ETM® Unified Trunk Application (UTA)

Installation and Configuration Guide

For Version 7.0.2 ETM Application Software

Updated: 03/18/2013

About SecureLogix

SecureLogix, a Gartner designated "Cool Vendor" is the leader in enterprise voice/UC policy enforcement and ROI intelligence. SecureLogix 7th generation solutions enable customers to save money through securing and optimizing IP Telephony and legacy voice networks, allowing cost efficient and confident migration to SIP Trunking and Unified Communications. SecureLogix solutions are currently protecting and managing over three-and-a-half million enterprise phone lines.

The highly patented SecureLogix® ETM® System helps to secure, optimize and simplify the management of complex enterprise voice/UC networks through enterprise-wide voice network intelligence and unified policy enforcement. Available as an appliance-based solution or deployed via a software-only model running on the Cisco Enterprise router family, the ETM System enables a hard-dollar ROI payback in less than 12 months by securing the enterprise from attack, fraud, data leakage, financial losses and service abuse over TDM and VoIP (SIP) enterprise phone lines, while optimizing voice service and infrastructure expenses.

For more information about SecureLogix and its products and services, visit us on the Web at *www.securelogix.com* and *www.voipsecurityblog.com*.

Corporate Headquarters:
SecureLogix Corporation
13750 San Pedro, Suite 820
San Antonio, Texas 78232
Telephone: 210-402-9669 (non-sales)
Fax: 210-402-6996
Email: *info@securelogix.com*
Website: *http://www.securelogix.com*

Sales:
Telephone: 1-800-817-4837 (North America)
Email: *sales@securelogix.com*

Customer Support:
Telephone: 1-877-SLC-4HELP
Email: *support@securelogix.com*
Web Page: *http://support.securelogix.com*

Training:
Telephone: 210-402-9669
Email: *training@securelogix.com*
Web Page: *http://training.securelogix.com*

Documentation:
Email: *docs@securelogix.com*
Web Page: *http://support.securelogix.com*

IMPORTANT NOTICE:

This manual and the software and/or Products described in it are furnished under license with SecureLogix Corporation ("SecureLogix") and may be used only in accordance with the terms of such license.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without prior written permission of SecureLogix.

The content of this manual is subject to change without notice. SecureLogix assumes no responsibility or liability for any errors or inaccuracies that may be contained herein or to correct the same.

ETM, TeleWatch Secure, TWSA, We See Your Voice, SecureLogix, SecureLogix Corporation, the ETM Emblem, the SecureLogix Emblem and the SecureLogix Diamond Emblem are trademarks and/or service marks or registered trademarks and/or service marks of SecureLogix Corporation in the U.S.A. and other countries. All other trademarks mentioned herein are believed to be trademarks of their respective owners.

This product is protected by one or more of the following patents: US 6,249,575 B1, US 6,320,948 B1, US 6,687,353 B1, US 6,718,024 B1, US 6,760,420 B2, US 6,760,421 B2, US 6,879,671 B1, US 7,133,511 B2, US 7,231,027 B2, US 7,440,558 B2, US 8,150,013 B2, CA 2,354,149, DE 1,415,459 B1, FR 1,415,459 B1, and GB 1,415,459 B1.

ETM is used herein as shorthand notation to refer to the ETM® System.

This product includes:

Data Encryption Standard software developed by Eric Young (eay@mincom.oz.au),
 (see DESLicense.txt on ETM software CD)

Style Report software owned and licensed exclusively by InetSoft Technology Corp.

Software developed by The Apache Software Foundation (http://www.apache.org/)

(See ApacheLicense.txt on ETM software CD.)

Linux kernel software developed by Linus Torvalds and others; and Busy Box software developed by Bruce Perens and others. Distributed pursuant to the General Public License (GPL). See the Open Source Code directory on the ETM software CD for related copyrights, licenses, and source code.

GNU C Library software; Distributed pursuant to the Library General Public License (LGPL). See the Open Source Code directory on the ETM software CD for related copyrights, licenses, and source code.

