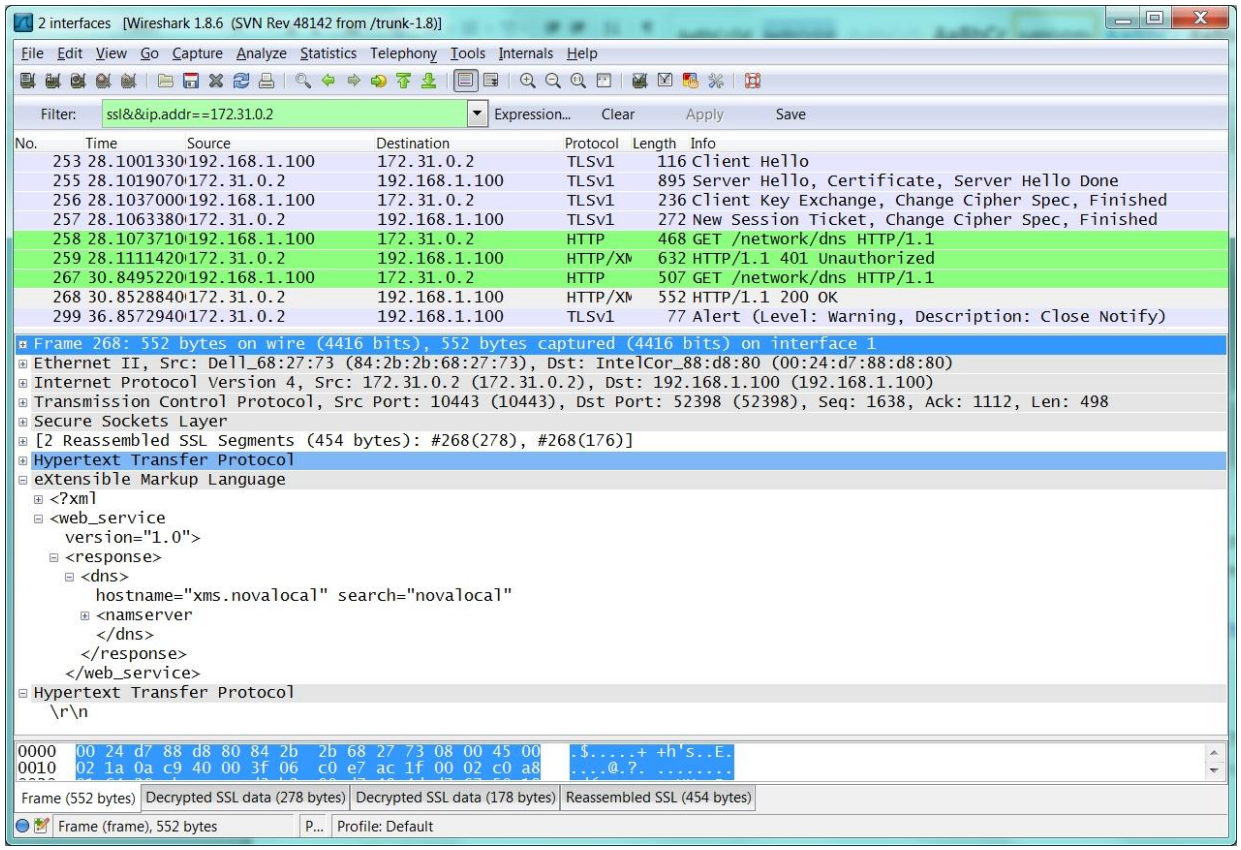
Decryption Using Firefox REST Client

The PowerMedia XMS RESTful Management API is used to provide a general purpose interface to operation, configuration and management of PowerMedia XMS. One easy way of exercising it is using the Firefox web browser with a RESTClient plug-in. Detailed instructions for setting up and using the RESTClient can be found in the PowerMedia XMS RESTful Management API Developer's Guide.

