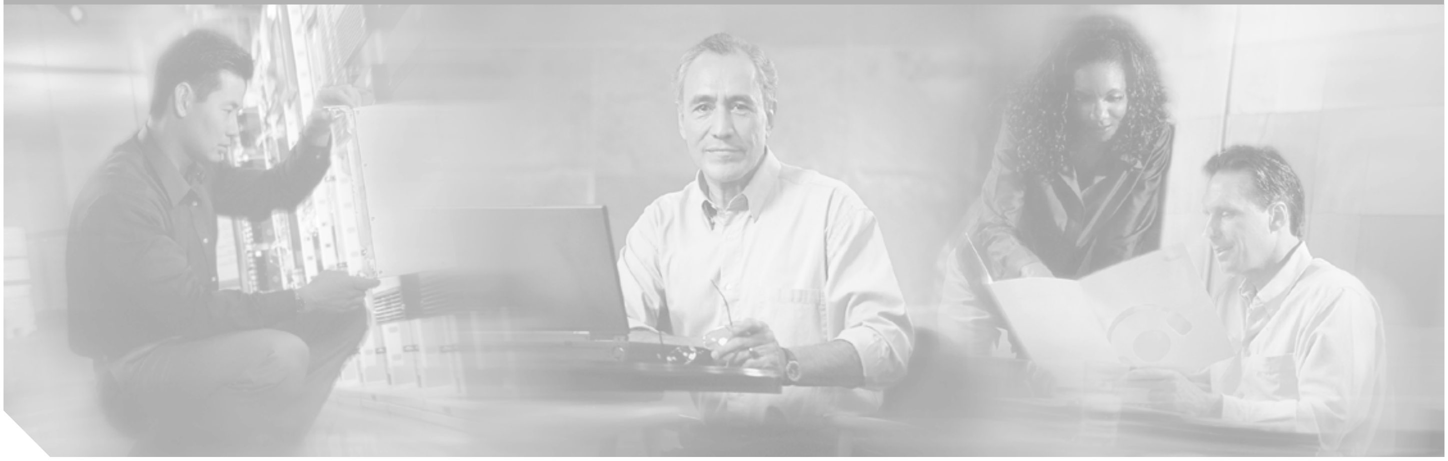


Catalyst 6500 Series SSL Services Module Installation and Configuration Note

Software Release 1.2
May, 2003

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Text Part Number: 78-15609-01

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment 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 users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: The equipment described in this manual generates and may radiate radio-frequency energy. If it is not installed in accordance with Cisco's installation instructions, it may cause interference with radio and television reception. This equipment has been tested and found to comply with the limits for a Class B digital device in accordance with the specifications in part 15 of the FCC rules. These specifications are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation.

Modifying the equipment without Cisco's written authorization may result in the equipment no longer complying with FCC requirements for Class A or Class B digital devices. In that event, your right to use the equipment may be limited by FCC regulations, and you may be required to correct any interference to radio or television communications at your own expense.

You can determine whether your equipment is causing interference by turning it off. If the interference stops, it was probably caused by the Cisco equipment or one of its peripheral devices. If the equipment causes interference to radio or television reception, try to correct the interference by using one or more of the following measures:

- Turn the television or radio antenna until the interference stops.

- Move the equipment to one side or the other of the television or radio.

- Move the equipment farther away from the television or radio.

- Plug the equipment into an outlet that is on a different circuit from the television or radio. (That is, make certain the equipment and the television or radio are on circuits controlled by different circuit breakers or fuses.)

Modifications to this product not authorized by Cisco Systems, Inc. could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

Catalyst 6500 Series SSL Services Module Installation and Configuration Note

Copyright Notices

Third-party software used under license accompanies the Cisco SSL Services Module Software release 1.1(1). One or more of the following notices may apply in connection with the license and use of such third-party software.

GNU General Public License

The Catalyst 6500 Series SSL Services Module contains software covered under the GNU Public License (listed below). If you would like to obtain the source for the modified GPL code in the SSL Services Module, please send a request to ssl_sw_req@Cisco.com.

License Text

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program,” below, refers to any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you.”

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. END OF TERMS AND CONDITIONS.

CONTENTS

CHAPTER 3

APPENDIX A

APPENDIX B

Preface

This preface describes who should read the *Catalyst 6500 Series SSL Services Module Installation and Configuration Note*, how it is organized, and its document conventions.

This publication does not contain the instructions to install the Catalyst 6500 series switch chassis. For information on installing the switch chassis, refer to the *Catalyst 6500 Series Switch Installation Guide*.

Audience

Only trained and qualified service personnel (as defined in IEC 60950 and AS/NZS3260) should install, replace, or service the equipment described in this publication.

Organization

This publication is organized as follows:

Chapter	Title	Description
Chapter 1	Overview	Presents an overview of the Catalyst 6500 series SSL Services Module.
Chapter 2	Installing and Removing the SSL Services Module	Describes how to install and remove the SSL Services Module.
Chapter 3	Configuring the SSL Services Module	Describes how to configure the SSL Services Module.
Appendix A	Testing SSL Proxy Services	Contains information for testing or troubleshooting SSL proxy services.
Appendix B	Command Reference	Contains the commands that allow you to set up and manage the SSL Services Module.
Appendix C	System Messages	Lists and describes the system messages for the SSL Services Module.

Conventions

This publication uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
`italic screen` font	Arguments for which you supply values are in *`italic screen`* font.
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

Notes use the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Tips use the following conventions:

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Cautions use the following conventions:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warnings use the following conventions:

Warning

This warning symbol means *danger*. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. To see translations of the warnings that appear in this publication, refer to the *Regulatory Compliance and Safety Information* document that accompanied this device.

Waarschuwing

Dit waarschuwingssymbool betekent gevaar. U verkeert in een situatie die lichamelijk letsel kan veroorzaken. Voordat u aan enige apparatuur gaat werken, dient u zich bewust te zijn van de bij elektrische schakelingen betrokken risico's en dient u op de hoogte te zijn van standaard maatregelen om ongelukken te voorkomen. Voor vertalingen van de waarschuwingen die in deze publicatie verschijnen, kunt u het document *Regulatory Compliance and Safety Information* (Informatie over naleving van veiligheids- en andere voorschriften) raadplegen dat bij dit toestel is ingesloten.

Varoitus

Tämä varoitusmerkki merkitsee vaaraa. Olet tilanteessa, joka voi johtaa ruumiinvammaan. Ennen kuin työskentelet minkään laitteiston parissa, ota selvää sähkökytkentöihin liittyvistä vaaroista ja tavanomaisista onnettomuuksien ehkäisykeinoista. Tässä julkaisussa esiintyvien varoitusten käännökset löydät laitteen mukana olevasta *Regulatory Compliance and Safety Information* -kirjasesta (määräysten noudattaminen ja tietoa turvallisuudesta).

Attention

Ce symbole d'avertissement indique un danger. Vous vous trouvez dans une situation pouvant causer des blessures ou des dommages corporels. Avant de travailler sur un équipement, soyez conscient des dangers posés par les circuits électriques et familiarisez-vous avec les procédures couramment utilisées pour éviter les accidents. Pour prendre connaissance des traductions d'avertissements figurant dans cette publication, consultez le document *Regulatory Compliance and Safety Information* (Conformité aux règlements et consignes de sécurité) qui accompagne cet appareil.

Warnung

Dieses Warnsymbol bedeutet Gefahr. Sie befinden sich in einer Situation, die zu einer Körperverletzung führen könnte. Bevor Sie mit der Arbeit an irgendeinem Gerät beginnen, seien Sie sich der mit elektrischen Stromkreisen verbundenen Gefahren und der Standardpraktiken zur Vermeidung von Unfällen bewußt. Übersetzungen der in dieser Veröffentlichung enthaltenen Warnhinweise finden Sie im Dokument *Regulatory Compliance and Safety Information* (Informationen zu behördlichen Vorschriften und Sicherheit), das zusammen mit diesem Gerät geliefert wurde.

Avvertenza

Questo simbolo di avvertenza indica un pericolo. La situazione potrebbe causare infortuni alle persone. Prima di lavorare su qualsiasi apparecchiatura, occorre conoscere i pericoli relativi ai circuiti elettrici ed essere al corrente delle pratiche standard per la prevenzione di incidenti. La traduzione delle avvertenze riportate in questa pubblicazione si trova nel documento *Regulatory Compliance and Safety Information* (Conformità alle norme e informazioni sulla sicurezza) che accompagna questo dispositivo.

Advarsel **Dette varselsymbolet betyr fare. Du befinner deg i en situasjon som kan føre til personskade. Før du utfører arbeid på utstyr, må du vare oppmerksom på de faremomentene som elektriske kretser innebærer, samt gjøre deg kjent med vanlig praksis når det gjelder å unngå ulykker. Hvis du vil se oversettelser av de advarslene som finnes i denne publikasjonen, kan du se i dokumentet *Regulatory Compliance and Safety Information* (Overholdelse av forskrifter og sikkerhetsinformasjon) som ble levert med denne enheten.**

Aviso **Este símbolo de aviso indica perigo. Encontra-se numa situação que lhe poderá causar danos físicos. Antes de começar a trabalhar com qualquer equipamento, familiarize-se com os perigos relacionados com circuitos eléctricos, e com quaisquer práticas comuns que possam prevenir possíveis acidentes. Para ver as traduções dos avisos que constam desta publicação, consulte o documento *Regulatory Compliance and Safety Information* (Informação de Segurança e Disposições Reguladoras) que acompanha este dispositivo.**

¡Advertencia! **Este símbolo de aviso significa peligro. Existe riesgo para su integridad física. Antes de manipular cualquier equipo, considerar los riesgos que entraña la corriente eléctrica y familiarizarse con los procedimientos estándar de prevención de accidentes. Para ver una traducción de las advertencias que aparecen en esta publicación, consultar el documento titulado *Regulatory Compliance and Safety Information* (Información sobre seguridad y conformidad con las disposiciones reglamentarias) que se acompaña con este dispositivo.**

Varning! **Denna varningssymbol signalerar fara. Du befinner dig i en situation som kan leda till personskada. Innan du utför arbete på någon utrustning måste du vara medveten om farorna med elkretsar och känna till vanligt förfarande för att förebygga skador. Se förklaringar av de varningar som förkommer i denna publikation i dokumentet *Regulatory Compliance and Safety Information* (Efterrättelse av föreskrifter och säkerhetsinformation), vilket medföljer denna anordning.**

Related Documentation

For more detailed installation and configuration information, refer to the following publications:

- *Release Notes for Catalyst 6500 Series SSL Services Module Software Release 1.x*
- *Regulatory Compliance and Safety Information for the Catalyst 6500 Series Switches*
- *Catalyst 6500 Series Switch Installation Guide*
- *Catalyst 6500 Series Switch Module Installation Guide*
- *Catalyst 6500 Series Switch Software Configuration Guide*
- *Catalyst 6500 Series Switch Command Reference*
- *Catalyst 6500 Series Switch IOS Software Configuration Guide*
- *Catalyst 6500 Series Switch IOS Command Reference*
- *Site Preparation and Safety Guide*
- *System Message Guide—Catalyst 6500 Series, 5000 Family, 4000 Family, 2926G Series, 2948G and 2980G Switches*

Obtaining Documentation

Cisco provides several ways to obtain documentation, technical assistance, and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

International Cisco websites can be accessed from this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which may have shipped with your product. The Documentation CD-ROM is updated regularly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual or quarterly subscription.

Registered Cisco.com users can order a single Documentation CD-ROM (product number DOC-CONDOCCD=) through the Cisco Ordering tool:

http://www.cisco.com/en/US/partner/ordering/ordering_place_order_ordering_tool_launch.html

All users can order monthly or quarterly subscriptions through the online Subscription Store:

http://www.cisco.com/go/subscription

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/en/US/partner/ordering/index.shtml

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. On the Cisco Documentation home page, click **Feedback** at the top of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com, which includes the Cisco Technical Assistance Center (TAC) website, as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from the Cisco TAC website. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC website, including TAC tools and utilities.

Cisco.com

Cisco.com offers a suite of interactive, networked services that let you access Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

To obtain customized information and service, you can self-register on Cisco.com at this URL:

http://tools.cisco.com/RPF/register/register.do

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two types of support are available: the Cisco TAC website and the Cisco TAC Escalation Center. The type of support that you choose depends on the priority of the problem and the conditions stated in service contracts, when applicable.

We categorize Cisco TAC inquiries according to urgency:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration. There is little or no impact to your business operations.
- Priority level 3 (P3)—Operational performance of the network is impaired, but most business operations remain functional. You and Cisco are willing to commit resources during normal business hours to restore service to satisfactory levels.
- Priority level 2 (P2)—Operation of an existing network is severely degraded, or significant aspects of your business operations are negatively impacted by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.
- Priority level 1 (P1)—An existing network is "down," or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Cisco TAC Website

The Cisco TAC website provides online documents and tools to help troubleshoot and resolve technical issues with Cisco products and technologies. To access the Cisco TAC website, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC website. Some services on the Cisco TAC website require a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://tools.cisco.com/RPF/register/register.do

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC website, you can open a case online at this URL:

http://www.cisco.com/tac/caseopen

If you have Internet access, we recommend that you open P3 and P4 cases online so that you can fully describe the situation and attach any necessary files.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- The *Cisco Product Catalog* describes the networking products offered by Cisco Systems, as well as ordering and customer support services. Access the *Cisco Product Catalog* at this URL:

  http://www.cisco.com/en/US/products/products_catalog_links_launch.html

- Cisco Press publishes a wide range of networking publications. Cisco suggests these titles for new and experienced users: *Internetworking Terms and Acronyms Dictionary, Internetworking Technology Handbook, Internetworking Troubleshooting Guide,* and the *Internetworking Design Guide.* For current Cisco Press titles and other information, go to Cisco Press online at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco quarterly publication that provides the latest networking trends, technology breakthroughs, and Cisco products and solutions to help industry professionals get the most from their networking investment. Included are networking deployment and troubleshooting tips, configuration examples, customer case studies, tutorials and training, certification information, and links to numerous in-depth online resources. You can access *Packet* magazine at this URL:

  http://www.cisco.com/go/packet

- iQ Magazine is the Cisco bimonthly publication that delivers the latest information about Internet business strategies for executives. You can access iQ Magazine at this URL:

  http://www.cisco.com/go/iqmagazine

- Internet Protocol Journal is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets. You can access the Internet Protocol Journal at this URL:

  http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html

- Training—Cisco offers world-class networking training. Current offerings in network training are listed at this URL:

  http://www.cisco.com/en/US/learning/le31/learning_recommended_training_list.html

Overview

The SSL Services Module is a Layer 4-through-Layer 7 service module that you can install into the Catalyst 6500 series switch. The module terminates secure sockets layer (SSL) transactions and accelerates the encryption and decryption of data used in SSL sessions.

The module operates either in a standalone configuration or with the Content Switching Module (CSM). In a standalone configuration, secure traffic is directed to the module using policy-based routing (PBR). When used with the CSM, only encrypted client traffic is forwarded to the module, while clear text traffic is forwarded to the real servers.

The SSL Services Module uses the SSL protocol to enable secure transactions of data through privacy, authentication, and data integrity; the protocol relies upon certificates, public keys, and private keys.

The certificates, which are similar to digital ID cards, verify the identity of the server to the clients. The certificates, which are issued by certificate authorities, include the name of the entity to which the certificate was issued, the entity's public key, and the time stamps that indicate the certificate's expiration date.

The public and private keys are the ciphers that are used to encrypt and decrypt information. The public key is shared without any restrictions, but the private key is never shared. Each public-private key pair works together; data that is encrypted with the public key can only be decrypted with the corresponding private key.

These sections describe the SSL Services Module:

- Features, page 1-1
- Front Panel Description, page 1-5

Features

The SSL Services Module has these features:

- Accelerates SSL transactions to help alleviate the server processing load
- Enables intelligent content switching using the CSM to load balance traffic through the server
- Provides centralized management (for the key, certificate, and configuration management)

Table 1-1 lists the available features.

Table 1-1 Feature Set Description

Features
Supported Hardware
Supervisor Engine 2 with MSFC2[1] and PFC2[2]
Supported Software
– Cisco IOS Release 12.1(13)E or later on the MSFC2 – Cisco IOS Release 12.1(13)E3 or later on the MSFC2 and Catalyst software release 7.5(1) or later on the Supervisor Engine 2 – SSL Services Module software release 1.1(1) or later on the SSL Services Module
Handshake Protocol
SSL 3.0
SSL 3.1/TLS 1.0
SSL 2.0 (only ClientHello support)
Session reuse
Session renegotiation
Session timeout[3]
Symmetric Algorithms
ARC4
DES
3DES
Asymmetric Algorithms
RSA
Hash Algorithms
MD5
SHA1
Cipher Suites
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CSC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
Public Key Infrastructure
RSA key pair generation for server certificates up to 2048-bit
Secure server key storage in SSL Services Module Flash memory device
Server certificate enrollment
Importing and exporting of server key and certificate (PKCS12 and PEM[3])
Duplicating keys and certificates on standby SSL Services Module using the key and certificate import and export mechanism

Table 1-1 Feature Set Description (continued)

Features
Public Key Infrastructure (continued)
Manual key archival, recovery, and backup
Key and certificate renewal using the CLI
Graceful rollover of expiring server keys and certificates
Auto-enrollment of server certificates
Exporting of PKCS10 CRT file for manual or offline certificate enrollment[3]
Importing of CA certificates by cut-and-paste or TFTP[3]
Up to 8 levels of Certificate Authority in a certificate chain[3]
Generating of self-signed certificate[3]
Manual certificate enrollment using cut-and-paste or TFTP[3]
TCP Termination
RFC 1323
Connection aging
Connection rate
Up to 64,000 concurrent client connections
Up to 192,000 concurrent connections (includes 2 MSL[4])
Up to 300 Mbps throughput
NAT[5]
Client NAT
Server NAT/PAT[6]
Scalability
Multiple modules in a single chassis when used with the CSM[7]; the CSM provides server load balancing (SLB)
High Availability
Failure detection (SLB[8] health monitoring schemes)
System-level redundancy (stateless) (when used with the CSM)
Module-level redundancy (stateless) (when used with the CSM)
Serviceability
OIR[9] (after properly shutdown)
Graceful shutdown
Statistics and Accounting
Total SSL connections attempt per virtual server
Total SSL connections successfully established per virtual server
Total SSL connections failed per virtual server
Total SSL alert errors per virtual server

Table 1-1 Feature Set Description (continued)

Features
Statistics and Accounting (continued)
Total SSL resumed sessions per virtual server
Total encrypted/decrypted packets/bytes per virtual server
Statistics displayed at 1 second, 1 minute, and 5 minutes traffic rate for CPU utilization and SSL-specific counters
Configuration and Management
Direct connection to the module console port
Secure Shell (SSHv1) session
Telnet
Automatic backup and restore of configuration file to NVRAM[3]
System Capacity and Performance
Up to 300 Mbps throughput
Up to 256 proxy servers
Up to 64,000 simultaneous sessions
Stores up to 356 key pairs
Stores up to 356 certificates
Supports the following RSA key sizes: – 512-bits – 768-bits – 1024-bits – 1536-bits – 2048-bits
Up to 3000 sessions per second

[1] MSFC = Multilayer Switch Feature Card

[2] PFC = Policy Feature Card

[3] New feature in SSL software release 1.2(1).

[4] MSL = Maximum Segment Lifetime

[5] NAT = Network Address Translation

[6] PAT = Port Address Translation

[7] CSM = Content Switching Module

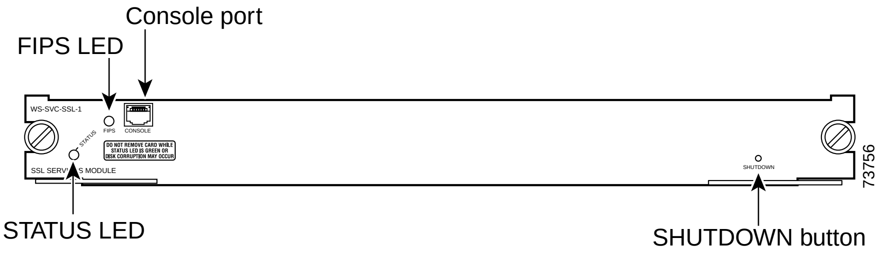
[8] SLB = Server Load Balancing

[9] OIR = Online Insertion and Removal

Front Panel Description

The SSL Services Module front panel (see Figure 1-1) includes a STATUS LED, a Federal Information Processing Standards (FIPS) LED, a SHUTDOWN button, and a console port.

Figure 1-1 SSL Services Module Front Panel



These sections describe the SSL Services Module front panel:

- STATUS LED, page 1-5
- FIPS LED, page 1-5
- SHUTDOWN Button, page 1-6

STATUS LED

The STATUS LED indicates the operating states of the module. Table 1-2 describes the LED operation.

Table 1-2 STATUS LED Description

Color	State	Description
Green	On	All diagnostic tests pass. The module is receiving power.
Red	On	A diagnostic other than an individual port test failed.
Orange	On	Indicates one of three conditions: • The module is running through its boot and self-test diagnostic sequence. • The module is disabled. • The module is in the shutdown state.
	Off	The module power is off.

FIPS LED

The FIPS LED currently is not used.

SHUTDOWN Button

Caution Do not remove the SSL Services Module from the switch until the module has shut down completely and the STATUS LED is orange. You can damage the module if you remove it from the switch before it completely shuts down.

To avoid corrupting the SSL Services Module hard disk, you must correctly shut down the SSL Services Module before you remove it from the chassis or disconnect the power. You can shut down the module by entering the **hw-mod module** *mod* **shutdown** command in privileged mode from the router CLI.

If the SSL Services Module fails to respond to this command, shut down the module by pressing the SHUTDOWN button on the front panel.

The shutdown procedure may require several minutes. The STATUS LED turns off when the module shuts down.

CHAPTER 2

Installing and Removing the SSL Services Module

This chapter describes how to install the SSL Services Module into the Catalyst 6500 series switch and contains these sections:

System Requirements

Before you install the SSL Services Module into the Catalyst 6500 series switch, make sure that the switch meets the hardware and software requirements listed in Table 1.

Table 1 System Requirements

Hardware	Minimum SSL Software Version		Recommended SSL Software Version		Minimum Cisco IOS Software	Minimum Catalyst Software
	Application Image	Maintenance Image	Application Image	Maintenance Image		
Supervisor Engine 2 with an MSFC2	1.1(1)	1.2(1)[1]	1.2(1)	1.2(1)[1]	12.1(13)E3	7.5(1)
					12.1(13)E	–

[1] Do not use 1.2(2) maintenance image.

Safety Overview

Safety warnings appear throughout this publication in procedures that, if performed incorrectly, may harm you. A warning symbol precedes each warning statement.

Warning **IMPORTANT SAFETY INSTRUCTIONS**

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. To see translations of the warnings that appear in this publication, refer to the translated safety warnings that accompanied this device.

Note: SAVE THESE INSTRUCTIONS

Note: This documentation is to be used in conjunction with the specific product installation guide that shipped with the product. Please refer to the Installation Guide, Configuration Guide, or other enclosed additional documentation for further details.

Warning **Before you install, operate, or service the system, read the *Site Preparation and Safety Guide*. This guide contains important safety information you should know before working with the system.**

Warning **Only trained and qualified personnel should be allowed to install, replace, or service this equipment.**

Warning **During this procedure, wear grounding wrist straps to avoid ESD damage to the card. Do not directly touch the backplane with your hand or any metal tool, or you could shock yourself.**

Warning **Blank faceplates and cover panels serve three important functions: they prevent exposure to hazardous voltages and currents inside the chassis; they contain electromagnetic interference (EMI) that might disrupt other equipment; and they direct the flow of cooling air through the chassis. Do not operate the system unless all cards, faceplates, front covers, and rear covers are in place.**

Installing the SSL Services Module

The following sections describe how to install the SSL Services Module into the Catalyst 6500 series switch:

- Preparing to Install the SSL Services Module, page 2-3
- Required Tools, page 2-3
- Installing the SSL Services Module, page 2-3

Preparing to Install the SSL Services Module

Before installing the SSL Services Module, make sure that the following items are available:

- Catalyst 6500 series switch chassis
- Management station that is available through a Telnet or a console connection to perform configuration tasks

Required Tools

Warning **Only trained and qualified personnel should be allowed to install, replace, or service this equipment.**

These tools are required to install the SSL Services Module into the Catalyst 6500 series switch:

- Flat-blade screwdriver
- Wrist strap or other grounding device
- Antistatic mat or antistatic foam

Installing the SSL Services Module

Note Before installing the SSL Services Module, you must install the Catalyst 6500 series switch chassis and at least one supervisor engine. For information on installing the switch chassis, refer to the *Catalyst 6500 Series Installation Guide*.

This section describes how to install the SSL Services Module into the Catalyst 6500 series switch.

Note All modules, including the supervisor engine (if you have redundant supervisor engines), support hot swapping. You can add, replace, or remove modules without interrupting the system power or causing other software or interfaces to shut down. For more information about hot-swapping modules, refer to the *Catalyst 6500 Series Module Installation Guide*.

Warning **During this procedure, wear grounding wrist straps to avoid ESD damage to the card. Do not directly touch the backplane with your hand or any metal tool, or you could shock yourself.**

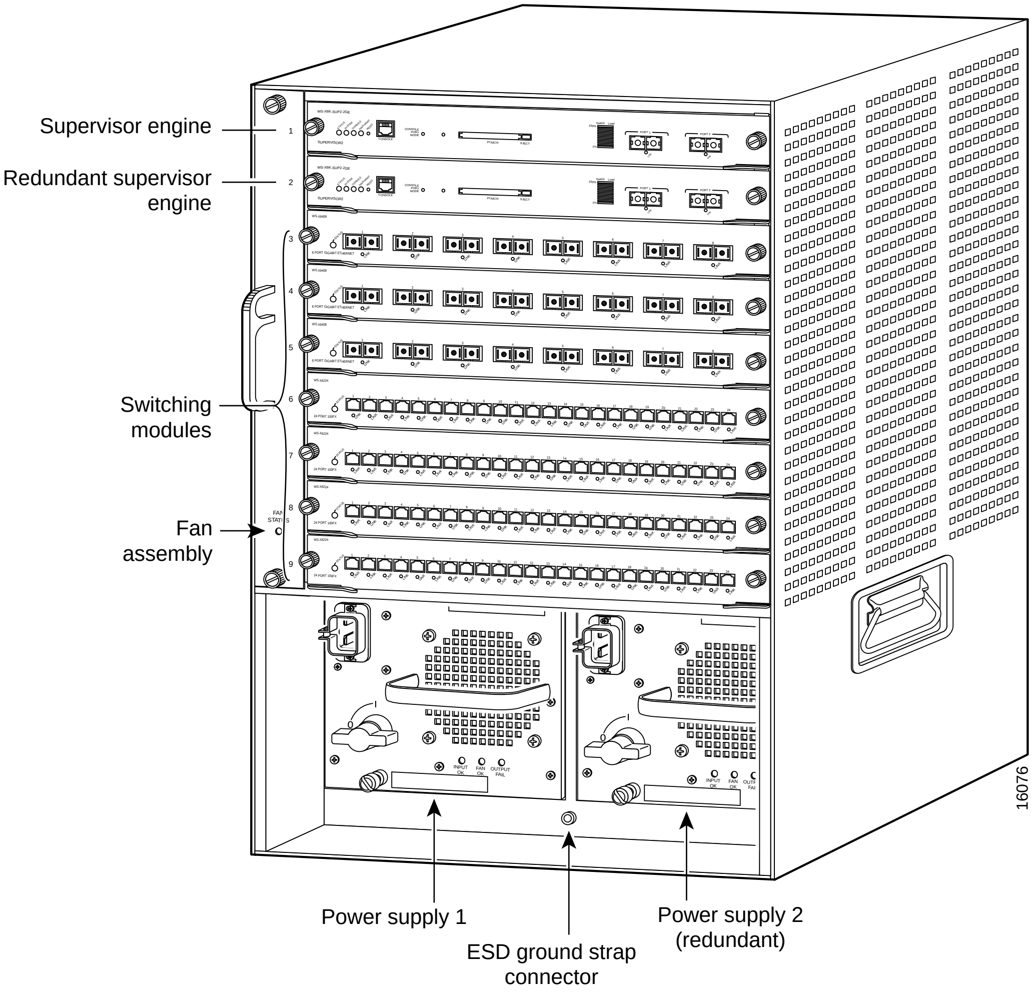
To install the SSL Services Module into the Catalyst 6500 series switch, perform these steps:

Step 1 Make sure that you take the necessary precautions to prevent ESD damage.

Step 2 Choose a slot for the SSL Services Module. See Figure 1 for the slot numbers on a Catalyst 6500 series switch.

Note Slot 1 is reserved for the supervisor engine. Slot 2 can contain an additional supervisor engine in case the supervisor engine in slot 1 fails. If a redundant supervisor engine is not required, you can insert the module in slots 2 through 6 on a 6-slot chassis, slots 2 through 9 on the 9-slot chassis, or slots 2 through 13 on the 13-slot chassis.

Figure 1 Slot Numbers on Catalyst 6500 Series Switches



Step 3 Check that there is enough clearance to accommodate any interface equipment that you will be connecting directly to the supervisor engine or switching module ports.

Note If possible, place switching modules between the empty slots that contain only switching-module filler plates (Cisco part number 800-00292-01).

Warning **Blank faceplates (filler panels) serve three important functions: they prevent exposure to hazardous voltages and currents inside the chassis; they contain electromagnetic interference (EMI) that might disrupt other equipment; and they direct the flow of cooling air through the chassis. Do not operate the system unless all cards and faceplates are in place.**

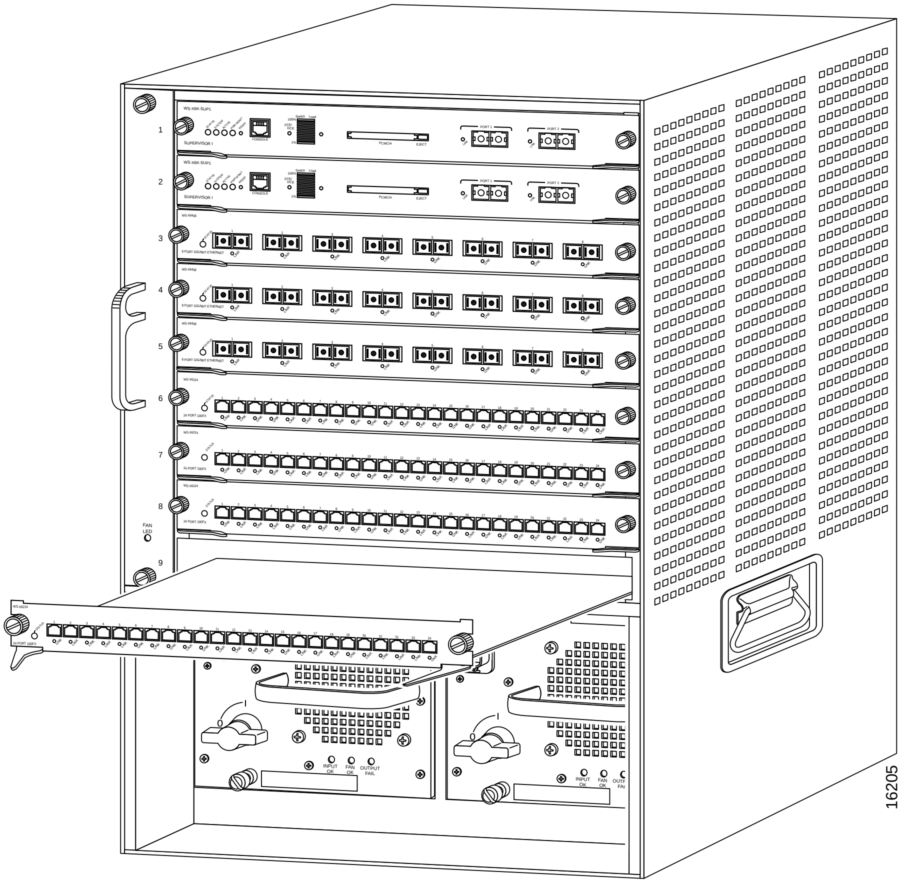
Step 4 Loosen the captive installation screws that secure the switching module filler plate (or an existing switching module) to the desired slot.

Step 5 Remove the switching module filler plate (or an existing switching module).

Step 6 Hold the handle of the SSL Services Module with one hand, and place your other hand under the carrier support. Do not touch the printed circuit boards or connector pins.

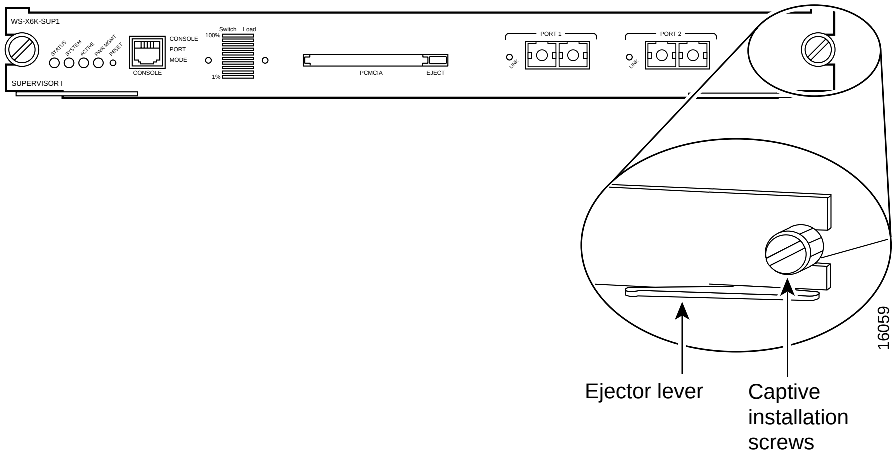
Step 7 Place the SSL Services Module in the slot. Align the notch on the sides of the switching module carrier with the groove in the slot. (See Figure 2.)

Figure 2 Installing Modules in the Catalyst 6500 Series Switch



Step 8 Keep the SSL Services Module at a 90-degree angle to the backplane and carefully slide the SSL Services Module into the slot until the switching module faceplate contacts the ejector levers. (See Figure 3.)

Figure 3 Ejector Levers and Captive Installation Screws



Step 9 Using the thumb and forefinger of each hand, simultaneously push in the left and right levers to fully seat the SSL Services Module in the backplane connector.

Caution Always use the ejector levers when installing or removing the SSL Services Module. A module that is partially seated in the backplane will cause the system to halt and subsequently crash.

