Configuration Guide

AudioCodes Mediant™ Multi-Service Business Routers (MSBR) Series

Security Setup

Version 7.2

Table of Contents

This page is intentionally left blank.

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: March 13, 2024

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Stay in the Loop with AudioCodes

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Throughout this manual, unless otherwise specified, the term *device* refers to AudioCodes Mediant MSBR products.

Related Documentation

Document Name
Mediant 500Li MSBR Hardware Installation Manual
Mediant 500Li MSBR User's Manual
Mediant 500L MSBR Hardware Installation Manual
Mediant 500L MSBR User's Manual
Mediant 500 MSBR Hardware Installation Manual
Mediant 500 MSBR User's Manual
Mediant 800 MSBR Hardware Installation Manual
Mediant 800 MSBR User's Manual

Document Revision Record

LTRT	Description
31641	Initial document release for Ver. 7.2
31643	Removal of the auto-VPN section; added new section for DNS query randomization
31646	Updates for IPSec configuration
31647	Management Access List added (domain name support)
31648	Updates to Section Configuring IPSec with GRE
31820	Security updated with SHA-256; L2TP updated with enabling NAT traversal
31821	Configuring IPSec with RSA added
31822	Typo in Section Configuring Port Forwarding
31823	New command (config-isakmp)# ike
31825	Typos (incorrect IP addresses) in Configuring IPSec section
31826	Command added ip tcp adjust-mss (Mediant 500Li MSBR)
31827	IPSec with VTI removed
31828	IKEv2 added
31829	Multiple access lists per IPSec tunnel.
31830	IPSec updated
31831	16 ACS rules per IPSec tunnel; 802.1x as a client (supplicant)

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our website at https://online.audiocodes.com/documentation-feedback.

.

1 Introduction

This document describes configuration of the security functionality of AudioCodes Mediant Multi-Service Business Routers (MSBR), hereafter referred to as *device*, using the command-line interface (CLI).

The document describes the CLI commands required for configuring each aspect of security, providing typical configuration examples for some of the features.

This page is intentionally left blank.

2 Access Control List

The device supports access control lists (ACL). The ACLs are tools to categorize traffic based on source IP or/and destination IP, protocols or ports used by traffic. The categorization is done by matching traffic to rules defined in the ACL. The ACLs usually work in combination with other features such as QoS, Firewall, IPSec and NAT. The ACLs are used to select which traffic to apply to which feature. The device supports two types of ACLs – connectionless and connection-aware or stateful. Connection-aware access lists only match first packets based on a rule, for example, traffic from source to destination. Subsequent packets with the same rule are categorized without matching. This saves CPU and memory resources. The ACLs can only be configured on Layer-3 interfaces.

To configure ACLs, use the following commands:

Table 2-1: Access Control List

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# access-list [number or word] [deny or permit] <protocol> <source> <source port> <destination> <destination port> <mode> [log]`	▪ [number or word] – ACL can be addressed using a number or a word. **Note:** access-list names are case sensitive. ▪ [deny or permit] – connection using this rule is denied or permitted using this keyword. ▪ <protocol> - connection is matched using one of the protocols: tcp, udp, ah, esp, gre, icmp, igmp, ip or manually selected using a number, 0 to 255, that represents the protocol field of the IP packet. ▪ <source> - source can be selected as a single host IP address, range of IP addresses with mask or local address. It also can be "any" address. Range of IP addresses need to be selected using wildcard. ▪ <source port> - source can be matched using TCP or UDP port. The <source port> can be omitted. ▪ <destination> - destination can be selected as a single host IP address, range of IP addresses with mask or local address. It also can be "any" address. Range of IP addresses needs to be selected using a wildcard. ▪ <destination port> - destination can be matched using TCP or UDP port. The <destination port> can be omitted. ▪ <mode> - mode of the ACL. If the keyword "established" is used, the ACL will be connection aware. If the keyword "stateless" is used, the ACL will be connectionless. The keyword "dscp" can be used to match the DSCP field of the IP packet. By default, the ACL will be connection aware. The <mode> can be omitted. ▪ [LOG] – if the log keyword is used, if a packet matches the rule, the event is logged

Command	Description
	and a counter will increment in the `show` command.
`(config-data)# ip access-list [extended or standard] [Name or number]`	Alternative method to configure ACLs is by using the `ip access-list` command. This accesses the ACL with the [name or number] configuration level. In the configuration level, the commands start with deny or permit as if the `access-list` command is used instead of `ip access-list`.
`# sh data access-lists`	Displays configured ACLs.
`(config-data)# no access-list <Name>`	Deletes the ACL with the name <Name>.

From Version 6.8, ACL numbering is supported. Every line in the ACL has a number. Every next line number is incremented by 10. To add a line between line number 10 and 20, start the ACL command with a number, as shown in the example table below:

Table 2-2: ACL Commands Example

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# ip access-list [extended or standard] [Name or number]`	Enter the ACL configuration level.
`(config-ext-nacl)# 15 permit ip <source> <destination>`	Add line number 15.
`(config-data)# ip access-list resequence <ACL No. or name> <start line> <step>`	Allows the sequencing of the line numbers of the ACL. <start line>: starting line number of the ACL. <step>: jump in numbers from line to line.

2.1 Configuration Example

This example configures an ACL rule called "DC-Access" that allows traffic from any source to a specific class C subnet:

```
# configure data
(config-data)# access-list DC-Access permit ip any 192.168.100.0
0.0.0.255 log

(config-data)# access-list DC-Access permit ip any 192.168.110.0
0.0.0.255 log

(config-data)# access-list DC-Access permit ip any 192.168.120.0
0.0.0.255 log

(config-data)# access-list DC-Access deny ip any any log
# show data access-lists
Extended IP access list DC-Access
DC-Access permit ip any 192.168.100.0 0.0.0.255 log   (0 matches)
DC-Access permit ip any 192.168.110.0 0.0.0.255 log   (0 matches)
DC-Access permit ip any 192.168.120.0 0.0.0.255 log   (0 matches)
DC-Access deny ip any any log   (0 matches)
#
```

The following example allows access from any IP to segment 192.168.199.0/24 only for SSH (TCP port 22), Telnet (TCP port 23), SNMP (UDP port 162) and UDP port 2032. For everything else, traffic is denied.

```
(config-data)# access-list DC-Access permit tcp any 192.168.199.0
0.0.0.255 eq 22 log

(config-data)# access-list DC-Access permit tcp any 192.168.199.0
0.0.0.255 eq 23 log

(config-data)# access-list DC-Access permit udp any 192.168.199.0
0.0.0.255 eq 162 stateless log

(config-data)# access-list DC-Access permit udp any 192.168.199.0
0.0.0.255 eq 2032 stateless log

(config-data)# access-list DC-Access deny ip any any

(config-data)#
```

The following example configures an ACL using the `ip access-list` command:

```
(config-data)# ip access-list extended DC-Access
(config-ext-nacl)# permit ip any 192.168.10.0 0.0.0.255 log
(config-ext-nacl)# deny ip any any log
(config-ext-nacl)#
```

This page is intentionally left blank.

3 ACLv6

The device supports ACL for the IPv6 protocol. Configuration rules are the same as for IPv4.

Table 3-1: ACLv6 Commands

Command	Description
`# configure data`	Configuration of ACLs is in the data level.
`(config-data)# ipv6 access-list [extended or standard] [Name or number]`	Accesses the ACL with the [name or number] configuration level.
`(config-data)# [line number] [deny or permit] <protocol> <source> <source port> <destination> <destination port> <mode> [log]`	▪ [line number]: Every line starts with a line number. This defines the number of this line. (from Version 6.8). ▪ [deny or permit]: connection using this rule is denied or permitted using. ▪ <protocol>: connection is matched using one of the protocols: tcp, udp, ah, esp, gre, icmp, igmp, ip or manually selected using a number, 0 to 255, that represents the protocol field of the IP packet. ▪ <source>: selects the source. The source can be selected as a single host IP address, range of IP addresses with mask or local address. It also can be "any" address. Range of IP addresses can be defined using a wildcard. ▪ <source port>: source can be matched using TCP or UDP port. The <source port> can be omitted. ▪ <destination>: selects the destination. The destination can be selected as a single host IP address, range of IP addresses with mask or local address. It also can be "any" address. Range of IP addresses can be defined using a wildcard. ▪ <destination port>: destination can be matched using TCP or UDP port. The <destination port> can be omitted. ▪ <mode>: the mode of the ACL. If the keyword "established" is used, the ACL is connection aware. If the keyword "stateless" is used, the ACL is connectionless. The keyword "dscp" can be used to match the DSCP field of the IP packet. By default, the ACL is connection aware. The <mode> can be omitted. ▪ [LOG]: if the log keyword is used, if a packet matches the rule, the event is logged and a counter will increment in the `show` command.
`# sh data access-lists`	Displays configured ACLs.
`(config-data)# no access-list <Name>`	Deletes the ACL with the name <Name>.

3.1 Configuration Example

This example configures an IPv6 ACL rule. Configuration is applied at firewall index for line 10, 20, and then 15.

```
# configure data
(config-data)# ipv6 access-list extended 150

(config-ext6-nacl)# 10 permit ipv6 2000:100:1::0/64
2000:100:2::0/64 log
(config-ext6-nacl)# 20 permit ipv6 2000:102:1::0/64
2000:100:2::0/64 log

(config-ext6-nacl)# 15 permit ipv6 2000:101:1::0/64
2000:100:2::0/64 log
(config-ext6-nacl)# exit

(config-data)# exit
#
```

You can view the configured ACL using the following command:

```
(config-data)#
# show data access-lists
Extended IP access list 150
150  10 permit ipv6 2000:100:1::0/64 2000:100:2::0/64 log    (0
matches)
150  15 permit ipv6 2000:101:1::0/64 2000:100:2::0/64 log    (0
matches)
150  20 permit ipv6 2000:102:1::0/64 2000:100:2::0/64 log    (0
matches)
```

You can add lines to the end of the ACL:

```
# configure data

(config-data)#

(config-data)# ipv access-list extended 150
(config-ext6-nacl)# 999 deny ip any any

(config-ext6-nacl)# exit
```

The ACL can be organized using the `resequence` command:

```
(config-data)# ipv6 access-list resequence 150 10 10
```

The result can be shown using the `show data access-lists` command:

```
(config-data)# exit
# show data access-lists
Extended IP access list 150
150  10 permit ipv6 2000:100:1::0/64 2000:100:2::0/64 log   (0
matches)
150  20 permit ipv6 2000:101:1::0/64 2000:100:2::0/64 log   (0
matches)
150  30 permit ipv6 2000:102:1::0/64 2000:100:2::0/64 log   (0
matches)
150  40 deny ipv6 any any   (0 matches)
```

This page is intentionally left blank.

4 Management Access Lists

When an access list is created for management using the protocols SNMP, Telnet, SSH or CWMP, it is possible to use DNS names instead of IP or IPv6 addresses. The device resolves the name to an IP address and acts upon the ACL rules. If the DNS resolution fails within one second, the device denies this connection.

4.1 Configuration Example

This example shows how to use access lists to permit or deny DNS hostnames through a WAN interface. In the example, the Telnet connection configured in the access list has the hostname “telnet_mgmt” (Telnet management workstation). This host permits access to “mgmt_ws” (any management IP address of the device).

```
configure data
   access-list telnet_mgmt permit ip host mgmt_ws local log
   access-list telnet_mgmt deny ip any any log
```

Configure the ACL for the Telnet connection:

```
configure system
   cli-terminal
    wan-telnet-allow on
    set telnet-acl "telnet_mgmt"
    activate
   exit
```

In the example below, the DNS name resolves locally on the device using the following command:

```
ip host mgmt_ws 10.1.1.44 3600
```

In other environments, an external DNS server can be used. To configure an external DNS, use the following command:

```
ip name-server <DNS Server IP address>
```

To verify the ACL, run two Telnet commands, once from mgmt_ws and once from a different location. Use the command `show data access-lists`. The counter should be incremented once for the mgmt_ws interface and once for the telnet_mgmt interface.

```
# sh d access-lists
Extended IP access list telnet_mgmt
 telnet_mgmt  10 permit ip host mgmt_ws local log    (1 matches)
 telnet_mgmt  20 deny ip any any log    (1 matches)
```

This page is intentionally left blank.

5 NAT and NAPT

The device supports the NAT and PAT protocols. The PAT protocol for the device is addressed as Network Address and Port Translation (NAPT). NAT changes the inside address of your network with an external address. NAPT changes the inside addresses of your network with a single external address with several ports.

NAT and NAPT provide two major benefits:

- The inside of a network behind NAT or NAPT is hidden and cannot be accessed from outside networks.
- Saves IP addresses on the Internet, by using one address toward the outside and many addresses on the inside.

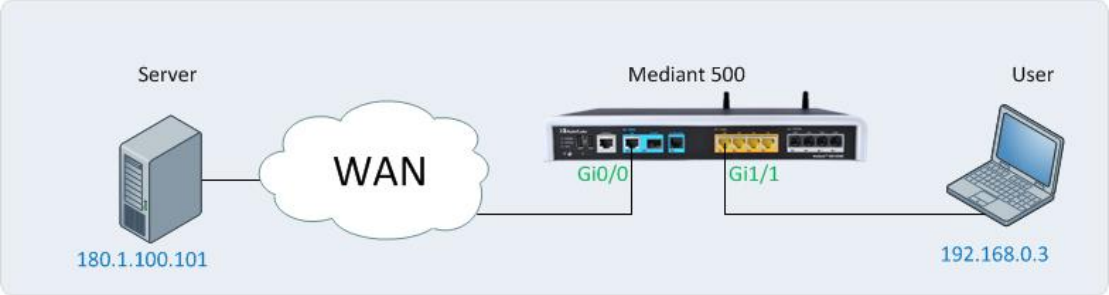
By default, NAPT is activated on the Gigabitethernet0/0 interface. To disable NAPT per interface, use the following commands:

Table 5-1: NAT and NAPT Commands

Command	Description
`# configure data`	Configuration of ACLs is in the data level.
`(config-data)# interface gigabitethernet 0/0`	Configure interface gigabitethernet0/0.
`(conf-if-GE 0/0)# no napt`	Disable NAPT on the interface.

After disabling NAPT on the interface, the interface becomes a routing interface and packets from the inside IP addresses are forwarded using the routing table through the interface gigabitethernet0/0.

Figure 5-1: NAPT and NAT Topology



In Figure 5-1: NAPT and NAT Topology, when NAPT is disabled, in every packet sent to the server from the user, the source is the user's IP address. When NAPT is enabled, the source IP of every packet is the IP address configured on the WAN interface. The WAN interface in the example is port Gi0/0.

Both NAT and NAPT can use a pool of addresses to contact (or to show) the outside word (the WAN). For NAT and NAPT, a range of IP addresses and ports can be configured using ACLs. This range of IP addresses is called a *NAT pool*. To configure the NAT pool, use the following commands.

Table 5-2: NAT Pool Commands

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# access-list tcp_nat permit tcp 192.168.0.0 0.0.0.255 any`	Mark the traffic of the inside addresses. These addresses will be hidden behind NAT.
`(config-data)# ip nat pool tcp_pool 180.1.100.50 180.1.100.50`	Configure a NAT pool that starts with the address 180.1.100.50 and ends with the address 180.1.100.50. This means that there is only one address in the NAT pool.