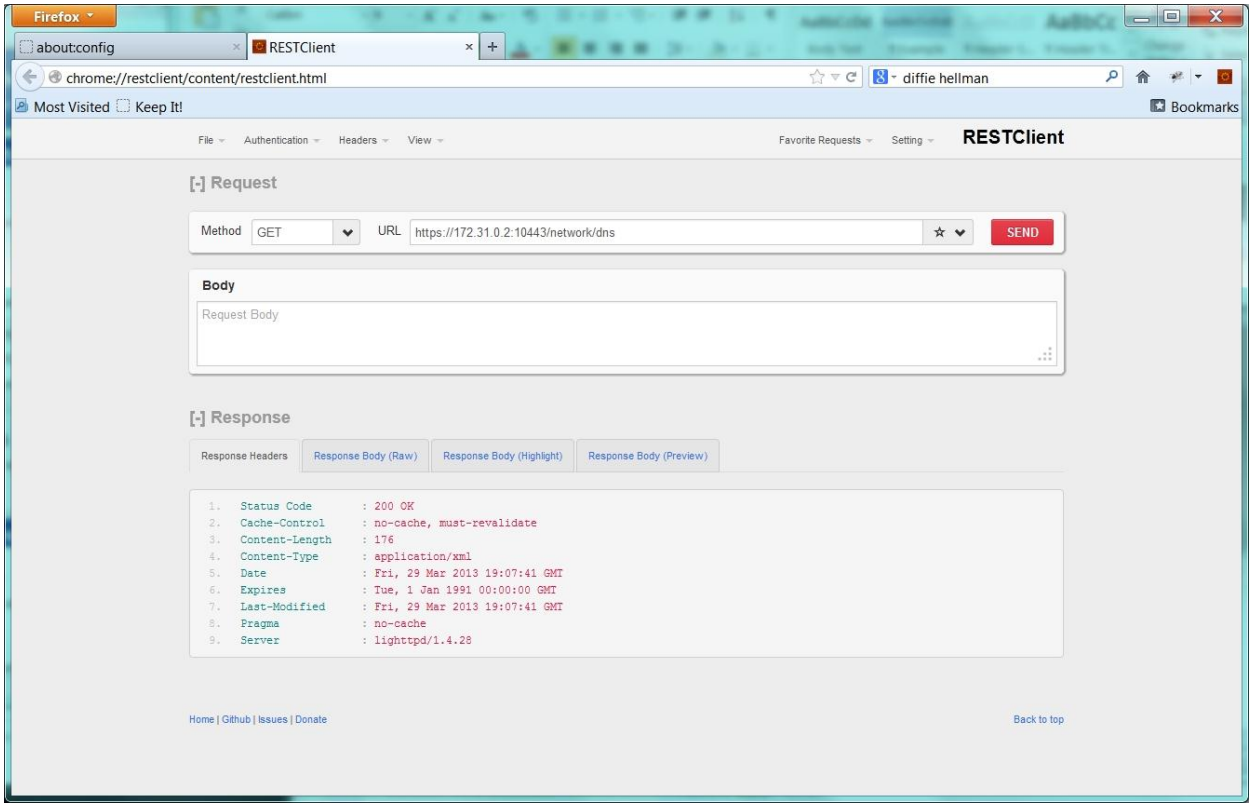
In addition, the choice of cipher suites offered by Firefox must be restricted as Wireshark is not able to decrypt if a Diffie-Hellman key exchange has been used. Thus, any cipher suite in Firefox with a "DH" or DHE" in its name should be eliminated. Parameter settings for Firefox can be found by going to the local URL about:config. Once there, find the encryption suites under "security.ssl3.suite_name". Double clicking on a line will toggle its value from true to false. To insure that a simple, non-Diffie-Hellaman suite is always chosen, disable all security.ssl3 entries except security.ssl3.rs_rc4_128_md5. To see the suites offered by the client, look into a Client Hello packet. Available cipher suites will be listed. The Server Hello that follows will indicate with suite was chosen.

For decryption to work, Wireshark must also capture the complete key exchange, which is done in a series of message exchanges. The best way of insuring this is to start the Wireshark capture before starting the Firefox web browser. Before starting Wireshark, make sure that the proper Ethernet interface is being monitored.

Then, start Firefox. Once it is running, try a simple RESTful Management request such as GET https://<xms_ip_addr>/network/dns:



