



# **Installing Avaya Mobile Video Server and Media Broker**

Release 3.2.3

Issue 1.2

July 2017

© 2016-2017 Avaya Inc.

All Rights Reserved.

### **Notice**

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

### **Documentation disclaimer**

“Documentation” means information published in varying mediums which may include product information, operating instructions and performance specifications that are generally made available to users of products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of Documentation unless such modifications, additions, or deletions were performed by or on the express behalf of Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

### **Link disclaimer**

Avaya is not responsible for the contents or reliability of any linked websites referenced within this site or Documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

### **Warranty**

Avaya provides a limited warranty on Avaya hardware and software. Refer to your sales Installing Avaya Mobile Video 3.2 Server and Media Broker

agreement to establish the terms of the limited warranty. In addition, Avaya's standard warranty language, as well as information regarding support for this product while under warranty is available to Avaya customers and other parties through the Avaya Support website: <http://support.avaya.com/helpcenter/getGenericDetails?detailId=C20091120112456651010> under the link “Warranty & Product Lifecycle” or such successor site as designated by Avaya. Please note that if You acquired the product(s) from an authorized Avaya Channel Partner outside of the United States and Canada, the warranty is provided to You by said Avaya Channel Partner and not by Avaya.

“**Hosted Service**” means an Avaya hosted service subscription that You acquire from either Avaya or an authorized Avaya Channel Partner (as applicable) and which is described further in Hosted SAS or other service description documentation regarding the applicable hosted service. If You purchase a Hosted Service subscription, the foregoing limited warranty may not apply but You may be entitled to support services in connection with the Hosted Service as described further in your service description documents for the applicable Hosted Service. Contact Avaya or Avaya Channel Partner (as applicable) for more information.

### **Hosted Service**

THE FOLLOWING APPLIES ONLY IF YOU PURCHASE AN AVAYA HOSTED SERVICE SUBSCRIPTION FROM AVAYA OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE), THE TERMS OF USE FOR HOSTED SERVICES ARE AVAILABLE ON THE AVAYA WEBSITE,

[HTTP://SUPPORT.AVAYA.COM/LICENSEINFO](http://support.avaya.com/licenseinfo)

UNDER THE LINK “Avaya Terms of Use for Hosted Services” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, AND ARE APPLICABLE TO ANYONE WHO ACCESSES OR USES THE HOSTED SERVICE. BY ACCESSING OR USING THE HOSTED SERVICE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE DOING SO (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THE TERMS OF USE. IF

YOU ARE ACCEPTING THE TERMS OF USE ON BEHALF A COMPANY OR OTHER LEGAL ENTITY, YOU REPRESENT THAT YOU HAVE THE AUTHORITY TO BIND SUCH ENTITY TO THESE TERMS OF USE. IF YOU DO NOT HAVE SUCH AUTHORITY, OR IF YOU DO NOT WISH TO ACCEPT THESE TERMS OF USE, YOU MUST NOT ACCESS OR USE THE HOSTED SERVICE OR AUTHORIZE ANYONE TO ACCESS OR USE THE HOSTED SERVICE.

**Licenses** THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, [HTTP://SUPPORT.AVAYA.COM/LICENSEINFO](http://support.avaya.com/licenseinfo), UNDER THE LINK “AVAYA SOFTWARE LICENSE TERMS (Avaya Products)” OR SUCH SUCCESSOR SITE AS DESIGNATED BY AVAYA, ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AVAYA CHANNEL PARTNER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AVAYA CHANNEL PARTNER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA CHANNEL PARTNER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Avaya grants You a license within the scope of the license types described below, with the exception of Heritage Nortel Software, for which the scope of the license is detailed below. Where the order documentation does not expressly identify a license type, the applicable license will be a Designated System License as set forth below in

Section M(i)1 or 2 as applicable. The applicable number of licenses and units of capacity for which the license is granted will be one (1), unless a different number of licenses or units of capacity is specified in the documentation or other materials available to You. “**Software**” means computer programs in object code, provided by Avaya or an Avaya Channel Partner, whether as stand-alone products, pre-installed on hardware products, and any upgrades, updates, patches, bug fixes, or modified versions thereto. “**Designated Processor**” means a single stand-alone computing device. “**Server**” means a set of Designated Processors that hosts (physically or virtually) a software application to be accessed by multiple users. “**Instance**” means a single copy of the Software executing at a particular time: (i) on one physical machine; or (ii) on one deployed software virtual machine (“**VM**”) or similar deployment.

#### **License types**

**Concurrent User License (CU).** End User may install and use the Software on multiple Designated Processors or one or more Servers, so long as only the licensed number of Units are accessing and using the Software at any given time. A “**Unit**” means the unit on which Avaya, at its sole discretion, bases the pricing of its licenses and can be, without limitation, an agent, port or user, an e-mail or voice mail account in the name of a person or corporate function (e.g., webmaster or helpdesk), or a directory entry in the administrative database utilized by the Software that permits one user to interface with the Software. Units may be linked to a specific, identified Server or an Instance of the Software.

#### **Heritage Nortel Software**

“Heritage Nortel Software” means the software that was acquired by Avaya as part of its purchase of the Nortel Enterprise Solutions Business in December 2009. The Heritage Nortel Software is the software contained within the list of Heritage

Nortel Products located at <http://support.avaya.com/LicenseInfo/> under the link "Heritage Nortel Products," or such successor site as designated by Avaya. For Heritage Nortel Software, Avaya grants Customer a license to use Heritage Nortel Software provided hereunder solely to the extent of the authorized activation or authorized usage level, solely for the purpose specified in the Documentation, and solely as embedded in, for execution on, or for communication with Avaya equipment. Charges for Heritage Nortel Software may be based on extent of activation or use authorized as specified in an order or invoice.

### **Copyright**

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, Hosted Service, or hardware provided by Avaya. All content on this site, the documentation, Hosted Service, and the product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

### **Virtualization**

The following applies if the product is deployed on a virtual machine. Each product has its own ordering code and license types. Unless otherwise stated, each Instance of a product must be separately licensed and ordered. For example, if the end user customer or Avaya Channel Partner would like to install two Instances of the same type of products, then two products of that type must be ordered.

### **Third Party Components**

"Third Party Components" mean certain software programs or portions thereof included in the Software or Hosted Service may contain software (including open source software) distributed under third party agreements ("Third Party Components"), which contain terms regarding the rights to use certain portions of the Software ("Third Party Terms"). As required, information regarding distributed Linux OS source code (for those products that have distributed Linux OS source code) and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply is available in the products, Documentation or on Avaya's website at: <http://support.avaya.com/Copyright> or such successor site as designated by Avaya. The open source software license terms provided as Third Party Terms are consistent with the license rights granted in these Software License Terms, and may contain additional rights benefiting You, such as modification and distribution of the open source software. The Third Party Terms shall take precedence over these Software License Terms, solely with respect to the applicable Third Party Components, to the extent that these Software License Terms impose greater restrictions on You than the applicable Third Party Terms.

The following applies only if the H.264 (AVC) codec is distributed with the product. THIS PRODUCT IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO (i) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (ii) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://WWW.MPEGLA.COM)

### **Service Provider**

THE FOLLOWING APPLIES TO AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS OR SERVICES. THE PRODUCT OR HOSTED SERVICE

MAY USE THIRD PARTY COMPONENTS SUBJECT TO THIRD PARTY TERMS AND REQUIRE A SERVICE PROVIDER TO BE INDEPENDENTLY LICENSED DIRECTLY FROM THE THIRD PARTY SUPPLIER. AN AVAYA CHANNEL PARTNER'S HOSTING OF AVAYA PRODUCTS MUST BE AUTHORIZED IN WRITING BY AVAYA AND IF THOSE HOSTED PRODUCTS USE OR EMBED CERTAIN THIRD PARTY SOFTWARE, INCLUDING BUT NOT LIMITED TO MICROSOFT SOFTWARE OR CODECS, THE AVAYA CHANNEL PARTNER IS REQUIRED TO INDEPENDENTLY OBTAIN ANY APPLICABLE LICENSE AGREEMENTS, AT THE AVAYA CHANNEL PARTNER'S EXPENSE, DIRECTLY FROM THE APPLICABLE THIRD PARTY SUPPLIER.

WITH RESPECT TO CODECS, IF THE AVAYA CHANNEL PARTNER IS HOSTING ANY PRODUCTS THAT USE OR EMBED THE G.729 CODEC, H.264 CODEC, OR H.265 CODEC, THE AVAYA CHANNEL PARTNER ACKNOWLEDGES AND AGREES THE AVAYA CHANNEL PARTNER IS RESPONSIBLE FOR ANY AND ALL RELATED FEES AND/OR ROYALTIES. THE G.729 CODEC IS LICENSED BY SIPRO LAB TELECOM INC. SEE [WWW.SIPRO.COM/CONTACT.HTML](http://WWW.SIPRO.COM/CONTACT.HTML). THE H.264 (AVC) CODEC IS LICENSED UNDER THE AVC PATENT PORTFOLIO LICENSE FOR THE PERSONAL USE OF A CONSUMER OR OTHER USES IN WHICH IT DOES NOT RECEIVE REMUNERATION TO: (I) ENCODE VIDEO IN COMPLIANCE WITH THE AVC STANDARD ("AVC VIDEO") AND/OR (II) DECODE AVC VIDEO THAT WAS ENCODED BY A CONSUMER ENGAGED IN A PERSONAL ACTIVITY AND/OR WAS OBTAINED FROM A VIDEO PROVIDER LICENSED TO PROVIDE AVC VIDEO. NO LICENSE IS GRANTED OR SHALL BE IMPLIED FOR ANY OTHER USE. ADDITIONAL INFORMATION FOR H.264 (AVC) AND H.265 (HEVC) CODECS MAY BE OBTAINED FROM MPEG LA, L.L.C. SEE [HTTP://WWW.MPEGLA.COM](http://WWW.MPEGLA.COM).

### **Compliance with Laws**

You acknowledge and agree that it is Your responsibility for complying with any applicable laws and regulations, including, but not limited to laws and regulations related to call recording, data privacy, intellectual property, trade secret, fraud, and music performance rights, in the

country or territory where the Avaya product is used.

### **Preventing Toll Fraud**

"Toll Fraud" is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

### **Avaya Toll Fraud intervention**

If You suspect that You are being victimized by Toll Fraud and You need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support website: <http://support.avaya.com>, or such successor site as designated by Avaya.

### **Security Vulnerabilities**

Information about Avaya's security support policies can be found in the Security Policies and Support section of <https://support.avaya.com/security>

Suspected Avaya product security vulnerabilities are handled per the Avaya Product Security Support Flow (<https://support.avaya.com/css/P8/documents/100161515>).

### **Trademarks**

The trademarks, logos and service marks ("Marks") displayed in this site, the Documentation, Hosted Service(s), and product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, its licensors, its suppliers, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation, Hosted Service(s) and product(s) should be construed as granting, by implication, estoppel, or otherwise,

any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

### **Downloading Documentation**

For the most current versions of Documentation, see the Avaya Support website: <http://support.avaya.com>, or such successor site as designated by Avaya.

### **Contact Avaya Support**

See the Avaya Support website: <http://support.avaya.com> for product or Hosted Service notices and articles, or to report a problem with your Avaya product or Hosted Service. For a list of support telephone numbers and contact addresses, go to the Avaya Support website: <http://support.avaya.com> (or such successor site as designated by Avaya), scroll to the bottom of the page, and select Contact Avaya Support.

# Contents

<b>Chapter 1: Introduction .....</b>	<b>10</b>
Purpose .....	10
<b>Chapter 2: Overview .....</b>	<b>11</b>
Components .....	11
<i>Avaya Mobile Video Server</i> .....	11
<i>Avaya Mobile Video Media Broker</i> .....	11
Recommended deployment .....	12
<b>Chapter 3: Pre-Installation .....</b>	<b>13</b>
Prerequisites .....	13
Recommended server specification .....	13
<i>Avaya Mobile Video Gateway</i> .....	13
<i>Avaya Mobile Video Media Broker</i> .....	14
Network .....	14
<i>Supported SDK client versions</i> .....	14
<i>Supported Agent Versions</i> .....	14
<i>Supported browsers for management</i> .....	15
Planning .....	15
Capacities .....	15
<i>Supported number of Audio Only calls per Avaya Mobile Video Media Broker</i> .....	16
<i>Support for 300 Audio Only calls</i> .....	17
<i>Supported number of Video calls per Avaya Mobile Video Media Broker</i> .....	17
<i>Support for 100 Video calls</i> .....	19
<i>Examples of Traffic Mix</i> .....	21
<b>Chapter 4: Installing and configuring the operating system and software .....</b>	<b>22</b>
Installation checklist .....	22
Installation prerequisites .....	23
<i>Install a supported OS</i> .....	23
<i>Post OS Installation</i> .....	24
<i>Create an administrative user</i> .....	26
<i>Avaya Mobile Video Media Broker network configuration</i> .....	26
Running the installer .....	27
<i>Required configuration</i> .....	27
<i>Configuring security</i> .....	29
<i>Preparing the installer</i> .....	30
<i>Installing Avaya Mobile Video Server software</i> .....	30
Post-installation .....	31
<i>Increase the Avaya Mobile Video Application Server JVM size</i> .....	31
<i>Install patches</i> .....	32
<i>Configuring Avaya Mobile Video Media Broker</i> .....	32
<i>Configuring additional Media Brokers</i> .....	33
<i>Configuring Call Admission Control</i> .....	34



<i>Installing the support website</i> .....	35
<i>Configuring Avaya Aura® components</i> .....	35
<i>Configure Web Application ID</i> .....	36
Configuration notes .....	37
<i>Enabling video calling on Communication Manager</i> .....	37
<i>Configuring Communication Manager VDNs</i> .....	38
<i>Configuring Communication Manager UI settings</i> .....	38
<b>Chapter 5: Security .....</b>	<b>39</b>
Changing the Avaya Mobile Video Gateway user interface password .....	39
Setting the Avaya Mobile Video Server password constraints .....	39
Enabling secure communication between Avaya Mobile Video Media Broker and Avaya Mobile Video Gateway .....	41
<i>Creating the trust certificate</i> .....	41
<i>Importing the trust certificate</i> .....	42
<i>Configuring Avaya Mobile Video Media Broker security</i> .....	42
Enabling secure SIP communication between Avaya Mobile Video Gateway and Session Manager .....	43
<i>Configuring Avaya Mobile Video Gateway</i> .....	43
<i>Configuring trust certificates</i> .....	44
Enabling secure Media between Avaya Mobile Video Media Broker and Avaya One-X® Agent.....	45
<i>Configuring the Avaya Mobile Video Media Broker</i> .....	45
<i>Configuring the Communication Manager</i> .....	46
<i>Configuring the signaling group between Communication Manager and Session Manager</i> .....	46
<i>Changing the configured HTTP/HTTPS ports</i> .....	46
<i>Configuring the Avaya Mobile Video Server HA SIP profile</i> .....	47
<i>Configuring the Load Balancer sockets</i> .....	48
Managing the Avaya Mobile Video Server SSL certificates.....	50
<i>Certificate import process</i> .....	50
Additional security .....	52
<b>Chapter 6: Upgrade.....</b>	<b>53</b>
<i>Upgrade process</i> .....	53
<i>Upgrade prerequisites</i> .....	54
<i>Installation structure</i> .....	54
<i>Upgrading the software</i> .....	54
<i>Rollback</i> .....	55
<b>Chapter 7: Administration .....</b>	<b>57</b>
Changing the Avaya Aura SIP domain .....	57
Changing the configured Session Manager.....	58
Changing the Gateway External Address .....	58
Changing the certificate expiry warning interval .....	59
<b>Chapter 8: Configuring video settings.....</b>	<b>60</b>
Typical video bandwidths .....	60
Configuring Avaya Mobile Video Gateway video .....	60
<i>Adaptive bitrate</i> .....	61



<i>Fixed bitrate</i> .....	62
Configuring Communication Manager video .....	62
<b>Chapter 9: Media Tuning</b> .....	<b>63</b>
<b>Chapter 10: Uninstall</b> .....	<b>64</b>
Uninstalling the Avaya Mobile Video Gateway .....	64
Uninstalling the Avaya Mobile Video Media Broker .....	64
<b>Chapter 11: Resources</b> .....	<b>65</b>
Documentation .....	65
Training .....	65
Support .....	66
<b>Appendix A: OS hardening</b> .....	<b>67</b>
<b>Appendix B: Glossary</b> .....	<b>77</b>

# Chapter 1: Introduction

---

## Purpose

This document details how to install and configure Avaya Mobile Video.

