BIG-IP® System: SSL Administration

Version 11.5

Table of Contents

Legal Notices

Publication Date

This document was published on August 29, 2014.

Publication Number

MAN-0527-00

Copyright

Trademarks

Patents

This product may be protected by one or more patents indicated at:
http://www.f5.com/about/guidelines-policies/patents

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes software with glib library utility functions, which is protected under the GNU Public License.

This product includes software with grub2 bootloader functions, which is protected under the GNU Public License.

This product includes software with the Intel Gigabit Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes software with the Intel 10 Gigabit PCI Express Linux driver, which is protected under the GNU Public License. Copyright ©1999 - 2012 Intel Corporation.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes software developed by Andrew Tridgell, which is protected under the GNU Public License, copyright ©1992-2000.

This product includes software developed by Jeremy Allison, which is protected under the GNU Public License, copyright ©1998.

This product includes software developed by Guenther Deschner, which is protected under the GNU Public License, copyright ©2008.

This product includes software developed by www.samba.org, which is protected under the GNU Public License, copyright ©2007.

This product includes software from Allan Jardine, distributed under the MIT License.

This product includes software from Trent Richardson, distributed under the MIT License.

This product includes vmbus drivers distributed by Microsoft Corporation.

This product includes software from Cavium.

This product includes software from Webroot, Inc.

This product includes software from Maxmind, Inc.

This product includes software from OpenVision Technologies, Inc. Copyright ©1993-1996, OpenVision Technologies, Inc. All Rights Reserved.

This product includes software developed by Matt Johnson, distributed under the MIT License. Copyright ©2012.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software from NLnetLabs. Copyright ©2001-2006. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of NLnetLabs nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes GRand Unified Bootloader (GRUB) software developed under the GNU Public License, copyright ©2007.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes gd-libgd library software developed by the following in accordance with the following copyrights:

- Portions copyright ©1994, 1995, 1996, 1997, 1998, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.
- Portions copyright ©1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.
- Portions relating to GD2 format copyright ©1999, 2000, 2001, 2002 Philip Warner.
- Portions relating to PNG copyright ©1999, 2000, 2001, 2002 Greg Roelofs.
- Portions relating to gdttf.c copyright ©1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).
- Portions relating to gdft.c copyright ©2001, 2002 John Ellson (ellson@lucent.com).
- Portions copyright ©2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 2008 Pierre-Alain Joye (pierre@libgd.org).
- Portions relating to JPEG and to color quantization copyright ©2000, 2001, 2002, Doug Becker and copyright ©1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group.
- Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande. Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This product includes software developed by Oracle America, Inc. Copyright ©2012.

1. Java Technology Restrictions. Licensee shall not create, modify, change the behavior of, or authorize licensees of licensee to create, modify, or change the behavior of, classes, interfaces, or subpackages that are in any way identified as "java", "javax", "sun" or similar convention as specified by Oracle in any naming convention designation. In the event that Licensee creates an additional API(s) which: (a) extends the functionality of a Java Environment; and (b) is exposed to third party software developers for the purpose of developing additional software which invokes such additional API, Licensee must promptly publish broadly an accurate specification for such API for free use by all developer.
2. Trademarks and Logos. This License does not authorize an end user licensee to use any Oracle America, Inc. name, trademark, service mark, logo or icon. The end user licensee acknowledges that Oracle owns the Java trademark and all Java-related trademarks, logos and icon including the Coffee Cup and Duke ("Java Marks") and agrees to: (a) comply with the Java Trademark Guidelines at http://www.oraclc.com/html/3party.html; (b) not do anything harmful to or inconsistent with Oracle's rights in the Java Marks; and (c) assist Oracle in protecting those rights, including assigning to Oracle any rights acquired by Licensee in any Java Mark.
3. Source Code. Software may contain source code that, unless expressly licensed for other purposes, is provided solely for reference purposes pursuant to the terms of your license. Source code may not be redistributed unless expressly provided for in the terms of your license.
4. Third Party Code. Additional copyright notices and license terms applicable to portion of the Software are set forth in the THIRDPARTYLICENSEREADME.txt file.
5. Commercial Features. Use of the Commercial Features for any commercial or production purpose requires a separate license from Oracle. "Commercial Features" means those features identified in Table I-I (Commercial Features In Java SE Product Editions) of tile Software documentation accessible at http://www.oracle.com/technetwork/java/javase/documentation/index.html.

This product includes utilities developed by Linus Torvalds for inspecting devices connected to a USB bus.

This product includes perl-PHP-Serialization software, developed by Jesse Brown, copyright ©2003, and distributed under the Perl Development Artistic License (http://dev.perl.org/licenses/artistic.html).

This product includes software developed by members of the CentOS Project under the GNU Public License, copyright ©2004-2011 by the CentOS Project.

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software licensed from Rémi Denis-Courmont under the GNU Library General Public License. Copyright ©2006 - 2011.

This product includes software developed by jQuery Foundation and other contributors, distributed under the MIT License. Copyright ©2014 jQuery Foundation and other contributors (http://jquery.com/).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Trent Richardson, distributed under the MIT License. Copyright ©2012 jQuery Foundation and other contributors (http://jquery.com/).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Allan Jardine, distributed under the MIT License. Copyright ©2008 - 2012, Allan Jardine, all rights reserved, jQuery Foundation and other contributors (http://jquery.com/).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,

OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Douglas Gilbert. Copyright ©1992 - 2012 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

This product includes software developed as open source software. Copyright ©1994 - 2012 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of the authors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). Copyright ©1998 - 2011 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software licensed from William Ferrell, Selene Scriven and many other contributors under the GNU General Public License, copyright ©1998 - 2006.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. Copyright ©1990-1994 Regents of the University of California. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.

4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes the ixgbevf Intel Gigabit Linux driver, Copyright © 1999 - 2012 Intel Corporation, and distributed under the GPLv2 license, as published by the Free Software Foundation.

This product includes libwebp software. Copyright © 2010, Google Inc. All rights reserved.

This product includes Angular software developed by Google, Inc., http://angulargs.org, copyright © 2010-2012 Google, Inc., and distributed under the MIT license.

This product includes node.js software, copyright © Joyent, Inc. and other Node contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product MAY include Intel SSD software subject to the following license; check your hardware specification for details.

1. LICENSE. This Software is licensed for use only in conjunction with Intel solid state drive (SSD) products. Use of the Software in conjunction with non-Intel SSD products is not licensed hereunder. Subject to the terms of this Agreement, Intel grants to You a nonexclusive, nontransferable, worldwide, fully paid-up license under Intel's copyrights to:

- copy the Software onto a single computer or multiple computers for Your personal, noncommercial use; and
- make appropriate back-up copies of the Software, for use in accordance with Section 1a) above.

The Software may contain the software or other property of third party suppliers, some of which may be identified in, and licensed in accordance with, any enclosed "license.txt" file or other text or file.

Except as expressly stated in this Agreement, no license or right is granted to You directly or by implication, inducement, estoppel or otherwise. Intel will have the right to inspect or have an independent auditor inspect Your relevant records to verify Your compliance with the terms and conditions of this Agreement.

2. RESTRICTIONS. You will not:

a. copy, modify, rent, sell, distribute or transfer any part of the Software, and You agree to prevent unauthorized copying of the Software; and,

b. reverse engineer, decompile, or disassemble the Software; and,

c. sublicense or permit simultaneous use of the Software by more than one user; and,

d. otherwise assign, sublicense, lease, or in any other way transfer or disclose Software to any third party, except as set forth herein; and,

e. subject the Software, in whole or in part, to any license obligations of Open Source Software including without limitation combining or distributing the Software with Open Source Software in a manner that subjects the Software or any portion of the Software provided by Intel hereunder to any license obligations of such Open Source Software. "Open Source Software" means any software that requires as a condition of use, modification and/or distribution of such software that such software or other software incorporated into, derived from or distributed with such software:

a. be disclosed or distributed in source code form; or
b. be licensed by the user to third parties for the purpose of making and/or distributing derivative works; or
c. be redistributable at no charge.

Open Source Software includes, without limitation, software licensed or distributed under any of the following licenses or distribution models, or licenses or distribution models substantially similar to any of the following:

a. GNU's General Public License (GPL) or Lesser/Library GPL (LGPL),
b. the Artistic License (e.g., PERL),
c. the Mozilla Public License,
d. the Netscape Public License,
e. the Sun Community Source License (SCSL),
f. vi) the Sun Industry Source License (SISL),
g. vii) the Apache Software license, and
h. viii) the Common Public License (CPL).

3. OWNERSHIP OF SOFTWARE AND COPYRIGHTS. Title to all copies of the Software remains with Intel or its suppliers. The Software is copyrighted and protected by the laws of the United States and other countries, and international treaty provisions. You may not remove any copyright notices from the Software. Intel may make changes to the Software, or to materials referenced therein, at any time and without notice, but is not obligated to support or update the Software. Except as otherwise expressly provided, Intel grants no express or implied right or license under Intel patents, copyrights, trademarks, or other intellectual property rights.
4. Entire Agreement. This Agreement contains the complete and exclusive statement of the agreement between You and Intel and supersedes all proposals, oral or written, and all other communications relating to the subject matter of this Agreement. Only a written instrument duly executed by authorized representatives of Intel and You may modify this Agreement.
5. LIMITED MEDIA WARRANTY. If the Software has been delivered by Intel on physical media, Intel warrants the media to be free from material physical defects for a period of ninety (90) days after delivery by Intel. If such a defect is found, return the media to Intel for replacement or alternate delivery of the Software as Intel may select.
6. EXCLUSION OF OTHER WARRANTIES. EXCEPT AS PROVIDED ABOVE, THE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND, INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, OR FITNESS FOR A PARTICULAR PURPOSE. Intel does not warrant or assume responsibility for any errors, the accuracy or completeness of any information, text, graphics, links or other materials contained within the Software.
7. LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION OR LOST INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS PROHIBIT EXCLUSION OR LIMITATION OF LIABILITY FOR IMPLIED WARRANTIES OR CONSEQUENTIAL OR INCIDENTAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. YOU MAY ALSO HAVE OTHER LEGAL RIGHTS THAT VARY FROM JURISDICTION TO JURISDICTION.
8. TERMINATION OF THIS AGREEMENT. Intel may terminate this Agreement at any time if You violate its terms. Upon termination, You will immediately destroy the Software or return all copies of the Software to Intel.
9. APPLICABLE LAWS. Claims arising under this Agreement will be governed by the laws of Delaware, excluding its principles of conflict of laws and the United Nations Convention on Contracts for the Sale

of Goods. You may not export the Software in violation of applicable export laws and regulations. Intel is not obligated under any other agreements unless they are in writing and signed by an authorized representative of Intel.

10. GOVERNMENT RESTRICTED RIGHTS. The Software is provided with "RESTRICTED RIGHTS." Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or their successors. Use of the Software by the Government constitutes acknowledgment of Intel's proprietary rights therein. Contractor or Manufacturer is Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95054.

Chapter 1

About SSL Administration on the BIG-IP System

About SSL administration on the BIG-IP system

The BIG-IP® system offers a robust set of features for managing SSL traffic. With the BIG-IP system, you can:

- Install, create, and manage digital certificates on the BIG-IP system for communication with other devices on the network.
- Optimize SSL application traffic by terminating incoming client connections and re-initiating server connections. By terminating client connections, the BIG-IP system can optimize and manipulate the data in user-defined ways before sending the data on to the target server.
- Offload client authentication and encryption/decryption tasks from the target server.

Chapter 2

Digital Certificate Management

About SSL digital certificates on the BIG-IP system

An *SSL digital certificate* is an electronic key pair that allows devices on a network to exchange data securely, using the public key infrastructure (PKI). PKI is based on public and private cryptographic key pairs used to encrypt and decrypt messages sent between two devices.

The BIG-IP® system uses digital certificates with the SSL/TLS protocol to grant authentication to clients on the external network that are generally untrusted. In high-security environments, the BIG-IP system can also use certificates to communicate securely with other systems on the internal network, such as web servers and other BIG-IP systems.