Contents

1. Introduction 6

2. Getting Started 8

2.1 ETM® 5000-Series Appliance Hardware and Software Requirements 8

2.2 ISR Hardware Requirements 8

2.7 IOS Software Requirements 9

2.3 SRE-V Hardware Requirements 9

3. Performance 9

3.1 Cisco Hardware Performance considerations 9

3.2 Using this guide 10

4. SecureLogix® 5000-Series UTA Appliance Installation and Configuration 10

4.1 Installing the 5xxx Appliance 10

4.2 Installing ETM® software on the Appliance 10

4.3 Enabling the XCC and XSVC API 11

4.4 Configure UTA Provider Broadcasting 13

4.5 Configure Trunk Groups 14

4.5.1 VoIP (SIP) example: 14

4.5.2 TDM (PRI) example 15

4.5.3 TDM (CAS/Analog) example: 17

4.6 ETM UTA Call Direction Considerations: 18

4.6.1 Weak Call Direction 19

4.6.2 Strong Call Direction 19

4.7 Configuring the UTA Provider HTTP Server and Client 20

4.8 Apply Access Control Lists to the HTTP Service 22

4.9 Configuring Appliance ETM® software 23

6.0 SRE-V UTA installation and configuration 23

6.1 Install the SecureLogix UTA RPM's 24

6.2 Create a snapshot (Optional) 24

6.3 ETM UTA Configuration 25

6.3.1 Enabling the XCC and XSVC API 25

6.3.2 Configure UTA Provider Broadcasting 26

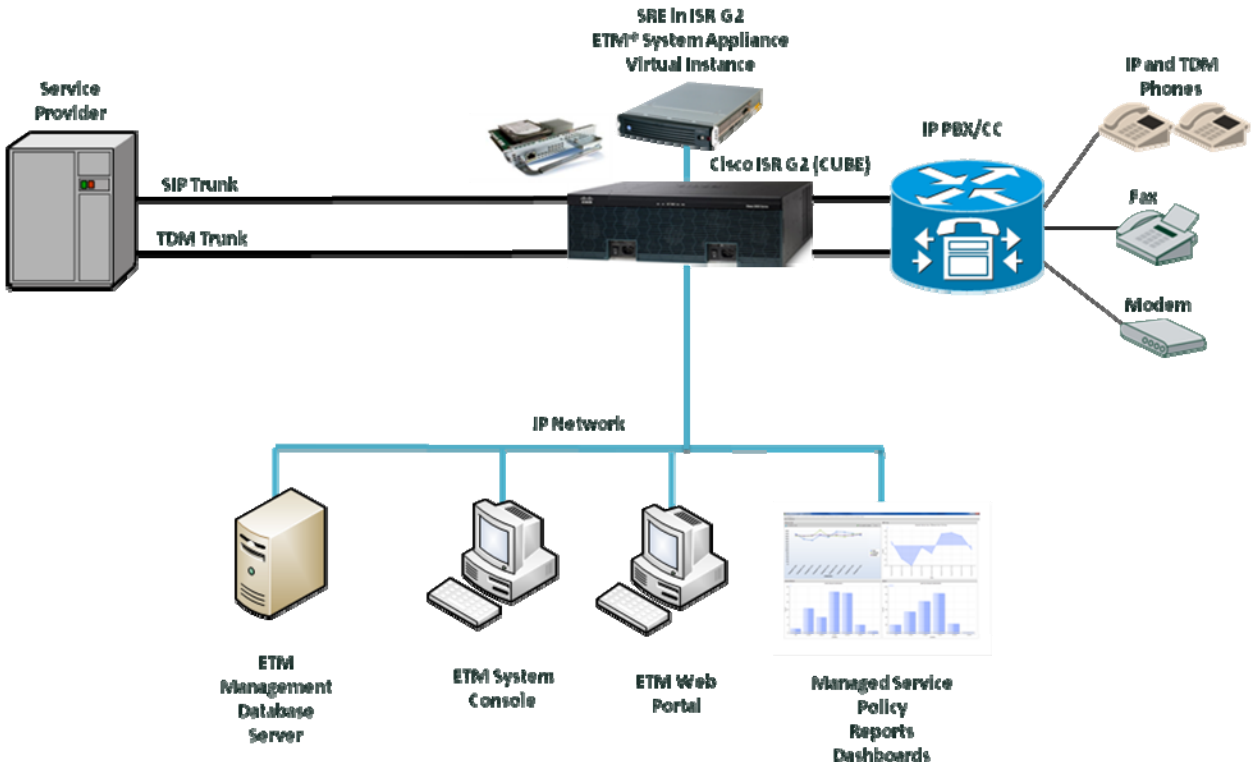
6.3.3 Trunk Configuration 27
6.3.3.1 VoIP (SIP) example: 27
6.3.3.2 TDM (PRI) example 29
6.3.3.3 TDM (CAS/Analog) example: 31
6.3.3.*4 ETM UTA Call Direction Considerations:* 32
6.4 Enable HTTP service on UTA Provider 33
6.5 Apply Access Control Lists to the HTTP Service 34
6.6 Configuring Appliance ETM® software 36
6.6.1 Confirm UTA Connection 36
7. Debugging and Diagnostics 37
7.1 Ensuring ETM® application to UTA provider connectivity 37
7.2 Debugging support 38
Addendum 39
Cisco IOS 15.1(2)+ Toll Fraud Features 39
Behavior Before 15.1(2)T 39
Behavior with 15.1(2)T and Later Releases 39
SIP URI Processing versus E164 on UTA Provider 39
Trunk Serviceability and Monitoring 39
ETM UTA TDM Trunk Status and D-Channel 39
Authorizing ICMP or SSH to external Network Management Devices 40
AXP Boot Loader config 41
AXP Boot Loader bypass on SRE only 41

1. Introduction

The SecureLogix® ETM® (Enterprise Telephony Management) System provides real-time voice security and usage policy enforcement, voice network management and monitoring tools, call recording, and enterprise-wide reporting. The ETM System is a distributed client/server/Voice Network Monitoring & Security (VNM&S) Application system. Expandable, managed VNM&S Applications are deployed on customer-premises devices on the voice network. These Applications are controlled by remote ETM Servers. The Servers and VNM&S Applications are managed from remote clients that can be used to manage multiple Servers and hundreds of VNM&S Applications that support various TDM and VoIP protocols and run on a variety of hardware platforms to support any voice network. These applications monitor and control voice traffic based on the ETM VNM&S Policies installed on them

The ETM Unified Trunk Application (UTA) is an implementation of the VNM&S Application that is integrated with the Cisco Integrated Services Router (ISR) G2 family via that is integrated with the Cisco Integrated Services Router (ISR) G2 family via the Cisco Unified Communications (UC) Gateway Services API available in router iOS versions 15.2(2)T and later. ETM UTA can be deployed on a Services-Ready Engine (SRE) module installed in the router and running SRE-V, on an ETM 5000 Series UTA appliance, or on customer-supplied COTS hardware that meets minimum system and resource requirements, such as a Cisco UCS. When deployed on a 5000-series UTA appliance or COTS hardware, the UTA application supports virtualization of either a single instance with call capacity stated for the selected 5000-Series Appliance, or of up to 15 instances on appropriately sized hardware (up 50 calls per instance in this model).

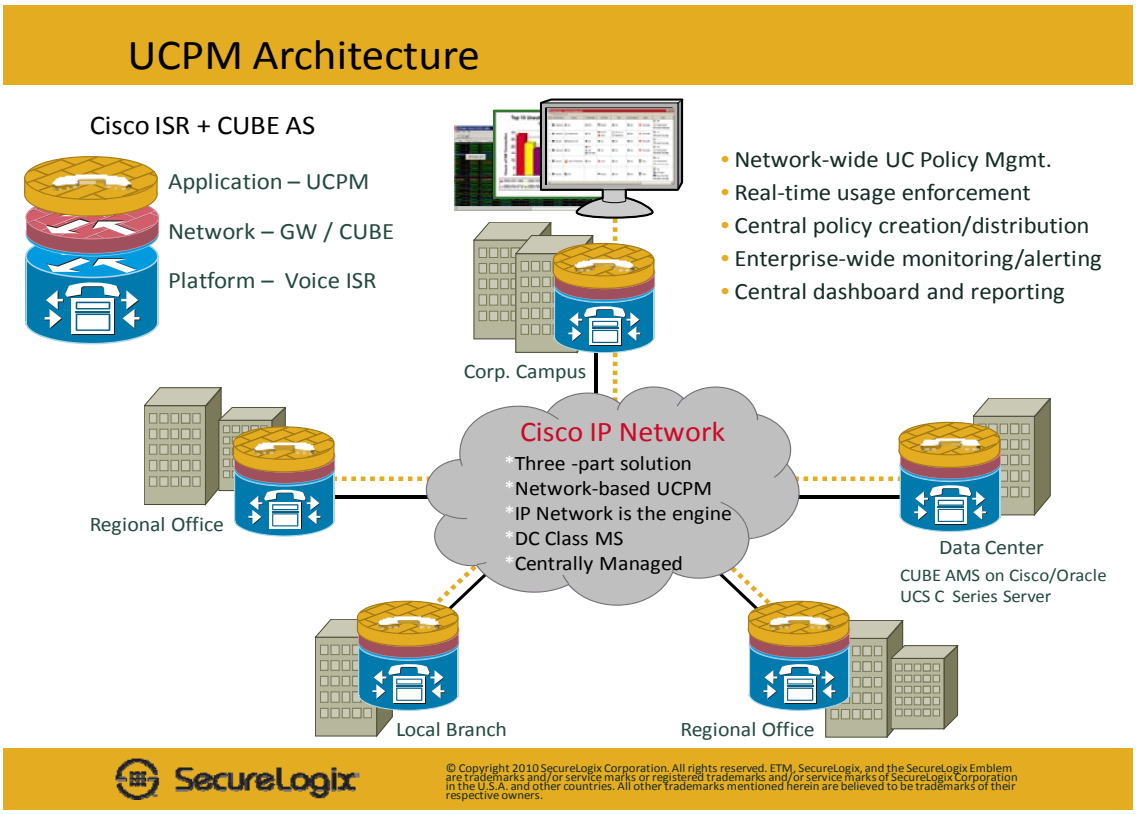
The illustration below provides a high-level overview of a sample ETM System UTA deployment.