Note If you perform a hot swap, the console displays the message “Module *n* has been inserted.” This message does not appear if you are connected to the Catalyst 6500 series switch through a Telnet session.

Step 10 Use a screwdriver to tighten the captive installation screws on the left and right ends of the SSL Services Module.

This completes the SSL Services Module installation procedure.

Verifying the Installation

When you install the SSL Services Module into the Catalyst 6500 series switch, the module goes through a boot sequence that requires no intervention. At the successful conclusion of the boot sequence, the green STATUS LED will light and remain on. If the STATUS LED is not green, or is a different color, see Table 1-2 on page 1-5 to determine the module’s status.

Removing the SSL Services Module

This section describes how to remove the SSL Services Module from the Catalyst 6500 series switch.

Caution Do not remove the SSL Services Module from the switch until the module has shut down completely and the STATUS LED is orange or off. You can damage the module if you remove it from the switch before it completely shuts down.

Warning **During this procedure, wear grounding wrist straps to avoid ESD damage to the card. Do not directly touch the backplane with your hand or any metal tool, or you could shock yourself.**

To remove the SSL Services Module, perform these steps:

Step 1 Shut down the module by one of these methods:

- In privileged mode from the router prompt, enter the **hw-mod module** *mod* **shutdown** command.

  Note If you enter this command to shut down the module, you will have to enter the following commands in config mode to restart (power down, and then power up) the module:

  ```
  Router# no power enable module mod
  Router# power enable module mod
  ```

- If the module does not respond to any commands, press the SHUTDOWN button located on the front panel of the module.

Note Shutdown may require several minutes.

Step 2 Verify that the SSL Services Module shuts down. Do not remove the module from the switch until the STATUS LED is off or orange.

Step 3 Use a screwdriver to loosen the captive installation screws at the left and right sides of the module.

Step 4 Grasp the left and right ejector levers. Simultaneously, pull the left lever to the left and the right lever to the right to release the module from the backplane connector.

Step 5 As you pull the module out of the slot, place one hand under the carrier to support it. Avoid touching the module itself.

Step 6 Carefully pull the module straight out of the slot, keeping one hand under the carrier to guide it. Keep the module at a 90-degree orientation to the backplane (horizontal to the floor).

Step 7 Place the removed module on an antistatic mat or antistatic foam.

Warning **Blank faceplates and cover panels serve three important functions: they prevent exposure to hazardous voltages and currents inside the chassis; they contain electromagnetic interference (EMI) that might disrupt other equipment; and they direct the flow of cooling air through the chassis. Do not operate the system unless all cards, faceplates, front covers, and rear covers are in place.**

Step 8 If the slot is to remain empty, install a module filler plate to keep dust out of the chassis and to maintain proper airflow through the module compartment.

CHAPTER 3

Configuring the SSL Services Module

This chapter describes how to configure the SSL Services Module from the Command Line Interface (CLI) of the module:

- Using the CLI, page 3-1
- Preparing to Configure the SSL Services Module, page 3-1
- Upgrading the Images, page 3-13
- Configuring the SSL Services Module, page 3-20
- Configuring Different Modes of Operation, page 3-55
- Advanced Configuration, page 3-46

Using the CLI

The software interface for the SSL Services Module is the Cisco IOS CLI. To understand the Cisco IOS CLI and Cisco IOS command modes, refer to Chapter 2, “Command-Line Interfaces,” in the *Catalyst 6500 Series Switch Cisco IOS Software Configuration Guide*.

Unless your switch is located in a fully trusted environment, we recommend that you configure the SSL Services Module through a direct connection to the module’s console port or through an encrypted session using Secure Shell (SSH). See the “Configuring SSH” section on page 3-4 for information on configuring SSH on the module.

Note The initial SSL Services Module configuration must be made through a direct connection to the module’s console port.

Preparing to Configure the SSL Services Module

Before you configure services on the SSL Services Module, you must do the following:

- Initial SSL Services Module Configuration, page 3-2
- Initial Catalyst 6500 Series Switch Configuration, page 3-6

Initial SSL Services Module Configuration

Note You are required to make the following initial SSL Services Module configurations through a direct connection to the SSL Services Module console port. After the initial configurations, you can make an SSH or Telnet connection to the module to further configure the module.

The initial SSL Services Module configuration consists of the following tasks:

- Configuring VLANs on the SSL Services Module, page 3-2
- Configuring Telnet Remote Access, page 3-3
- Configuring the Fully Qualified Domain Name, page 3-3
- Configuring SSH, page 3-4

Configuring VLANs on the SSL Services Module

When you configure VLANs on the SSL Services Module, configure one of the VLANs as an admin VLAN. The admin VLAN is used for all management traffic, including SSH, public key infrastructure (PKI), secure file transfer (SCP), and TFTP operations. The system adds the default route through the gateway of the admin VLAN.

Note Configure only one VLAN on the SSL Services Module as the admin VLAN.

Note VLAN IDs must be the same for the switch and the module. Refer to the "Configuring VLANs" chapter in the *Catalyst 6500 Series Switch Software Configuration Guide* for details.

Note The SSL software supports only the normal-range VLANs (2 through 1005). Limit the SSL Services Module configuration to the normal-range VLANs.

To configure VLANs on the SSL Services Module, perform this task:

	Command	Purpose
Step 1	ssl-proxy(config)# **ssl-proxy vlan** *vlan*	Configures the VLANs and enters VLAN mode.
Step 2	ssl-proxy(config-vlan)# **ipaddr** *ip_addr netmask*	Configures an IP address for the VLAN.
Step 3	ssl-proxy(config-vlan)# **gateway** *gateway_addr*	Configures the client-side gateway IP address. **Note** Configure the gateway IP address in the same subnet as the VLAN IP address.
Step 4	ssl-proxy(config-vlan)# **route** *ip_addr netmask* **gateway** *ip_addr*	(Optional) Configures a static route for servers that are one or more Layer 3 hops away from the SSL Services Module.
Step 5	ssl-proxy(config-vlan)# **admin**	(Optional) Configures the VLAN as the admin VLAN[1].

[1] The admin VLAN is for management traffic (PKI, SSH, SCP and TFTP). Specify only one VLAN as the admin VLAN.

This example shows how to configure the VLAN, specify the IP address, the subnet mask, and the global gateway, and also specifies the VLAN as the admin VLAN:

```
ssl-proxy(config)# ssl-proxy vlan 100
ssl-proxy(config-vlan)# ipaddr 10.1.0.20 255.255.255.0
ssl-proxy(config-vlan)# gateway 10.1.0.1
ssl-proxy(config-vlan)# admin
ssl-proxy(config-vlan)# ^Z
ssl-proxy#
```

Configuring Telnet Remote Access

To configure the SSL Services Module for Telnet remote access, perform this task:

	Command	Purpose
Step 1	`ssl-proxy(config)#` **`enable password`** *`password`*	Specifies a local enable password.
Step 2	`ssl-proxy(config)#` **`line vty`** *`starting-line-number ending-line-number`*	Identifies a range of lines for configuration and enters line configuration mode.
Step 3	`ssl-proxy(config-line)#` **`login`**	Enables password checking at login.
Step 4	`ssl-proxy(config-line)#` **`password`** *`password`*	Specifies a password on the line.

This example shows how to configure the SSL Services Module for remote access:

```
ssl-proxy(config)#line vty 0 4
ssl-proxy(config-line)#login
ssl-proxy(config-line)#password cisco
ssl-proxy(config-line)#end
ssl-proxy#
```

Configuring the Fully Qualified Domain Name

If you are using the SSL Services Module to enroll for certificates from a certificate authority, you must configure the Fully Qualified Domain Name (FQDN) on the module. The FQDN is the hostname and domain name of the module.

To configure the FQDN, perform this task:

	Command	Purpose
Step 1	`ssl-proxy(config)#` **`hostname`** *`name`*	Configures the hostname.
Step 2	`ssl-proxy(config)#` **`ip domain-name`** *`name`*	Configures the domain name.

This example shows how to configure the FQDN on the SSL Services Module:

```
ssl-proxy(config)# hostname ssl-proxy2
ssl-proxy2(config)# ip domain-name example.com
ssl-proxy2(config)# end
ssl-proxy2(config)#
```

Configuring SSH

After you complete the initial configuration for the module, enable SSH on the module, and then configure the user name and password for the SSH connection using either a simple user name and password or using an authentication, authorization, and accounting (AAA) server.

These sections describe how to enable and configure SSH:

- Enabling SSH on the Module, page 3-4
- Configuring the User Name and Password for SSH, page 3-5
- Configuring Authentication, Authorization, and Accounting for SSH, page 3-5

Enabling SSH on the Module

SSH uses the first key pair generated on the module. In the following task, you generate a key pair used specifically for SSH.

Note If you generate a general-purpose key pair (as described in the "Generating RSA Key Pairs" section on page 3-23) without specifying the SSH key pair first, SSH is enabled and uses the general-purpose key pair. If this key pair is later removed, SSH is disabled. To reenable SSH, generate a new SSH key pair.

To generate an SSH key pair and enable SSH, perform this task:

	Command	Purpose
Step 1	`ssl-proxy#` **`configure terminal`**	Enters configuration mode, selecting the terminal option.
Step 2	`ssl-proxy(config)#` **`ip ssh rsa keypair-name`** *`ssh_key_name`*	Assigns the key pair name to SSH.
Step 3	`ssl-proxy(config)#` **`crypto key generate rsa general-keys label`** *`ssh_key_name`*	Generates the SSH key pair. SSH is now enabled.
Step 4	`ssl-proxy(config)#` **`end`**	Exits configuration mode.
Step 5	`ssl-proxy#` **`show ip ssh`**	Shows the current state of SSH.

This example shows how to enable SSH on the module, and how to verify that SSH is enabled:

```
ssl-proxy(config)# ip ssh rsa keypair-name ssh-key
Please create RSA keys to enable SSH.
ssl-proxy(config)# crypto key generate rsa general-keys label ssh-key
The name for the keys will be: ssh-key
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys ...[OK]
```

```
ssl-proxy(config)#
*Aug 28 11:07:54.051: %SSH-5-ENABLED: SSH 1.5 has been enabled
ssl-proxy(config)# end

ssl-proxy# show ip ssh
SSH Enabled - version 1.5
Authentication timeout: 120 secs; Authentication retries: 3
ssl-proxy#
```

Configuring the User Name and Password for SSH

To configure the user name and password for the SSH connection, perform this task:

	Command	Purpose
Step 1	ssl-proxy# **configure terminal**	Enters configuration mode, selecting the terminal option.
Step 2	ssl-proxy(config)# **enable password** *password*	Specifies a local enable password, if not already specified.
Step 3	ssl-proxy(config)# **username** *username* {**password** | **secret**} *password*	Specifies the user name and password.
Step 4	ssl-proxy(config)# **line vty** *line-number ending-line-number*	Identifies a range of lines for configuration and enters line configuration mode.
Step 5	ssl-proxy(config-line)# **login local**	Enables local username authentication.

This example shows how to configure the user name and password for the SSH connection to the SSL Services Module:

```
ssl-proxy# configure terminal
ssl-proxy(config)# enable password cisco
ssl-proxy(config)# username admin password admin-pass
ssl-proxy(config)# line vty 0 4
ssl-proxy(config-line)# login local
ssl-proxy(config-line)# end
```

After you configure the user name and password, see the "Initial Catalyst 6500 Series Switch Configuration" section on page 3-6 to configure the switch.

Configuring Authentication, Authorization, and Accounting for SSH

To configure authentication, authorization, and accounting (AAA) for SSH, perform this task:

	Command	Purpose
Step 1	ssl-proxy# **configure terminal**	Enters configuration mode, selecting the terminal option.
Step 2	ssl-proxy(config)# **username** *username* **secret** {**0** | **5**} *password*	Enables enhanced password security for the specified, unretrievable username.

	Command	Purpose
Step 3	`ssl-proxy(config)# enable password password`	Specifies a local enable password, if not already specified.
Step 4	`ssl-proxy(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA).
Step 5	`ssl-proxy(config)# aaa authentication login default local`	Specifies the module to use the local username database for authentication.
Step 6	`ssl-proxy(config)# line vty line-number ending-line-number`	Identifies a range of lines for configuration and enters line configuration mode.
Step 7	`ssl-proxy(config-line)# transport input ssh`	Configures SSH as the only protocol used on a specific line (to prevent non-SSH connections).

This example shows how to configure AAA for the SSH connection to the SSL Services Module:

```
ssl-proxy# configure terminal
ssl-proxy(config)# username admin secret admin-pass
ssl-proxy(config)# enable password enable-pass
ssl-proxy(config)# aaa new-model
ssl-proxy(config)# aaa authentication login default local
ssl-proxy(config)# line vty 0 4
ssl-proxy(config-line)# transport input ssh
ssl-proxy(config-line)# end
ssl-proxy#
```

After you configure AAA, see the "Initial Catalyst 6500 Series Switch Configuration" section on page 3-6 to configure the switch.

Initial Catalyst 6500 Series Switch Configuration

How you configure the Catalyst 6500 series switch depends on whether you are using Cisco IOS software or the Catalyst operating system software.

The following sections describe how to configure the switch from the CLI for each switch operating system:

- Cisco IOS Software, page 3-6
- Catalyst Operating System Software, page 3-10

Cisco IOS Software

The initial Catalyst 6500 series switch configuration consists of the following:

- Configuring VLANs on the Switch, page 3-7
- Configuring Layer 3 Interfaces, page 3-7
- Configuring a LAN Port for Layer 2 Switching, page 3-8
- Adding the SSL Services Module to the Corresponding VLAN, page 3-8
- Verifying the Initial Configuration, page 3-9

Configuring VLANs on the Switch

Note VLAN IDs must be the same for the switch and the module. Refer to the "Configuring VLANs" chapter in the *Catalyst 6500 Series Switch Software Configuration Guide* for details.

Note The SSL software supports only the normal-range VLANs (2 through 1005). Limit the SSL Services Module configuration to the normal-range VLANs.

To configure VLANs on the switch, perform this task:

	Command	Purpose
Step 1	`Router#` **`configure terminal`**	Enters configuration mode, selecting the terminal option.
Step 2	`Router(config)#` **`vlan`** *`vlan_ID`*	Enters VLAN configuration mode and adds a VLAN. The valid range is 2 through 1001. **Note** Do not add an external VLAN.
Step 3	`Router(config-vlan)#` **`end`**	Updates the VLAN database and returns to privileged EXEC mode.

This example shows how to configure VLANs on the switch:

```
Router> enable
Router# configure terminal
Router(config)# vlan 100
VLAN 100 added:
     Name: VLAN100

Router(config-vlan)# end
```

Configuring Layer 3 Interfaces

To configure the corresponding Layer 3 VLAN interface, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface vlan`** *`vlan_ID`*	Selects an interface to configure.
Step 2	`Router(config-if)#` **`ip address`** *`ip_address subnet_mask`*	Configures the IP address and IP subnet.
Step 3	`Router(config-if)#` **`no shutdown`**	Enables the interface.
Step 4	`Router(config-if)#` **`exit`**	Exits configuration mode.

This example shows how to configure the Layer 3 VLAN interface:

```
Router# configure terminal
Router(config)# interface vlan 100
Router(config-if)# ip address 10.10.1.10 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
```

Configuring a LAN Port for Layer 2 Switching

To place physical interfaces that connect to the servers or the clients in the corresponding VLAN, perform this task:

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type`*[1] *`mod/port`*	Selects the LAN port to configure.
Step 2	`Router(config-if)#` **`switchport`**	Configures the LAN port for Layer 2 switching. **Note** You must enter the **switchport** command once without any keywords to configure the LAN port as a Layer 2 port before you can enter additional **switchport** commands with keywords.
Step 3	`Router(config-if)#` **`switchport mode access`**	Puts the LAN port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The LAN port becomes a nontrunk port even if the neighboring LAN port does not agree to the change.
Step 4	`Router(config-if)#` **`switchport access vlan`** *`vlan_ID`*	Configures the default VLAN, which is used if the interface stops trunking.
Step 5	`Router(config-if)#` **`no shutdown`**	Activates the interface.

[1] *type* = **ethernet**, **fastethernet**, **gigabitethernet**, or **tengigabitethernet**

This example shows how to configure a physical interface as a Layer 2 interface and assign it to a VLAN:

```
Router(config)# interface gigabitethernet 1/1
Router(config-if)# switchport
Router(config-if)# switchport mode access
Router(config-if)# switchport access vlan 100
Router(config-if)# no shutdown
Router(config-if)# exit
```

Adding the SSL Services Module to the Corresponding VLAN

Note By default, the SSL Services Module is in trunking mode with native VLAN 1.

To add the SSL Services Module to the corresponding VLAN, enter this command:

Command	Purpose
`Router (config)#` **`ssl-proxy module`** *`mod`* **`allowed-vlan`** *`vlan_ID`*	Configures the VLANs allowed over the trunk to the SSL Services Module. **Note** One of the allowed VLANs must be the admin VLAN.

This example shows how to add an SSL Services Module installed in slot 6 to a specific VLAN:

```
Router>
Router> enable
Router# configure terminal
Router (config)# ssl-proxy module 6 allowed-vlan 100
Router (config)# end
```

Verifying the Initial Configuration

To verify the configuration, enter these commands:

Command	Purpose
Router# **show spanning-tree vlan** *vlan_ID*	Displays the spanning tree state for the specified VLAN.
Router# **show ssl-proxy mod** *mod* **state**	Displays the trunk configuration.

Note In the following examples, the SSL Services Module is installed in slot 4 (Gi4/1).

This example shows how to verify that the module is in forwarding (FWD) state:

```
Router# show spanning-tree vlan 100

VLAN0100
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     0009.e9b2.b864
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     0009.e9b2.b864
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time 15

Interface        Role Sts Cost      Prio.Nbr Type
---------------- ---- --- --------- -------- --------------------------------
Gi3/1            Desg FWD 4         128.129  P2p
Gi4/1            Desg FWD 4         128.193  P2p
Po261            Desg FWD 3         128.833  P2p
Router
```

This example shows how to verify that the VLAN information displayed matches the VLAN configuration:

```
Router# show ssl-proxy mod 6 state
SSL-services module 6 data-port:
 Switchport:Enabled
Administrative Mode:trunk
Operational Mode:trunk
Administrative Trunking Encapsulation:dot1q
```

```
Operational Trunking Encapsulation:dot1q
Negotiation of Trunking:Off
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Trunking VLANs Enabled:100
Pruning VLANs Enabled:2-1001
Vlans allowed on trunk:100
Vlans allowed and active in management domain:100
Vlans in spanning tree forwarding state and not pruned:
100
Allowed-vlan :100
```

Catalyst Operating System Software

The initial Catalyst 6500 series switch configuration consists of the following:

- Configuring VLANs on the Switch, page 3-10
- Configuring Layer 3 Interfaces on the MSFC, page 3-11
- Adding the SSL Services Module to the Corresponding VLAN, page 3-11
- Verifying the Initial Configuration, page 3-12

Configuring VLANs on the Switch

Note VLAN IDs must be the same for the switch and the module. Refer to the “Configuring VLANs” chapter in the *Catalyst 6500 Switch Series Software Configuration Guide* for details.

Note The SSL software supports only the normal-range VLANs (2 through 1005). Limit the SSL Services Module configuration to the normal-range VLANs.

To configure VLANs on the switch, perform this task:

	Command	Purpose
Step 1	`Console>` **`enable`**	Enters privileged mode.
Step 2	`Console> (enable)` **`set vlan`** *`vlan_id`*	Adds a VLAN. The valid range is 2 through 1001. **Note** Do not add an external VLAN.

This example shows how to configure VLANs on the switch:

```
Console> enable
Enter Password: <password>
Console> (enable) set vlan 100
Vlan 100 configuration successful
Console> (enable)
```

Configuring Layer 3 Interfaces on the MSFC

To configure the corresponding Layer 3 VLAN interface on the multilayer switch feature card (MSFC), perform this task:

	Command	Purpose
Step 1	Console> (enable) **session** [*mod*][1]	Accesses the MSFC from the switch CLI using a Telnet session[2].
Step 2	Router> **enable**	Enters enable mode.
Step 3	Router# **configure terminal**	Enters global configuration mode.
Step 4	Router(config)# **interface vlan** *vlan_id*	Specifies a VLAN interface on the MSFC.
Step 5	Router(config-if)# **ip address** *ip_address subnet_mask*	Assigns an IP address to the VLAN.
Step 6	Router(config-if)# **no shutdown**	Enables the interface.
Step 7	Router(config-if)# **exit**	Exits the MSFC CLI and returns to the switch CLI.

[1] The *mod* keyword specifies the module number of the MSFC; either 15 (if the MSFC is installed on the supervisor engine in slot 1) or 16 (if the MSFC is installed on the supervisor engine in slot 2). If no module number is specified, the console will switch to the MSFC on the active supervisor engine.

[2] To access the MSFC from the switch CLI directly connected to the supervisor engine console port, enter the **switch console** *mod* command. To exit from the MSFC CLI and return to the switch CLI, press **Ctrl-C** three times at the Router> prompt.

This example shows how to configure the Layer 3 VLAN interface on the MSFC:

```
Console> (enable) session 15
Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
Router> config t
Router(config)# interface vlan 100
Router(config-if)# ip address 10.10.1.10 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# exit
Console> (enable)
```

Adding the SSL Services Module to the Corresponding VLAN

Note By default, the SSL Services Module is in trunking mode with native VLAN 1.

To add the SSL Services Module to the corresponding VLAN, enter this command:

Command	Purpose
Console> (enable) **set trunk** *mod/port vlan_id*	Configures the VLANs allowed over the trunk to the SSL Services Module. **Note** One of the allowed VLANs must be the admin VLAN.

This example shows how to add an SSL Services Module installed in slot 6 to a specific VLAN:

```
Console> (enable) set trunk 6/1 100
Adding vlans 100 to allowed list.
Console> (enable)
```

Verifying the Initial Configuration

To verify the configuration, enter one of these commands:

Command	Purpose
Console> **show spanntree** *vlan_ID*	Displays the spanning tree state for the specified VLAN.
Console> **show trunk** *mod/port*	Displays the trunk configuration.

Note In the following examples, the SSL Services Module is installed in slot 6.

This example shows how to verify that the module is in forwarding (FWD) state:

```
Console> show spantree 100
VLAN 100
Spanning tree mode          PVST+
Spanning tree type          ieee
Spanning tree enabled

Designated Root             00-06-2a-db-a5-01
Designated Root Priority    32768
Designated Root Cost        0
Designated Root Port        1/0
Root Max Age   20 sec   Hello Time 2  sec   Forward Delay 15 sec

Bridge ID MAC ADDR          00-06-2a-db-a5-01
Bridge ID Priority          32768
Bridge Max Age 20 sec   Hello Time 2  sec   Forward Delay 15 sec

Port                     Vlan Port-State    Cost      Prio Portfast Channel_id
------------------------ ---- ------------- --------- ---- -------- ----------
 6/1                     100  forwarding          100   32 enabled  033
Console>
```

This example shows how to verify that the VLAN information displayed matches the VLAN configuration:

```
Console> show trunk 6/1
* - indicates vtp domain mismatch
# - indicates dot1q-all-tagged enabled on the port
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 6/1      nonegotiate  dot1q          trunking      1

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 6/1      100

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 6/1      100

Port      Vlans in spanning tree forwarding state and not pruned
--------  ---------------------------------------------------------------------
 6/1      100
```

Upgrading the Images

The compact Flash on the SSL Services Module has two bootable partitions: application partition (AP) and maintenance partition (MP). By default the application partition boots every time. The application partition contains the binaries necessary to run the SSL image. The maintenance partition is booted if you need to upgrade the application partition.

You can upgrade both the application software and the maintenance software.

The entire application and maintenance partitions are stored on the FTP or TFTP server. The images are downloaded and extracted to the application partition or maintenance partition depending on which image is being upgraded.

To upgrade the application partition, change the boot sequence to boot the module from the maintenance partition. To upgrade the maintenance partition, change the boot sequence to boot the module from the application partition. Set the boot sequence for the module using the supervisor engine CLI commands. The maintenance partition downloads and installs the application image. The supervisor engine must be executing the run-time image to provide network access to the maintenance partition.

Before starting the upgrade process, you will need these software images:

- The application partition image for the module
- The maintenance partition image for the module

A TFTP and FTP server are required to copy the images. The TFTP server should be connected to the switch, and the port connecting to the TFTP server should be included in VLAN 1 on the switch.

Another TFTP server is required in the network. This TFTP server must be reachable from the module when the module image is booted up.

These sections describe how to upgrade the images:

- Upgrading the Application Software, page 3-13.
- Upgrading the Maintenance Software, page 3-17.

Upgrading the Application Software

How you upgrade the application software depends on whether you are using Cisco IOS software or the Catalyst operating system software.

The following sections describe how to upgrade the application software from the CLI for each switch operating system:

- Cisco IOS Software, page 3-14
- Catalyst Operating System Software, page 3-16

Cisco IOS Software

Note Do not reset the module until the image is upgraded. The total time to upgrade the image takes up to 8 minutes.

To upgrade the application partition software, perform this task:

	Command	Purpose
Step 1	`Router# hw-module module mod reset cf:1`	Reboots the module from the maintenance partition. **Note** It is normal to see messages, such as "Press Key," on the module console after entering this command.
Step 2	`Router# show module`	Displays that the maintenance partition for the module has booted.
Step 3	`Router# copy tftp: pclc#mod-fs:`	Downloads the image.
Step 4	`Router# hw-module module mod reset`	Resets the module.
Step 5	`Router# show module`	Displays that the application partition for the module has booted.

This example shows how to upgrade the application partition software:

```
Router# hw-module module 6 reset cf:1
hw mod 6 reset cf:1
Device BOOT variable for reset = <cf:1>
Warning: Device list is not verified.

Proceed with reload of module? [confirm]y

% reset issued for module 6

02:11:18: SP: The PC in slot 6 is shutting down. Please wait ...
02:11:31: SP: PC shutdown completed for module 6
02:11:31: %C6KPWR-SP-4-DISABLED: power to module in slot 6 set off (Reset)
02:14:21: SP: OS_BOOT_STATUS(6) MP OS Boot Status: finished booting
02:14:28: %DIAG-SP-6-RUN_MINIMUM: Module 6: Running Minimum Online Diagnostics...
02:14:34: %DIAG-SP-6-DIAG_OK: Module 6: Passed Online Diagnostics
02:14:34: %OIR-SP-6-INSCARD: Card inserted in slot 6, interfaces are now online

Router# show module
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  1    2  Catalyst 6000 supervisor 2 (Active)    WS-X6K-S2U-MSFC2   SAD055006RZ
  2   48  48 port 10/100 mb RJ45                 WS-X6348-RJ-45     SAL052794UW
  6    1  SSL Module (MP)                        WS-SVC-SSL-1       SAD060702VK

...<output truncated>...
```

```
Router# copy tftp: pclc#6-fs:
copy tftp: pclc#6-fs:
Address or name of remote host []? 10.1.1.1

Source filename []? c6svc-ssl-k9y9.1-x-y.bin

Destination filename [c6svc-ssl-k9y9.1-x-y.bin]?

Accessing tftp://10.1.1.1/c6svc-ssl-k9y9.1-x-y.bin...
Loading c6svc-ssl-k9y9.1-x-y.bin from 10.1.1.1 (via Vlan2):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

<output truncated>

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 14918353 bytes]

14918353 bytes copied in 643.232 secs (23193 bytes/sec)
Router#
02:29:23: %SVCLC-SP-5-STRRECVD: mod 6: <Application upgrade has started>
02:29:23: %SVCLC-SP-5-STRRECVD: mod 6: <Do not reset the module till upgrade completes!!>
02:36:07: %SVCLC-SP-5-STRRECVD: mod 6: <Application upgrade has succeded>
02:36:07: %SVCLC-SP-5-STRRECVD: mod 6: <You can now reset the module>>

Router# hw-module module 6 reset
Device BOOT variable for reset = <empty>
Warning:Device list is not verified.

Proceed with reload of module? [confirm]y
% reset issued for module 6
Router#
02:36:57:SP:The PC in slot 6 is shutting down. Please wait ...
02:37:17:SP:PC shutdown completed for module 6
02:37:17:%C6KPWR-SP-4-DISABLED:power to module in slot 6 set off (Reset)
02:38:39:SP:OS_BOOT_STATUS(6) AP OS Boot Status:finished booting
02:39:27:%DIAG-SP-6-RUN_COMPLETE:Module 6:Running Complete Online Diagnostics...
02:39:29:%DIAG-SP-6-DIAG_OK:Module 6:Passed Online Diagnostics
02:39:29:%OIR-SP-6-INSCARD:Card inserted in slot 6, interfaces are now online

Router# show module

Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
  1    2  Catalyst 6000 supervisor 2 (Active)    WS-X6K-S2U-MSFC2   SAD055006RZ
  2   48  48 port 10/100 mb RJ45                 WS-X6348-RJ-45     SAL052794UW
  6    1  SSL Module                             WS-SVC-SSL-1       SAD060702VK

...<output truncated>...
```

Catalyst Operating System Software

Note Do not reset the module until the image is upgraded. The total time to upgrade the image takes up to 8 minutes.

To upgrade the application partition software, perform this task:

	Command	Purpose
Step 1	Console (enable) **set boot device cf:1** *mod*	Sets the module to boot the maintenance partition.
Step 2	Console (enable) **reset** *mod*	Resets the module to the maintenance partition. **Note** The SUP_OSBOOTSTATUS system message shows that the maintenance partition (MP) has booted.
Step 3	Console (enable) **session** [*mod*]	Access the MSFC from the switch CLI using a Telnet session[1].
Step 4	Router# **copy tftp: pclc#***mod***-fs:**	Downloads the image.
Step 5	Router# **exit**	Exits the MSFC CLI and returns to the switch CLI.
Step 6	Console (enable) **set boot device cf:4** *mod*	Sets the module to boot the application partition.
Step 7	Console (enable) **reset** *mod*	Resets the module to the application partition. **Note** The SUP_OSBOOTSTATUS system message shows that the application partition (AP) has booted.

[1] To access the MSFC from the switch CLI directly connected to the supervisor engine console port, enter the **switch console** *mod* command. To exit from the MSFC CLI and return to the switch CLI, press **Ctrl-C** three times at the Router> prompt.

This example shows how to upgrade the application partition software:

```
Console> (enable) set boot device cf:1 6
Device BOOT variable = cf:1
Memory-test set to PARTIAL
Warning:Device list is not verified but still set in the boot string.
Console> (enable)
Console> (enable) reset 6 cf:1
This command will reset module 6.
Unsaved configuration on module 6 will be lost
Do you want to continue (y/n) [n]? y
Module 6 shut down in progress, please don't remove module until shutdown completed.
Console> (enable) Module 6 shutdown completed. Module resetting...
2003 Jan 17 08:34:07 %SYS-3-SUP_OSBOOTSTATUS:MP OS Boot Status:finished booting
2003 Jan 17 08:34:23 %SYS-5-MOD_OK:Module 6 is online
2003 Jan 17 08:34:23 %DTP-5-TRUNKPORTON:Port 6/1 has become dot1q trunk
```

```
Console> (enable) session 15
Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
Router>

Router# copy tftp: pclc#6-fs:
copy tftp: pclc#6-fs:
Address or name of remote host []? 10.1.1.1

Source filename []? c6svc-ssl-k9y9.1-x-y.bin

Destination filename [c6svc-ssl-k9y9.1-x-y.bin]?

Accessing tftp://10.1.1.1/c6svc-ssl-k9y9.1-x-y.bin...
Loading c6svc-ssl-k9y9.1-x-y.bin from 10.1.1.1 (via Vlan2):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

<output truncated>

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 14918353 bytes]

14918353 bytes copied in 643.232 secs (23193 bytes/sec)
Router#
02:29:23: %SVCLC-SP-5-STRRECVD: mod 6: <Application upgrade has started>
02:29:23: %SVCLC-SP-5-STRRECVD: mod 6: <Do not reset the module till upgrade completes!!>
02:36:07: %SVCLC-SP-5-STRRECVD: mod 6: <Application upgrade has succeded>
02:36:07: %SVCLC-SP-5-STRRECVD: mod 6: <You can now reset the module>>
Router# exit
Console> (enable) set boot device cf:4 6
Device BOOT variable = cf:4
Memory-test set to PARTIAL
Warning:Device list is not verified but still set in the boot string.
Console> (enable) reset 6
This command will reset module 6.
Unsaved configuration on module 6 will be lost
Do you want to continue (y/n) [n]? y
Module 6 shut down in progress, please don't remove module until shutdown completed.
Console> (enable) Module 6 shutdown completed. Module resetting...
2003 Jan 17 08:36:58 %SYS-3-SUP_OSBOOTSTATUS:AP OS Boot Status:finished booting
2003 Jan 17 08:37:51 %SYS-5-MOD_OK:Module 6 is online
2003 Jan 17 08:37:51 %DTP-5-TRUNKPORTON:Port 6/1 has become dot1q trunk
```

Upgrading the Maintenance Software

How you upgrade the maintenance software depends on whether you are using Cisco IOS software or the Catalyst operating system software.

The following sections describe how to upgrade the maintenance software from the CLI for each switch operating system:

- Cisco IOS Software, page 3-18
- Catalyst OS Software, page 3-19

Cisco IOS Software

Note Do not reset the module until the image is upgraded. The total time to upgrade the image takes up to 8 minutes.

To upgrade the maintenance partition software, perform this task:

	Command	Purpose
Step 1	`Router# hw-module module mod reset`	Reboots the module from the application partition.
Step 2	`Router# copy tftp: pclc#mod-fs:`	Downloads the image.
Step 3	`Router# hw-module module mod reset cf:1`	Resets the module in the maintenance partition.
Step 4	`Router# show module`	Displays that the maintenance partition for the module has booted.