The BIG-IP system can sign a digital certificate in either of two ways:

- By generating and submitting a request to a third-party trusted certificate authority (CA)
- By creating a self-signed certificate. Self-signed certificates are typically used for testing purposes.

Once a certificate is installed or created on the BIG-IP system, other BIG-IP administrative users can specify those certificates in BIG-IP SSL profiles to manage SSL application traffic. Moreover, the BIG-IP system uses digital certificates to establish device trust in device service clustering (DSC™) configurations.

Supported certificate/key types

The BIG-IP® system supports multiple cipher suites when offloading SSL operations from a target server on the network. The BIG-IP system can support cipher suites that use these algorithms:

- Rivest Shamir Adleman (RSA)
- Elliptic Curve Digital Signature Algorithm (ECDSA)
- Digital Signature Algorithm (DSA)

When you generate a certificate request or a self-signed certificate, you specify the type of private key, which determines that specific signing or encryption algorithm that is used to generate the private key.

About RSA certificates

RSA (Rivest Shamir Adleman) is the original encryption algorithm that is based on the concept of a public and a private key. When a public site attempts to communicate with a device such as the BIG-IP® system, the device sends the site a public key that the site uses to encrypt data before sending that data back to the device. The device uses its private key associated with the public key to decrypt the data. Only the device on which the certificate resides has access to this private key.

The RSA encryption algorithm includes an authentication mechanism.

About DSA certificates

DSA (Digital Signature Algorithm) uses a different algorithm for signing key exchange messages than that of RSA. DSA is paired with a key exchange method such as Diffie-Hellman or Elliptical Curve Diffie-Hellman to achieve a comparable level of security to RSA. Because DSA is generally endorsed by federal agencies, specifying a DSA key type makes it easier to comply with new government standards, such as those for specific key lengths.

About ECDSA certificates

When creating certificates on the BIG-IP® system, you can create a certificate with a key type of ECDSA (Elliptic Curve Digital Signature Algorithm). An *ECDSA key* is based on Elliptic Curve Cryptography (ECC), and provides better security and performance with significantly shorter key lengths.

For example, an RSA key size of 2048 bits is equivalent to an ECC key size of only 224 bits. As a result, less computing power is required, resulting in faster, more secure connections. Encryption based on ECC is ideally suited for mobile devices that cannot store large keys. The BIG-IP system supports both the prime256v1 and secp384r1 curve names, although only prime256v1 can be associated with an SSL profile.

About certificate management

You can obtain a certificate for the BIG-IP system by using the BIG-IP® Configuration utility to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA). The CA then issues a signed certificate.

In addition to requesting CA-signed certificates, you can create self-signed certificates. You create self-signed certificates primarily for testing purposes within an organization.

When you install the BIG-IP software, the application includes a default self-signed certificate. The BIG-IP system also includes a default CA bundle certificate. This certificate bundle contains certificates from most of the well-known CAs.

Note: *To manage digital certificates for the BIG-IP system, you must have a role of Certificate Manager, Administrator, or Resource Administrator assigned to your BIG-IP user account.*

Creating a self-signed digital certificate

If you are configuring the BIG-IP® system to manage client-side HTTP traffic, you perform this task to create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you must repeat this task to create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.

13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select a key type.
    Possible values are: **RSA**, **DSA**, and **ECDSA**.
15. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
16. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
17. Click **Finished**.

Requesting a certificate from a certificate authority

You perform this task to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA).

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select a key type.
    Possible values are: **RSA**, **DSA**, and **ECDSA**.
17. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
18. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
19. Click **Finished**.
    The Certificate Signing Request screen displays.
20. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
21. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
22. Click **Finished**.

The Certificate Signing Request screen displays.

The generated certificate signing request is submitted to a trusted certificate authority for signature.

Importing a certificate signed by a certificate authority

Before performing this task, confirm that a digital certificate signed by a certificate authority is available.

You can install an SSL certificate that is signed by a certificate authority by importing a certificate that already exists on the system hard drive. You can import a private key, a certificate or certificate bundle, or an archive.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Import**.
3. From the **Import Type** list, select **Certificate**.
4. For the **Certificate Name** setting, do one of the following:
   - Select the **Create New** option, and type a unique name in the field.
   - Select the **Overwrite Existing** option, and select a certificate name from the list.
5. For the **Certificate Source** setting, do one of the following:
   - Select the **Upload File** option, and browse to the location of the certificate file.
   - Select the **Paste Text** option, and paste the certificate text copied from another source.
6. Click **Import**.

The SSL certificate that was signed by a certificate authority is installed.

Exporting a digital certificate

You perform this task to export a digital certificate to another device.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click the name of the certificate you want to export.
   The General Properties screen displays.
3. Click **Export**.
   The Certificate Export screen displays the contents of the certificate in the **Certificate Text** box.
4. To obtain the certificate, do one of the following:
   - Copy the text from the **Certificate Text** field, and paste it as needed into an interface on another system.
   - At the **Certificate File** option, click **Download filename** where filename is the name of the certificate file, such as `mycert.crt`.

Viewing a list of certificates on the system

You can perform this task to view a list of existing digital certificates on the BIG-IP® system.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. In the Name column, view the list of certificates on the system.

Digital certificate properties

When you use the BIG-IP® Configuration utility to view the list of digital certificates that you have installed on the BIG-IP® system, you can see information for each certificate.

Property	Description
Certificate	The name of the certificate.
Content	The type of certificate content, for example, Certificate Bundle or Certificate and Key.
Common name	The common name (CN) for the certificate. The common name embedded in the certificate is used for name-based authentication. The default common name for a self-signed certificate is `localhost.localdomain`.
Expiration date	The date that the certificate expires. If the certificate is a bundle, this information shows the range of expiration dates that apply to certificates in the bundle.
Organization	The organization name for the certificate. The organization name embedded in the certificate is used for name-based authentication. The default organization for a self-signed certificate is `MyCompany`.

Chapter 3

SSL Traffic Management

About SSL traffic management

You can manage the way that the BIG-IP system processes SSL application traffic by configuring two types of SSL profiles: A Client SSL profile, a Server SSL profile, or both. These profiles affect the way that the system manages SSL traffic passing through the system.

When you configure Client SSL or Server SSL profiles and assign them to a virtual server, the BIG-IP system offloads SSL processing from the destination server. This offloading not only conserves resource on destination servers, but enables the BIG-IP system to customize SSL traffic processing according to your configuration specifications.

About SSL offload

When you want the BIG-IP system to process application traffic over SSL, you can configure the system to perform the SSL handshake that destination servers normally perform. This ability for the BIG-IP system to offload SSL processing from a destination server is an important feature of the BIG-IP system.

The most common way to configure the BIG-IP system is to create a Client SSL profile, which makes it possible for the BIG-IP system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client.

Within a Client SSL profile specifically, you can specify multiple certificate/key pairs, one per key type. This enables the system to accept all types of cipher suites that a client might support as part of creating a secure connection. The system then decrypts the client data, manipulates any headers or payload according to the way that you configured the Client SSL profile, and by default, sends the request in clear text to the target server for processing.

For those sites that require enhanced security on their internal network, you can configure a Server SSL profile. With a Server SSL profile, the BIG-IP system re-encrypts the request before sending it to the destination server. When the server returns an encrypted response, the BIG-IP system decrypts and then re-encrypts the response, before sending the response back to the client.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select a certificate name.
      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, `default`.
   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.
      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, you can specify the name of an existing key, `default`.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.
      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.
   e) Click **Add** and repeat the process for all certificate key chains that you want to specify.

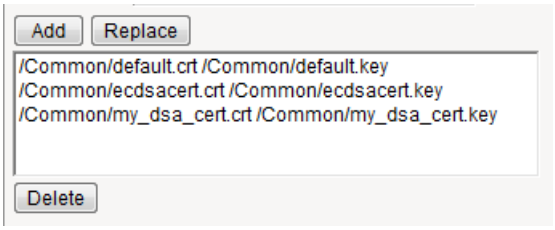


Figure 1: Sample configuration with three key types specified

      The result is that all specified key chains appear in the box.

7. If you want to use a cipher suite other than `DEFAULT`:
   a) From the Configuration list, select **Advanced**.
   b) For the **Ciphers** setting, type the name of a cipher.
      You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.
      Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.
8. Configure all other profile settings as needed.
9. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With an Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The SSL Server profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.
9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.
10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. If you want to use a cipher suite other than `DEFAULT`:
    a) From the Configuration list, select **Advanced**.
    b) For the **Ciphers** setting, type the name of a cipher.

       You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.

       Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.
12. Select the **Custom** check box for **Server Authentication**.
13. Modify the settings, as required.
14. Click **Finished**.

After performing this task, you can see the custom Server SSL profile in the list of Server SSL profiles on the system.

You must also assign the profile to a virtual server.

Assigning SSL profiles to a virtual server

The final task in the process of implementing SSL profiles is to assign the SSL profile to a virtual server. If the relevant virtual server does not yet exist, you can assign the SSL profile (or profiles) to the virtual server when you create it.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the name of a virtual server.

3. From the **Configuration** list, select **Advanced**.
4. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
5. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
6. Click **Update** to save the changes.

After you perform this task, you must assign the profile to a virtual server.

Cipher support on the BIG-IP system

The BIG-IP® system supports a large set of cipher suites that offer various combinations of encryption algorithms and authentication mechanisms, including RSA (Rivest Shamir Adleman), DSA (Digital Signature Algorithm), and ECDSA (Elliptic Curve Digital signature Algorithm).

Within an SSL profile, you can specify either a particular string to indicate the cipher suites that you want the BIG-IP system to use or not use for SSL negotiation. If you do not specify a cipher string, the BIG-IP system uses the default cipher string, `DEFAULT`. Examples of cipher values that you can specify are: `ECDHE`, or `DEFAULT:!ECDHE`. The `ECDHE` and `DEFAULT:!ECDHE` values instruct the BIG-IP system to either negotiate with elliptic curve Diffie-Hellman Ephemeral (DHE) cipher suites, or negate the use of those cipher suites.

It is important to note that if you are assigning both a Client SSL and a Server SSL profile to the virtual server, the connections on each side of the BIG-IP system must use common ciphers. Otherwise, the handshake between the virtual server and the server fails and the connection closes.

The list of supported ciphers that you can specify varies depending on whether the transport layer being used is TCP or UDP.

Note: *The following are not included in the DEFAULT cipher suite:*

- The DHE cipher suites
- Elliptic curve ciphers with DSA

Viewing the list of supported ciphers

Before using this command, confirm that your user account grants you access to the advanced shell.

You perform this task when you want to display the full set of ciphers that the BIG-IP system supports.

1. Access the advanced shell on the BIG-IP system.
2. At the system prompt, type this command: `tmm --clientciphers all`
   The BIG-IP system displays the list of all supported ciphers.

Support for multiple key types

For client-side traffic specifically, you can configure a Client SSL profile to specify multiple certificate key chains on the BIG-IP® system, one for each key type: RSA, DSA, and ECDSA. By configuring a Client

SSL profile with different digital certificates and keys, the system can accept all types of cipher suites that clients might request as part of creating a secure connection.

Important: *To ensure successful negotiation, the BIG-IP system requires you to specify an RSA-based certificate key chain at a minimum, to accommodate any RSA-based ciphers that the client presents. However, F5 Networks highly recommends that you also specify DSA and ECDSA certificate key chains.*

About the DEFAULT cipher suite

The BIG-IP system supports a large suite of ciphers. Some of these ciphers are included in a default cipher suite named `DEFAULT`. The `DEFAULT` cipher suite appears as the default value in the **Ciphers** setting of the Client SSL and Server SSL profiles.

If you want the BIG-IP to accept ciphers that are not included in the `DEFAULT` cipher suite, or you want the system to reject ciphers that are included in the `DEFAULT` cipher suite, you can configure an SSL profile accordingly.

Viewing the list of DEFAULT ciphers

Before using this command, confirm that your user account grants you access to the advanced shell.

You perform this task when you want to display the entire list of ciphers included in the DEFAULT cipher set.

1. Access the advanced shell on the BIG-IP system.
2. At the system prompt, type this command: `tmm --clientciphers DEFAULT`
   The BIG-IP system displays a list of the ciphers included in the DEFAULT cipher set.