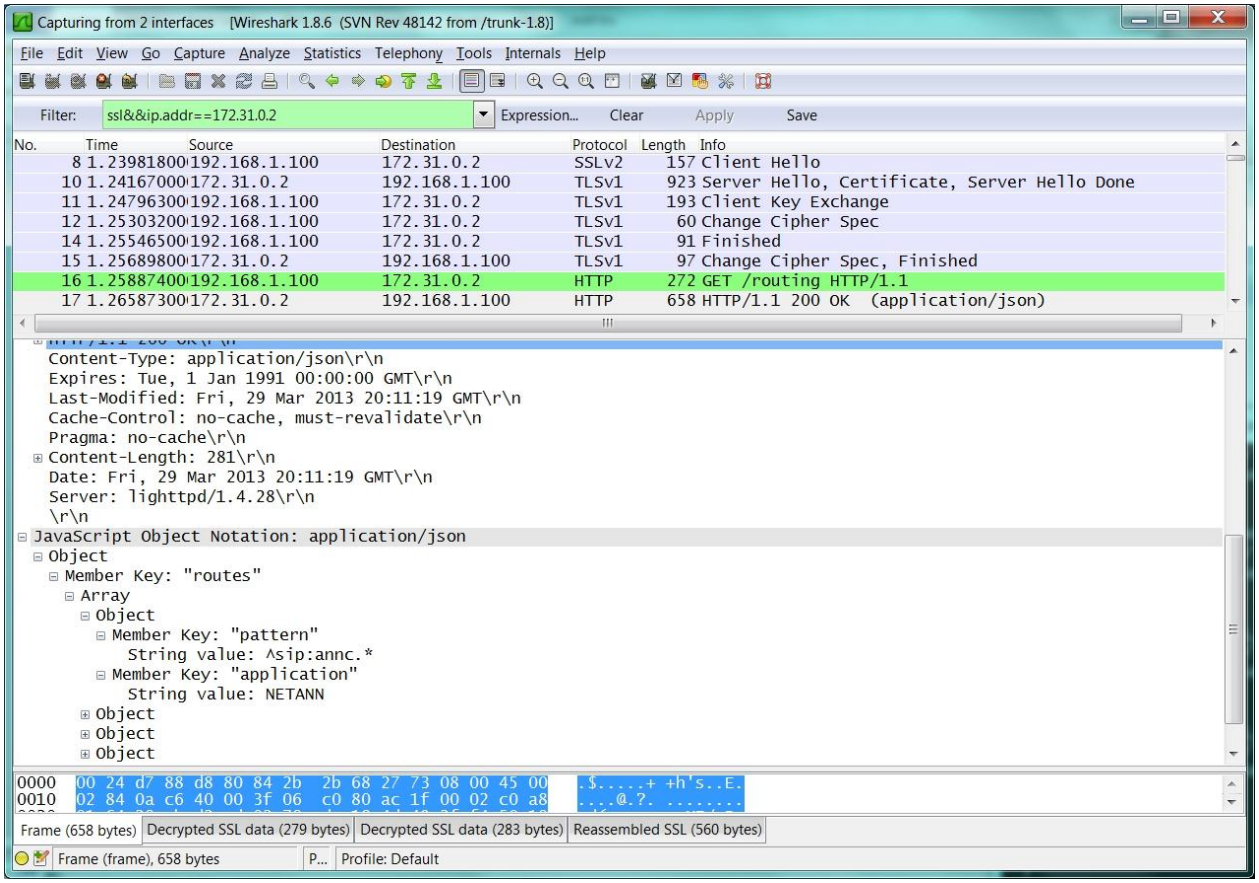
This will return the information seen in the Response Header and Response Body on the screen. On Wireshark, the HTTPS packets are found by filtering on "ssl" and any other relevant parameters, such as PowerMedia XMS IP address. The HTTP in the packet headers and body should be decoded and readable:



Decryption Using the Java Management Demo

A Java demo of how to use most aspects of the RESTful Management API is available. (See the technote "Using the Dialogic® PowerMedia™ Extended Media Server (XMS) 2.0 RESTFul Management API - A Java Demo) This uses the same RESTful API as the RESTClient demo mentioned above. Cipher suites are offered according to those available as part of Java Secure Socket Extensions. However, the first suite offered is the relatively simple TLS_RSA_WITH_RC4_128_MD5, which is the same that was recommended for use with the Firefox RESTClient. PowerMedia XMS will accept with this cipher suite and the results can be decoded by Wireshark.