Table 5-3: NAT Rules

Command	Description
`(config-data)# ip nat inside source list tcp_nat interface gigabitethernet 0/0 pool tcp_pool`	Configure IP NAT translation for devices behind NAT. For every address?? selected by the tcp_nat ACL, on the interface gi0/0 and use the tcp_pool NAT pool.

Table 5-4: NAPT Rules

Command	Description
`(config-data)# ip nat inside source list tcp_nat interface gigabitethernet 0/0 pool tcp_pool port 5000 5010`	Configure IP NAPT translation for IP addresses behind the NAT. For every address selected by the tcp_nat ACL, on the interface gi0/0, map multiple IP addresses to the tcp_pool addresses using ports range 5000-5010.

The process of changing the LAN IP address to WAN IP address is called *NAT translation*. To verify that the NAT translation is working, use the following command:

Table 5-5: NAT Translation

Command	Description
`# show data ip nat translations`	Displays NAT translations.

To access a specific port on an IP address on the inside network while using NAT, configure port forwarding using the following configuration steps:

Table 5-6: NAT Port Forwarding Configuration

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# ip nat inside source static <protocol> <inside IP address> <inside port> <outside interface> <outside port>`	Configures NAT port forwarding. ▪ <protocol>: protocols (gre, ip, tcp, udp). ▪ <inside IP address>: IP address of the device on the inside. ▪ <inside port>: port on the inside. ▪ <outside interface>: physical interface to witch the outside world is connected to. ▪ <outside port>: port to which the users from the outside connect to.

The device supports load balancing using NAT. If there are more than two servers on the LAN side of the device, a connection to the WAN address can be forwarded to one of the servers in a round-robin fashion. To configure load balancing, see the following table:

Table 5-7: Configuring NAT Load Balancing

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# ip nat pool <pool name> <start address> <end address> rotary`	Configure the NAT pool. ▪ <pool name>: NAT pool name. The <start address> is the first IP to load balance connections to. ▪ <end address>: last IP to load balance connections to. ▪ rotary: activates the load balance feature
`(config-data)# ip nat inside destination <WAN IP> port <port> pool <pool name>`	▪ <WAN IP>: outside address accessible from the WAN side of the device. ▪ <port>: port on the WAN side to which the users connect. The same port is used to access the servers on the inside. ▪ <pool name>: NAT pool name configured using the `ip nat pool` command.

5.1 Configuration Examples

5.1.1 Configuring TCP and ICMP NAT

This example configures a NAT for TCP and ICMP traffic. UDP traffic will not use NAT.

```
# configure data
(config-data)# access-list gen_nat permit tcp 192.168.0.0
0.0.0.255 any
# gen_nat is a short for general NAT

(config-data)# access-list gen_nat permit icmp 192.168.0.0
0.0.0.255 any log
(config-data)# ip nat pool nat_pool 180.1.100.50 180.1.100.50
(config-data)#  ip nat inside source list gen_nat interface
GigabitEthernet 0/0 pool nat_pool
```

This example configures a NAPT for TCP only:

```
# configure data
(config-data)# access-list gen_nat permit tcp 192.168.0.0
0.0.0.255 any
# gen_napt is a short for general NAPT
(config-data)# ip nat pool nat_pool 180.1.100.50 180.1.100.50
(config-data)#  ip nat inside source list gen_nat interface
GigabitEthernet 0/0 pool nat_pool port 4000 5000
```

Below is the output of the `show data ip nat translations` command:

```
# show data ip nat translations
(Note: static translations are not shown)
NAT summary: 1 TCP, 0 UDP, 2 ICMP. Total 3 NAT connections.
.Pro Inside global         Inside local          Outside local
Outside global        Timeout
ICMP180.1.100.50 512     192.168.0.3 512      180.1.100.100
180.1.100.100         0
ICMP180.1.100.50 512     192.168.0.3 512      180.1.100.101
180.1.100.101         0
TCP 180.1.100.50:2046    192.168.0.3:2046     180.1.100.100:80
180.1.100.100:80      7199
```

The output displays only TCP and ICMP sessions that have been translated. The output does not display UDP sessions because the UDP traffic is not included in the gen_nat access list.

5.1.2 Configuring Port Forwarding

This example configures port forwarding to forward port 2080 to port 80, from the WAN side to the LAN side:

```
# configure data

(config-data)# ip nat inside source static tcp 192.168.0.200 80
GigabitEthernet 0/0 2080
```

The IP address of the interface gigabitEthernet 0/0 is 180.1.1.1. Every connection made to IP address 180.1.1.1 on port 2080 is forwarded to IP address 192.168.0.200 on port 80.

5.1.3 Configuring Load Balancing using NAT

This example includes two HTTP servers on the NAT side. One with IP address 192.168.0.3 and one with IP address 192.168.0.4. Both are identical HTTP server with main page. To access these servers, a secondary IP address of the WAN interface GigabitEthernet 0/0 is configured. The main IP address of the WAN interface is 180.1.100.1 and the secondary is 180.1.100.10.

```
# configure data
(config-data)# interface gigabitethernet 0/0
(conf-if-GE 0/0)# ip address 180.1.100.1 255.255.255.0
(conf-if-GE 0/0)# ip address 180.1.100.10 255.255.255.0 secondary
(conf-if-GE 0/0)# exit
(config-data)# ip nat pool L-balancing 192.168.0.3 192.168.0.4
rotary
(config-data)# ip nat inside destination 180.1.100.10 port 80 pool
L-balancing
(config-data)#
```

The output of the `show data ip nat translations` command displays a source address 180.1.100.20 from port 4355 that accesses IP address 180.1.100.10 on port 80. The connection is then NATed to the inside address of 192.168.0.3:80.

```
# show data ip nat translations
(Note: static translations are not shown)
NAT summary: 1 TCP, 0 UDP, 0 ICMP. Total 1 NAT connections.
.Pro Inside global        Inside local         Outside local
Outside global       Timeout
TCP 180.1.100.10:80       192.168.0.3:80      180.1.100.20:4355
180.1.100.20:4355    86395
```

After waiting a while, a refresh command is issued at the source and the source accesses the external IP address again. Now the output of the `show data ip nat translations` command displays that the other HTTP server with the IP address 192.168.0.4 was accessed:

```
# show data ip nat translations
(Note: static translations are not shown)
NAT summary: 1 TCP, 0 UDP, 0 ICMP. Total 1 NAT connections.
.Pro Inside global        Inside local         Outside local
Outside global       Timeout
TCP 180.1.100.10:80       192.168.0.4:80      180.1.100.20:4356
180.1.100.20:4356    86397
```

This page is intentionally left blank.

6 SPI Firewall

The device provides a built-in firewall feature. The firewall allows or denies traffic using a rule set. The firewall rules are set using ACLs. The firewall can be session-aware or stateless. There are two modes of firewall: manual and automatic. To configure the firewall in automatic mode, use the following commands:

Table 6-1: Firewall - Automatic Mode

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# interface gigabitethernet 0/0`	Enter the interface.
`(conf-if-GE 0/0)# firewall enable`	Enables the firewall.
`(conf-if-GE 0/0)# no firewall enable`	Disables firewall.

An automatic firewall performs a stateful packet inspection and keeps track of the state of each connection and is able to drop inbound protocol data units if they do not belong to a known connection. For example, if a user initiates an HTTP request to a sever on the WAN (anything connected to the WAN interface), the device allows that server to respond to the user.

To configure a manual firewall, use ACLs and apply the ACL rules on an interface IN or OUT direction. The firewall can only be configured on Layer-3 interfaces.

Table 6-2: Firewall – Manual Configuration

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# interface gigabitethernet 0/0`	Enter the interface.
`(conf-if-GE 0/0)# ip access-group name {in|out}`	Apply an access-list to the interface (inbound or outbound).
`(conf-if-GE 0/0)# no ip access-group name {in|out}`	Remove an access-list to the interface (inbound or outbound).

To view whether the firewall "caught" packets, use the following command:

Table 6-3: Firewall –Verification

Command	Description
`# show data access-lists`	Displays all access lists and packets that have been caught.
`# show data ip access-list FW_out`	Displays specific ACL and packets caught.

Note that when a firewall is enabled, all inbound traffic is denied access; however, the user can still explicitly permit only ICMP inbound traffic.

Table 6-4: Firewall – Permit ICMP Inbound Traffic

Command	Description
(config-data)# ip firewall allow-icmp	Allow ICMP (ping) on interfaces without an access-list.

6.1 Configuration Example

This example configures a firewall on the G0/0 interface to allow traffic on TCP ports 20 to 23 and UDP ports 5000-5004 at the destination, from the 192.168.0.0/24 to any network. The firewall also allows ping from and to any host. The firewall ends with a deny any any rule, which blocks all other traffic.

```
# configure data
; Create the ACL
(config-data)# ip access-list extended FW_out
(config-ext-nacl)# permit tcp 192.168.0.0 0.0.0.255 any eq 20 log
(config-ext-nacl)# permit tcp 192.168.0.0 0.0.0.255 any eq 21 log
(config-ext-nacl)# permit tcp 192.168.0.0 0.0.0.255 any eq 22 log
(config-ext-nacl)# permit tcp 192.168.0.0 0.0.0.255 any eq 23 log
(config-ext-nacl)# permit udp 192.168.0.0 0.0.0.255 any eq 5000
log
(config-ext-nacl)# permit udp 192.168.0.0 0.0.0.255 any eq 5001
log
(config-ext-nacl)# permit udp 192.168.0.0 0.0.0.255 any eq 5002
log
(config-ext-nacl)# permit udp 192.168.0.0 0.0.0.255 any eq 5003
log
(config-ext-nacl)# permit udp 192.168.0.0 0.0.0.255 any eq 5004
log
(config-ext-nacl)# permit icmp any any log
(config-ext-nacl)# deny ip any any log
(config-ext-nacl)#

@ Apply ACL on an interface
(config-ext-nacl)# exit
(config-data)# interface gigabitethernet 0/0
(conf-if-GE 0/0)# ip access-group FW_out out
```

After simulating the ICMP, UDP traffic on port 5000 and traffic on other ports that are not allowed by the firewall, the output of the `show data access` command displays the following:

```
# show data access-lists
Extended IP access list FW_out
FW_out permit tcp 192.168.0.0 0.0.0.255 any eq 20 log   (0
matches)
FW_out permit tcp 192.168.0.0 0.0.0.255 any eq 21 log   (0
matches)
FW_out permit tcp 192.168.0.0 0.0.0.255 any eq 22 log   (0
matches)
FW_out permit tcp 192.168.0.0 0.0.0.255 any eq 23 log   (0
matches)
FW_out permit udp 192.168.0.0 0.0.0.255 any eq 5000 log   (2
matches)
FW_out permit udp 192.168.0.0 0.0.0.255 any eq 5001 log   (0
matches)
FW_out permit udp 192.168.0.0 0.0.0.255 any eq 5002 log   (0
matches)
FW_out permit udp 192.168.0.0 0.0.0.255 any eq 5003 log   (0
matches)
FW_out permit udp 192.168.0.0 0.0.0.255 any eq 5004 log   (0
matches)
FW_out permit icmp any any log   (1298 matches)
FW_out deny ip any any log   (701523 matches)

#
```

Note that the traffic counter incremented after specific traffic passed through the ACL.

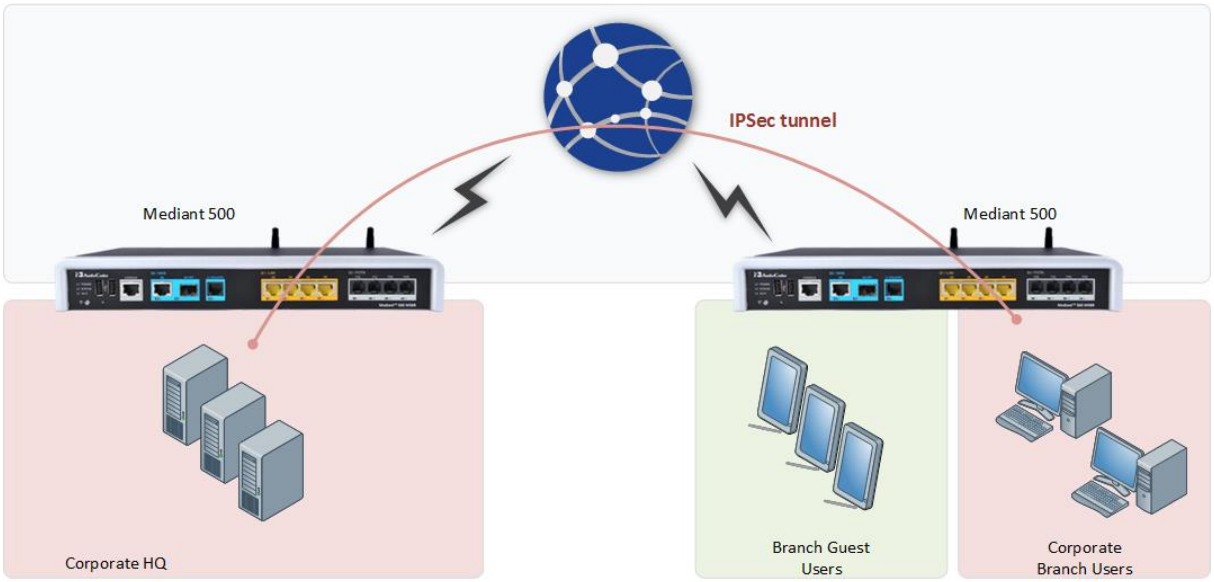
This page is intentionally left blank.

7 IPSec Tunneling

The device supports the IPSec tunnel protocol. IPSec tunnels encrypt sessions between two points. These points could be single computers, network segment or selected hosts. The IPSec encryption uses the AES, 3DES or DES algorithms.

There are many practical uses for encrypting data. For example, if some corporation would like to provide guest access to the internet for the corporation guests, but also the corporation would like to protect itself from corporate espionage, it is a good practice to use IPSec.

Figure 7-1: IPSec and Guest Access



In the example above, the Corporate Branch Users are connected through the IPSec tunnel to the Corporate HQ. The communication is encrypted using IPSec, and the Guest Users, or anyone on the Internet are not able to "read" and understand the traffic between the segments. This solution is also applicable to other applications that need to encrypt traffic such as protecting classified project in the same organization.

To configure IPSec, use the following commands:

Table 7-1: IPsec Tunneling

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# access-list ipsec permit ip 192.168.0.0 0.0.0.255 10.0.0.0 0.0.0.255`	Create an ACL to capture traffic for IPSec. This will later become an entry in the routing table. Up to 16 rules can be used per IPSec tunnel, enabling multiple subnets to "reside" behind an IPSec tunnel.
`(config-data)# crypto isakmp policy 1`	Configure the isakmp policy.
`(config-isakmp)# encryption aes 128`	Configure the encryption protocol (AES, DES or 3DES). The number is the amount of bits for the encryption protocol.
`(config-isakmp)# authentication pre-share`	Choose an authentication method (pre-shared key or Rivest-Shamir-Adleman Signature).