RSA ciphers in the DEFAULT cipher suite

This table lists the RSA ciphers in the DEFAULT cipher suite that include AES, DES, and RC4 ciphers.

Table 1: RSA ciphers in the DEFAULT cipher suite

Suite	Bits	Protocol	Cipher	MAC	Key Type
AES256-SHA256	256	TLS1.2	AES	SHA256	RSA
AES256-SHA	256	TLS1	AES	SHA	RSA
AES256-SHA	256	TLS1.1	AES	SHA	RSA
AES256-SHA	256	TLS1.2	AES	SHA	RSA
AES256-SHA	256	DTLS1	AES	SHA	RSA
AES128-SHA256	128	TLS1.2	AES	SHA256	RSA
AES128-SHA	128	TLS1	AES	SHA	RSA
AES128-SHA	128	TLS1.1	AES	SHA	RSA
AES128-SHA	128	TLS1.2	AES	SHA	RSA
AES128-SHA	128	DTLS1	AES	SHA	RSA
DES-CBC3-SHA	192	TLS1	DES	SHA	RSA
DES-CBC3-SHA	192	TLS1.1	DES	SHA	RSA

Suite	Bits	Protocol	Cipher	MAC	Key Type
DES-CBC3-SHA	192	TLS1.2	DES	SHA	RSA
DES-CBC3-SHA	192	DTLS1	DES	SHA	RSA
RC4-SHA	128	TLS1	RC4	SHA	RSA
RC4-SHA	128	TLS1.1	RC4	SHA	RSA
RC4-SHA	128	TLS1.2	RC4	SHA	RSA

ECDHE ciphers in the DEFAULT cipher suite

This table lists all ECDHE ciphers in the `DEFAULT` cipher suite. In the `DEFAULT` cipher suite, Elliptic Curve (EC) ciphers are the only ciphers that include DHE as the key exchange method.

Table 2: ECDHE ciphers in the DEFAULT cipher suite

Suite	Bits	Protocol	Cipher	MAC	Key Type
ECDHE-RSA-AES256-SHA384	256	TLS1.2	AES	SHA384	ECDHE_RSA
ECDHE-RSA-AES256-CBC-SHA	256	TLS1	AES	SHA	ECDHE_RSA
ECDHE-RSA-AES256-CBC-SHA	256	TLS1.1	AES	SHA	ECDHE_RSA
ECDHE-RSA-AES256-CBC-SHA	256	TLS1.2	AES	SHA	ECDHE_RSA
ECDHE-RSA-AES128-SHA256	128	TLS1.2	AES	SHA56	ECDHE_RSA
ECDHE-RSA-AES128-CBC-SHA	128	TLS1	AES	SHA	ECDHE_RSA
ECDHE-RSA-AES128-CBC-SHA	128	TLS1.1	AES	SHA	ECDHE_RSA
ECDHE-RSA-AES128-CBC-SHA	128	TLS1.2	AES	SHA	ECDHE_RSA
ECDHE-RSA-DES-CBC3-SHA	192	TLS1	DES	SHA	ECDHE_RSA
ECDHE-RSA-DES-CBC3-SHA	192	TLS1.1	DES	SHA	ECDHE_RSA
ECDHE-RSA-DES-CBC3-SHA	192	TLS1.2	DES	SHA	ECDHE_RSA

About Diffie-Hellman Ephemeral key exchange

The BIG-IP® system supports the Diffie-Hellman Ephemeral key exchange method, as well as other Diffie-Hellman variations. A *Diffie–Hellman key exchange method* is an alternative to RSA key exchange and allows the client and the BIG-IP system to establish a shared secret session key to use for communication.

Supported Diffie-Hellman variations

The BIG-IP system supports all three Diffie-Hellman key exchange methods. They are:

Diffie-Hellman Ephemeral (DHE)

Diffie-Hellman Ephemeral uses temporary public keys. The authenticity of a temporary key can be verified by checking the digital signature included in the key exchange messages. The key exchange messages are signed using either the RSA or DSA algorithms, depending on the cipher being used. For example, DHE-RSA uses RSA to sign the key exchange messages. DHE includes Perfect Forward Secrecy (PFS), which means that a compromise of the system's long-term signing key does not affect the privacy of past sessions. Like FIPS, DHE prevents private key disclosure.

Diffie-Hellman (DH)
Diffie-Hellman embeds the system's public parameter in the certificate, and the CA then signs the certificate. That is, the certificate contains the Diffie-Hellman public-key parameters, and those parameters never change.

Anonymous Diffie-Hellman (ADH)
Anonymous Diffie-Hellman uses DH, but without authentication. The keys used in the exchange are not authenticated, resulting in keys being susceptible to security attacks.

About Perfect-Forward-Privacy

The Diffie-Hellman Ephemeral (DHE) key exchange method provides Perfect Forward Privacy (PFP). With standard Diffie-Hellman, multiple key exchanges all use the same session key, which can compromise security. By contrast, DHE uses *PFP*, which generates a disposable key per session and thereby ensures that the same session key is never used twice. No key remains to be disclosed, and if the private key of the server is discovered, past communication remains secure.

About DHE cipher support

Because Diffie-Hellman key exchange methods do not include authentication, use of Diffie-Hellman Ephemeral (DHE) requires that it be paired with an authentication mechanism. The DHE ciphers that the BIG-IP system supports are:

- DHE-RSA-* (Diffie-Hellman Ephemeral-RSA)
- DHE-DSS-* (Diffie-Hellman Ephemeral-DSS)
- ECDHE-RSA-* (Elliptic Curve Diffie-Hellman Ephemeral-RSA)
- ECDHE-ECDSA-* (Elliptic Curve Diffie-Hellman Ephemeral-DSA)

Note: *For DHE, the* `DEFAULT` *cipher suite includes Elliptic Curve cipher suites only. DHE ciphers for RSA and DSS encryption are not included.*

Viewing a list of supported DHE ciphers

Before using this command, confirm that your user account grants you access to the advanced shell.

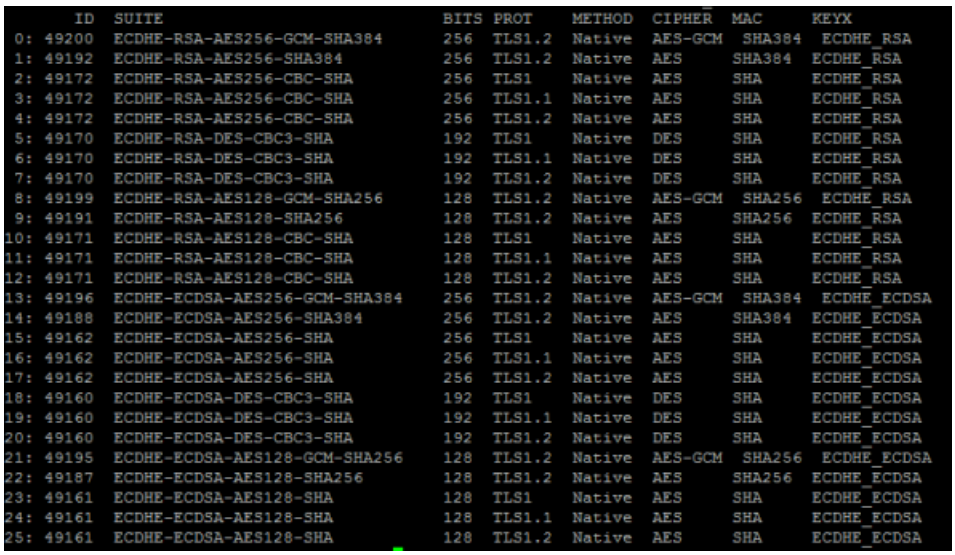
You perform this task when you want to display a specific set of ciphers that the BIG-IP system supports.

1. Access the advanced shell on the BIG-IP system.
2. At the system prompt, type the command `tmm --clientciphers` *ciphers*.
   a) For example, to see a list of DHE+DES ciphers, type `tmm --clientciphers DHE:DHE_DSS`
   The BIG-IP system displays the list of all DHE+DES ciphers that the BIG-IP system supports:

```
      ID  SUITE                         BITS PROT    METHOD  CIPHER  MAC     KEY
0:    21  DHE-RSA-DES-CBC-SHA             64  SSL3    Native  DES     SHA     EDH
1:    21  DHE-RSA-DES-CBC-SHA             64  TLS1    Native  DES     SHA     EDH
2:    21  DHE-RSA-DES-CBC-SHA             64  TLS1.1  Native  DES     SHA     EDH
3:    21  DHE-RSA-DES-CBC-SHA             64  TLS1.2  Native  DES     SHA     EDH
```

Figure 2: Supported DHE+DES ciphers on the BIG-IP system

   b) To see a list of ECDHE ciphers, type `tmm --clientciphers ECDHE:ECDHE_ECDSA`.
   The BIG-IP system displays the list of all ECDHE ciphers that the BIG-IP system supports:



	ID	SUITE	BITS	PROT	METHOD	CIPHER	MAC	KEYX
0:	49200	ECDHE-RSA-AES256-GCM-SHA384	256	TLS1.2	Native	AES-GCM	SHA384	ECDHE_RSA
1:	49192	ECDHE-RSA-AES256-SHA384	256	TLS1.2	Native	AES	SHA384	ECDHE_RSA
2:	49172	ECDHE-RSA-AES256-CBC-SHA	256	TLS1	Native	AES	SHA	ECDHE_RSA
3:	49172	ECDHE-RSA-AES256-CBC-SHA	256	TLS1.1	Native	AES	SHA	ECDHE_RSA
4:	49172	ECDHE-RSA-AES256-CBC-SHA	256	TLS1.2	Native	AES	SHA	ECDHE_RSA
5:	49170	ECDHE-RSA-DES-CBC3-SHA	192	TLS1	Native	DES	SHA	ECDHE_RSA
6:	49170	ECDHE-RSA-DES-CBC3-SHA	192	TLS1.1	Native	DES	SHA	ECDHE_RSA
7:	49170	ECDHE-RSA-DES-CBC3-SHA	192	TLS1.2	Native	DES	SHA	ECDHE_RSA
8:	49199	ECDHE-RSA-AES128-GCM-SHA256	128	TLS1.2	Native	AES-GCM	SHA256	ECDHE_RSA
9:	49191	ECDHE-RSA-AES128-SHA256	128	TLS1.2	Native	AES	SHA256	ECDHE_RSA
10:	49171	ECDHE-RSA-AES128-CBC-SHA	128	TLS1	Native	AES	SHA	ECDHE_RSA
11:	49171	ECDHE-RSA-AES128-CBC-SHA	128	TLS1.1	Native	AES	SHA	ECDHE_RSA
12:	49171	ECDHE-RSA-AES128-CBC-SHA	128	TLS1.2	Native	AES	SHA	ECDHE_RSA
13:	49196	ECDHE-ECDSA-AES256-GCM-SHA384	256	TLS1.2	Native	AES-GCM	SHA384	ECDHE_ECDSA
14:	49188	ECDHE-ECDSA-AES256-SHA384	256	TLS1.2	Native	AES	SHA384	ECDHE_ECDSA
15:	49162	ECDHE-ECDSA-AES256-SHA	256	TLS1	Native	AES	SHA	ECDHE_ECDSA
16:	49162	ECDHE-ECDSA-AES256-SHA	256	TLS1.1	Native	AES	SHA	ECDHE_ECDSA
17:	49162	ECDHE-ECDSA-AES256-SHA	256	TLS1.2	Native	AES	SHA	ECDHE_ECDSA
18:	49160	ECDHE-ECDSA-DES-CBC3-SHA	192	TLS1	Native	DES	SHA	ECDHE_ECDSA
19:	49160	ECDHE-ECDSA-DES-CBC3-SHA	192	TLS1.1	Native	DES	SHA	ECDHE_ECDSA
20:	49160	ECDHE-ECDSA-DES-CBC3-SHA	192	TLS1.2	Native	DES	SHA	ECDHE_ECDSA
21:	49195	ECDHE-ECDSA-AES128-GCM-SHA256	128	TLS1.2	Native	AES-GCM	SHA256	ECDHE_ECDSA
22:	49187	ECDHE-ECDSA-AES128-SHA256	128	TLS1.2	Native	AES	SHA256	ECDHE_ECDSA
23:	49161	ECDHE-ECDSA-AES128-SHA	128	TLS1	Native	AES	SHA	ECDHE_ECDSA
24:	49161	ECDHE-ECDSA-AES128-SHA	128	TLS1.1	Native	AES	SHA	ECDHE_ECDSA
25:	49161	ECDHE-ECDSA-AES128-SHA	128	TLS1.2	Native	AES	SHA	ECDHE_ECDSA

Figure 3: Supported ECDHE ciphers on the BIG-IP system

Specifying the use of Diffie-Hellman ciphers

Use this task to modify an existing Client SSL profile to enable support for Diffie-Hellman key exchange.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client** or **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Client SSL or Server SSL profile list screen opens.
2. In the Name column, click the name of the profile you want to modify.
3. Select the **Custom** check box.
   The settings become available for change.
4. To specify DHE ciphers:
   a) From the Configuration list, select **Advanced**.
   b) In the **Ciphers** field, type `DHE:DHE_DSS`.
5. Click **Update**.