The integration of the ETM UTA application with the Cisco ISR decouples the ETM Application from network and signaling specifics for both VoIP (Voice over Internet Protocol) and TDM (Time Division Multiplexing) traffic, providing full ETM System functionality without the need to be inline. Policy decision processing is performed by the ETM System, and Cisco routers, gateways, and software ensure network connectivity and integrity across various telephony protocols.

The UTA application is not inline with either signaling or media. Signaling and call information is exchanged through Web Services calls with the iOS API, and the API forks a copy of the media and sends it over a socket connection to the UTA Media Proxy (MP). The API alos provides call state call control functionality and trunk status to the UTA application.

When the UTA application is installed on an SRE module, Cisco SRE-V is used as the hosting environment to embed ETM® UTA application into the routers. The Cisco SRE_V hosting environment provides the infrastructure to securely host, install, upgrade, and manage the application. This integrated solution complements the existing ETM hardware instrumentation strategy by extending the ETM Appliance further out into the network to provide cost-effective management and security. This enhanced visibility further strengthens the Cisco Borderless Networks initiative that enables organizations and individuals to communicate anytime, anywhere, in any way they wish.



The audience of this document is technical. Please refer to the ETM System user guides for information pertaining to architecture and terminology used in this document.

2. Getting Started

The ETM® Appliance software has hardware and operational requirements that should be analyzed in deciding the best platform for which to run the application.

The ETM® Appliance software package is found on the CD provided. There are a number of initial steps that must be performed in order to load and ensure the software is running on your Cisco ISRG2 (AXP, SRE-V) or 5xxx Appliance.

2.1 ETM® 5000-Series Appliance Hardware and Software Requirements

ETM 5000-series UTA appliances are shipped with the ETM® Appliance operating system a tailored version of Linux CentOS 5.2) and UTA software (packaged as RPMs) already loaded and ready to be configured for the network on which they are to be deployed.

2.2 ISR Hardware Requirements

The UTA application supports only the ISR G2 family running the necessary version of iOS, using an SRE module, with SRE-V 5000-series UTA appliance, or virtual instances on COTS hardware to support SecureLogix UTA. When using an SRE, a 900/910 or greater module is recommended, although small

deployments can be supported on the SRE 700/710. A one-to-one relationship exists between a router and a UTA instance.

2.7 IOS Software Requirements

UTA requires IOS 15.2(2)T or later to be installed on the ISRG2 prior to installation of ETM® software. The current release that supports UTA is 15.2.4M3

Refer to Cisco IOS documentation for your specific ISRG2 for instructions on how to upgrade and install IOS software images.

2.3 SRE-V Hardware Requirements

Table 2. Cisco SRE Modules Support on Cisco Integrated Services Routers

Model	Max # of Cisco SRE SM	Cisco SRE 700/710 900/ 910
Cisco 2911	2	1
Cisco 2921	2	1
Cisco 2951	3	2
Cisco 3925	3	2
Cisco 3925E	2	2
Cisco 3945	5	4
Cisco 3945E	4	4

3. Performance

3.1 Cisco Hardware Performance considerations

The API that supports UTA represents additional processing requirements on the hosting Cisco platform. UTA uses a fairly verbose API to communicate with controlling parties and makes heavy use of DSP resources for media forking. An important factor determining the hardware platform is call capacity. The larger the concurrent call load, the faster the platform must be. The table below below provides the

tested call capacities for two supported SRE-V modules. The tests were performed using 2.5 minute calls with 5000 object Voice Firewall and IPS policies installed.

SRE-V Module	Concurrent Sessions	
	Signaling Only	Signaling Plus Call Recording (CR)
SRE 700	1000 calls	50 CR calls + 100 Signaling Only calls (150 total)
SRE 900	1500 calls	50 CR calls + 1000 Signaling Only calls (1050 total)

3.2 Using this guide

Please note there are numerous command line examples throughout the document. The table below indicates where the specific commands are intended to be executed and on which platform.

Command Prompt	Platform
#root> **[root@ETM ~]**	Execute commands on 5000 Series Appliances
C2951#	Execute commands on ISR Routers
10-1-1-20#	Execute command on the Service Module

4. SecureLogix® 5000-Series UTA Appliance Installation and Configuration

The 5000-series UTA appliance platform hosts ETM® Appliance software. A number of important steps must be taken to deploy and correctly configure ETM® Appliance software on this platform.

4.1 Installing the 5xxx Appliance

Ensure that the 5xxx series appliance has a fresh OS install as per ETM SIP application specifications.

4.2 Installing ETM® software on the Appliance

Configure the network settings for the ETM VM	
Step	**Command**
1	Obtain the following RPM packages: • **ETM_5003_callprocessor_<VERSION>.rpm** • **ETM_5003_callmanager_<VERSION>.rpm**

	• **ETM_5003_mediaproxy_<VERSION>.rpm**
2	Login remotely to the appliance as root and copy the RPM files to /root
3	Install of the RPM files using a terminal window on the 5000 series appliance using this syntax: **#root>rpm --force -ivh <RPM FILENAME>.rpm**
4	Using a terminal window on the ETM VM, Install the SLC Media Proxy RPM using this command: **# rpm -ivh --force ETM-mediaproxy-*.rpm**
5	Using a terminal window on the ETM VM, insure the RPMS are installed using this command: **#root>rpm –qa | grep –i etm** **ETM-callprocessor-<VERSION>** **ETM-callmanager-<VERSION>** **ETM-mediaproxy-<VERSION>**

4.3 Enabling the XCC and XSVC API

Turn on and enable all UTA and have them point to the correct ETM UTA application address that we will configure in subsequent steps.

Enabling the XCC and XSVC API on the Router		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# uc wsapi**	Enter the configuration mode for the UTA
4		

	Example: **C2951(config-uc)#message-exchange max-failures 1**	
5	Example: **C2951 (config-uc)# probing interval negative 10**	
6	C2951(config-uc)#**probing max-failures 2**	
7	C2951(config-uc)#**probing interval keepalive 60**	Determines how often a host is pinged to detect failures
8	**Call Control Provider** Example: **C2951 (config-uc)# provider xcc**	Specifies the Call Control Provider
9	Example: **C2951 (config-xcc)# remote-url http://10.1.1.20:8888**	Tells the provider to point to the ETM appliance IP:UTA port
10	**Exit Interface Config** Example: **C2951 (config- xcc)#exit**	Returns to global configuration mode
11	**Specify Status Provider** Example: **C2951 (config-uc)# provider xsvc**	Specifies the Status Provider
12	Example: **C2951 (**config-xsvc**)# remote-url 1 http://10.1.1.20:8889**	Tells the status provider to point to the ETM appliance IP:UTA port
13	Example: **C2951 (**config-xsvc**)#end**	Ends the configuration mode
14	**Save config**	Saves the configuration

	Example: **C2951 #wr**	

4.4 Configure UTA Provider Broadcasting

When the UTA Provider restarts via a hardware reload or hardware restart it must broadcast the fact that it is going away to the ETM UTA application to properly clear active calls from ETM tracking. To enable this feature the Provider must be configured.