The primary audience for this document is anyone who installs, configures, and verifies Avaya Mobile Video. The audience includes implementation engineers, field technicians, business partners, solution providers, and customers.

# Chapter 2: Overview

Mobile Video SDK (MVSDK) components are installed on the Avaya Mobile Video Server. These components can be used to add voice and video to new and existing Web or Mobile applications.

The Avaya Mobile Video Server is the signalling component that communicates securely to web browsers and mobile apps via HTTP or HTTPS and connects those clients to Avaya one-X®Agent/Media Client endpoints using Avaya Aura infrastructure. It also controls the Avaya Mobile Video Media Broker that relays real-time media between clients inside and outside of the network. Avaya Mobile Video Media Broker secures real-time media, handles the complexities of firewall and NAT traversal and transcodes audio and video between clients.

In a typical production environment the recommended configuration is that the Avaya Mobile Video Server be installed in the internal network with HTTPS traffic being relayed to it using an Avaya Session Border Controller for Enterprise (Avaya SBCE). The Avaya SBCE is installed in the DMZ between the internal network and the Internet to ensure that only appropriate traffic is being sent to the Avaya Mobile Video Server. The Avaya Mobile Video Media Broker would also be installed in the DMZ and receive media traffic directly from the outside firewall, sending it on to internal endpoints.

---

## Components

---

### Avaya Mobile Video Server

Avaya Mobile Video includes the Avaya Mobile Video Server, which removes the complexity in the signaling between the clients and Avaya Session Manager so that the two can communicate together seamlessly. The Avaya Mobile Video Server communicates with the client using the TCP-based WebSockets protocol, providing a standardized way for the server to send content to the client without being solicited, and allowing for messages to be passed back and forth while keeping the connection open.

The primary functions of the Avaya Mobile Video Server are to:

- Provide signaling conversion between the client and Avaya Session Manager.
- Only allow clients that have been authorized by the Web application to create sessions. The Web Application will create session tokens for permitted users, and the Avaya Mobile video Server will only allow clients with a session token to connect and make calls.

---

### Avaya Mobile Video Media Broker

The Avaya Mobile Video Media Broker is responsible for media transcoding and SRTP routing between the client applications and the SIP network. Routing is configured based on the SDP passing through the Avaya Mobile Video Server. For communication into the enterprise, its role is to simplify and limit the SRTP into a form that is supported by the users' devices. For communication

going to the client application, the SRTP is augmented to add the WEBRTC-compliant features that are required.

The primary functions of the Avaya Mobile Video Media Broker are the following:

- Convert between client SRTP streams and SRTP/RTP streams compatible with SIP entities.
- Tailor the video frame rate and/or resolution for client capabilities.
- SRTP signaling encryption.

## Recommended deployment



For more detail on the recommended deployment and the roles each of the elements perform, see *Avaya Mobile Video Planning and Security Reference*.

### Note:

- This release of the Avaya Mobile Video Server does not support High Availability.
- The internal firewall is optional in the Avaya Mobile Video Server deployment.

# Chapter 3: Pre-Installation

---

## Prerequisites

Before commencing installation, ensure that you complete the *Deployment planning* section in *Avaya Mobile Video Planning and Security Reference*.

- A standard Avaya Aura® installation must be available, including Session Manager.
- Avaya Aura® Communication Manager is supported. The Avaya Media Client software is only supported on one Communication Manager system in a deployment.
- Avaya one-X® Agent and Avaya Media Client Application must be installed. These are available at the Avaya Mobile Video Server release location.
- An Avaya Session Border Controller for Enterprise (Avaya SBCE) must be installed in two-wire deployment mode, with one interface on the internal (Enterprise) network and the other on the external (public) network, protected by an external firewall .
- Two servers must be available, one for the Avaya Mobile Video Gateway and one for the Avaya Mobile Video Media Broker for the typical deployment scenario. Extra Media Broker servers can then be added to support the required call capacity. See [Recommended server specification](#).
- A Web Server needs to be available that will host the Web Application responsible for authorizing users and distributing session tokens.

For the latest and most accurate compatibility information for Avaya Mobile Video, go to:

<https://support.avaya.com/CompatibilityMatrix/Index.aspx>

---

## Recommended server specification

This product can be installed on a virtualized environment, for example VMWare.

---

### Avaya Mobile Video Gateway

2.6 Ghz 8 Core Processor

16 GB RAM

50 GB hard disk free space

One 1 GB Network Interface Card

---

## Avaya Mobile Video Media Broker

2.6 Ghz 8 Core Processor

16 GB RAM

50 GB hard disk free space

Three 1 GB Network Interface Cards

- One for the management traffic
- One for the external traffic (to the SDK Client endpoints)
- One for the internal traffic (to the H.323 Avaya one-X® Agent endpoints)

---

## Network

To support the transfer of RTP, Avaya Mobile Video Media Brokers require the following bandwidth:

- Video calls require up to 2 Mbit/s, upload and download. For more information see [Capacities](#).
- Audio calls require up to 100 Kbit/s. This depends on the codec used.

**Important:** Avaya Mobile Video calls made over 3G and 4G/LTE mobile networks might experience poor video quality, poor audio quality, and dropped calls caused by low mobile network bandwidth. Avaya does not provide support to troubleshoot 3G and 4G/LTE mobile network issues that might cause problems with Avaya Mobile Video calls.

Network switches used in the deployment should not be utilized above 50%.

---

## Supported SDK client versions

- Android smartphone and tablet devices using ARM architecture running 5.0 and later software.
- [iPhone 5c and above, and iPad air and above, running IOS 8.0 and later software.](#)
- Chrome browser on Microsoft Windows, running stable version 49 and later software.

---

## Supported Agent Versions

- Avaya one-X® Agent (H.323)
- Avaya Media Client

---

## Supported browsers for management

Avaya Mobile Video can be administered using either the Avaya Mobile Video Application Server Management interface or the Avaya Mobile Video Gateway Administration interface. Recent versions of all major browsers can be used, for example:

- Microsoft Internet Explorer
- Google Chrome
- Mozilla Firefox
- Apple Safari

---

## Planning

#	Action	Notes	✓
1	Obtain Red Hat 6.8 or 6.9 Minimal Installation Media and install to Avaya Mobile Video Gateway and Avaya Mobile Video Media Broker servers		
2	Apply Red Hat licenses to installation to allow updates, or obtain the software disc to allow installing updates from the disc.		

---

## Capacities

**Note:** All figures stated in this section are based on the Avaya Mobile Video Gateway and Avaya Mobile Video Media Broker specifications detailed in [Recommended server specification](#).

This deployment provides support for:

1. 100 simultaneous video calls or 300 simultaneous audio calls.
2. A mixture of video and audio calls based on the above figures.

Only a single Avaya Mobile Video Gateway is supported in this release, and as such no redundancy or High Availability is supported.

The number of Avaya Mobile Video Media Brokers required for the deployment depends on the following factors:

1. The type of the call:
  - a. **Video Pass-through**—this is a call where both endpoints in the call are using the same video codec.
  - b. **Video Transcoded**—this is where the endpoints in a call are using a different video codec. In this situation the Avaya Mobile Video Media Broker has to convert the one video stream in to the other codec type. This is done for both legs of the call.



- c. **Audio Pass-through**—this is a call where both endpoints in the call are using the same audio codec and the call has no video.
  - d. **Audio Transcoded**—this is where the endpoints in a call are using a different audio codec and the call has no video. In this situation the Avaya Mobile Video Media Broker has to convert the one audio stream in to the other codec type. This is done for both legs of the call.
2. The resolution (for video calls only).
  3. The bit rate used or available.

## Supported number of Audio Only calls per Avaya Mobile Video Media Broker

The following tables show the number of calls that can be achieved by a single Avaya Mobile Video Media Broker when processing audio only calls. These figures assume all calls are audio only calls and there are no video calls.

### Pass-through

Codec	# Calls
g.711 (64 Kbps)	300

### Transcoded (g.711 to Opus)

56
----

The figures in each cell above denote the maximum number of calls supported if all the calls were of the same type.

If a mixture of pass-through calls and transcoded calls are received, the figures will be different. This is because pass-through calls and transcoded calls are not equal. Transcoded calls are more resource intensive; therefore more pass-through calls can be made per transcoded call.

Below are some examples of the number of calls expected when a mixture of transcoded audio only calls and pass-through audio only calls are received.

Call Type	Number of supported calls										
Audio Only Pass through calls (g.711)	300	270	240	210	180	150	120	90	60	30	0
Audio Only Transcoded calls (g.711 -> Opus)	0	5	11	16	22	28	33	39	44	50	56

---

## Support for 300 Audio Only calls

Based on the figures in the tables above, the table below shows the number of Avaya Mobile Video Media Brokers needed to support 300 audio only calls at each of the different configurations.

### Pass-through – Total number of Media Brokers required for 300 call support

Codec	# Media Brokers
g.711 (64 Kbps)	1

### Transcoded - Total number of Media Brokers required for 300 call support

9
---

The tables below give some examples of the Media Brokers required given different mixtures of call traffic:

<b>g.711 Pass through calls</b>	300	270	240	210	180	150	120	90	60	30	0
<b>Transcoded calls</b>	0	30	60	90	120	150	180	210	240	270	300
<b>Number of Media Brokers Required</b>	1	2	2	3	4	4	5	5	6	6	6

---

## Supported number of Video calls per Avaya Mobile Video Media Broker

The following tables show the number of calls that can be achieved by a single Avaya Mobile Video Media Broker at various video resolutions and bit rates. These figures assume all calls are video calls and there are no audio only calls. The resolutions in the tables below are the only supported resolutions for this deployment.

**Important:** 768 Kbps is the maximum supported bandwidth for this product.

### Pass-through

Resolution	450 Kbps	620 Kbps	768 Kbps
320x240@30 fps	66	50	
640x480@30 fps	66	50	50
1280x720@30 fps		50	50

### Transcoded

Resolution	450 Kbps	620 Kbps	768 Kbps
320x240@30 fps	13	12	
640x480@30 fps	10	7	7
1280x720@30 fps		6	6

The figures in each cell above denote the maximum number of calls supported at that specified bandwidth and resolution if all the calls were of that identical bandwidth and resolution. For example, at 640x480@30 fps and 768 Kbps, 50 pass-through calls can be made as long as no transcoded calls are received.

If a mixture of pass-through calls and transcoded calls are received, the figures will be different. This is because pass-through calls and transcoded calls are not equal. Transcoded calls are more resource intensive; therefore more pass-through calls can be made per transcoded call.

Below are some examples of the number of calls expected when a mixture of transcoded calls and pass-through calls are received.

#### 320x240@30 fps – 450 Kbps

Call Type	Number of supported calls							
Pass through calls	66	56 to 60	51 to 55	41 to 45	31 to 35	21 to 25	11 to 15	0
Transcoded calls	0	1	2	4	6	8	10	13

#### 320x240@30 fps – 620 Kbps

Call Type	Number of supported calls							
Pass through calls	50	42 to 45	38 to 41	30 to 33	21 to 25	13 to 16	5 to 8	0
Transcoded calls	0	1	2	4	6	8	10	12

### 640x480@30 fps – 768 Kbps

Call Type	Number of supported calls							
Pass through calls	50	36 to 42	29 to 35	22 to 28	15 to 21	8 to 14	1 to 7	0
Transcoded calls	0	1	2	3	4	5	6	7

For more information on how pass-through and transcoded calls affect each other, and how to configure these values in the deployment, see [Configuring Call Admission Control](#).

## Support for 100 Video calls

Based on the figures in the tables above, the table below shows the number of Avaya Mobile Video Media Brokers needed to support 100 calls at each of the different configurations.

**Important:** 768 Kbps is the maximum supported bandwidth for this product.

### Pass-through – Total number of Media Brokers required for 100 call support

Resolution	450 Kbps	620 Kbps	768 Kbps
320x240@30 fps	2	2	
640x480@30 fps	2	2	2
1280x720@30 fps		2	2

### Transcoded – Total number of Media Brokers required for 100 call support

Resolution	450 Kbps	620 Kbps	768 Kbps
320x240@30 fps	8	9	
640x480@30 fps	10	15	15
1280x720@30 fps		17	17

The figures in each cell above denote the maximum number of Media Brokers required if all the calls were of that identical bandwidth and resolution. For example, at 640x480@30 fps and 768 Kbps, 15 Media Brokers are required if all 100 calls are transcoded.

If a mixture of pass-through calls and transcoded calls are received, then the number of Media Brokers required will depend on the mixture of calls received. The tables below give some examples of the Media Brokers required given different mixtures of call traffic:

**320x240@30 fps – 450 Kbps**

Pass through calls	100	90	80	70	60	50	40	30	20	10	0
Transcoded calls	0	10	20	30	40	50	60	70	80	90	100
Number of Media Brokers Required	2	3	3	4	4	5	6	6	7	8	8

**320x240@30 fps – 620 Kbps**

Pass through calls	100	90	80	70	60	50	40	30	20	10	0
Transcoded calls	0	10	20	30	40	50	60	70	80	90	100
Number of Media Brokers Required	2	3	4	4	5	6	6	7	8	8	9

**640x480@30 fps – 768 Kbps**

Pass through calls	100	90	80	70	60	50	40	30	20	10	0
Transcoded calls	0	10	20	30	40	50	60	70	80	90	100
Number of Media Brokers Required	2	4	5	6	7	9	10	11	12	14	15

---

## Examples of Traffic Mix

The table below shows an example of what calls can be expected for each of the different call types for a single Media Broker instance:

Video Transcoded	Audio Transcoded	Video Pass-through	Audio Pass-through
9	5	0	0
9	3	3	0
5	20	9	1
5	20	4	24
5	4	28	1
5	5	16	49
1	40	3	42
1	30	14	45
1	20	35	3
1	20	19	76

The table below shows an example of what calls can be expected for each of the different call types for the supported solution:

Video Transcoded	Audio Transcoded	Video Pass-through	Audio Pass-through
149	5	0	3
149	4	1	4
100	32	250	192
100	150	150	14
70	200	250	192
52	513	30	55

# Chapter 4: Installing and configuring the operating system and software

This chapter contains the steps you must perform to:

- Configure the operating system (OS) for the Avaya Mobile Video Server installation.
- Install the Avaya Mobile Video Server software.
- Configure the Avaya Mobile Video Server software.

---

## Installation checklist

Action	Notes	✓
Install and configure software on all servers, which includes:		
Install the OS		
Connect to the network		
Configure DNS		
Deploy security updates		
Install required packages		
Install Java	Record the installation directory here:	
Configure open file limit		
Disable SELinux		
Create a user for the Avaya Mobile Video Server		
Create an administrative user		



Run the Avaya Mobile Video Server installer	Run the installer, providing the details prepared in the section <a href="#">Required configuration</a> .	
Install all applicable patches	Install any patches included in the \Patches folder of the downloaded AvayaMobileVideox.x.x.zip file.	
Configure the Avaya Mobile Video Media Broker		
Configure Avaya Aura®		
Enable secure communication between Avaya Mobile Video Media Broker and Avaya Mobile Video Gateway (optional)		
Enable secure communication between Avaya Mobile Video Gateway and Session Manager (optional)		
Enable secure Media between Avaya Mobile Video Media Broker and Avaya one-X® Agent (optional)		
Configure a Web Application ID		

---

## Installation prerequisites

---

### Install a supported OS

**Important:** Perform this on all servers in the deployment.

The Avaya Mobile Video Server supports the 64-bit x86 Red Hat Enterprise Linux Advanced Platform version 6.8 or 6.9. The **Minimal** package group must be selected during the Red Hat installation. After installation, ensure that the Red Hat subscription is applied to the server, as several packages may need to be installed from the repository.