After you perform this task and assign the profile to a virtual server, the BIG-IP uses the DHE key exchange method to establish secure communication with the relevant client or server.

Viewing DHE key exchange statistics

You can use the Traffic Management Shell (tmsh) to view statistics about the use of Diffie-Hellman ciphers in SSL negotiation.

1. Access the system prompt on the BIG-IP system.
2. From the BIG-IP system prompt, type `tmsh show ltm profile client-ssl` *profile_name*.
   An example of a name for a profile that specifies DHE ciphers is `my_dhe_profile`.

After you type this command, the BIG-IP system displays output such as the following. In this example, the profile statistics show that Diffie-Hellman Ephemeral (DHE) with RSA certificates has been used once:

Key Exchange Method	
Anonymous Diffie-Hellman	0
Diffie-Hellman w/ RSA Certs	0
Ephemeral Diffie-Hellman w/ RSA Certs	1
RSA Certs	0

Figure 4: Sample profile statistics for key exchange method

About Elliptic Curve encryption

The BIG-IP system supports Elliptic Curve Cryptography (ECC). Because Elliptic Curve key sizes are significantly smaller than those of other key types, ECC is ideally suited for smaller, mobile devices for which key storage is an issue. On the BIG-IP system, ECC works with the SSL offload feature.

About Elliptic Curve cipher support

The BIG-IP system supports multiple ciphers that use Elliptic Curve Cryptography (ECC) encryption with Diffie-Hellman key exchange. On the BIG-IP system, EC is used with DHE to establish the shared secret; however, the subsequent bulk encryption of data cannot be done with any ECC-based algorithm and must be done using conventional crypto algorithms such as AES and 3DES. For example, a typical Elliptic Curve cipher is: ECDHE-RSA-AES128-CBC-SHA.

The specific ECC ciphers that the BIG-IP system supports are:

- ECDHE-RSA-*
- ECDHE-ECDSA-*
- ECDH-ECDSA-*

Because ECC with Diffie-Hellman does not include a mechanism for digitally signing handshake messages, the RSA or DSA algorithms are used to digitally sign the handshake messages to thwart Man-in-the-Middle attacks. For example, an ECDHE-ECDSA-* cipher suite uses the ECC DSA certificate specified in the Client SSL profile to digitally sign the handshake messages.

Note: *Although the BIG-IP system supports both the prime256v1 and secp384r1 curve names, only prime256v1 can be associated with an SSL profile. Also note that Elliptic Curve ciphers with DSA are not included in the* ***DEFAULT*** *cipher suite.*

Specifying the use of Elliptic Curve ciphers

Use this task to modify an existing Client SSL profile to enable support for Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) key exchange.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client** or **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The Client SSL or Server SSL profile list screen opens.
2. In the Name column, click the name of the profile you want to modify.
3. Select the **Custom** check box.
   The settings become available for change.
4. To specify ECDHE ciphers:
   a) From the Configuration list, select **Advanced**.
   b) In the **Ciphers** field, type `ECDHE`.

5. Click **Update**.

After you perform this task and assign the profile to a virtual server, the BIG-IP system uses the ECDHE key exchange method to establish secure communication with the relevant client or server.

Viewing ECDH key exchange statistics

You can use the Traffic Management Shell (tmsh) to view statistics about the use of Elliptic Curve Diffie-Hellman ciphers in SSL negotiation.

1. Access the system prompt on the BIG-IP system.
2. From the BIG-IP system prompt, type `tmsh show ltm profile client-ssl` *profile_name*`|` grep ECDH.
   An example of a name for a profile that specifies DHE ciphers is `my_ecdh_profile`.

After you type this command, the BIG-IP system displays output such as the following. In this example, the profile statistics show that ECDH with RSA certificates has been used six times:

```
[root@server35:Active:Standalone] config # tmsh show ltm profile client-ssl myclientssl
| grep ECDH
  Ephemeral ECDH w/ RSA Certs                 6
```

Figure 5: Sample profile statistics for key exchange method

Client and server certificate authentication

There are several settings that you can configure on an SSL profile to manage client-side SSL authentication.

Requirement for a client certificate

You can cause Local Traffic Manager™ to handle authentication of clients or servers in certain ways. For client-side processing, the possible behaviors are:

Ignore
Ignore a certificate (or lack of one) and therefore never authenticate the client. The **Ignore** option is the default, and when used, causes any per-session authentication setting to be ignored.

Require
Require a client to present a valid and trusted certificate before granting access.

Request
Request and verify a client certificate. In this case, the SSL profile always grants access regardless of the status or absence of the certificate.

Warning: *If you are using the LDAP-based client authorization feature, use of the* ***Request*** *or* ***Ignore*** *values can sometimes cause a connection to terminate.*

Tip: *The* ***Request*** *value works well with the header insertion feature. Configuring the SSL profile to insert client certificate information into an HTTP client request, and to authenticate clients based on the* ***Request*** *option, enables the BIG-IP® system or a server to then perform actions such as redirecting the request to*

another server, sending different content back to the client, or performing client certificate or session ID persistence.

For server-side processing, the possible behaviors are:

Require
Require a server to present a valid and trusted certificate before granting access.

Ignore
Ignore a certificate (or lack of one) and therefore never authenticate the server. The **Ignore** value is the default setting, and when used, causes any per-session authentication setting to be ignored.

Frequency of authentication

You can configure an SSL profile to require authentication either once per SSL session (once), or once upon each subsequent re-use of an SSL session (always). The default setting for this option is once.

Whether you set this value to once or always depends on your application. A well-designed web application should need to verify a certificate only once per session. F5 recommends for performance reasons that you use the default setting (once) whenever possible.

You can modify the SSL profile to require authentication not only once per session, but also upon each subsequent re-use of an SSL session.

Certificate chain traversal depth

You can use the **Certificate Chain Traversal Depth** setting of an SSL profile to configure the maximum number of certificates that can be traversed in the certificate chain. The default value is `9`. If a longer chain is provided, and the client has not been authenticated within this number of traversals, client or server certificate verification fails. If the authentication depth value is set to zero, then only the client or server certificate, and one of the chain files, are examined.

Trusted certificate authorities

For client-side and server-side SSL processing, you can use the **Trusted Certificate Authorities** setting on an SSL profile to configure an SSL profile to verify certificates presented by a client or a server. You can specify a client trusted CAs file name, which the BIG-IP® system then uses to verify client or server certificates. If you do not configure a trusted CAs file, the system uses a default file.

The trusted CAs file that you specify for certificate verification contains one or more certificates, in Privacy Enhanced Mail (PEM) format. Built manually, this file contains a list of the client or server certificates that the SSL profile will trust. If you do not specify a trusted CAs file, or the specified trusted CAs file is not accessible to the BIG-IP system, the system uses the default file name.

Advertised certificate authorities

For client-side profiles only, if you intend to configure the SSL profile to require or request client certificates for authentication, you will want the profile to send to clients a list of CAs that the BIG-IP® system is likely to trust.

This list, known as the *Client Certificate CA file*, is different from the client Trusted CAs file. This is because, in some cases, you might have a client that does not possess a valid client certificate, in which case you might not want to reveal the actual list of CAs that the BIG-IP system trusts. The client certificate CA file solves this problem by allowing the BIG-IP system to advertise a list of CAs that is different from the actual client trusted CAs file configured as part of certificate verification.

Although the contents of the Client Certificate CA file can differ from that of the Client Trusted CAs file, it is best, for compatibility reasons, to set the **Advertised Certificate Authorities** setting to match the **Trusted Certificate Authorities** setting. This is because modern browsers might not permit SSL session negotiation to proceed if the peer that requests the client certificate does not provide a list of trusted CAs.

Note: *The maximum size of native SSL handshake messages that Local Traffic Manager™ allows is 14304 bytes. Consequently, if the SSL handshake is negotiating a native cipher and the total length of all messages in the handshake exceeds this byte threshold, the handshake can fail. Although typical use does not cause message length to exceed this threshold, we recommend that when configuring a Client SSL profile to request or require client certificates, you avoid specifying large numbers of certificates through the* ***Advertised Certificate Authorities*** *setting. This minimizes the number of certificates that must be exchanged during a Client SSL handshake.*

Name-based authentication

For server-side profiles only, Local Traffic Manager™ supports name-based authentication, which guards against man-in-the-middle attacks. When you configure the Authenticate Name setting for a server-side profile, Local Traffic Manager checks the name against the Common Name (CN) listed in the certificate that the target server presents to the BIG-IP® system. If the name attribute that you specify does not match the CN in the server certificate, Local Traffic Manager closes the connection. An example of a CN is `www.f5.com`.

Certificate revocation

The **Certificate Revocation List (CRL)** setting of an SSL profile allows Local Traffic Manager™ to use CRLs to check revocation status of a certificate prior to authenticating a client or server.

Important: *CRL files can become outdated, and might need to be updated as often as every day, or as seldom as every 30 days. If your CRL file is out-of-date, Local Traffic Manager rejects all certificates, both valid and invalid. For this reason, it is important to keep your CRL files up-to-date at all times. You can do this by accessing the CRL in the* `/config/ssl/ssl.crl` *directory and then using the* `openssl crl` *command. For more information, see* `http://www.openssl.org/docs/`.

As an alternative to using CRLs, you can use the Online Certificate Status Protocol (OCSP) feature, which ensures up-to-date information on certificate revocation status.

Chapter 4

Additional SSL Configuration Options

- *Options*
- *ModSSL methods*
- *SSL session cache size and timeout*
- *Alert timeout*
- *Handshake timeout*
- *Renegotiation of SSL sessions*
- *Server name*
- *Default SSL Profile for SNI*
- *Require Peer SNI Support*
- *Unclean SSL shutdowns*
- *Strict Resume*
- *About session tickets*
- *Generic alerts*
- *Acceptance of non-SSL connections*
- *SSL sign hash*

Options

OpenSSL supports a set of SSL options and defect workarounds. You can enable these workarounds and options as settings of an individual client-side or server-side SSL profile. The default value for the **Options** setting is **Options List**. Retaining the default value enables one option, which is **Don't insert empty fragments**. You can then enable other options that appear in the **Available Options** list.

Important: *For security reasons, when you enable the Proxy SSL setting, the BIG-IP® system automatically disables the* ***Don't insert empty fragments option****. Disabling this option when Proxy SSL is enabled guards against a particular type of cryptographic attack.*

Note that when configuring protocol versions, you must ensure that the protocol versions configured for the BIG-IP system match those of the system's peer. That is, protocol versions specified in the client-side SSL profile must match those of the client, and protocol versions specified in the server-side SSL profile must match those of the server. Thus, for both client-side and server-side SSL connections, you can specify the protocol versions that you do not want the BIG-IP system to allow.

You can declare up to two of the three protocol versions to be invalid: SSLv2 , SSLv3, and TLSv1. If no protocol versions are specified, Local Traffic Manager™ allows all SSL protocol versions.

Note: *F5 Networks recommends that, at a minimum, you specify protocol version SSLv2 as invalid.*

Workarounds and other SSL options

This table lists and describes the possible workarounds and options that you can configure for an SSL profile.