Configure UTA Provider Broadcasting		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# uc wsapi**	Enter the configuration mode for the UTA
4	Example: **C2951 (config-uc)#source 10.1.1.254**	Router/Provider IP that is used for UTA communications
5	Example: **C2951 (config-uc)#end**	Ends the configuration mode
6	**Save config** Example: **C2951 #wr**	Saves the configuration

4.5 Configure Trunk Groups

Now that Call Control (XCC) is turned for all Trunking on that router it is critical that you create and define VoIP trunk groups which will be used for Health & Status, DB reporting and many other critical ETM services. Moreover, definition of these elements is critical as it establishes Call Direction contexts via description markers, so that policy execution can take place correctly. You can configure more than one trunk or trunk pair.

4.5.1 VoIP (SIP) example:

SIP Trunk Configuration Example		
Step	**Command**	**Description**
1	**Enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# voip trunk group AT&T**	Create a VoIP Trunk named AT&T
4	**Enable the trunk created** Example: **C2951 (config-voip)#xsvc**	Enable this trunk
5	**Mark the trunk as External** Example: **C2951 (config-uc)# description EXTERNAL**	Mark this trunk as EXTERNAL
6	**Attach trunk to dial-peer** Example: **C2951 (config-voip)# session target ipv4:10.1.2.237**	Attach it to a dial-peer
7	**Exit trunk configuration** Example: **C2951 (config-voip)#exit**	

8	**Exit Interface Config** Example: **C2951 (config)# voip trunk group CUCM**	Create a VoIP Trunk named CUCM
9	 Example: **C2951 (config-voip)#xsvc**	Enable this trunk
10	 Example: **C2951 (config-voip)#description INTERNAL**	Mark this trunk as INTERNAL
11	 Example: **C2951 (config-voip)#session target ipv4:10.1.2.248**	Attach it to a dial-peer
12	 Example: **C2951 (config-if)#end**	Ends the configuration mode
13	**Save config** Example: **C2951 #wr**	Saves the configuration

4.5.2 TDM (PRI) example

TDM (PRI) Trunk Configuration Example		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3		create a PRI Trunk named PRI_TRUNK

	Example: **C2951(config)# trunk group PRI_TRUNK**	
4	**Enable the trunk created** Example: **C2951 (config-trunk)#xsvc**	Enable this trunk
5	**Mark the trunk as External** Example: **C2951 (config-trunk)# description EXTERNAL**	Mark this trunk as EXTERNAL
6	Example: **C2951 (config-trunk)#exit**	
7	**Configure the serial interface for this trunk** Example: **C2951 (config)#interface Serial0/0/1:23**	Configure the serial interface for this trunk
8	Example: **C2951 (config-interface)# no ip address**	
9	Example: **C2951 (config-interface)# encapsulation hdlc**	
10	Example: **C2951 (config-interface)# isdn switch-type primary-ni**	
11	Example: **C2951 (config-interface)# isdn incoming-voice voice**	
12	Example:	Map it to the trunk group

	C2951 (config-interface)# trunk-group PRI_TRUNK	
	Example: **C2951 (config-interface)# no cdp enable**	
	Example: **C2951 (config-interface)# end**	Ends the configuration mode
13	**Save config** Example: **C2951 #wr**	Saves the configuration

4.5.3 TDM (CAS/Analog) example:

TDM (PRI) Trunk Configuration Example		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# trunk group ANALOG_TRUNK**	create a Analog Trunk named ANALOG_TRUNK
4	**Enable the trunk created** Example: **C2951 (config-trunk)#xsvc**	Enable this trunk
5	**Mark the trunk as External** Example:	Mark this trunk as **INTERNAL**

	C2951 (config-trunk)# description INTERNAL	
6	Example: **C2951 (config-trunk)#exit**	
7	**Configure the serial interface for this trunk** Example: **C2951 (config)#voice-port 0/0/0**	Configure the voice port for this trunk
8	Example: **C2951 (config-voice)# secondary dialtone**	
9	Example: **C2951 (config-voice)# ring-number 2**	
10	Example: **C2951 (config-voice)# caller-id enable**	
11	Example: **C2951 (config-voice)# trunk-group ANALOG_TRUNK**	Map it to the trunk group
12	Example: **C2951 (config-interface)# end**	Ends the configuration mode
13	**Save config** Example: **C2951 #wr**	Saves the configuration

4.6 ETM UTA Call Direction Considerations:

It is important that you carry over the EXTERNAL and INTERNAL markings over to the Management Server configuration to ensure that the Provider configuration and ETM application are in synch.

The following are the Call Direction rules that the UTA applications apply when processing Call Information into and out of the UTA provider:

4.6.1 Weak Call Direction

Weak call direction, call direction relative to the router, is an initial call direction based only on information for the caller leg. Weak call direction must be determined to support policy processing if only the caller leg information is available at the time policy processing must be performed. An example of its use case is Firewall reject processing

The rules are included in the following table and are based on the assumption that most traffic will either be Internal→External or External→Internal.

Connection 1		Connection 2		Call Direction
Location	**Direction**	**Location**	**Direction**	
Internal	Outbound	Undefined	Undefined	Outbound
External	Inbound	Undefined	Undefined	Inbound
Undefined	Undefined	Internal	Inbound	Inbound
Undefined	Undefined	External	Outbound	Outbound
Internal	Inbound	Undefined	Undefined	Outbound
External	Inbound	Undefined	Undefined	Inbound
Undefined	Inbound	Undefined	Undefined	Inbound

4.6.2 Strong Call Direction

Strong call direction, call direction relative to the organization, requires data for both connections. Strand call direction is determined in a two-step process. First the direction of each of the call's connections is determined, and then the direction of the call itself is determined from the connection directions.

To determine call direction relative to the organization to ETM application follows 2 steps:

1) Determine the direction of each connection relative to the call using the **<connDirectionType>** and **<routeDescription>** element values communicated via the XSVC UTA API and the following rules.
   a) If a Connection is Incoming (Caller leg) and the ETM configuration of the trunk for which that Connection belongs to is an "Internal" trunk then the Connection Direction is OUTGOING.
   b) If a Connection is Outgoing (Callee leg) and the ETM configuration of the trunk for which that Connection belongs to is an "External" trunk then the Connection Direction is OUTGOING.
   c) If a Connection is Incoming (Caller leg) and the ETM configuration of the trunk for which that Connection belongs to is an "External" trunk then the Connection Direction is INCOMING.
   d) If a Connection is Outgoing (Callee leg) and the ETM configuration of the trunk for which that Connection belongs to is an "Internal" trunk then the Connection Direction is INCOMING.

e) If a Connection's trunk has no matching ETM configuration that can mark is as a INTERNAL or EXTERNAL connection, then the Connection's Direction is UNKNOWN.