Command	Description
`(config-isakmp)# hash sha`	Configures the hashing protocol (sha, sha256, or md5). The sha protocol is stronger than md5.
`(config-isakmp)# group 2`	Configures the Diffie-Hellman group.
`(config-isakmp)# ike v1`	Selects IKE version 1 or IKE version 2
`(config-isakmp)# lifetime 28800`	The lifetime is the period of re-authentication. In this case, the tunnel is re-authenticated every hour.
`(config-isakmp)# exit`	Exit policy configuration level.
`(config-data)# crypto ipsec transform-set crypto_set1 esp-aes 128 esp-sha-hmac`	Configure the transform set, and select encrypting type and key length in bits.
`(cfg-crypto-trans)# mode tunnel`	Select the operation mode.
`(cfg-crypto-trans)# exit`	Exit transform set configuration level.
`(config-data)# crypto map MAP1 1 ipsec-isakmp`	Configure the crypto map.
`(config-crypto-map)# set peer 180.1.100.21`	Configure the peer IP address.
`(config-crypto-map)# set transform-set crypto_set1`	Configure the transform set.
`(config-crypto-map)# set security-association lifetime seconds 3600`	Configure the lifetime timer. When the timer expires, re authentication commences.
`(config-crypto-map)# match address ipsec`	Assign an ACL to the transform set.
`(config-crypto-map)# exit`	Exit the transform set configuration level.
`(config-data)# crypto isakmp key P@ssw0rd address 180.1.100.21`	Configure the key from the IPSec.
`(config-data)# interface GigabitEthernet 0/0`	Configure interface g0/0.
`(conf-if-GE 0/0)# crypto map MAP1`	Assign the IPSec policy to the interface.
`(conf-if-GE 0/0)# ip tcp adjust-mss 1374`	Ensures that IPSec traffic is accelerated, resulting in high performance of the IPSec traffic. **Note:** This is applicable only to Mediant 500Li MSBR.
`# show data crypto status`	Displays the IPSec status.

IPSec is supported only on the main VRF.

Multiple crypto maps configuration to a single peer is not supported. To encrypt several LAN networks to a single destination, use several lines in the access list under a crypto map.

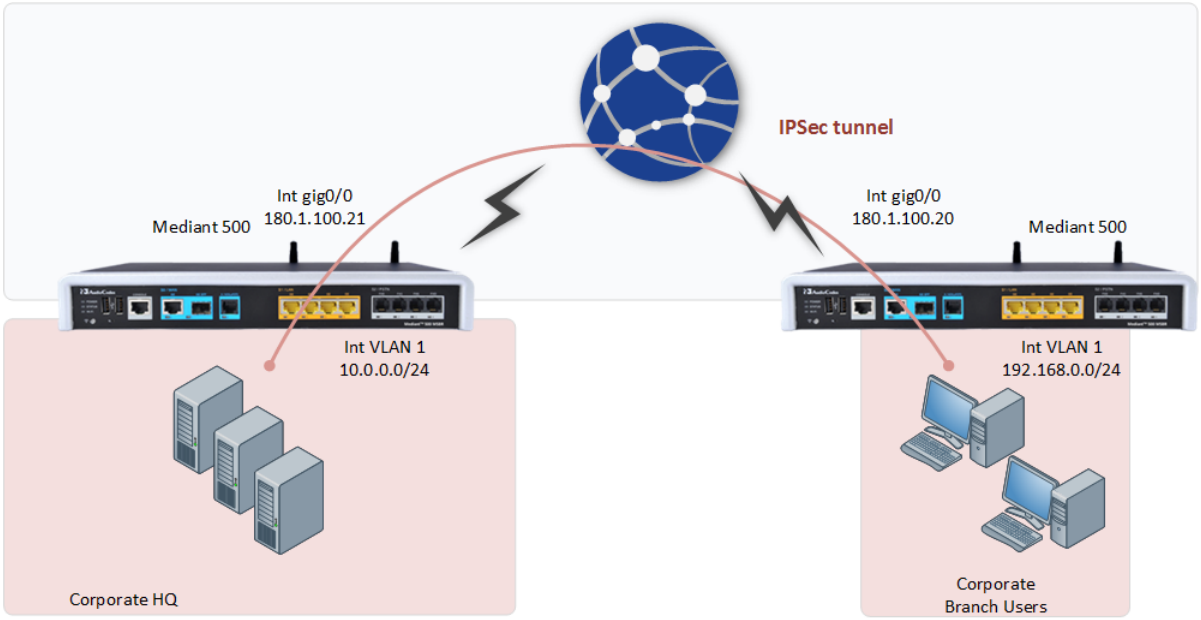
7.1 Configuration Examples

This section provides configuration examples for IPSec.

7.1.1 Configuring IPSec

This example includes two routers connected back to back using interface Gigabitethernet0/0, as shown in Figure 7-2: IPSec Example. All traffic captured in the access-list is encrypted.

Figure 7-2: IPSec Example



IPSec configuration of the device on the right-hand side (Corporate Branch Users) is as follows:

```
access-list ipsec permit ip 192.168.0.0 0.0.0.255 10.0.0.0
0.0.0.255
crypto isakmp policy 1
 encryption aes 128
 authentication pre-share
 hash sha
 group 2
 lifetime 28800
 exit
crypto ipsec transform-set crypto_set1 esp-aes 128 esp-sha-hmac
 mode tunnel
 exit

crypto map MAP1 1 ipsec-isakmp
 set peer 180.1.100.21
 set transform-set crypto_set1
 set security-association lifetime seconds 3600
 match address ipsec
 exit
```

```
crypto isakmp key P@ssw0rd address 180.1.100.21

interface GigabitEthernet 0/0
 crypto map MAP1
```

IPSec configuration of the device on the Corporate HQ is as follows:

```
access-list ipsec permit ip 10.0.0.0 0.0.0.255 192.168.0.0
0.0.0.255
crypto isakmp policy 1
 encryption aes 128
 authentication pre-share
 hash sha
 group 2
 lifetime 28800
exit
crypto ipsec transform-set crypto_set1 esp-aes 128 esp-sha-hmac
 mode tunnel
 exit

crypto map MAP1 1 ipsec-isakmp
 set peer 180.1.100.20
 set transform-set crypto_set1
 set security-association lifetime seconds 3600
 match address ipsec
 exit

crypto isakmp key P@ssw0rd address 180.1.100.20

interface GigabitEthernet 0/0
 crypto map MAP1
```

Note: If configuration requires NAPT and IPsec for the WAN interface, the user should configure a selective NAPT rule which applies the NAPT to all traffic, except the IPSec subnet. This allows access to the Internet for the workstations in the LAN.

Example of Corporate Branch:

```
access-list  selective_nat deny ip 192.168.0.0 0.0.0.255 10.0.0.0
0.0.0.255
access-list  selective_nat permit ip any any
interface GigabitEthernet 0/0
 no napt
 crypto map eth1_MAP
 exit
ip nat inside source list selective_nat interface GigabitEthernet
0/0
```

Use the `show data crypto status` command to view the IPSec status. The following is the output from the command on the device on the branch site:

```
# show data crypto status

IKE peer  [180.1.100.21]
        map      [MAP1-1]
        status   [connected]
        Interface(s): [GigabitEthernet 0/0][2][7][eth1.4010]
```

Use the `show data crypto status` command to view the IPSec status. The following is the output from the command on the device on the Corporate HQ site:

```
MSBR-2# show data crypto status

IKE peer  [180.1.100.20]
        map      [MAP1-1]
        status   [connected]
        Interface(s): [GigabitEthernet 0/0][2][0][eth1]
```

If configuration requires two subnets to be connected using two IPSec tunnels, then in addition to the previous primary configuration, the following configuration needs to be added to the device on the branch site:

```
access-list ipsec permit ip 192.168.2.0 0.0.0.255 10.0.2.0
0.0.0.255
```

The following configuration needs to be added to the device on the Corporate HQ site:

```
access-list ipsec permit ip 10.0.2.0 0.0.0.255 192.168.2.0
0.0.0.255
```

The configuration additions above assume that the subnets 192.168.2.0/24 and 10.0.2.0/24 need to be added.

If configuration requires two devices connected to the Corporate HQ device, then instead of the previous addition to the device, the following configuration needs to be applied to the Corporate HQ device:

```
access-list ipsec permit ip 10.0.2.0 0.0.0.255 192.168.2.0
0.0.0.255
crypto map MAP1 2 ipsec-isakmp
 set peer 180.1.100.40
 set transform-set crypto_set1
 set security-association lifetime seconds 3600
 match address ipsec
 exit
crypto isakmp key P@ssw0rd address 180.1.100.21
```

The above configuration assumes that the third router's GigabitEthernet 0/0 address is 180.1.100.40.

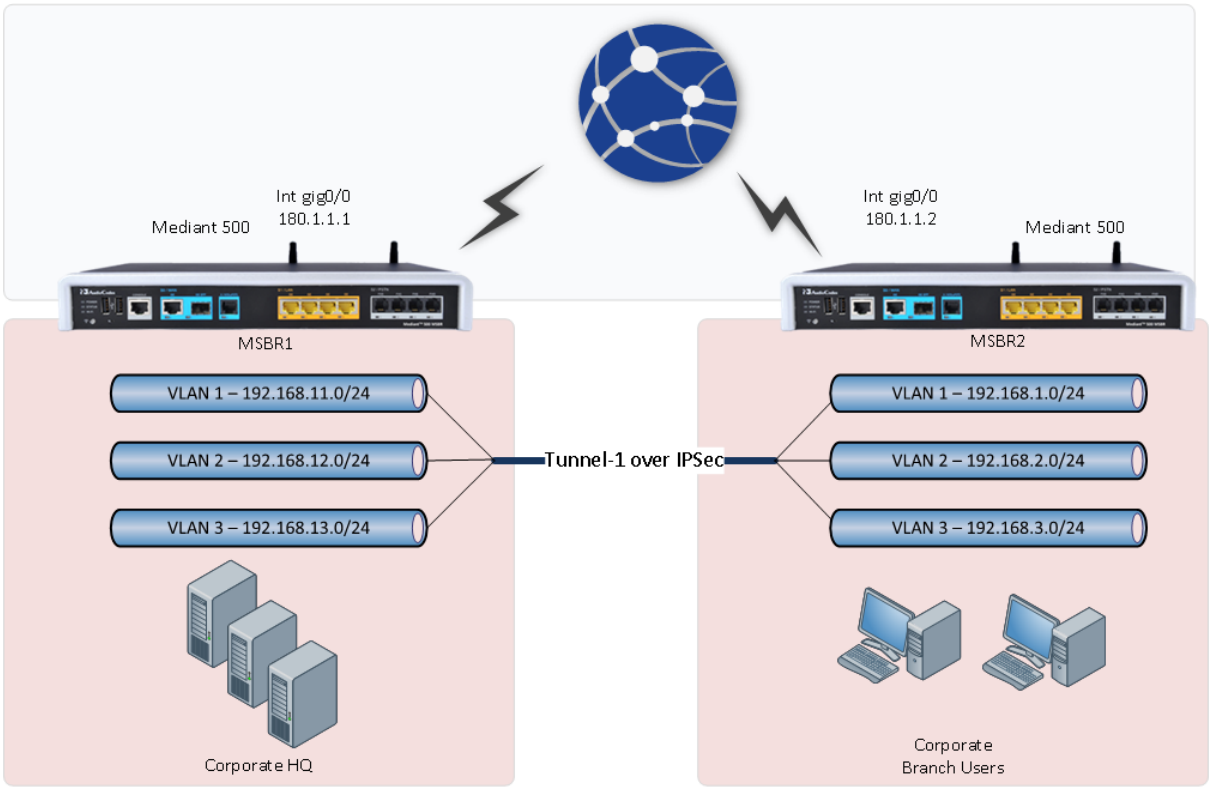
Configuration of the third device is as follows:

```
interface gig 0/0
ip address 180.1.100.40
access-list ipsec permit ip 10.0.2.0 0.0.0.255 192.168.2.0
0.0.0.255
crypto isakmp policy 1
 encryption aes 128
 authentication pre-share
 hash sha
 group 2
 lifetime 28800
 exit

crypto ipsec transform-set crypto_set1 esp-aes 128 esp-sha-hmac
 mode tunnel
 exit

crypto map MAP1 1 ipsec-isakmp
 set peer 180.1.100.21
 set transform-set crypto_set1
 set security-association lifetime seconds 3600
 match address ipsec
 exit

crypto isakmp key P@ssw0rd address 180.1.100.21

interface GigabitEthernet 0/0
crypto map MAP1
```

7.1.2 Configuring IPSec with GRE

This example includes IPSec with GRE where two devices are connected back to back via the Gigabit Ethernet 0/0 interface. Only GRE traffic that is being "caught" by the access list permit gre any any, between the Gigabit Ethernet interfaces is encrypted.

Figure 7-3: GRE over IPSec



The following shows the MSBR1 configuration:

```
conf d
int gigabitethernet 0/0
 ip address 180.1.1.1 255.255.255.0
 no firewall enable
exit
int vla 1
 ip address 192.168.11.1 255.255.255.0
 exit
int vla 2
ip address 192.168.12.1 255.255.255.0
 no shutdown
 exit
int vla 3
 ip address 192.168.13.1 255.255.255.0
 no shutdown
 exit
interface gre 1
 ip address 1.1.1.1 255.255.255.0
 tunnel destination 180.1.1.2
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 180.1.1.2 gigabitethernet 0/0
```

```
ip route 192.168.1.0 255.255.255.0 gre 1
ip route 192.168.2.0 255.255.255.0 gre 1
ip route 192.168.3.0 255.255.255.0 gre 1
access-list  ipsec permit gre any any log
crypto isakmp key Aa123456 address 180.1.1.2
crypto isakmp policy 10
 encr aes 128
 authentication pre-share
 hash sha
 group 2
 lifetime 28800
 exit
crypto ipsec transform-set crypto_set1 esp-3des esp-sha-hmac
 mode tunnel
 exit
crypto map MAP1 10 ipsec-isakmp
 set peer 180.1.1.2
 set transform-set crypto_set1
 match address ipsec
 exit
interface GigabitEthernet 0/0
 crypto map MAP1
```

The following shows the MSBR2 configuration:

```
conf d
int gigabitethernet 0/0
 ip address 180.1.1.2 255.255.255.0
 no firewall enable
exit
int vla 1
 ip address 192.168.1.1 255.255.255.0
 exit
int vla 2
 ip address 192.168.2.1 255.255.255.0
 no shutdown
 exit
int vla 3
 ip address 192.168.3.1 255.255.255.0
 no shutdown
 exit
interface gre 1
 ip address 1.1.1.2 255.255.255.0
 tunnel destination 180.1.1.1
 no shutdown
 exit
ip route 0.0.0.0 0.0.0.0 180.1.1.1 gigabitethernet 0/0
ip route 192.168.11.0 255.255.255.0 gre 1
ip route 192.168.12.0 255.255.255.0 gre 1
ip route 192.168.13.0 255.255.255.0 gre 1
access-list  ipsec permit gre any any log
crypto isakmp key Aa123456 address 180.1.1.1
```

```
crypto isakmp policy 10
 encryption aes 128
 authentication pre-share
 hash sha
 group 2
 lifetime 28800
 exi
crypto ipsec transform-set crypto_set1 esp-3des esp-sha-hm
 mode tunnel
 exit
crypto map MAP1 10 ipsec-isakmp
 set peer 180.1.1.1
 set transform-set crypto_set1
 set security-association lifetime seconds 3600
 match address ipsec
 exit
int gigabitethernet 0/0
crypto map MAP1
```