SSL Attribute	Description
Cipher server preference	When the BIG-IP® system chooses a cipher, this option uses the server's preferences instead of the client preferences. When this option is not set, the SSL server always follows the client's preferences. When this option is set, the SSLv3/TLSv1 server chooses by using its own preferences. Due to the different protocol, for SSLv2 the server sends its list of preferences to the client, and the client always chooses the cipher.
Don't insert empty fragments	This option disables a countermeasure against a SSL 3.0/TLS 1.0 protocol vulnerability affecting CBC ciphers. These ciphers cannot be handled by certain broken SSL implementations. This option has no effect for connections using other ciphers. This is the default value for the Options list. ***Note:*** *For security reasons, this option is not available when you enable the Proxy SSL setting.*
Ephemeral RSA	This option uses ephemeral (temporary) RSA keys when doing RSA operations. According to the specifications, this is only done when an RSA key can only be used for signature operations (namely under export ciphers with restricted RSA key length). By setting this option, Local Traffic Manager™ always uses ephemeral RSA keys. This option breaks compatibility with the SSL/TLS specifications and can lead to interoperability problems with clients, and we therefore do not recommend it. You should use ciphers with EDH (ephemeral Diffie-Hellman) key exchange instead. This option is ignored for server-side SSL.
Microsoft session ID bug	This option handles a Microsoft® session ID problem.

SSL Attribute	Description
Netscape CA DN bug workaround	This option handles a defect regarding system instability. If the system accepts a Netscape® browser connection, demands a client cert, has a non-self-signed CA that does not have its CA in Netscape, and the browser has a certificate, then the system crashes or hangs.
Netscape challenge bug	This option handles the Netscape challenge problem.
Netscape demo cipher change bug workaround	This option deliberately manipulates the SSL server session resumption behavior to mimic that of certain Netscape servers (see the Netscape reuse cipher change bug workaround description). We do not recommend this option for normal use and it is ignored for server-side SSL processing.
Netscape reuse cipher change bug workaround	This option handles a defect within Netscape-Enterprise/2.01, only appearing when connecting through SSLv2/v3 then reconnecting through SSLv3. In this case, the cipher list changes. First, a connection is established with the RC4-MD5 cipher list. If it is then resumed, the connection switches to using the DES-CBC3-SHA cipher list. However, according to RFC 2246, (section 7.4.1.3, cipher_suite) the cipher list should remain RC4-MD5. As a workaround, you can attempt to connect with a cipher list of DES-CBC-SHA:RC4-MD5 and so on. For some reason, each new connection uses the RC4-MD5 cipher list, but any re-connect ion attempts to use the DES-CBC-SHA cipher list. Thus Netscape, when reconnecting, always uses the first cipher in the cipher list.
No SSLv2	Do not use the SSLv2 protocol.
No SSLv3	Do not use the SSLv3 protocol.
No session resumption on renegotiation	When Local Traffic Manager performs renegotiation as an SSL server, this option always starts a new session (that is, session resumption requests are only accepted in the initial handshake). The system ignores this option for server-side SSL processing.
N0 TLSv1	Do not use the TLSv1 protocol.
Microsoft big SSLV3 buffer	This option enables a workaround for communicating with older Microsoft® applications that use non-standard SSL record sizes.
Microsoft IE SSLV2 RSA padding	This option enables a workaround for communicating with older Microsoft® applications that use non-standard RSA key padding. This option is ignored for server-side SSL.
Passive close	Specifies that the SSL filter helps prevent packets from getting into the TCP half-closed state by waiting for a connection shutdown from the server. This is a workaround for HTTP/1.0 and HTTP/0.9 clients that send an HTTP request followed by a FIN, which immediately closes the connection for server-SSL-only proxies. Instead of closing immediately, the proxy waits for the server to close.
PKCS1 check 1	This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We do not recommend this option for normal use. The system ignores this option for client-side SSL processing.
PKCS1 check 2	This debugging option deliberately manipulates the PKCS1 padding used by SSL clients in an attempt to detect vulnerability to particular SSL server vulnerabilities. We do not recommend this option for normal use. The system ignores this option for client-side SSL processing.
Single DH use	This option creates a new key when using temporary/ephemeral DH parameters. You must use this option if you want to prevent small subgroup attacks, when the DH parameters were not generated using strong primes (for example, when using DSA-parameters). If strong primes were used, it is not strictly necessary to generate a

SSL Attribute	Description
	new DH key during each handshake, but we do recommend this. You should enable the Single DH use option whenever temporary/ephemeral DH parameters are used.
SSLEAY 080 client DH bug workaround	This option enables a workaround for communicating with older SSLeay-based applications that specify an incorrect Diffie-Hellman public value length. This option is ignored for server-side SSL.
SSL Ref2 reuse cert type bug	This option handles the SSL re-use certificate type problem.
TLS D5 bug workaround	This option is a workaround for communicating with older TLSv1-enabled applications that specify an incorrect encrypted RSA key length. This option is ignored for server-side SSL.
TLS block padding bug workaround	This option enables a workaround for communicating with older TLSv1-enabled applications that use incorrect block padding.
TLS rollback bug workaround	This option disables version rollback attack detection. During the client key exchange, the client must send the same information about acceptable SSL/TLS protocol levels as it sends during the first hello. Some clients violate this rule by adapting to the server's answer. For example, the client sends an SSLv2 hello and accepts up to SSLv3.1 (TLSv1), but the server only understands up to SSLv3. In this case, the client must still use the same SSLv3.1 (TLSv1) announcement. Some clients step down to SSLv3 with respect to the server's answer and violate the version rollback protection. This option is ignored for server-side SSL.

ModSSL methods

You can enable or disable ModSSL method emulation. You enable ModSSL method emulation when the OpenSSL methods are inadequate. When you enable this setting, you can then write an iRule, using the `HTTP::header insert_modssl_fields` command, which inserts some of the ModSSL options as headers into HTTP requests.

ModSSL options for use with iRules

This table lists the options that you can insert into an HTTP request.

Header Type	Header Name and Format	Description
Certificate status	SSLClientCertStatus: [status]	The status of the client certificate. The value of [status] can be NoClientCert, OK, or Error. If status is NoClientCert, only this header is inserted into the request. If status is Error, the error is followed by a numeric error code.
Certificate version	SSLClientCertVersion: [version]	The version of the certificate.
Certificate serial number	SSLClientCertSerialNumber: [serial]	The serial number of the certificate.
Signature algorithm of the certificate	SSLClientCertSignatureAlgorithm: [alg]	The signature algorithm of the certificate.

Header Type	Header Name and Format	Description
Issuer of the certificate	SSLClientCertIssuer: [issuer]	The issuer of the certificate.
Certificate validity dates	SSLClientCertNotValidBefore: [before] SSLClientCertNotValidAfter: [after]	The validity dates for the certificate. The certificate is not valid before or after the dates represented by `[before]` and `[after]`, respectively.
Certificate subject	SSLClientCertSubject: [subject]	The subject of the certificate.
Public key of the subject	SSLClientCertSubjectPublicKey: [key]	The type of public key type. The allowed types are RSA ([size] bit), DSA, or Unknown public key.
The certificate itself	SSLClientCert: [cert]	The actual client certificate.
MD5 hash of the certificate	SSLClientCertHash: [hash]	The MD5 hash of the client certificate.

SSL session cache size and timeout

You can configure timeout and size values for the SSL session cache. Because each profile maintains a separate SSL session cache, you can configure the values on a per-profile basis.

SSL session cache size

You can specify the maximum size of the SSL session cache. The default value for the size of the SSL session cache is 262144 entries. A value of `0` disallows session caching.

SSL session cache timeout

You can specify the number of usable lifetime seconds of negotiated SSL session IDs. The default timeout value for the SSL session cache is `3600` seconds. If you specify a timeout value, valid values are integers greater than or equal to 1.

Clients attempting to resume an SSL session with an expired session ID are forced to negotiate a new session.

Warning: *If the timeout value for the client-side SSL session cache is set to zero, the SSL session IDs negotiated with that profile's clients remain in the session cache until the cache is filled and the purging of entries begins. Setting a value of zero can introduce a significant security risk if valuable resources are available to a client that is reusing those session IDs. It is therefore common practice to set the SSL session cache timeout to a length of time no greater than 24 hours, and for significantly shorter periods.*

Alert timeout

You can specify the duration in seconds that the BIG-IP® system waits while trying to close an SSL connection, before the connection is reset. The default timeout value for this setting is 10 seconds.

Handshake timeout

You can specify the amount of time in seconds that the BIG-IP® system spends attempting to perform an SSL handshake. The default timeout value for this setting is 10 seconds.

Renegotiation of SSL sessions

Long-lived connections are susceptible to man-in-the-middle attacks. To prevent such attacks, you can force Local Traffic Manager™ to renegotiate SSL sessions, based on either time period or application size. You can also force Local Traffic Manager to terminate an SSL session after receiving a specified number of records.

Sessions based on a time period

You can specify the number of seconds from the initial connect time that the system renegotiates an SSL session. The options are a number you specify, indefinite, and default. The default is indefinite, meaning that you do not want the system to renegotiate SSL sessions. Each time the session renegotiation is successful, essentially a new connection is started. Therefore, the system attempts to renegotiate the session again in the specified amount of time following the successful session renegotiation. For example, setting the **renegotiate period** to **3600** seconds triggers session renegotiation at least once an hour.

Sessions based on application data size

You can force Local Traffic Manager™ to renegotiate an SSL session after the specified number of megabytes of application data have been transmitted over the secure channel. The default value for this setting is `Indefinite`.

Maximum record delay

You can force Local Traffic Manager™ to terminate an SSL session after receiving the specified maximum number of SSL records. If Local Traffic Manager receives more than the maximum number of SSL records, it closes the connection. The default value for this setting, in seconds, is `10`.

Secure renegotiation

The Secure Renegotiation setting specifies the method of secure renegotiation for SSL connections. The default value for the Client SSL profile is Require; the default value for the Server SSL profile is Require Strict. If your configuration does not require secure SSL renegotiation, set this value to Request. The possible values for this setting are:

Request
Specifies that the system requests secure renegotiation of SSL connections.

Require
Specifies that the system requires secure renegotiation of SSL connections. In this mode, the system permits initial SSL handshakes from clients, but terminates renegotiations from unpatched clients.

Require Strict
Specifies that the system requires strict secure renegotiation of SSL connections. In this mode, the system refuses new SSL connections to unsecure servers and terminates existing SSL connections to unsecure servers.

Server name

The **Server Name** setting in an SSL profile specifies the name of the specific domain from which the client requests a certificate. This setting supports a feature known as TLS Server Name Indication (TLS SNI), used when a single virtual IP server needs to host multiple domains.

For example, suppose that the BIG-IP™ system needs to host the two domains `domain1.com` and `domain2.com`, on the same HTTP virtual server. Each domain has its own server certificate to use, such as domain1.crt and domain2.crt, and each has different security requirements.

To ensure that the BIG-IP system presents the correct certificate to the browser, you enable SNI, which sends the name of a domain as part of the TLS negotiation. This, in turn, enables the BIG-IP system to select this domain rather than waiting to read the domain name in the request header.

To enable SNI, you configure the **Server Name** and other TLS-related settings on an SSL profile, and then assign the profile to a virtual server. Note that the wildcard character (*) is supported within any domain name that you specify.

Default SSL Profile for SNI

When you enable the **Default SSL Profile for SNI** setting on an SSL profile, you are specifying that this is the default SSL profile to use when the client provides either no Server Name Indication (SNI) extension, or provides a non-matching SNI extension.

When assigning multiple SSL profiles to a single virtual server, you can enable this setting on one Client SSL profile only and one Server SSL profile only.

Require Peer SNI Support

If you enable the **Require Peer SNI Support** setting on an SSL profile, the domain name of the peer must match the domain name that you specify in the **Default SSL Profile for SNI** field.

Unclean SSL shutdowns

In an *unclean shutdown*, underlying TCP connections are closed without exchanging the required SSL shutdown alerts. However, you can disable unclean shutdowns and thus force the SSL profile to perform a clean shutdown of all SSL connections by configuring this setting.