2) Determine the direction of the call using the connection directions determined in the first step and the following rules.

a) Normal Case:

Connection 1		Connection 2		Call Direction
Location	**Direction**	**Location**	**Direction**	
Internal	Outbound	External	Outbound	Outbound
External	Inbound	Internal	Inbound	Inbound

b) Internal and Tandem Calling (in lieu of having a Direction "Internal" and "Tandem"):

Connection 1		Connection 2		Call Direction
Location	**Direction**	**Location**	**Direction**	
Internal	Outbound	Internal	Inbound	Outbound
External	Inbound	External	Outbound	Outbound

c) Undefined Cases (the directions for the first four are based upon the assumption that most traffic we see will either be Internal->External or External->Internal):

Connection 1		Connection 2		Call Direction
Location	**Direction**	**Location**	**Direction**	
Internal	Outbound	Undefined	Undefined	Outbound
External	Inbound	Undefined	Undefined	Inbound
Undefined	Undefined	Internal	Inbound	Inbound
Undefined	Undefined	External	Outbound	Outbound
Undefined	Undefined	Undefined	Undefined	Inbound

4.7 Configuring the UTA Provider HTTP Server and Client

The HTTP server and client on the router must be enabled to allow UTA to operate. Additional settings that define timeout policies for application failures are also advised.

Enable HTTP service on UTA Provider		
Step	**Command**	**Description**

1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Enable HTTP** Example: **C2951(config)# ip http server**	Enables HTTP Server
4	**Configure max connections** Example: **C2951 (config)# ip http max-connections 200**	Limits the maximum number of connections
5	**Ensure connection is persistent** Example: **C2951 (config)# http client connection persistent**	Persistent indicates that the application needs the HTTP client to maintain connection after data transfer from itself to the remote server
6	**Configure timeout** Example: **C2951 (config)# ip http client connection idle timeout 60**	Determine how long a connection can be idle before timing it out
7	**Configure application connection timeout** C2951(config)# **http client connection idle timeout 60**	
8	**Configure application connection/server timeout response** C2951(config)# **http client connection response timeout 10** C2951(config)# **ip http timeout-policy idle 60 life 86400 requests 86400**	Defines the HTTP timeout for requests. This value dictates how long calls could be held in cases when an application fails. A higher value is required under load.
9	**End configuration** Example: **C2951 (config-if)#end**	Ends the configuration mode

10	**Save configuration** Example: **C2951 #wr**	Saves the configuration

IMPORTANT NOTE: The above settings affect how quickly the UTA provider will detect and take back call control in the event of application un-responsiveness. The above settings are a good default however if your particular deployment mandates for example that call setup cannot be delayed more than 1 second - for any reason - you would adjust the http client connection timeout parameters to 1 versus 2.

Moreover, if the connection path between the UTA provider and the application could have latency that exceeds your settings it is advised to increase the timeouts to ensure that the Call control events are given sufficient time to reach the ETM application and for the ETM application to do its Call controlling processing avoiding the UTA provider from abruptly taking back call control.

4.8 Apply Access Control Lists to the HTTP Service

The HTTP server operates unprotected by default. It is advisable that IP ACL's are used to "Lock-down" access to and from the HTTP service on the Provider.

Apply Access Control Lists to the HTTP Service		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Enable HTTP** Example: **C2951(config)# access-list 101 permit tcp host <ETM appliance IP> <Router IP> 0.0.0.0 eq 8090** **C2951(config)# access-list 101 permit tcp host 10.1.1.25 10.1.1.254 eq 8090**	Enables HTTP Server
4	Example: **C2951 (config)# access-list 101 deny tcp any <Router IP> 0.0.0.0 eq 8090**	

	C2951(config)# access-list 101 deny tcp any 10.1.1.254 eq 8090	
5	Example: **C2951 (config)# access-list 101 permit tcp any any**	
6	Example: **C2951 (config)# interface <IP interface>**	Apply ACL to the UTA communication interface
7	Example: **C2951 (config)# ip access-group 101 in**	
8	Example: **C2951 (config-if)#end**	Ends the configuration mode
9	**Save config** Example: **C2951 #wr**	Saves the configuration

4.9 Configuring Appliance ETM® software

Current ETM® Appliance software is designed to be initially configured via its own CLI prior to entering an operational state. Step through the ETM configuration wizard. After it is completed, configure your ETM® Appliance software UTA provider Addresses and Ports as you specified in **Section 4.3** in the Management Server. Please refer to the ETM System user guides for further information.

[root@ETM ~]$**cd opt/slc**
[root@ETM ~]$**./ETM_5000_configure.pl**

6.0 SRE-V UTA installation and configuration

The SRE-V platform hosts ETM® Appliance software on a virtual machine created by SecureLogix. A Service Module hosts the SRE-V platform. A Cisco ISRG2 hosts a Service Module. This tiered hosted approach introduces several layers of configuration management. The current ETM® Appliance software does not automatically provision and install itself. Regardless, a number of important steps must be taken to deploy and correctly configure ETM® Appliance software on this platform. This guide presumes the user has followed the steps in the "**ETM Application on Cisco Services-Ready Engine Virtualization Installation and Configuration Guide**" document where the user has finished installing and configuring the SecureLogix virtual machine in the SRE-V environment.

6.1 Install the SecureLogix UTA RPM's

When the VM has connectivity to the network, you can move the install the SecureLogix UTA RPM's onto the VM. The RPM's **should already be installed** on the VM, but this step is included in the event they are not. To do this use the vClient software and follow these steps:

Configure the network settings for the ETM VM	
Step	**Command**
1	Copy the ETM RPM files from the FTP server to the / directory using this command in a terminal server window on the virtual machine: **# scp admin@10.1.1.185:/home/admin/ETM* /.** **# admin@10.1.1.185's password:**
2	Using a terminal window on the ETM VM, Install the SLC Callprocessor RPM using this command: **# rpm -ivh --force ETM-callmanager-*.rpm**
3	Using a terminal window on the ETM VM, Install the SLC Signal Proxy RPM using this command: **# rpm -ivh --force ETM-callprocessor-*.rpm**
4	Using a terminal window on the ETM VM, Install the SLC Media Proxy RPM using this command: **# rpm -ivh --force ETM-mediaproxy-*.rpm**
5	Using a terminal window on the ETM VM, insure the RPMS are installed using this command: # rpm –qa | grep -i etm **ETM-callprocessor-<VERSION>** **ETM-callmanager-<VERSION>** **ETM-mediaproxy-<VERSION>**

6.2 Create a snapshot (Optional)

After the RPMS are installed on the VM, you may optionally create a snapshot of the installation in its initial state. To do this use the vClient software and follow these steps:

Configure the network settings for the ETM VM	
Step	**Command**
1	Shut down the VM in the terminal window by performing the following command: **#init 0**
2	When the VM has shut down, use vClient to right-click on the VM and select “Snapshot” in the pop-up menu and then select “take snapshot”
3	Create a name for the snapshot and include a brief description.
4	Click on OK when finished.

6.3 ETM UTA Configuration

If you are using the ETM UTA software, you will have to ensure the router has been configured correctly to ensure that the appliance will be able to communicate with it.