**Important:** Red Hat Enterprise Linux 7.x is not supported for this release of the Avaya Mobile Video Server.

---

## Post OS Installation

**Important:** Perform these procedures on all servers in the deployment.

After installation of RHEL 6.8 or 6.9, there are several steps that need to be undertaken prior to installing Avaya Mobile Video Server.

### Stop the iptables service

Before installation is attempted, you must ensure the `iptables` service is stopped. To do so, run the command:

```
service iptables stop
```

### Configure DNS

Avaya Mobile Video Server requires that the Gateway and Media Broker hostnames can be resolved.

At a minimum, the Avaya Mobile Video Server Gateway and Media Broker must have their own FQDN added to their respective `/etc/hosts` file. To do this:

1. Open the `/etc/hosts` file.
2. Add an entry for the server, for example `192.168.8.65 mobilevideomb.avaya.com`.
3. Ping the hostname again to check that it now is resolved.

If hostnames are used during the installation process, the FQDN for the Avaya Mobile Video Gateway and Media Broker servers should also be configured on the DNS server being used in the deployment. To check if it has been configured correctly:

1. Run the `hostname` command. This displays the configured hostname provided during installation, for example `mobilevideo.avaya.com`.
2. Ping the hostname to see if it is resolvable, for example:  

```
ping mobilevideo.avaya.com
```
3. If the hostname cannot be resolved, correct the configuration of the DNS server being used.
4. Ping the hostname again to check that it is now is resolved.

### Security updates

To install the latest security updates, run the following command:

```
yum update
```

### Install required packages

The following packages need to be installed prior to Avaya Mobile Video Server installation:

- `pango`
- `libXv`

- libxml2
- tcpdump
- wget
- unzip

To test if the packages are already installed, run the following command:

```
rpm -qa | grep "pango\|libXv\|libxml2\|unzip\|tcpdump\|wget"
```

If a package is missing, install it using the yum command, for example `yum -y install pango`

## Install Java

Avaya Mobile Video Server requires 64-bit Oracle JDK 8 update 131 to run. To install Java, perform the following steps:

1. Run the install command, for example `rpm -ivh jdk-8u131-linux-x64.rpm`
2. Run `java -version` to check java has been installed. The output from this shows the version, for example:

```
[root@avayagw ~]# java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b13, mixed mode)
```

Java is installed into `/usr/java` and a soft link `/usr/java/latest` is created (this is used later during the Avaya Mobile Video Server install).

Java also needs to be added to the path for all users. To do this, create the `/etc/profile.d/java.sh` file, specifying the following content:

```
export JAVA_HOME=/usr/java/latest
export PATH=$PATH:$JAVA_HOME/bin
```

Change the Java path in the above command (`/usr/java/latest`) as necessary depending on the location of the Java install. Apply the changes to the current session by running the command:  
`source /etc/profile.d/java.sh.`

## Configure open file limits

The number of open files required by the Avaya Mobile Video Gateway and Avaya Mobile Video Media Broker components must be configured at 30000.

1. To see the current maximum number of open files, run the following command: `ulimit -n`
2. To change the number of open files, open `/etc/security/limits.conf` and add or change the setting for the open file limit, for example `* - nofile 30000`
3. To confirm that the setting has changed, log out and log back in again. Run `ulimit -n` again to check the maximum number of open files.

Note: If the open file limit is changed at a later stage, the Mobile Video service (Gateway and Media Broker) will have to be restarted to pick up the change.

## Disable SELinux

To disable SELinux, add the following line to `/etc/selinux/config`

```
SELINUX=disabled
```

After changing this setting, the server needs to be rebooted. Once restarted, run the following command which should now show SELinux as disabled:

```
getenforce
```

## User configuration

The Mobile Video Server installer must be executed by the root user. During the installation, the installer is requested to enter the name of the OS user that will run the Mobile Video Server processes. This user cannot be the root user. A specific user should be created to run the Mobile Video Server processes. Create this user if necessary, for example:

```
adduser amvsuser
```

**Note:** Do not assign this user a password. This user is for running the Avaya Mobile Video Server services only, and does not need login permission. Remove login by running the following command:

```
usermod -s /sbin/nologin amvsuser
```

After creating the user, the following line needs to be added to `/etc/security/limits.conf`

```
<user> - nproc 8192
```

Where `<user>` is the user that was created.

---

## Create an administrative user

**Important:** Perform this on all servers in the deployment.

The Avaya Mobile Video Server installation process locks down the server, removing SSH access to root. Before running the install, an administrative user should be created to ensure the server can still be accessed remotely and administered when necessary. This administrative user should be created with the name 'admin'.

---

## Avaya Mobile Video Media Broker network configuration

In order to communicate with an external WebRTC Client, the Avaya Mobile Video Media Broker network configuration must be configured to route traffic correctly:

- Traffic destined for the Internal network should be routed to the interface configured on the Internal network.
- Traffic destined for the External network should be routed to the interface configured on the External network.
- Management Traffic should be routed to the interface configured to allow communication with the Avaya Mobile Video Gateway.

- All other/public service traffic should be routed to the default external gateway.

See the following example routing configuration:

```
[root@mb-a ~]# route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
172.135.3.0      0.0.0.0         255.255.255.0   U        0      0        0 eth0
10.10.10.10      0.0.0.0         255.255.255.255 UH       0      0        0 eth1
10.10.10.0       0.0.0.0         255.255.255.0   U        0      0        0 eth2
0.0.0.0          172.135.3.97    0.0.0.0         UG       0      0        0 eth0
```

In this example:

- Management Traffic destined for the Avaya Mobile Video Gateway (10.10.10.10) is routed to eth1
- Internal traffic (10.10.10.x) is routed to eth2
- External traffic (172.135.3.x) is routed to eth0 (the default external gateway)
- All other traffic is routed to eth0 (the default external gateway)

## Running the installer

### Required configuration

**Important:** Configure these parameters on all servers in the deployment.

The following configuration items are required when you run the installer.

Parameter	Your value
<b>JDK Path</b> Avaya Mobile Video Server requires java to run. This should have been installed when configuring the installation prerequisites. Enter the path to the JDK installation when prompted, for example /usr/java/latest	
<b>IP Address</b> The IP address of the network interface to be used by the Avaya Mobile Video Application Server or Avaya Mobile Video Media Broker. This should be a valid, configured IP Address on the RHEL server that the installation is being deployed to.	
<b>Cluster Address</b> The DNS resolvable fully qualified domain name that the Avaya Mobile Video Application Server is contactable on. This is the domain that clients will use to connect to the cluster.	

Parameter	Your value
<b>OS User</b> Avaya Mobile Video Server requires a user to be specified which launches the Avaya Mobile Video Application Server and Avaya Mobile Video Media Broker processes. This user is created during the User configuration step in the Post OS Installation steps.	
<b>Avaya Mobile Video Server Only Configuration</b>	
<b>Administration User Credentials</b> In order to administer the Avaya Mobile Video Application Server, an administration account must be created. Choose the username and password to use for the administration account. The default user name and password are: <code>administrator/administrator</code> <b>Security Alert:</b> Change the default password immediately after your first login. The system will not be secure if you do not change the password.	
<b>Controlled Domain</b> This is the SIP domain that the Avaya Mobile Video Gateway manages—any requests to this domain are serviced by the gateway. For Avaya Mobile Video Server, all calls are routed to Session Manager, and none are serviced by the gateway, so this should be set to the IP Address or FQDN on the RHEL server that Avaya Mobile Video Server is installed on.	
<b>Gateway External Address</b> This specifies the IP address or hostname that the client uses to create the WebSocket. If an Avaya SBCE is used in the deployment, this address is the external address, otherwise it is the address of the IP Address or FQDN on the RHEL server that the Avaya Mobile Video Server is installed on.	
<b>Use secure connection</b> This specifies whether the WebSocket connection should be secure (HTTPS) or non-secure (HTTP). If a reverse proxy is used in the deployment, this refers to the external connection to the reverse proxy.	

Parameter	Your value
<b>Gateway External Port</b> This specifies the port that the client uses to create the WebSocket. If a reverse proxy is used in the deployment, this address is a port on the reverse proxy, otherwise it is a port on the server running Avaya Mobile Video Media Broker.	
<b>Session Manger Address</b> Avaya Mobile Video Server routes all calls to the Avaya Aura Session Manager. This is the IP address of the Session Manager SIP entity.	
<b>Avaya Aura SIP Domain</b> The SIP domain for Avaya Aura. The Avaya Mobile Video Server routes any inbound calls to this domain to the Session Manager.	

---

## Configuring security

**Important:** Configure these parameters on all servers in the deployment.

Towards the end of the install, you are given the option to secure the operating system on which you are installing the Avaya Mobile Video Server. When selecting **Yes** you will be required to configure the following settings:

Parameter	Your value
<b>Remote Login Message</b> This is the message that is shown to users before attempting to log in to the OS from a remote machine	
<b>Local Login Message</b> This is the message that is shown to users before attempting to log in to the OS from the local machine	
<b>Login Success Message</b> The message displayed to a user when they successfully login.	



---

## Preparing the installer

Copy the `MobileVideoSDK-x.x.x.tar.gz` to the RHEL server in a temporary directory, ready for install.

**Note:** The installer *must* be run as root.

1. Run the following command to unzip the file:

```
tar -xzvf MobileVideoSDK-X.X.X.tar.gz
```

2. Run the following command to start the installation:

```
./install.sh
```

---

## Installing Avaya Mobile Video Server software

The first installation step is to choose the components to install. The installer allows installation of:

- Avaya Mobile Video Gateway
- Avaya Mobile Video Media Broker

You are prompted to answer a number of questions during the installation process where you provide the information gathered in the table in [Required configuration](#). Some questions relate only to the Avaya Mobile Video Gateway; others are only required when installing Avaya Mobile Video Media Broker—you are only presented with questions relevant to your product selection.

Once the installation is complete, the installer prompts whether to secure the Operating System. This process performs any necessary tasks to secure the Operating System so that potential vulnerabilities are resolved, for example:

- `iptables` is configured to only allow traffic on authorized ports to prevent DoS.
- Deny root login via SSH.
- Disable insecure network services.
- Disable TCP and ICMP timestamps.

It is recommended that you secure the operating system on both Avaya Mobile Video Gateway and Avaya Mobile Video Media Broker. For more information on what the O/S Hardening does, see [Appendix A: O/S hardening](#).

If you choose not to secure the operating system, this can be done at a later date. To do this:

1. Navigate to the directory that the Avaya Mobile Video Server installer was extracted to, or extract the installer again if it has been deleted after initial installation.
2. Execute the script that secures the O/S by running the command `./configure-os.sh`
3. Follow the on-screen commands, and provide the necessary input. For input reference, see [Security configuration](#). After inputting all the required information, the O/S is secured.
4. To test the secured installation, run the following command:

```
./test-os.sh
```

---

## Post-installation

---

### Increase the Avaya Mobile Video Application Server JVM size

The JVM settings dictate how much memory is allocated to the Java processes for the Avaya Mobile Video Application Server. A default Avaya Mobile Video Application Server installation has the JVM parameters set to the following:

```
-XX:PermSize=256m -XX:MaxPermSize=256m -Xms256m -Xmx512
```

512 MB may not be enough for some scenarios, for production systems it is recommended to increase this to at least 2048 MB.

Edit `/opt/avaya/awmvs/<version>/awmvs/domain/configuration/domain.xml`

Set the heap size to 1024 and 2048 as follows:

```
<server-groups>
  <server-group name="main-server-group" profile="ha">
    <jvm name="main-jvm">
      <heap size="1024m" max-size="2048m"/>
      <permgen size="256m" max-size="256m"/>
      <jvm-options>
        <option value="-server"/>
        <option value="-XX:+UseG1GC"/>
        <option value="-XX:MaxGCPauseMillis=50"/>
        <option value="-XX:+HeapDumpOnOutOfMemoryError"/>
        <option value="-XX:HeapDumpPath=./heapdump_as.hprof"/>
      </jvm-options>
    </jvm>
  </server-group>
</server-groups>
```

Then restart the Avaya Mobile Video Server:

```
service awmvs restart
```

The settings can be confirmed with:

```
ps -ef | grep appserver

root      8342   8272  44 13:15 pts/0      00:00:29 /opt/java/bin/java
-D[Server:appserver-awmvs] -XX:PermSize=256m -XX:MaxPermSize=256m
-Xms1024m -Xmx2048m
```

---

## Install patches

**Important:** Install patches on all servers in the deployment.

Consult the `\Patches\ReadMe` file in the downloaded `AvayaMobileVideox.x.x.zip` bundle for any patch files. Install the patches on the Gateway and/or Media Broker Server as instructed in the `ReadMe` file.

---

## Configuring Avaya Mobile Video Media Broker

Before configuring Avaya Mobile Video Media Brokers, the number of WebRTC and SIP ports required must be determined.

### WebRTC ports

These are the ports used for sending media traffic to the browser side (external). These ports need to be configured on the external firewall.

Five WebRTC ports must be configured per Media Broker, for example 16000 to 16004.

### SIP ports

These are the ports used for sending media traffic to the SIP side (internal).

Enough ports should be allocated for the supported number of calls on the system. 8 ports should be allocated for each supported call. For example, to support 100 calls, 800 ports should be configured.

If not enough ports are configured for the supported number of calls, the system becomes unstable and experiences call failures.

The start port in the range must be an even number and the end port in range should be an odd number. This is because media should be sent over 2 consecutive pairs of port. The RTP is sent on an even-numbered port, and the RTCP is sent on the next (odd-numbered) port. This complies with RFC 3550.

## Adding Avaya Mobile Video Media Brokers

The Avaya Mobile Video Media Broker must be configured with three network interface cards (NICs). This allows management, external traffic, and internal traffic to each be serviced by an individual NIC.

To configure Avaya Mobile Video Media Broker after initial installation:

1. Log on to the Avaya Mobile Video Gateway Administration Console at:

`https://<GatewayIPAddress>:8443/web_plugin_framework/webcontroller`

**Security Alert:** Change the default password immediately after your first login. The system will not be secure if you do not change the password.

2. Select **Gateway > Media Brokers** then click **Add Record**.
3. Under **General Configuration**, set the **Control Address** to the internal IP address of the Avaya Mobile Video Media Broker.

4. Retain the default values for all other items in the General Configuration section.
5. Under **SIP Network**, click **Add**.
  - a. Set the **Local Address CIDR** to the internal address of the Avaya Mobile Video Media Broker with a /32 to indicate just 1 address, for example 192.168.1.1/32.  
If using a single NIC, this can be set to **all**.
  - b. Set the **Start Port Range** to the low value, for example, 17000.
  - c. Set the **Finish Port Range** to the high value, for example, 17799 for 100 call support.
  - d. Click **Submit**.
  - e. Under **WebRTC Client** click **Add**.
  - f. Set the **Source Address CIDR** so that it matches the Avaya SBCE internal address, for example 192.168.1.2/32.  
Another valid value for this is **all** to indicate that the following IP and port apply to all source WebRTC clients.
  - g. Click **Submit**.
6. Click the symbol alongside all to add the address and port details.

**Important:** The WebRTC Client configuration references the customer/external side of the Media Broker and is required so that the Media Broker can deal with scenarios where there is Network Address Translation, and to allow the media from the customer device on the internet to reach the external facing interface on the Media Broker inside the DMZ.

If there is a firewall in place on the external side of the Media Broker then the Public address will be that firewall external IP address, while the Local address will be the IP address of the “external facing” interface on the Media broker.
7. Under **RTP Public and Local Port** click **Add**.
  - a. Set the **Public Address**. This is the Public IP address (external side of the firewall) for the external side of the Media Broker.
  - b. Set the **Public Port** to the required value, for example, 16000.
  - c. Set the **Local Address** to the IP address of the “external facing” interface on the Media Broker.
  - d. Set the **Local Port** to the required value, for example, 16000.
8. Repeat step 7 above four more times to add in the remaining WEBRTC side ports (there must be five in total), setting the **Public Port** and **Local Port** to the other port numbers, for example, 16001 to 16004.
9. Click **Submit**.
10. Click **Save**.

---

## Configuring additional Media Brokers

To add more than one Avaya Mobile Video Media Broker:

1. Install the Avaya Mobile Video Media Broker on the new server
2. Follow the Configuring Avaya Mobile Video Media Broker section above for the new Media Broker

When there is more than one Avaya Mobile Video Media Broker in a deployment, the call load is distributed among the servers equally.