The following is the output of the routing table of MSBR1. Note that the route through GigabitEthernet 0/0 is marked with [IPSec].

```
MSBR1# sh d ip route
Codes: K - kernel route, C - connected, S - static,
       R - RIP, O - OSPF, B - BGP

 C   1.1.1.0/24 [1/1] is directly connected, GRE 1
 C   180.1.1.0/24 [1/3] is directly connected, GigabitEthernet 0/0
 S   180.1.1.2/32 [1/0] is directly connected, GigabitEthernet 0/0
[IPSec]
 S   192.168.1.0/24 [1/1] is directly connected, GRE 1
 S   192.168.2.0/24 [1/1] is directly connected, GRE 1
 S   192.168.3.0/24 [1/1] is directly connected, GRE 1
 C   192.168.11.0/24 [1/4] is directly connected, VLAN 1
 C   192.168.12.0/24 [1/4] is directly connected, VLAN 2
 C   192.168.13.0/24 [1/4] is directly connected, VLAN 3
```

The following is the output of the routing table of MSBR2. Note that the route through GigabitEthernet 0/0 is marked with [IPSec]:

```
MSBR2# sh d ip route
Codes: K - kernel route, C - connected, S - static,
       R - RIP, O - OSPF, B - BGP

 C   1.1.1.0/24 [1/1] is directly connected, GRE 1
 C   180.1.1.0/24 [1/3] is directly connected, GigabitEthernet 0/0
 S   180.1.1.1/32 [1/0] is directly connected, GigabitEthernet 0/0
[IPSec]
 C   192.168.0.0/24 [1/4] is directly connected, BVI 1
 C   192.168.1.0/24 [1/4] is directly connected, VLAN 1
 C   192.168.2.0/24 [1/4] is directly connected, VLAN 2
 C   192.168.3.0/24 [1/4] is directly connected, VLAN 3
```

```
 S    192.168.11.0/24 [1/1] is directly connected, GRE 1
 S    192.168.12.0/24 [1/1] is directly connected, GRE 1
 S    192.168.13.0/24 [1/1] is directly connected, GRE 1

MSBR2#
```

A debug capture is run while pinging from MSBR1 VLAN 1 to MSBR2 VLAN 1, using the command:

```
debug capture data interface gigabitethernet 0/0 proto all host
180.1.1.1
```

While the ping command was issued in the following matter:

```
ping 192.168.1.1 source data source-address interface vlan 1
```

Note that the traffic is encrypted and therefore, only ESP packets are shown.

```
# debug capture data interface gigabitethernet 0/0 proto all host
180.1.1.1
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on eth1.4010, link-type EN10MB (Ethernet), capture size
96 bytes
10:17:24.936266 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 150: 180.1.1.1 > 180.1.1.2:
ESP(spi=0xce91a06e,seq=0xc), length 11
10:17:24.936858 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 150: 180.1.1.2 > 180.1.1.1:
ESP(spi=0x3647ff5a,seq=0xc), length 11
10:17:25.933155 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 150: 180.1.1.1 > 180.1.1.2:
ESP(spi=0xce91a06e,seq=0xd), length 11
10:17:25.933653 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 150: 180.1.1.2 > 180.1.1.1:
ESP(spi=0x3647ff5a,seq=0xd), length 11
10:17:26.935143 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 150: 180.1.1.1 > 180.1.1.2:
ESP(spi=0xce91a06e,seq=0xe), length 11
10:17:26.935625 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 150: 180.1.1.2 > 180.1.1.1:
ESP(spi=0x3647ff5a,seq=0xe), length 11
10:17:27.934135 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 150: 180.1.1.1 > 180.1.1.2:
ESP(spi=0xce91a06e,seq=0xf), length 11
10:17:27.934665 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 150: 180.1.1.2 > 180.1.1.1:
ESP(spi=0x3647ff5a,seq=0xf), length 11
10:17:29.934720 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
ARP (0x0806), length 60: arp who-has 180.1.1.2 tell 180.1.1.1
```

Note the output of the capture with the "ipsec" keyword, which allows viewing encrypted traffic:

```
debug capture data interface gigabitethernet 0/0 ipsec proto all
host 180.1.1.1
```

Note that traffic is upon the GRE tunnel:

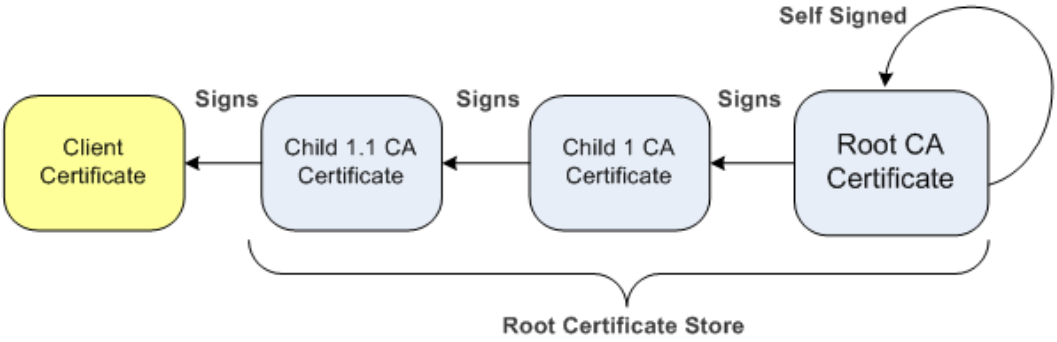
```
debug capture data interface gigabitethernet 0/0 ipsec proto all
host 180.1.1.1
tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode
listening on ipsec2, link-type EN10MB (Ethernet), capture size 96
bytes
10:21:06.709636 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 98: 180.1.1.1 > 180.1.1.2: GREv0, proto IPv4
(0x0800), length 64: 192.168.11.1 > 192.168.1.1: ICMP echo
request, id 27378, seq 1, length 40
10:21:06.710405 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 98: 180.1.1.2 > 180.1.1.1: GREv0, proto IPv4
(0x0800), length 64: 192.168.1.1 > 192.168.11.1: ICMP echo reply,
id 27378, seq 1, length 40
10:21:07.702933 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 98: 180.1.1.1 > 180.1.1.2: GREv0, proto IPv4
(0x0800), length 64: 192.168.11.1 > 192.168.1.1: ICMP echo
request, id 27378, seq 2, length 40
10:21:07.703292 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 98: 180.1.1.2 > 180.1.1.1: GREv0, proto IPv4
(0x0800), length 64: 192.168.1.1 > 192.168.11.1: ICMP echo reply,
id 27378, seq 2, length 40
10:21:08.703879 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 98: 180.1.1.1 > 180.1.1.2: GREv0, proto IPv4
(0x0800), length 64: 192.168.11.1 > 192.168.1.1: ICMP echo
request, id 27378, seq 3, length 40
10:21:08.704280 00:90:8f:89:35:a9 > 00:90:8f:59:4b:56, ethertype
IPv4 (0x0800), length 98: 180.1.1.2 > 180.1.1.1: GREv0, proto IPv4
(0x0800), length 64: 192.168.1.1 > 192.168.11.1: ICMP echo reply,
id 27378, seq 3, length 40
10:21:09.702894 00:90:8f:59:4b:56 > 00:90:8f:89:35:a9, ethertype
IPv4 (0x0800), length 98: 180.1.1.1 > 180.1.1.2: GREv0, proto IPv4
(0x0800), length 64: 192.168.11.1 > 192.168.1.1: ICMP echo
request, id 27378, seq 4, length 40
```

7.1.3 Configuring IPSec with RSA

It is possible to use certificates instead of pre-shared password for authentication. The device provides its own Trusted Root Certificate store. This store lets you manage trusted CA certificates to authenticate the remote side. You can import up to 20 certificates to the store (this amount might be less depending on certificate file size).

This storage can also be used for trusted certificate chains. A certificate chain is a sequence of certificates where each certificate in the chain is signed by the subsequent certificate. The last certificate in the list of certificates is the Root CA certificate, which is self-signed. The purpose of a certificate chain is to establish a chain of trust from a child certificate to the trusted root CA certificate. The CA vouches for the identity of the child certificate by signing upon it. A client certificate is considered trusted if one of the CA certificates in the certificate chain is present in the server certificate directory. For the device to trust a whole chain of certificates, all of them must be imported.

Figure 7-4: Configuring IPSec Using RSA



Each certificate in the file must be Base64 encoded (PEM). When copying-and-pasting the certificates to the device, each Base64 ASCII encoded certificate string must be enclosed between "**-----BEGIN CERTIFICATE-----**" and "**-----END CERTIFICATE-----**".

You must configure the device clock settings, preferably with an NTP server, to make sure that the expiration date for the certificates are correctly validated.

For the IPSEC to authenticate using PKI, the CA certificate or CA chain certificates need to be imported to the device. A certificate signing request (CSR) needs to be first generated and then the signed certificate needs to be imported to the device. In the generation of the signing request, a private key is used. The private key needs to be generated or imported prior to the signing request. Using this signing request, the CA generates a certificate that can then be imported to the device.

This "MSBR certificate" is later used to establish an IPSec connection.

7.1.3.1 Importing Certificates

This procedure describes how to import certificates.

7.1.3.1.1 Private Key

A private key needs to be generated or imported. The private key is used to generate enrollment requests to the CA. To generate a private key, use the following command:

```
(config-isakmp-pki)# private-key generate 2048
Generating new 2048-bit private key, this might take some time...
New 2048-bit private key generated.

(config-isakmp-pki)#
```

7.1.3.1.2 Root Certificate or Chain Certificates

When importing CA certificate or CA chain certificates, you must first import a root CA certificate, then child certificates. All certificate manipulations must be performed using CLI under the PKI (public key infrastructure) configuration section.

1. Enter the following commands:

```
Configure data
crypto isakmp pki 0
```

The relevant available commands in the PKI section are shown in the table below:

Table 7-2: Root Certificate or Chain Certificates

Command	Sub-commands	Description
`certificate`	`-`	Imports device certificate.
`certificate`	`create-self-signed`	Creates a self-signed certificate.
	`delete`	Deletes certificate.
	`detail`	Displays certificates.
	`export`	Exports certificates.
	`import`	Imports certificates.
	`signing-request`	Generates signing requests.
	`status`	Displays current certificate status.
	`subject`	Configures subject name for CSRs and new certificates.
`trusted-root`	`-`	Imports root certificate.
`trusted-root`	`clear-and-import`	Clears Trusted Root certificates and imports new ones in textual PEM format.
	`delete`	Deletes an individual Trusted Root certificate.
	`detail`	Details of a specific root certificate, by number.
	`export`	Exports individual Trusted Root certificate.
	`import`	Imports Trusted Root certificate, in textual PEM format.
	`summary`	Summary of Trusted Root certificates.
`private-key`		Local private key manipulation.
	`delete`	Deletes current private key (use with caution).
	`generate`	Generates new private key and self-signed certificate.

Command	Sub-commands	Description
	`import`	Imports private key, in textual PEM format.

7.1.3.1.3 Import Root Certificates Procedure

1. Go to the PKI CLI section:

```
#configure data
(config-data)#crypto isakmp
```

2. Use the following command to import the Root certificate:

```
(config-isakmp-pki)# trusted-root import
```

The following message is displayed:

```
Enter data below. Type a period (.) on an empty line to
finish.
```

3. Paste a root certificate:

```
-----BEGIN CERTIFICATE-----
MIIFxz…
---output omitted---
…tjkjeqG
-----END CERTIFICATE-----
```

4. Enter dot "." to end root certificate:

If there are other "child root" certificates, repeat from trusted-root import to add more certificates. After the certificate has been imported, check the root certificate using " trusted-root summary" command:

```
(config-isakmp-pki)# trusted-root summary
1 trusted certificates.
Num Subject                          Issuer
Expires
-------------------------------------------------------------------
----------------
  1 ca.local                         ca.local
6/15/2028

MSBR(config-isakmp-pki)#
```

7.1.3.1.4 Import Device Certificate Using Signing Request

1. Go to the PKI CLI section:

```
#configure data
MSBR(config-data)#crypto isakmp
```

2. Create certificate fields names, such as country codes, state, Organization name etc using the command "certificate subject field-set <FIELD NAME> <FIELD VALUE>":

```
(config-isakmp-pki)#certificate subject field-set organization
AC
(config-isakmp-pki)#certificate subject field-set country IL
(config-isakmp-pki)#certificate subject field-set common-name
MSBR-7
```

3. Generate a signing request:

```
(config-isakmp-pki)#certificate signing-request
Certificate signing request:

-----BEGIN CERTIFICATE REQUEST-----
MIICgTCC…
---output omitted---
…zxcsF
-----END CERTIFICATE REQUEST-----

Send this request to your security administrator for signing,
then upload the new signed certificate to the device.
```

4. Using the signing request, obtain the device certificate and then import the obtained certificate using the import command "Certificate import".

```
(config-isakmp-pki)#certificate import
Enter data below. Type a period (.) on an empty line to
finish.
-----BEGIN CERTIFICATE-----
MIIEoDCCAoigAwIB
---output omitted---

-----END CERTIFICATE-----
.
File replaced.

MSBR(config-isakmp-pki)#
```

5. Check if the imported certificate matches the private key with which it was generated:

```
MSBR-31(config-isakmp-pki)# certificate status
Certificate subject: /C=IL/CN=MSBR-31
Certificate issuer :
/C=IL/ST=CENTER/L=LOD/O=Audiocodes/OU=R&D/CN=ca.local/emailAdd
ress=timg@audiocodes.com
Signature Algorithm: sha256WithRSAEncryption
Time to expiration : 369 days

Key size: 2048 bits
Active sockets: 0
The currently-loaded private key matches this certificate.
```

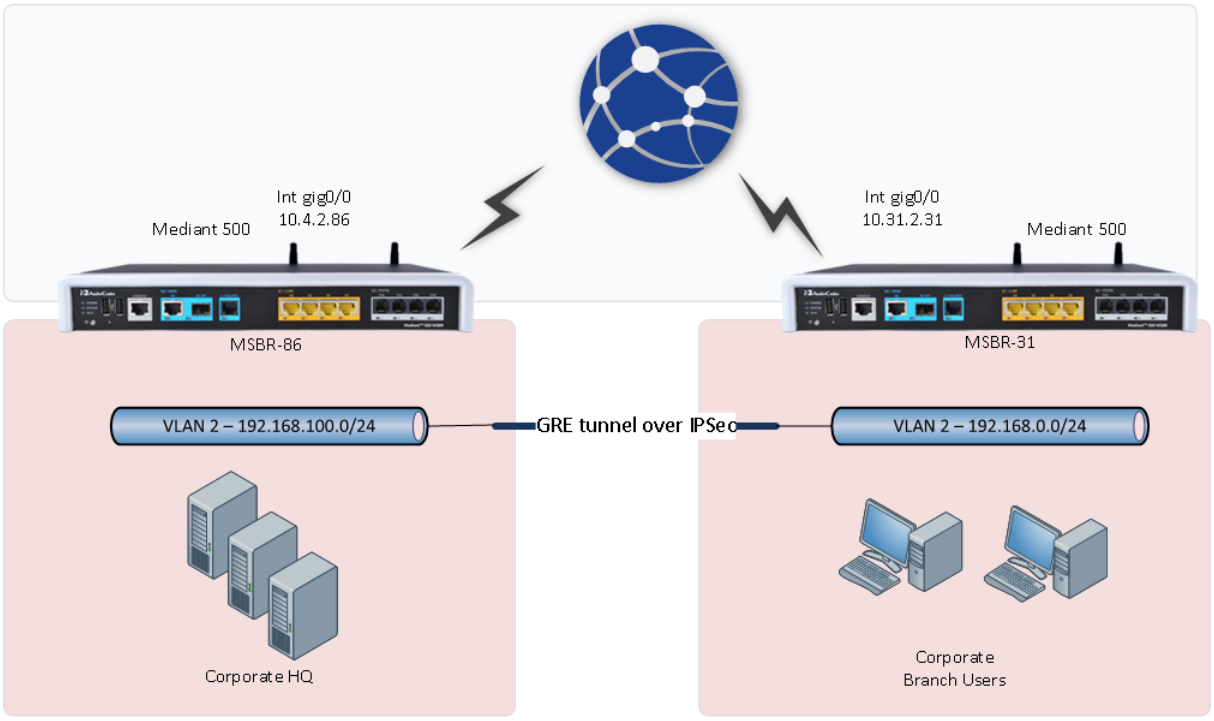
If the imported certificate does not match the generated key, the output is as follows:

```
MSBR-99(config-isakmp-pki)# certificate status
Certificate subject:
/C=IL/ST=Center/L=Lod/O=AC/OU=R&D/CN=ca.local/emailAddress=timg@audiocodes.com
Certificate issuer :
/C=IL/ST=Center/L=Lod/O=AC/OU=R&D/CN=ca.local/emailAddress=timg@audiocodes.com
Signature Algorithm: sha256WithRSAEncryption
Time to expiration : 3522 days

Key size: 1024 bits
Active sockets: 0
The currently-loaded private key DOES NOT match this certificate.
```

7.1.3.1.5 Device PKI Configuration Example

The following is an example of the configuration of IPSec using PKI authentication between two routers using a GRE tunnel. Both devices have an NTP server configured, and certificates were imported as described in the previous sections.