6.3.1 Enabling the XCC and XSVC API

Turn on and enable all UTA and have them point to the correct ETM UTA application address that we will configure in subsequent steps.

Enabling the XCC and XSVC API on the Router		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# uc wsapi**	Enter the configuration mode for the UTA
4	Example:	

	C2951 (config-uc)#message-exchange max-failures 2	
5	Example: **C2951 (config-uc)# probing interval negative 10**	
6	**Call Control Provider** Example: **C2951 (config-uc)# provider xcc**	Specifies the Call Control Provider
7	Example: **C2951 (config-xcc)# remote-url http://10.1.1.25:8888**	Tells the provider to point to the ETM appliance IP:UTA port
8	**Exit Interface Config** Example: **C2951 (config- xcc)#exit**	Returns to global configuration mode
9	**Specify Status Provider** Example: **C2951 (config-uc)# provider xsvc**	Specifies the Status Provider
10	Example: **C2951 (**config-xsvc**)# remote-url 1 http://10.1.1.25:8889**	Tells the status provider to point to the ETM appliance IP:UTA port
11	Example: **C2951 (**config-xsvc**)#end**	Ends the configuration mode
12	**Save config** Example: **C2951 #wr**	Saves the configuration

6.3.2 Configure UTA Provider Broadcasting

When the UTA Provider restarts via a hardware reload or hardware restart it must broadcast the fact that it is going away to the ETM UTA application to properly clear active calls from ETM tracking. To enable this feature the Provider must be configured.

Configure UTA Provider Broadcasting		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# uc wsapi**	Enter the configuration mode for the UTA
4	Example: **C2951 (config-uc)#source 10.1.1.254**	Router/Provider IP that is used for UTA communications
5	Example: **C2951 (config-uc)#end**	Ends the configuration mode
6	**Save config** Example: **C2951 #wr**	Saves the configuration

6.3.3 Trunk Configuration

Now that Call Control (XCC) is turned for all Trunking on that router it is critical that you create and define VoIP trunk groups which will be used for Health & Status, DB reporting and many other critical ETM services. Moreover, definition of these elements is critical as it establishes Call Direction contexts via description markers, so that policy execution can take place correctly. You can configure more than one trunk or trunk pair.

6.3.3.1 VoIP (SIP) example:

SIP Trunk Configuration Example		
Step	**Command**	**Description**

1	**Enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# voip trunk group AT&T**	Create a VoIP Trunk named AT&T
4	**Enable the trunk created** Example: **C2951 (config-voip)#xsvc**	Enable this trunk
5	**Mark the trunk as External** Example: **C2951 (config-uc)# description EXTERNAL**	Mark this trunk as EXTERNAL
6	**Attach trunk to dial-peer** Example: **C2951 (config-voip)# session target ipv4:10.1.2.237**	Attach it to a dial-peer
7	**Exit trunk configuration** Example: **C2951 (config-voip)#exit**	
8	**Exit Interface Config** Example: **C2951 (config)# voip trunk group CUCM**	Create a VoIP Trunk named CUCM
9	 Example: **C2951 (config-voip)#xsvc**	Enable this trunk
10		Mark this trunk as INTERNAL

	Example: **C2951 (config-voip)#description INTERNAL**	
11	Example: **C2951 (config-voip)#session target ipv4:10.1.2.248**	Attach it to a dial-peer
12	Example: **C2951 (config-if)#end**	Ends the configuration mode
13	**Save config** Example: **C2951 #wr**	Saves the configuration

6.3.3.2 TDM (PRI) example

TDM (PRI) Trunk Configuration Example		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	Example: **C2951(config)# trunk group PRI_TRUNK**	create a PRI Trunk named PRI_TRUNK
4	**Enable the trunk created** Example: **C2951 (config-trunk)#xsvc**	Enable this trunk
5	**Mark the trunk as External**	Mark this trunk as EXTERNAL

	Example: **C2951 (config-trunk)# description EXTERNAL**	
6	Example: **C2951 (config-trunk)#exit**	
7	**Configure the serial interface for this trunk** Example: **C2951 (config)#interface Serial0/0/1:23**	Configure the serial interface for this trunk
8	Example: **C2951 (config-interface)# no ip address**	
9	Example: **C2951 (config-interface)# encapsulation hdlc**	
10	Example: **C2951 (config-interface)# isdn switch-type primary-ni**	
11	Example: **C2951 (config-interface)# isdn incoming-voice voice**	
12	Example: **C2951 (config-interface)# trunk-group PRI_TRUNK**	Map it to the trunk group
	Example: **C2951 (config-interface)# no cdp enable**	
	Example:	Ends the configuration mode

	C2951 (config-interface)# end	
13	**Save config** Example: **C2951 #wr**	Saves the configuration

6.3.3.3 TDM (CAS/Analog) example:

TDM (PRI) Trunk Configuration Example		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Configure UTA** Example: **C2951(config)# trunk group ANALOG_TRUNK**	create a Analog Trunk named ANALOG_TRUNK
4	**Enable the trunk created** Example: **C2951 (config-trunk)#xsvc**	Enable this trunk
5	**Mark the trunk as External** Example: **C2951 (config-trunk)# description INTERNAL**	Mark this trunk as **INTERNAL**
6	Example: **C2951 (config-trunk)#exit**	
7	**Configure the serial interface for this trunk** Example: **C2951 (config)#voice-port 0/0/0**	Configure the voice port for this trunk

8	Example: **C2951 (config-voice)# secondary dialtone**	
9	Example: **C2951 (config-voice)# ring-number 2**	
10	Example: **C2951 (config-voice)# caller-id enable**	
11	Example: **C2951 (config-voice)# trunk-group ANALOG_TRUNK**	Map it to the trunk group
12	Example: **C2951 (config-interface)# end**	Ends the configuration mode
13	**Save config** Example: **C2951 #wr**	Saves the configuration

6.3.3.4 ETM UTA Call Direction Considerations:

It is important that you carry over the EXTERNAL and INTERNAL markings over to the Management Server configuration to ensure that the Provider configuration and ETM application are in synch.

The following are the Call Direction rules that the UTA applications apply when processing Call Information into and out of the UTA provider:

Undefined is defined as a Trunk that does NOT have a description marker (due to operator ERROR) or does NOT have a description marker that is in-synch with the UTA application configuration.

Normal Case:

- Internal -> External => Outbound
  External -> Internal => Inbound

Internal and Tandem Calling (in lieu of having a Direction "Internal" and "Tandem"):

- Internal -> Internal => Outbound
  External -> External => Outbound

Undefined Cases (the directions for the first four are based upon the assumption that most traffic we see will either be Internal->External or External->Internal):

- Internal -> Undefined => Outbound
  External -> Undefined => Inbound
  Undefined -> Internal => Inbound
  Undefined -> External => Outbound
  Undefined -> Undefined => Inbound

The UTA application evaluates Firewall/IPS policy in the Call Rejection case when it receives the Caller Leg and has no Callee Leg yet. At this moment it needs to assign the call a direction to the Call. Given that it does NOT have a Callee Leg yet, it cannot make an accurate Call direction determination as per the Trunk direction markings at this point. Given that the Callee Leg can be inferred to have an Unknown direction, the UTA applications will apply the following rules during for this Call as per:

- Internal -> Undefined => Outbound
  External -> Undefined => Inbound
  Undefined -> Undefined => Inbound

6.4 Enable HTTP service on UTA Provider

The HTTP server on the router must be enabled to allow CAUGA to operate.