**Note:** A separate range of ports is required for each Avaya Mobile Video Media Broker. The SIP ports defined do not have to be different from another Media Broker’s SIP ports as each port range is defined per Media Broker.

---

## Configuring Call Admission Control

Each Media Broker, based on the specification in the [Recommended server specification](#) section, can support a specific number of Video and Audio calls, each of which can be either pass-through or transcoded.

Call Admission Control is designed to protect an Avaya Mobile Video Media Broker against overloading when one is being selected to handle a new call.

**Note:** Call Admission Control is *not* enabled by default, as the properties are not set.

When enabled, and an Avaya Mobile Video Media Broker is deemed unable to handle another call, the Load Balancer attempts to select another Media Broker—this introduces the risk that a new call is rejected due to no Media Brokers being available.

Call Admission Control allows the definition of call limits for an individual Avaya Mobile Video Media Broker. Call Admission Control is configured from the Media Brokers page by editing or adding a media broker record.

This feature works by setting the maximum allowed number of calls for a given type (Video or Audio Only, pass-through or transcoded) and then working out the allowed combinations based on these maximum values.

### Allowed call combinations

Based on the tables in the Capacities section, if you require a video call resolution of 640x480@30 fps and have a bandwidth of 450kbps, you need to set the CAC values on each Media Broker as follows:

- Max Audio Video Pass-through Calls = 66
- Max Audio Video Transcodable Calls = 10

The settings for audio only calls should also be set as follows:

- Max Audio Only Pass-through Calls = 300
- Max Audio Only Transcodable Calls = 56

With this configuration you can expect to achieve:

- 66 concurrent pass-through video calls if there are no other call types received/ongoing.
- 10 concurrent transcoded video calls if there are no other call types received/ongoing.
- 300 concurrent pass-through audio only calls if there are no other call types received/ongoing.
- 7 concurrent transcoded audio only calls if there are no other call types received/ongoing.

### Setting the CAC limits

To set the CAC limits, log on to the Avaya Mobile Video Gateway Administration Console at:

`https://<GatewayIPAddress>:8443/web_plugin_framework/webcontroller`

Edit each Media Broker configuration in turn and configure the following values as required:

- **Max Audio Video Pass-through Calls** – The maximum allowed pass-through video calls (given no other type of calls exist)
- **Max Audio Video Transcodable Calls** – The maximum allowed transcodable video calls (given no other type of calls exist)
- **Max Audio Only Pass-through Calls** – The maximum allowed pass-through audio only calls (given no other type of calls exist)
- **Max Audio Only Transcodable Calls** – The maximum allowed transcodable audio only calls (given no other type of calls exist)

**Note:** If any of the values are set to zero, the feature is disabled.

---

## Installing the support website

### About this task

The support website is used in addition to the reference clients to debug Avaya Mobile Video Server. For instructions on how to use it, consult the *Using the support website* section in the *Troubleshooting Avaya Mobile Video* document.

### Procedure

1. Open the JBoss administration console at:  
`https://<AWMVS_IPAddress>:9990`
2. Once authenticated, click **Manage Deployments**.
3. Click **Add**.
4. Select the `support.war` file from the `..\Install\Support` folder of the downloaded software.
5. Click **Next**.
6. Click **Save**.
7. Once the upload has completed, find the `support.war` file in the *Available Deployment Content* table and select it.
8. Click **Assign**.
9. Select the main-server-group.
10. Click **Save**.

The support website is now available at

`https://<GatewayExternalAddress>:8443/support`

Where `<GatewayExternalAddress>` the IP address or hostname of the Avaya Mobile Video server. If the deployment uses a reverse proxy, this is the external Avaya Mobile Video Server IP address or hostname.

---

## Configuring Avaya Aura® components

To integrate the Avaya Mobile Video Server with Avaya Aura®, it needs to be configured as a SIP entity in the System Manager Administration interface. To do this, browse to the System Manager Administration Interface and log in, then:

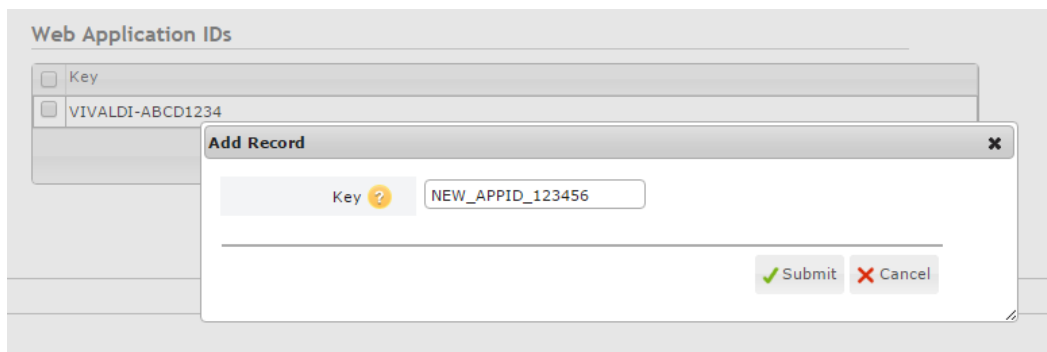
1. Click **Elements > Routing**
2. Click **SIP Entities**
3. Click **New**
4. Set the **Name** field
5. Set the **FQDN or IP Address** field to the Avaya Mobile Video Gateway Address
6. Set the **Type** to **Other**
7. In the **Entity Links** section:
8. Click **Add**
9. Set the **Name** field to `SM_GW_5060_TCP`
10. Set **SIP Entity 1** to the Session Manager Entity
11. Set the **Port** to `5060`
12. Set The **SIP Entity 2** to the Avaya Mobile Video Gateway Entity
13. Set the **Port** to `5060`
14. Set the Connection policy to **trusted**
15. Click **Add**
16. Set the **Name** field to `SM_GW_5061_TLS`
17. Set **SIP Entity 1** to the Session Manager Entity
18. Set the **Port** to `5061`
19. Set The **SIP Entity 2** to the Avaya Mobile Video Gateway Entity
20. Set the **Port** to `5061`
21. Set the Connection policy to **trusted**
22. Click **Commit** to save the new SIP Entity

---

## Configure Web Application ID

To Configure the Web Application ID that is used by the Web Application to create session tokens to be used by clients, do the following:

1. Browse to the **General Configuration** page on the Avaya Mobile Video Server web administration console:  
`https://<GatewayIPAddress>:8443/web_plugin_framework/webcontroller`
2. In the **Web Application IDs** section, click **Add** and enter the 16-character ID of your choice, for example:



The screenshot shows the 'Web Application IDs' section of the Avaya Mobile Video Server web administration console. A table lists existing records with columns for 'Key' and 'Value'. One record is visible with 'Key' as 'VIVALDI-ABCD1234'. An 'Add Record' dialog box is open, prompting for a 'Key' (with a help icon) and a 'Value'. The 'Value' field contains 'NEW\_APPID\_123456'. At the bottom of the dialog are 'Submit' and 'Cancel' buttons.

3. Click **Submit** to close the dialog and then click **Save** to save the changes.



---

# Configuration notes

---

## Enabling video calling on Communication Manager

If video calling is not already enabled on the Communication Manager, there are some SAT changes required on Communication Manager to enable video.

1. On the SAT screen, type `change system-parameters customer-options`.
2. On page 4, set the **Multimedia IP SIP Trunking** field to `y`.
3. To enable video for the SIP signaling group, on the Communication Manager SAT screen, type `change signaling-group <group name>`, and set the following parameters:

**IP Video:** `y`

**Direct IP-IP Audio Connections:** `y`

**Initial IP-IP Direct Media:** `y`

**Note:** The value of the **Far-end Network Region** field of the **Signaling Group** is used in the following steps.

4. To identify the codec set, on the Communication Manager SAT screen, type `display ip-network-region n`, where `n` is the `ip-network-region` shown in the SIP signaling group, and navigate to the Inter Network Region Connection Management pages. Ensure that the destination region number is identical to the far-end network region of the signaling group. Note the codec set number of the network region.
5. To add **video to ip-codec-set**, on the SAT screen, type `change ip-codec-set <codec set number used by the SIP signaling group>`.
6. Add or change the required audio codecs, and set the following parameters on page 2:

**Allow Direct-IP Multimedia:** `y`

**Maximum Call Rate for Direct-IP Multimedia:** `768 Kbps`

**Maximum Call Rate for Priority Direct-IP Multimedia:** `768 Kbps`

7. Enable Video on each station by typing `change station <station Extension>` and set the following parameters:

### Page 1

- a. **IP Softphone:** `y`
- b. **IP Video softphone:** `y`

### Page 2

- a. **IP-IP Audio Connections:** `y`
- b. **Service Link Mode:** `as-needed`

---

## Configuring Communication Manager VDNs

The time that an incoming call spends in the queue (before being dropped) when dialing a VDN depends on a configuration value specified in the Communication Manager signaling group. The default time is 3 minutes.

To change this value:

1. Open the Communication Manager system administration tool by running the `sat` command from the Communication Manager command line.
2. Edit the signaling group that is used for communication with Communication Manager, for example `change signaling-group 1`.
3. Find the `Session Establishment Timer (min)` setting.
4. Set the value desired: valid values are 3 – 120 minutes.

---

## Configuring Communication Manager UUI settings

In many use cases, a Context ID (a string encapsulating some business logic, such as a customer reference number or an Order number) is passed upwards by the mobile video client applications when making a call to the Avaya Mobile Video server. This context ID will be passed verbatim through the call and available on the One-X Agent interface, under the identifier “UUI” (User-to-user information), when the call is presented. In order to support the passing of this UUI, the following changes should be made on the CM:

1. Open the CM system administration tool by running the `sat` command from the CM command line
2. Edit the class of restriction that is associated with the agent stations, for example `change cor 1`
3. On Page 2, set the **Station-button display of UUI IE Data** to `y`.
4. Edit the trunk group associated with the SIP trunk, for example `change tru 1`.
5. On Page 3, set **UUI Treatment** to `shared`.
6. Edit each agent station, for example `change station 40000`.
7. On Page 4 or 5 add a **uui-info** button assignment to an available button.

The customer must ensure that their choice of `<context whatever>` length does not exceed the maximum available length that UUI supports or the `<context whatever>` will be truncated. This length is typically 32 characters, but may change depending on the customer’s usage of UUI. For full details on UUI, see the Avaya Aura® Call Center Elite documentation.

# Chapter 5: Security

---

## Changing the Avaya Mobile Video Gateway user interface password

After installation, the default passwords should be changed for Avaya Mobile Video Gateway Administration Interface. The process below details how to do this. The Admin username can also be changed.

1. Log into the Avaya Mobile Video Gateway Administration Interface at:  
`https://<GatewayIPAddress>:8443/web_plugin_framework/webcontroller`  
The default credentials after installation are:  
Username: admin  
Password: admin
2. To change the password, open a browser and navigate to the User Credentials page:  
`https://<GatewayIPAddress>:8443/web_plugin_framework/webcontroller/credentials`

---

## Setting the Avaya Mobile Video Server password constraints

By default the only constraint enforced when setting a password for the Avaya Mobile Video Gateway administration interface is that the password must be at least four characters long. You can use the Avaya Mobile Video Application Server Management Console to add a system property which defines a regular expression which any new password must match before it is accepted.

To set password constraints:

1. Open a web browser and navigate to the Avaya Mobile Video Application Server Management Console at:  
`https://<AWMVS_IPAddress>:9990`
2. Click **Server > Server Groups**. The Server Groups page displays:

Host: master-prod-fusionwe...

**Server**

- Server Configurations
- Server Groups**

**Host Settings**

- JVM Configurations
- Interfaces
- Host Properties

**Group Configurations**

**Server Groups**

A Server Group does specify a common management policy for a set of servers. Server Groups are associated with profiles.

Available Group Configurations

Add Remove

Group Name	Profile
lb-server-group	lb
main-server-group	ha
mgmt-server-group	management

<< 1-3 of 3 >>

Attributes JVM Configuration System Properties

Add

Key	Value	Boot-Time?	Option
saltedPasswordHashPBKDF2WithHmacSHA1	srZgdXaJA++0gXv1dzSD03zRvCQ=	true	Remove
sips.identity.group	main-server-group	true	Remove
sips.trust.group	default-trust	true	Remove
wpf.gateway.rest.host	prod-fusionweb.thrupoint.com	true	Remove
wpf.gateway.rest.url	http://prod-fusionweb.thrupoint.com:8080	true	Remove

- In the **Available Group Configurations** list, select `main-server-group`.
- To add the new system property, select the **System Properties** tab and click **Add**. The **Create System Property** dialog displays.

**Create System Property**

Name:

Value:

Boot-Time: ☐

Save Cancel

- In the **Name** field, enter `appserver.admin.password.validation.regexp`
- In the **Value** field, enter an appropriate Java-style regex.

#### Example regular expressions:

- At least 6 characters:  
`.{6,}`
- At least 6 alphanumeric characters:  
`^[a-zA-Z0-9]{6,}$`
- At least 6 alphabetic characters, with a mix of both upper- and lower-case characters:  
`^(?=.*[a-z])(?=.*[A-Z])[a-zA-Z]{6,}$`

- At least 6 alphanumeric characters, including both alphabetic and numeric characters:

```
^(?=.*\d) (?=.*[a-zA-Z]) [a-zA-Z0-9]{6,}$
```

7. Check the **Boot-Time** checkbox.
8. Click **Save**.

The new system property is now displayed in the System Properties list for main-server-group on the Server Groups page.

---

## Enabling secure communication between Avaya Mobile Video Media Broker and Avaya Mobile Video Gateway

There is a REST service running on the Media Broker which services requests from the Avaya Mobile Video Gateway to set up and tear down media routes, send DTMF and also to monitor the health of all Avaya Mobile Video Media Brokers. By default, this connection is unsecured after installation and you need to configure it for HTTPS if you require this connection to be secure.

When HTTPS is set up, the Avaya Mobile Video Media Broker is authenticated by Avaya Mobile Video Application Server. Hostname verification is done via the Subject Alternative Name (SAN) entries in the server certificate.

---

### Creating the trust certificate

1. To create the server certificate and keystore, run the following command on the Avaya Mobile Video Media Broker install directory:

```
keytool -genkeypair -alias <alias_name> -keyalg RSA -keystore  
<keystore_name> -keysize 2048 -ext  
san=ip:<ipaddress>,dns:<fqdn> -dname "CN=<common_name>"
```

Where:

- `alias_name`—The alias that the certificate is given (for example, `mediabroker`).
  - `keystore_name`—The name of the key store file that is generated (for example, `mediabroker.jks`).
  - `common_name`—The common name to use in the certificate (for example, `selfsigned`).
  - `ipaddress` and `fqdn`—These are the IP address and FQDN of the server that the Avaya Mobile Video Media Broker is running on. If an FQDN has not been configured, only use `ipaddress`.
2. When prompted for a password for the new keystore and certificate, it is recommended that you keep these the same.
  3. To export the public certificate for installation in the Avaya Mobile Video Application Server truststore, run the following command:

```
keytool -export -alias <alias_name> -file <pem_name> -keystore  
<keystore_name> -rfc
```

Where `pem_name` is the name of the `.pem` file to be generated. For example, `mediabroker.pem`.

4. Move the generated keystore (from step 1) to the `/opt/avaya/awmvs` directory.
5. Update the following settings in `{MEDIA_BROKER_INSTALL_DIR}/controller.properties`:
  - Set the HTTPS port. For example, 8093.
  - To turn off HTTP, set the HTTP port to 0.
  - Set `keystore.file.path` to point to the `keystore_name` (.jks file) created in step 1 above.
  - Set `keystore.file.password` to the password used in step 2 above.

---

## Importing the trust certificate

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
2. From the top-right menu select **Profiles**.
3. From the **Profile** drop-down list, select the **Management** profile.
4. From the menu on the left, expand **Subsystems > Trust Management** and select **Trust Certificates**.
5. Select the trust certificate group that you want to work with (the default trust group).
6. Click **Import**.
7. Enter a meaningful name, preferably indicating the CA whose certificate you want to import, and the security password (default password is: `changeit`).
8. Open the certificate from the unknown CA in a text editor and copy all of the contents, including the start and end tags.
9. In the **Encoded certificate** field, paste the certificate text, and click **Save**.
10. The certificate is imported into the trust certificate group directory and then copied to each server in the group.