Here is a decryption of a JSON payload requested by the Java demo:



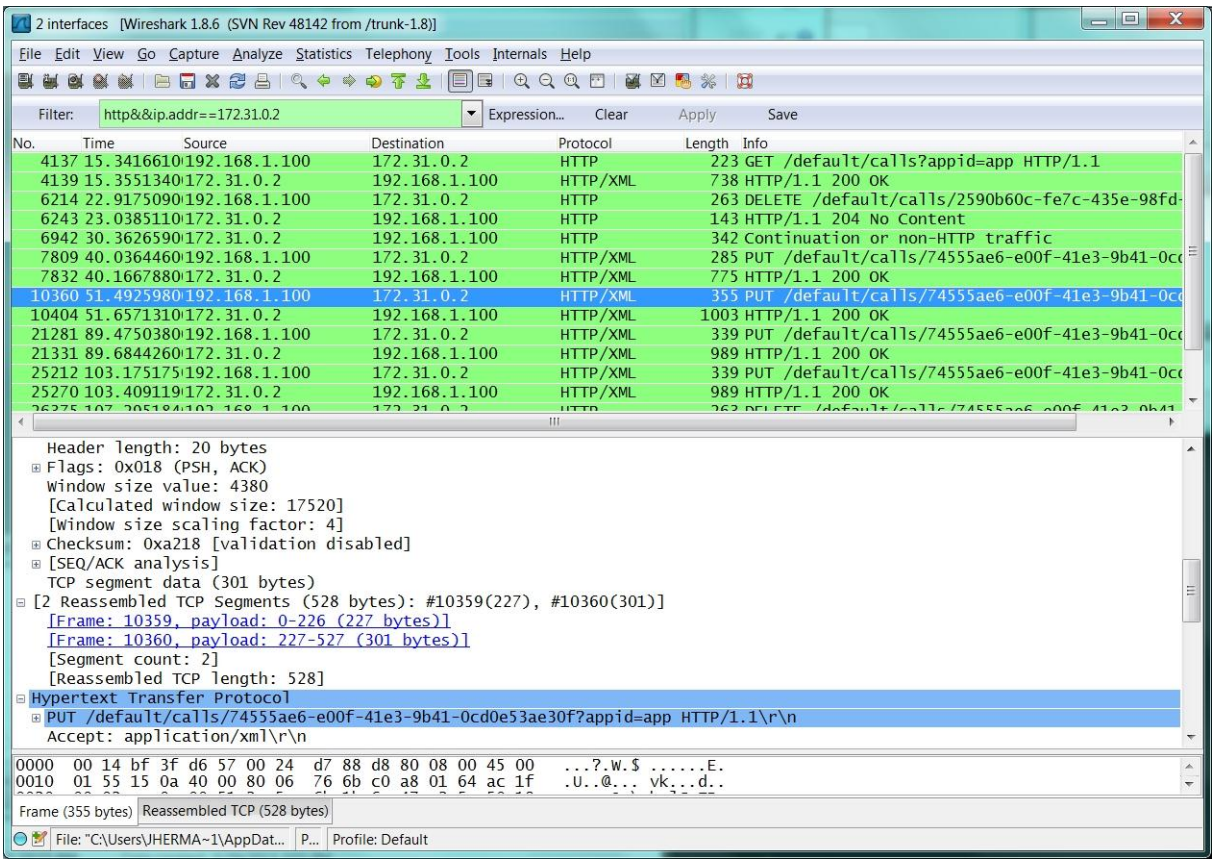
RESTful Call and Media Control API Traffic

The RESTful call and media control API is used by an application to control calls, audio and video media streaming and conferencing. It does not use HTTPS, making it easier to view in Wireshark.

Usually, some sort of filter is needed when looking at a Wireshark trace cut down on unwanted packets. For viewing HTTP, enter the following filter expressions in the Filter box:

- **http** – show only HTTP packets
- **http&&ip.addr==<xms_ip_address>** - only view PowerMedia XMS HTTP
- **http&&xml** - if a PowerMedia RESTful request contains XML, (such as a POST or PUT) this will help cut down on other unwanted HTTP

An example screen is shown below:



Open Source

This article discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Dialogic®

www.dialogic.com

Dialogic Inc
1504 McCarthy Boulevard
Milpitas, California 95035-7405
USA

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Inc. and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see http://www.dialogic.com/company/terms-of-use.aspx for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Inc. at the address indicated below or on the web at *www.dialogic.com*.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 6700 de la Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, Dialogic Blue, Veraz, Brooktrout, Diva, BorderNet, PowerMedia, ControlSwitch, I-Gate, Mobile Experience Matters, Network Fuel, Video is the New Voice, Making Innovation Thrive, Diastar, Cantata, TruFax, SwitchKit, Eiconcard, NMS Communications, SIPcontrol, Exnet, EXS, Vision, inCloud9, NaturalAccess and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Inc. and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 6700 de la Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners. 03/13