Enable HTTP service on UTA Provider		
Step	**Command**	**Description**
1	**enable** Example: **C2951#enable**	Enters EXEC mode on the router
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Enable HTTP** Example: **C2951(config)# ip http server**	Enables HTTP Server
4	**Configure max connections** Example:	Limits the maximum number of connections

	C2951 (config)# ip http max-connections 100	
5	**Ensure connection is persistent** Example: **C2951 (config)# http client connection persistent**	Persistent indicates that the application needs the HTTP client to maintain connection after data transfer from itself to the remote server
6	**Configure timeout** Example: **C2951 (config)# ip http client connection idle timeout 60**	Determine how long a connection can be idle before timing it out
7	**Configure application connection timeout** C2951(config)# **http client connection idle timeout 60**	
8	**Configure application connection/server timeout response** C2951(config)# **http client connection response timeout 10** C2951(config)# **ip http timeout-policy idle 60 life 86400 requests 86400**	Defines the HTTP timeout for requests. This value dictates how long calls could be held in cases when an application fails. A higher value is required under load.
9	**End configuration** Example: **C2951 (config-if)#end**	Ends the configuration mode
10	**Save configuration** Example: **C2951 #wr**	Saves the configuration

6.5 Apply Access Control Lists to the HTTP Service

The HTTP server operates unprotected by default. It is advisable that IP ACL's are used to "Lock-down" access to and from the HTTP service on the Provider.

Apply Access Control Lists to the HTTP Service		
Step	**Command**	**Description**
1	**enable** Example:	Enters EXEC mode on the router

	C2951#enable	
2	**configure terminal** Example: **C2951#config terminal**	Enters configuration mode
3	**Enable HTTP** Example: **C2951(config)# access-list 101 permit tcp host <ETM appliance IP> <Router IP> 0.0.0.0 eq 8090** **C2951(config)# access-list 101 permit tcp host 10.1.1.25 10.1.1.254 eq 8090**	Enables HTTP Server
4	Example: **C2951 (config)# access-list 101 deny tcp any <Router IP> 0.0.0.0 eq 8090** **C2951(config)# access-list 101 deny tcp any 10.1.1.254 eq 8090**	
5	Example: **C2951 (config)# access-list 101 permit tcp any any**	
6	Example: **C2951 (config)# interface <IP interface>**	Apply ACL to the UTA communication interface
7	Example: **C2951 (config)# ip access-group 101 in**	
8	Example: **C2951 (config-if)#end**	Ends the configuration mode
9	**Save config** Example:	Saves the configuration

	C2951 #wr	

6.6 Configuring Appliance ETM® software

Current ETM® Appliance software is designed to be initially configured via its own CLI prior to entering an operational state. Step through the ETM configuration wizard. After it is completed, configure your ETM® Appliance software UTA provider Addresses and Ports as you specified in **Section 6.3.1**in the Management Server. Please refer to the ETM System user guides for further information.

```
[root@ETM ~]$cd opt/slc
[root@ETM ~]$./ETM_5000_configure.pl
```

6.6.1 Confirm UTA Connection

When the ETM_5000_configure.pl script has completed you will be able to observe the VM appliance connecting to the management server. When the appliance has connected if you are using the UTA RPM's, you will have to edit the span and ensure the API value in the GUI is the value of the IP address that is the router to hypervisor IP address configured in Section 3.2.1 Step 4 of the "ETM Application on Cisco Services-Ready Engine Virtualization Installation and Configuration Guide" document.

You can determine if the appliance is connected to the router by performing the "show wsapi reg all" which will tell you if the UTA is connected.

```
C2951# show wsapi reg all
Provider XCC
=====================================================
registration
id: 8B78788:XCC:XCCapp10:5
appUrl:http://10.1.1.25:8888
appName: XCCapp10
provUrl: http://10.1.1.100:8090/cisco_xcc
prober state: STEADY
connEventsFilter:
CREATED|AUTHORIZE_CALL|ADDRESS_ANALYZE|REDIRECTED|ALERTING|CONNEC
TED|TRANSFERRED|CALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_L
EAVE
mediaEventsFilter: MODE_CHANGE
blockingEventTimeoutSec: 5
blockingTimeoutHandle: CONTINUE_PROCESSING


Provider XSVC
=====================================================
registration index: 1
id: 8B78784:XSVC:XSVCapp9:5
appUrl:http://10.1.1.25:8889
appName: XSVCapp9
provUrl: http://10.1.1.100:8090/cisco_xsvc
```

```
prober state: STEADY
route filter:
event filter: off
```

7. Debugging and Diagnostics

7.1 Ensuring ETM® application to UTA provider connectivity

After you have configured the appliance using the **ETM_5000_configure.pl** script, you should be able to observe the appliance connecting to the management server and router sending messages to the service module. Remember you will have to edit the UTA SPAN API tab for the SM on the management server to ensure the router's IP that was configured in Section 4.3, 5.3 or 6.3.1 Step 3 is entered.

You can also confirm the successful connectivity state of the UTA provider and ETM applications by using the following command on the router:

```
C2951# show wsapi reg all
Provider XCC
=====================================================
registration
id: 8B78788:XCC:XCCapp10:5
appUrl:http://10.1.1.20:8888
appName: XCCapp10
provUrl: http://10.1.2.254:8090/cisco_xcc
prober state: STEADY
connEventsFilter:
CREATED|AUTHORIZE_CALL|ADDRESS_ANALYZE|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|C
ALL_DELIVERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
mediaEventsFilter: MODE_CHANGE
blockingEventTimeoutSec: 5
blockingTimeoutHandle: CONTINUE_PROCESSING


Provider XSVC
=====================================================
registration index: 1
id: 8B78784:XSVC:XSVCapp9:5
appUrl:http://10.1.1.20:8889
appName: XSVCapp9
provUrl: http://10.1.2.254:8090/cisco_xsvc
prober state: STEADY
route filter:
event filter: off
```

If neither of the above is present then there is likely a transport or connectivity issue between the UTA provider and the ETM application. Settings to check and ensure are:

- Ensure that the “Appliance IP” and “Router IP” configuration are correct and deployed to the ETM application.
- Refer to section *Authorizing ICMP or SSH to external Network Management Devices* to enable SSH and PING to the ETM application from a remote party. Can you ping the ETM application “Appliance IP” as defined in the API tab from the router? If you cannot then check the routing and network settings on both the router and application to ensure they cannot communicate.
- Refer to section *Authorizing ICMP or SSH to external Network Management Devices* to enable SSH and PING to the ETM application from a remote party. Can you ping the UTA provider “Router IP” as defined in the API tab from the ETM application? If you cannot then check the routing and network settings on both the router and application to ensure they cannot communicate.
- Ensure that the routing tables on the UTA provider and UTA application direct HTTP TCP traffic over the interface which is equivalent to the “Router IP” setting.