---

## Configuring Avaya Mobile Video Media Broker security

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:8443/web_plugin_framework/webcontroller`
2. Choose the **Gateway** option on the top menu, select the **Media Broker** menu item.

3. Change the **control port** to the port specified in the `controller.properties` file (the default is 8092).
4. Change the **Control Type** to **Secure**.
5. Click **Save** at the bottom of the page.
6. Restart the Avaya Mobile Video Media Broker, from the command line on the media broker, run the command: `service media_broker restart`.

The Avaya Mobile Video Gateway should also be restarted by executing the restart command: `service awmvs restart`.

The communication between the Avaya Mobile Video Media Broker and the Avaya Mobile Video Gateway is now secure.

---

## Enabling secure SIP communication between Avaya Mobile Video Gateway and Session Manager

The Avaya Mobile Video Gateway and Avaya Session Manager communicate only using SIP. This communication can be changed to use secure SIPS communication by following the procedures below.

---

### Configuring Avaya Mobile Video Gateway

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:8443/web_plugin_framework/webcontroller`
2. Chose the **Gateway** option on the top menu, select the **General Administration** menu item.
3. In the **Outbound SIP Servers** table, `sip:sessionmanager` is listed. Click the entry in the table to make it editable and change the value to `sips:sessionmanager;transport=tls`.
4. Click **Save** at the bottom of the page.
5. On the Avaya Mobile Video Gateway server, locate and download locally the file `/opt/avaya/awmvs/x.x.x/awmvs_res/adaptation-db.xml`.
6. Modify the two lines that specify the **Session Manager** address to use a sips URI, for example:  
`<action>route:sips:10.30.244.134;transport=tls</action>`
7. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
8. Log in with the administrative credentials supplied at install time.
9. Select **Runtime** on the top right of the page.
10. Select **Manage Deployments** on the left side of the page.
11. Find and select the `adaptation-db.xml` file in the Content Repository list.

12. Click **Update**.
13. Select the `adaptation-db.xml` file that was modified and stored locally.
14. Click **Next**.
15. Click **Save**.

---

## Configuring trust certificates

The following sections detail how to configure the certificates correctly on the Avaya Mobile Video Gateway and Session Manager to allow secure communication.

### Exporting the Avaya Mobile Video Gateway trust certificate

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
2. On the top right of the page, click **Profiles**.
3. On the left of the page, in the **Profile** drop down box, select **Management**.
4. Select **Trust Management > Trust Certificates** from the left hand menu.
5. In the **Trust Certificate Group** list, select **default-trust**.
6. Select the **installer-ca** certificate and click **Export**.
7. Enter the password (default is `changeit`).
8. Copy the certificate displayed, including the begin and end tags.

### Importing the Avaya Mobile Video Gateway trust certificate into Session Manager

1. In Avaya System Manager, navigate to **Home > Services > Inventory > Manage Elements**, select the Session Manager, click **More Actions** then select **Trusted Certificates**.
2. Click **Add**.
3. In **Select Store Type to add trusted certificate** select `SECURITY_MODULE_SIP`.
4. Select **Import as PEM Certificate**.
5. Paste the Avaya Mobile Video Gateway Trust Certificate (copied in step 8 of the previous section) into the textbox and click **Commit**.

### Importing the Session Manager certificate into Avaya Mobile Video Gateway

1. In Avaya System Manager, navigate to **Home > Services > Inventory > Manage Elements**, select the **Session Manager**, click **More Actions** then select **Trusted Certificates**.
2. Select the **Security Module SIP** root certificate, for example `CN=SIP Product Certificate Authority, OU=SIP Product Certificate Authority, O=Avaya Inc., C=US` and click **Export**.
3. Save the PEM file to a local directory.



4. Open the PEM file in a text editor and copy the certificate, including the begin and end tags.
5. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
6. On the top right of the page, click **Profiles**.
7. On the left of the page, in the **Profile** drop down box, select `management`.
8. Select **Trust Management->Trust Certificates** from the left hand menu.
9. Click **Import**, change the name to `avaya-sm`, supply the password and paste the Session Manager certificate copied in step 4 in the **Encoded Certificate** text box.
10. Click **Import**.

At this point, secure communication is possible between the Avaya Mobile Video Server and Session Manager.

---

## Enabling secure Media between Avaya Mobile Video Media Broker and Avaya One-X® Agent

By default, the media between the Avaya Mobile Video Media Broker and the Avaya one-X® Agent (the SIP leg of the call) is unencrypted. Media between the Client and the Avaya Mobile Video Media Broker is always encrypted.

Encryption can be enabled using the following steps.

**Note:** Encryption is either on or off, so if it is turned on and Avaya one-X® Agent is not configured to use encryption, calls will fail.

The Communication Manager in this release only supports encrypted audio, not encrypted video.

The encryption used is `AESCM128_HMAC80`.

---

## Configuring the Avaya Mobile Video Media Broker

1. Connect using SSH to the Avaya Mobile Video Media Broker.
2. Navigate to the Avaya Mobile Video Media Broker installation directory at `/opt/avaya/awmvs/x.x.x/mv sdk/media_broker`.
3. Edit the `proxy.properties` file and edit the values to enable SRTP. The recommended settings for Avaya one-X® Agent are:
  - `srtp.enabled=true`
  - `srtp.video.encrypted=false`
  - `srtp.audio.encrypted=true`
  - `srtcp.enabled=false`
  - `srtp.rtp.protocol=SAVP`

4. Restart the Avaya Mobile Video Media Broker using the command: `service media_broker restart`.

---

## Configuring the Communication Manager

The Communication Manager needs to have the license applied that allows Media Encryption.

The Communication Manager `ip-codec-set` should be modified to support the `AESCM128_HMAC80` cipher:

1. Open the Communication Manager system administration tool by running the `sat` command from the Communication Manager command line.
2. Edit the `ip-codec-set`, for example change `ip-codec-set 1`.
3. Under the **Media Encryption** settings, enter `1-srtp-aescm128-hmac80`.
4. Below the `1-srtp-aescm128-hmac80` setting, enter the setting: `'none'`. This is to allow un-encrypted video to be accepted for H323 video softphones, which do not support encrypted video. With this configuration, the audio portion of the call will still be encrypted using `-aescm128-hmac80`.

---

## Configuring the signaling group between Communication Manager and Session Manager

1. Open the Communication Manager system administration tool by running the command `sat` from the Communication Manager command line.
2. Busy-out the signaling group that is used between Communication Manager and Session Manager. For example, if you are using signaling group 3, enter the following command:

```
busy signaling-group 3
```

3. Enter the following command:

```
change signaling-group 3
```

4. Set the **Enforce SIPS URI for SRTP?** option to `n`.
5. Save the change.
6. Release the busy-out on the signaling group between Communication Manager and Session Manager. For example, enter the following command:

```
release signaling-group 3
```

---

## Changing the configured HTTP/HTTPS ports

By default, Avaya applications listen for service traffic on port 8443 for HTTPS, and 8080 for HTTP traffic.

If you want to change from the default, the simplest way to change the HTTPS port is to edit the `domain.xml` file, as follows:

1. Stop the Avaya Mobile Video Server service by entering the following command:  
`service awmvs stop`

2. Edit `/opt/avaya/awmvs/x.x.x/awmvs/domain/configuration/domain.xml`

3. Change all references to the existing port to the new port number. In a basic install of the Avaya Mobile Video Server the following 4 lines will need changing.

```
<connector name="https" protocol="HTTP/1.1" scheme="https" socket-
binding="https" proxy-name="server.example.com" proxy-port="8443"
secure="true" executor="http-connector">
.....
<socket-binding name="https" interface="lb-public" port="8443"/>
.....
<property name="gateway.rest.url"
value="https://192.168.1.1:8443/admin/gateway/1.0"/>
<property name="wpf.gateway.rest.url"
value="https://192.168.1.1:8443/admin/gateway/1.0"/>
```

**Note:** Additional applications may also have their own configuration that references the HTTPS port; if so they should also be changed.

4. Edit the `/opt/avaya/awmvs/x.x.x/awmvs/bin/URL_monitor.list` file and update the URL with the appropriate HTTP/HTTPS and port number values.

5. Start the awmvs service by entering the following command:

```
service awmvs start
```

For changing the HTTP port a similar procedure should be followed, replacing all references to the default 8080 port.

To change the SIP ports for the Avaya Mobile Video Server you must change settings in two locations on the Application server web interface.

**Note:** Before making these changes you will need 3 new ports, the SIP defaults are:

5060  
5061  
5062

In this example we are going to change them to:

5075  
5076  
5077

You must confirm that these ports are free and not already in use before making these changes. These ports are examples and others may be used.

---

## Configuring the Avaya Mobile Video Server HA SIP profile

1. In a web browser, navigate to:

```
https://<AWMVS_IPAddress>:9990
```

2. Log in as administrator. The default credentials are

**Username:** administrator

**Password:** administrator

3. Click **Profiles** on the top right of the screen.
4. Click **Sip** then **Sip Servlets** on the left hand side panel.
5. Click **Connectors** tab at the top.
6. Edit each of the **Connectors Static Server Ports** by clicking **Edit** and changing the appropriate field.

For example:

Change **sip-tcp** from 5060 to 5075

Change **sip-tls** from 5061 to 5076

Change **sip-udp** from 5060 to 5075

Change **sip-ws** from 5062 to 5077

**Note:** Save each setting after making a change.

The screenshot shows the 'Application Server 2.1.24' configuration interface. The top navigation bar includes 'Profiles', 'Server', and 'Runtime'. The left sidebar shows a tree view with 'Subsystems' expanded, containing 'Connector', 'Container', 'Core', 'Infinispan', 'License Client', 'OSGi', 'Security', 'Sip', and 'Web'. Under 'Sip', 'Sip Servlet' is selected, showing 'Application Config', 'Application Routers', and 'Web'. The main content area is titled 'Sip Connectors' and shows a table of available connectors. Below the table is a 'Selection' section with fields for Name, Protocol, Scheme, Enabled?, and Static Server Port.

Name	Protocol	Enabled?
sip-tcp	SIP/2.0	<input checked="" type="checkbox"/>
sip-tls	SIP/2.0	<input checked="" type="checkbox"/>
sip-udp	SIP/2.0	<input checked="" type="checkbox"/>
sip-ws	SIP/2.0	<input checked="" type="checkbox"/>

Selection

Name: sip-tcp Protocol: SIP/2.0 Scheme: sip Enabled?: ☒ Use Static Address?: ☐

Static Server Address: Static Server Port: 5075

## Configuring the Load Balancer sockets

1. On the top left, in the **Profile** drop-down box, select **lb**.
2. In the lower half of the left hand side panel, click **Socket Bindings**.
3. Click **View > For lb-sockets**.

4. In the **Available socket bindings** for the names stated above, change the ports to match your previous configurations.

**Note:** Save these configurations each time you make a change.

The screenshot shows the 'Application Server 2.1.24' configuration window. The left sidebar has a tree view with 'Subsystems' expanded, showing 'Core', 'Infinispan', 'Load Balancer', and 'General Configuration'. Under 'General Configuration', 'Socket Binding' is selected. The main area is titled 'Socket Bindings' and shows a list of 'Available Socket Bindings'. The list has columns for Name, Port, and MCast Port. The 'sip-tcp' entry is highlighted. Below the list, there is a 'Selection' section with 'Save' and 'Cancel' buttons. The 'Name' field is set to 'sip-tcp', the 'Interface' is 'lb-public', and the 'Port' is '5075'. There is also a 'Fixed Port?' checkbox which is unchecked. At the bottom, there is a 'Multicast' checkbox which is checked.

Name	Port	MCast Port
jgroups-udp	55180	45688
jgroups-udp-fd	54180	
osgi-http	8070	
remoting	4427	
sip-tcp	5075	
sip-tls	5076	
sip-udp	5075	
sip-ws	5077	

Once you have completed all the above, restart the Avaya Mobile Video Server using the following command on your server:

```
service awmvs restart
```

Once that is restarted, you can confirm that the services are listening on the new port by running the following command:

```
netstat -anp | grep <new port number>
```

The screenshot shows a terminal window with the command `netstat -anp | grep 5075` executed. The output shows two lines: one for TCP and one for UDP, both listening on 192.168.8.143:5075. The process is identified as java.

```
[root@cs-rmorgan2 log]# netstat -anp | grep 5075
tcp        0      0 192.168.8.143:5075 0.0.0.0:*        LISTEN      18534/java
udp        0      0 192.168.8.143:5075 0.0.0.0:*
```

---

# Managing the Avaya Mobile Video Server SSL certificates

Avaya Mobile Video Server is installed with certificates signed by the installer Certificate Authority which are not trusted outside the server. For lab installations, the procedures found in [Enabling secure communication between Avaya Mobile Video Media Broker and Avaya Mobile Video Gateway](#), [Enabling secure SIP communication between Avaya Mobile Video Gateway and Session Manager](#), and [Enabling secure Media between Avaya Mobile Video Media Broker and Avaya One-X® Agent](#) can be achieved using the default installer certificates. The steps detailed below need only be carried out when you are moving to using signed certificates in a production environment.”

For production installations it is recommended to use certificates signed by well-known CAs. If self-signed certificates are used instead, you should ensure that the domain name and server name match and they should use SHA-2.

Certificates need to be configured for:

1. `https` for the `main-loadbalancer-group`

**Note:** This is not needed if you are SSL offloading at the Avaya SBCE and are satisfied to use unsecured http between the Avaya SBCE and the Avaya Mobile Video Gateway.

2. `sips` for the `main-loadbalancer-group`

Required if you are receiving internal SIP calls over TLS

**Note:** To make internal SIP calls over TLS you need to import external SIP entity certificates into your trust store.

---

## Certificate import process

### Generating Certificate Signing Requests

First, generate a Certificate Signing Request (CSR) to send to the third-party CA. Do the following:

1. In the **Management Console**, from the top-right menu select **Profiles**.
2. From the **Profile** drop-down list, select the management profile.
3. From the menu on the left, expand **Subsystems > Trust Management** and select **ID Certificates**.
4. Select the identity certificate group that you want to work with (for example, `main-loadbalancer-group`).
5. Select the certificate that you want to be signed in the list (for example, `https`).
6. Click **Generate CSR**, a pop-up window appears.
7. Enter the security password.
8. Optionally enter a challenge Password for revocation purposes.

9. Enter the DN for the component for which you are generating a certificate. The Subject DN is already pre-populated for HTTPS and SIPS, for example `CN=my.server.avaya.com`, but you can change this if you need to.
10. Click **Generate**; a dialog containing the CSR text is displayed.
11. Copy all of the displayed text, including the start and end tags, and paste it into a text editor, then save as a `.csr` file.
12. Close the dialog.

## **Sending a certificate to the external CA for signing**

The procedure for getting your certificate signed by a third-party CA depends upon the requirements of that CA—see the guidance from the CA.

## **Importing the signed certificate**

1. When you receive the certificate back from the CA you must then import it into the right identity certificate group and named listing, from which the CSR was generated.
2. The certificate that you import needs to contain the complete certificate chain from your server down to the root certificate. Some CAs provide the certificate chain in multiple separate files. If this is the case, these need to be concatenated into one file, without empty lines in it, before importing.
3. From the page that you generated your CSR from, select the identity certificate group that you want to import into.
4. Select the named certificate entry that you requested the CSR for, for example SIPS or HTTPS.
5. Click **Import**, a dialog displays.
6. Enter the security password for the named certificate.
7. Open the certificate in a text editor, and copy all of the contents, including the start and end tags.
8. In the **Encoded Certificate** field, paste the certificate text.
9. Click **Import**.  
The Console is updated to reflect the new certificate details, such as the issuer DN and the expiry date. The updated identity certificate group directory is then copied to each Avaya Mobile Video Application Server or Load Balancer in the Server Group.
10. Restart each server for the changes to take effect.

---

## Additional security

It is recommended that to further secure the installation, Server CBC mode Ciphers and Weak Mac algorithms should be disabled.