Figure 7-5: Device PKI Configuration Example



Configuration of MSBR-31 is as follows:

```
configure data
   access-list  IPSEC permit gre any any
   access-list  ALL_BUT_IPSEC deny gre any any
   access-list  ALL_BUT_IPSEC permit ip any any
   crypto isakmp policy 1
    encr aes 256
    authentication rsa-sig
    hash sha
    group 5
    lifetime 3600
    exit
   crypto ipsec transform-set crypto_set esp-aes 256 esp-sha-hmac
    mode tunnel
    exit
   crypto map MAP1 1 ipsec-isakmp
    set peer 10.4.40.86
    set transform-set crypto_set
    set security-association lifetime seconds 3600
    match address IPSEC
    set default-route
    exit
   interface GigabitEthernet 0/0
    ip address 10.31.2.31 255.255.255.0
```

```
    mtu auto
    desc "WAN Copper"
    no ipv6 enable
    speed auto
    duplex auto
    no service dhcp
    ip dns server auto
    no napt
    crypto map MAP1
    firewall enable
    no shutdown
   exit
interface VLAN 1
    ip address 192.168.0.1 255.255.255.0
    mtu auto
    desc "LAN switch VLAN 1"
    no ipv6 enable
    ip dhcp-server network 192.168.0.3 192.168.0.8 255.255.255.0
    ip dhcp-server dns-server 0.0.0.0
    ip dhcp-server netbios-name-server 0.0.0.0
    ip dhcp-server lease 0 1 0
    ip dhcp-server provide-host-name
    ip dhcp-server ntp-server 0.0.0.0
    ip dhcp-server tftp-server 0.0.0.0
    ip dhcp-server override-router-address 0.0.0.0
    ip dhcp-server next-server 0.0.0.0
    service dhcp
    ip dns server static
    ip name-server 1.1.1.1 8.8.8.8
    no napt
    no firewall enable
    no link-state monitor
    no shutdown
   exit
interface GRE 2
    ip address 16.0.0.2 255.255.255.252
    mtu 1400
    desc "WAN GRE 2"
    no napt
    tunnel source GigabitEthernet 0/0
    tunnel destination 10.4.40.86
    keepalive 1 2
    no firewall enable
    no shutdown
   exit

  ip nat inside source list ALL_BUT_IPSEC interface
GigabitEthernet 0/0
 ip route 10.4.2.0 255.255.255.0 10.31.2.1 GigabitEthernet 0/0
 ip route 192.168.100.0 255.255.255.0 gre 2
```

Configuration of MSBR-86 is as follows:

```
configure data
   access-list  IPSEC permit gre any any
   access-list  ALL_BUT_IPSEC deny gre any any
   access-list  ALL_BUT_IPSEC permit ip any any
   crypto isakmp policy 1
    encr aes 256
    authentication rsa-sig
    hash sha
    group 5
    lifetime 3600
    exit
   crypto ipsec transform-set crypto_set esp-aes 256 esp-sha-hmac
    mode tunnel
    exit
   crypto map MAP1 1 ipsec-isakmp
    set peer 10.31.2.31
    set transform-set crypto_set
    set security-association lifetime seconds 3600
    match address IPSEC
    set default-route
    exit
   interface GigabitEthernet 0/0
    ip address 10.4.2.86 255.255.255.0
    mtu auto
    desc "WAN Copper"
    no ipv6 enable
    speed auto
    duplex auto
    no service dhcp
    ip dns server auto
    no napt
    crypto map MAP1
    firewall enable
    no shutdown
   exit
interface VLAN 1
    ip address 192.168.100.1 255.255.255.0
    mtu auto
    desc "LAN switch VLAN 1"
    no ipv6 enable
    no service dhcp
    ip dns server static
    ip name-server 1.1.1.1 8.8.8.8
    no napt
    no firewall enable
    no link-state monitor
    no shutdown
   exit
interface GRE 2
    ip address 16.0.0.1 255.255.255.252
```

```
    mtu 1400
    desc "WAN GRE 2"
    no napt
    tunnel source GigabitEthernet 0/0
    tunnel destination 10.31.2.31
    keepalive 1 2
    no firewall enable
    no shutdown
   exit

  ip nat inside source list ALL_BUT_IPSEC interface
GigabitEthernet 0/0
 ip route 10.31.2.0 255.255.255.0 10.4.2.1 GigabitEthernet 0/0
 ip route 192.168.0.0 255.255.255.0 gre 2
```

To check that IPSec is up, use the `show data crypto status` command. The expected output is as follows:

```
MSBR-31# show data crypto status

IKE peer  [10.4.40.86]
        map      [MAP1-1]
        status   [connected]
        interface(s): [GigabitEthernet 0/0]
        15-seconds input rate: 512 bits/sec
        15-seconds output rate: 1088 bits/sec
        uptime: 22.40 Minutes

MSBR-31#
```

7.1.4 Configuring IPSec with IKEv2

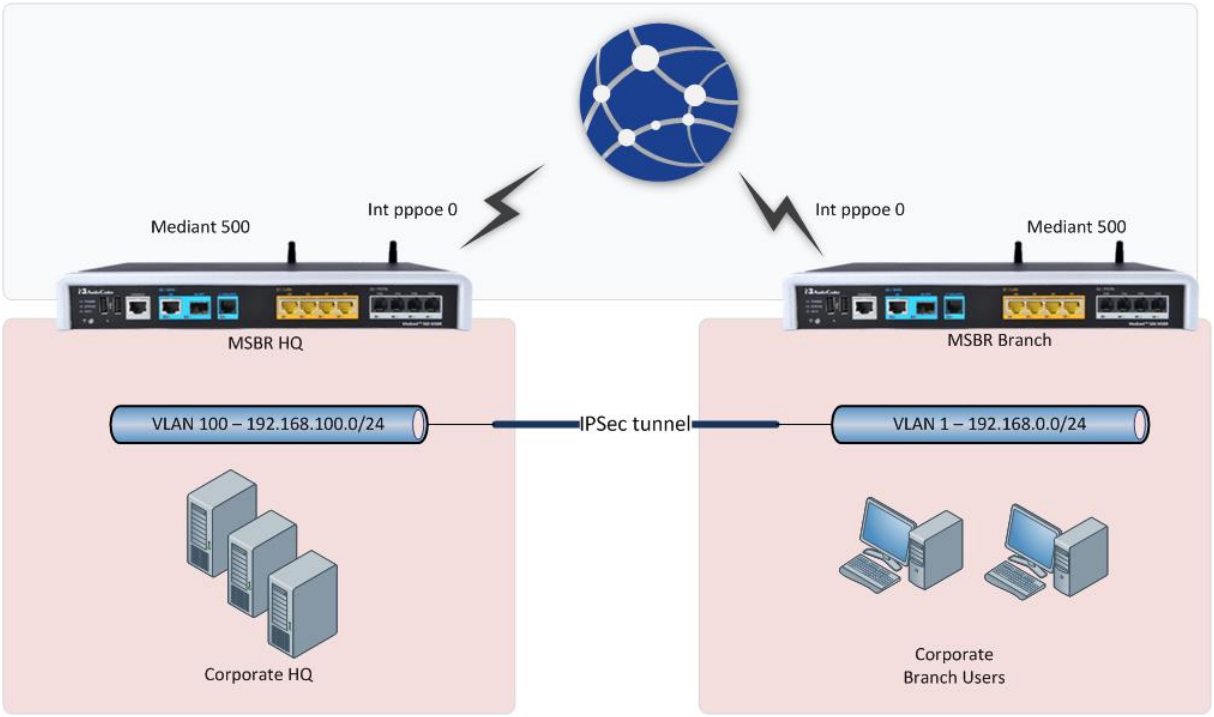
The MSBR supports Internet Key Exchange (IKE) version 2. With IKEv2, the MSBR supports configuring the peer by IP address or FQDN.

For the identity of the IKEv2 peer, the MSBR supports:

- IP address
- Email
- FQDN

7.1.4.1 Configuration Example

This configuration example is based on the following topology:



The IP address of the "MSBR HQ" is constant (fixed), while the IP address of the "MSBR Branch" may be dynamic and change every time the interface PPPoE 0 reconnects. In this scenario, the identity of the MSBR Branch should therefore, not be by IP address because it changes; instead, it should be by FQDN or email address.

Configuration of MSBR Branch:

```
  configure data
   access-list  ipsec permit ip 192.168.0.0 0.0.0.255
192.168.100.0 0.0.0.255 log
   access-list  all_but_ipsec deny ip 192.168.0.0 0.0.0.255
192.168.100.0 0.0.0.255 log
   access-list  all_but_ipsec permit ip any any log
   crypto isakmp obscured-key Vhc2aWtpb2lr address 82.80.170.113
   crypto isakmp identity fqdn home.timg.pro
   crypto isakmp policy 1
    encr aes 256
    authentication pre-share
    hash sha
    group 5
    lifetime 3600
```

```
 ike v2
 exit
crypto ipsec transform-set crypto_set esp-aes 256 esp-sha-hmac
 mode tunnel
 exit
crypto map MAP1 1 ipsec-isakmp
 set peer 82.80.170.113
 set transform-set crypto_set
 set security-association lifetime seconds 3600
 match address ipsec
 exit
interface VLAN 1
 no ip address
 bridge-group 1
 mtu auto
 desc "LAN switch VLAN 1"
 no ipv6 enable
 no service dhcp
 no link-state monitor
 no shutdown
exit
interface dsl 0/2
 #DSL configuration is automatic
 #Termination cpe
 no shutdown
exit
interface EFM 0/2
 no ip address
 mtu auto
 desc "VDSL"
 no ipv6 enable
 no service dhcp
 ip dns server static
 no shutdown
exit
interface BVI 1
 ip address 192.168.0.1 255.255.255.0
 mtu auto
 desc "LAN Bridge"
 no ipv6 enable
 ip dhcp-server network 192.168.0.3 192.168.0.123 255.255.255.0
 ip dhcp-server dns-server 0.0.0.0
 ip dhcp-server netbios-name-server 0.0.0.0
 ip dhcp-server lease 0 1 0
 ip dhcp-server provide-host-name
 ip dhcp-server ntp-server 0.0.0.0
 ip dhcp-server tftp-server 0.0.0.0
 ip dhcp-server override-router-address 0.0.0.0
 ip dhcp-server next-server 0.0.0.0
 service dhcp
 ip dns server static
```

```
    no napt
    no firewall enable
    no shutdown
   exit
   interface pppoe 0
    firewall enable
    napt
    mtu auto
    ppp user 0543150513@014 obscured-pass vu/atLSt8g==
    ppp authentication chap
    ppp authentication ms-chap
    ppp authentication ms-chap-v2
    ppp authentication pap
    ppp lcp-echo 6 5
    no ppp compression
    ip address auto
    ipv6 address autoconfig
    ip dns server auto
    underlying EFM 0/2
    crypto map MAP1
    network wan
    no shutdown
   exit
   ip nat inside source list all_but_ipsec interface PPPOE 0
   ip route 0.0.0.0 0.0.0.0 PPPOE 0 1
  exit
```

The MSBR Branch configuration defines the IKEv2 peer as an IP address. It's important to note that the identity of the MSBR Branch is set to **home.timg.pro**.

Configuration of MSBR HQ:

```
configure data
   access-list  all_but_ipsec deny ip 192.168.100.0 0.0.0.255
192.168.0.0 0.0.0.255 log
   access-list  all_but_ipsec permit ip any any log
   access-list  ipsec permit ip 192.168.100.0 0.0.0.255
192.168.0.0 0.0.0.255 log
   crypto isakmp key Aa123456 address home.timg.pro
   crypto isakmp policy 1
    encr aes 256
    authentication pre-share
    hash sha
    group 5
    lifetime 3600
    ike v2
    exit
   crypto ipsec transform-set crypto_set esp-aes 256 esp-sha-hmac
    mode tunnel
    exit
   crypto map MAP1 1 ipsec-isakmp
    set peer home.timg.pro
    set transform-set crypto_set
```

```
    set security-association lifetime seconds 3600
    match address ipsec
    set default-route
    exit
   interface dsl 0/2
    #DSL configuration is automatic
    #Termination cpe
    no shutdown
   exit
   interface EFM 0/2
    no ip address
    mtu auto
    desc "VDSL"
    no ipv6 enable
    no service dhcp
    ip dns server static
    no shutdown
   exit
   interface VLAN 100
    no ip address
    mtu auto
    desc "LAN switch VLAN 100"
    no ipv6 enable
    no service dhcp
    ip dns server static
    no firewall enable
    no link-state monitor
    no ipv6 nd ra suppress
    ipv6 address autoconfig
    no shutdown
   exit
   interface BVI 100
    ip address 192.168.100.1 255.255.255.0
    mtu auto
    desc "Bridge"
    ip dhcp-server network 192.168.100.3 192.168.100.33
255.255.255.0
    ip dhcp-server dns-server 8.8.8.8
    ip dhcp-server netbios-name-server 0.0.0.0
    ip dhcp-server lease 0 1 0
    ip dhcp-server provide-host-name
    ip dhcp-server time-offset 0
    ip dhcp-server netbios-node-type 0
    ip dhcp-server ntp-server 0.0.0.0
    ip dhcp-server tftp-server 0.0.0.0
    ip dhcp-server override-router-address 0.0.0.0
    ip dhcp-server next-server 0.0.0.0
    service dhcp
    ip dns server static
    no napt
    no firewall enable
    no shutdown
```

```
  exit
  ip nat inside source list all_but_ipsec interface PPPOE 0
  ip route 0.0.0.0 0.0.0.0 PPPOE 0 1
exit
```

The MSBR HQ has an IKEv2 peer that is configured with an FQDN as **home.timg.pro**. This DNS resolves into the MSBR Branch's IP address and serves as the MSBR Branch identity.