7.2 Debugging support

If you are still unable to reach a STEADY state and all of the above are OK, then it is advised to solicit help from ETM support. In order to facilitate the interaction produce a tech-support package that can be sent to support. Assuming you have SSH access to the ETM application:

```
[root@ETM ~]$cd opt/slc
[root@ETM ~]$[sudo] ./techsupport.sh
```

This will create a zipped file called **techsupport.tar.gz.** This file contains important debugging information including network captures of the traffic leaving and entering the system where the ETM application is running. Send this package to support.

This package can be collected at any time; however we recommend its collection be limited to periods of inactivity in the ETM application as to avoid potential degradation of application performance.

Moreover, if any other issues are detected with the application, this package is the primary means of communicating field debugging information to support.

Addendum

Cisco IOS 15.1(2)+ Toll Fraud Features

Behavior Before 15.1(2)T

For all IOS releases before 15.1(2)T, the default behavior for IOS voice gateways is to accept call setups from all sources. As long as voice services are running on the router, the default configuration will treat a call setup from any source IP address as a legitimate and trusted source to set a call up for. Also, FXO ports and inbound calls on ISDN circuits will present secondary-dial tone for inbound calls, allowing for two-stage dialing. This assumes a proper inbound dial-peer is being matched.

Behavior with 15.1(2)T and Later Releases

Starting with 15.1(2)T, the router's default behavior is to not trust a call setup from a VoIP source. This feature adds an internal application named TOLLFRAUD_APP to the default call control stack, which checks the source IP of the call setup before routing the call. If the source IP does not match an explicit entry in the configuration as a trusted VoIP source, the call is rejected.

For further explanation see:
http://www.cisco.com/en/US/tech/tk652/tk90/technologies_tech_note09186a0080b3e123.shtml

SIP URI Processing versus E164 on UTA Provider

To enable the Provider to send SIP URI's as address information the ETM UTA application you must ensure that you enable the following feature on the Provider. Given that Cisco CUBE CANNOT yet Route calls based on a destination that is a SIP URI, this feature is only to enable SIP URI address processing in the ETM UTA application.

```
C2951#config t                                  <enter config mode
C2951(config)# voice service voip
C2951(config-voi-serv)#sip
C2951(conf-serv-sip)#header-passing
C2951(conf-serv-sip)#end
C2951(config)# wr                               < save config
```

Trunk Serviceability and Monitoring

To fully realize Trunk monitoring and serviceability the Provider Trunk configuration must be configured properly to take advantage of the full features offered by the Provider to satisfy monitoring of the Trunk status. More information can be found at
http://www.cisco.com/en/US/prod/collateral/voicesw/ps6790/gatecont/ps5640/white_paper_c11-613550_ps10536_Products_White_Paper.html

ETM UTA TDM Trunk Status and D-Channel

The UTA application and UTA Provider allow TDM trunks to be monitored for Alarms and Health of the Trunk and D-Channel. This functionality is similar to the native ETM TDM appliances, but has fewer

capabilities. It is important to understand these capabilities in order to set expectations as it relates to Trunk Health and Status in the Management Server and ETM System Events.

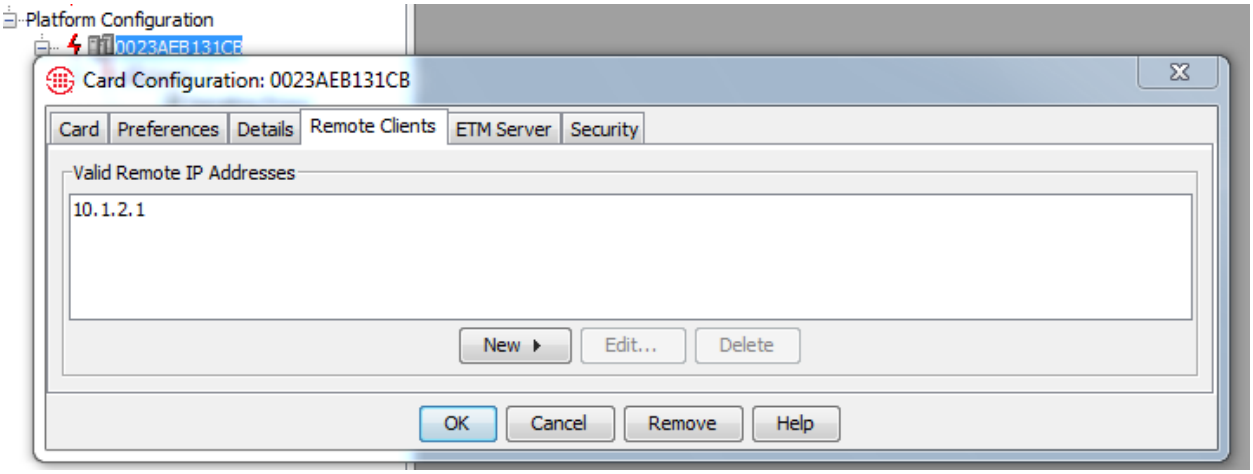
Below is a matrix that captures the environmental state and maps it to expected output in the ETM applications.

Trunk Status from LAST Heartbeat from UTA Application	D-Channel Status from LAST Heartbeat from UTA Application	Trunk Status as Reported by the UTA Provider via XSVC	Trunk Status reflected in the Management Server/System Events	D-Channel Status reflected in the Management Server
Unknown	Unknown	UP	UP (but don't send Telco)	UP (but don't send Telco)
Unknown	Unknown	DOWN	UP (but don't send Telco)	DOWN
Unknown	Unknown	DOWN	DOWN	DOWN
UP	UP	UP	Already UP	Already UP
UP	UP	DOWN	Already UP	DOWN
UP	UP	DOWN	DOWN	DOWN
UP	DOWN	UP	Already UP	UP
UP	DOWN	DOWN	Already UP	Already DOWN
UP	DOWN	DOWN	DOWN	Already DOWN
DOWN	DOWN	UP	UP	UP
DOWN	DOWN	DOWN	UP	Already DOWN
DOWN	DOWN	DOWN	Already DOWN	Already DOWN

NOTE: The cases marked in RED are not detectable and supported due to limitations with the UTA Provider event generation strategy.

Authorizing ICMP or SSH to external Network Management Devices

The ETM Linux OS that is configured locks down network access to authorized Endpoints and the ETM® Management Server. To enable additional parties to PING the host or enable SSH access, enable Remote Clients in the Management Server interface.



AXP Boot Loader config

The AXP Boot-loader guide can be found at

http://www.cisco.com/en/US/docs/interfaces_modules/services_modules/ax/1.6/user/guide/axpap1.html#wpxref86381

AXP Boot Loader bypass on SRE only

To install a base image instead of going through the boot helper mode you can use the boot loader to load the AXP Operating system via the link above or use the standard IOS command:

Router#**service-module sm 0/0 install url ftp://x.x.x.x/ axp-k9.sme.1.6.1.pkg**