To do this, on the Avaya Mobile Video Gateway and Media Broker:

1. Open the `/etc/ssh/sshd_config` file.
2. Modify/Add the Ciphers and Mac lines as follows:
  - o Ciphers `aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128`
  - o MACs `hmac-sha1,umac-64@openssh.com,hmac-ripemd160`
3. Restart the sshd service: `service sshd restart`.



# Chapter 6: Upgrade

This chapter takes you through the process for upgrading Avaya Mobile Video Server and Mobile Video SDK.

The pre-requisite version of Oracle JDK required for Avaya Mobile Video changed from JDK 7 Update 80 for Avaya Mobile Video 3.0.1 to JDK 8 Update 131 for Avaya Mobile Video 3.2. Upgrades from 3.0.1 to 3.2 or 3.2.2 are therefore not supported. Users running Avaya Mobile Video 3.0.1 must follow the uninstall instructions in Chapter 10 of the Installation guide to uninstall 3.0.1, then upgrade the JDK to `jdk-8u131-linux-x64.rpm` and install Avaya Mobile Video 3.2.2.

**NOTE:** While upgrades from Avaya Mobile Video 3.2 to 3.2.2 are supported; this is not recommended for production installs. New Apple App Store requirements mandate that iOS Apps can support IPV6 and in order to facilitate this, Avaya Mobile Video must be installed with a Cluster FQDN that is entered at install time on the 3.2.2 installer.

Lab users running Avaya Mobile Video 3.2 can follow the upgrade instructions below to upgrade from 3.2 to 3.2.2, but will not be able to use approved Apple App Store clients with the server. Unofficial Apps, built in Xcode, will still work as before with an upgraded server, but will not be accepted in the Apple App Store.

---

## Upgrade process

The process to upgrade the Avaya Mobile Video Server and Mobile Video SDK can be executed by a customer or a support engineer. The choice of who performs the upgrade is usually specified by the customer and is related to how comfortable the customer is with performing technical tasks.

If the customer is to run the procedure, it should be performed by a user that has the necessary administrative rights and knowledge.

A customer does not normally give SSH access to a third party to perform the upgrade. The most common scenarios are:

- Customer runs the upgrade themselves and only contacts a support engineer if issues occur.
- Customer runs the upgrade themselves with telephone assistance from a support engineer.
- Customer runs the upgrade themselves but allows a support engineer access to view the procedure using remote assistance software so they can advise.
- Customer provides access to support engineer using remote assistance software and support engineer performs the upgrade.

The choice of remote assistance software used is usually decided by the customer, and is more often than not initiated by the customer. It is however useful to have a remote assistance solution available to use if the customer does not have one in place.

The upgrade is performed by a single script that detects what has been installed on the server that

it is executed on (for example, Avaya Mobile Video Gateway only or Avaya Mobile Video Media Broker only), and upgrades the relevant components.

To upgrade the whole Avaya Mobile Video Server, the process is:

1. Run the upgrade on the Avaya Mobile Video Server.
2. Run the upgrade on each Avaya Mobile Video Media Broker in turn.

The Avaya Mobile Video Server Gateway and Media Broker can be upgraded individually if a specific patch is made available for the individual elements.

---

## Upgrade prerequisites

Before executing the upgrade, the upgrade artefact (`.tar.gz`) file needs to be placed in a location on the target server prior to upgrade, for example `/opt/x.x.x_upgrade`.

The `awmvs` and `media_broker` processes do not have to be stopped before upgrade. The upgrade script controls the services as necessary.

**Note:** Performing an upgrade will affect traffic currently on the Avaya Mobile Video Server. On production systems, we recommend that upgrades are carried out in a scheduled maintenance window during times of low traffic.

---

## Installation structure

Following installation of the Avaya Mobile Video Server, the file system has the following structure:

```
/opt/avaya/awmvs
+-- x0.y0.z0
|   +-- awmvs
|   +-- awmvs_res
|   +-- bin
|   +-- mvsdk
+-- bin (softlink)
```

In this structure, `x0.y0.z0` is the version of Avaya Mobile Video Server that was installed, and `/opt/avaya/awmvs/bin` is a soft link that link to resources in the version directory. The currently running Avaya Mobile Video Server components are those residing in the `x0.y0.z0` directory.

The `bin` directory contains commands for upgrade and version management. These are:

- `upgrade`
- `rollback`

---

## Upgrading the software

To perform an upgrade of Avaya Mobile Video Server to a newer version, the `upgrade` command is used. The command should be accessed via the `bin` soft link and run with root privileges. To display the usage of the `upgrade` command run:

```
% /opt/avaya/awmvs/bin/upgrade.sh
```

This also displays when the `-h` flag is used.

The help output shows that an upgrade is performed using the `-f` option, together with an upgrade pack file, for example:

```
% /opt/avaya/awmvs/bin/upgrade.sh -f MobileVideoSDKUpgrade-2.0.3.tar.gz
```

The upgrade process follows a query/answer model. When the upgrade has been completed, a message indicating the successful upgrade is presented. If anything goes wrong during the upgrade, the system is rolled back so that the previous version is reinstated.

Following successful upgrade, the file system looks like the following diagram:

```
/opt/avaya/awmvs
+-- x0.y0.z0
|   +-- awmvs
|   +-- awmvs_res
|   +-- bin
|   +-- mvsdk
+-- x1.y1.z1
|   +-- awmvs
|   +-- awmvs_res
|   +-- bin
|   +-- mvsdk
+-- bin (softlink)
```

In this diagram, `x1.y1.z1` is the version of Avaya Mobile Video Server upgraded to. The soft link `bin` is changed to link to resources within the newer version of Avaya Mobile Video Server. The currently running Avaya Mobile Video Server components are now those residing in the `x1.y1.z1` directory.

This completes the upgrade.

---

## Rollback

A previous version of Avaya Mobile Video Server can be reinstated by using the `rollback` command. The command should be accessed via the `bin` soft link and run with superuser privileges.

- To display the usage of the `rollback` command run:

```
% /opt/avaya/awmvs/bin/rollback.sh
```

This is also displayed when the `-h` flag is used.

- To list the versions of Avaya Mobile Video Server that have been installed, run the following command:

```
% /opt/avaya/awmvs/bin/rollback.sh -l
```

This lists the versions of Avaya Mobile Video Server that have been installed, either via normal installation or as a result of upgrading.

- To roll back to a specific version, the `-v` option is used. For example:

```
% /opt/avaya/awmvs/bin/rollback.sh -v x.y.z
```

This reinstates version `x.y.z` (assuming that is not the version already operating).

- The rollback command can be used to establish any version listed by `rollback.sh -l`, including rolling forward to newer versions relative to an established older version.

# Chapter 7: Administration

Some of the configuration for Avaya Mobile Video Server is in the following configuration files which are deployed to the server:

- `vivaldi.properties`
- `vivaldi_token_json.template`
- `adaptation-db.xml`

On initial install, the versions of these files that are deployed are also saved locally on the server in the event that the configuration may need to be modified. Some of the configuration sections below refer to these files. The files can be found in the following server directory:

```
/opt/avaya/awmvs/x.x.x/awmvs_res
```

---

## Changing the Avaya Aura SIP domain

To change the Avaya Aura® SIP domain, locate and download locally, then modify the following files and settings, found in the `/opt/avaya/awmvs/x.x.x/awmvs_res` location on the Avaya Mobile Video Gateway Server:

- `vivaldi.properties`  
Change `default.domain` to the desired Avaya Aura® SIP domain.
- `vivaldi_token_json.template`  
Change the `domain` value in the `identity` section to the desired Avaya Aura® SIP domain.
- `adaptation-db.xml`  
Change `<entity address="incorrect.sip.domain">` to the desired Avaya Aura® SIP domain.

Once modified, redeploy the changed files in place of the existing files:

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
2. Log in with the administrative credentials supplied at install time.
3. Select **Runtime** on the top right of the page.
4. Select **Manage Deployments** on the left side of the page.
5. Find and select the existing file e.g. `adaptation-db.xml` file in the Content Repository list.
6. Click **Update**.
7. Select the `adaptation-db.xml` file that was modified and stored locally.
8. Click **Next**.
9. Repeat the deployment process for the other configuration files.

---

## Changing the configured Session Manager

If the Session Manager address has been incorrectly configured, or if it needs to be changed to point to a new Session Manager system, this can be done by editing the `adaptation-db.xml` file.

On the Avaya Mobile Video Gateway server, locate and download locally the following file:

```
/opt/avaya/awmvs/x.x.x/awmvs_res/adaptation-db.xml
```

Open the file in a text editor and change both action/route lines which refer to the incorrect Session Manager address to the correct Session Manager address, for example:

```
<adaptations>
  <entity address="collaboratory.avaya.com">
    <outbound>
      <action>servlet:VivaldiAdaptation</action>
      <action>route:sip:new.sm.address</action>
    </outbound>
  </entity>
  <entity address="sessionmanager">
    <outbound>
      <action>servlet:VivaldiAdaptation</action>
      <action>route:sip:new.sm.address</action>
    </outbound>
  </entity>
</adaptations>
```

To apply the changes to the running Avaya Mobile Video Gateway server:

1. In a web browser, navigate to:

```
https://<AWMVS_IPAddress>:9990
```

2. Log in with the administrative credentials supplied at install time.
3. Select **Runtime** on the top right of the page.
4. Select **Manage Deployments** on the left side of the page.
5. Find and select the `adaptation-db.xml` file in the Content Repository list.
6. Click **Update**.
7. Select the `adaptation-db.xml` file that was modified and stored locally.
8. Click **Next**.
9. Click **Save**.

---

## Changing the Gateway External Address

The Gateway External IP address specifies the IP address or host name that the client uses to create the WebSocket. If an Avaya SBCE is used in the deployment, this address will be the external listen address configured on the SBCE. Otherwise it is the IP Address or FQDN of the Mobile Video Gateway server.

To change the Gateway External IP address, download the `vivaldi_token_json.template` file from the `/opt/awmvs/x.x.x/awmvs_res` folder on the Avaya Mobile Video Gateway Server.

Open the file with a text editor and edit the host entry under the `serverUrlDetails` area and enter the new Gateway External IP address.

```
...
...
"serverUrlDetails":
{
"secure":true,
"host":"XX.XX.XX.XX",
"port":"8443"
}
...
...
```

Once modified, redeploy the `vivaldi_token_json.template` file in place of the existing one using the following procedure:

1. In a web browser, navigate to:  
`https://<AWMVS_IPAddress>:9990`
2. Log in with the administrative credentials supplied at install time.
3. Select **Runtime** on the top right of the page.
4. Select **Manage Deployments** on the left side of the page.
5. Find and select the existing `vivaldi_token_json.template` file in the Content Repository list.
6. Click **Update**.
7. Select the `vivaldi_token_json.template` file that was modified and stored locally.
8. Click **Next**.
9. Click **Save**.

**Note:** The files stored in the `/opt/avaya/awmvs/3.0.1/awmvs_res/` folder are there for reference. These are copies of the files that were originally deployed at install time and will not change when you redeploy a modified version.

---

## Changing the certificate expiry warning interval

To change the interval at which the Administrator is warned (via the Performance Dashboard) that certificates are going to expire, change the following value in the `vivaldi.properties` file:

```
certificate.expiry.warning.days
```

The value is the number of days before expiry to alert.

# Chapter 8: Configuring video settings

The quality of a video call can be affected by many different factors and it is important to ensure that the configuration is optimally applied based on the network resources and bandwidth available.

---

## Typical video bandwidths

Bandwidth consumption predominantly depends on the following:

- Video resolution
- Frame rate
- Target bitrate

The following examples show the bandwidth used for different calls:

Resolution	Video Format (Aspect)	Quality	Typical Bandwidth
320 x 240	QVGA (4:3)	Standard Definition (SD)	256 Kbps - 620 Kbps
640 x 480	VGA (4:3)	Standard Definition (SD)	450 Kbps – 768 Kbps
1280 x 720	720p (16:9)	High Definition (HD)	768 Kbps - 2048 Kbps

These figures are per side of a call, for example an HD 720p video call would need a minimum 1536 kbps of network bandwidth available.

Depending on the business requirement, enough network bandwidth must be made available to support the required number of calls at the required resolution and frame rate.

Networks switches used in the deployment should not be utilized above 50%.

---

## Configuring Avaya Mobile Video Gateway video

On the Avaya Mobile Video Gateway, the bandwidth to be used for a call can be found in the `Bitrate Settings` section on the **Media Configuration** page.



---

## Adaptive bitrate

By default, the Avaya Mobile Video Gateway is configured to use Adaptive Bitrate Adjustment. Video endpoints distributed over the internet cannot guarantee a stable bitrate required for Real Time Communications. It is important for the stream of packets which construct a video call to arrive at their destination in a timely fashion; depending on the network pathway between a client and Avaya Mobile Video Media Broker; network buffering or QoS restrictions may limit or severely impact Video performance.

**Note:** Adaptive bitrate is only supported for traffic between the external client and Avaya Mobile Video Media Broker.

Avaya Mobile Video Media Broker implements REMB (for WebRTC Clients) specifications so that it can monitor the RTCP Reports and react to the ever-changing environment that clients may face.

These protocols request more or less bandwidth from clients if the network conditions change; if there is no change then the protocol maintains the current bitrate.

Avaya Mobile Video Media Broker allows the administrator to configure constraints on the bandwidth to maintain video quality parameters that can be met by the business requirement. Configure the following parameters for adaptive bitrate:

- **Initial Adaptive Bitrate** - When a call starts, this is the Initial value utilized by the Avaya Mobile Video Media Broker's encoder.
- **Minimum Adaptive Bitrate** - This is the minimum acceptable value of video produced by the encoder. It is also the minimum value that the Avaya Mobile Video Media Broker requests from clients.
- **Maximum Adaptive Bitrate** - This is the maximum value given to the video encoder; it is also the maximum value that the Avaya Mobile Video Media Broker requests from clients.

The **Maximum** and **Minimum Adaptive Bitrate** values give bounds to the threshold that the Avaya Mobile Video Media Broker goes to when rendering video. This value affects Avaya Mobile Video Media Broker in the following ways:

- Avaya Mobile Video Media Broker requests no more and no less than these value from clients
- Avaya Mobile Video Media Broker does not use a value outside this range to encode video

The **Initial Adaptive Bitrate** value is used by Avaya Mobile Video Media Broker when sending video at the beginning of a call before there is enough data collected from the RTCP to behave appropriately.

In most consumer cases it is appropriate to set this value equal to the Minimum Adaptive Bitrate; if the network is sufficient the bitrate of the call improves shortly after the call starts. Some video solutions may prefer that the video starts at a higher bitrate, in which case clients on an insufficient network have a worse experience until the bitrate falls.

The recommended Adaptive Bitrate settings are:

- **Initial Adaptive Bitrate** - 256
- **Minimum Adaptive Bitrate** - 256
- **Maximum Adaptive Bitrate** – This should be set based on the call bandwidth required (see [Typical video bandwidths](#)) and network resources available.

---

## Fixed bitrate

Using fixed bitrate is not the recommended setting for video, but it can be used if necessary. One issue with this approach is that if there is more bandwidth available on the network, the video quality could be improved, but as the rate is fixed, the quality remains the same. Using adaptive bitrate in this situation would improve the video quality using the extra bandwidth available.

---

## Configuring Communication Manager video

On the Communication Manager system, the bandwidth to be used by a call can be found in the `ip-codec-set` that is in use. There are two values that need to be configured:

- Maximum Call Rate for Direct-IP Multimedia
- Maximum Call Rate for Priority Direct-IP Multimedia

These settings relate to the combined audio and video transmit rate or receive rate for video calls. You can use these settings to limit the amount of bandwidth used. These values relate to the bandwidth available for the each side of the call.

These values should be set based on the call bandwidth required (see [Typical video bandwidths](#)) and the network resources available. To set the values:

1. Open the Communication Manager system administration tool by running the `sat` command from the Communication Manager command line.
2. Edit the `ip-codec-set`, for example change `ip-codec-set 1`.
3. Navigate to the second settings page.
4. Set Maximum Call Rate for Direct-IP Multimedia to the required value.
5. Set Maximum Call Rate for Priority Direct-IP Multimedia to the required value.