This example shows how to upgrade the maintenance partition software:

```
Router# hw module 6 reset
Device BOOT variable for reset = <empty>
Warning:Device list is not verified.
Proceed with reload of module? [confirm]y
% reset issued for module 6
Router#
02:36:57:SP:The PC in slot 6 is shutting down. Please wait ...
02:37:17:SP:PC shutdown completed for module 6
02:37:17:%C6KPWR-SP-4-DISABLED:power to module in slot 6 set off (Reset)
1w0d:SP:OS_BOOT_STATUS(6) AP OS Boot Status:finished booting
1w0d:%OIR-SP-6-INSCARD:Card inserted in slot 6, interfaces are now online
Router# copy tftp:pclc#6-fs:
Address or name of remote host []? 10.1.1.1
Source filename []? mp.1-2-0-16.bin.gz
Destination filename [mp.1-2-0-16.bin.gz]?
Accessing tftp://10.1.1.1/mp.1-2-0-16.bin.gz...
Loading mp.1-2-0-16.bin.gz from 10.1.1.1 (via Vlan2):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
<output truncated>
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 9818951 bytes]
9818951 bytes copied in 164.388 secs (59730 bytes/sec)
ssl-proxy>
1w0d:%SVCLC-SP-6-STRRECVD:mod 6:<MP upgrade started. Do not reset the card.>
1w0d:%SVCLC-SP-6-STRRECVD:mod 6:<Upgrade of MP was successful. You can now boot MP.>
Router# hw mod 6 reset cf:1
Device BOOT variable for reset = <cf:1>
Warning:Device list is not verified.
Proceed with reload of module? [confirm]y
% reset issued for module 6
Router# show module
Mod Ports Card Type                              Model              Serial No.
--- ----- -------------------------------------- ------------------ -----------
1   2     Catalyst 6000 supervisor 2 (Active)    WS-X6K-S2U-MSFC2   SAD055006RZ
2   48    48 port 10/100 mb RJ45                 WS-X6348-RJ-45     SAL052794UW
6   1     SSL Module (MP)                       WS-SVC-SSL-1        SAD060702VK

...<output truncated>...
```

Catalyst OS Software

Note Do not reset the module until the image is upgraded. The total time to upgrade the image takes up to 8 minutes.

To upgrade the maintenance partition software, perform this task:

	Command	Purpose
Step 1	`Console (enable)` **`set boot device cf:4`** *`mod`*	Sets the module to boot the application partition.
Step 2	`Console (enable)` **`reset`** *`mod`*	Resets the module to the application partition. **Note** The SUP_OSBOOTSTATUS system message shows that the application partition (AP) has booted.
Step 3	`Console (enable)` **`session`** `[`*`mod`*`]`	Access the MSFC from the switch CLI using a Telnet session[1].
Step 4	`Router#` **`copy tftp: pclc#`***`mod`***`-fs:`**	Downloads the image.
Step 5	`Router#` **`exit`**	Exits the MSFC CLI and returns to the switch CLI.
Step 6	`Console (enable)` **`set boot device cf:1`** *`mod`*	Sets the module to boot the maintenance partition.
Step 7	`Console (enable)` **`reset`** *`mod`*	Resets the module to the maintenance partition. **Note** The SUP_OSBOOTSTATUS system message shows that the maintenance partition (MP) has booted.

[1] To access the MSFC from the switch CLI directly connected to the supervisor engine console port, enter the **switch console** *mod* command. To exit from the MSFC CLI and return to the switch CLI, press **Ctrl-C** three times at the Router> prompt.

This example shows how to upgrade the maintenance partition software:

```
Console> (enable) set boot device cf:4 6
Device BOOT variable = cf:4
Memory-test set to PARTIAL
Warning:Device list is not verified but still set in the boot string.
Console> (enable) reset 6
This command will reset module 6.
Unsaved configuration on module 6 will be lost
Do you want to continue (y/n) [n]? y
Module 6 shut down in progress, please don't remove module until shutdown completed.
Console> (enable) Module 6 shutdown completed. Module resetting...
2003 Jan 17 08:36:58 %SYS-3-SUP_OSBOOTSTATUS:AP OS Boot Status:finished booting
2003 Jan 17 08:37:51 %SYS-5-MOD_OK:Module 6 is online
2003 Jan 17 08:37:51 %DTP-5-TRUNKPORTON:Port 6/1 has become dot1q trunk
Console> (enable) session 15
Trying Router-15...
Connected to Router-15.
Type ^C^C^C to switch back...
Router>
```

```
Router# copy tftp:pclc#6-fs:
Address or name of remote host []? 10.1.1.1
Source filename []? mp.1-2-0-16.bin.gz
Destination filename [mp.1-2-0-16.bin.gz]?
Accessing tftp://10.1.1.1/mp.1-2-0-16.bin.gz...
Loading mp.1-2-0-16.bin.gz from 10.1.1.1 (via Vlan2):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

<output truncated>

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 9818951 bytes]

9818951 bytes copied in 164.388 secs (59730 bytes/sec)
ssl-proxy>
1w0d:%SVCLC-SP-6-STRRECVD:mod 6:<MP upgrade started. Do not reset the card.>
1w0d:%SVCLC-SP-6-STRRECVD:mod 6:<Upgrade of MP was successful. You can now boot MP.>
Router# exit
Console> (enable) set boot device cf:1 6
Device BOOT variable = cf:1
Memory-test set to PARTIAL
Warning:Device list is not verified but still set in the boot string.
Console> (enable)
Console> (enable) reset 6 cf:1
This command will reset module 6.
Unsaved configuration on module 6 will be lost
Do you want to continue (y/n) [n]? y
Module 6 shut down in progress, please don't remove module until shutdown completed.
Console> (enable) Module 6 shutdown completed. Module resetting...
2003 Jan 17 08:34:07 %SYS-3-SUP_OSBOOTSTATUS:MP OS Boot Status:finished booting
2003 Jan 17 08:34:23 %SYS-5-MOD_OK:Module 6 is online
2003 Jan 17 08:34:23 %DTP-5-TRUNKPORTON:Port 6/1 has become dot1q trunk
```

Configuring the SSL Services Module

These sections describe how to configure the SSL Services Module:

- Configuring Public Key Infrastructure, page 3-20
- Configuring SSL Proxy Services, page 3-44

Configuring Public Key Infrastructure

The SSL Services Module uses the SSL protocol to enable secure transactions of data through privacy, authentication, and data integrity; the protocol relies upon certificates, public keys, and private keys.

The certificates, which are similar to digital ID cards, verify the identity of the server to the clients. The certificates, which are issued by certificate authorities, include the name of the entity to which the certificate was issued, the entity’s public key, and the time stamps that indicate the certificate’s expiration date.

Public and private keys are the ciphers that are used to encrypt and decrypt information. The public key is shared without any restrictions, but the private key is never shared. Each public-private key pair works together; data that is encrypted with the public key can only be decrypted with the corresponding private key.

Each SSL Services Module acts as an SSL proxy for up to 256 web servers. You must configure a pair of keys for each web server in order to apply for a server certificate for authentication.

We recommend that the certificates be stored in NVRAM so that when you boot up, the module does not need to query the certificate authority to obtain the certificates or to automatically enroll. See the "Saving Your Configuration" section on page 3-37 for more information.

These sections describe how to configure the public key infrastructure (PKI):

- Configuring Keys and Certificates, page 3-21
- Verifying Certificates and Trustpoints, page 3-36
- Saving Your Configuration, page 3-37
- Backing Up Keys and Certificates, page 3-38
- Monitoring and Maintaining Keys and Certificates, page 3-39
- Assigning a Certificate to a Proxy Service, page 3-40
- Renewing a Certificate, page 3-41
- Enabling Key and Certificate History, page 3-44

Configuring Keys and Certificates

You can configure keys and certificates using one of the following methods:

- If you are using Simple Certificate Enrollment Protocol (SCEP), configure the keys and certificates by doing the following:
  - Generate a key pair.
  - Declare the trustpoint.
  - Get the certificate authority certificate.
  - Send an enrollment request to a certificate authority on behalf of the SSL server.

  See the "Configuring the Trustpoint Using SCEP" section on page 3-23 for details.
- If you are not using SCEP, configure the keys and certificates using the manual certificate enrollment (TFTP and cut-and-paste) feature by doing the following:
  - Generate or import a key pair.
  - Declare the trustpoint.
  - Get the certificate authority certificate and enroll the trustpoint using TFTP or cut-and-paste to create a PKCS10 file.
  - Request the SSL server certificate offline using the PKCS10 package.
  - Import the SSL server certificate using TFTP or cut-and-paste.

  See the "Manual Certificate Enrollment" section on page 3-28 for details.
- If you are using an external PKI system, do the following:
  - Generate PKCS12 or PEM files.
  - Import this file to the module.

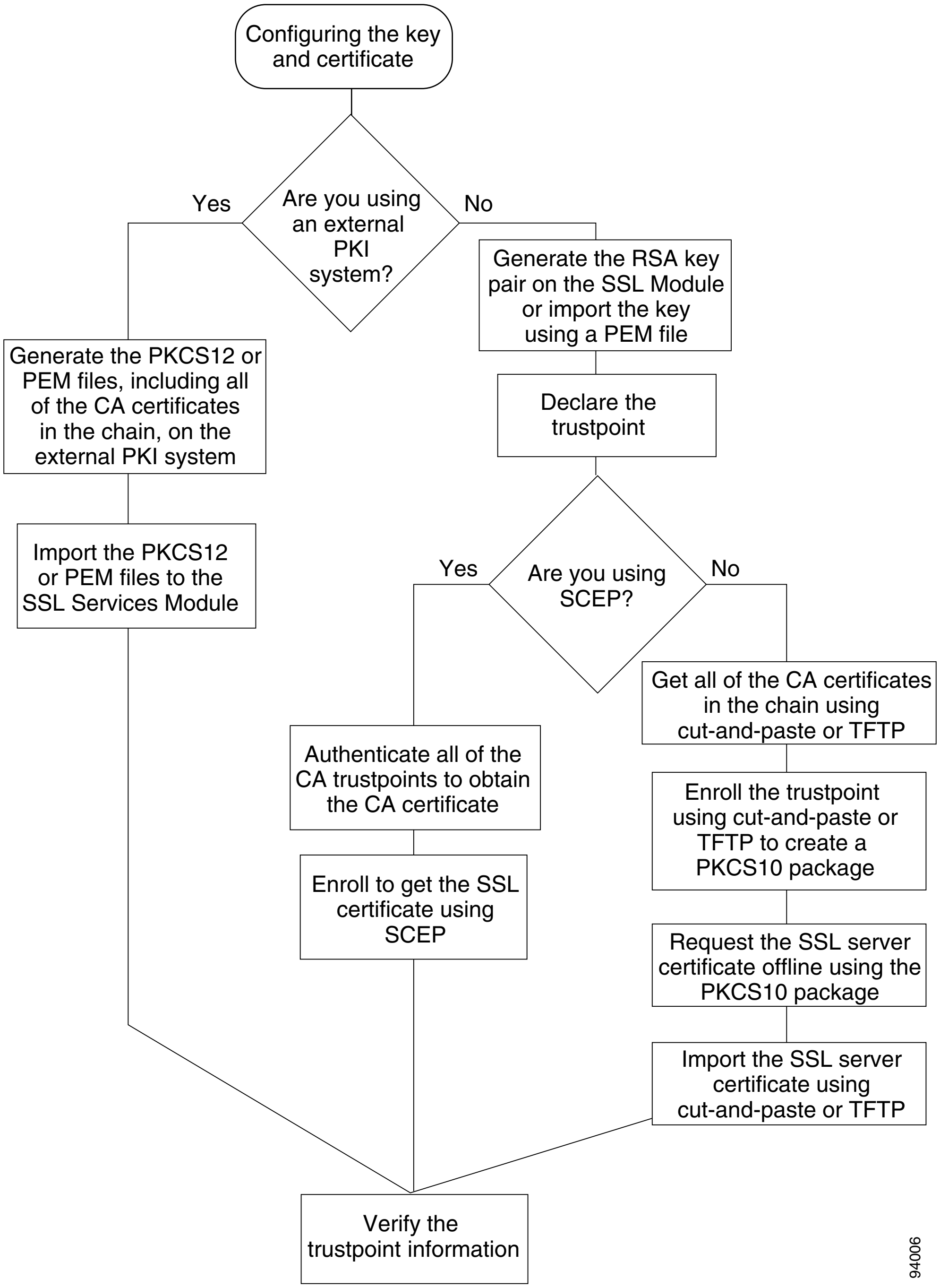
  See the "Importing and Exporting Key Pairs and Certificates" section on page 3-29 for details.

An external PKI system is a server or a PKI administration system that generates key pairs and enrolls for certificates from a certificate authority or a key and certificate archival system. The Public-Key Cryptography Standards (PKCS) specifies the transfer syntax for personal identity information, including the private keys and certificates. This information is packaged into an encrypted file. To open the encrypted file, you must know a pass phrase. The encryption key is derived from the pass phrase.

Note You do not need to configure a trustpoint before importing the PKCS12 or PEM files. If you import keys and certificates from PKCS12 or PEM files, the trustpoint is created automatically, if it does not already exist.

See Figure 3-1 for an overview on configuring keys and certificates.

Figure 3-1 Key and Certificate Configuration Overview


Configuring the key and certificate
Are you using an external PKI system?
Yes
No
Generate the PKCS12 or PEM files, including all of the CA certificates in the chain, on the external PKI system
Import the PKCS12 or PEM files to the SSL Services Module
Generate the RSA key pair on the SSL Module or import the key using a PEM file
Declare the trustpoint
Are you using SCEP?
Yes
No
Authenticate all of the CA trustpoints to obtain the CA certificate
Enroll to get the SSL certificate using SCEP
Get all of the CA certificates in the chain using cut-and-paste or TFTP
Enroll the trustpoint using cut-and-paste or TFTP to create a PKCS10 package
Request the SSL server certificate offline using the PKCS10 package
Import the SSL server certificate using cut-and-paste or TFTP
Verify the trustpoint information
94006

Configuring the Trustpoint Using SCEP

To configure a trustpoint using SCEP, complete the following tasks:

- Generating RSA Key Pairs, page 3-23
- Declaring the Trustpoint, page 3-25
- Obtaining the Certificate Authority Certificate, page 3-26
- Requesting a Certificate, page 3-26

Generating RSA Key Pairs

Note The first key pair generated enables SSH on the module. If you are using SSH, configure a key pair for SSH. See the "Configuring SSH" section on page 3-4.

RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Aldeman. RSA algorithm is widely used by certificate authorities and SSL servers to generate key pairs. Each certificate authority and each SSL server has its own RSA key pair. The SSL server sends its public key to the certificate authority when enrolling for a certificate. The SSL server uses the certificate to prove its identity to clients when setting up the SSL session.

The SSL server keeps the private key in a secure storage, and sends only the public key to the certificate authority, which uses its private key to sign the certificate that contains the server's public key and other identifying information about the server.

Each certificate authority keeps the private key secret and uses the private key to sign certificates for its subordinate certificate authorities and SSL servers. The certificate authority has a certificate that contains its public key.

The certificate authorities form a hierarchy of one or more levels. The top-level certificate authority is called the root certificate authority. The lower level certificate authorities are called Intermediate or subordinate certificate authorities. The root certificate authority has a self-signed certificate, and it signs the certificate for the next level subordinate certificate authority, which in turn signs the certificate for the next lower level certificate authority, and so on. The lowest level certificate authority signs the certificate for the SSL server.

Note In SSL software release 1.2, the SSL Services Module supports up to eight levels of certificate authority (one root certificate authority and up to seven subordinate certificate authorities). For an example of a three-level (3-tier) enrollment, see the "Example of Three-Tier Certificate Authority Enrollment" section on page 3-27.

These certificates form a chain with the server certificate at the bottom and the root certificate authority's self-signed certificate at the top. Each signature is formed by using the private key of the issuing certificate authority to encrypt a hash digest of the certificate body. The signature is attached to the end of the certificate body to form the complete certificate.

When setting up an SSL session, the SSL server sends its certificate chain to the client. The client verifies the signature of each certificate up the chain by retrieving the public key from the next higher-level certificate to decrypt the signature attached to the certificate body. The decryption result is compared with the hash digest of the certificate body. Verification terminates when one of the certificate authority certificates in the chain matches one of the trusted certificate authority certificates stored in the client's own database.

If the top-level certificate authority certificate is reached in the chain, and there is no match of trusted self-signed certificates, the client may terminate the session, or prompt the user to view the certificates and determine if they can be trusted.

After the SSL client authenticates the server, it uses the public key from the server certificate to encrypt a secret and send it over to the server. The SSL server uses its private key to decrypt the secret. Both sides use the secret and two random numbers they exchanged to generate the key material required for the rest of the SSL session for data encryption, decryption, and integrity checking.

Note The SSL Services Module supports only general-purpose keys.

When you generate general-purpose keys, only one pair of RSA keys is generated. Named key pairs allow you to have multiple RSA key pairs, enabling the Cisco IOS software to maintain a different key pair for each identity certificate. We recommend that you specify a name for the key pairs.

Note The generated key pair resides in system memory (RAM). They will be lost on power failure or module reset. You must enter the **copy system:running-config nvram:startup-config** command to save the running configuration, as well as save the key pairs to the private configuration file in the module NVRAM.

To generate RSA key pairs, perform this task:

Command	Purpose
`ssl-proxy(config)#` **`crypto key generate rsa general-keys label`** *`key-label`* `[`**`exportable`**`[1]] [`**`modulus`** *`size`*`]`	Generates RSA key pairs.

[1] The **exportable** keyword specifies that the key is allowed to be exported. You can specify that a key is exportable during key generation. Once the key is generated as either exportable or not exportable, it cannot be modified for the life of the key.

Note When you generate RSA keys, you are prompted to enter a modulus length in bits. The SSL Services Module supports modulus lengths of 512, 768, 1024, 1536, and 2048 bits. Although you can specify 512 or 768, we recommend a minimum modulus length of 1024. A longer modulus takes longer to generate and takes longer to use, but offers stronger security.

This example shows how to generate general-purpose RSA keys:

```
ssl-proxy(config)# crypto key generate rsa general-keys label kp1 exportable

The name for the keys will be: kp1

Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 1024

Generating RSA keys.... [OK].
```

Note After you generate a key pair, you can test the SSL service by generating a self-signed certificate. To generate a self-signed certificate for testing, see the "Generating a Self-Signed Certificate" section on page A-1.

Declaring the Trustpoint

You should declare one trustpoint to be used by the module for each certificate.

To declare the trustpoint that your module uses and specify characteristics for the trustpoint, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`ssl-proxy(config)#` **`crypto ca trustpoint`** *`trustpoint-label`*[1]	Declares the trustpoint that your module should use. Enabling this command puts you in ca-trustpoint configuration mode.
Step 2	`ssl-proxy(ca-trustpoint)#` **`rsakeypair`** *`key-label`*	Specifies which key pair to associate with the certificate.
Step 3	`ssl-proxy(ca-trustpoint)#` **`enrollment`** `[`**`mode ra`**`] [`**`retry`** `[`**`period`** *`minutes`*`] [`**`count`** *`count`*`]]` **`url`** *`url`*	Specifies the enrollment parameters for your certificate authority.
Step 4	`ssl-proxy(ca-trustpoint)#` **`ip-address`** *`server_ip_addr`*	(Optional) Specifies the IP address of the proxy service which will use this certificate[2].
Step 5	`ssl-proxy(ca-trustpoint)#` **`subject-name`** *`line`*[3]	(Optional) Configures the host name of the proxy service[4].
Step 6	`ssl-proxy(ca-trustpoint)#` **`password`** *`password`*	(Optional) Configures a challenge password.
Step 7	`ssl-proxy(ca-trustpoint)#` **`exit`**	Exits ca-trustpoint configuration mode.

[1] The *trustpoint-label* should match the *key-label* of the keys; however, this is not a requirement.

[2] Some web browsers compare the IP address in the SSL server certificate with the IP address that might appear in the URL. If the IP addresses do not match, the browser may display a dialog box and ask the client to accept or reject this certificate.

[3] For example, **subject-name CN=***server1.domain2.com*, where *server1* is the name of the SSL server that appears in the URL. The **subject-name** command uses the Lightweight Directory Access Protocol (LDAP) format.

[4] Some browsers compare the CN field of the subject name in the SSL server certificate with the hostname that might appear in the URL. If the names do not match, the browser may display a dialog box and ask the client to accept or reject the certificate. Also, some browsers will reject the SSL session setup and silently close the session if the CN field is not defined in the certificate.

This example shows how to declare the trustpoint PROXY1 and verify connectivity:

```
ssl-proxy(config)# crypto ca trustpoint PROXY1
ssl-proxy(ca-trustpoint)# rsakeypair PROXY1
ssl-proxy(ca-trustpoint)# enrollment url http://exampleCA.cisco.com
ssl-proxy(ca-trustpoint)# ip-address 10.0.0.1
ssl-proxy(ca-trustpoint)# password password
ssl-proxy(ca-trustpoint)# serial-number
ssl-proxy(ca-trustpoint)# subject-name C=US; ST=California; L=San Jose; O=Cisco; OU=Lab;
CN=host1.cisco.com
ssl-proxy(ca-trustpoint)# end
ssl-proxy#
ssl-proxy# ping example.cisco.com
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms
ssl-proxy#
```

Obtaining the Certificate Authority Certificate

For each trustpoint, you must get a certificate that contains the public key of the certificate authority; multiple trustpoints can use the same certificate authority.

Note Contact the certificate authority to obtain the correct fingerprint of the certificate and verify the fingerprint displayed on the console.

To obtain the certificate that contains the public key of the certificate authority, perform this task in global configuration mode:

Command	Purpose
`ssl-proxy(config)# crypto ca authenticate` *`trustpoint-label`*	Obtains the certificate that contains the public key of the certificate authority. Enter the same *trustpoint_label* that you entered when declaring the trustpoint.

This example shows how to obtain the certificate of the certificate authority:

```
ssl-proxy(config)# crypto ca authenticate PROXY1
Certificate has the following attributes:
Fingerprint: A8D09689 74FB6587 02BFE0DC 2200B38A
% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
ssl-proxy(config)# end
ssl-proxy#
```

Requesting a Certificate

You must obtain a signed certificate from the certificate authority for each trustpoint.

To request signed certificates from the certificate authority, perform this task in global configuration mode:

Command	Purpose
`ssl-proxy(config)# crypto ca enroll` *`trustpoint-label`*[1]	Requests a certificate for the trustpoint.

[1] You have the option to create a challenge password that is not saved with the configuration. This password is required in the event that your certificate needs to be revoked, so you must remember this password.

Note If your module or switch reboots after you have entered the **crypto ca enroll** command but before you have received the certificates, you must reenter the command and notify the certificate authority administrator.

This example shows how to request a certificate:

```
ssl-proxy(config)# crypto ca enroll PROXY1
%
% Start certificate enrollment ..

% The subject name in the certificate will be: C=US; ST=California; L=San Jose; O=Cisco;
OU=Lab; CN=host1.cisco.com
% The subject name in the certificate will be: host.cisco.com
% The serial number in the certificate will be: 00000000
% The IP address in the certificate is 10.0.0.1

% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.
Fingerprint:  470DE382 65D8156B 0F84C2AF 4538B913

ssl-proxy(config)#end
```

After you configure the trustpoint, see the "Verifying Certificates and Trustpoints" section on page 3-36 to verify the certificate and trustpoint information.

Example of Three-Tier Certificate Authority Enrollment

In SSL software release 1.2, the SSL Services Module supports up to eight levels of certificate authority (one root certificate authority and up to seven subordinate certificate authorities). The following example shows how to configure three levels of certificate authority:

Generating the Keys

```
ssl-proxy(onfig)# crypto key generate rsa general-keys label key1 exportable
The name for the keys will be:key1
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]:1024
% Generating 1024 bit RSA keys ...[OK]
```

Defining the Trustpoints

```
ssl-proxy(config)# crypto ca trustpoint 3tier-root
ssl-proxy(ca-trustpoint)# enrollment url http://10.1.1.1
ssl-proxy(ca-trustpoint)#
ssl-proxy(ca-trustpoint)# exit
ssl-proxy(config)# crypto ca trustpoint 3tier-sub1
ssl-proxy(ca-trustpoint)# enrollment url http://10.1.1.2
ssl-proxy(ca-trustpoint)#
ssl-proxy(ca-trustpoint)# exit
ssl-proxy(config)# crypto ca trustpoint tp-proxy1
ssl-proxy(ca-trustpoint)# enrollment url http://10.1.1.3
ssl-proxy(ca-trustpoint)# serial-number
ssl-proxy(ca-trustpoint)# password cisco
ssl-proxy(ca-trustpoint)# subject CN=ste.cisco.com
ssl-proxy(ca-trustpoint)# rsakeypair key1
ssl-proxy(ca-trustpoint)#
ssl-proxy(ste(ca-trustpoint)#show
 enrollment url http://10.1.1.3
 serial-number
 password 7 02050D480809
 subject-name CN=ste.cisco.com
 rsakeypair key1
end

ssl-proxy(ca-trustpoint)# exit
```

Authenticating the Three Certificate Authorities (One Root And Two Subordinate Certificate Authorities)

```
ssl-proxy(config)# crypto ca authenticate 3tier-root
Certificate has the following attributes:
Fingerprint:84E470A2 38176CB1 AA0476B9 C0B4F478
% Do you accept this certificate? [yes/no]:yes
```

```
Trustpoint CA certificate accepted.
ssl-proxy(config)#
ssl-proxy(config)# crypto ca authenticate 3tier-sub1
Certificate has the following attributes:
Fingerprint:FE89FB0D BF8450D7 9934C926 6C66708D
Certificate validated - Signed by existing trustpoint CA certificate.
Trustpoint CA certificate accepted.
ssl-proxy(config)#
ssl-proxy(config)# crypto ca authenticate tp-proxy1
Certificate has the following attributes:
Fingerprint:6E53911B E29AE44C ACE773E7 26A098C3
Certificate validated - Signed by existing trustpoint CA certificate.
Trustpoint CA certificate accepted.
```

Enrolling with the Third Level Certificate Authority

```
ssl-proxy(config)# crypto ca enroll tp-proxy1
%
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be:ste.
% The subject name in the certificate will be:ste.
% The serial number in the certificate will be:B0FFF0C2
% Include an IP address in the subject name? [no]:
Request certificate from CA? [yes/no]:yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

ssl-proxy(config)#    Fingerprint: 74390E57 26F89436 6FC52ABE 24E23CD9

ssl-proxy(config)#
*Apr 18 05:10:20.963:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority
```

Manual Certificate Enrollment

The Manual Certificate Enrollment (TFTP and Cut-and-Paste) feature allows you to generate a certificate request and accept certification authority certificates as well as the router's certificates. These tasks are accomplished with a TFTP server or manual cut-and-paste operations. You may want to use TFTP or manual cut-and-paste enrollment in the following situations:

- Your certificate authority does not support Simple Certificate Enrollment Protocol (SCEP) (which is the most commonly used method for sending and receiving requests and certificates).
- A network connection between the router and certificate authority is not possible (which is how a router running Cisco IOS software obtains its certificate).

The following example shows how to configure certificate enrollment using TFTP:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# crypto ca trustpoint tftp_example
ssl-proxy(ca-trustpoint)# enrollment url tftp://10.1.1.2/win2k
ssl-proxy(ca-trustpoint)# rsakeypair pair3
ssl-proxy(ca-trustpoint)# exit
ssl-proxy(config)# crypto ca auth tftp_example
Loading win2k.ca from 10.1.1.2 (via Ethernet0/0.168): !
[OK - 1436 bytes]

Certificate has the following attributes:
Fingerprint: 2732ED87 965F8FEB F89788D4 914B877D
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
ssl-proxy(config)#
```

```
ssl-proxy(config)# crypto ca enroll tftp_example
% Start certificate enrollment ..

% The fully-qualified domain name in the certificate will be: ssl-proxy.cisco.com
% The subject name in the certificate will be: ssl-proxy.cisco.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: 00000000
% Include an IP address in the subject name? [no]:
Send Certificate Request to tftp server? [yes/no]: yes
% Certificate request sent to TFTP Server
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

ssl-proxy(config)#    Fingerprint:  D012D925 96F4B5C9 661FEC1E 207786B7
!!
ssl-proxy(config)# crypto ca import tftp_example cert
% The fully-qualified domain name in the certificate will be: ssl-proxy.cisco.com
Retrieve Certificate from tftp server? [yes/no]: yes
% Request to retrieve Certificate queued

ssl-proxy(config)#
Loading win2k.crt from 10.1.1.2 (via Ethernet0/0.168): !
[OK - 2112 bytes]

ssl-proxy(config)#
*Apr 15 12:02:33.535: %CRYPTO-6-CERTRET: Certificate received from Certificate Authority
ssl-proxy(config)#
```

Configure the Manual Certificate Enrollment (TFTP and cut-and-paste) feature as described at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftmancrt.htm

Importing and Exporting Key Pairs and Certificates

You can import and export key pairs and certificates using either the PKCS12 file format or privacy-enhanced mail (PEM) file format.

To import or export key pairs and certificates, see one of these sections:

- Importing and Exporting a PKCS12 File, page 3-29
- Importing and Exporting PEM Files, page 3-31

Note A test PKCS12 file (test/testssl.p12) is embedded in the SSL software on the module. You can install the file into NVRAM for testing purposes and for proof of concept. After the PKCS12 file is installed, you can import it to a trustpoint, and then assign it to a proxy service configured for testing. For information on installing the test PKCS12 file, see the "Importing the Embedded Test Certificate" section on page A-4.

Importing and Exporting a PKCS12 File

You can use an external PKI system to generate a PKCS12 file, and then import this file to the module.

Note When creating a PKCS12 file, include the entire certificate chain, from server certificate to root certificate, and public and private keys. You can also generate a PKCS12 file from the module and export it.

Note Imported key pairs cannot be exported.

Note If you are using SSH, we recommend using SCP (secure file transfer) when importing or exporting a PKCS12 file. SCP authenticates the host and encrypts the transfer session.

To import or export a PKCS12 file, perform this task:

Command	Purpose
ssl-proxy(config)# **crypto ca** {**import** | **export**} *trustpoint_label* **pkcs12** {**scp:**| **ftp:**| **nvram:**| **rcp:**| **tftp:**} [*pkcs12_filename*[1]] *pass_phrase*[2]	Imports or exports a PKCS12 file. **Note** You do not need to configure a trustpoint before importing the PKCS12 file. Importing keys and certificates from a PKCS12 file creates the trustpoint automatically, if it does not already exist.

[1] If you do not specify the *pkcs12_filename* value, you will be prompted to accept the default filename (the default filename is the *trustpoint_label* value) or enter the filename. For **ftp:** or **tftp:**, include the full path in the *pkcs12_filename* value.

[2] You will receive an error if you enter the pass phrase incorrectly.

This example shows how to import a PKCS12 file using SCP:

```
ssl-proxy(config)# crypto ca import TP2 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Source username [ssl-proxy]? admin-1
Source filename [TP2]? /users/admin-1/pkcs12/TP2.p12

Password:password
Sending file modes:C0644 4379 TP2.p12
!
ssl-proxy(config)#
*Aug 22 12:30:00.531:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
ssl-proxy(config)#
```

This example shows how to export a PKCS12 file using SCP:

```
ssl-proxy(config)# crypto ca export TP1 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Destination username [ssl-proxy]? admin-1
Destination filename [TP1]? TP1.p12

Password:

Writing TP1.p12 Writing pkcs12 file to scp://admin-1@10.1.1.1/TP1.p12

Password:
!
CRYPTO_PKI:Exported PKCS12 file successfully.
ssl-proxy(config)#
```

This example shows how to import a PKCS12 file using FTP:

```
ssl-proxy(config)# crypto ca import TP2 pkcs12 ftp: sky is blue
Address or name of remote host []? 10.1.1.1
Source filename [TP2]? /admin-1/pkcs12/PK-1024
Loading /admin-1/pkcs12/PK-1024 !
[OK - 4339/4096 bytes]
ssl-proxy(config)#
```

This example shows how to export a PKCS12 file using FTP:

```
ssl-proxy(config)# crypto ca export TP1 pkcs12 ftp: sky is blue
Address or name of remote host []? 10.1.1.1
Destination filename [TP1]? /admin-1/pkcs12/PK-1024
Writing pkcs12 file to ftp://10.1.1.1//admin-1/pkcs12/PK-1024

Writing /admin-1/pkcs12/PK-1024 !!
CRYPTO_PKI:Exported PKCS12 file successfully.
ssl-proxy(config)#
```

After you import the PKCS12 file, see the "Verifying Certificates and Trustpoints" section on page 3-36 to verify the certificate and trustpoint information.

Importing and Exporting PEM Files

Note The **crypto ca import pem** command imports only the private key (.prv), the server certificate (.crt), and the issuer certificate authority certificate (.ca). If you have more than one level of certificate authority in the certificate chain, you need to import the root and subordinate certificate authority certificates before this command is issued for authentication. Use cut-and-paste or TFTP to import the root and subordinate certificate authority certificates.

Note Imported key pairs cannot be exported.

Note If you are using SSH, we recommend using SCP (secure file transfer) when importing or exporting PEM files. SCP authenticates the host and encrypts the transfer session.

To import or export PEM files, perform one of these tasks:

Command	Purpose
ssl-proxy(config)# **crypto ca import** *trustpoint_label* **pem** [**exportable**] {**terminal** | **url** {**scp:**| **ftp:**| **nvram:**| **rcp:**| **tftp:**} | **usage-keys**} *pass_phrase*[1,2]	Imports PEM files. **Note** You do not need to configure a trustpoint before importing the PEM files. Importing keys and certificates from PEM files creates the trustpoint automatically, if it does not already exist.
ssl-proxy(config)# **crypto ca export** *trustpoint_label* **pem** {**terminal** | **url** {**scp:**| **ftp:**| **nvram:**| **rcp:**| **tftp:**} [**des** | **3des**] *pass_phrase*[1,2]	Exports PEM files. **Note** Only the key, the server certificate, and the issuer certificate authority of the server certificate are exported. All higher level certificate authorities need to be exported using cut-and-paste of TFTP.

[1] You will receive an error if you enter the pass phrase incorrectly.

[2] A pass phrase protects a PEM file that contains a private key. The PEM file is encrypted by DES or 3DES. The encryption key is derived from the pass phrase. A PEM file containing a certificate is not encrypted and is not protected by a pass phrase.

This example shows how to import PEM files using TFTP:

Note The TP5.ca, TP5.prv, and TP5.crt files should be present on the server.