This feature is especially useful with respect to the Internet Explorer browser. Different versions of the browser, and even different builds within the same version of the browser, handle shutdown alerts differently. Some versions or builds require shutdown alerts from the server, while others do not, and the SSL profile cannot always detect this requirement or lack of it. In the case where the browser expects a shutdown alert but the SSL profile has not exchanged one (the default setting), the browser displays an error message.

By default, this setting is enabled, which means that Local Traffic Manager™ performs unclean shutdowns of all SSL connections.

Strict Resume

You can configure Local Traffic Manager™ to discontinue an SSL session after an unclean shutdown. By default, this setting is disabled, which causes Local Traffic Manager (LTM®) to resume SSL sessions after an unclean shutdown. If you enable this setting, LTM does not resume SSL sessions after an unclean shutdown.

About session tickets

To enhance system performance, you can enable the use of session tickets, a TLS extension defined in RFC 5077. The use of session tickets is an alternative to the standard session caching mechanism that systems such as the BIG-IP system typically use to resume sessions.

When you enable this feature, the BIG-IP system, acting as a server to terminate SSL connections, sends a special message to the client as part of the SSL handshake. This message includes a *session ticket*, which contains complete session state information. Sending the session state information to the client removes the need for the BIG-IP system to maintain a server-side cache for storing session information. With session tickets, the entire session state is remembered by the client.

The session state information in the ticket includes the master secret negotiated between the client and the BIG-IP system, as well as the cipher suite used.

Generic alerts

By default, when an SSL failure occurs, the BIG-IP system sends an alert message with a numeric code indicating the type of failure. If you do not want alert messages to indicate the specific reason for the failure, for security reasons, you can enable the **Generic Alerts** setting on an SSL profile. Enabling this setting causes the BIG-IP system to label all failure alerts as handshake failures, with a code of `40`. For example, a failure due to a certificate revocation is normally flagged with a code of `48`, but with the **Generic Alerts** setting enabled, the alert is coded as `40`.

Acceptance of non-SSL connections

You can configure Local Traffic Manager™ to accept connections that are not SSL connections. In this case, connections pass through the BIG-IP® system in clear-text format. By default, this setting is disabled.

SSL sign hash

You can specify the specific hash algorithm that you want the BIG-IP® system to use for server key exchange with Elliptic Curve ciphers. Possible choices are **SHA1**, **SHA256**, **SHA384**, or **Any**. When you select **Any**, you authorize the system to choose any one of the hash algorithms. Note that in this case, the BIG-IP system chooses **SHA1** whenever possible.

Chapter 5

SSL Persistence

- *SSL persistence*
- *Criteria for session persistence*
- *Creating an SSL persistence profile*

SSL persistence

SSL persistence is a type of persistence that tracks SSL sessions using the SSL session ID, and it is a property of each individual pool. Using SSL persistence can be particularly important if your clients typically have translated IP addresses or dynamic IP addresses, such as those that Internet service providers typically assign. Even when the client's IP address changes, BIG-IP system® still recognizes the session as being persistent based on the session ID.

You might want to use SSL persistence and source address affinity persistence together. In situations where an SSL session ID times out, or where a returning client does not provide a session ID, you might want the BIG-IP system to direct the client to the original node based on the client's IP address. As long as the client's simple persistence record has not timed out, the BIG-IP system can successfully return the client to the appropriate node.

Criteria for session persistence

Regardless of the type of persistence you are implementing, you can specify the criteria that the BIG-IP® system uses to send all requests from a given client to the same pool member. These criteria are based on the virtual server or servers that are hosting the client connection. To specify these criteria, you use the **Match Across Services**, **Match Across Virtual Servers**, and **Match Across Pools** profile settings. Before configuring a persistence profile, it is helpful to understand these settings.

Creating an SSL persistence profile

You create an SSL persistence profile when you want to customize the way that the BIG-IP®system persists SSL traffic.

Important: *The BIG-IP system includes a default SSL persistence profile named* `ssl`*. If you do not need to customize the way that the system persists SSL traffic, you can skip this task. Instead, simply use the* ***Default Persistence Profile*** *setting on the relevant virtual server to specify the default* `ssl` *profile.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **Persistence**.
   The Persistence profile list screen opens.
2. Click **Create**.
   The New Persistence Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Persistence Type** list, select **SSL**.
5. For the **Parent Profile** setting, confirm that **ssl** appears.
6. Select the **Custom** check box.
7. Configure settings as needed.
8. Click **Finished**.

The custom SSL persistence profile now appears in the persistence profiles list.

After creating a persistence profile, you must assign the profile to the relevant virtual server.

Chapter 6

Managing Client-Side HTTP Traffic Using a CA-Signed RSA Certificate

Overview: Managing client-side HTTP traffic using a CA-signed RSA certificate

When you want to manage HTTP traffic over SSL, you can configure the BIG-IP® system to perform the SSL handshake that target web servers normally perform.

A common way to configure the BIG-IP system is to enable client-side SSL, which makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.

This implementation uses a certificate signed by an RSA certificate authority (CA) to authenticate HTTP traffic.

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

Requesting an RSA certificate from a certificate authority
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a pool to process HTTP traffic
Creating a virtual server for client-side HTTP traffic

Requesting an RSA certificate from a certificate authority

You can generate a request for an RSA digital certificate and then copy or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.

14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select **RSA**.
17. From the **Size** list, select a key size, in bits.
18. Click **Finished**.
    The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated RSA certificate request is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you specify an RSA type of key chain.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.

The settings become available for change.

6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select a certificate name.
      This is the name of an RSA certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, `default`.
   b) From the **Key** list, select a key name.
      This is the name of an RSA key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, you can specify the name of an existing key, `default`.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to the SSL certificate/key pair.
      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.
   e) Click **Add**.
      The result is that the specified key chain appears in the box.

7. If you want to use a cipher suite other than `DEFAULT`:
   a) From the Configuration list, select **Advanced**.
   b) For the **Ciphers** setting, type the name of a cipher.
      You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.
      Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.

8. Configure all other profile settings as needed.
9. Click **Finished**.

After performing this task, you must assign the profile to a virtual server.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click **<<** to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system can authenticate and decrypt HTTP traffic coming from a client system, using an RSA digital certificate. The BIG-IP system can also re-encrypt server responses before sending them back to the client.

Chapter 7

Managing Client-Side HTTP Traffic Using a Self-Signed RSA Certificate

Overview: Managing client-side HTTP traffic using a self-signed RSA certificate

This implementation uses an RSA self-signed certificate to authenticate HTTP traffic. When you want to manage HTTP traffic over SSL, you can configure the BIG-IP® system to perform the SSL handshake that target web servers typically perform.

A common way to configure the BIG-IP system is to enable client-side SSL, which makes it possible for the system to decrypt client requests before forwarding them to a server, and to encrypt server responses before returning them to the client. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.

Task summary

To implement client-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

Creating a self-signed RSA certificate
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a pool to process HTTP traffic
Creating a virtual server for client-side HTTP traffic

Creating a self-signed RSA certificate

If you are configuring the BIG-IP® system to manage client-side HTTP traffic, you create an RSA self-signed digital certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you create a second RSA self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.

13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select **RSA**.
15. From the **Size** list, select a key size, in bits.
16. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you specify an RSA type of key chain.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select a certificate name.
      This is the name of an RSA certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, `default`.
   b) From the **Key** list, select a key name.
      This is the name of an RSA key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, you can specify the name of an existing key, `default`.

c) From the **Chain** list, select the chain that you want to include in the certificate key chain.

A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

---

d) For the **Passphrase** field, type a string that enables access to the SSL certificate/key pair.

This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.

e) Click **Add**.
The result is that the specified key chain appears in the box.

7. If you want to use a cipher suite other than `DEFAULT`:
   a) From the Configuration list, select **Advanced**.
   b) For the **Ciphers** setting, type the name of a cipher.

   You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.

   Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.

8. Configure all other profile settings as needed.
9. Click **Finished**.

After performing this task, you must assign the profile to a virtual server.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click **<<** to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.

7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.

c) (Optional) Type a priority number in the **Priority** field.
d) Click **Add**.

8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system can authenticate and decrypt HTTP traffic coming from a client system, using an RSA self-signed certificate. The BIG-IP system can also re-encrypt server responses before sending them back to the client.

Chapter 8

Managing Client-side HTTP Traffic Using a CA-Signed Elliptic Curve DSA Certificate

Overview: Managing client-side HTTP traffic using a CA-signed, ECC-based certificate

When you configure the BIG-IP® system to decrypt client-side HTTP requests and encrypt the server responses, you can optionally configure the BIG-IP system to use the Elliptic Curve Digital Signature Algorithm (ECDSA) as part of the BIG-IP system's certificate key chain. The result is that the BIG-IP system performs the SSL handshake usually performed by target web servers, using an ECDSA key type in the certificate key chain.

This particular implementation uses a certificate signed by a certificate authority (CA).

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

Requesting an RSA certificate from a certificate authority
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a pool to process HTTP traffic
Creating a virtual server for client-side HTTP traffic

Requesting a signed certificate that includes an ECDSA key

You can generate a certificate that includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key type, and then copy it or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.

14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select **ECDSA**.
17. From the **Curve Name** list, select **prime256v1**.
18. Click **Finished**.
    The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server. When you perform this task, you specify a certificate key chain that includes Elliptic Curve Digital Signature Algorithm (ECDSA) as the key type.

Note: *In addition to specifying an ECDSA certificate key chain, you must also specify an RSA key chain. Specifying an RSA key chain is a minimum requirement for all Client SSL profiles that you configure.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.

The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify both an ECDSA and an RSA certificate key chain:
   a) From the **Certificate** list, select the name of a certificate with a key of type ECDSA.
   b) From the **Key** list, select the name of an ECDSA key.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
   d) Click **Add**.
   e) Repeat this process and specify an RSA certificate key chain.
7. To specify ECDHE ciphers:
   a) From the Configuration list, select **Advanced**.
   b) In the **Ciphers** field, type `ECDHE`.
8. Configure all other profile settings as needed.
9. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click **<<** to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system encrypts client-side ingress HTTP traffic using an SSL certificate key chain. The BIG-IP system also re-encrypts server responses before sending the responses back to the client.

The certificate in the certificate key chain includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key and certificate.

Chapter 9

Managing Client-side HTTP Traffic Using a Self-Signed Elliptic Curve DSA Certificate

Overview: Managing client-side HTTP traffic using a self-signed, ECC-based certificate

When you configure the BIG-IP® system to decrypt client-side HTTP requests and encrypt the server responses, you can optionally configure the BIG-IP system to use the Elliptic Curve Digital Signature Algorithm (ECDSA) as part of the BIG-IP system's certificate key chain. The result is that the BIG-IP system performs the SSL handshake, usually performed by target web servers, using an ECDSA key type in the certificate key chain.

This particular implementation uses a self-signed certificate.

Task summary

To implement client-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

Creating a self-signed RSA certificate
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a pool to process HTTP traffic
Creating a virtual server for client-side HTTP traffic

Creating a self-signed SSL certificate

If you are configuring the BIG-IP system to manage client-side HTTP traffic, you create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.

14. From the **Key Type** list, select **ECDSA**.
15. From the **Curve Name** list, select **prime256v1**.
16. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server. When you perform this task, you specify a certificate key chain that includes Elliptic Curve Digital Signature Algorithm (ECDSA) as the key type.

Note: *In addition to specifying an ECDSA certificate key chain, you must also specify an RSA key chain. Specifying an RSA key chain is a minimum requirement for all Client SSL profiles that you configure.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify both an ECDSA and an RSA certificate key chain:
   a) From the **Certificate** list, select the name of a certificate with a key of type ECDSA.
   b) From the **Key** list, select the name of an ECDSA key.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
   d) Click **Add**.
   e) Repeat this process and specify an RSA certificate key chain.

7. To specify ECDHE ciphers:
   a) From the Configuration list, select **Advanced**.
   b) In the **Ciphers** field, type `ECDHE`.
8. Configure all other profile settings as needed.
9. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click **<<** to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTP traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

The IP address you type must be available and not in the loopback network.