# Chapter 9: Media Tuning

The default configuration is often close to optimal for most deployments. This section details some guidelines for achieving the best video performance.

1. Configure the Communication Manager bitrate (see [Communication Manager video configuration](#)) as low as you find acceptable. This makes the video experience more reliable at lower bandwidths.
2. Keep adaptive bitrate enabled and the initial adaptive bitrate low. If you find the quality unacceptable at the start of calls you can tune the initial bitrate higher; lower initial bitrate helps initial quality on low bandwidths; higher initial bitrate helps quality improve quicker if the available bandwidth is higher than the initial bitrate you have set.
3. Avoid transcoding - use H.264 for Video calls and g.711 for audio calls as far as possible. You may need to prioritize H.264 and g.711, or ban VP8 (note that banning VP8 will prevent Chrome calls from working).
4. Keeping video resolutions appropriate to the target bitrate and device can help. If you only need to use a small video window then lower resolutions are appropriate.
5. `Fast Picture Update Poll Period (mSecs)` should only be used when absolutely required as this is detrimental to quality. Values under 1000 (polling more than once per second) should only be used for very small numbers of calls as the impact is very high. Normally this should be disabled or set to a value of 5000 or more.
6. Configure CAC correctly to avoid overloading any Avaya Mobile Video Media Broker. The *Capacities* section details the call limits for Avaya Mobile Video Media Brokers.

# Chapter 10: Uninstall

This chapter details how to uninstall the Avaya Mobile Video Gateway and Avaya Mobile Video Media Broker .

---

## Uninstalling the Avaya Mobile Video Gateway

1. Connect to the Avaya Mobile Video Gateway using SSH.
2. Stop the Avaya Mobile Video Gateway by running the command:  
`service awmvs stop`
3. Check that the service is stopped by running:  
`service awmvs status`  
This should display the following:  
`Avaya Mobile Video Server is not running`
4. Remove the `/opt/avaya/awmvs/x.x.x` directory:  
`rm -rf /opt/avaya/awmvs/x.x.x`
5. Remove the `/etc/awmvs.conf` file.
6. Remove the `/etc/csdk.conf` file.
7. Remove the `/etc/init.d/awmvs` file.

---

## Uninstalling the Avaya Mobile Video Media Broker

1. Connect to the Avaya Mobile Video Media Broker using SSH.
2. Stop the Avaya Mobile Video Media Broker by running the command:  
`service media_broker stop`
3. Check that the service is stopped by running:  
`service media_broker status`  
  
This should display the following:  
`Media Broker is not running`
4. Remove the `/opt/avaya/awmvs/x.x.x` directory:  
`rm -rf /opt/avaya/awmvs/x.x.x`
5. Remove the `/etc/csdk.conf` file
6. Remove the `/etc/init.d/media_broker` file

**Note:** it is not possible to unhardened the operating system (which is optionally performed during installation). If you require the O/S to be unhardened then the O/S will need to be reinstalled. If using a virtualized environment you may be able to revert to a snapshot created prior to installation of Avaya Mobile Video Server.

# Chapter 11: Resources

---

## Documentation

The following table lists the related documents for Avaya Mobile Video Server. Download the documents from the Avaya Support website at <http://support.avaya.com>.

Title	Description
Avaya Mobile Video Overview and Specification	Describes the features and specifications for Avaya Mobile Video.
Avaya Mobile Video Release Notes	Describes any late-changing information about the release and known issues for the product.
Avaya Mobile Video Planning and Security Reference	Describes the components, deployment, and security options for Avaya Mobile Video.
Installing Avaya Mobile Video Server and Media Broker	Describes how to install Avaya Mobile Video.
Installing Avaya Media Client	Describes how to install Avaya Media Client.
Using Avaya Media Client	Describes how to use the features of Avaya Media Client.
Administering Avaya Mobile Video	Describes how to administer Avaya Mobile Video.
Avaya Mobile Video Server Software Development Guide	Describes how to develop Mobile Video applications.

---

## Training

Course code	Course title
<b>Avaya Oceana™ Solution Training</b>	
3420W	Avaya Oceana™ Solutions Design Fundamentals
3470T	Avaya Oceana™ Solutions Design Fundamentals APDS Online Test
2402W	Avaya Oceana™ Workspaces Agent Desktop Training
2404W	Avaya Oceana™ Workspaces Supervisor Desktop Training

---

## Support

Visit the Avaya Support website at <http://support.avaya.com> for the most up-to-date documentation, product notices, and knowledge articles. You can also search for release notes, downloads, and resolutions to issues. Use the online service request system to create a service request. Chat with live agents to get answers to questions, or request an agent to connect you to a support team if an issue requires additional expertise.

# Appendix A: OS hardening

This appendix details what occurs when the operating system is secured during installation. Refer to the comments in the script for a description of what security requirements are being applied.

```
#!/bin/bash

# This script, among other things, adds iptables rules based on what is being installed. So we need
# to know which components are being installed.
# It takes two arguments that can either be 'y' or 'n' designating if the Gateway and Media Broker
# are being installed, respectively.
if [ -z "$1" ] || [ -z "$2" ]
then
    echo "This script must be run with two arguments. USAGE: ./configure-os.sh <Gateway install y|n>
    <Media Broker install y|n>"
    exit 1
else
    case $1 in
        [yYnN] )
            ;;
        *) echo "You need to enter 'y' or 'n' for the two parameters";
            exit 1
            ;;
    esac

    case $2 in
        [yYnN] )
            ;;
        *) echo "You need to enter 'y' or 'n' for the two parameters";
            exit 1
            ;;
    esac
fi

GW_INSTALL=$1
MB_INSTALL=$2

function _change_key_value_pair_in_file()
{
    local _key=$1
    local _value=$2
    local _file=$3
    local _sep=$4

    if [ -w ${_file} ]
    then
        local _key_regexp="^([[:space:]]*${_key}[[:space:]]*${_sep}[[:space:]]*)"
        grep -qE "${_key_regexp}${_value}" ${_file}
        if [ $? -ne 0 ]
        then
            #Not already set
            if grep -qE "${_key_regexp}" ${_file}
            then
                #Comment out existing line
                sed -e "s/^([[:space:]]*${_key}[[:space:]]*.*\)/#\1/g" -i ${_file}
            fi
            echo "${_key}${_sep}${_value}" >> ${_file}
        fi
    fi
}

function _backup_file()
{

```

```

    cp $1{,-$(date +%Y%m%d_%H%M%S)}
}

function _disable_service()
{
    local _service=$1

    if [ -r /etc/init.d/${_service} ]
    then
        service ${_service} stop > /dev/null
        chkconfig --del ${_service}
    fi
}

function _populate_text_file()
{
    local _label=$1
    local _file=$2

    echo -n "$1"
    echo " (Enter Ctrl-d on a blank line when done)"
    echo
    local _content=$(cat)

    _backup_file "${_file}"
    echo "${_content}" > "${_file}"
}

# 147849-140 Logon Warning Banner - The system will provide a customer configurable warning banner
# for all interactive login sessions
function 147849-140_Logon_Warning_Banner()
{
    _populate_text_file "Enter the message to be displayed before a remote user attempts to log in"
/etc/issue.net
    _populate_text_file "Enter the message to be displayed before a local user attempts to log in"
/etc/issue
    _populate_text_file "Enter the message to be displayed after a user successfully logs in"
/etc/motd

    local _file=/etc/ssh/sshd_config

    _backup_file "${_file}"
    _change_key_value_pair_in_file "Banner" "/etc/issue.net" "${_file}" " "
    _change_key_value_pair_in_file "PrintMotd" "yes" "${_file}" " "

    if [ -r /etc/init.d/sshd ]
    then
        service sshd restart > /dev/null
    fi
}

#147849-191 SSH Service hardening - -
# Secure Shell (SSH) must be hardened to disable SSH Protocol version 1 (SSHv1), disable X11
# forwarding,
# restrict root access to console only and use only strong asymmetrikeys (RSA 2048, 4096 bit keys)
function 147849-191_SSH_Service_hardening()
{
    local _file=/etc/ssh/sshd_config

    _backup_file "${_file}"
    _change_key_value_pair_in_file "Protocol" "2" "${_file}" " "
    _change_key_value_pair_in_file "X11Forwarding" "no" "${_file}" " "
    _change_key_value_pair_in_file "PermitRootLogin" "no" "${_file}" " "

    if [ -r /etc/init.d/sshd ]
    then
        service sshd restart > /dev/null
    fi
}

# 147849-240 Disable insecure network services - - The following network services should be disabled
# by default and require administrator action to re-enable them:

```



```

# telnet, ftp, tftp, nfs, chargen, finger, http,X-windows, rlogin, rsh, rexec, netdump, portmap,
rwhod, smb, yppasswd, ypserv, ypxfrd
function 147849-240_Disable_insecure_xwindows_service()
{
    local _file=/etc/inittab

    grep -q "^id:3:initdefault:" ${_file}
    if [ $? -ne 0 ]
    then
        _backup_file "${_file}"
        grep -q "^id:.*:initdefault:" ${_file}
        if [ $? -eq 0 ]
        then
            sed -e "s/^\\(id:.*:initdefault:.*\\)\\$/#\\1/" -i ${_file}
        fi
        echo "id:3:initdefault:" >> ${_file}
    fi
}

# 147849-240 Disable insecure network services - - The following network services should be disabled
by default and require administrator action to re-enable them:
# telnet, ftp, tftp, nfs, chargen, finger, http,X-windows, rlogin, rsh, rexec, netdump, portmap,
rwhod, smb, yppasswd, ypserv, ypxfrd
function 147849-240_Disable_insecure_network_services()
{
    for service in vsftpd nfs httpd netdump rpcbind rwhod smb ypserv yppasswd ypxfrd
    do
        _disable_service "${service}"
    done
}

# 147849-240 Disable insecure network services - - The following network services should be disabled
by default and require administrator action to re-enable them:
# telnet, ftp, tftp, nfs, chargen, finger, http,X-windows, rlogin, rsh, rexec, netdump, portmap,
rwhod, smb, yppasswd, ypserv, ypxfrd
function 147849-240_Disable_insecure_network_xinetd_services()
{
    local _xinet_restart_required=0

    for service in telnet tftp chargen-dgram chargen-stream finger rlogin rsh rexec
    do
        local _file=/etc/xinetd.d/${service}

        if [ -w ${_file} ]
        then
            grep -q "^[[[:space:]]*disable[[[:space:]]]*=[[[:space:]]*yes" ${_file}
            if [ $? -ne 0 ]
            then
                _backup_file "${_file}"
                sed -e "s/^\\([[[:space:]]*disable[[[:space:]]]*=[[[:space:]]*.*\\)\\$/#\\1/" -i ${_file}
                sed -e "/" /i \
                disable
                = yes" -i ${_file}
                _xinet_restart_required=1
            fi
        fi
    done

    if [ ${_xinet_restart_required} -ne 0 ]
    then
        local _pid=$(pidof xinetd)
        if [ -n "${_pid}" ]
        then
            kill -USR2 ${_pid}
        fi
    fi
}

function _change_network_param()
{
    local _key=$1
    local _value=$2
    local _file=$3

```

```

sysctl -e -w ${_key}=${_value} > /dev/null
_change_key_value_pair_in_file "${_key}" "${_value}" "${_file}" "="
}

# 147849-250 Linux/Unix Networking parameters - - Linux or Unix operating systems should configure
the host network parameters as follows:
# net.ipv4.ip_forward = 0
# net.ipv4.conf.all.send_redirects = 0
# net.ipv4.conf.default.send_redirects = 0
# net.ipv4.conf.all.accept_source_route = 0
# net.ipv4.conf.all.accept_redirects = 0
# net.ipv4.conf.all.secure_redirects = 0
# net.ipv4.conf.all.log_martians = 1
# net.ipv4.conf.default.accept_source_route = 0
# net.ipv4.conf.default.accept_redirects = 0
# net.ipv4.conf.default.secure_redirects = 0
# net.ipv4.icmp_echo_ignore_broadcasts = 1
# net.ipv4.icmp_ignore_bogus_error_messages = 1
# net.ipv4.tcp_syncookies = 1
# net.ipv4.conf.all.rp_filter = 1
# net.ipv4.conf.default.rp_filter = 1
function 147849-250_Linux_networking_parameters()
{
    local _file=/etc/sysctl.conf

    _backup_file "${_file}"
    _change_network_param "net.ipv4.ip_forward" "0" "${_file}"
    _change_network_param "net.ipv4.conf.all.send_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.conf.default.send_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.conf.all.accept_source_route" "0" "${_file}"
    _change_network_param "net.ipv4.conf.all.accept_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.conf.all.secure_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.conf.all.log_martians" "1" "${_file}"
    _change_network_param "net.ipv4.conf.default.accept_source_route" "0" "${_file}"
    _change_network_param "net.ipv4.conf.default.accept_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.conf.default.secure_redirects" "0" "${_file}"
    _change_network_param "net.ipv4.icmp_echo_ignore_broadcasts" "1" "${_file}"
    _change_network_param "net.ipv4.icmp_ignore_bogus_error_messages" "1" "${_file}"
    _change_network_param "net.ipv4.tcp_syncookies" "1" "${_file}"
    _change_network_param "net.ipv4.conf.all.rp_filter" "1" "${_file}"
    _change_network_param "net.ipv4.conf.default.rp_filter" "1" "${_file}"
}

# 147849-040 Single user mode - Avaya products must require a password for single user mode.
# The root account must be password enabled and the following quoted text must be present in
/etc/inittab: "~~:S:wait:/sbin/sulogin"
function 147849-040_Single_user_mode()
{
    local _file=/etc/sysconfig/init

    _backup_file "${_file}"
    _change_key_value_pair_in_file "SINGLE" "/sbin/sulogin" "${_file}" "="
}

# 147849-130 Session Limits - Users should be restricted to a maximum number of 5 simultaneous
logins.
# Thislimit can be set by configuring maxlogins in /etc/limits.conf to 5.
# For Linux/Unix and Solaris users, users should be restricted to a maximum number of simultaneous
processes
# they can have running to 20, by setting ulimit -u 20 > /dev/null 2>&1 in /etc/profile.
function 147849-130_Session_Limits()
{
    local _limits_file=/etc/security/limits.conf

    grep -qE "^[[:space:]]*\^[[:space:]]*-[[:space:]]*maxlogins[[:space:]]*5" "${_limits_file}"
    if [ $? -ne 0 ]
    then
        #Not already set
        _backup_file "${_limits_file}"
        grep -qE "^[[:space:]]*\^[[:space:]]*-[[:space:]]*maxlogins[[:space:]]*" "${_limits_file}"
        if [ $? -eq 0 ]

```

```

        then
            #Comment out previous setting
            sed -e "s/^\(.*[[:space:]]*-[[:space:]]*maxlogins[[:space:]]*.*)$/#\1/" -i
${_limits_file}
        fi
        sed -e "/# End of file/i \
*           -           maxlogins           5 " -i ${_limits_file}
        fi
    }

# 147849-170 Timeout inactive sessions - - The system shall provide a customer configurable
inactivity timeout with a default of 10 minutes.
# All Inactive sessions whether for shell or GUI or web access, must be closed once the inactivity
timeout expires
function 147849-170_Timeout_inactive_sessions()
{
    local _file=/etc/profile

    grep -qE "^[[:space:]]*(readonly|export)?[[:space:]]*TMOUT=600[[:space:]]*$" ${_file}
    if [ $? -ne 0 ]
    then
        #Not already set
        _backup_file "${_file}"
        grep -qE "^[[:space:]]*(readonly|export)?[[:space:]]*TMOUT=.*$" ${_file}
        if [ $? -eq 0 ]
        then
            #Comment out previous setting
            sed -e "s/^\(.*[[:space:]]*TMOUT=.*\)$/#\1/g" -i ${_file}
        fi
        echo "readonly TMOUT=600" >> "${_file}"
    fi
}

# 147849-210 Linux: Disable Kudzu - The Kudzu services must be disabled to prevent bootable devices
from being plugged into the system.
function 147849-210_Linux_Disable_Kudzu()
{
    _disable_service "kudzu"
}

function _change_home_dir_perms()
{
    local _user=$1
    local _home_dir=$2

    local _priv_dirs=(/ /bin /dev /sbin /proc /var/empty/sshd) #List of directories we will not
change

    for _priv_dir in "${_priv_dirs[@]}"
    do
        if [ "${_priv_dir}" == "${_home_dir}" ]
        then
            return 0
        fi
    done

    if [ -d "${_home_dir}" ]
    then
        chown "${_user}:" "${_home_dir}"
        chmod 700 "${_home_dir}"
    fi
}

# 147849-120 Unique directory for each user - Each user should have their own directory with the
default permissions for their home directory set to 700 to allow owner access.
function 147849-120_Unique_directory_for_each_user()
{
    awk -F ":" '{print $1 " " $6}' /etc/passwd | while read user home_dir
    do
        _change_home_dir_perms "${user}" "${home_dir}"
        #_change_duplicate_home_dir "${home_dir}"
    done
}