```
ssl-proxy(config)# crypto ca import TP5 pem url tftp://10.1.1.1/TP5 password
% Importing CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.ca]?
Reading file from tftp://10.1.1.1/TP5.ca
Loading TP5.ca from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1976 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.prv]?
Reading file from tftp://10.1.1.1/TP5.prv
Loading TP5.prv from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 963 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.crt]?
Reading file from tftp://10.1.1.1/TP5.crt
Loading TP5.crt from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1692 bytes]
% PEM files import succeeded.
ssl-proxy(config)#end
ssl-proxy#
*Apr 11 15:11:29.901: %SYS-5-CONFIG_I: Configured from console by console
```

This example shows how to export PEM files using TFTP:

```
ssl-proxy(config)# crypto ca export TP5 pem url tftp://10.1.1.1/tp99 3des password
% Exporting CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.ca]?
% File 'tp99.ca' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.ca!
% Key name: key1
    Usage: General Purpose Key
% Exporting private key...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.prv]?
% File 'tp99.prv' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.prv!
% Exporting router certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [tp99.crt]?
% File 'tp99.crt' already exists.
% Do you really want to overwrite it? [yes/no]: yes
!Writing file to tftp://10.1.1.1/tp99.crt!
ssl-proxy(config)#
```

After you import the PEM files, see the "Verifying Certificates and Trustpoints" section on page 3-36 to verify the certificate and trustpoint information.

Example of Importing PEM Files for Three Levels of Certificate Authority

In this section, the root certificate authority certificate (Tier 1) and intermediate certificate authority certificate (Tier 2) are obtained using the cut-and-paste option of offline enrollment. The intermediate certificate authority certificate (Tier 3), private keys, and router certificate are obtained by importing PEM files.

Using Cut-and-Paste to Obtain Root Certificate Authority-Tier 1 Certificate

```
ssl-proxy(config)# crypto ca trustpoint 3tier-root
ssl-proxy(ca-trustpoint)# enrollment terminal
ssl-proxy(ca-trustpoint)# exit
ssl-proxy(config)# crypto ca authenticate 3tier-root

Enter the base 64 encoded CA certificate.
End with a blank line or the word ‘quit’ on a line by itself

-----BEGIN CERTIFICATE-----
MIID+DCCA2GgAwIBAgIQf9WyCbXCRIxH938UBiXSZTANBgkqhkiG9w0BAQQFADCB
rDEhMB8GCSqGSIb3DQEJARYSaHRoaWFnYXJAY2lzY28uY29tMQswCQYDVQQGEwJV
UzELMAkGA1UECBMCY2ExETAPBgNVBAcTCHNhbiBqb3NlMRwwGgYDVQQKExNjaXNj
byBzeXN0ZW1zLCBpbmMuMQ0wCwYDVQQLEwRpc2J1MS0wKwYDVQQDEyQ2ZWJmOWIz
ZS05YTZkLTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWIwHhcNMDIwNjEzMDAwNTMyWhcN
MDQwNjEzMDAxMTU4WjCBrDEhMB8GCSqGSIb3DQEJARYSaHRoaWFnYXJAY2lzY28u
Y29tMQswCQYDVQQGEwJVUzELMAkGA1UECBMCY2ExETAPBgNVBAcTCHNhbiBqb3Nl
MRwwGgYDVQQKExNjaXNjbyBzeXN0ZW1zLCBpbmMuMQ0wCwYDVQQLEwRpc2J1MS0w
KwYDVQQDEyQ2ZWJmOWIzZS05YTZkLTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWIwgZ8w
DQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBALtGJFtJNwpKKNqu6j6j+C5zxzFyLFvM
v0qAsTIiHWZ2/isSefidLqnoi6L7T2eEnIY/9CCwjjU5DysUnWtCUOEPyO1nmE34
WLYdJnNRxudBLxgZjHfbKl9htfeIdcFv7g3G0Rv6xss6SG9fb3aZbGp+pT1HBJLA
902F057QkrvJAgMBAAGjggEXMIIBEzALBgNVHQ8EBAMCAcYwDwYDVR0TAQH/BAUw
AwEB/zAdBgNVHQ4EFgQUk0u7DvArNTvDbsb/RzzTh6F96w8wgcEGA1UdHwSBuTCB
tjBYoFagVIZSaHR0cDovL3dpbjJrLXNlcnZlcjEuY2lzY28uY29tL0NlcnRFbnJv
bGwvNmViZjliM2UtOWE2ZC00NDAwLTg5M2MtZGQ4NWRjZmU5MTFiLmNybDBaoFig
VoZUZmlsZTovL1xcd2luMmstc2VydmVyMS5jaXNjby5jb21cQ2VydEVucm9sbFw2
ZWJmOWIzZS05YTZkLTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWIuY3JsMBAGCSsGAQQB
gjcVAQQDAgEAMA0GCSqGSIb3DQEBBAUAA4GBAB4bOQTm8Ja9Gk0XZ6cIJ8RuurJ4
nNQJ1jMXQM2pvPhv6Y2zhbjr2VIjfWsqePAkMZOEl7SGAqgk+iomrq1Ivd3Zob2/
U5G4Sn2q0mux0wWV61aG+au0ynz7iIFCSvyr9Ms47VMC1YGWomR8lJsP762bD8lx
oJGL8AEVJHMS4IgF
-----END CERTIFICATE-----

Certificate has the following attributes:
Fingerprint:2732ED87 965F8FEB F89788D4 914B877D
% Do you accept this certificate? [yes/no]:yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Using Cut-and-Paste to Obtain Intermediate Certificate Authority-Tier 2 Certificate

```
ssl-proxy(config)#
ssl-proxy(config)# crypto ca trustpoint 3tier-sub1
ssl-proxy(ca-trustpoint)# enrollment terminal
ssl-proxy(ca-trustpoint)# exit
ssl-proxy(config)# crypto ca authenticate 3tier-sub1

Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself

-----BEGIN CERTIFICATE-----
MIIFhTCCBO6gAwIBAgIKKaR97wAAAAAE6TANBgkqhkiG9w0BAQQFADCBrDEhMB8G
CSqGSIb3DQEJARYSaHRoaWFnYXJAY2lzY28uY29tMQswCQYDVQQGEwJVUzELMAkG
```

```
A1UECBMCY2ExETAPBgNVBAcTCHNhbiBqb3NlMRwwGgYDVQQKExNjaXNjbyBzeXN0
ZW1zLCBpbmMuMQ0wCwYDVQQLEwRpc2J1MS0wKwYDVQQDEyQ2ZWJmOWIzZS05YTZk
LTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWIwHhcNMDIxMjEyMjA1NTE3WhcNMDMxMjEy
MjEwNTE3WjCBkjEdMBsGCSqGSIb3DQEJARYOaW5mb0BjaXNjby5jb20xCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpjYWxpZm9ybmlhMREwDwYDVQQHEwhzYW4gam9zZTEb
MBkGA1UEChMSY2lzY28gc3lzdGVtcywgaW5jMQ0wCwYDVQQLEwRpc2J1MRAwDgYD
VQQDEwdzdWJ0ZXN0MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANK2mcLEc2q1QFLN
PnYhPcluGa927YVznf5NGDzPkZOym+QaTETоUxrQFluY1KyWqKPBed104etyIDyU
5y2DEksCAwEAAaOCAwgwggMEMBAGCSsGAQQBgjcVAQQDAgEAMB0GA1UdDgQWBBSr
fPPd4af6awKMcss9MNWzfYojOTALBgNVHQ8EBAMCAcYwDwYDVR0TAQH/BAUwAwEB
/zCB6AYDVR0jBIHgMIHdgBSTS7sO8Cs1O8Nuxv9HPNOHoX3rD6GBsqSBrzCBrDEh
MB8GCSqGSIb3DQEJARYSaHRoaWFnYXJAY2lzY28uY29tMQswCQYDVQQGEwJVUzEL
MAkGA1UECBMCY2ExETAPBgNVBAcTCHNhbiBqb3NlMRwwGgYDVQQKExNjaXNjbyBz
eXN0ZW1zLCBpbmMuMQ0wCwYDVQQLEwRpc2J1MS0wKwYDVQQDEyQ2ZWJmOWIzZS05
YTZkLTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWKCEH/Vsgm1wkSMR/d/FAYl0mUwgcEG
A1UdHwSBuTCBtjBYoFagVIZSaHR0cDovL3dpbjJrLXNlcnZlcjEuY2lzY28uY29t
L0NlcnRFbnJvbGwvNmViZjliM2UtOWE2ZC00NDAwLTg5M2MtZGQ4NWRjZmU5MTFi
LmNybDBaoFigVoZUZmlsZTovL1xcd2luMmstc2VydmVyMS5jaXNjby5jb21cQ2Vy
dEVucm9sbFw2ZWJmOWIzZS05YTZkLTQ0MDAtODkzYy1kZDg1ZGNmZTkxMWIuY3Js
MIIBAgYIKwYBBQUHAQEEgfUwgfIwdgYIKwYBBQUHMAKGamh0dHA6Ly93aW4yay1z
ZXJ2ZXIxLmNpc2NvLmNvbS9DZXJ0RW5yb2xsL3dpbjJrLXNlcnZlcjEuY2lzY28u
Y29tXzZlYmY5YjNlLTlhNmQtNDQwMC04OTNjLWRkODVkY2ZlOTExYi5jcnQweAYI
KwYBBQUHMAKGbGZpbGU6Ly9cXHdpbjJrLXNlcnZlcjEuY2lzY28uY29tXENlcnRF
bnJvbGxcd2luMmstc2VydmVyMS5jaXNjby5jb21fNmViZjliM2UtOWE2ZC00NDAw
LTg5M2MtZGQ4NWRjZmU5MTFiLmNydDANBgkqhkiG9w0BAQQFAAOBgQANmtHd1fUk
mAIHMgmG4eJsvtLLjaeX9WKt04m5SX2aSUcPUsYKuk+ijAkefZGUD2RWs2gpo8Xm
DURATnOw3WDXekNMPcNWAcrqas8V2TCQAOralycOUuUCuB4LuLY7USlPV9+f5zAT
/43avjGho4oXfSwIqFhAuKQb7RIMRIz8Ag==
-----END CERTIFICATE-----

Certificate has the following attributes:
Fingerprint:D5DBC20D 07EF26BD 2BA3D13F 82CFE648
Certificate validated - Signed by existing trustpoint CA certificate.
Trustpoint CA certificate accepted.
% Certificate successfully imported
```

Importing the PEM Files (Intermediate Certificate Authority-Tier 3, Private Key, and Router Certifiate)

```
ssl-proxy(config)# crypto ca import tp-proxy1 pem url tftp: password
% Importing CA certificate...
Address or name of remote host []? 10.1.1.2
Source filename [tp-proxy1.ca]?
Reading file from tftp://10.1.1.2/tp-proxy1.ca
Loading tp-proxy1.ca from 10.1.1.2 (via Ethernet0/0.3):!
[OK - 1667 bytes]

% Importing private key PEM file...
Address or name of remote host []? 10.1.1.2
Source filename [tp-proxy1.prv]?
Reading file from tftp://10.1.1.2/tp-proxy1.prv
Loading tp-proxy1.prv from 10.1.1.2 (via Ethernet0/0.3):!
[OK - 963 bytes]

% Importing  certificate PEM file...
Address or name of remote host []? 10.1.1.2
Source filename [tp-proxy1.crt]?
Reading file from tftp://10.1.1.2/tp-proxy1.crt
Loading tp-proxy1.crt from 10.1.1.2 (via Ethernet0/0.3):!
[OK - 1509 bytes]
% PEM files import succeeded.
ssl-proxy(config)# ^Z
```

Displaying Certificate Information

```
ssl-proxy# show crypto ca certificates tp-proxy1
Certificate
  Status:Available
  Certificate Serial Number:04A0147B00000000010E
  Certificate Usage:General Purpose
  Issuer:
    CN = sub3ca
     C = US
Subject:
    Name:ssl-proxy.
    Serial Number:B0FFF0C2
    OID.1.2.840.113549.1.9.2 = ssl-proxy.
     OID.2.5.4.5 = B0FFF0C2
  CRL Distribution Point:
    http://sample.cisco.com/sub3ca.crl
  Validity Date:
    start date:18:04:09 UTC Apr 23 2003
    end   date:21:05:17 UTC Dec 12 2003
    renew date:00:00:00 UTC Jan 1 1970
  Associated Trustpoints:tp-proxy1

CA Certificate
  Status:Available
  Certificate Serial Number:6D1E6B0F000000000007
  Certificate Usage:Signature
  Issuer:
    CN = subtest
     C = US
Subject:
    CN = sub3ca
     C = US
CRL Distribution Point:
    http://sample.cisco.com/subtest.crl
  Validity Date:
    start date:22:22:52 UTC Mar 28 2003
    end   date:21:05:17 UTC Dec 12 2003
  Associated Trustpoints:tp-proxy1

ssl-proxy# show crypto ca certificates 3tier-sub1
CA Certificate
  Status:Available
  Certificate Serial Number:29A47DEF0000000004E9
  Certificate Usage:Signature
  Issuer:
    CN = 6ebf9b3e-9a6d-4400-893c-dd85dcfe911b
     C = US
Subject:
    CN = subtest
     C = US
CRL Distribution Point:
    http://sample.cisco.com/6ebf9b3e-9a6d-4400-893c-dd85dcfe911b.crl
  Validity Date:
    start date:20:55:17 UTC Dec 12 2002
    end   date:21:05:17 UTC Dec 12 2003
  Associated Trustpoints:3tier-sub1
```

```
ssl-proxy# show crypto ca certificates 3tier-root
CA Certificate
  Status:Available
  Certificate Serial Number:7FD5B209B5C2448C47F77F140625D265
  Certificate Usage:Signature
  Issuer:
    CN = 6ebf9b3e-9a6d-4400-893c-dd85dcfe911b
     C = US
Subject:
    CN = 6ebf9b3e-9a6d-4400-893c-dd85dcfe911b
     C = US
CRL Distribution Point:
    http://sample.cisco.com/6ebf9b3e-9a6d-4400-893c-dd85dcfe911b.crl
  Validity Date:
    start date:00:05:32 UTC Jun 13 2002
    end   date:00:11:58 UTC Jun 13 2004
  Associated Trustpoints:3tier-root
```

Verifying Certificates and Trustpoints

To verify information about your certificates and trustpoints, perform this task in EXEC mode:

	Command	Purpose
Step 1	`ssl-proxy(ca-trustpoint)#` **`show crypto ca certificates`** [*`trustpoint_label`*]	Displays information about the certificates associated with the specified trustpoint, or all of your certificates, the certificates of the certificate authority, and registration authority certificates.
Step 2	`ssl-proxy(ca-trustpoint)#` **`show crypto ca trustpoints`** [*`trustpoint_label`*]	Displays information about all trustpoints or the specified trustpoint.

Sharing Keys and Certificates

The SSL Services Module supports the sharing of the same key pair by multiple certificates. However, this is not a good practice, because if one key pair is compromised, all the certificates must be revoked and replaced.

Because proxy services are added and removed at different times, the certificates also expire at different times. Some certificate authorities require you to refresh the key pair at the time of renewal. If certificates share one key pair, you need to renew the certificates at the same time. In general, it is easier to manage certificates if each certificate has its own key pair.

The SSL Module does not impose any restrictions on sharing certificates among multiple proxy services and multiple SSL Services Modules. The same trustpoint can be assigned to multiple proxy services.

From a business point of view, the certificate authority may impose restrictions (for example, on the number of servers in a server farm that can use the same certificate). There may be contractual or licensing agreements regarding certificate sharing. Consult with the certificate authority or the legal staff regarding business contractual aspects.

In practice, some web browsers compare the subject name of the server certificate with the hostname or the IP address that appears on the URL. In case the subject name does not match the hostname or IP address, a dialog box appears, prompting the user to verify and accept the certificate. To avoid this step, limit the sharing of certificates based on the hostname or IP address.

Saving Your Configuration

Caution RSA key pairs are saved only to NVRAM. RSA keys are *not* saved with your configuration when you specify any other file system with the **copy system:running-config** *file_system***:** command.

Always remember to save your work when you make configuration changes.

To save your configuration to NVRAM, perform this task:

Command	Purpose
`ssl-proxy# copy [/erase] system:running-config nvram:startup-config`	Saves the configuration, key pairs, and certificate to NVRAM. The key pairs are stored in the private configuration file, and each certificate is stored as a binary file in NVRAM. On bootup, the module will not need to query the certificate authority to obtain the certificates or to auto-enroll. **Note** For security reasons, we recommend that you enter the **/erase** option to erase the public and the private configuration files before updating the NVRAM. If you do not enter the **/erase** option, the key pairs from the old private configuration file may remain in the NVRAM. **Caution** When you enter the **/erase** option, both the current and the backup buffers in NVRAM are erased before the running configuration is saved into NVRAM. If a power failure or reboot occurs after the buffers are erased, but before the running configuration is saved, both configurations might be lost.

Note If you have a large number of files in NVRAM, this task may take up to 2 minutes to finish.

In SSL software release 1.2, the automatic backup of the configuration to NVRAM feature automatically backs up the last saved configuration. If the current write process fails, the configuration is restored to the previous configuration automatically.

Oversized Configuration

If you save an oversized configuration with more than 256 proxy services and 356 certificates, you may encounter a situation where you could corrupt the contents in the NVRAM.

We recommend that you always copy to running-config before saving to NVRAM. When you save the running-config file to a remote server, each certificate is saved as a hex dump in the file. If you copy the running-config file back to running-config and then save it to NVRAM, the certificates are saved again, but as binary files. However, if you copy the running-config file directly from the remote server to startup-config, the certificates saved as hex dumps are also saved, resulting in two copies of the same certificate: one in hex dump and one as a binary file. This is unnecessary, and if the remote file is very large, it may overwrite part of the contents in the NVRAM, which could corrupt the contents.

Verifying the Saved Configuration

To verify the saved configuration, perform this task:

	Command	Purpose
Step 1	`ssl-proxy# show startup-config`	Displays the startup configuration.
Step 2	`ssl-proxy# directory nvram:`	Displays the names and sizes of the files in NVRAM.

Note With the maximum number or proxy services (256) and certificates (356) configured, the output takes up to seven minutes to display.

Erasing the Saved Configuration

To erase a saved configuration, perform this task:

Command	Purpose
`ssl-proxy# erase nvram:`	Erases the startup configuration and the key pairs.
`ssl-proxy# erase /all nvram:`	Erases the startup configuration, the key pairs, the certificates, and all other files from the NVRAM.

Note If you have a large number of files in NVRAM, this task may take up to 2 minutes to finish.

Caution If you erase the saved configuration, the automatic backup configuration in NVRAM is also erased.

Backing Up Keys and Certificates

If an event occurs that interrupts the process of saving the keys and certificates to NVRAM (for example, a power failure), you could lose the keys and certificates being saved. You can obtain public keys and certificates from the certificate authority. However, you cannot recover private keys.

If a secure server is available, back up key pairs and the associated certificate chain by exporting each trustpoint to a PKCS12 file. You can then import the PKCS12 files to recover the keys and certificates.

Security Guidelines

When backing up keys and certificates, observe the following guidelines:

- For each PKCS12, you must select a pass phrase that cannot be easily guessed, and keep the pass phrase well protected. Do not store the PKCS12 file in clear form.
- The backup server must be secure. Allow only authorized personnel to access the backup server.
- When importing or exporting the PKCS12 file (in which you are required to enter a pass phrase), connect directly to the module console or use an SSH session.
- Use SCP for file transfer.

Monitoring and Maintaining Keys and Certificates

The following tasks in this section are optional:

- Deleting RSA Keys from the Module, page 3-39
- Viewing Keys and Certificates, page 3-39
- Deleting Certificates from the Configuration, page 3-40

Deleting RSA Keys from the Module

Caution Deleting the SSH key will disable SSH on the module. If you delete the SSH key, generate a new key. See the "Configuring SSH" section on page 3-4.

Under certain circumstances you may want to delete the module's RSA keys. For example, if you believe the RSA keys were compromised in some way and should no longer be used, you should delete the keys.

To delete all RSA keys from the module, perform this task in global configuration mode:

Command	Purpose
`ssl-proxy(config)#` **`crypto key zeroize rsa`** `[`*`key-label`*`]`	Deletes all RSA key pairs, or the specified key pair. **Caution** If a key is deleted, all certificates that are associated with the key are deleted.

After you delete a module's RSA keys, complete these two additional tasks:

- Ask the certificate authority administrator to revoke your module's certificates at the certificate authority; you must supply the challenge password that you created when you originally obtained the module's certificates with the **crypto ca enroll** command.
- Manually remove the trustpoint from the configuration, as described in the "Deleting Certificates from the Configuration" section on page 3-40.

Viewing Keys and Certificates

To view keys and certificates, enter these commands in EXEC mode:

Command	Purpose
`ssl-proxy#` **`show crypto key mypubkey rsa`**	Displays your module's RSA public keys.
`ssl-proxy#` **`show crypto ca certificates`** `[`*`trustpoint_label`*`]`	Displays information about your certificate, the certificate authority certificate, and any registration authority certificates.

Command	Purpose
`ssl-proxy#` **`show running-config`** `[`**`brief`**`]`	Displays the public keys and the certificate chains. If the *brief* option is specified, the hex dump of each certificate is not displayed.
`ssl-proxy#` **`show ssl-proxy service`** *`proxy-name`*	Displays the key pair and the serial number of the certificate chain used for a specified proxy service. **Note** The *proxy-name* value is case-sensitive.

Deleting Certificates from the Configuration

The module saves its own certificates and the certificate of the certificate authority. You can delete certificates that are saved on the module.

To delete the certificate from the module configuration, perform this task in global configuration mode:

Command	Purpose
`ssl-proxy(config)#` **`no crypto ca trustpoint`** *`trustpoint-label`*	Deletes the certificate.

Assigning a Certificate to a Proxy Service

When you enter the **certificate rsa general-purpose trustpoint** *trustpoint_label* subcommand (under the **ssl-proxy service** *proxy_service* command), a certificate to the specified proxy service is assigned. You can enter the **certificate rsa general-purpose trustpoint** subcommand multiple times for the proxy service.

If the trustpoint label is modified, the proxy service is momentarily taken out of service during the transition. Existing connections continue to use the old certificate until the connections are closed or cleared. New connections use the certificate from the new trustpoint, and the service is available again.

However, if the new trustpoint does not have a certificate yet, the operational status of the service remains down. New connections are not established until the new certificate is available. If the certificate is deleted by entering the **no certificate rsa general-purpose trustpoint** subcommand, the existing connections continue to use the certificate until the connections are closed or cleared. Although the certificate is obsolete, it is not removed from the proxy service until all the connections are closed or cleared.

The following example shows how to assign a trustpoint to a proxy service:

```
ssl-proxy# configure terminal
ssl-proxy(config)# ssl-proxy service s2
ssl-proxy(config-ssl-proxy)# virtual ip 10.1.1.2 p tcp p 443
ssl-proxy(config-ssl-proxy)# server ip 20.0.0.3 p tcp p 80
ssl-proxy(config-ssl-proxy)# inservice
ssl-proxy(config-ssl-proxy)# certficate rsa general trustpoint tp-1
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#
```

```
ssl-proxy# show ssl-proxy service s2
Service id:6, bound_service_id:262
Virtual IP:10.1.1.2, port:443
Server IP:20.0.0.3, port:80
rsa-general-purpose certificate trustpoint:tp-1
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:tp-1
       Serial Number:3C2CD2330001000000DB
    Root CA Certificate:
       Serial Number:313AD6510D25ABAE4626E96305511AC4
  Certificate chain complete
Admin Status:up
Operation Status:up
ssl-proxy#
```

The following example shows how to change a trustpoint for a proxy service:

Note The existing connections continue to use the old certificate until the connections are closed. The operational status of the service changes from up to down, and then up again. New connections use the new certificate.

```
ssl-proxy# configure terminal
ssl-proxy(config)# ssl-proxy service s2
ssl-proxy(config-ssl-proxy)# certificate rsa general trustpoint tp-2
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#
ssl-proxy# show ssl-proxy service s2
Service id:6, bound_service_id:262
Virtual IP:10.1.1.2, port:443
Server IP:20.0.0.3, port:80
rsa-general-purpose certificate trustpoint:tp-2
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:k2
       Serial Number:70FCBFEC000100000D65
    Root CA Certificate:
       Serial Number:313AD6510D25ABAE4626E96305511AC4
  Obsolete certificate chain in use for old connections:
    Server Certificate:
       Key Label:tp-1
       Serial Number:3C2CD2330001000000DB
    Root CA Certificate:
       Serial Number:313AD6510D25ABAE4626E96305511AC4
  Certificate chain complete
Admin Status:up
Operation Status:up
ssl-proxy#
```

Renewing a Certificate

Some certificate authorities require you to generate a new key pair to renew a certificate, while other certificate authorities allow you to use the key pair of the expiring certificate to renew a certificate. Both cases are supported on the SSL Services Module.

The SSL server certificates usually expire in one or two years. Graceful rollover of certificates avoids sudden loss of services.

In the following example, proxy service s2 is assigned trustpoint t2:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy service s2
ssl-proxy(config-ssl-proxy)# certificate rsa general-purpose trustpoint t2
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#

ssl-proxy# show ssl-proxy service s2
Service id:0, bound_service_id:256
Virtual IP:10.1.1.1, port:443
Server IP:10.1.1.10, port:80
Nat pool:pool2
rsa-general-purpose certificate trustpoint:t2
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:k2
       Serial Number:1DFBB1FD000100000D48
    Root CA Certificate:
      Serial Number:313AD6510D25ABAE4626E96305511AC4
  Certificate chain complete
Admin Status:up
Operation Status:up
```

In the following example, the key pair for trustpoint t2 is refreshed, and the old certificate is deleted from the IOS database. Graceful rollover starts automatically for proxy service s2.

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# crypto key generate rsa general-key k2 exportable
% You already have RSA keys defined named k2.
% Do you really want to replace them? [yes/no]:yes
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]:1024
% Generating 1024 bit RSA keys ...[OK]
ssl-proxy(config)#end

ssl-proxy# show ssl-proxy service s2
Service id:0, bound_service_id:256
Virtual IP:10.1.1.1, port:443
Server IP:10.1.1.10, port:80
Nat pool:pool2
rsa-general-purpose certificate trustpoint:t2
  Certificate chain in graceful rollover, being renewed:
    Server Certificate:
       Key Label:k2
       Serial Number:1DFBB1FD000100000D48
    Root CA Certificate:
      Serial Number:313AD6510D25ABAE4626E96305511AC4
  Server certificate in graceful rollover
Admin Status:up
Operation Status:up
```

In the following example, existing and new connections use the old certificate until trustpoint t2 reenrolls. After trustpoint t2 reenrolls, new connections use the new certificate; existing connections continue to use the old certificate until the connections are closed.

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# crypto ca enroll t2
%
% Start certificate enrollment ..

% The subject name in the certificate will be:CN=host1.cisco.com
% The subject name in the certificate will be:ssl-proxy.cisco.com
% The serial number in the certificate will be:00000000
% The IP address in the certificate is 10.1.1.1

% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

    Fingerprint: 6518C579 A0498063 C5795057 A6170 075

ssl-proxy(config)# end
*Sep 24 15:19:34.339:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

ssl-proxy# show ssl-proxy service s2
Service id:0, bound_service_id:256
Virtual IP:10.1.1.1, port:443
Server IP:10.1.1.10, port:80
Nat pool:pool2
rsa-general-purpose certificate trustpoint:t2
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:k2
       Serial Number:2475A2FC000100000D4D
    Root CA Certificate:
      Serial Number:313AD6510D25ABAE4626E96305511AC4
  Obsolete certificate chain in use for old connections:
    Server Certificate:
       Key Label:k2
       Serial Number:1DFBB1FD000100000D48
    Root CA Certificate:
      Serial Number:313AD6510D25ABAE4626E96305511AC4
  Certificate chain complete
Admin Status:up
Operation Status:up
```

In the following example, the obsolete certificate is removed after all of the existing connections are closed.

```
ssl-proxy# show ssl-proxy service s2
Service id:0, bound_service_id:256
Virtual IP:10.1.1.1, port:443
Server IP:10.1.1.10, port:80
Nat pool:pool2
rsa-general-purpose certificate trustpoint:t2
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:k2
       Serial Number:2475A2FC000100000D4D
    Root CA Certificate:
      Serial Number:313AD6510D25ABAE4626E96305511AC4
  Certificate chain complete
Admin Status:up
Operation Status:up
```

Enabling Key and Certificate History

When you enter the **ssl-proxy pki history** command, the SSL proxy services key and certificate history are enabled. This history creates a record for each addition or deletion of the key pair and certificate chain for a proxy service.

When you enter the show **ssl-proxy certificate-history** command, the records are displayed. Each record logs the service name, key pair name, time of generation or import, trustpoint name, certificate subject name and issuer name, serial number, and date.

You can store up to 512 records in memory. For each record, a syslog message is generated. The oldest records are deleted after the limit of 512 records is reached.

To enable key and certificate history and display the records, perform this task:

	Command	Purpose
Step 1	`ssl-proxy(config)#` **`ssl-proxy pki history`**	Enables key and certificate history.
Step 2	`ssl-proxy#` **`show ssl-proxy certificate-history`** `[`**`service`** *`proxy_service`*`]`	Displays key and certificate history records for all services or the specified service.

This example shows how to enable key and certificate history and display the records for a specified proxy service:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)#ssl-proxy pki history
ssl-proxy(config)#end

ssl-proxy# show ssl-proxy certificate-history service s2
  Record 1, Timestamp:00:00:22, 17:44:18 UTC Sep 29 2002
    Installed Server Certificate, Index 0
    Proxy Service:s2,  Trust Point:t2
    Key Pair Name:k2,  Key Usage:RSA General Purpose, Not Exportable
    Time of Key Generation:06:29:08 UTC Sep 28 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 = ssl-proxy.cisco.com,
OID.1.2.840.113549.1.9.8 = 10.1.1.1
    Issuer Name:CN = TestCA, OU = Lab, O = Cisco Systems, L = San Jose, ST = CA, C = US,
EA =<16> simpson-pki@cisco.com
    Serial Number:3728ADCD000100000D4F
    Validity Start Time:15:56:55 UTC Sep 28 2002
    End Time:16:06:55 UTC Sep 28 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record
Total number of certificate history records displayed = 1
```

Configuring SSL Proxy Services

You define SSL proxy services using the **ssl-proxy service** *ssl_proxy_name* command. You can configure the virtual IP address and port associated with the proxy service and the associated target IP address and port. You also can define TCP and SSL policies for both client (**virtual**) and server (**server**) sides of the proxy.

To configure SSL proxy services, perform this task:

	Command	Purpose
Step 1	ssl-proxy(config)# **ssl-proxy service** *proxy_name*	Defines the name of the SSL proxy service. **Note** The *proxy-name* value is case-sensitive.
Step 2	ssl-proxy(config-ssl-proxy)# **virtual ipaddr** *ip_addr* **protocol tcp port** *port* [**secondary**[1,2,3]]	Defines the virtual server IP address, transport protocol (TCP), and port number for which the SSL Services Module is the proxy.
Step 3	ssl-proxy(config-ssl-proxy)# **virtual policy tcp** *tcp_policy_name*[4]	Applies a TCP policy to the client side of the proxy server. See the "Configuring TCP Policy" section on page 3-48 for TCP policy parameters.
Step 4	ssl-proxy(config-ssl-proxy)# **virtual policy ssl** *ssl_policy_name*[4]	Applies an SSL policy to the client side of the proxy server. See the "Configuring SSL Policy" section on page 3-46 for SSL policy parameters.
Step 5	ssl-proxy(config-ssl-proxy)# **server ipaddr** *ip_addr* **protocol tcp port** *port*	Defines the IP address, port number, and the transport protocol of the target server for the proxy. **Note** The target server IP address can be a virtual IP address of an SLB device or a real IP address of a web server.
Step 6	ssl-proxy(config-ssl-proxy)# **server policy tcp** *tcp_policy_name*	Applies a TCP policy to the server side of the proxy server. See the "Configuring TCP Policy" section on page 3-48
Step 7	ssl-proxy(config-ssl-proxy)# **nat** {**server** | **client** *natpool_name*}	Specifies the usage of either server NAT[5] or client NAT for the server-side connection opened by the SSL Services Module. See the "Configuring NAT" section on page 3-49.
Step 8	ssl-proxy(config-ssl-proxy)# **certificate rsa general-purpose trustpoint** *trustpoint_label*	Applies a trustpoint configuration to the proxy server[6]. **Note** The trustpoint defines the certificate authority server, the key parameters and key-generation methods, and the certificate enrollment methods for the proxy server. See the "Declaring the Trustpoint" section on page 3-25 for information on configuring the trust point.
Step 9	ssl-proxy(config-ssl-proxy)# **inservice**	Sets the proxy server as administratively Up.

1. When you enter the **secondary** keyword, the SSL Services Module does not respond to ARP requests of the virtual IP address.
2. You can enter the **secondary** keyword when the SSL Services Module is used in a standalone configuration or when the SSL Services Module is used as a real server on a load balancer (like the CSM) configured in dispatch mode (MAC address rewrite). See the "Configuring Different Modes of Operation" section on page 3-55 for information on configuring the SSL Services Module with the CSM.
3. You can enter the **secondary** keyword if you configure multiple devices using the same virtual IP address. The virtual IP address can be any legal IP address, and does not have to be in the VLAN (subnet) connected to the SSL Services Module.
4. If you create a policy without specifying any parameters, the policy is created using the default values.
5. NAT = network address translation
6. If the key (modulus) size is other than 512, 768, 1024, 1536, or 2048, you will receive an error and the trustpoint configuration is not applied. Replace the key by generating a key (using the same *key_label*) and specifying a supported modulus size, then repeat Step 8.

This example shows how to configure SSL proxy services:

```
ssl-proxy(config)# ssl-proxy service proxy1
ssl-proxy(config-ssl-proxy)# virtual ipaddr 10.1.1.100 protocol tcp port 443
ssl-proxy(config-ssl-proxy)# server ipaddr 10.1.1.1 protocol tcp port 80
ssl-proxy(config-ssl-proxy)# virtual policy tcp tcp2
ssl-proxy(config-ssl-proxy)# server policy tcp tcp2
ssl-proxy(config-ssl-proxy)# virtual policy ssl ssl1
ssl-proxy(config-ssl-proxy)# nat client t2
ssl-proxy(config-ssl-proxy)# certificate rsa general-purpose trustpoint tp1
ssl-proxy(config-ssl-proxy)# inservice
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#
```

Advanced Configuration

This section describes the following advanced configurations:

- Configuring Policies, page 3-46
- Configuring NAT, page 3-49
- Enabling the Cryptographic Self-Test, page 3-50
- Collecting Crash Information, page 3-52
- Enabling VTS Debugging, page 3-54

Configuring Policies

See the "Configuring SSL Proxy Services" section on page 3-44 for procedures for applying policies to a proxy service.

This section describes how to configure SSL and TCP policies:

- Configuring SSL Policy, page 3-46
- Configuring TCP Policy, page 3-48

Configuring SSL Policy

Note The SSL commands for the SSL Services Module apply either globally or to a particular proxy server.

The SSL policy template allows you to define parameters associated with the SSL stack.

If you do not associate an SSL policy with a particular proxy server, the proxy server enables all the supported cipher suites and versions by default.