5. In the **Service Port** field, type `443`, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

After performing this task, the virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system encrypts client-side ingress HTTP traffic using an SSL certificate key chain. The BIG-IP system also re-encrypts server responses before sending the responses back to the client.

The certificate in the certificate key chain includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key and certificate.

Chapter 10

Managing Client- and Server-side HTTP Traffic using a CA-signed Certificate

Overview: Managing client and server HTTP traffic using a CA-signed certificate

One of the ways to configure the BIG-IP system to manage SSL traffic is to enable both client-side and server-side SSL termination:

- *Client-side SSL termination* makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. This ensures that client-side HTTP traffic is encrypted. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.
- *Server-side SSL termination* makes it possible for the system to decrypt and then re-encrypt client requests before sending them on to a server. Server-side SSL termination also decrypts server responses and then re-encrypts them before sending them back to the client. This ensures security for both client- and server-side HTTP traffic. In this case, you need to install two SSL key/certificate pairs on the BIG-IP system. The system uses the first certificate/key pair to authenticate the client, and uses the second pair to request authentication from the server.

This implementation uses a CA-signed certificate to manage HTTP traffic.

Task summary

To implement client-side and server-side authentication using HTTP and SSL with a CA-signed certificate, you perform a few basic configuration tasks.

Task list

Requesting a certificate from a certificate authority
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a custom Server SSL profile
Creating a pool to manage HTTPS traffic
Creating a virtual server for client-side and server-side HTTPS traffic

Requesting a certificate from a certificate authority

You perform this task to generate a certificate signing request (CSR) that can then be submitted to a third-party trusted certificate authority (CA).

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.

8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select a key type.
    Possible values are: **RSA**, **DSA**, and **ECDSA**.
17. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
18. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
19. Click **Finished**.
    The Certificate Signing Request screen displays.
20. Do one of the following to download the request into a file on your system.
    - In the **Request Text** field, copy the certificate.
    - For **Request File**, click the button.
21. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
22. Click **Finished**.
    The Certificate Signing Request screen displays.

The generated certificate signing request is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select a certificate name.
      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, `default`.
   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.
      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, you can specify the name of an existing key, `default`.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.
      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.
   e) Click **Add** and repeat the process for all certificate key chains that you want to specify.

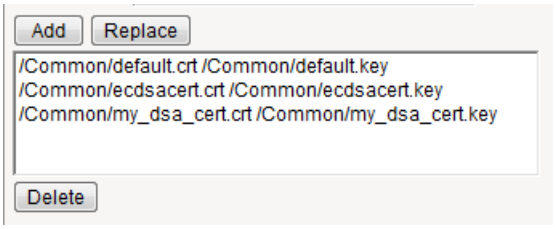


Figure 6: Sample configuration with three key types specified

The result is that all specified key chains appear in the box.

7. If you want to use a cipher suite other than `DEFAULT`:
   a) From the Configuration list, select **Advanced**.
   b) For the **Ciphers** setting, type the name of a cipher.
      You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.
      Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.
8. Configure all other profile settings as needed.
9. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With an Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The SSL Server profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.
9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.
10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. If you want to use a cipher suite other than `DEFAULT`:
    a) From the Configuration list, select **Advanced**.
    b) For the **Ciphers** setting, type the name of a cipher.

You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.

Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.

12. Select the **Custom** check box for **Server Authentication**.
13. Modify the settings, as required.
14. Click **Finished**.

After performing this task, you can see the custom Server SSL profile in the list of Server SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a pool to manage HTTPS traffic

You can create a pool (a logical set of devices, such as web servers, that you group together to receive and process HTTPS traffic) to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Assign the **https** or **https_443** health monitor from the **Available** list by moving it to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Add each resource that you want to include in the pool using the **New Members** setting:
   a) Type an IP address in the **Address** field.
   b) Type `443` in the **Service Port** field, or select **HTTPS** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The HTTPS load balancing pool now appears in the Pool List screen.

Creating a virtual server for client-side and server-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.

2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. Type `443` in the **Service Port** field, or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system ensures that SSL authentication and encryption occurs for both client-side and server-side HTTP traffic. The system performs this authentication and encryption according to the values you specify in the Client SSL and Server SSL profiles.

Chapter 11

Managing Client- and Server-side HTTP Traffic using a Self-signed Certificate

Overview: Managing client and server HTTP traffic using a self-signed certificate

One of the ways to configure the BIG-IP system to manage SSL traffic is to enable both client-side and server-side SSL processing:

- *Client-side SSL termination* makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. This ensures that client-side HTTP traffic is encrypted. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.
- *Server-side SSL termination* makes it possible for the system to decrypt and then re-encrypt client requests before sending them on to a server. Server-side SSL termination also decrypts server responses and then re-encrypts them before sending them back to the client. This ensures security for both client- and server-side HTTP traffic. In this case, you need to install two SSL key/certificate pairs on the BIG-IP system. The system uses the first certificate/key pair to authenticate the client, and uses the second pair to request authentication from the server.

This implementation uses a self-signed certificate to authenticate HTTP traffic.

Task summary

To implement client-side and server-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

Creating a self-signed digital certificate
Creating a custom HTTP profile
Creating a custom Client SSL profile
Creating a custom Server SSL profile
Creating a pool to manage HTTPS traffic
Creating a virtual server for client-side and server-side HTTPS traffic

Creating a self-signed digital certificate

If you are configuring the BIG-IP® system to manage client-side HTTP traffic, you perform this task to create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you must repeat this task to create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System** > **File Management** > **SSL Certificate List**.
   The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.

7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the or **State or Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.
    This name is embedded in the certificate for X509 extension purposes.
    By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select a key type.
    Possible values are: **RSA**, **DSA**, and **ECDSA**.
15. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
16. If the BIG-IP system contains an internal HSM module, specify a location for storing the private key.
17. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP®system to manage HTTP traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
   The HTTP profile list screen opens.
2. Click **Create**.
   The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and encrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these decryption/encryption functions from the destination server. When you perform this task, you can specify multiple certificate key chains, one for each key type (RSA, DSA, and ECDSA). This allows the BIG-IP system to negotiate secure client connections using different cipher suites based on the client's preference.

Note: *At a minimum, you must specify a certificate key chain that includes an RSA key pair. Specifying certificate key chains for DSA and ECDSA key pairs is optional, although highly recommended.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.

The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
   The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
   a) From the **Certificate** list, select a certificate name.
      This is the name of a certificate that you installed on the BIG-IP® system. If you have not generated a certificate request nor installed a certificate on the BIG-IP system, you can specify the name of an existing certificate, `default`.
   b) From the **Key** list, select the name of the key associated with the certificate specified in the previous step.
      This is the name of a key that you installed on the BIG-IP® system. If you have not installed a key on the BIG-IP system, you can specify the name of an existing key, `default`.
   c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
      A certificate chain can contain either a series of public key certificates in Privacy Enhanced Mail (PEM) format or a series of one or more PEM files. A certificate chain can contain certificates for Intermediate certificate Authorities (CAs).

      Note: *The default self-signed certificate and the default CA bundle certificate are not appropriate for use as a certificate chain.*

   d) For the **Passphrase** field, type a string that enables access to SSL certificate/key pairs that are stored on the BIG-IP system with password protection.
      This setting is optional. For added security, the BIG-IP system automatically encrypts the pass phrase itself. This pass phrase encryption process is invisible to BIG-IP® system administrative users.
   e) Click **Add** and repeat the process for all certificate key chains that you want to specify.

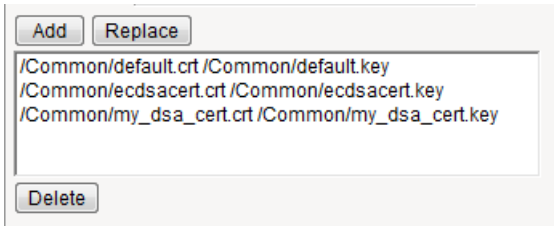


Figure 7: Sample configuration with three key types specified

   The result is that all specified key chains appear in the box.

7. If you want to use a cipher suite other than `DEFAULT`:
   a) From the Configuration list, select **Advanced**.
   b) For the **Ciphers** setting, type the name of a cipher.
      You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.
      Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.
8. Configure all other profile settings as needed.
9. Click **Finished**.

After performing this task, you can see the custom Client SSL profile in the list of Client SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a custom Server SSL profile

With an Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The SSL Server profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
   The settings become available for change.
7. From the **Certificate** list, select the name of an SSL certificate on the BIG-IP system.
8. From the **Key** list, select the name of an SSL key on the BIG-IP system.
9. In the **Pass Phrase** field, select a pass phrase that enables access to the certificate/key pair on the BIG-IP system.
10. From the **Chain** list, select the name of an SSL chain on the BIG-IP system.
11. If you want to use a cipher suite other than `DEFAULT`:
    a) From the Configuration list, select **Advanced**.
    b) For the **Ciphers** setting, type the name of a cipher.

       You can specify a particular string to indicate the ciphers that you want the BIG-IP system to use for SSL negotiation, or you can specify ciphers that you do not want the system to use.

       Examples of cipher values that you can specify are `ECDHE` and `DEFAULT:!ECDHE`.
12. Select the **Custom** check box for **Server Authentication**.
13. Modify the settings, as required.
14. Click **Finished**.

After performing this task, you can see the custom Server SSL profile in the list of Server SSL profiles on the system.

You must also assign the profile to a virtual server.

Creating a pool to manage HTTPS traffic

You can create a pool (a logical set of devices, such as web servers, that you group together to receive and process HTTPS traffic) to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.

3. In the **Name** field, type a unique name for the pool.
4. Assign the **https** or **https_443** health monitor from the **Available** list by moving it to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Add each resource that you want to include in the pool using the **New Members** setting:
   a) Type an IP address in the **Address** field.
   b) Type `443` in the **Service Port** field, or select **HTTPS** from the list.
   c) (Optional) Type a priority number in the **Priority** field.
   d) Click **Add**.
8. Click **Finished**.

The HTTPS load balancing pool now appears in the Pool List screen.

Creating a virtual server for client-side and server-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTP traffic over SSL.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

   The IP address you type must be available and not in the loopback network.
5. Type `443` in the **Service Port** field, or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system ensures that SSL authentication and encryption occurs for both client-side and server-side HTTP traffic. The system performs this authentication and encryption according to the values you specify in the Client SSL and Server SSL profiles.

Chapter

12

Implementing SSL Forward Proxy on a Single BIG-IP System

Overview: SSL forward proxy client and server authentication

With the BIG-IP® system's *SSL forward proxy* functionality, you can encrypt all traffic between a client and the BIG-IP system, by using one certificate, and to encrypt all traffic between the BIG-IP system and the server, by using a different certificate.

A client establishes a three-way handshake and SSL connection with the wildcard IP address of the BIG-IP system virtual server. The BIG-IP system then establishes a three-way handshake and SSL connection with the server, and receives and validates a server certificate (while maintaining the separate connection with the client). The BIG-IP system uses the server certificate to create a second unique server certificate to send to the client. The client receives the second server certificate from the BIG-IP system, but recognizes the certificate as originating directly from the server.

Important: *To enable SSL forward proxy functionality, you can either:*

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
- Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.

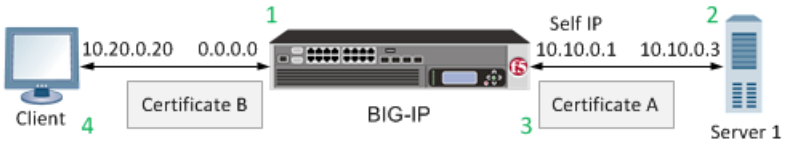


Figure 8: A virtual server configured with Client and Server SSL profiles for SSL forward proxy functionality

1. Client establishes three-way handshake and SSL connection with wildcard IP address.
2. BIG-IP system establishes three-way handshake and SSL connection with server.
3. BIG-IP system validates a server certificate (Certificate A), while maintaining the separate connection with the client.
4. BIG-IP system creates different server certificate (Certificate B) and sends it to client.

Task summary

To implement SSL forward proxy client-to-server authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the SSL Forward Proxy feature in both profiles.