This page is intentionally left blank.

8 L2TP VPN Server

The device supports L2TP VPN servers. With this feature, the client can connect to the device from other locations using Windows dialer. To configure the L2TP VPN server, use the following commands:

Table 8-1: L2TP VPN Servers

Command	Description
`# configure data`	Configuration of the L2TP server on data level.
`(config-data)# l2tp-server`	Configuration of L2TP server.
`(conf-l2tps)# ppp authentication mschap`	Enable mschap authentication.
`(conf-l2tps)# ppp authentication mschapv2`	Enable mschap version 2 authentication.
`(conf-l2tps)# ipsec key <password>`	Enable IPSec with password <password>.
`# show data l2tp-server`	Displays users connected to the L2TP server.

For users to connect to the device using L2TP, the users need to be configured. Use the following commands to configure the users:

Table 8-2: L2TP VPN User Configuration

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# user <user name> password <password>`	Configure a user with a name <user name> and password <password>.

Some operating systems don't have “NAT traversal” (NAT-T) enabled by default. Depending on network topology and in some cases, this is required.

In Windows 10 operating system, NAT traversal can be enabled by editing the registry (regedit.exe):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PolicyAgent
```

A DWORD 32 type value named “AssumeUDPEncapsulationContextOnSendRule” should be set to 2 to enable NAT traversal. If the parameter doesn’t exist, then assume it is set to 0, meaning that there is no connection to servers behind NAT. If the parameter exists, then 1 means a connection where VPN server is behind NAT, and 2 means a connection where server and the client are behind NAT.

8.1 Configuration Example

This example configures an L2TP VPN server and a Windows 7 client to connect to the server.

The following must be configured on the device that acts as an L2TP server:

```
l2tp-server
     ip range 192.168.1.3 192.168.1.8
     no ppp authentication pap
     ppp authentication chap
```

```
    ppp authentication mschap
    ppp authentication mschapv2
    idle-timeout 60
    ipsec key LinePass!1
    no shutdown
    exit
```

The above configuration configures address 192.168.1.3 to 192.168.1.8 for L2TP clients. The chap, mschap, mschap version two protocols are selected for the authentication. The key "LinePass!1" is used for the IPSec encryption between the client and server.

The following is the user configuration for the clients:

```
  vpn-users
    user AudioCodes key P@ssw0rd
    exit
```

Note that `show running-config` displays the passwords and keys in obscured format.

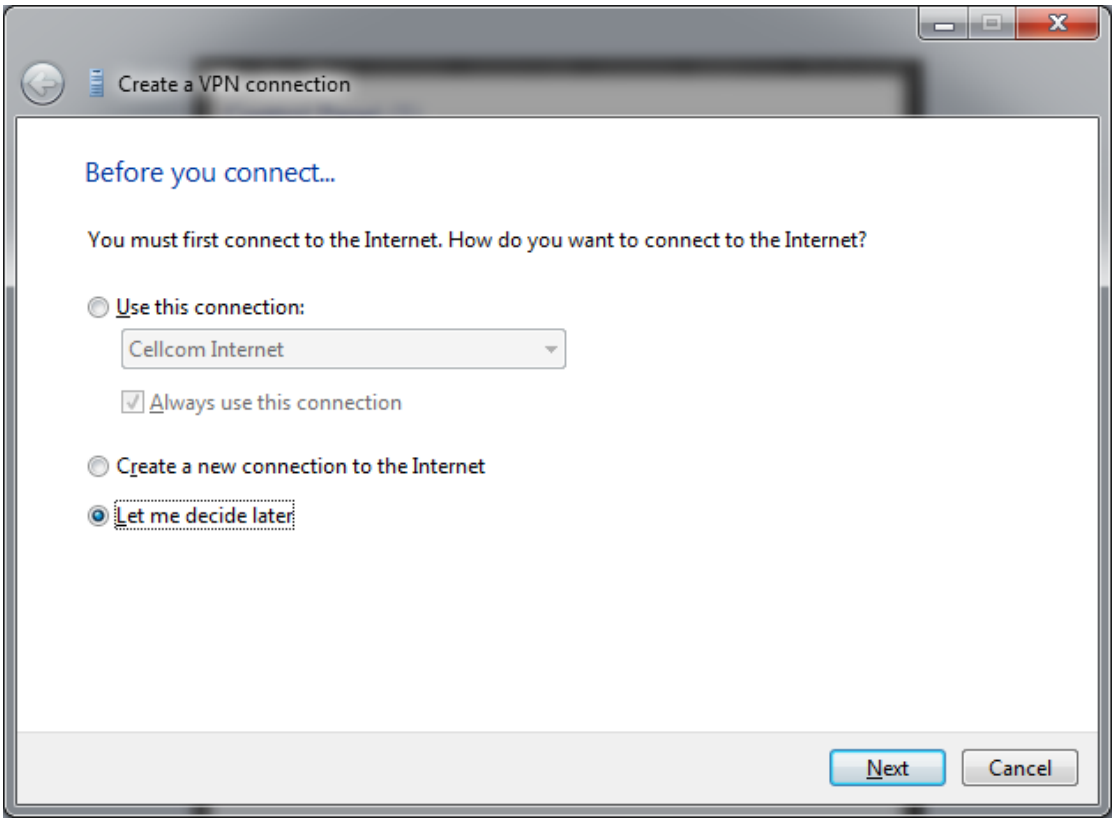
➢ To configure Windows 7 to connect to the L2TP server:

1. Click the Windows icon on the left, and in the search text box, type "vpn".

Figure 8-1: VPN Console

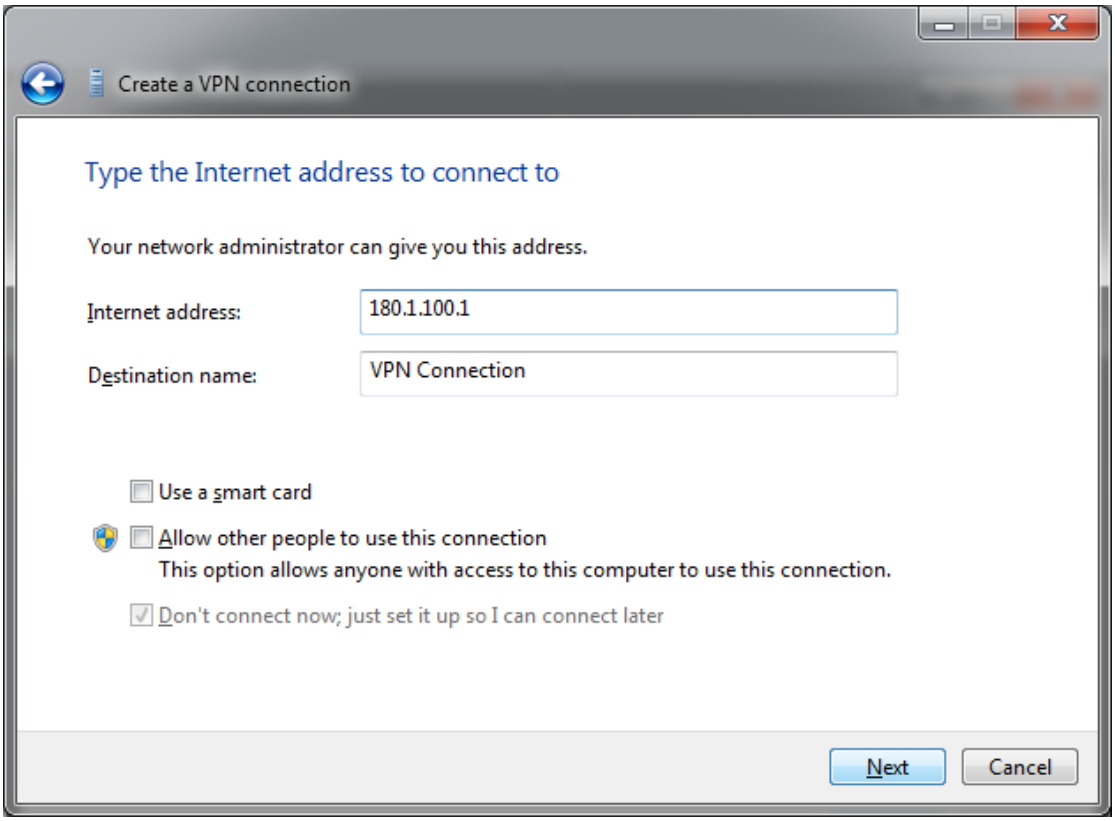
2. Click the **Set up a virtual private network (VPN) connection** link.

Figure 8-2: Select Connection Type



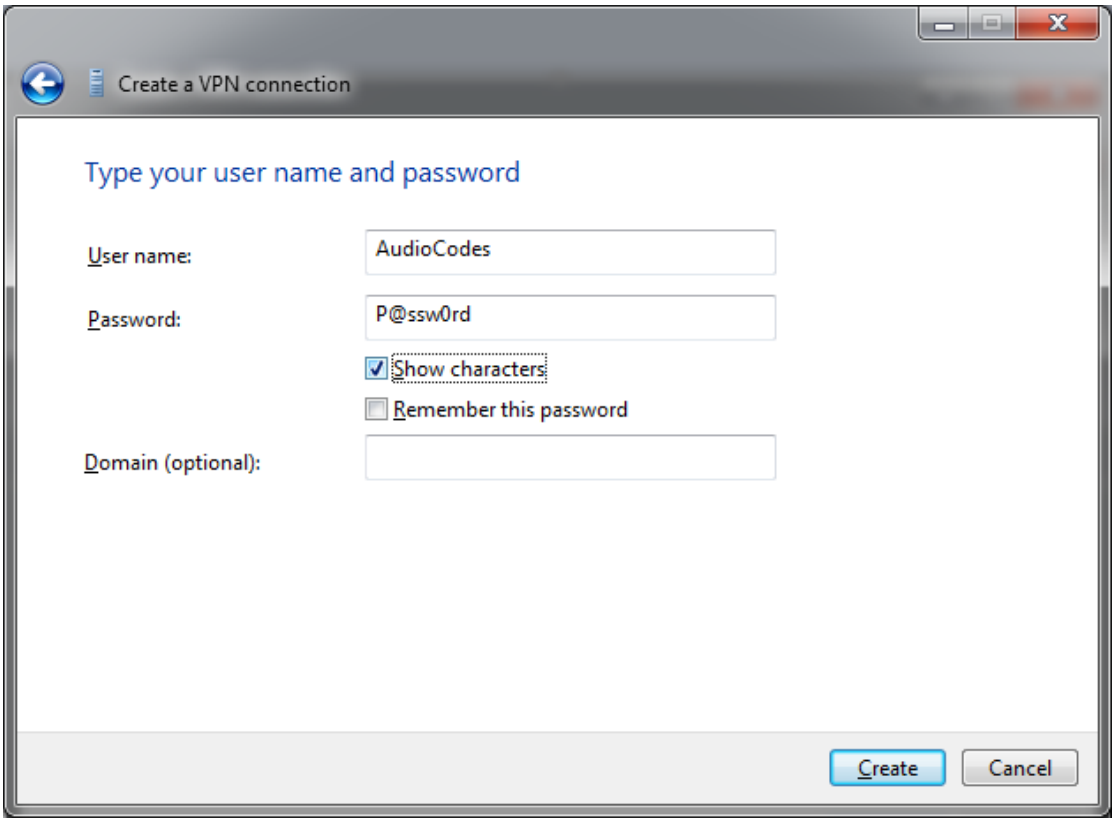
3. Select the **Let me decide later** option, and then click **Next**.

Figure 8-3: VPN Server IP Address



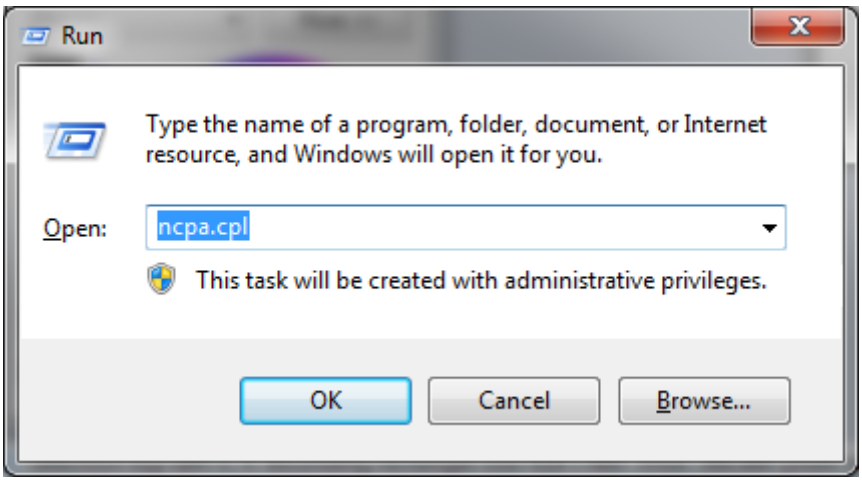
4. In the 'Internet address' field, enter the VPN IP address (typically, the device's WAN interface).
5. In the 'Destination name' field, enter the destination name, which will later become the dialer's name in the Network Connection window.
6. Click **Next**.

Figure 8-4: L2TP Username and Password



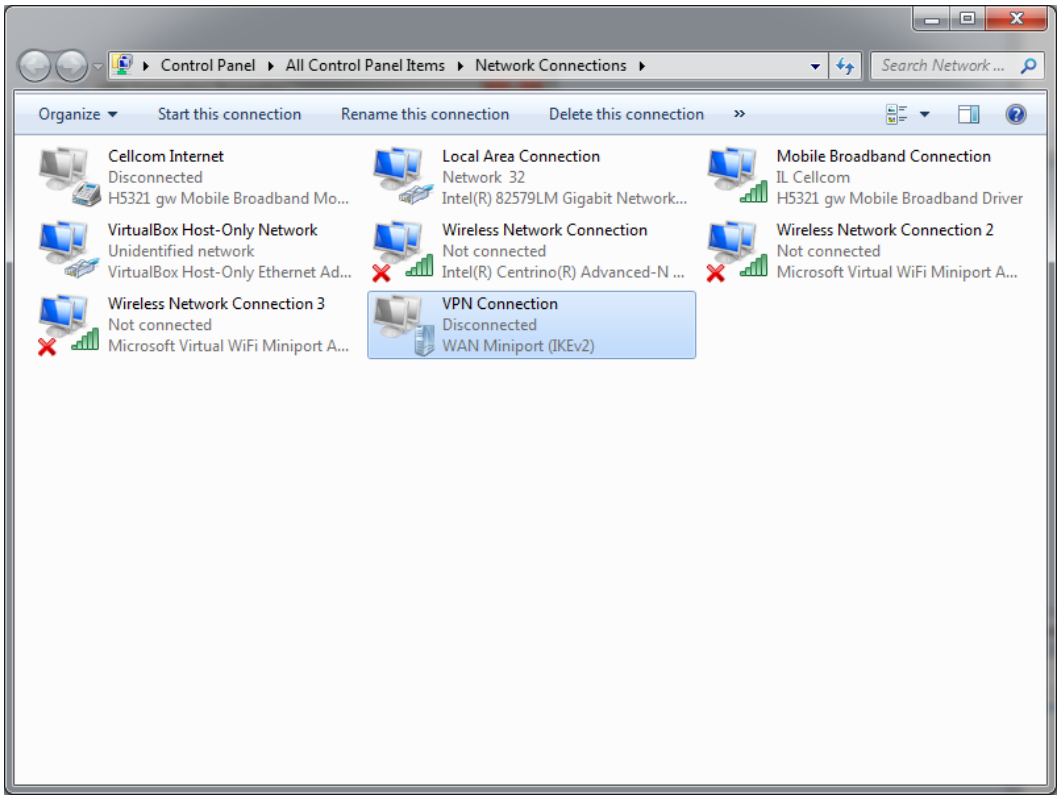
7. Enter the user name and password that was previously configured on the device, and then click **Create**.
8. Open the Network Connections window:
   a. Press the WINDOWS+R key combination; the Run window appears:

Figure 8-5: Run Window



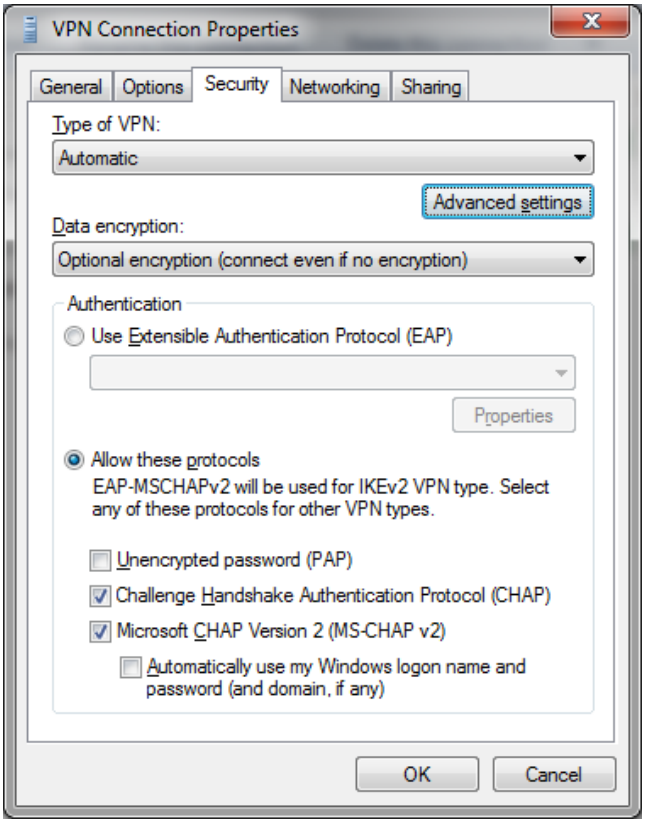
   b. In the 'Open' field, enter "ncpa.cpl", and then click **OK**.

Figure 8-6: Network Connections Window



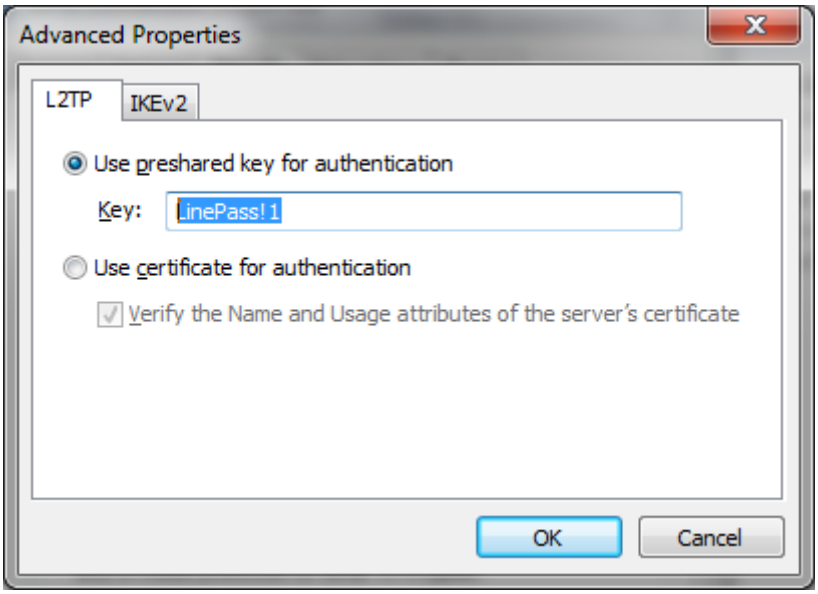
9. Right-click **VPN Connection** that you just created, and then choose **Properties**.

Figure 8-7: VPN Connection Properties Security Tab



10. Click the **Security** tab, and then click **Advanced settings**.

Figure 8-8: VPN Connection Advanced Properties



11. Select the **Use preshared key for authentication** option, and then enter the key previously configured on device, and then click **OK**.
12. Click **OK** until you're back at the Network Connections window.
13. Double-click **VPN Connection**.

Figure 8-9: VPN Connection Dialer

14. Enter the username and password, and then click **Connect**.

15. When the connection is successfully established, in the device use the `show data l2tp-server` command to view the connected users:

```
MSBR-1# show data l2tp-server
Conn# Username                              IP
Rx/Tx    Uptime
----- ------------------------------------ ----------------- --
--------- ------
300   AudioCodes                            192.168.1.3
3832/1514  1220
Total 1 connections.

MSBR-1#
```

This page is intentionally left blank.

9 802.1X

From Version 6.8, the device supports dot1x. The dot1x is a protocol that allows or denies access of a host to the network based on the hosts' authentication. To configure 802.1x using an authentication server, perform the following configuration steps:

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# dot1x radius-server host 192.168.0.200 auth-port 1812 key P@ssw0rd`	Configure a RADIUS server with IP address 192.168.0.200 on port 1812, with the key "P@ssw0rd". Instead of specifying the host, the "local" keyword can be used. In this case, local users configured on the device will be used.
`(config-data)# dot1x lan-authentication enable`	Enable dot1x authentication globally.
`(config-data)# interface gigabitethernet 4/3`	Configure the interface gigabitethernet 4/3.
`(conf-if-GE 4/3)# authentication dot1x single-host |multi-host`	Configure dot1x on the interface, using a single-host – only one MAC address of the supplicant is allowed on the port, or multi-host, allow any connected MAC.
`# show data dot1x-status`	Displays dot1x status.

To configure dot1x authentication using a local server, do the following:

Command	Description
`# configure data`	Enter the data configuration menu.
`(config-data)# dot1x radius-server local`	Use local users configured on the device to allow access to the network.
`(config-data)# dot1x local-user administrator password P@ssw0rd`	Configure username "administrator" with password "P@ssw0rd".
`(conf-if-GE 4/3)# authentication dot1x single-host |multi-host`	Configure dot1x on the interface, using single-host – only one MAC address of the supplicant is allowed on the port, or multi-host, allow any connected MAC.

From Version 7.2 (see Release Notes for exact version), the device supports 802.1x as a client (supplicant) - the authenticated device. To configure 802.1x authentication as a client (supplicant), do the below. Once configured (and run `exit`), configuration is loaded and negotiation with the 802.1x authenticator (e.g., secure LAN switch) begins. If the supplicant's credentials are valid, the authenticator authorizes traffic on the secure port connected to the device. You can use the `show data dot1x-supplicant-status` to view the status.

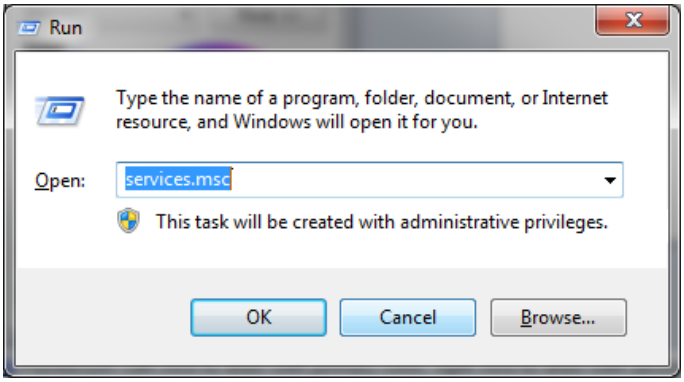
Command	Description
`# configure data`	**Enter the data configuration menu.**
`(config-data)# dot1x supplicant`	**Enter the supplicant command mode.**
`(config-dot1x-supplicant)# set identity`	**Define the supplicant's identity string.**
`(config-dot1x-supplicant)# set mode`	**Enable the 802.1x supplicant mode using EAP-MD5, EAP-PEAP, or EAP-TLS.**
`(config-dot1x-supplicant)# set password`	**Define the supplicant's password.**
`(config-dot1x-supplicant)# set port-type`	**Define the supplicant's port type to run on.**
`(config-dot1x-supplicant)# set tls-ctx`	**Define the TLS Context (ID) for the supplicant.**

9.1 Activating dot1x Authentication on Windows 7

➢ **To activate dot1x authentication on Windows 7:**

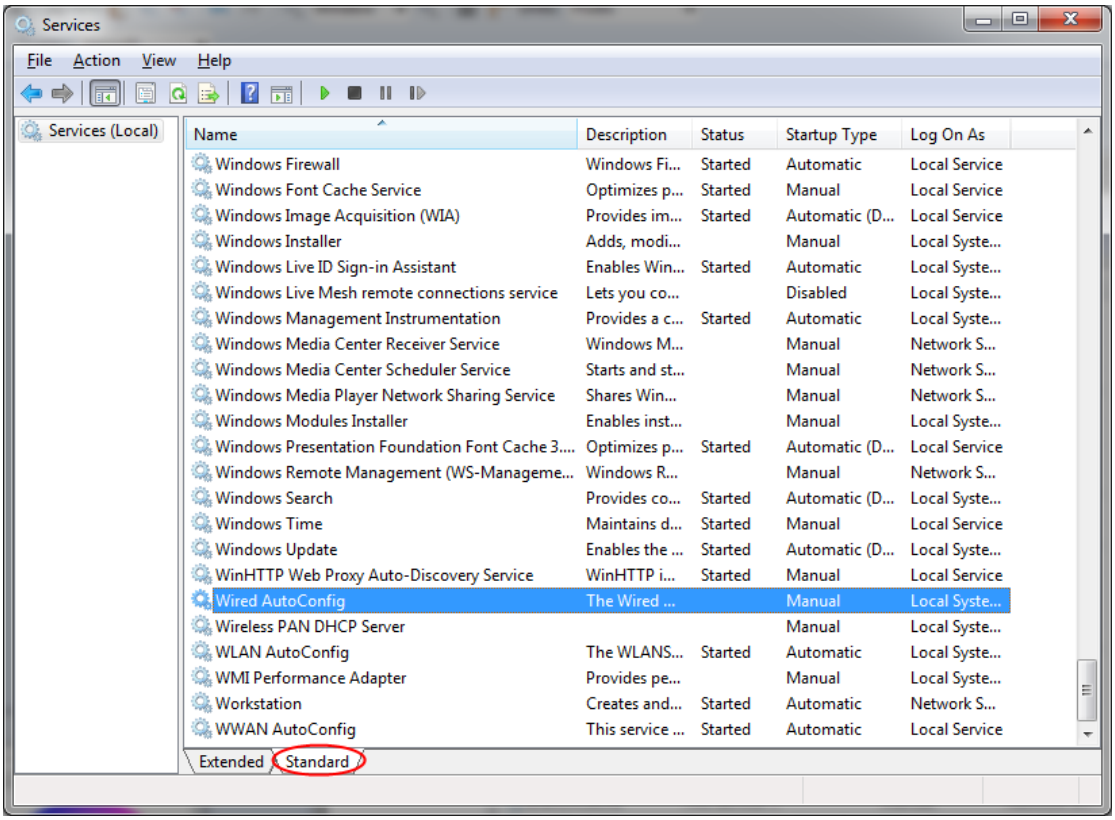
1. Press the Windows + R key combination to open the Run window.

Figure 9-1: Run Window



2. In the 'Open' field, type "services.msc", and then click **OK**.

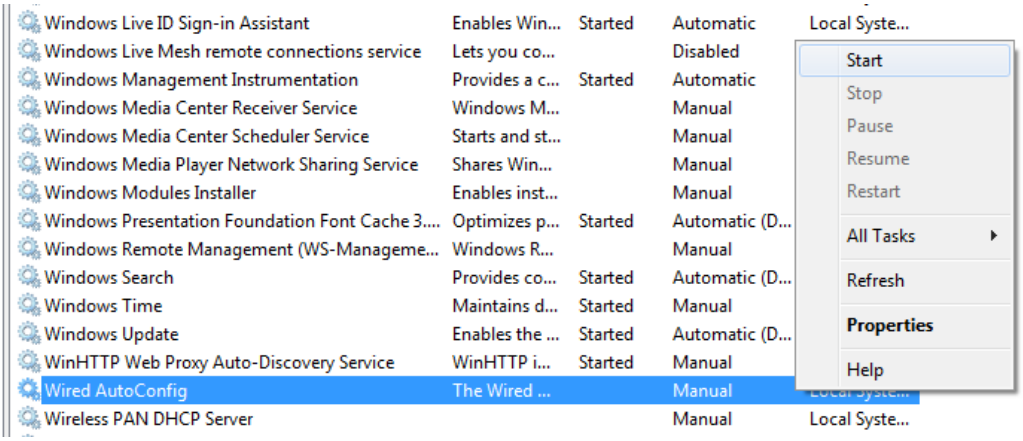
Figure 9-2: Services Window



3. Navigate to the **Standard** tab, and locate the "Wired AutoConfig" entry.

4. Right-click **Wired AutoConfig**, and then from the shortcut menu, choose **Start**, as shown below:

Figure 9-3: Wired AutoConfig Service



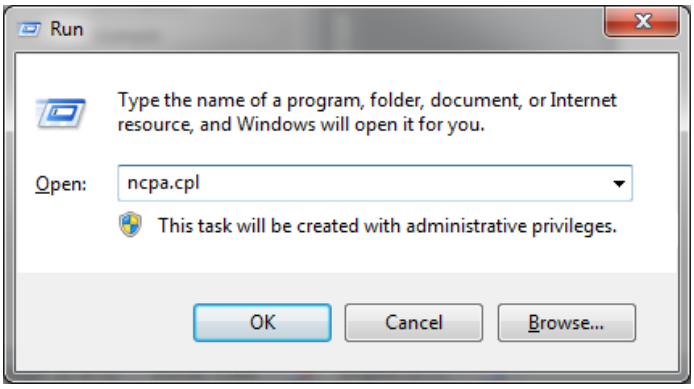