```

```

}

# 147849-220 Disable TCP and ICMP Timestamps:
# ICMP timestamp requests may allow an attacker to determine the time/date set on the server which
# may help the attacker defeat time-based authentication pro-tocols.
function 147849-220_Disable_TCP_and_ICMP_Timestamps()
{
    local _file=/etc/sysctl.conf

    _backup_file "${_file}"
    _change_network_param "net.ipv4.tcp_timestamps" "0" "${_file}"
}

# 147849-080 Remove unneeded accounts - Accounts created by the operating system that are not
# required by the product must be removed as part of the installation.
function 147849-080_Remove_unneeded_accounts()
{
    for account_to_delete in \
        adm \
        ftp \
        games \
        gopher \
        halt \
        lp \
        mail \
        nfsnobody \
        operator \
        shutdown \
        uucp
    do
        grep -q "^${account_to_delete}" /etc/passwd
        if [ $? -eq 0 ]
        then
            userdel ${account_to_delete}
        fi
    done
}

setPrintLastLogInSshdConfig() {
    local sshdConfigBackedUp=$1
    local fileUpdated=false
    echo -n "Checking /etc/ssh/sshd_config for PrintLastLog ... "
    if ! grep -qE
    "^([[:blank:]]*[Pp][Rr][Ii][Nn][Tt][Ll][Aa][Ss][Tt][Ll][Oo][Gg][[:blank:]]*yes[[:blank:]]*$"
    /etc/ssh/sshd_config
    then
        [ "$sshdConfigBackedUp" = false ] && _backup_file /etc/ssh/sshd_config
        sed -i
        's|^([[:blank:]]*[Pp][Rr][Ii][Nn][Tt][Ll][Aa][Ss][Tt][Ll][Oo][Gg][[:blank:]]*)|#\1g'
        /etc/ssh/sshd_config
        echo "PrintLastLog yes" >> /etc/ssh/sshd_config
        echo "entry added, file updated"
        fileUpdated=true
    fi
    [ "$fileUpdated" = false ] && echo "entry present, file not updated"
    [ "$fileUpdated" = true ]
}

setUsePamInSshdConfig() {
    local sshdConfigBackedUp=$1
    local fileUpdated=false
    echo -n "Checking /etc/ssh/sshd_config for UsePAM ... "
    if ! grep -qE
    "^([[:blank:]]*[Uu][Ss][Ee][Pp][Aa][Mm][[:blank:]]*yes[[:blank:]]*$"
    /etc/ssh/sshd_config
    then
        [ "$sshdConfigBackedUp" = false ] && _backup_file /etc/ssh/sshd_config
        sed -i "s|^([[:blank:]]*[Uu][Ss][Ee][Pp][Aa][Mm][[:blank:]]*)|#\1g" /etc/ssh/sshd_config
        echo "UsePAM yes" >> /etc/ssh/sshd_config
        echo "entry added, file updated"
        fileUpdated=true
    fi
    [ "$fileUpdated" = false ] && echo "entry present, file not updated"
}

```

```

    [ "$fileUpdated" = true ]
}

addShowFailedToModule() {
    local file="$1"
    local fileUpdated=false
    echo -n "Checking $file for pam_lastlog.so showfailed ... "
    if ! grep -qE
"^[[[:blank:]]*session[[[:blank:]]*required[[[:blank:]]*pam_lastlog\.so.*[[[:blank:]]]+showfailed($|[[[:blank:]]+)]+)" "$file"
    then
        _backup_file "$file"
        echo "session required pam_lastlog.so showfailed" >> "$file"
        echo "entry added, file updated"
        fileUpdated=true
    fi
    [ "$fileUpdated" = false ] && echo "entry present, file not updated"
    [ "$fileUpdated" = true ]
}

#147849-150 Logon Display - Upon successful logon, the system will display the date and time the last
success-ful logon and then number of failed logon attempts since the last successful log-in.
147849-150_Logon_Display() {
    local changesMade=false

    setPrintLastLogInSshdConfig $changesMade && changesMade=true
    setUsePamInSshdConfig $changesMade && changesMade=true
    addShowFailedToModule /etc/pam.d/sshd && changesMade=true
    addShowFailedToModule /etc/pam.d/login && changesMade=true

    [ "$changesMade" = true ] && service sshd restart
}

#147049-030 Network DoS Protection
#Avaya products shall survive denial of service (DoS) attacks at any packet rate without spontaneous
rebooting, restarting, or reloading, and shall automatically recover to full service after the denial
of service attack is over.
# DoS attacks shall include, at a minimum, but are not limited to the following list:
# SYN Flood (TCP SYN) Attack;
# Land and LaTierra;
# Smurf/Pong; Fraggle;
# Jolt1 & Jolt2;
# Packet Replay attack;
# Ping Flood;
# Finger of Death;
# Chargen Packet Storm;
# OOB Nuke;
# SPANK DoS;
# other Flooding Attacks;
# Replay Attacks;
# Malformed Packet Attacks inc. Teardrop, Overlap, or Fragmented Packets; SNMP Protos;
# H.225v4 PROTOS;
# SDP & SIP PROTOS;
# ARP Attacks
#
# In addition , the following packet types shall be discarded without responding to them:
#IP:
# 1) Datagrams in which the protocol field contains a value for a protocol that is not supported.
# ICMP:
# 1) Time Stamps,
# 2.) Address Mask,
# 3.) Fragments,
# 4.) Information Requests,
# 5.) Broadcast messages,
# 6.) Redirects
# 7) ICMP error messages in response to multicast messages if multicast is not supported.
# TCP:
# 1) TCP Time Stamps IGMP:
# 1) IGMP packets that are not addressed to a multicast IP address
147049-030_Network_Dos_Protection(){
    # Check the service is running and in-memory config is synchronised with disk
    service iptables restart
}

```

```

# Remove any pre-existing rules
iptables -F

# Add a new chain to send all dropped packets to. These is to make administration easier
# A rule is added to this chain to drop anything that is sent to it.
iptables -N DROPPED
# For example, you could log all dropped packets along the following lines:
# iptables -A DROPPED -j LOG --log-prefix "iptables - Dropped packet: " --log-level 4
iptables -A DROPPED -j DROP

# Allow established connections to receive replies
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Block fragments
iptables -A INPUT -f -j DROPPED

# Allow access to the localhost interface
iptables -A INPUT -i lo -j ACCEPT

# Drop null packets
iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROPPED
# Block SYN-flood
iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROPPED
# Block Christmas tree (aka kamikaze, nastygram, lamp test segment) packets
iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROPPED
# Block final packets trying to sync (similar to Christmas tree)
iptables -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROPPED
# Block final packets trying to reset (similar to Christmas tree)
iptables -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROPPED
# Another similar to Christmas tree
iptables -A INPUT -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROPPED
# Block dubious final urgent push messages
iptables -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROPPED
# Block FIN packet scans (All packets after initial SYN should ACK)
iptables -A INPUT -p tcp --tcp-flags FIN,ACK FIN -j DROPPED

# Allow ICMP echo (ping)
iptables -A INPUT -p icmp --icmp-type 8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
# Allow ICMP echo reply (pong)
iptables -A INPUT -p icmp --icmp-type 0 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# Allow SSH access
iptables -A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
# The following is better if we know where we're allowed to connect from:
#iptables -A INPUT -p tcp -s YOUR_IP_ADDRESS -m tcp --dport 22 -j ACCEPT

if [ "$GW_INSTALL" = "y" ] || [ "$GW_INSTALL" = "Y" ]
then
    # Multicast Control for LB and AS
    iptables -A INPUT -m pkttype --pkt-type multicast -j ACCEPT
    # LB Remoting
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 4427 -j ACCEPT
    # AS Remoting
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 4447 -j ACCEPT
    # Txn recovery
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 4712 -j ACCEPT
    # Txn status
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 4713 -j ACCEPT
    # LB internals
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 5065 -j ACCEPT
    iptables -A INPUT -p udp -m state --state NEW -m udp --dport 5065 -j ACCEPT
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 5066 -j ACCEPT
    # AJP Port
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8009 -j ACCEPT
    # LB OSGI
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8070 -j ACCEPT
    # Default LB HTTP port
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8080 -j ACCEPT
    # AS OSG
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8090 -j ACCEPT
    # Default AS HTTP port

```

```

iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8100 -j ACCEPT
# Default AS SNMP port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8161 -j ACCEPT
# Default LB HTTPS port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8443 -j ACCEPT
# Default AS HTTPS port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8463 -j ACCEPT
# Default FAS Management port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 9990 -j ACCEPT
# Default FAS Management REST port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 9999 -j ACCEPT
# Default FAS Management Unsecure port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 9100 -j ACCEPT
# Default FAS Management Secure port
iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 9463 -j ACCEPT
# Default SIP Signalling ports
iptables -A INPUT -p udp -m state --state NEW -m udp -m multiport --dports 5060,5080 -j
ACCEPT
iptables -A INPUT -p tcp -m state --state NEW -m tcp -m multiport --dports
5060,5061,5062,5080,5081,5082 -j ACCEPT
# Cluster ports
iptables -A INPUT -p tcp -m state --state NEW -m tcp -m multiport --dports
7480,7500,7580,7600,45700,57580,57600 -j ACCEPT
iptables -A INPUT -p udp -m state --state NEW -m udp -m multiport --dports
54180,54200,55180,55200 -j ACCEPT
fi
if [ "$MB_INSTALL" = "y" ] || [ "$MB_INSTALL" = "Y" ]
then
    DEFAULT_MB_WEBRTC_PORT="16000"
    DEFAULT_MB_SIP_PORT_RANGE="17000:17099"

    read -p "Please enter the port that will be configured for WEBRTC side MB traffic [default
$DEFAULT_MB_WEBRTC_PORT]: " MB_WEBRTC_PORT
    read -p "Please enter the port range (in the form min-port:max-port), that will be configured
for SIP side MB traffic [default $DEFAULT_MB_SIP_PORT_RANGE]: " MB_SIP_PORT_RANGE

    if [ "$MB_WEBRTC_PORT" = "" ]
    then
        MB_WEBRTC_PORT=$DEFAULT_MB_WEBRTC_PORT
    fi

    if [ "$MB_SIP_PORT_RANGE" = "" ]
    then
        MB_SIP_PORT_RANGE=$DEFAULT_MB_SIP_PORT_RANGE
    fi

    # Default Media Broker Control Port
    iptables -A INPUT -p tcp -m state --state NEW -m tcp --dport 8092 -j ACCEPT
    # Default WEBRTC Media Port
    iptables -A INPUT -p udp -m state --state NEW -m udp --dport $MB_WEBRTC_PORT -j ACCEPT
    # Default SIP Media Ports
    iptables -A INPUT -p udp -m state --state NEW -m udp --dport $MB_SIP_PORT_RANGE -j ACCEPT
fi
iptables -A INPUT -j DROPPED

# Accept all outgoing traffic
iptables -P OUTPUT ACCEPT
# Block all incoming traffic
iptables -P INPUT ACCEPT
# Block all forwarding traffic
iptables -P FORWARD DROP

# Save the rules
service iptables save
# Restart the service
service iptables restart
# List rules
service iptables status

# Start iptables with the OS
chkconfig iptables on
}

```

147849-140\_Logon\_Warning\_Banner  
147849-191\_SSH\_Service\_hardenening  
147849-240\_Disable\_insecure\_network\_services  
147849-240\_Disable\_insecure\_network\_xinetd\_services  
147849-240\_Disable\_insecure\_xwindows\_service  
147849-250\_Linux\_networking\_parameters  
147849-040\_Single\_user\_mode  
147849-130\_Session\_Limits  
147849-170\_Timeout\_inactive\_sessions  
147849-210\_Linux\_Disable\_Kudzu  
147849-080\_Remove\_unneeded\_accounts  
147849-120\_Unique\_directory\_for\_each\_user  
147849-220\_Disable\_TCP\_and\_ICMP\_Timestamps  
147849-150\_Logon\_Display  
147049-030\_Network\_Dos\_Protection



# Appendix B: Glossary

Item	Description
Avaya Aura® Communication Manager	The Avaya telecommunications system used for unified communications and collaboration.
Avaya one-X® Agent	A desktop application for contact center agents and supervisors.
Avaya SBCE	Avaya Session Border Control for Enterprise—a reverse proxy server that retrieves resources on behalf of a client from one or more servers. These resources are then returned to the client as though they originated from the SBCE itself.
AMVS	<p>Avaya Mobile Video Server—platform for delivering web applications to make voice and video calls directly from a Web browser, iOS device, or Android device, to an Avaya one-X Agent.</p> <p>The AMVS Web Administration interface is used to configure the services facilitating this communication.</p>
CAC	Call Admission Control
CIDR	<p>Classless Inter-Domain Routing. CIDR notation is a compact representation of an IP address and its associated routing prefix. The notation is constructed from an IP address, a slash (/) character, and a decimal number representing the network mask, for example:</p> <p>192.0.2.0/24</p>
CODEC	“Coder-decoder” encodes a data stream or signal for transmission and decodes it for playback in voice over IP and video conferencing applications.
DMZ	A demilitarized zone (also referred to as a perimeter network) is a physical or logical subnetwork that contains and exposes an organization's external-facing services to the Internet.
FQDN	<p>A fully qualified domain name—the complete domain name for a specific computer, or host, on the Internet, for example</p> <p>examplehost.example.com</p>
G.711	PCMU/A 8-bit audio codec used for base telephony applications.
G.729a	Low bit rate audio codec for VoIP applications.

Item	Description
H.264	Video codec. H.264 is the dominant video compression technology, or codec, in industry that was developed by the International Telecommunications Union (as H.264 and MPEG-4 Part 10, Advanced Video Coding, or AVC).
Media Broker	Intercepts SDP messages, performs transcoding where required, and can remove any banned codes. Multiple Media Brokers can be installed on the same network, for load balancing and scaling.
MVSDK	Mobile Video SDKs. Includes three distinct SDKs for iOS, Android and web/JavaScript developers.
MVSDK Client	A web/JavaScript, iOS, or Android client with which a connection is established using the MVSDK.
Opus	Low bit rate, high definition audio codec for VoIP applications. See RFC 6716.
Ping	Query (ICMP echo request) made to another computer on a network to determine whether there is a connection to it.
PLI	A feedback mechanism of the Real-time Transport Control Protocol (RTCP) which enables the sender to resend keyframe packets to re-establish a full video picture when communicating over the Internet or poor network conditions.
Pong	A response made to a Ping request, confirming that a connection exists.
REMB	Receiver Estimated Maximum Bitrate
RTCP	Real-time Transport Control Protocol
RTP	Real-time Transport Protocol
SDP	Session Description Protocol
SIP	Session Initiation Protocol. SIP is a VoIP call setup protocol that operates at the application layer. It sets up calls that then use RTP to actually send the voice data between phones.
TIMMBR	Temporary Maximum Media-Stream Bit Rate Request
UC	Unified Communications
VP8	Video codec. VP8 is a video compression format owned by Google. VP8 is roughly equivalent in processor usage, bandwidth and quality to H.264.
Web Gateway	Permits users to make calls to one-X agent endpoints.

Item	Description
WebRTC	Web Real Time Communications for communications without plug-ins.
WebSockets	A protocol providing full-duplex communication channels over a single TCP connection, standardized by the IETF as RFC 6455.