To define an SSL policy template and associate an SSL policy with a particular proxy server, perform this task:

	Command	Purpose
Step 1	ssl-proxy (config)# **ssl-proxy policy ssl** *ssl_policy_name*	Defines SSL policy templates.
Step 2	ssl-proxy (config-ssl-policy)# **cipher** {**rsa-with-rc4-128-md5** | **rsa-with-rc4-128-sha** | **rsa-with-des-cbc-sha** | **rsa-with-3des-ede-cbc-sha** | **others...**}	Configures a list of cipher-suite names acceptable to the proxy server. The cipher-suite names follow the same convention as that of existing SSL stacks.
Step 3	ssl-proxy (config-ssl-policy)# **protocol** {**ssl3** | **tls1** | **all**}	Defines the various protocol versions supported by the proxy server.
Step 4	ssl-proxy (config-ssl-policy)# **timeout handshake** *time*	Configures how long the module keeps the connection in handshake phase. The valid range is 0 to 65535 seconds.
Step 5	ssl-proxy (config-ssl-policy)# **close-protocol strict**	Configures the SSL close-protocol behavior. When enabled, a close-notify alert message is sent to the client, and a close-notify alert message is expected from the client. When disabled, the server sends a close-notify alert message to the client; however, the server does not expect a close-notify alert before tearing down the session. Close-protocol is disabled by default.
Step 6	ssl-proxy (config-ssl-policy)# **session-cache**	Enables the session-caching feature. Session caching is enabled by default.
Step 7	ssl-proxy (config-ssl-policy)# **timeout session** *timeout* [**absolute**[1]]	Configures the amount of time that an entry is kept in the session cache. The valid range is 1 to 72000 seconds. **Note** The **absolute** keyword is required in order to configure session-cache size. **Note** The **absolute** keyword specifies that the session entry is kept in the session cache for the specified *timeout*. When the **absolute** keyword is specified, new incoming connections are rejected if there are no free entries available in the session cache.
Step 8	ssl-proxy (config-ssl-policy)# **session-cache size** *size*	(Optional) Specifies the size of the session cache[1]. The valid range is 1 to 262143 entries. **Note** Specify the session cache size when you enter the **absolute** keyword with the **timeout session** command. If this command is not entered or if no *size* is specified, the session cache size is the maximum size (262,144).

[1] When the **absolute** keyword is configured, the session entry is not reused until the configured session timeout expires. When **absolute** is configured, the number of session entries required is equal to (new_connection_rate * absolute_timeout). Depending on the timeout configuration and the new connection rate, the number of session entries might be very large. In this case, you can limit the number of session entries used by configuring the session-cache size.

Configuring TCP Policy

Note The TCP commands for the SSL Services Module apply either globally or to a particular proxy server.

The TCP policy template allows you to define parameters associated with the TCP stack.

To define an TCP policy template and associate an TCP policy with a particular proxy server, perform this task:

	Command	Purpose
Step 1	`ssl-proxy (config)# ssl-proxy policy tcp tcp_policy_name`	Defines TCP policy templates. All defaults are assumed unless otherwise specified.
Step 2	`ssl-proxy (config-ssl-policy)# mss max_segment_size`	Configures the maximum segment size (MSS), in bytes, that the connection will identify in the SYN packet that it generates. **Note** This command allows you to configure a different MSS for the client side and server side of the proxy server. The default is 1460 bytes. The valid range is from 256 to 2460 bytes[1].
Step 3	`ssl-proxy (config-ssl-policy)# timeout syn time`	Configures the connection establishment timeout. The default is 75 seconds. The valid range is from 5 to 75 seconds.
Step 4	`ssl-proxy (config-ssl-policy)# timeout reassembly time`	Configures the amount of time, in seconds, before the reassembly queue is cleared. If the transaction is not complete within the specified time, the reassembly queue is cleared and the connection is dropped. The default is 60 seconds. The valid range is 0 to 960 seconds (0 = disabled).
Step 5	`ssl-proxy (config-ssl-policy)# timeout inactivity time`	Configures the amount of time, in seconds, that an established connection can be inactive. The default is 600 seconds. The valid range is 0 to 960 seconds (0 = disabled).
Step 6	`ssl-proxy (config-ssl-policy)# timeout fin-wait time`	Configures the FIN wait timeout in seconds. The default value is 600 seconds. The valid range is from 75 to 600 seconds.
Step 7	`ssl-proxy (config-ssl-policy)# buffer-share rx buffer_limit`	Configures the maximum receive buffer share per connection in bytes. The default value is 32768 bytes. The valid range is from 8192 to 262144 bytes.
Step 8	`ssl-proxy (config-ssl-policy)# buffer-share tx buffer_limit`	Configures the maximum transmit buffer share per connection in bytes. The default value is 32768 bytes. The valid range is from 8192 to 262144 bytes.

[1] If fragmentation occurs, decrease the MSS value until there is no fragmentation.

Configuring NAT

Client connections originate from the client and are terminated on the SSL Services Module. Server connections originate from the SSL Services Module.

You can configure client NAT, server NAT, or both, on the server connection.

Server NAT

The server IP address configured with the **ssl-proxy service** command specifies the IP address and port for the destination device, either the CSM or the real server for which the SSL Services Module acts as a proxy. If you configure server NAT, the server IP address is used as the destination IP address for the server connection. If the server NAT is not configured, the destination IP address for the server connection is the same as the **virtual ipaddress** for which SSL Services Module is a proxy. The SSL Services Module always performs the port translation by using the port number entered in the **server ipaddress** subcommand.

To configure server NAT, enter the **nat server** subcommand under the **ssl-proxy service** command:

	Command	Purpose
Step 1	`ssl-proxy (config)#` **`ssl-proxy service`** *`ssl_proxy_name`*	Defines the SSL proxy service.
Step 2	`ssl-proxy (config-ssl-proxy)#` **`nat server`**	Enables a NAT server address for the server connection of the specified service SSL offload.

Client NAT

If you configure client NAT, the server connection source IP address and port are derived from a NAT pool. If client NAT is not configured, the server connection source IP address and port are derived from the source IP address and source port of the client connection.

Allocate enough IP addresses to satisfy the total number of connections supported by the SSL Services Module (256,000 connections). Assuming you have 32,000 ports per IP address, configure 8 IP addresses in the NAT pool. If you try to configure fewer IP addresses than required by the total connections supported by the SSL Services Module, the command is rejected.

To configure a NAT pool and assign the NAT pool to the proxy service, perform this task:

	Command	Purpose
Step 1	`ssl-proxy (config)#` **`ssl-proxy natpool`** *`natpool_name start_ip_addr end_ip_addr netmask`*	Defines a pool of IP addresses which the SSL Services Module uses for implementing the client NAT.
Step 2	`ssl-proxy (config-ssl-proxy)#` **`ssl-proxy service`** *`ssl_proxy_name`*	Defines the SSL proxy service.
Step 3	`ssl-proxy (config-ssl-proxy)#` **`nat client`** *`natpool_name`*	Configures a NAT pool for the client address used in the server connection of the specified service SSL offload.

Enabling the Cryptographic Self-Test

Note The power-on crypto chip self-test and key test are run only once at bootup.

Note Use the self-test for troubleshooting only. Running this test will impact run-time performance.

To run the self-test, perform this task:

	Command	Purpose
Step 1	`ssl-proxy(config)#` **`ssl-proxy crypto self-test time-interval`** *`time`*	Enables the crytographic self-test. The default value for *time* is 3 seconds; valid values are 1 though 8.
Step 2	`ssl-proxy(config)#` **`show ssl-proxy stats`** {**`crypto`** | **`ipc`** | **`pki`** | **`service`** | **`ssl`** | **`tcp`**}	Displays specified statistics information.

This example shows how to enable the cryptographic self-test and display cryptographic information:

```
ssl-proxy(config)# ssl-proxy crypto self-test time-interval 1
ssl-proxy(config)# end
ssl-proxy# show ssl-proxy stats crypto
Crypto Statistics from SSL Module:1
  Self-test is running
  Current device index is 1
  Time interval between tests is 1 seconds
  Device 0 statistics:
    Total Number of runs:50
    Runs all passed:50
    Number of timer error:0
    ---------------------------------------------------------
    Test Name                     Passed  Failed  Did-not-run
    ---------------------------------------------------------
      0  Power-on Crypto chip sel    1      0      0
      1  Power-on Crypto chip key    1      0      0
      2  Hash Test Case 1           50      0      0
      3  Hash Test Case 2           50      0      0
      4  Hash Test Case 3           50      0      0
      5  Hash Test Case 4           50      0      0
      6  SSL3 MAC Test Case 1       50      0      0
      7  SSL3 MAC Test Case 2       50      0      0
      8  TLS1 MAC Test Case 1       50      0      0
      9  TLS1 MAC Test Case 2       50      0      0
     10  DES Server Test            50      0      0
     11  DES Encrypt Test 1         50      0      0
     12  DES Decrypt Test 1         50      0      0
     13  DES Encrypt Test 2         50      0      0
     14  DES Decrypt Test 2         50      0      0
     15  ARC4 Test Case 1           50      0      0
     16  ARC4 Test Case 2           50      0      0
     17  ARC4 Test Case 3           50      0      0
     18  ARC4 State Test Case 1     50      0      0
     19  ARC4 State Test Case 2     50      0      0
     20  ARC4 State Test Case 3     50      0      0
     21  ARC4 State Test Case 4     50      0      0
     22  HMAC Test Case 1           50      0      0
     23  HMAC Test Case 2           50      0      0
     24  Random Bytes Generation    50      0      0
     25  RSA Encrypt/Decrypt Test   50      0      0
     26  Master Secret Generation   50      0      0
     27  Key Material Generation    50      0      0
     28  SSL3 Handshake Hash Test   50      0      0
     29  TLS1 Handshake Hash Test   50      0      0

  Device 1 statistics:
    Total Number of runs:49
    Runs all passed:49
    Number of timer error:0
    ---------------------------------------------------------
    Test Name                     Passed  Failed  Did-not-run
    ---------------------------------------------------------
      0  Power-on Crypto chip sel    1      0      0
      1  Power-on Crypto chip key    1      0      0
      2  Hash Test Case 1           50      0      0
      3  Hash Test Case 2           50      0      0
      4  Hash Test Case 3           50      0      0
      5  Hash Test Case 4           50      0      0
      6  SSL3 MAC Test Case 1       50      0      0
      7  SSL3 MAC Test Case 2       50      0      0
      8  TLS1 MAC Test Case 1       50      0      0
      9  TLS1 MAC Test Case 2       50      0      0
     10  DES Server Test            50      0      0
     11  DES Encrypt Test 1         50      0      0
     12  DES Decrypt Test 1         50      0      0
     13  DES Encrypt Test 2         50      0      0
     14  DES Decrypt Test 2         50      0      0
     15  ARC4 Test Case 1           50      0      0
     16  ARC4 Test Case 2           50      0      0
     17  ARC4 Test Case 3           50      0      0
     18  ARC4 State Test Case 1     49      0      0
     19  ARC4 State Test Case 2     49      0      0
     20  ARC4 State Test Case 3     49      0      0
     21  ARC4 State Test Case 4     49      0      0
     22  HMAC Test Case 1           49      0      0
     23  HMAC Test Case 2           49      0      0
     24  Random Bytes Generation    49      0      0
     25  RSA Encrypt/Decrypt Test   49      0      0
     26  Master Secret Generation   49      0      0
     27  Key Material Generation    49      0      0
     28  SSL3 Handshake Hash Test   49      0      0
     29  TLS1 Handshake Hash Test   49      0      0
```

This example shows how to display PKI information:

```
ssl-proxy# show ssl-proxy stats pki
PKI Memory Usage Counters:
  Malloc count:252
  Setstring count:46
  Free count:222
  Malloc failed:0
  Ipc alloc count:56
  Ipc free count:84
  Ipc alloc failed:0
PKI IPC Counters:
  Request buffer sent:28
  Request buffer received:0
  Request duplicated:0
  Request send failed:0
  Response buffer sent:0
  Response buffer received:28
  Response timeout:0
  Response failed:0
  Response with error reported by SSL Processor:0
  Response with no request:0
  Response duplicated:0
  Message type error:0
  Message length error:0
Key Certificate Table Current Usage (cannot be cleared):
  Total number of entries in table:8192
  Entries in use:7
  Free entries:8185
  Complete server entries:5
  Incomplete new/renew server entries:1
  Retiring server entries:0
  Obsolete server entries:0
  Complete intermediate CA cert:0
  Complete root CA cert:1
  Obsolete intermediate CA cert:0
  Obsolete root CA cert:0
PKI Accumulative Counters (cannot be cleared):
  Proxy service trustpoint added:7
  Proxy service trustpoint deleted:1
  Proxy service trustpoint modified:0
  Keypair added:6
  Keypair deleted:1
  Wrong key type:1
  Server certificate added:6
  Server certificate deleted:1
  Server certificate rolled over:0
  Server certificate completed:6
  Intermediate CA certificate added:0
  Intermediate CA certificate deleted:0
  Root CA certificate added:1
  Root CA certificate deleted:0
  Certificate overwritten:0
  No free table entries:0
  Rollover failed:0
  History records written:4
  History records currently kept in memory:4
  History records have been cleared:0 times

ssl-proxy#
```

Collecting Crash Information

The crash-info feature collects information necessary for developers to fix software-forced resets. Enter the **show ssl-proxy crash-info** command to collect software-forced reset information. You can retrieve only the latest crash-info in case of multiple software-forced resets. The **show ssl-proxy crash-info** command takes 1 to 6 minutes to complete the information collection process.

Note The **show stack** command is not a supported command to collect software-forced reset information on the SSL Service Module.

The following example shows how to collect software-forced reset information:

```
ssl-proxy# show ssl-proxy crash-info

===== SSL SERVICE MODULE - START OF CRASHINFO COLLECTION =====


------------- COMPLEX 0 [FDU_IOS] ----------------------

NVRAM CHKSUM:0xEB28
NVRAM MAGIC:0xC8A514F0
NVRAM VERSION:1

++++++++++ CORE 0 (FDU) ++++++++++++++++++++++

   CID:0
   APPLICATION VERSION:2003.04.15 14:50:20 built for cantuc
   APPROXIMATE TIME WHEN CRASH HAPPENED:14:06:04 UTC Apr 16 2003
   THIS CORE DIDN'T CRASH
   TRACEBACK:222D48 216894
   CPU CONTEXT  -----------------------------

$0 :00000000, AT :00240008, v0 :5A27E637, v1 :000F2BB1
a0 :00000001, a1 :0000003C, a2 :002331B0, a3 :00000000
t0 :00247834, t1 :02BFAAA0, t2 :02BF8BB0, t3 :02BF8BA0
t4 :02BF8BB0, t5 :00247834, t6 :00000000, t7 :00000001
s0 :00000000, s1 :0024783C, s2 :00000000, s3 :00000000
s4 :00000001, s5 :0000003C, s6 :00000019, s7 :0000000F
t8 :00000001, t9 :00000001, k0 :00400001, k1 :00000000
gp :0023AE80, sp :031FFF58, s8 :00000019, ra :00216894
LO :00000000, HI :0000000A, BADVADDR :828D641C
EPC :00222D48, ErrorEPC :BFC02308, SREG :34007E03
Cause 0000C000 (Code 0x0):Interrupt exception

CACHE ERROR registers  -------------------

CacheErrI:00000000, CacheErrD:00000000
ErrCtl:00000000, CacheErrDPA:0000000000000000

   PROCESS STACK -----------------------------
      stack top:0x3200000

   Process stack in use:

   sp is close to stack top;

   printing 1024 bytes from stack top:

031FFC00:06405DE0 002706E0 0000002D 00000001  .@]`.'.`...-....
031FFC10:06405DE0 002706E0 00000001 0020B800  .@]`.'.`..... 8.
031FFC20:031FFC30 8FBF005C 14620010 24020004  ..|0.?.\.b..$...
...........
...........
...........
FFFFFFD0:00000000 00000000 00000000 00000000 ................
FFFFFFE0:00627E34 00000000 00000000 00000000 .b~4............
FFFFFFF0:00000000 00000000 00000000 00000006 ................


===== SSL SERVICE MODULE - END OF CRASHINFO COLLECTION =======
```

Enabling VTS Debugging

A virtual terminal server (VTS) is built into the SSL Service Module for debugging different processors (FDU, TCP, SSL) on the module.

Note Use the TCP debug commands only to troubleshoot basic connectivity issues under little or no load conditions (for instance, when no connection is being established to the virtual server or real server).

If you use TCP debug commands, the TCP module displays large amounts of debug information on the console, which can significantly slow down module performance. Slow module performance can lead to delayed processing of TCP connection timers, packets, and state transitions.

From a workstation or PC, make a Telnet connection to one of the module's VLAN IP addresses to reach the FDU (port 2001), TCP (port 2002), and SSL (port 2003) processor on the SSL Services Module.

To display debugging information, perform this task:

Command	Purpose
`ssl-proxy# [`**`no`**`]` **`debug ssl-proxy`** `{`**`fdu`**` | `**`ssl`**` | `**`tcp`**`} [`*`type`*`]`	Turns on or off the debug flags for the specified system component.

After you make the Telnet connection, enter the **debug ssl-proxy** {**tcp** | **fdu** | **ssl**} command from the SSL Services Module console. One connection is sent from a client and displays the logs found in TCP console.

The following example shows how to display the log for TCP states for a connection and verify the debugging state:

```
ssl-proxy# debug ssl-proxy tcp state
ssl-proxy# show debugging
STE Mgr:
  STE TCP states debugging is on
```

The following example shows the output from the workstation or PC:

```
Conn 65066 state CLOSED --> state SYN_RECEIVED
Conn 65066 state SYN_RECEIVED --> state ESTABLISHED
Conn 14711 state CLOSED --> state SYN_SENT
Conn 14711 state SYN_SENT --> state ESTABLISHED
Conn 14711 state ESTABLISHED --> state CLOSE_WAIT
Conn 65066 state ESTABLISHED --> state FIN_WAIT_1
Conn 65066 state FIN_WAIT_1 --> state FIN_WAIT_2
Conn 65066 state FIN_WAIT_2 --> state TIME_WAIT
Conn 14711 state CLOSE_WAIT --> state LAST_ACK
Conn 14711 state LAST_ACK --> state CLOSED
##############Conn 65066 state TIME_WAIT --> state CLOSED
```

Configuring Different Modes of Operation

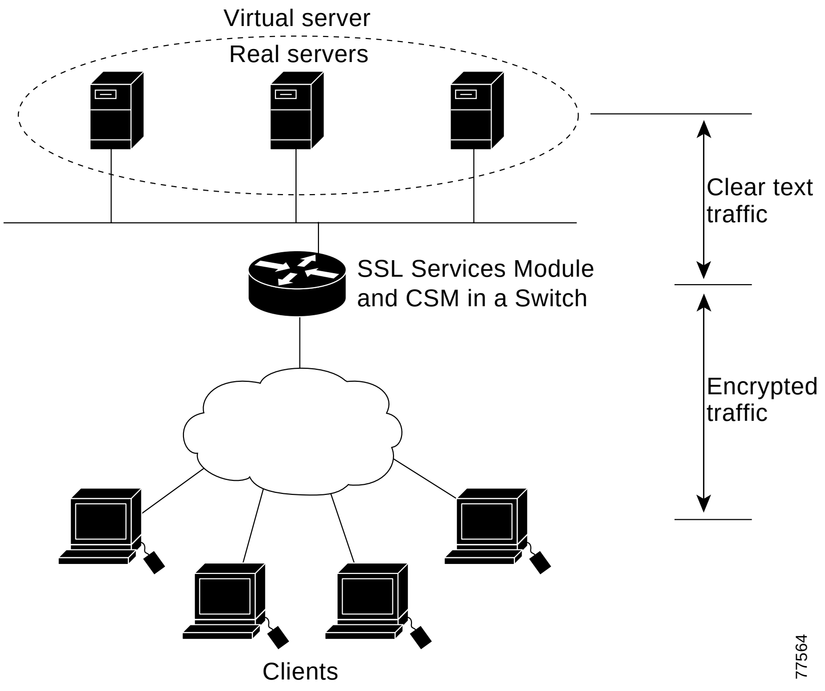
The SSL Services Module operates either in a standalone configuration or with a Content Switching Module (CSM). In a standalone configuration, secure traffic is directed to the SSL Services Module using policy-based routing. When used with a CSM, only encrypted client traffic is forwarded to the SSL Services Module, while clear text traffic is forwarded to real servers.

The following sections describe how to configure the SSL Services Module in a standalone configuration or with a CSM:

- Configuring Policy-Based Routing, page 3-55
- Configuring the Content Switching Module, page 3-61

Figure 3-2 shows a sample network topology with an SSL Services Module and a CSM in a single Catalyst 6500 series switch.

Figure 3-2 Sample Network Layout—SSL Services Module with CSM



Configuring Policy-Based Routing

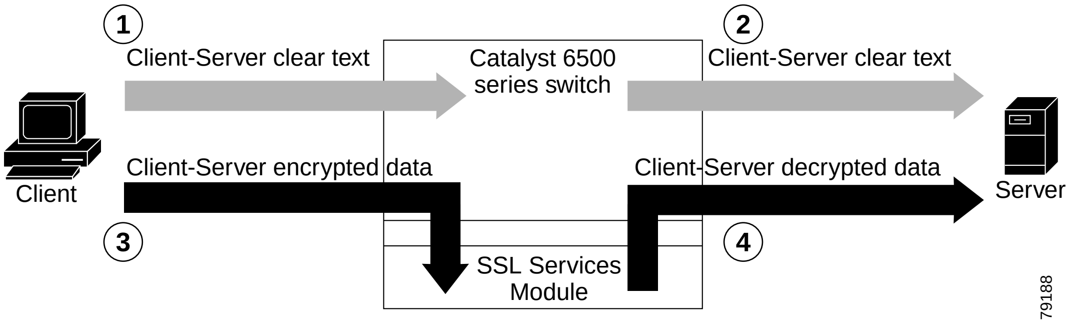
In a standalone configuration, encrypted SSL traffic is directed to the SSL Services Module using policy-based routing.

When you configure policy-based routing on the SSL Services Module, use the following guidelines:

- Configure clients and servers on separate subnets.
- Configure two VLANs (one for each subnet) on the switch.
- Configure IP interfaces on each VLAN.
- Configure an IP interface on the server-side VLAN of the SSL Services Module.

Two flows exist for each direction of traffic. In the client-to-server direction, traffic flow originates from the client as either clear text or as encrypted data. (See Figure 3-3.) In the server-to-client direction, all traffic originates from the server as clear text. However, depending on the source port, the traffic in the server-to-client direction may or may not be encrypted by the SSL Services Module before being forwarded to the client.

Figure 3-3 Client-to-Server Traffic Flow—Standalone Configuration



In Figure 3-3, the client sends clear text traffic to the server (as shown in flow 1). The switch then forwards clear text traffic to the server (flow 2).

The client sends encrypted traffic to the server (port 443); policy-based routing intercepts the traffic and forwards it to the SSL Services Module (flow 3). The SSL Services Module decrypts the traffic and forwards the stream to a well known port (a port that has been configured on the server to expect decrypted traffic) (flow 4).

To enable policy-based routing, perform this task beginning in global configuration mode:

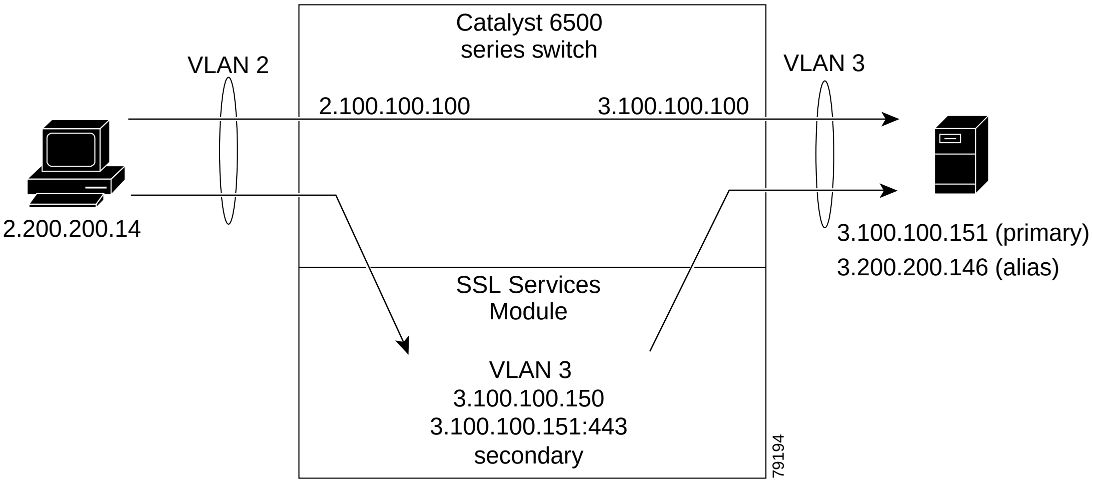
	Command	Purpose
Step 1	`Router(config)# ip access-list extended name`	Defines an IP extended access list.
Step 2	`Router(config-ext-nacl)# permit tcp source source-wildcard operator port destination destination-wildcard operator port`	Specifies conditions for the named access list. **Note** Use the **any** keyword as an abbreviation for a *source* and *source-wildcard* or *destination* and *destination-wildcard* of 0.0.0.0 255.255.255.255.
Step 3	`Router(config-ext-nacl)# route-map map-tag [permit | deny] [sequence-number]`	Defines a route map to control where packets are output. **Note** This command puts the switch into route map configuration mode.
Step 4	`Router(config-route-map)# match ip address name`	Specifies the match criteria. Matches the source and destination IP address that is permitted by one or more standard or extended access lists.
Step 5	`Router(config-route-map)# set ip next-hop ip-address`	Sets the next hop to which to route the packet (the next hop must be adjacent).

	Command	Purpose
Step 6	`Router(config-route-map)# interface interface-type interface-number`	Specifies the interface. **Note** This command puts the switch into interface configuration mode.
Step 7	`Router(config-if)# ip policy route-map map-tag`	Identifies the route map to use for policy-based routing. **Note** One interface can only have one **route-map** tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If there is no match, packets will be routed as usual.

Policy-Based Routing Configuration Example

This section shows a policy-based routing configuration example using a real client and a real server.

Figure 3-4 Client-to-Server Traffic Flow Example



In Figure 3-4, the SSL Services Module and the real server both have the IP address 3.100.100.151. The IP address on the SSL Services Module is configured as **secondary** and will not reply to ARP requests for this address, which avoids the duplicate IP address issue.

The client (2.200.200.14) is attached to a VLAN 2 switchport (access mode). The client's default gateway is 2.100.100.100 (VLAN 2 IP address on the supervisor engine).

The real server is attached to a VLAN 3 switchport (access mode). The real server's default gateway is 3.100.100.100 (VLAN 3 IP address on the supervisor engine). The real server has two addresses: 3.100.100.151 (primary) and 3.200.200.146 (alias).

Clear-text (HTTP) traffic destined for 3.100.100.151 port 80 is sent directly to the real server, which bypasses the SSL Services Module.

With policy-based routing, SSL traffic destined for 3.100.100.151 port 443 is redirected to the SSL Services Module for decryption. The decrypted traffic is sent to 3.200.200.146 port 81 (the alias IP address for the real server). The return traffic from the real server is forwarded to the SSL Services Module. The module encrypts the traffic and sends it to client.

Configuring the Allowed VLANs

These examples show how to allow VLAN 3 between the SSL Services Module and the supervisor engine:

Cisco IOS:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ssl-proxy module 8 allowed-vlan 3
Router(config)# ^Z
Router#
Router# show ssl-proxy module 8 state
SSL-proxy module 8 data-port:
 Switchport:Enabled
Administrative Mode:trunk
Operational Mode:trunk
Administrative Trunking Encapsulation:dot1q
Operational Trunking Encapsulation:dot1q
Negotiation of Trunking:Off
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Trunking VLANs Enabled:3
Pruning VLANs Enabled:2-1001
Vlans allowed on trunk:3
Vlans allowed and active in management domain:3
Vlans in spanning tree forwarding state and not pruned:
   3
Allowed-vlan :3

Router#
```

Catalyst Operating System Software:

```
Console> (enable) set trunk 8/1
Adding vlans 3 to allowed list.
Console> (enable) show trunk 8/1
* - indicates vtp domain mismatch
# - indicates dot1q-all-tagged enabled on the port
Port      Mode         Encapsulation  Status        Native vlan
--------  -----------  -------------  ------------  -----------
 8/1      nonegotiate  dot1q          not-trunking  1

Port      Vlans allowed on trunk
--------  ---------------------------------------------------------------------
 8/1      3

Port      Vlans allowed and active in management domain
--------  ---------------------------------------------------------------------
 8/1      3

Port      Vlans in spanning tree forwarding state and not pruned
--------  ---------------------------------------------------------------------
 8/1      3
```

Configuring the Access List and Route Map

This example shows how to configure the access list and route map for redirecting SSL traffic from the client to the SSL Services Module, and for redirecting clear text traffic from the real server to the SSL Services Module:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)# ip access-list extended redirect_ssl
Router(config-ext-nacl)# permit tcp any 3.0.0.0 0.255.255.255 eq 443
Router(config-ext-nacl)# !
Router(config-ext-nacl)# ip access-list extended reverse_traffic
Router(config-ext-nacl)# permit tcp 3.0.0.0 0.255.255.255 eq 81 any
Router(config-ext-nacl)# !
Router(config-ext-nacl)# route-map redirect_ssl permit
Router(config-route-map)# match ip address redirect_ssl
Router(config-route-map)# set ip next-hop 3.100.100.150
Router(config-route-map)# !
Router(config-route-map)# route-map reverse_traffic permit
Router(config-route-map)# match ip address reverse_traffic
Router(config-route-map)# set ip next-hop 3.100.100.150
Router(config-route-map)# !
Router(config-route-map)# interface Vlan2
Router(config-if)# ip address 2.100.100.100 255.0.0.0
Router(config-if)# ip policy route-map redirect_ssl
Router(config-if)# !
Router(config-if)# interface Vlan3
Router(config-if)# ip address 3.100.100.100 255.0.0.0
Router(config-if)# ip policy route-map reverse_traffic
Router(config-if)# !
Router(config-if)#^Z
Router#
```

Importing a Test Certificate

This example shows how to import the test certificate. For information on configuring a trustpoint and obtaining a certificate, see the “Configuring Keys and Certificates” section on page 3-21

```
ssl-proxy# test ssl-proxy certificate install
% Opening file, please wait ...
% Writing, please wait ...........
% Please use the following config command to import the file.
  "crypto ca import <trustpoint-name> pkcs12 nvram:test/testssl.p12 sky is blue"
% Then you can assign the trustpoint to a proxy service for testing.

*Oct  9 19:49:17.570:%STE-6-PKI_TEST_CERT_INSTALL:Test key and certificate was installed
into NVRAM in a PKCS#12 file.
ssl-proxy# configure terminal
ssl-proxy(config)# crypto ca import sample pkcs12 nvram:sky is blue
Source filename [sample]? test/testssl.p12
ssl-proxy(config)#
*Oct  9 19:51:04.674:%SSH-5-ENABLED:SSH 1.5 has been enabled
*Oct  9 19:51:04.678:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
ssl-proxy(config)# ^Z
ssl-proxy#
```

Configuring the SSL Proxy VLAN

This example shows how to add an interface to VLAN 3 on the SSL Services Module:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy vlan 3
ssl-proxy(config-vlan)# ipaddr 3.100.100.150 255.0.0.0
ssl-proxy(config-vlan)# gateway 3.100.100.100
ssl-proxy(config-vlan)# admin
ssl-proxy(config-vlan)# ^Z
ssl-proxy#
```

Configuring the SSL Proxy Service

This example shows how to add a specific proxy service that identifies a virtual IP address and a server IP address for each proxy:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy service sample
ssl-proxy(config-ssl-proxy)# virtual ipaddr 3.100.100.151 protocol tcp port 443 secondary
ssl-proxy(config-ssl-proxy)# server ipaddr 3.200.200.146 protocol tcp port 81
ssl-proxy(config-ssl-proxy)# cert rsa general-purpose trustpoint sample
ssl-proxy(config-ssl-proxy)# inservice
ssl-proxy(config-ssl-proxy)# ^Z
ssl-proxy#
```

Verifying Service and Connections

This example shows how to verify the SSL proxy service and connections:

```
ssl-proxy# show ssl-proxy service sample
Service id:3, bound_service_id:259
Virtual IP:3.100.100.151, port:443
Server IP:3.200.200.146, port:81
rsa-general-purpose certificate trustpoint:sample
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label:sample
       Serial Number:01
    Root CA Certificate:
       Serial Number:00
  Certificate chain complete
Admin Status:up
Operation Status:up
ssl-proxy#

ssl-proxy# show ssl-proxy conn
Connections for TCP module 1
Local Address       Remote Address          VLAN   Conid  Send-Q Rwind  Recv-Q State
------------------- -------------------     -----  ------ ------ ------ ------ ------
3.100.100.151.443   2.200.200.14.37820      3      470     0     32768    0    ESTABLISHED
2.200.200.14.37820  3.200.200.146.81        3      471     0     32768    0    ESTABLISHED
ssl-proxy#
```

Configuring the Content Switching Module

Note For detailed information on configuring the CSM, refer to the *Catalyst 6500 Series Switch Content Switching Module Installation and Configuration Note,* Release 3.1, at this URL:

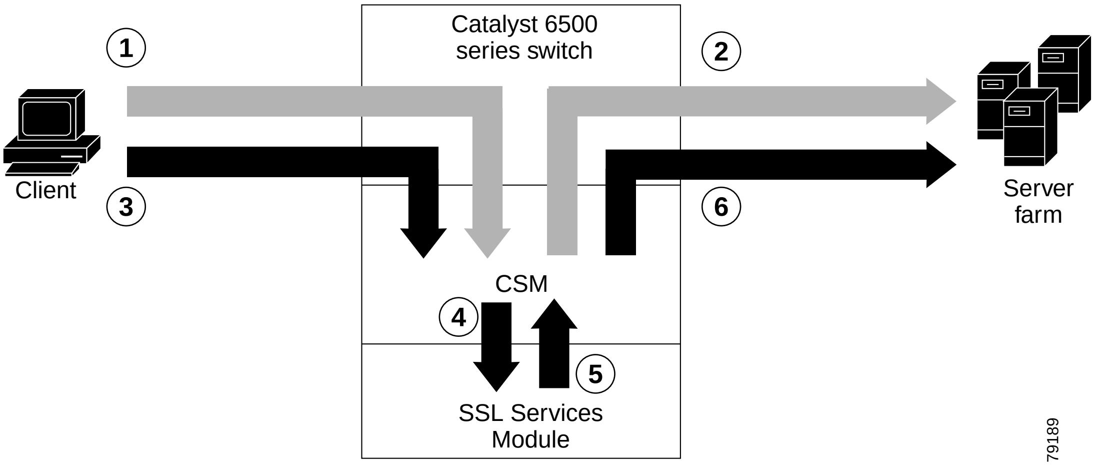
http://www.cisco.com/univercd/cc/td/doc/product/lan/cat6000/cfgnotes/csm_3_1/index.htm

The Content Switching Module (CSM) provides high-performance server load balancing (SLB) between network devices and server farms based on Layer 4 through Layer 7 packet information.