The actions above should activate dot1x authentication for all interfaces on Windows 7.

9.2 Configuring dot1x on Windows 7

➢ **To configure dot1x on Windows 7:**

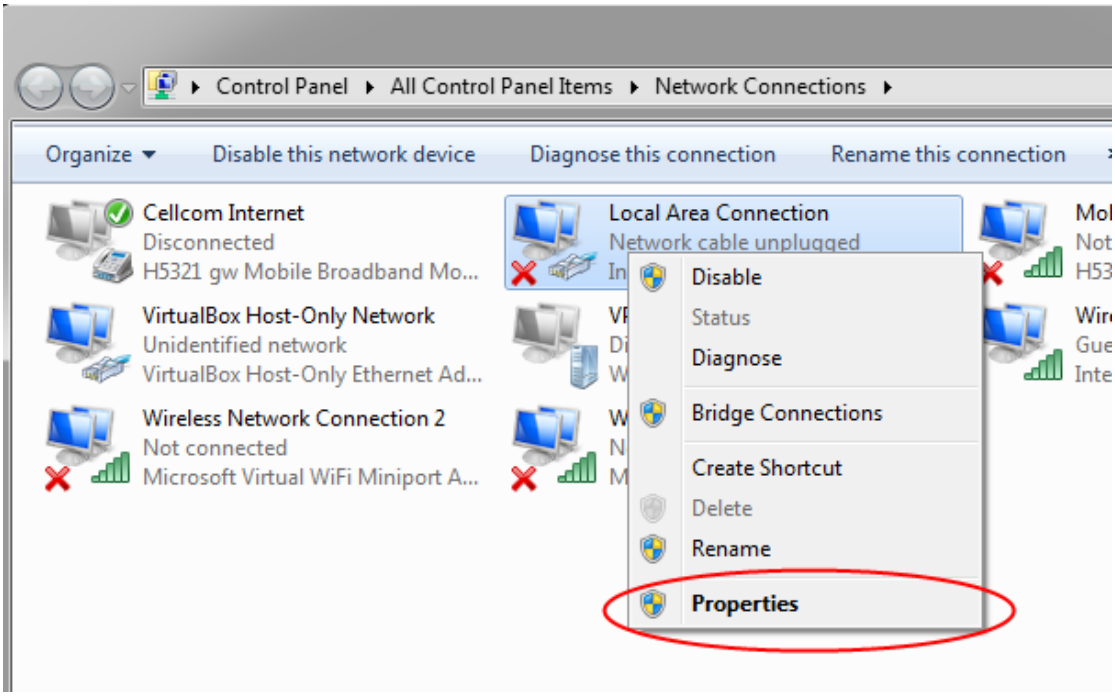
1. Press the Windows+R key combination to open the Run window.

Figure 9-4: Run Window



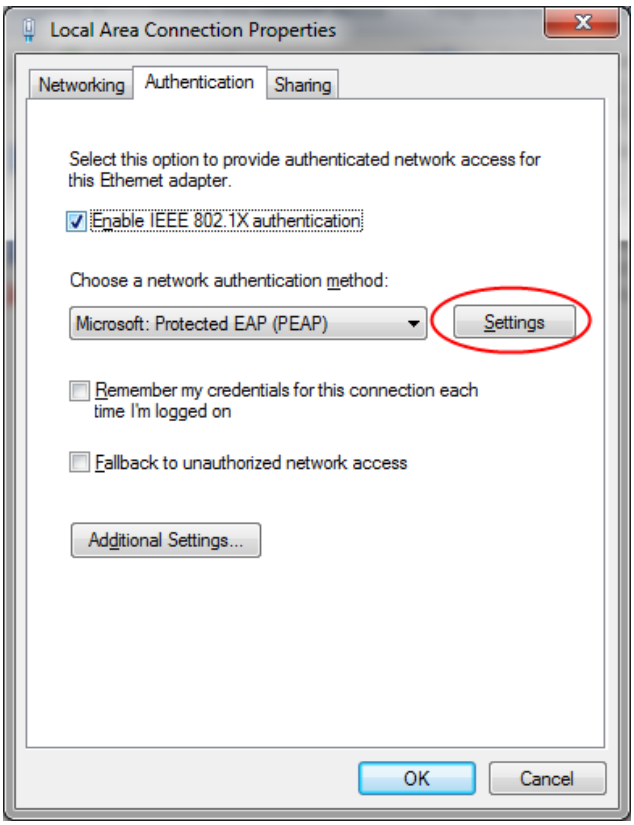
2. In the 'Open' field, type "ncpa.cpl ", and then click **OK**; the Network Connections window appears:

Figure 9-5: Local Area Connection Properties



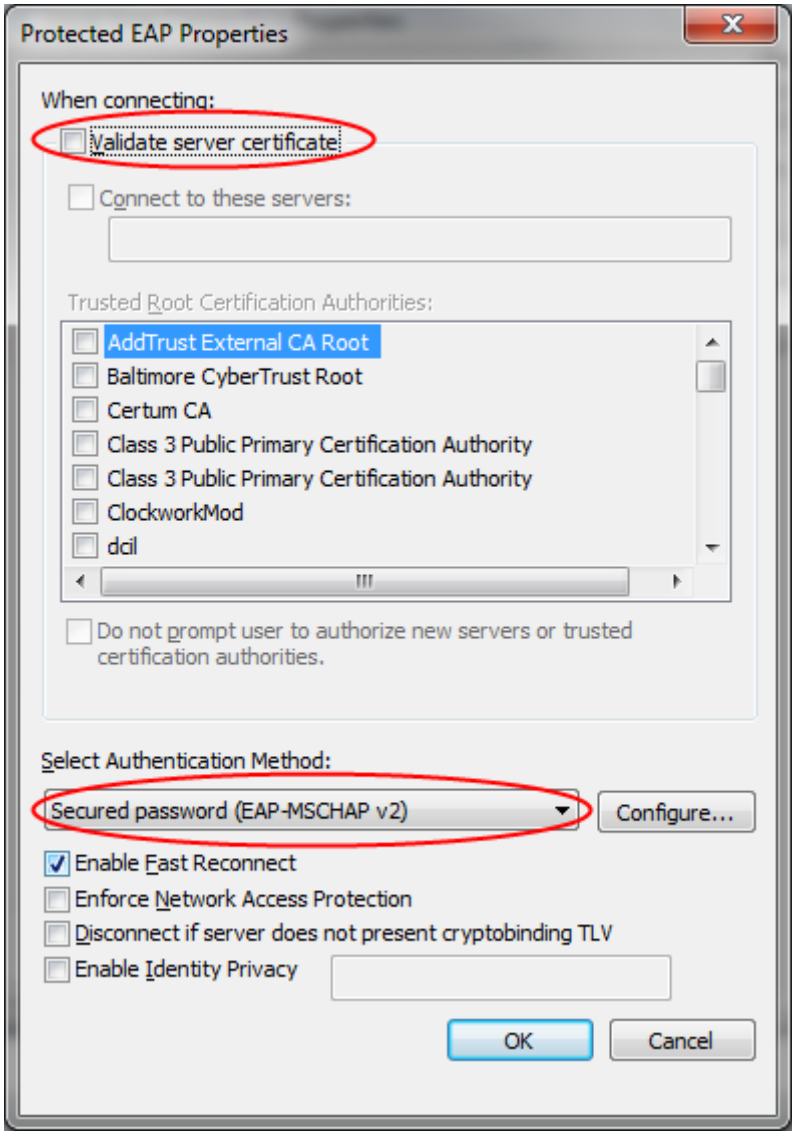
3. Right-click an interface that dot1x needs to be configured on, and then choose **Properties**; the following dialog box appears:

Figure 9-6: Local Area Connection



4. Select the 'Enable IEEE 802.1X authentication' check box.
5. Set the authentication method to **Microsoft: Protected EAP (PEAP)**.
6. Click **Settings**; the following dialog box appears:

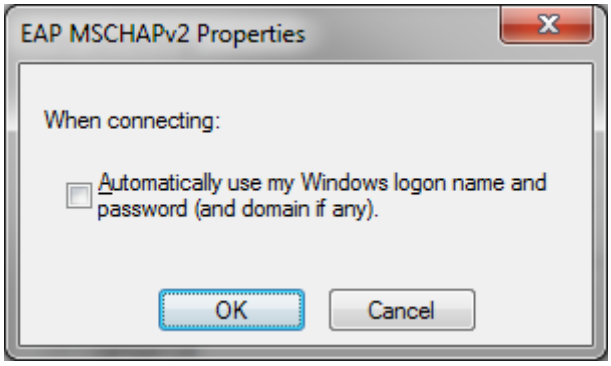
Figure 9-7: Protected EAP Properties



7. Clear the 'Validate server certificate' check box, and make sure that **Secured Password (EAP-MSCHAP v2)** is selected.

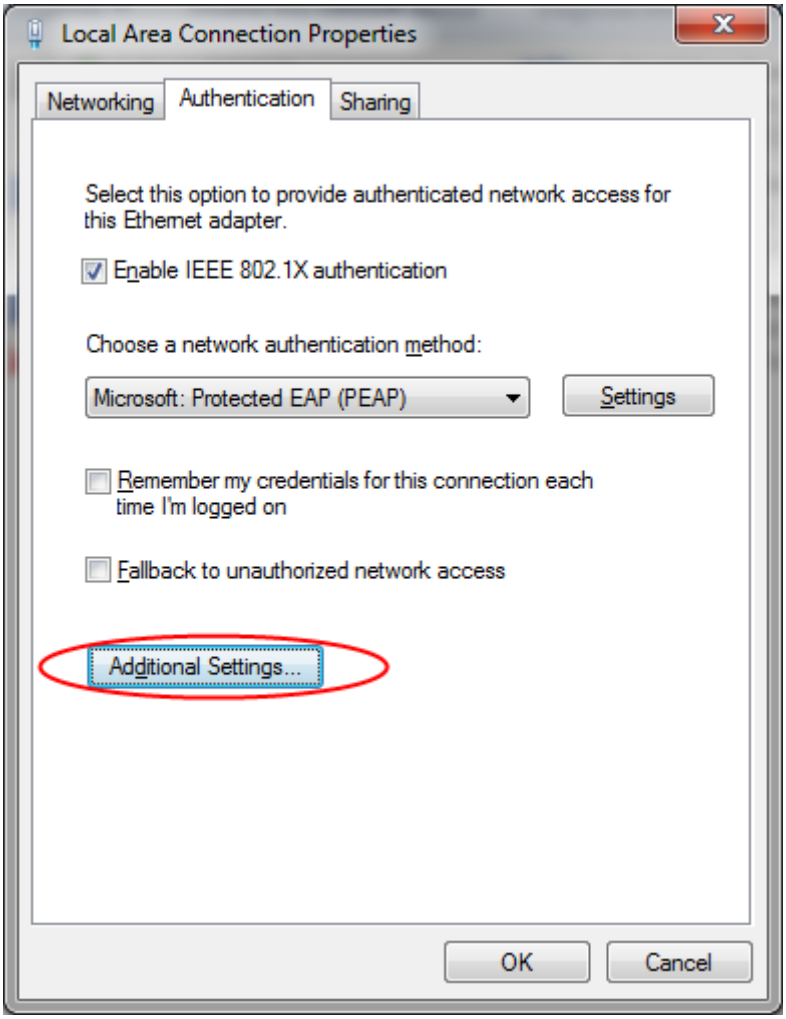
8. Click **Configure**; the following dialog box appears:

Figure 9-8: EAP MSCHAPv2 Properties



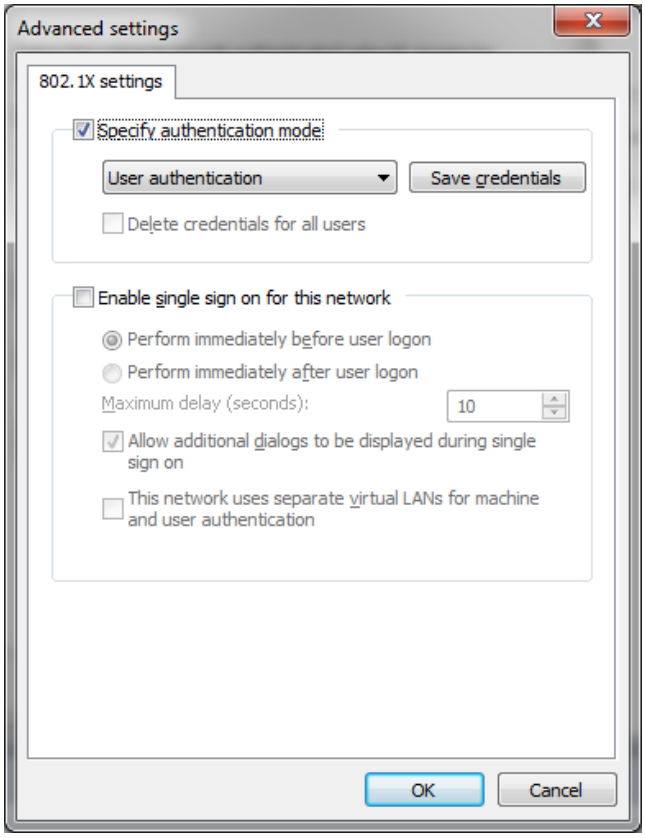
9. When internal, meaning device's, dot1x server is used, or anytime that windows logon is not used, clear the 'Automatically use my …' check box. If Windows authentication is used, select the check box.
10. Click **OK** until you're back at the **Authentication** tab in the Local Area Connection Properties window:

Figure 9-9: Authentication Tab



11. Click **Additional Settings**; the following dialog box appears:

Figure 9-10: Advanced Settings



12. Make sure that the 'Specify Authentication mode' check box is selected.
13. Select **User authentication** for user authentication. You can also enter the credentials at this step be clicking **Save credentials**.
14. Click **OK** until the interface settings close.

9.3 Example of Local Authentication Configuration

This example describes how to use the device's internal dot1x RADIUS to authenticate users:

```
# configure data
(config-data)# dot1x radius-server local
(config-data)# dot1x local-user AudioCodes password P@ssw0rd
(config-data)# dot1x lan-authentication enable
(config-data)# interface gigabitethernet 4/1
(conf-if-GE 4/1)# authentication dot1x single-host
```

Displays the dot1x connected users:

```
# show data dot1x-status
Port    Auth      State         Timeout Username
----    ----      -----         ------- --------
  1     Enabled   Forwarding          0 AudioCodes
  2     Disabled  Idle                0
  3     Disabled  Idle                0
  4     Disabled  Idle                0

#
```

This page is intentionally left blank.

10 DNS Query Randomization

The device supports DNS query source port and Query ID randomization from Version 6.8. The purpose of this feature is to prevent DNS spoofing attacks.

There are two modes of operation for DNS Query Randomization:

- **Forwarding Plan mode:** An external DNS server on the device's WAN side is advertised); only the source port is randomized.
- **DNS proxy mode:** The device is configured as a DNS server on its LAN side.

  Both the DNS Query ID and source port used on the device's WAN side are randomized. This option activates the randomization feature on all outgoing DNS queries from the device to the WAN side.

10.1 Configuration Example

This example shows how to activate the DNS query randomization feature above:

```
# configure data
(config-data)# ip dns randomization
(config-data)# exit
#
```

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
80 Kingsbridge Rd
Piscataway, NJ 08854, USA
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com

Document #: LTRT-31831