Task list

Creating a custom Client SSL forward proxy profile
Creating a custom Server SSL forward proxy profile
Creating a load balancing pool
Creating a virtual server for client-side and server-side SSL traffic

Creating a custom Client SSL forward proxy profile

You perform this task to create a Client SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. From the **SSL Forward Proxy** list, select **Advanced**.
6. Select the **Custom** check box for the SSL Forward Proxy area.
7. Modify the SSL Forward Proxy settings.
   a) From the **SSL Forward Proxy** list, select **Enabled**.
   b) From the **CA Certificate** list, select a certificate.
   c) From the **CA Key** list, select a key.
   d) In the **CA Passphrase** field, type a passphrase.
   e) In the **Confirm CA Passphrase** field, type the passphrase again.
   f) In the **Certificate Lifespan** field, type a lifespan for the SSL forward proxy certificate in days.
   g) (Optional) From the **Certificate Extensions** list, select **Extensions List**.
   h) (Optional) For the **Certificate Extensions List** setting, select the extensions that you want in the **Available extensions** field, and move them to the **Enabled Extensions** field using the **Enable** button.
   i) Select the **Cache Certificate by Addr-Port** check box if you want to cache certificates by IP address and port number.
   j) From the **SSL Forward Proxy Bypass** list, select **Enabled**.
   Additional settings display.
   k) From the **Bypass Default Action** list, select **Intercept** or **Bypass**.

   The default action applies to addresses and hostnames that do not match any entry specified in the lists that you specify. The system matches traffic first against destination IP address lists, then source IP address lists, and lastly, hostname lists. Within these, the default action also specifies whether to search the intercept list or the bypass list first.

   Note: *If you select **Bypass** and do not specify any additional settings, you introduce a security risk to your system.*

8. Click **Finished**.

The custom Client SSL forward proxy profile now appears in the Client SSL profile list screen.

Creating a custom Server SSL forward proxy profile

You perform this task to create a Server SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.

The SSL Server profile list screen opens.

2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list select **serverssl**.
5. Select the **Custom** check box for the Configuration area.
6. From the **SSL Forward Proxy** list, select **Enabled**.
7. Click **Finished**.

The custom Client SSL forward proxy profile now appears in the Client SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. Specify the **Destination** settings.
   - For a Host, in the **Address** field, type `0.0.0.0` for the virtual server address.
   - For a Network, in the **Address** field, type `0.0.0.0` for the virtual server address, and in the **Mask** field, type `0.0.0.0` for the mask.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable SSL forward proxy functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
   - Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.
7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable SSL forward proxy functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
   - Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.
8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system ensures that the client system and server system can authenticate each other independently. After client and server authentication, the BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Chapter

13

Implementing Proxy SSL on a Single BIG-IP System

Overview: Direct client-server authentication with application optimization

When setting up the BIG-IP® system to process application data, you might want the destination server to authenticate the client system directly, for security reasons, instead of relying on the BIG-IP system to perform this function. Retaining direct client-server authentication provides full transparency between the client and server systems, and grants the server final authority to allow or deny client access.

The feature that makes it possible for this direct client-server authentication is known as *Proxy SSL*. You enable this feature when you configure the Client SSL and Server SSL profiles.

Note: *To use this feature, you must configure both a Client SSL and a Server SSL profile.*

Without the Proxy SSL feature enabled, the BIG-IP system establishes separate client-side and server-side SSL connections and then manages the initial authentication of both the client and server systems.

With the Proxy SSL feature, the BIG-IP system makes it possible for direct client-server authentication by establishing a secure SSL tunnel between the client and server systems and then forwarding the SSL handshake messages from the client to the server and vice versa. After the client and server successfully authenticate each other, the BIG-IP system uses the tunnel to decrypt the application data and intelligently manipulate (optimize) the data as needed.

Task summary

To implement direct client-to-server SSL authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the Proxy SSL feature in both profiles.

Before you begin, verify that the client system, server system, and BIG-IP® system contain the appropriate SSL certificates for mutual authentication.

Important: *The BIG-IP certificate and key referenced in a Server SSL profile must match those of the server system.*

As you configure your network for Proxy SSL, keep in mind the following considerations:

- Proxy SSL supports only the RSA key exchange. For proper functioning, the client and server must not negotiate key exchanges or cipher suites that Proxy SSL does not support, such as the Diffie-Hellman (DH) and Ephemeral Diffie-Hellman (DHE) key exchanges, and the Elliptic Curve Cryptography (ECC) cipher suite. To avoid this issue, you can either configure the client so that the ClientHello packet does not include DH, DHE, or ECC; or configure the server to not accept DH, DHE, or ECC.
- Proxy SSL supports only the NULL compression method.

Task list

Creating a custom Server SSL profile
Creating a custom Client SSL profile
Creating a load balancing pool
Creating a virtual server for client-side and server-side SSL traffic

Creating a custom Server SSL profile

You perform this task to create a Server SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL traffic only.

Important: *The certificate and key that you specify in this profile must match the certificate/key pair that you expect the back-end server to offer. If the back-end server has two or more certificates to offer, you must create a separate Server SSL profile for each certificate and then assign all of the Server SSL profiles to a single virtual server.*

---

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Server**.
   The SSL Server profile list screen opens.
2. Click **Create**.
   The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Certificate** list, select a relevant certificate name.
6. From the **Key** list, select a relevant key name.
7. For the **Proxy SSL** setting, select the check box.
8. From the **Configuration** list, select **Advanced**.
9. Modify all other settings, as required.
10. Choose one of the following actions:
    - If you need to create another Server SSL profile, click **Repeat**.
    - If you do not need to create another Server SSL profile, click **Finished**.

All relevant Server SSL profiles now appear on the SSL Server profile list screen.

Creating a custom Client SSL profile

You perform this task to create a Client SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL traffic only.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. For the **Proxy SSL** setting, select the check box.
6. From the **Configuration** list, select **Advanced**.
7. Modify all other settings, as required.
8. Click **Finished**.

The custom Client SSL profile now appears in the Client SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: *You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic** > **Pools**.
   The Pool List screen opens.
2. Click **Create**.
   The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click **<<** to move the monitor to the **Active** list.

   Tip: *Hold the Shift or Ctrl key to select more than one monitor at a time.*
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

   The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
   - Select **Disabled** to disable priority groups. This is the default option.
   - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
   a) Type an IP address in the **Address** field.
   b) Type a port number in the **Service Port** field, or select a service name from the list.
   c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
   d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select the type, and type an address, or an address and mask, as appropriate for your network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.

6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the custom Client SSL proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable proxy SSL functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
   - Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable proxy SSL functionality.

7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the custom Server SSL proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

   Important: *To enable SSL proxy functionality, you can either:*

   - Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
   - Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

   Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL proxy functionality.

8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system ensures that the client system and server system can initially authenticate each other directly. After client-server authentication, the BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Chapter 14

Securing Client-side SMTP Traffic

Overview: Securing client-side SMTP traffic

You can add SSL encryption to SMTP traffic quickly and easily, by configuring an SMTPS profile on the BIG-IP® system. *SMTPS* is a method for securing Simple Mail Transport Protocol (SMTP) connections at the transport layer.

Normally, SMTP traffic between SMTP servers and clients is unencrypted. This creates a privacy issue because SMTP traffic often passes through routers that the servers and clients do not trust, resulting in a third party potentially changing the communications between the server and client. Also, two SMTP systems do not normally authenticate each other. A more secure SMTP server might only allow communications from other known SMTP systems, or the server might act differently with unknown systems.

To mitigate these problems, the BIG-IP system includes an SMTPS profile that you can configure. When you configure an SMTPS profile, you can activate support for the industry-standard STARTTLS extension to the SMTP protocol, by instructing the BIG-IP system to either allow, disallow, or require STARTTLS activation for SMTP traffic. The STARTTLS extension effectively upgrades a plain-text connection to an encrypted connection on the same port, instead of using a separate port for encrypted communication.

This illustration shows a basic configuration of a BIG-IP system that uses SMTPS to secure SMTP traffic between the BIG-IP system and an SMTP mail server.

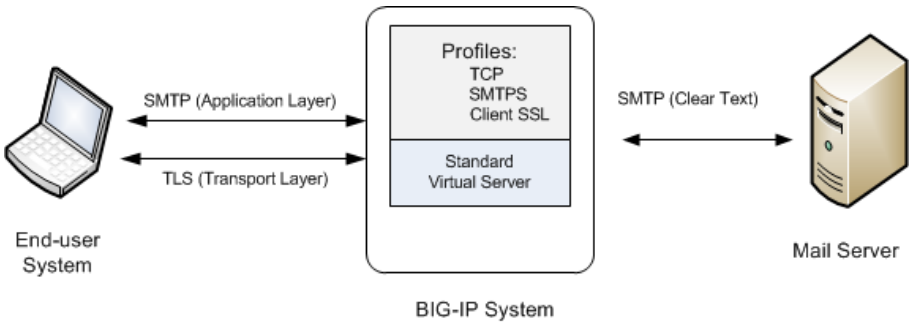


Figure 9: Sample BIG-IP configuration for SMTP traffic with STARTTLS activation

Task summary

To configure the BIG-IP ®system to process Simple Mail Transport Protocol (SMTP) traffic with SSL functionality, you perform a few basic tasks.

Task list

Creating an SMTPS profile
Creating a Client SSL profile
Creating a virtual server and load-balancing pool

Creating an SMTPS profile

This task specifies that STARTTLS authentication and encryption should be required for all client-side Simple Mail Transport Protocol (SMTP) traffic. When you require STARTTLS for SMTP traffic, the BIG-IP® system effectively upgrades SMTP connections to include SSL, on the same SMTP port.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **SMTPS**.
   The SMTPS profile list screen opens.
2. Click **Create**.
   The New SMTPS Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. From the **STARTTLS Activation Mode** list, select **Require**.
6. Click **Finished**.

The BIG-IP system is now required to activate STARTTLS for all client-side SMTP traffic.

Creating a Client SSL profile

You create a Client SSL profile when you want the BIG-IP® system to authenticate and decrypt/encrypt client-side application traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
   The Client profile list screen opens.
2. Click **Create**.
   The New Client SSL Profile screen opens.
3. Configure all profile settings as needed.
4. Click **Finished**.

After creating the Client SSL profile and assigning the profile to a virtual server, the BIG-IP system can apply SSL security to the type of application traffic for which the virtual server is configured to listen.

Creating a virtual server and load-balancing pool

You use this task to create a virtual server, as well as a default pool of Simple Mail Transport Protocol (SMTP) servers. The virtual server listens for, and applies SSL security to, client-side SMTP application traffic. The virtual server then forwards the SMTP traffic on to the specified server pool.

Note: *Using this task, you assign an SMTPS profile to the virtual server instead of an SMTP profile. You must also assign a Client SSL profile.*

---

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select the type, and type an address, or an address and mask, as appropriate for your network.
5. In the **Service Port** field, type `25` or select **SMTP** from the list.
6. From the **Configuration** list, select **Basic**.
7. For the **SSL Profile (Client)** setting, in the **Available** box, select a profile name, and using the Move button, move the name to the **Selected** box.
8. From the **SMTPS Profile** list, select the SMTPS profile that you previously created.
9. In the Resources area of the screen, for the **Default Pool** setting, click the **Create (+)** button.

The New Pool screen opens.

10. In the **Name** field, type a unique name for the pool.
11. In the Resources area, for the **New Members** setting, select the type of new member you are adding, then type the appropriate information in the **Node Name**, **Address**, and **Service Port** fields, and click **Add** to add as many pool members as you need.
12. Click **Finished** to create the pool.
    The screen refreshes, and reopens the New Virtual Server screen. The new pool name appears in the **Default Pool** list.
13. Click **Finished**.

After performing this task, the virtual server applies the custom SMTPS and Client SSL profiles to incoming SMTP traffic.

Implementation result

After you have created an SMTPS profile and a Client SSL profile and assigned them to a virtual server, the BIG-IP system listens for client-side SMTP traffic on port 25. The BIG-IP system then activates the STARTTLS method for that traffic, to provide SSL security on that same port, before forwarding the traffic on to the specified server pool.

Index

O

P

R

S

T

U

V