When you use the SSL Services Module with the CSM, only encrypted client traffic is forwarded to the SSL Services Module, while clear text traffic is forwarded to real servers.

The CSM parses for traffic destined to the server farm virtual IP address, port 443. The CSM forwards this traffic to the SSL Services Module without modifying the destination IP address. If there are multiple SSL Services Modules in the configuration, the CSM load balances the traffic across the SSL Services Modules. The SSL Services Module decrypts the traffic and forwards the new stream back to the CSM. The SSL Services Module does not change the destination IP address (the original server farm virtual IP address), but it does perform a port translation. With this new virtual IP address and port combination, the CSM balances the data across the servers in the server farm. (See Figure 3-5.)

Figure 3-5 Client-to-Server Traffic Flow—SSL Services Module and CSM



In Figure 3-5, clear text traffic is sent from the client to a virtual IP address, non-SSL port (for example, 80) (shown in flow 1). The CSM balances the clear text traffic across the servers in the server farm (flow 2).

Encrypted traffic is sent from the client to a virtual IP address, SSL port (443) (flow 3). The CSM forwards the encrypted traffic to the SSL Services Module (flow 4); if there is more than one SSL Services Module, the CSM balances the encrypted traffic across SSL Services Modules.

The SSL Services Module decrypts the traffic and forwards it to a virtual IP address and port on the CSM (flow 5).

The CSM balances the decrypted traffic across the servers in the server farm (flow 6).

On the return path, the CSM must monitor the port from which the server transmits data. If it is the standard clear text port (for example, 80), the data is forwarded back to the client unaltered, with the exception of the source address. If server NAT is configured on the clear text flow, the virtual IP address replaces the source IP address.

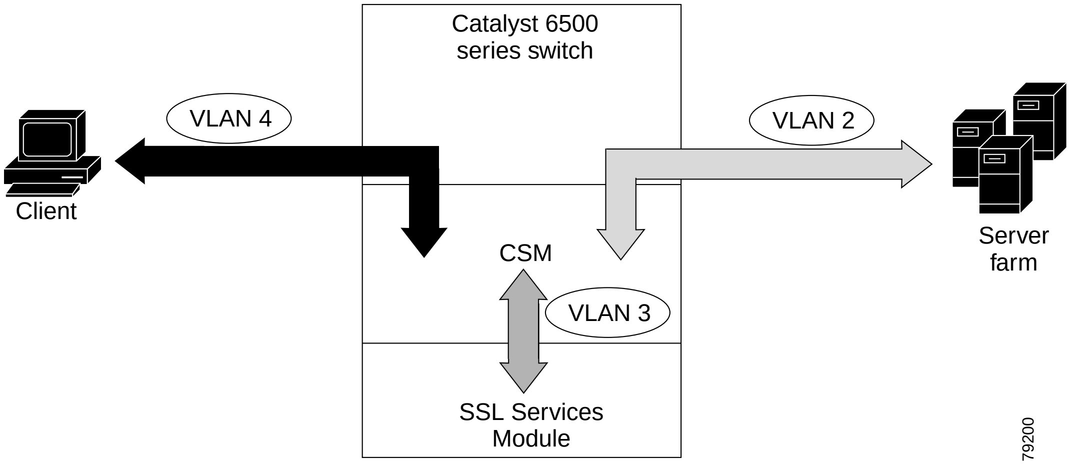
If traffic is destined to the virtual IP address and port 443, the CSM forwards this flow to the SSL Services Module. The SSL Services Module encrypts the traffic and performs port translation on the packet header. The SSL Services Module directs the traffic to the CSM with source port 443 (the SSL port to which the client originally directed encrypted traffic) so that the CSM can handle the reverse path traffic.

VLANs

As with normal CSM operation, you are required to configure separate client and server VLANs. If the CSM client and server VLANs are not on the same subnet, the CSM acts as a switch between the client and server VLANs.

To allow traffic to pass between the CSM and the SSL Services Module, you need to configure a single VLAN between them (see Figure 3-6); all flows between the CSM and the SSL Services Module are on that VLAN.

Figure 3-6 SSL Services Module with CSM—3-VLAN Configuration



In Figure 3-6, VLAN 4 involves clear text and encrypted traffic between the client and the CSM virtual IP address.

VLAN 2 involves the following types of traffic between the server and the client:

- Clear text traffic between the client and the server
- Traffic sent by the client that was decrypted by the SSL Services Module
- Traffic sent by the server that needs to be encrypted by the SSL Services Module

VLAN 3 involves the following types of traffic between the CSM and the SSL Services Module:

- Encrypted client traffic that needs to be decrypted
- Decrypted client traffic that needs to be forwarded to the server farm
- Unencrypted server traffic that needs to be encrypted
- Encrypted server traffic that needs to be forwarded back to the client

To configure VLANs on the CSM, perform this task:

	Command	Purpose
Step 1	Router(config)# **mod csm** *slot*	Specifies the slot of the CSM.
Step 2	Router(config-module-csm)# **vlan** *vlan* {**client**|**server**}	Configures the VLAN as either a client or a server on the CSM.
Step 3	Router(config-slb-vlan-client)# **ip address** *ip_addr netmask*	Configures the IP address and netmask of the interface on the VLAN.
Step 4	Router(config-slb-vlan-client)# **gateway** *ip_addr*	Configures the gateway IP address.

Server Farms

When you use the SSL Services Module with a CSM, the CSM sees two types of server farms. The first server farm is the traditional farm consisting of a group of real servers and is mapped to one or more virtual server IP addresses. You may or may not choose to allow server or client NAT to act on traffic going to these servers.

The second type of server farm consists of the SSL Services Modules that are present in the chassis. The CSM views these SSL Services Modules as real servers and balances SSL traffic across the modules.

To configure a server farm on the CSM, perform this task:

	Command	Purpose
Step 1	Router(config)# **mod csm** *slot*	Specifies the slot of the CSM.
Step 2	Router(config-module-csm)# **serverfarm** *server_farm*	Configures the name of the server farm.
Step 3	Router(config-slb-sfarm)# **no nat server**	(Optional) Disables server NAT.
Step 4	Router(config-slb-sfarm)# **nat client** *natpool_name*	(Optional) Enables client NAT.
Step 5	Router(config-slb-sfarm)# **real** *ip_addr*	Configures the real IP address of the server.
Step 6	Router(config-slb-real)# **inservice**	Puts the server farm in service.

Virtual Servers

Three types of virtual servers are required for every real server farm supported in a CSM and SSL Services Module configuration. The main distinction between the three types of virtual servers is the port number. The clear text virtual server and the SSL virtual server have the same virtual IP address. The decryption virtual server may or may not have the same virtual IP address. The three types of virtual servers are as follows:

- Clear text virtual server—The clear text virtual server is the destination for any clear text traffic sent by the client. Typically, this traffic is destined to port 80. The CSM balances traffic sent to this virtual server directly to a real server in the server farm. The SSL Services Module is uninvolved.
- SSL virtual server—The SSL virtual server should be the destination for any SSL-encrypted traffic from the client to the server. This traffic is destined to port 443. The CSM forwards this type of traffic to the SSL Services Module for decryption.

- Decryption virtual server—After the SSL Services Module decrypts SSL traffic from the client, it forwards it back to the CSM, destined for the decryption virtual server. The CSM balances the traffic to a real server in the server farm, similar to the action it took for traffic destined to the clear text virtual server. The port associated with this decryption virtual server should match the port from which the real server has been configured to expect traffic decrypted by the SSL Services Module.

To configure a virtual server on the CSM, perform this task:

	Command	Purpose
Step 1	`Router(config)# mod csm slot`	Specifies the slot of the CSM.

Step 2	`Router(config-module-csm)#`
	`vserver vserver`
		Configures the name of the virtual server.

Step 3	`Router(config-slb-vserver)#`
	`virtual ip_address tcp port`
		Configures the IP address, protocol, and port of the virtual server.

Step 4	`Router(config-slb-vserver)#`
	`serverfarm server_farm`
		Configures the destination server farm.

Step 5	`Router(config-slb-vserver)#`
	`vlan vlan`
		Specifies the VLAN from where the CSM accepts traffic for a specified virtual server.
		Note For security reasons, this command is required for the decryption virtual server.

Step 6	`Router(config-slb-vserver)#`
	`inservice`
		Puts the virtual server in service.

Sticky Connections

Note Configuring the SSL sticky feature requires CSM software release 3.1(1a) or later releases on the CSM.

If a CSM and SSL Services Module configuration consists of multiple SSL Services Modules connected to a single CSM, configure the SSL sticky feature on the CSM to ensure that the CSM always forwards traffic from a particular client to the same SSL Services Module.

A 32-byte SSL session ID is created for each connection between a client and an SSL Services Module. With the SSL sticky feature configured, the CSM looks at a specific portion of the SSL session ID (the MAC address of the SSL Services Module) and load balances SSL traffic among the SSL Services Modules.

Note The MAC address of the SSL Services Module is always located at bytes 21 though 26 of the SSL session ID, even when the session ID is renegotiated.

To configure a sticky connection on the CSM, perform this task:

	Command	Purpose
Step 1	`Router(config)# mod csm mod`	Specifies the slot of the CSM.

Step 2	`Router(config-module-csm)#`
	`sticky group ssl`
		Configures the sticky group ID.

	Command	Purpose

Step 3	`Router(config-module-csm)#`
	`vserver` *`vserver`*
		Associates the group ID with the virtual server.

Step 4	`Router(config-slb-vserver)#`
	`sticky` *`group`* **`timeout`** *`time`*
		Specifies the amount of time, in minutes, that the connection remains sticky.

Step 5	`Router(config-slb-vserver)#`
	`ssl-sticky offset 20 length 6`
		Specifies the location of the SSL Services Module MAC address in the SSL ID.

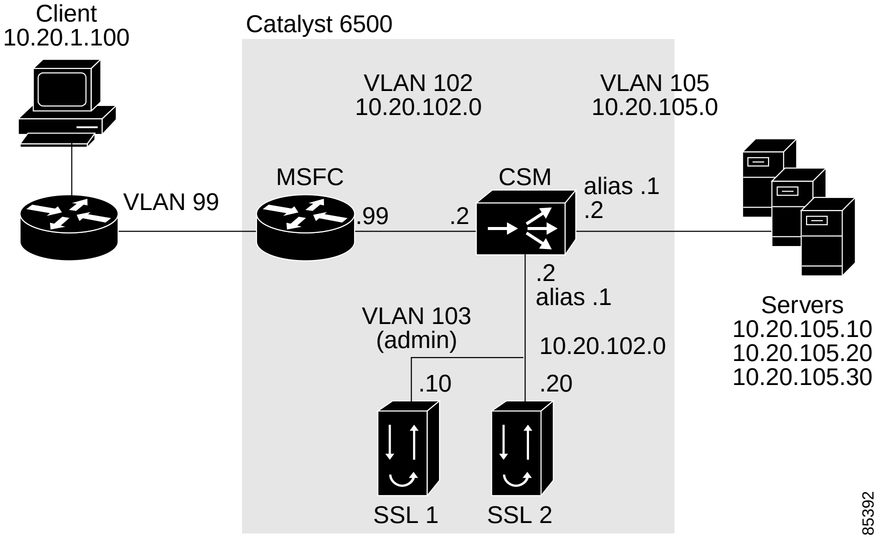
CSM and SSL Services Module Configuration Example (Bridge Mode, No NAT)

This section describes a CSM and SSL Services Module configuration that contains two SSL Services Modules, a CSM, a client network, and a server farm that has three web servers (IP addresses 10.20.105.10, 10.20.105.20, 10.20.105.30).

In this example, the CSM client VLAN and CSM server VLAN for the SSL Services Modules are configured in the same IP subnet (bridge mode), while the CSM server VLAN for the web servers is in a separate IP subnet. (See Figure 3-7.)

The CSM is configured to not perform NAT operations when it is load balancing encrypted traffic to the SSL Services Modules. The SSL Services Modules are also configured to not perform NAT operations when they are sending decrypted traffic back to the CSM. The CSM is then configured to perform NAT for the decrypted traffic to the selected destination server.

Figure 3-7 Bridge Mode, No NAT Configuration Example



CSM Virtual Servers:

- Client clear text traffic—10.20.102.100:80
- Client SSL traffic—10.20.102.100:443
- Decrypted traffic from SSL Services Modules—10.20.102.100:80

SSL Virtual Server:

- 10.20.103.100:443 secondary

Figure 3-7 shows VLAN 102 and VLAN 103 in the same subnet, and VLAN 105 in a separate subnet.

Add all the required VLANs to the VLAN database, and configure the IP interface for VLAN 102 on the MSFC. Configure VLANs 102, 103 and 105 on the CSM. See the "Preparing to Configure the SSL Services Module" section on page 3-1 for information on how to configure VLANs and IP interfaces.

Note While VLAN 102 exists as Layer 3 interface on the MSFC, both VLAN 103 and VLAN 105 exist only as VLANs in the VLAN database and as CSM VLANs, but do not have a corresponding Layer 3 interface on the MSFC.

This example shows how to create the client and server VLANs on the CSM installed in slot number 5:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# vlan 102 client
Router(config-slb-vlan-client)# ip address 10.20.102.2 255.255.255.0
Router(config-slb-vlan-client)# gateway 10.20.102.99
Router(config-slb-vlan-client)# exit
Router(config-module-csm)# vlan 103 server
Router(config-slb-vlan-server)# ip address 10.20.102.2 255.255.255.0
Router(config-slb-vlan-server)# alias 10.20.102.1 255.255.255.0
Router(config-slb-vlan-server)# exit
Router(config-module-csm)# vlan 105 server
Router(config-slb-vlan-server)# ip address 10.20.105.2 255.255.255.0
Router(config-slb-vlan-server)# alias 10.20.105.1 255.255.255.0
Router(config-slb-vlan-server)# end
```

This example shows how to allow VLAN 103 between the SSL Services Module and the CSM:

Cisco IOS:

```
Router(config)# ssl-proxy module 4 allowed-vlan 103
```

Catalyst Operating System Software:

```
Console> (enable) set trunk 4/1 103
```

This example shows how to create the server farm of web servers (configured with server NAT) and the server farm of SSL Services Modules (configured with no server NAT):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# serverfarm SSLFARM
Router(config-slb-sfarm)# no nat server
Router(config-slb-sfarm)# real 10.20.102.10
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.102.20
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-module-csm)# serverfarm WEBSERVERS
Router(config-slb-sfarm)# nat server
Router(config-slb-sfarm)# real 10.20.105.10
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.105.20
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.105.30
Router(config-slb-real)# inservice
Router(config-slb-real)# end
```

This example shows how to configure the three virtual servers. In this example, the web servers are only receiving traffic to port 80, either directly from the clients or as decrypted traffic from the SSL Services Modules (since no port translation is configured).

The CSM distinguishes between requests received directly from the clients and requests received from the SSL Services Modules based on the VLAN from where the connections are received.

A sticky group is also configured to maintain stickiness based on the SSL ID.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# sticky 100 ssl timeout 30
Router(config-module-csm)# vserver CLEAR_VIP
Router(config-slb-vserver)# virtual 10.20.102.100 tcp www
Router(config-slb-vserver)# vlan 102
Router(config-slb-vserver)# serverfarm WEBSERVERS
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
Router(config-module-csm)# vserver DECRYPT_VIP
Router(config-slb-vserver)# virtual 10.20.102.100 tcp www
Router(config-slb-vserver)# vlan 103
Router(config-slb-vserver)# serverfarm WEBSERVERS
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
Router(config-module-csm)# vserver SSL_VIP
Router(config-slb-vserver)# virtual 10.20.102.100 tcp https
Router(config-slb-vserver)# vlan 102
Router(config-slb-vserver)# serverfarm SSLFARM
Router(config-slb-vserver)# sticky 30 group 100
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# end
```

This example shows how to configure the SSL Services Module to communicate with the CSM over VLAN 103, the admin VLAN:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy vlan 103
ssl-proxy(config-vlan)# ipaddr 10.20.102.10 255.255.255.0
ssl-proxy(config-vlan)# gateway 10.20.102.99
ssl-proxy(config-vlan)# admin
ssl-proxy(config-vlan)# end
```

To complete the configuration, enter the **ssl-proxy service** command to create a new service on the SSL Services Module (**test1**). This example shows how to configure a virtual IP address that matches the virtual server created on the CSM (this virtual IP address is configured as **secondary** so that the SSL Services Module does not reply to ARP requests for this IP address). The service is configured to send decrypted traffic back to the CSM without performing NAT.

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy service test1
ssl-proxy(config-ssl-proxy)# virtual ipaddr 10.20.102.100 protocol tcp port 443 secondary
ssl-proxy(config-ssl-proxy)# server ipaddr 10.20.102.1 protocol tcp port 80
ssl-proxy(config-ssl-proxy)# certificate rsa general-purpose trustpoint testtp
ssl-proxy(config-ssl-proxy)# no nat server
ssl-proxy(config-ssl-proxy)# inservice
ssl-proxy(config-ssl-proxy)# end
```

The following examples show the output of the various **show** commands on the MSFC and CSM:

```
Router# show module csm 5 vlan detail
vlan   IP address       IP mask          type
---------------------------------------------------
102    10.20.102.2      255.255.255.0    CLIENT
  GATEWAYS
  10.20.102.99
103    10.20.102.2      255.255.255.0    SERVER
  ALIASES
  IP address       IP mask
  --------------------------------
  10.20.102.1      255.255.255.0
105    10.20.105.2      255.255.255.0    SERVER
  ALIASES
  IP address       IP mask
  --------------------------------
  10.20.105.1      255.255.255.0

Router# show module csm 5 vserver detail
SSL_VIP, type = SLB, state = OPERATIONAL, v_index = 13
  virtual = 10.20.102.100/32:443, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 102, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 2
  Default policy:
    server farm = SSLFARM, backup = <not assigned>
    sticky: timer = 30, subnet = 0.0.0.0, group id = 100
  Policy          Tot Conn     Client pkts  Server pkts
  -------------------------------------------------------
  (default)       2            22           15

CLEAR_VIP, type = SLB, state = OPERATIONAL, v_index = 14
  virtual = 10.20.102.100/32:80, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 102, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 0
  Default policy:
    server farm = WEBSERVERS, backup = <not assigned>
    sticky: timer = 0, subnet = 0.0.0.0, group id = 0
  Policy          Tot Conn     Client pkts  Server pkts
  -------------------------------------------------------
  (default)       0            0            0

DECRYPT_VIP, type = SLB, state = OPERATIONAL, v_index = 15
  virtual = 10.20.102.100/32:80, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 103, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 2
  Default policy:
    server farm = WEBSERVERS, backup = <not assigned>
    sticky: timer = 0, subnet = 0.0.0.0, group id = 0
  Policy          Tot Conn     Client pkts  Server pkts
  -------------------------------------------------------
  (default)       2            11           7
```

The following examples show the output of the various **show** commands on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service test1
Service id: 0, bound_service_id: 256
Virtual IP: 10.20.102.100, port: 443 (secondary configured)
Server IP: 10.20.102.1, port: 80
rsa-general-purpose certificate trustpoint: testtp
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label: testtp
       Serial Number: 01
    Root CA Certificate:
      Serial Number: 00
  Certificate chain complete
Admin Status: up
Operation Status: up
ssl-proxy#
ssl-proxy# show ssl-proxy stats
TCP Statistics:
    Conns initiated   : 2             Conns accepted   : 2
    Conns established : 4             Conns dropped    : 4
    Conns closed      : 4             SYN timeouts     : 0
    Idle timeouts     : 0             Total pkts sent  : 26
    Data packets sent : 15            Data bytes sent  : 8177
    Total Pkts rcvd   : 27            Pkts rcvd in seq : 11
    Bytes rcvd in seq : 5142

SSL stats:
    conns attempted     : 2             conns completed     : 2
    full handshakes     : 2             resumed handshakes  : 0
    active conns        : 0             active sessions     : 0
    renegs attempted    : 0             conns in reneg      : 0
    handshake failures  : 0             data failures       : 0
    fatal alerts rcvd   : 0             fatal alerts sent   : 0
    no-cipher alerts    : 0             ver mismatch alerts : 0
    no-compress alerts  : 0             bad macs received   : 0
    pad errors          : 0

FDU Statistics
    IP Frag Drops     :  0               Serv_Id Drops     :  0
    Conn Id Drops     :  0               Checksum Drops    :  0
    IOS Congest Drops :  0               IP Version Drops  :  0
    Hash Full Drops   :  0               Hash Alloc Fails  :  0
    Flow Creates      :  4               Flow Deletes      :  4
    conn_id allocs    :  4               conn_id deallocs  :  4
    Tagged Drops      :  0               Non-Tagged Drops  :  0
    Add ipcs          :  0               Delete ipcs       :  0
    Disable ipcs      :  0               Enable ipcs       :  0
    Unsolicited ipcs  :  0               Duplicate ADD ipcs:  0
ssl-proxy#
```

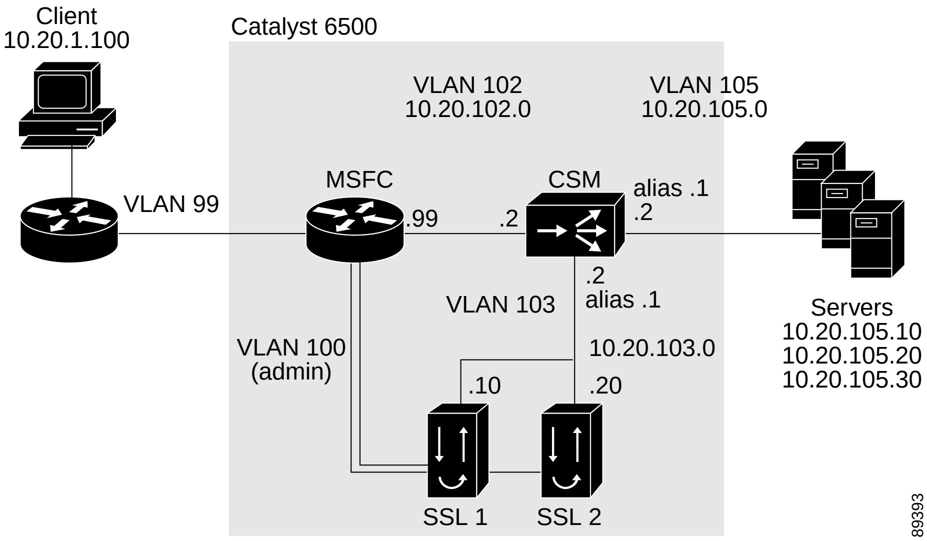
CSM and SSL Services Module Configuration Example (Router Mode, Server NAT)

This section describes a CSM and SSL Services Module configuration that contains two SSL Services Modules, a CSM, a client network, and a server farm that has three web servers (IP addresses 10.20.105.10, 10.20.105.20, 10.20.105.30).

In this example, the three CSM VLANs (client VLAN, server VLAN for the SSL Services Modules, and server VLAN for the web servers) are configured in distinct IP subnets (router mode). (See Figure 3-8.)

The CSM is configured to perform server NAT operations when it is load balancing the encrypted traffic to the SSL Services Modules. The SSL Services Modules are also configured to perform server NAT operations when they are sending decrypted traffic back to the CSM. The CSM is then configured to perform NAT on the decrypted traffic to the selected destination server.

Figure 3-8 Configuration Example—Router Mode, Server NAT



CSM Virtual Servers:

- Client clear text traffic—10.20.102.100:80
- Client SSL traffic—10.20.102.100:443
- Decrypted traffic from SSL Services Modules—10.20.103.100:80

SSL Virtual Servers:

- 10.20.103.110:443
- 10.20.103.120:443

In Figure 3-8, VLAN 102, VLAN 103 and VLAN 105 are in separate subnets. VLAN 100 (admin) is set up as a separate VLAN for management purposes.

Add all the required VLANs to the VLAN database, and configure the IP interfaces for VLAN 100 and VLAN 102 on the MSFC. Configure VLANs 102, 103, and 105 on the CSM. See the "Preparing to Configure the SSL Services Module" section on page 3-1 for information on how to configure VLANs and IP interfaces.

Note While VLAN 100 and VLAN 102 exist as Layer 3 interfaces on the MSFC, both VLAN 103 and VLAN 105 exist only as VLANs in the VLAN database and as CSM VLANs, but do not have a corresponding Layer 3 interface on the MSFC.

This example shows how to create the client and server VLANs on the CSM installed in slot number 5:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# vlan 102 client
Router(config-slb-vlan-client)# ip address 10.20.102.2 255.255.255.0
Router(config-slb-vlan-client)# alias 10.20.102.1 255.255.255.0
Router(config-slb-vlan-client)# gateway 10.20.102.99
Router(config-slb-vlan-client)# exit
Router(config-module-csm)# vlan 103 server
Router(config-slb-vlan-server)# ip address 10.20.103.2 255.255.255.0
Router(config-slb-vlan-server)# alias 10.20.103.1 255.255.255.0
Router(config-slb-vlan-server)# exit
Router(config-module-csm)# vlan 105 server
Router(config-slb-vlan-server)# ip address 10.20.105.2 255.255.255.0
Router(config-slb-vlan-server)# alias 10.20.105.1 255.255.255.0
Router(config-slb-vlan-server)# end
```

This example shows how to allow VLAN 103 (client VLAN) between the SSL Services Module and the CSM, and VLAN 100 (admin VLAN) between the SSL Services Module and the MSFC:

Cisco IOS

```
Router(config)# ssl-proxy module 4 allowed-vlan 100,103
```

Catalyst Operating System Software

```
Console> (enable) set trunk 4/1 100,103
```

This example shows how to create the server farm of web servers (configured with server NAT) and the server farm of SSL Services Modules (configured with server NAT):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# serverfarm SSLFARM
Router(config-slb-sfarm)# nat server
Router(config-slb-sfarm)# real 10.20.103.110
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.103.120
Router(config-slb-real)# inservice
Router(config-slb-real)# exit
Router(config-slb-sfarm)# exit
Router(config-module-csm)# serverfarm WEBSERVERS
Router(config-slb-sfarm)# nat server
Router(config-slb-sfarm)# real 10.20.105.10
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.105.20
Router(config-slb-real)# inservice
Router(config-slb-real)# real 10.20.105.30
Router(config-slb-real)# inservice
Router(config-slb-real)# end
```

This example shows how to configure the three virtual servers. In this example, the web servers receive requests to port 80 directly from the clients, and decrypted requests to port 81 from the SSL Services Modules (since IP and port translation are configured).

This example also shows how to configure a sticky group to maintain stickiness based on the SSL ID.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# module csm 5
Router(config-module-csm)# sticky 100 ssl timeout 30
Router(config-module-csm)# vserver CLEAR_VIP
Router(config-slb-vserver)# virtual 10.20.102.100 tcp www
Router(config-slb-vserver)# vlan 102
Router(config-slb-vserver)# serverfarm WEBSERVERS
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
Router(config-module-csm)# vserver DECRYPT_VIP
Router(config-slb-vserver)# virtual 10.20.103.100 tcp 81
Router(config-slb-vserver)# vlan 103
Router(config-slb-vserver)# serverfarm WEBSERVERS
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# exit
Router(config-module-csm)# vserver SSL_VIP
Router(config-slb-vserver)# virtual 10.20.102.100 tcp https
Router(config-slb-vserver)# vlan 102
Router(config-slb-vserver)# serverfarm SSLFARM
Router(config-slb-vserver)# sticky 30 group 100
Router(config-slb-vserver)# inservice
Router(config-slb-vserver)# end
```

This example shows how to configure the SSL Services Module to communicate with the CSM over VLAN 103 and to communicate with the MSFC over VLAN 100 (admin VLAN):

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy vlan 103
ssl-proxy(config-vlan)# ipaddr 10.20.103.10 255.255.255.0
ssl-proxy(config-vlan)# gateway 10.20.103.1
ssl-proxy(config-vlan)# exit
ssl-proxy(config)# ssl-proxy vlan 100
ssl-proxy(config-vlan)# ipaddr 10.20.100.10 255.255.255.0
ssl-proxy(config-vlan)# gateway 10.20.100.99
ssl-proxy(config-vlan)# admin
ssl-proxy(config-vlan)# end
```

To complete the configuration, enter the **ssl-proxy service** command to create a new service on the SSL Services Module (**test1**). This example shows how to configure a virtual IP address, which acts as a real server for the CSM (since this virtual IP address is required to reply to ARP, the **secondary** keyword is not entered). The service is configured to send decrypted traffic back to the CSM and to perform NAT on both the destination IP address and the port:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy service test1
ssl-proxy(config-ssl-proxy)# virtual ipaddr 10.20.103.110 protocol tcp port 443
ssl-proxy(config-ssl-proxy)# server ipaddr 10.20.102.100 protocol tcp port 81
ssl-proxy(config-ssl-proxy)# certificate rsa general-purpose trustpoint testtp
ssl-proxy(config-ssl-proxy)# nat server
ssl-proxy(config-ssl-proxy)# inservice
ssl-proxy(config-ssl-proxy)# end
```

The following examples show the output of the various **show** commands on the MSFC and CSM:

```
Router# show mod csm 5 vlan deta
vlan   IP address       IP mask          type
---------------------------------------------------
102    10.20.102.2      255.255.255.0    CLIENT
  GATEWAYS
  10.20.102.99
  ALIASES
  IP address       IP mask
  --------------------------------
  10.20.102.1      255.255.255.0
103    10.20.103.2      255.255.255.0    SERVER
  ALIASES
  IP address       IP mask
  --------------------------------
  10.20.103.1      255.255.255.0
105    10.20.105.2      255.255.255.0    SERVER
  ALIASES
  IP address       IP mask
  --------------------------------
  10.20.105.1      255.255.255.0

Router# show mod csm 5 vser deta
CLEAR_VIP, type = SLB, state = OPERATIONAL, v_index = 10
  virtual = 10.20.102.100/32:80, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 102, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 1
  Default policy:
    server farm = WEBSERVERS, backup = <not assigned>
    sticky: timer = 0, subnet = 0.0.0.0, group id = 0
  Policy          Tot Conn     Client pkts  Server pkts
  -----------------------------------------------------
  (default)       1            6            4

DECRYPT_VIP, type = SLB, state = OPERATIONAL, v_index = 11
  virtual = 10.20.103.100/32:81, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 103, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 2
  Default policy:
    server farm = WEBSERVERS, backup = <not assigned>
    sticky: timer = 0, subnet = 0.0.0.0, group id = 0
  Policy          Tot Conn     Client pkts  Server pkts
  -----------------------------------------------------
  (default)       2            11           7

SSL_VIP, type = SLB, state = OPERATIONAL, v_index = 13
  virtual = 10.20.102.100/32:443, TCP, service = NONE, advertise = FALSE
  idle = 3600, replicate csrp = none, vlan = 102, pending = 30
  max parse len = 600, persist rebalance = TRUE
  conns = 0, total conns = 2
  Default policy:
    server farm = SSLFARM, backup = <not assigned>
    sticky: timer = 30, subnet = 0.0.0.0, group id = 100
  Policy          Tot Conn     Client pkts  Server pkts
  -----------------------------------------------------
  (default)       2            21           15
```

The following examples show the output of the various **show** commands on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service test1
Service id: 0, bound_service_id: 256
Virtual IP: 10.20.103.110, port: 443
Server IP: 10.20.103.100, port: 81
rsa-general-purpose certificate trustpoint: testtp
  Certificate chain in use for new connections:
    Server Certificate:
       Key Label: testtp
       Serial Number: 01
    Root CA Certificate:
      Serial Number: 00
  Certificate chain complete
Admin Status: up
Operation Status: up
ssl-proxy#

ssl-proxy# show ssl-proxy stats
TCP Statistics:
    Conns initiated   : 2             Conns accepted   : 2
    Conns established : 4             Conns dropped    : 4
    Conns closed      : 4             SYN timeouts     : 0
    Idle timeouts     : 0             Total pkts sent  : 26
    Data packets sent : 15            Data bytes sent  : 8212
    Total Pkts rcvd   : 26            Pkts rcvd in seq : 11
    Bytes rcvd in seq : 5177

SSL stats:
    conns attempted     : 2             conns completed     : 2
    full handshakes     : 2             resumed handshakes  : 0
    active conns        : 0             active sessions     : 0
    renegs attempted    : 0             conns in reneg      : 0
    handshake failures  : 0             data failures       : 0
    fatal alerts rcvd   : 0             fatal alerts sent   : 0
    no-cipher alerts    : 0             ver mismatch alerts : 0
    no-compress alerts  : 0             bad macs received   : 0
    pad errors          : 0

FDU Statistics
    IP Frag Drops     :  0              Serv_Id Drops     :  0
    Conn Id Drops     :  0              Checksum Drops    :  0
    IOS Congest Drops :  0              IP Version Drops  :  0
    Hash Full Drops   :  0              Hash Alloc Fails  :  0
    Flow Creates      :  4              Flow Deletes      :  4
    conn_id allocs    :  4              conn_id deallocs  :  4
    Tagged Drops      :  0              Non-Tagged Drops  :  0
    Add ipcs          :  0              Delete ipcs       :  0
    Disable ipcs      :  0              Enable ipcs       :  0
    Unsolicited ipcs  :  0              Duplicate ADD ipcs:  0
```

CHAPTER A

Testing SSL Proxy Services

You can test or troubleshoot SSL proxy services by doing one of the following:

- Generating a Self-Signed Certificate, page A-1
- Importing the Embedded Test Certificate, page A-4

Generating a Self-Signed Certificate

You can generate multiple self-signed certificates for testing SSL proxy services, specifying the key label and the subject name, by entering the **test crypto pki self** command. You need to generate a key pair with a label before you generate the self-signed certificate. See the "Generating RSA Key Pairs" section on page 3-23 for details on generating key pairs.

After you enter the **test crypto pki self** command, you are prompted for the key pair label and the subject name of the certificate. A trustpoint with the key pair label as the trustpoint name is automatically created, and the hexadecimal dump of the self-signed certificate is displayed on the console. You can then assign the trustpoint to a proxy service for testing. You can repeat the procedure after reboot if necessary. The certificate is stored only in memory and cannot be saved in NVRAM as part of the configuration.

Note You cannot save the self-signed certificates as part of the configuration.

Note The **show crypto ca certificate** command does not display the self-signed certificates.

To generate a self-signed certificate and assign a trustpoint to the proxy service, perform this task:

	Command	Purpose
Step 1	`ssl-proxy# test crypto pki self`	Generates a self-signed certificate. **Note** After you enter this command, you are prompted for the subject name (using the LDAP format) and key pair name.
Step 2	`ssl-proxy# show crypto ca trustpoint label`	Displays information for the trustpoint.
Step 3	`ssl-proxy# configure terminal`	Enters configuration mode, selecting the terminal option.

	Command	Purpose
Step 4	ssl-proxy(config)# **ssl-proxy service** *proxy_name*	Defines the name of the SSL proxy service. **Note** The *proxy-name* is case-sensitive.
Step 5	ssl-proxy(config-ssl-proxy)# **certificate rsa general-purpose trustpoint** *trustpoint_label*	Assigns a trustpoint to a proxy service.
Step 6	ssl-proxy(config-ssl-proxy)# **end**	Exits configuration mode.
Step 7	ssl-proxy# **show ssl-proxy service** *proxy-name*	Displays the key pair and the serial number of the certificate chain used for a specified proxy service. **Note** The *proxy-name* is case-sensitive.

Note If the trustpoint already exists, it might be replaced by the test certificate. We recommend that you generate a unique key pair for the test certificate.

This example shows how to generate a key pair, generate a self-signed certificate, and assign the certificate to a proxy service:

```
ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# crypto key generate rsa general-keys label k1 modulus 1024
The name for the keys will be:k1

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys ...[OK]

ssl-proxy(config)# end
ssl-proxy#
*Mar 20 14:34:01.543:%SYS-5-CONFIG_I:Configured from console by console
ssl-proxy#
ssl-proxy# test crypto pki self
Enter subject name for certificate
CN=testhost.my.com, O=lab, OU=testgroup
Enter name of key to be used
k1
ssl-proxy#
     30 82 02 06 30 82 01 6F 02 20 45 32 30 38 39 32
     45 37 38 31 42 41 46 45 45 45 44 45 37 37 41 36
     43 41 44 37 44 43 45 38 34 37 30 0D 06 09 2A 86
     48 86 F7 0D 01 01 04 05 00 30 3C 31 12 30 10 06
     03 55 04 0B 13 09 74 65 73 74 67 72 6F 75 70 31
     0C 30 0A 06 03 55 04 0A 13 03 6C 61 62 31 18 30
     16 06 03 55 04 03 13 0F 74 65 73 74 68 6F 73 74
     2E 6D 79 2E 63 6F 6D 30 1E 17 0D 30 33 30 33 32
     30 31 34 33 35 30 30 5A 17 0D 31 33 30 33 31 37
     31 34 33 35 30 30 5A 30 3C 31 12 30 10 06 03 55
     04 0B 13 09 74 65 73 74 67 72 6F 75 70 31 0C 30
     0A 06 03 55 04 0A 13 03 6C 61 62 31 18 30 16 06
     03 55 04 03 13 0F 74 65 73 74 68 6F 73 74 2E 6D
     79 2E 63 6F 6D 30 81 9F 30 0D 06 09 2A 86 48 86
     F7 0D 01 01 01 05 00 03 81 8D 00 30 81 89 02 81
     81 00 EC 21 35 B5 0E BF 9C 1C 71 05 05 B2 8A 47
     C8 F9 13 6C 5A 14 77 63 BD 0C B7 D3 35 6A DB B8
     0F C2 D2 39 A8 62 67 EE CB BC 8D 5E F8 C2 1E 8E
     D6 39 62 07 B4 64 20 D8 29 25 1E 9E 06 C8 F8 F9
     A6 29 05 19 CC D9 00 E9 2D 96 6D CE CA E0 D7 BF
     DC 9D 1B 7E 71 C1 D7 3F 25 28 41 5A F9 FB 98 66
     B9 A7 81 18 79 71 2A AC 55 F8 CC A4 4A 90 35 A7
     E9 BD 79 66 BC 5B C5 98 16 B0 63 5B D3 6E 85 65
     42 1B 02 03 01 00 01 30 0D 06 09 2A 86 48 86 F7
     0D 01 01 04 05 00 03 81 81 00 A3 93 7A E6 60 54
     8C 3A FF 6A 72 A8 1F 4B AD 79 53 C4 37 DF C4 D4
     F9 F4 58 3C E4 D8 BE FF BB C5 F9 CD B0 20 7F 3D
     0E B5 11 8E FA 33 02 9E 5E 52 36 4D 0F AB 21 41
     97 A4 2D 94 4D DF D2 A0 B4 DE B0 2E 1C BA 16 A9
     4C 28 34 72 8E D5 82 F6 B6 B2 D6 4E B5 1A F0 BB
     6B 65 E7 85 52 72 9F 9C BC A7 D9 B4 79 AB 6B C2
     DC FD AD 02 D3 28 87 CD 06 8B 11 3C 22 85 28 1B
     DC 04 05 8D 4F 1D 07 8D D0 BC

ssl-proxy# show crypto ca trustpoint k1
Trustpoint k1:
    Subject Name:
    CN = testhost.my.com
     O = lab
     OU = testgroup
          Serial Number:4532303839324537383142414645454544453737413643414437444345383437
    Application generated trust point

ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# ssl-proxy service ser1
ssl-proxy(config-ssl-proxy)# certificate rsa general trustpoint k1
ssl-proxy(config-ssl-proxy)#
*Mar 20 14:36:09.567:%STE-6-PKI_SERVER_CERT_INSTALL:Proxy:ser1, Trustpoint:k1, Key:k1,
Serial#:4532303839324537383142414645454544453737413643414437444345383437, Index:3
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#
*Mar 20 14:36:16.363:%SYS-5-CONFIG_I:Configured from console by console
ssl-proxy#
ssl-proxy# show ssl-proxy service ser1
Service id:2, bound_service_id:258
Virtual IP address not configured
Server IP address not configured
rsa-general-purpose certificate trustpoint:k1
  Certificate chain for new connections:
    Server Certificate:
       Key Label:k1
       Serial Number:4532303839324537383142414645454544453737413643414437444345383437
       Self-signed
Certificate chain complete
Admin Status:down
Operation Status:down
ssl-proxy#
```

Importing the Embedded Test Certificate

A test PKCS12 file (test/testssl.p12) is embedded in the SSL software on the module. You can install the file into NVRAM for testing purposes and for proof of concept. After the PKCS12 file is installed, you can import it to a trustpoint, and then assign it to a proxy service that is configured for testing.

To install and import the test file, perform this task:

	Command	Purpose
Step 1	ssl-proxy# **test ssl-proxy certificate install**	Installs the test PKCS12 file to NVRAM.
Step 2	ssl-proxy(config)# **crypto ca import** *trustpoint_label* **pkcs12 nvram:test/testssl.p12** *passphrase*	Imports the test PKCS12 file to the module. **Note** For the test certificate, the *passphrase* is **sky is blue**.
Step 3	ssl-proxy(config)# **ssl-proxy service** *test_service*	Defines the name of the test proxy service.
Step 4	ssl-proxy(config-ssl-proxy)# **certificate rsa general-purpose trustpoint** *trustpoint_label*	Applies a trustpoint configuration to the proxy server.
Step 5	ssl-proxy# **show ssl-proxy stats** *test_service*	Displays test statistics information.

This example shows how to import the test PKCS12 file:

```
ssl-proxy# test ssl-proxy certificate install
% Opening file, please wait ...
% Writing, please wait ...........
% Please use the following config command to import the file.
  "crypto ca import <trustpoint-name> pkcs12 nvram:test/testssl.p12 sky is blue"
% Then you can assign the trustpoint to a proxy service for testing.

ssl-proxy# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ssl-proxy(config)# crypto ca import test-tp pkcs12 nvram:test/testssl.p12 sky is blue
Source filename [test/testssl.p12]?
ssl-proxy(config)#
ssl-proxy(config)# ssl-proxy service test-service
ssl-proxy(config-ssl-proxy)# certificate rsa general-purpose trustpoint test-tp
ssl-proxy(config-ssl-proxy)# end
ssl-proxy#
```

Command Reference

This appendix describes the SSL Services Module commands.

Table B-1 provides a brief description of the commands contained in this appendix.

Table B-1 Command Descriptions

Command	Description
clear ssl-proxy conn	Clears the SSL connections.
clear ssl-proxy session	Resets the statistics counters maintained in different SSL Services Module system components.
clear ssl-proxy stats	Resets the statistics counters maintained in different SSL Services Module system components.
crypto ca export pem	Exports privacy-enhanced mail (PEM) files from the SSL Services Module.
crypto ca import pem	Imports a PEM file to the SSL Services Module.
crypto ca export pkcs12	Exports a PKCS12 file from the SSL Services Module.
crypto ca import pkcs12	Imports a PKCS12 file to the SSL Services Module.
crypto key export rsa pem	Exports a PEM-formatted RSA key from the SSL Services Module.
crypto key import rsa pem	Imports a PEM-formatted RSA key to the SSL Services Module.
debug ssl-proxy	Turns on the debug flags in different system components.
show ssl-proxy admin-info	Displays the administration VLAN and related IP and gateway addresses.
show ssl-proxy buffers	Displays the TCP buffer usage information.
show ssl-proxy certificate-history	Displays the certificate event history information.
show ssl-proxy conn	Displays the TCP connections from the SSL Services Module.
show ssl-proxy crash-info	Displays the crash information.
show ssl-proxy mac address	Displays the current MAC address.
show ssl-proxy natpool	Displays NAT pool information.
show ssl-proxy policy	Displays the configured SSL or TCP policies.
show ssl-proxy service	Displays the configured SSL virtual server information.

Table B-1 Command Descriptions (continued)

Command	Description
show ssl-proxy stats	Displays statistics counter information.
show ssl-proxy status	Displays status information.
show ssl-proxy version	Displays the current image version.
show ssl-proxy vlan	Displays VLAN information.
ssl-proxy crypto selftest	Initiates a cryptographic self-test.
ssl-proxy mac address	Configures a MAC address.
ssl-proxy natpool	Defines a pool of IP addresses that the SSL module uses for implementing the client NAT.
ssl-proxy pki history	Enables the public key infrastructure (PKI) event history option.
ssl-proxy policy ssl	Enters the SSL-policy configuration submode where you can define the SSL of a TCP policy for one or more SSL proxy services.
ssl-proxy policy tcp	Enters the proxy-policy TCP configuration submode where you can define the TCP policy templates.
ssl-proxy service	Enters the proxy-service configuration submode where you can configure the virtual IP address and port associated with the proxy service and the associated target IP address and port. You can also define TCP and SSL policies for both the client side and the server side of the proxy.
ssl-proxy ssl ratelimit	Prohibits new connections during overload conditions.
ssl-proxy vlan	Enters the proxy VLAN configuration submode where you can configure a VLAN for the SSL Services Module.

Table B-2 lists the modes and submode commands.

Table B-2 Commands and Submode Commands

Commands	Submode Commands
ssl-proxy policy ssl	**cipher** {**rsa-with-3des-ede-cbc-sha** | **rsa-with-des-cbc-sha** | **rsa-with-rc4-128-md5** | **rsa-with-rc4-128-sha** | **all**}
	[**no**] **close-protocol**
	default {**cipher** | **close-protocol** | **session-cache** | **version**}
	exit
	help
	[**no**] **session-cache**
	[**no**] **session-cache size** *size*
	[**no**] **timeout handshake** *time*
	[**no**] **timeout session** *time* [**absolute**]
	version {**all** | **ssl3** | **tls1**}

Table B-2 Commands and Submode Commands (continued)

Commands	Submode Commands
ssl-proxy policy tcp	**exit**
	[**no**] **timeout fin-wait** *timeout-in-seconds*
	help
	[**no**] **timeout inactivity** *timeout-in-seconds*
	[**no**] **buffer-share rx** *buffer-limit-in-bytes*
	[**no**] **buffer-share tx** *buffer-limit-in-bytes*
	[**no**] **mss** *max-segment-size-in-bytes*
	[**no**] **timeout syn** *timeout-in-seconds*
	[**no**] **timeout reassembly** *time-in-seconds*
ssl-proxy service	**certificate rsa general-purpose trustpoint** *trustpoint-name*
	default {**nat**}
	exit
	help
	inservice
	nat {**server | client** *natpool-name*}
	server ipaddr *ip-addr* **protocol** *protocol* **port** *portno*
	server policy tcp *server-side-tcp-policy-name*
	virtual {**ipaddr** *ip-addr*} {**protocol** *protocol*} {**port** *portno*} [**secondary**]
	virtual {**policy ssl** *ssl-policy-name*}
	virtual {**policy tcp** *client-side-tcp-policy-name*}
ssl-proxy vlan	**admin**
	exit
	gateway *prefix* [**drop** | **forward**]
	help
	ipaddr *prefix mask*
	no
	route {*prefix mask*} {**gateway** *prefix*}

clear ssl-proxy conn

To clear all TCP connections on the entire system, use the **clear ssl-proxy conn** command.

clear ssl-proxy conn

Syntax Description

service *name*	(Optional) Clears the connections for the specified service.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

To reset all the statistics counters that the SSL Services Module maintained, use the **clear ssl-proxy connection** command without options.

Examples

This example shows how to clear the connections for the specified service:

```
ssl-proxy# clear ssl-proxy conn service S6
```

This example shows how to clear all TCP connections on the entire system:

```
ssl-proxy# clear ssl-proxy conn
ssl-proxy#
```

clear ssl-proxy session

To clear all entries from the session cache, use the **clear ssl-proxy session** command.

clear ssl-proxy session

Syntax Description

service *name*	(Optional) Clears the session cache for the specified service.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

To clear all entries from the session cache for all services, use the **clear ssl-proxy session** command without options.

Examples

These examples show how to clear the entries from the session cache for the specified service on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy session service S6
```

This example shows how to clear all entries in the session cache maintained on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy session
ssl-proxy#
```

clear ssl-proxy stats

To reset the statistics counters maintained in different SSL Services Module system components, use the **clear ssl-proxy stats** command.

clear ssl-proxy stats [**crypto** | **fdu** | **ipc** | **pki** | **service** | **ssl** | **tcp**]

Syntax Description

crypto	(Optional) Clears the crypto statistics information.
fdu	(Optional) Clears the F6DU statistics information
ipc	(Optional) Clears the inter-process communications (IPC) statistics information.
pki	(Optional) Clears the public key infrastruture (PKI) statistics information.
service *name*	(Optional) Clears the statistics information for a specific service.
ssl	(Optional) Clears the SSL statistics information
tcp	(Optional) Clears the TCP statistics information

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

To reset all the statistics counters that the SSL Services Module maintained, use the **clear ssl-proxy stats** command without options.

Examples

These examples show how to reset the statistics counters maintained in different system components on the SSL Services Module:

```
ssl-proxy# clear ssl-proxy stats crypto
ssl-proxy# clear ssl-proxy stats ipc
ssl-proxy# clear ssl-proxy stats pki
ssl-proxy# clear ssl-proxy stats service S6
```

This example shows how to clear all statistic counters that the SSL Services Module maintained:

```
ssl-proxy# clear ssl-proxy stats
ssl-proxy#
```

crypto ca export pem

To export privacy-enhanced mail (PEM) files from the SSL Services Module, use the **crypto ca export pem** command.

crypto ca export *trustpoint_label* **pem** {**terminal** {**des** | **3des**} {**url** *url*}} *pass_phrase*

Syntax Description

trustpoint-label	Name of the trustpoint.
terminal	Displays the request on the terminal.
des	Specifies the 56-bit DES-CBC encryption algorithm.
3des	Specifies the 168-bit DES (3DES) encryption algorithm.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Export to the FTP: file system • **null:**—Export to the NULL: file system • **nvram:**—Export to the NVRAM: file system • **rcp:**—Export to the RCP: file system • **scp:**—Export to the SCP: file system • **system:**—Export to the system: file system • **tftp:**—Export to the TFTP: file system
pass_phrase	Pass phrase used to protect the private key.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The *pass_phrase* can be any phrase including spaces and punctuation escept for "**?**", which has special meaning to the Cisco IOS parser.

Pass phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported, and when this key is imported the same pass phrase must be entered to decrypt it.

A key marked as unexportable cannot be exported.

You can change the default file extensions when prompted. The default file extensions are as follows:

- public key (.pub)
- private key (.prv)
- certificate (.crt)
- CA certificate (.ca)
- signature key (-sign)
- encryption key (-encr)

Note In SSL software release 1.2, only the private key (.prv), the server certificate (.crt), and the issuer CA certificate (.ca) of the server certificate are exported. To export the whole certificate chain, including all the CA certificates, use a PKCS12 file instead of PEM files."

Examples

This example shows how to export a PEM-formatted file on the SSL Services Module:

```
ssl-proxy(config)#crypto ca import TP5 pem url tftp://10.1.1.1/TP5 password
% Importing CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.ca]?
Reading file from tftp://10.1.1.1/TP5.ca
Loading TP5.ca from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1976 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.prv]?
Reading file from tftp://10.1.1.1/TP5.prv
Loading TP5.prv from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 963 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.crt]?
Reading file from tftp://10.1.1.1/TP5.crt
Loading TP5.crt from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1692 bytes]
% PEM files import succeeded.
ssl-proxy(config)#end
ssl-proxy#
*Apr 11 15:11:29.901: %SYS-5-CONFIG_I: Configured from console by console
```

Related Commands

crypto ca import pem

crypto ca import pem

To import a PEM-formatted file to the SSL Services Module, use the **crypto ca import pem** command.

crypto ca import *trustpoint_label* **pem** [**exportable**] {**terminal** | **url** *url* | **usage-keys**} *pass_phrase*

Syntax Description

trustpoint-label	Name of the trustpoint.
exportable	Specifies that key can be exported.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Export to the FTP: file system • **null:**—Export to the null: file system • **nvram:**—Export to the NVRAM: file system • **rcp:**—Export to the RCP: file system • **scp:**—Export to the SCP: file system • **system:**—Export to the system: file system • **tftp:**—Export to the TFTP: file system
pass_phrase	Pass phrase.
usage-keys	Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

You will receive an error if you enter the pass phrase incorrectly.The *pass_phrase* can be any phrase including spaces and punctuation except for “**?**”, which has special meaning to the Cisco IOS parser.

Pass phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported, and the same pass phrase must be entered when this key is imported to decrypt it.

When importing RSA keys, a public key or its corresponding certificate can be used.

The **crypto ca import pem** command imports only the private key (.prv), the server certificate (.crt), and the issuer CA certificate (.ca). If you have more than one level of CA in the certificate chain, you need to import the root and subordinate CA certificates before this command is issued for authentication. Use cut-and-paste or TFTP to import the root and subordinate CA certificates.

Examples

This example shows how to import a PEM-formatted file from the SSL Services Module:

```
ssl-proxy(config)# crypto ca import TP5 pem url tftp://10.1.1.1/TP5 password
% Importing CA certificate...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.ca]?
Reading file from tftp://10.1.1.1/TP5.ca
Loading TP5.ca from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1976 bytes]

% Importing private key PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.prv]?
Reading file from tftp://10.1.1.1/TP5.prv
Loading TP5.prv from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 963 bytes]

% Importing certificate PEM file...
Address or name of remote host [10.1.1.1]?
Destination filename [TP5.crt]?
Reading file from tftp://10.1.1.1/TP5.crt
Loading TP5.crt from 10.1.1.1 (via Ethernet0/0.168): !
[OK - 1692 bytes]
% PEM files import succeeded.
ssl-proxy(config)# end
ssl-proxy#
*Apr 11 15:11:29.901: %SYS-5-CONFIG_I: Configured from console by console
```

Related Commands

crypto ca export pem

crypto ca export pkcs12

To export a PKCS12 file from the SSL Services Module, use the **crypto ca export** command.

crypto ca export *trustpoint_label* **pkcs12** *file_system* [*pkcs12_filename*] *pass_phrase*

Syntax Description

trustpoint_label	Specifies the trustpoint label.
file_system	Specifies the file system. Valid values are **scp:, ftp:, nvram:, rcp:,** and **tftp:**
pkcs12_filename	Specifies the name of the PKCS12 file to import.
pass_phrase	Specifies the pass phrase of the PKCS12 file.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

Imported key pairs cannot be exported.

If you are using SSH, we recommend using SCP (secure file transfer) when exporting a PKCS12 file. SCP authenticates the host and encrypts the transfer session.

If you do not specify *pkcs12_filename*, you will be prompted to accept the default filename (the default filename is the *trustpoint_label*) or enter the filename. For the **ftp:** or **tftp:** value, include the full path in the *pkcs12_filename*.

You will receive an error if you enter the pass phrase incorrectly.

If there is more than one level of CA, the root CA and all the subordinate CA certificates are exported in the PKCS12 file.

Examples

This example shows how to export a PKCS12 file using SCP:

```
ssl-proxy(config)#crypto ca export TP1 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Destination username [ssl-proxy]? admin-1
Destination filename [TP1]? TP1.p12

Password:

Writing TP1.p12 Writing pkcs12 file to scp://admin-1@10.1.1.1/TP1.p12

Password:
!
CRYPTO_PKI:Exported PKCS12 file successfully.
ssl-proxy(config)#
```

crypto ca import pkcs12

To import a PKCS12 file to the SSL Services Module, use the **crypto ca import** command.

crypto ca import *trustpoint_label* **pkcs12** *file_system* [*pkcs12_filename*] *pass_phrase*

Syntax Description

trustpoint_label	Specifies the trustpoint label.
file_system	Specifies the file system. Valid values are **scp:, ftp:, nvram:, rcp:,** and **tftp:**
pkcs12_filename	Specifies the name of the PKCS12 file to import.
pass_phrase	Specifies the pass phrase of the PKCS12 file.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

If you are using SSH, we recommend using SCP (secure file transfer) when importing a PKCS12 file. SCP authenticates the host and encrypts the transfer session.

If you do not specify *pkcs12_filename*, you will be prompted to accept the default filename (the default filename is the *trustpoint_label*) or to enter the filename. For the **ftp:** or **tftp:** value, include the full path in the *pkcs12_filename*.

You will receive an error if you enter the pass phrase incorrectly.

If there is more than one level of CA, the root CA and all the subordinate CA certificates are exported in the PKCS12 file.

Examples

This example shows how to import a PKCS12 file using SCP:

```
ssl-proxy(config)# crypto ca import TP2 pkcs12 scp: sky is blue
Address or name of remote host []? 10.1.1.1
Source username [ssl-proxy]? admin-1
Source filename [TP2]? /users/admin-1/pkcs12/TP2.p12

Password:password
Sending file modes:C0644 4379 TP2.p12
!
ssl-proxy(config)#
*Aug 22 12:30:00.531:%CRYPTO-6-PKCS12IMPORT_SUCCESS:PKCS #12 Successfully Imported.
ssl-proxy(config)#
```

crypto key export rsa pem

To export a PEM-formatted RSA key to the SSL Services Module, use the **crypto key export rsa pem** command.

crypto key export rsa *keylabel* **pem** {**terminal** | **url** *url*} {{**3des** | **des**} *pass_phrase*}

Syntax Description

keylabel	Name of the key.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Export to the FTP: file system • **null:**—Export to the null: file system • **nvram:**—Export to the NVRAM: file system • **rcp:**—Export to the RCP: file system • **scp:**—Export to the SCP: file system • **system:**—Export to the system: file system • **tftp:**—Export to the TFTP: file system
des	Specifies the 56-bit DES-CBC encryption algorithm.
3des	Specifies the 168-bit DES (3DES) encryption algorithm.
exportable	(Optional) Specifies that key can be exported.
pass_phrase	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The pass phrase can be any phrase including spaces and punctuation except for “**?**”, which has special meaning to the Cisco IOS parser.

Pass phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported, and the same pass phrase must be entered when this key is imported to decrypt it.

Examples

This example shows how to export a key from the SSL Services Module:

```
ssl-proxy(config)# crypto key export rsa test-keys pem url scp: 3des password
% Key name:test-keys
   Usage:General Purpose Key
Exporting public key...
Address or name of remote host []? 7.0.0.7
Destination username [ssl-proxy]? lab
Destination filename [test-keys.pub]?

Password:

Writing test-keys.pub Writing file to scp://lab@7.0.0.7/test-keys.pub
Password:
!
Exporting private key...
Address or name of remote host []? 7.0.0.7
Destination username [ssl-proxy]? lab
Destination filename [test-keys.prv]?

Password:

Writing test-keys.prv Writing file to scp://lab@7.0.0.7/test-keys.prv
Password:
ssl-proxy(config)#
```

crypto key import rsa pem

To import a PEM-formatted RSA key from the SSL Services Module, use the **crypto key import rsa pem** command.

crypto key import rsa *keylabel* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*}

Syntax Description

keylabel	Name of the key.
usage-keys	(Optional) Specifies that two special-usage key pairs should be generated, instead of one general-purpose key pair.
terminal	Displays the request on the terminal.
url *url*	Specifies the URL location. Valid values are as follows: • **ftp:**—Export to the FTP: file system • **null:**—Export to the null: file system • **nvram:**—Export to the NVRAM: file system • **rcp:**—Export to the RCP: file system • **scp:**—Export to the SCP: file system • **system:**—Export to the system: file system • **tftp:**—Export to the TFTP: file system
exportable	(Optional) Specifies that key can be exported.
passphrase	Pass phrase.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
SSL Services Module Release 1.2(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The pass phrase can be any phrase including spaces and punctuation except “**?**”, which has special meaning to the Cisco IOS parser.

Pass phrase protection associates a pass phrase with the key. The pass phrase is used to encrypt the key when it is exported, and the same pass phrase must be entered when this key is imported to decrypt it.

Examples

This example shows how to import a PEM-formatted RSA key to the SSL Services Module:

```
ssl-proxy(config)# crypto key import rsa newkeys pem url scp: password
% Importing public key or certificate PEM file...
Address or name of remote host []? 7.0.0.7
Source username [ssl-proxy]? lab
Source filename [newkeys.pub]? test-keys.pub

Password:
Sending file modes:C0644 272 test-keys.pub
Reading file from scp://lab@7.0.0.7/test-keys.pub!
% Importing private key PEM file...
Address or name of remote host []? 7.0.0.7
Source username [ssl-proxy]? lab
Source filename [newkeys.prv]? test-keys.prv

Password:
Sending file modes:C0644 963 test-keys.prv
Reading file from scp://lab@7.0.0.7/test-keys.prv!% Key pair import succeeded.

ssl-proxy(config)#
```

debug ssl-proxy

To turn on the debug flags in different system components, use the **debug ssl-proxy** command. Use the **no** form of this command to turn off the debug flags.

debug ssl-proxy {**app** | **fdu** [*type*] | **ipc** | **pki** [*type*] | **ssl** [*type*] | **tcp** [*type*]}

Syntax Description

app	Turns on App debugging.
fdu [*type*]	Turns on FDU debugging; (optional) *type* valid values are **cli**, **hash**, **ipc**, and **trace**. See the "Usage Guidelines" section for additional information.
ipc	Turns on IPC debugging.
pki [*type*]	Turns on PKI debugging; (optional) *type* valid values are **cert**, **events**, **history**, **ipc**, and **key**. See the "Usage Guidelines" section for additional information.
ssl [*type*]	Turns on SSL debugging; (optional) *type* valid values are **alert**, **error**, **handshake**, and **pkt**. See the "Usage Guidelines" section for additional information.
tcp [*type*]	Turns on TCP debugging; (optional) *type* valid values are **event**, **packet**, **state**, and **timers**. See the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **fdu** *type* includes the following values:

- **cli**—Debugs the FDU CLI.
- **hash**—Debugs the FDU hash.
- **ipc** —Debugs the FDU IPC.
- **trace**—Debugs the FDU trace.

The **pki** *type* includes the following values:

- **certs**—Debugs the certificate management.
- **events**—Debugs events.
- **history**—Debugs the certificate history.
- **ipc**—Debugs the IPC messages and buffers.
- **key**—Debugs key management.

The **ssl** *type* includes the following values:

- **alert**—Debugs the SSL alert events.
- **error**—Debugs the SSL error events.
- **handshake**—Debugs the SSL handshake events.
- **pkt**—Debugs the received and transmitted SSL packets.

Note Use the TCP debug commands only to troubleshoot basic connectivity issues under little or no load conditions (for instance when no connection is being established to the virtual server or real server).

If you run TCP debug commands, the TCP module displays large amounts of debug information on the console, which can significantly slow down module performance. Slow module performance can lead to delayed processing of TCP connection timers, packets, and state transitions.

The **tcp** *type* includes the following values:

- **events**—Debugs the TCP events.
- **pkt**—Debugs the received and transmitted TCP packets.
- **state**—Debugs the TCP states.
- **timers**—Debugs the TCP timers.

Examples

This example shows how to turn on App debugging:

```
ssl-proxy# debug ssl-proxy app
ssl-proxy#
```

This example shows how to turn on FDU debugging:

```
ssl-proxy# debug ssl-proxy fdu
ssl-proxy#
```

This example shows how to turn on IPC debugging:

```
ssl-proxy# debug ssl-proxy ipc
ssl-proxy#
```

This example shows how to turn on PKI debugging:

```
ssl-proxy# debug ssl-proxy pki
ssl-proxy#
```

This example shows how to turn on SSL debugging:

```
ssl-proxy# debug ssl-proxy ssl
ssl-proxy#
```

This example shows how to turn on TCP debugging:

```
ssl-proxy# debug ssl-proxy tcp
ssl-proxy#
```

This example shows how to turn off TCP debugging:

```
ssl-proxy# no debug ssl-proxy tcp
ssl-proxy#
```

show ssl-proxy admin-info

To display the administration VLAN and related IP and gateway addresses, use the **show ssl-proxy admin-info** command.

show ssl-proxy admin-info

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the administration VLAN and related IP and gateway addresses:

```
ssl-proxy# show ssl-proxy admin-info
STE administration VLAN: 2
STE administration IP address: 207.57.100.18
STE administration gateway: 207.0.207.5
ssl-proxy#
```

Related Commands

ssl-proxy vlan

show ssl-proxy buffers

To display the TCP buffer usage information, use the **show ssl-proxy buffers** command.

show ssl-proxy buffers

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the buffer usage and other information in the TCP subsystem:

```
ssl-proxy# show ssl-proxy buffers
Buffers info for TCP module 1
TCP data buffers used 2816 limit 112640
TCP ingress buffer pool size 56320 egress buffer pool size 56320
TCP ingress data buffers min-thresh 7208960 max-thresh 21626880
TCP ingress data buffers used Current 0 Max 0
TCP ingress buffer RED shift 9 max drop prob 10
Conns consuming ingress data buffers 0
Buffers with App 0
TCP egress data buffers used Current 0 Max 0
Conns consuming egress data buffers 0
In-sequence queue bufs 0 OOO bufs 0
ssl-proxy#
```

Related Commands

ssl-proxy policy tcp

show ssl-proxy certificate-history

To display the certificate event history information, use the **show ssl-proxy certificate-history** command.

show ssl-proxy certificate-history [**service** [*name*]]

Syntax Description

service [*name*]	Displays all certificate records of a proxy service and (optionally) for a specific proxy service.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **show ssl-proxy certificate-history** command displays these records:

- service name
- keypair name
- generation or import time
- trustpoint name
- certificate subject name
- certificate issuer name
- serial number
- date

A syslog message is generated for each record. The oldest records are deleted after the limit of 512 records is reached.

Examples

This example shows how to display the event history of all the certificate processing:

```
ssl-proxy# show ssl-proxy certificate-history
Record 1, Timestamp:00:00:51, 16:36:34 UTC Oct 31 2002
    Installed Server Certificate, Index 5
    Proxy Service:s1,  Trust Point:t3
    Key Pair Name:k3,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:12:27:58 UTC Oct 30 2002
    Subject Name:OID.1.2.840.113549.1.9.2 = simpson5-2-ste.cisco.com,
OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5D3D1931000100000D99
    Validity Start Time:21:58:12 UTC Oct 30 2002
    End Time:22:08:12 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 2, Timestamp:00:01:06, 16:36:49 UTC Oct 31 2002
    Installed Server Certificate, Index 6
    Proxy Service:s5,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 3, Timestamp:00:01:34, 16:37:18 UTC Oct 31 2002
    Installed Server Certificate, Index 7
    Proxy Service:s6,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 4, Timestamp:00:01:40, 16:37:23 UTC Oct 31 2002
    Deleted Server Certificate, Index 0
    Proxy Service:s6,  Trust Point:t6
    Key Pair Name:k6,  Key Usage:RSA General Purpose, Not Exportable
    Time of Key Generation:00:28:28 UTC Mar 1 1993
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.8, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5CB5CFD6000100000D97
    Validity Start Time:19:30:26 UTC Oct 30 2002
    End Time:19:40:26 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record
% Total number of certificate history records displayed = 4
ssl-proxy#
```

This example shows how to display the certificate record for a specific proxy service:

```
ssl-proxy# show ssl-proxy certificate-history service s6
Record 3, Timestamp:00:01:34, 16:37:18 UTC Oct 31 2002
    Installed Server Certificate, Index 7
    Proxy Service:s6,  Trust Point:t10
    Key Pair Name:k10,  Key Usage:RSA General Purpose, Exportable
    Time of Key Generation:07:56:43 UTC Oct 11 2002
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.9, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:24BC81B7000100000D85
    Validity Start Time:22:38:00 UTC Oct 19 2002
    End Time:22:48:00 UTC Oct 19 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record

  Record 4, Timestamp:00:01:40, 16:37:23 UTC Oct 31 2002
    Deleted Server Certificate, Index 0
    Proxy Service:s6,  Trust Point:t6
    Key Pair Name:k6,  Key Usage:RSA General Purpose, Not Exportable
    Time of Key Generation:00:28:28 UTC Mar 1 1993
    Subject Name:CN = host1.cisco.com, OID.1.2.840.113549.1.9.2 =
simpson5-2-ste.cisco.com, OID.1.2.840.113549.1.9.8 = 207.79.1.8, OID.2.5.4.5 = B0FFF235
    Issuer Name:CN = SimpsonTestCA, OU = Simpson Lab, O = Cisco Systems, L = San Jose, ST
= CA, C = US, EA =<16> simpson-pki@cisco.com
    Serial Number:5CB5CFD6000100000D97
    Validity Start Time:19:30:26 UTC Oct 30 2002
    End Time:19:40:26 UTC Oct 30 2003
    Renew Time:00:00:00 UTC Jan 1 1970
  End of Certificate Record
Total number of certificate history records displayed = 2
```

Related Commands

ssl-proxy service

show ssl-proxy conn

To display the TCP connections from the SSL Services Module, use the **show ssl-proxy conn** command.

show ssl-proxy conn 4tuple [local {**ip** *local-ip-addr local-port*} [**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]]

show ssl-proxy conn 4tuple [local {**port** *local-port*} [**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]]

show ssl-proxy conn 4tuple [local {**remote** [{**ip** *remote-ip-addr* [**port** *remote-port*]} | {**port** *remote-port* [**ip** *remote-ip-addr*]}]]

show ssl-proxy conn service *name*

Syntax Description

4tuple	Displays the TCP connections for a specific address.
local	(Optional) Displays the TCP connections for a specific local device.
ip *local-ip-addr*	IP address of a local device.
local-port	Port number of a local device.
remote	(Optional) Displays the TCP connections for a specific remote device.
ip *remote-ip-addr*	IP address of a remote device.
port *remote-port*	Port number of a remote device.
port *local-port*	(Optional) Displays the TCP connections for a specific local port.
service *name*	Displays the TCP connections for a specific proxy service.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

These examples show different ways to display the TCP connection established from the SSL Services Module:

```
ssl-proxy# show ssl-proxy conn
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.0.0.10:4430         1.200.200.14:48582    2    0      0      0      ESTAB
1.200.200.14:48582    2.100.100.72:80       2    1      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48583    2    2      0      0      ESTAB
1.200.200.14:48583    2.100.100.72:80       2    3      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48584    2    4      0      0      ESTAB
1.200.200.14:48584    2.100.100.72:80       2    5      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48585    2    6      0      0      ESTAB
1.200.200.14:48585    2.100.100.72:80       2    7      0      0      ESTAB

2.0.0.10:4430         1.200.200.14:48586    2    8      0      0      ESTAB
1.200.200.14:48586    2.100.100.72:80       2    9      0      0      ESTAB

ssl-proxy# show ssl-proxy conn 4tuple local port 443
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.133:443       1.200.200.12:39728    2    113676 0      0      TWAIT
No Bound Connection

2.50.50.133:443       1.200.200.12:39729    2    113680 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:40599    2    113684 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48031    2    114046 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48032    2    114048 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48034    2    114092 0      0      TWAIT
No Bound Connection

2.50.50.132:443       1.200.200.13:48035    2    114100 0      0      TWAIT
No Bound Connection
```

```
ssl-proxy# show ssl-proxy conn 4tuple remote ip 1.200.200.14
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.131:443       1.200.200.14:38814    2    58796  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38815    2    58800  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38817    2    58802  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38818    2    58806  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38819    2    58810  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38820    2    58814  0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:38821    2    58818  0      0      TWAIT
No Bound Connection

ssl-proxy# show ssl-proxy conn service iis1
Connections for TCP module 1
Local Address         Remote Address        VLAN Conid  Send-Q Recv-Q State
--------------------- --------------------- ---- ------ ------ ------ ------
2.50.50.131:443       1.200.200.14:41217    2    121718 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41218    2    121722 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41219    2    121726 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41220    2    121794 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41221    2    121808 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41222    2    121940 0      0      TWAIT
No Bound Connection

2.50.50.131:443       1.200.200.14:41223    2    122048 0      0      TWAIT
No Bound Connection
```

show ssl-proxy crash-info

To collect software-forced reset information from the SSL Services Module, use the **show ssl-proxy crash-info** command.

show ssl-proxy crash-info [**brief** | **details**]

Syntax Description

brief	(Optional) Collects a subset of software-forced reset information, limited to processor registers.
details	(Optional) Collects the full set of software-forced reset information, including exception and interrupt stacks dump (this can take up to 10 minutes to complete printing)

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

The following example shows how to collect software-forced reset information:

```
ssl-proxy# show ssl-proxy crash-info

===== SSL SERVICE MODULE - START OF CRASHINFO COLLECTION =====


------------- COMPLEX 0 [FDU_IOS] ----------------------

NVRAM CHKSUM:0xEB28
NVRAM MAGIC:0xC8A514F0
NVRAM VERSION:1

++++++++++ CORE 0 (FDU) ++++++++++++++++++++++

   CID:0
   APPLICATION VERSION:2003.04.15 14:50:20 built for cantuc
   APPROXIMATE TIME WHEN CRASH HAPPENED:14:06:04 UTC Apr 16 2003
   THIS CORE DIDN'T CRASH
   TRACEBACK:222D48 216894
   CPU CONTEXT  -----------------------------

$0 :00000000, AT :00240008, v0 :5A27E637, v1 :000F2BB1
a0 :00000001, a1 :0000003C, a2 :002331B0, a3 :00000000
t0 :00247834, t1 :02BFAAA0, t2 :02BF8BB0, t3 :02BF8BA0
t4 :02BF8BB0, t5 :00247834, t6 :00000000, t7 :00000001
```

```
s0 :00000000, s1 :0024783C, s2 :00000000, s3 :00000000
s4 :00000001, s5 :0000003C, s6 :00000019, s7 :0000000F
t8 :00000001, t9 :00000001, k0 :00400001, k1 :00000000
gp :0023AE80, sp :031FFF58, s8 :00000019, ra :00216894
LO :00000000, HI :0000000A, BADVADDR :828D641C
EPC :00222D48, ErrorEPC :BFC02308, SREG :34007E03
Cause 0000C000 (Code 0x0):Interrupt exception

CACHE ERROR registers  -------------------

CacheErrI:00000000, CacheErrD:00000000
ErrCtl:00000000, CacheErrDPA:0000000000000000

   PROCESS STACK ----------------------------
      stack top:0x3200000

   Process stack in use:

   sp is close to stack top;

   printing 1024 bytes from stack top:

031FFC00:06405DE0 002706E0 0000002D 00000001  .@]`.'.`...-....
031FFC10:06405DE0 002706E0 00000001 0020B800  .@]`.'.`..... 8.
031FFC20:031FFC30 8FBF005C 14620010 24020004  ..|0.?.\.b..$...
...........
...........
...........
FFFFFFD0:00000000 00000000 00000000 00000000 ................
FFFFFFE0:00627E34 00000000 00000000 00000000 .b~4............
FFFFFFF0:00000000 00000000 00000000 00000006 ................


===== SSL SERVICE MODULE - END OF CRASHINFO COLLECTION =======
```

The following example shows how to collect software-forced reset information:

```
ssl-proxy# show ssl-proxy crash-info brief


===== SSL SERVICE MODULE - START OF CRASHINFO COLLECTION =====


------------- COMPLEX 0 [FDU_IOS] ----------------------

SKE CRASH INFO Error: wrong MAGIC # 0

CLI detected an error in FDU_IOS crash-info; wrong magic.

------------- COMPLEX 1 [TCP_SSL] ----------------------


Crashinfo fragment #0 from core 2 at offset 0 error:
Remote system reports wrong crashinfo magic.
Bad fragment received. Reception abort.

CLI detected an error in TCP_SSL crash-info;


===== SSL SERVICE MODULE - END OF CRASHINFO COLLECTION =======
```

show ssl-proxy mac address

To display the current MAC address, use the **show ssl-proxy mac address** command.

show ssl-proxy mac address

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the current MAC address used in the SSL Services Module:

```
ssl-proxy# show ssl-proxy mac address
STE MAC address: 00e0.b0ff.f232
ssl-proxy#
```

show ssl-proxy natpool

To display NAT pool information, use the **show ssl-proxy natpool** command.

show ssl-proxy natpool [*name*]

Syntax Description

name	(Optional) NAT pool name.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display information for a specific NAT address pool configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy natpool NP1
Start ip: 207.57.110.1
End ip: 207.57.110.8
netmask: 255.0.0.0
vlan associated with natpool: 2
SSL proxy services using this natpool:
S2
S3
S1
S6
Num of proxies using this natpool: 4
ssl-proxy#
```

Related Commands

ssl-proxy natpool

show ssl-proxy policy

To display the configured SSL or TCP policies, use the **show ssl-proxy policy** command.

show ssl-proxy policy {**ssl** | **tcp**} [*name*]

Syntax Description

ssl	Displays the configured SSL policies.
tcp	Displays the configured TCP policies.
name	(Optional) Policy name.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display policy information for a specific SSL policy configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy policy ssl ssl-policy1
Cipher suites: (None configured, default ciphers included)
     rsa-with-rc4-128-md5
     rsa-with-rc4-128-sha
     rsa-with-des-cbc-sha
     rsa-with-3des-ede-cbc-sha
SSL Versions enabled:SSL3.0, TLS1.0
strict close protocol:disabled
Session Cache:enabled
Handshake timeout not configured (never times out)
Num of proxies using this poilicy:0
```

This example shows how to display policy information for a specific TCP policy configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy policy tcp tcp-policy1
MSS                  1250
SYN timeout          75
Idle timeout          600
FIN wait timeout     75
Rx Buffer Share  32768
Tx Buffer Share  32768

Usage count of this policy:0
ssl-proxy#
```

show ssl-proxy service

To display the configured SSL virtual server information, use the **show ssl-proxy service** command.

show ssl-proxy service [*name*]

Syntax Description

name	(Optional) Service name.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display all SSL virtual services configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service

Proxy Service Name Admin Operation Events
status status
S2 up up
S3 up up
S1 up up
S6 down down
ssl-proxy#
```

This example shows how to display a specific SSL virtual service configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy service S6
Service id: 0, bound_service_id: 256
Virtual IP: 10.10.1.104, port: 443
Server IP: 10.10.1.100, port: 80
Virtual SSL Policy: SSL1_PLC
rsa-general-purpose certificate trustpoint: tptest
  Certificate chain for new connections:
    Server Certificate:
       Key Label: tptest
       Serial Number: 01
    Root CA Certificate:
       Serial Number: 00
  Certificate chain complete
Admin Status: up
Operation Status: down
Proxy status: No Client VLAN, No Server VLAN
ssl-proxy#
```

show ssl-proxy stats

To display statistics counter information, use the **show ssl-proxy stats** command.

show ssl-proxy stats [*type*]

Syntax Description

type	(Optional) Information type; valid values are **crypto**, **ipc**, **pki**, **service**, **ssl**, and **tcp**. See the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	The output of the **show ssl-proxy stats** command was changed to include session allocation failure and session limit-exceed table information.

Usage Guidelines

The *type* values are defined as follows:

- **crypto**—Displays crypto statistical information.
- **ipc**—Displays IPC statistical information.
- **pki**—Displays PKI statistical information.
- **service**—Displays proxy service statistical information.
- **ssl**—Displays SSL detailed statistical information.
- **tcp**—Displays TCP detailed statistical information.

Examples

This example shows how to display all the statistics counters collected on the SSL Services Module:

```
ssl-proxy# show ssl-proxy stats
TCP Statistics:
    Conns initiated     : 20636         Conns accepted      : 20636
    Conns established   : 28744         Conns dropped       : 28744
    Conns closed        : 41272         SYN timeouts        : 0
    Idle timeouts       : 0             Total pkts sent     : 57488
    Data packets sent   : 0             Data bytes sent     : 0
    Total Pkts rcvd     : 70016         Pkts rcvd in seq    : 0
    Bytes rcvd in seq   : 0
```

```
SSL Statistics:
    conns attempted     : 20636         conns completed     : 20636
    full handshakes     : 0             resumed handshakes  : 0
    active conns        : 0             active sessions     : 0
    renegs attempted    : 0             conns in reneg      : 0
    handshake failures  : 20636         data failures       : 0
    fatal alerts rcvd   : 0             fatal alerts sent   : 0
    no-cipher alerts    : 0             ver mismatch alerts : 0
    no-compress alerts  : 0             bad macs received   : 0
    pad errors          : 0             session fails       : 0

FDU Statistics:
    IP Frag Drops       : 0             Serv_Id Drops       : 9
    Conn Id Drops       : 0             Bound Conn Drops    : 0
    Vlan Id Drops       : 0             Checksum Drops      : 0
    IOS Congest Drops   : 0             IP Version Drops    : 0
    Hash Full Drops     : 0             Hash Alloc Fails    : 0
    Flow Creates        : 41272         Flow Deletes        : 41272
    conn_id allocs      : 41272         conn_id deallocs    : 41272
    Tagged Drops        : 0             Non-Tagged Drops    : 0
    Add ipcs            : 3             Delete ipcs         : 0
    Disable ipcs        : 3             Enable ipcs         : 0
    Unsolicited ipcs    : 0             Duplicate ADD ipcs  : 0
    IOS broadcast pkts  : 29433         IOS unicast pkts    : 5
    IOS total pkts      : 29438
ssl-proxy#
```

This example shows how to display PKI statistical information:

```
ssl-proxy# show ssl-proxy stats pki
PKI Memory Usage Counters:
  Malloc count: 0
  Setstring count: 0
  Free count: 0
  Malloc failed: 0
  Ipc alloc count: 0
  Ipc free count: 0
  Ipc alloc failed: 0
PKI IPC Counters:
  Request buffer sent: 0
  Request buffer received: 0
  Request duplicated: 0
  Response buffer sent: 0
  Response buffer received: 0
  Response timeout: 0
  Response with error status: 0
  Response with no request: 0
  Response duplicated: 0
  Message type error: 0
PKI Accumulative Certificate Counters:
  Proxy service trustpoint added: 0
  Proxy service trustpoint deleted: 0
  Proxy service trustpoint modified: 0
  Keypair added: 0
  Keypair deleted: 0
  Wrong key type: 0
  Server certificate added: 0
  Server certificate deleted: 0
  Server certificate rolled over: 0
  Server certificate completed: 0
  Intermediate CA certificate added: 0
  Intermediate CA certificate deleted: 0
  Root CA certificate added: 0
  Root CA certificate deleted: 0
  Certificate overwritten: 0
  History records written: 0
  History records read from NVRAM: 0
  Key cert table entries in use: 0
ssl-proxy#
```

show ssl-proxy status

To display status information, use the **show ssl-proxy status** command.

show ssl-proxy status

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	The output of the **show ssl-proxy statu s** command was changed to include statistics displayed at 1 second, 1 minute, and 5 minutes traffic rate for CPU utilization.

Examples

This example shows how to display the status on the SSL Services Module:

```
ssl-proxy# show ssl-proxy status
FDU cpu is alive!
FDU cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0x4D52D1B7         int cycles  : 0x6B6C9937
    total cycles: 0xB954D5BEB6FA
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0

    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0


TCP cpu is alive!
TCP cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0xA973D74D         int cycles  : 0xAA03E1D89A
    total cycles: 0xB958C8FF0E73
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0

    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0
```

```
SSL cpu is alive!
SSL cpu utilization:
    % process util   : 0             % interrupt util : 0

    proc cycles : 0xD475444          int cycles  : 0x21865088E
    total cycles: 0xB958CCEB8059
    % process util (5 sec)   : 0             % interrupt util (5 sec) : 0

    % process util (1 min)  : 0             % interrupt util (1 min): 0
    % process util (5 min)  : 0             % interrupt util (5 min) : 0
```

show ssl-proxy version

To display the current image version, use the **show ssl-proxy version** command.

show ssl-proxy version

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display the image version currently running on the SSL Services Module:

```
ssl-proxy# show ssl-proxy version
Cisco Internetwork Operating System Software
IOS (tm) SVCSSL Software (SVCSSL-K9Y9-M), Version 12.2(14.6)SSL(0.19)  INTERIM TEST
SOFTWARE
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Thu 10-Apr-03 03:03 by integ
Image text-base: 0x00400078, data-base: 0x00ABE000

ROM: System Bootstrap, Version 12.2(11)YS1 RELEASE SOFTWARE

ssl-proxy uptime is 3 days, 22 hours, 22 minutes
System returned to ROM by power-on
System image file is "tftp://10.1.1.1/unknown"
AP Version 1.2(1)

ssl-proxy#
```

show ssl-proxy vlan

To display VLAN information, use the **show ssl-proxy vlan** command.

show ssl-proxy vlan [*vlan-id* | **debug**]

Syntax Description

vlan-id	(Optional) VLAN ID. Displays information for a specific VLAN; valid values are from 1 to 1005.
debug	(Optional) Displays debug information.

Defaults

This command has no default settings.

Command Modes

EXEC mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to display all the VLANs configured on the SSL Services Module:

```
ssl-proxy# show ssl-proxy vlan
VLAN index 2 (admin VLAN)
   IP addr 10.1.1.1 NetMask 255.0.0.0 Gateway 10.1.1.5
   Network 10.1.1.2 Mask 255.0.0.0 Gateway 10.1.1.6
VLAN index 3
   IP addr 10.1.1.3 NetMask 255.0.0.0 Gateway 10.1.1.6
VLAN index 6
   IP addr 10.1.1.4 NetMask 255.0.0.0

ssl-proxy#
```

Related Commands

ssl-proxy vlan

ssl-proxy crypto selftest

To initiate a cryptographic self-test, use the **ssl-proxy crypto selftest** command. Use the **no** form of this command to disable the testing.

ssl-proxy crypto selftest [**time-interval** *seconds*]

no ssl-proxy crypto selftest

Syntax Description

time-interval *seconds*	(Optional) Sets the time interval between test cases; valid values are from 1 to 8 seconds.

Defaults

3 seconds

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **ssl-proxy crypto selftest** command enables a set of crypto algorithm tests to be run on the SSL processor in the background. Random number generation, hashing, encryption and decryption, and MAC generation are tested with a time interval in between test cases.

This test is run only for troubleshooting purposes. Running this test will impact run-time performance.

To display the results of the self-test, enter the **show ssl-proxy stats crypto** command.

Examples

This example shows how to start a cryptographic self-test:

```
ssl-proxy (config)# ssl-proxy crypto selftest
ssl-proxy (config)#
```

ssl-proxy mac address

To configure a MAC address, use the **ssl-proxy mac address** command.

ssl-proxy mac address *mac-addr*

Syntax Description

mac-addr	MAC address; see the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

Enter the MAC address in this format: H.H.H.

Examples

This example shows how to configure a MAC address:

```
ssl-proxy (config)# ssl-proxy mac address 00e0.b0ff.f232
ssl-proxy (config)#
```

Related Commands

show ssl-proxy mac address

ssl-proxy natpool

To define a pool of IP addresses, which the SSL Services Module uses for implementing the client NAT, use the **ssl-proxy natpool** command.

ssl-proxy natpool *nat-pool-name start-ip-addr* {**netmask** *netmask*}

Syntax Description

nat-pool-name	NAT pool name.
start-ip-addr	Start IP address.
netmask *netmask*	Netmask; see the "Usage Guidelines" section for additional information.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to define a pool of IP addresses:

```
ssl-proxy (config)# ssl-proxy natpool NP2 207.59.10.01 207.59.10.08 netmask 255.0.0.0
ssl-proxy (config)#
```

Related Commands

show ssl-proxy natpool

ssl-proxy pki history

To enable the PKI event history option, use the **ssl-proxy pki history** command. Use the **no** form of this command to disable the logging and clear the memory.

ssl-proxy pki history

no ssl-proxy pki history

Syntax Description

This command has no arguments or keywords.

Defaults

Disabled

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

The **ssl-proxy pki history** command enables logging of certificate history records per-proxy service into memory and generates a syslog message per record. Each record keeps track of the addition or deletion of a keypair or certificate into the proxy services key and the certificate table.

When the index of the table changes, this command logs the following information:

- Key pair name
- Trustpoint label
- Service name
- Subject name
- Serial number of the certificate

Up to 512 records can be stored in the memory at one time.

Examples

This example shows how to enable the PKI event history option:

```
ssl-proxy (config)# ssl-proxy pki history
ssl-proxy (config)#
```

Related Commands

show ssl-proxy stats

ssl-proxy policy ssl

To enter the SSL-policy configuration submode, use the **ssl-proxy policy ssl** command.

ssl-proxy policy ssl *ssl-policy-name*

Syntax Description

ssl-policy-name	SSL policy name.

Defaults

The defaults are as follows:

- **cipher** is all.
- **close-protocol** is enabled.
- **session-caching** is enabled.
- **version** is all.
- **session-cache size** *size* is 262143 entries.
- **timeout session** *timeout* is 0 seconds.
- **timeout handshake** *timeout* is 0 seconds.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	This command was changed to add the following subcommands: • **session-cache size** *size* • **timeout session** *timeout* [**absolute**]

Usage Guidelines

In the SSL-policy configuration submode, you can define the SSL policy for one or more SSL-proxy services.

Each SSL-policy configuration submode command is entered on its own line.

Table B-3 lists the commands available in SSL-policy configuration submode.

Table B-3 SSL-Policy Configuration Submode Command Descriptions

cipher-suite {RSA_WITH_3DES_EDE_CBC_SHA | RSA_WITH_DES_CBC_SHA | RSA_WITH_RC4_128_MD5 | RSA_WITH_RC4_128_SHA | all}	Allows you to configure a list of cipher-suites acceptable to the proxy-server; see the "Usage Guidelines" section for information about the cipher suites.
[**no**] **close-protocol enable**	Allows you to configure the SSL close-protocol behavior. Use the **no** form of this command to disable close-protocol.
default {**cipher** | **close-protocol** | **session-cache** | **version**}	Sets a command to its default settings.
exit	Exits from SSL-policy configuration submode.
help	Provides a description of the interactive help system.
[**no**] **session-cache enable**	Allows you to enable the session-caching feature. Use the **no** form of this command to disable session-caching.
session-cache size *size*	Specifies the maximum number of session entries to be allocated for a given service; valid values are from 1 to 262143 entries.
timeout handshake *timeout*	Allows you to configure how long the module keeps the connection in handshake phase; valid values are from 0 to 65535 seconds.
timeout session *timeout* [**absolute**]	Allows you to configure the session timeout. The syntax description is as follows: • *timeout*—Session timeout; valid values are from 0 to 72000 seconds. • **absolute—**(Optional) The session entry is not removed until the configured timeout has completed.
version {**all** | **ssl3** | **tls1**}	Allows you to set the version of SSL used to one of the following: • **all**—Both SSL3 and TLS1 versions are used. • **ssl3**—SSL version 3 is used. • **tls1**—TLS version 1 is used.

You can define the SSL policy templates using the **ssl-proxy policy ssl** *ssl-policy-name* command and associate a SSL policy with a particular proxy server using the proxy server configuration CLI. The SSL policy template allows you to define various parameters associated with the SSL handshake stack.

When close-notify is enabled, a close-notify alert message is sent to the client and a close-notify alert message is expected from the client as well. When disabled, the server sends a close-notify alert message to the client, however the server does not expect, nor wait for, a close-notify message from the client before tearing down the session.

The cipher-suite names follow the same convention as the existing SSL Stacks.

The cipher-suites acceptable to the proxy-server are as follows:

- RSA_WITH_3DES_EDE_CBC_SHA— RSA with 3des-sha
- RSA_WITH_DES_CBC_SHA—RSA with des-sha
- RSA_WITH_RC4_128_MD5—RSA with rc4-md5
- RSA_WITH_RC4_128_SHA—RSA with rc4-sha
- all—All supported ciphers

If you enter the **timeout session** *timeout* **absolute** command, the session entry is kept in the session cache for the configured timeout before it is cleaned up. If the session cache is full with the timers being active for all the entries and the **absolute** option is configured, all further new sessions are rejected.

If you enter the **timeout session** *timeout* command without the **absolute** option, the specified timeout is treated as the maximum timeout and a best-effort is made to keep the session entry in the session cache. If the session cache runs out of session entries, a session entry that is currently being used is removed for incoming new connections.

Examples

This example shows how to enter the SSL-policy configuration submode:

```
ssl-proxy (config)# ssl-proxy policy ssl sslpl1
ssl-proxy (config-ssl-policy)#
```

This example shows how to define the cipher suites supported for the SSL-policy:

```
ssl-proxy (config-ssl-policy)# cipher RSA_WITH_3DES_EDE_CBC_SHA
ssl-proxy (config-ssl-policy)#
```

This example shows how to enable the SSL session closing protocol:

```
ssl-proxy (config-ssl-policy)# close-protocol enable
ssl-proxy (config-ssl-policy)#
```

This example shows how to disable the SSL session closing protocol:

```
ssl-proxy (config-ssl-policy)# no close-protocol enable
ssl-proxy (config-ssl-policy)#
```

These examples show how to set a given command to its default setting:

```
ssl-proxy (config-ssl-policy)# default cipher
ssl-proxy (config-ssl-policy)# default close-protocol
ssl-proxy (config-ssl-policy)# default session-cache
ssl-proxy (config-ssl-policy)# default version
ssl-proxy (config-ssl-policy)#
```

This example shows how to enable the the session-cache option:

```
ssl-proxy (config-ssl-policy)# session-cache enable
ssl-proxy (config-ssl-policy)#
```

This example shows how to disable the the session-cache option:

```
ssl-proxy (config-ssl-policy)# no session-cache enable
ssl-proxy (config-ssl-policy)#
```

This example shows how to set the maximum number of session entries to be allocated for a given service:

```
ssl-proxy (config-ssl-policy)# session-cache size 22000
ssl-proxy (config-ssl-policy)#
```

This example shows how to configure the session timeout to absolute:

```
ssl-proxy (config-ssl-policy)# timeout session 30000 absolute
ssl-proxy (config-ssl-policy)#
```

These examples show how to enable the support of different SSL versions:

```
ssl-proxy (config-ssl-policy)# version all
ssl-proxy (config-ssl-policy)# version ssl3
ssl-proxy (config-ssl-policy)# version tls1
ssl-proxy (config-ssl-policy)#
```

This example shows how to print out a general help page:

```
ssl-proxy (config-ssl-policy)# help
ssl-proxy (config-ssl-policy)#
```

Related Commands

show ssl-proxy stats
show ssl-proxy stats ssl

ssl-proxy policy tcp

To enter the proxy policy TCP configuration submode, use the **ssl-proxy policy tcp** command. In proxy policy TCP configuration submode, you can define the TCP policy templates.

ssl-proxy policy tcp *tcp-policy-name*

Syntax Description

tcp-policy-name	TCP policy name.

Defaults

The defaults are as follows:

- **timeout inactivity** is 240 seconds.
- **timeout fin-wait** is 600 seconds.
- **buffer-share rx** is 32768 bytes.
- **buffer-share tx** is 32768 bytes.
- **mss** is 1500 bytes .
- **timeout syn** is 75 seconds.
- **timeout reassembly** is 60 seconds.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.
SSL Services Module Release 1.2(1)	This command was changed to add the **timeout reassembly** *time* subcommand.

Usage Guidelines

After you have defined the TCP policy, you can associate the TCP policy with a proxy server using the proxy-policy TCP configuration submode commands.

Each proxy-policy TCP configuration submode command is entered on its own line.

Table B-4 lists the commands available in proxy-policy TCP configuration submode.

Table B-4 Proxy-policy TCP Configuration Submode Command Descriptions

default	Sets a command to its default settings.
exit	Exits from proxy-service configuration submode.

Table B-4 Proxy-policy TCP Configuration Submode Command Descriptions (continued)

[**no**] **timeout fin-wait** *timeout-in-seconds*	Allows you to configure the FIN wait timeout; vaid values are from 75 to 600 seconds. Use the **no** form of this command to return to the default setting.
help	Provides a description of the interactive help system.
[**no**] **timeout inactivity** *timeout-in-seconds*	Allows you to configure the inactivity timeout; valid values are from 0 to 960 seconds. This allows you to set the aging timeout for an idle connection and helps protect the connection resources. Use the **no** form of this command to return to the default setting.
[**no**] **buffer-share rx** *buffer-limit-in-bytes*	Allows you to configure maximum size of the receive buffer share per connection; valid values are from 8192 to 262144. Use the **no** form of this command to return to the default setting.
[**no**] **buffer-share tx** *buffer-limit-in-bytes*	Allows you to configure maximum size of the transmit buffer share per connection; valid values are from 8192 to 262144. Use the **no** form of this command to return to the default setting.
[**no**] **mss** *max-segment-size-in-bytes*	Allows you to configure the maximum segment size the connection identifies in the generated SYN packet; valid values are from 64 to 1460. Use the **no** form of this command to return to the default setting.
[**no**] **timeout syn** *timeout-in-seconds*	Allows you to configure the connection establishment timeout; vaid values are from 5 to 75 seconds. Use the **no** form of this command to return to the default setting.
[**no**] **timeout reassembly** *time*	Allows you to configure the amount of time, in seconds, before the reassembly queue is cleared; valid values are from 0 to 960 seconds (0 = disabled). If the transaction is not complete within the specified time, the reassembly queue is cleared and the connection is dropped. Use the **no** form of this command to return to the default setting.

Usage Guidelines

TCP commands entered on the SSL Services Module can apply either globally or to a particular proxy server.

You can configure a different maximum segment size for the client side and the server side of the proxy server.

The TCP policy template allows you to define parameters associated with the TCP stack.

You can either enter the **no** form of the command to return to the default setting or use the **default** option.

Examples

This example shows how to enter the proxy-policy TCP configuration submode:

```
ssl-proxy (config)# ssl-proxy policy tcp tcppl1
ssl-proxy (config-tcp-policy)#
```

These examples show how to set a given command to its default value:

```
ssl-proxy (config-tcp-policy)# default timeout fin-wait
ssl-proxy (config-tcp-policy)# default inactivity-timeout
ssl-proxy (config-tcp-policy)# default buffer-share rx
ssl-proxy (config-tcp-policy)# default buffer-share tx
ssl-proxy (config-tcp-policy)# default mss
ssl-proxy (config-tcp-policy)# default timeout syn
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the FIN wait timeout in seconds:

```
ssl-proxy (config-tcp-policy)# timeout fin-wait 200
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the inactivity timeout in seconds:

```
ssl-proxy (config-tcp-policy)# timeout inactivity 300
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the maximum receive buffer size configuration:

```
ssl-proxy (config-tcp-policy)# buffer-share rx 16384
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the maximum transmit buffer size configuration:

```
ssl-proxy (config-tcp-policy)# buffer-share tx 13444
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the maximum segment size for TCP:

```
ssl-proxy (config-tcp-policy)# mss 1460
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the initial connection (SYN) timeout value:

```
ssl-proxy (config-tcp-policy)# timeout syn 5
ssl-proxy (config-tcp-policy)#
```

This example shows how to define the reassembly timeout value:

```
ssl-proxy (config-tcp-policy)# timeout reassembly 120
ssl-proxy (config-tcp-policy)#
```

Related Commands

show ssl-proxy policy

ssl-proxy service

To enter the proxy-service configuration submode, use the **ssl-proxy-service** command. In proxy-service configuration submode, you can configure the virtual IP address and port associated with the proxy service and the associated target IP address and port. You can also define TCP and SSL policies for both the client side (beginning with the virtual keyword) and the serve side of the proxy (beginning with the **server** keyword).

ssl-proxy service *ssl-proxy-name*

Syntax Description

ssl-proxy-name	SSL proxy name.

Defaults

Server NAT is enabled, and client NAT is disabled

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

Each proxy-service configuration submode command is entered on its own line.

Table B-5 lists the commands available in proxy-service configuration submode.

Table B-5 Proxy-service Configuration Submode Command Descriptions

Syntax	Description
certificate rsa general-purpose trustpoint *trustpoint-name*	Configures the certificate with RSA general purpose keys and associates a trustpoint to the certificate.
default {**certificate** | **inservice** | **nat** | **server** | **virtual**}	Sets a command to its default settings.
exit	Exits from ssl-proxy service configuration submode.
help	Provides a description of the interactive help system.
inservice	Declares a proxy server as administratively up.
nat {server | client *natpool-name***}**	Specifies the usage of either server NAT or client NAT for the server side connection opened by the SSL Services Module.
server ipaddr *ip-addr* **protocol** *protocol* **port** *portno*	Defines the IP address of the target server for the proxy server. You can also specify the port number and the transport protocol. The target IP address can be a virtual IP address of an SLB device or a real IP address of a web server.

Table B-5 Proxy-service Configuration Submode Command Descriptions (continued)

Syntax	Description
server policy tcp *server-side-tcp-policy-name*	Applies a TCP policy to the server side of a proxy server. You can specify the port number and the transport protocol as well.
virtual {**ipaddr** *ip-addr*} {**protocol** *protocol*} {**port** *portno*} [**secondary**]	Defines the virtual IP address of the virtual server that STE is proxying for. You can also specify the port number and the transport protocol. Valid value for *protocol* is **tcp**; valid values for *portno* is from 1 to 65535. The **secondary** option (optional) prevents the STE from replying to the ARP request coming to the virtual IP address.
virtual {**policy ssl** *ssl-policy-name*}	Applies an SSL policy with the client side of a proxy server.
virtual {**policy tcp** *client-side-tcp-policy-name*}	Applies a TCP policy to the client side of a proxy server.

Both secured and bridge mode between the Content Switching Module (CSM) and the SSL Services Module is supported.

Use the **secondary** option (optional) for bridge-mode topology.

Examples

This example shows how to enter the proxy-service configuration submode:

```
ssl-proxy (config)# ssl-proxy service S6
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure the certificate for the specified SSL proxy services:

```
ssl-proxy (config-ssl-proxy)# certificate rsa general-purpose trustpoint tp1
ssl-proxy (config-ssl-proxy)#
```

These examples show how to set a specified command to its default value:

```
ssl-proxy (config-ssl-proxy)# default certificate
ssl-proxy (config-ssl-proxy)# default inservice
ssl-proxy (config-ssl-proxy)# default nat
ssl-proxy (config-ssl-proxy)# default server
ssl-proxy (config-ssl-proxy)# default virtual
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure a virtual IP address for the specified virtual server:

```
ssl-proxy (config-ssl-proxy)# virtual ipaddr 207.59.100.20 protocol tcp port 443
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure the SSL policy for the specified virtual server:

```
ssl-proxy (config-ssl-proxy)# virtual policy ssl sslpl1
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure the TCP policy for the specified virtual server:

```
ssl-proxy (config-ssl-proxy)# virtual policy tcp tcppl1
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure a clear-text web server for the SSL Services Module to forward the decrypted traffic:

```
ssl-proxy (config-ssl-proxy)# server ipaddr 207.50.0.50 protocol tcp port 80
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure a TCP policy for the given clear-text web server:

```
ssl-proxy (config-ssl-proxy)# server policy tcp tcppl1
ssl-proxy (config-ssl-proxy)#
```

This example shows how to configure a NAT pool for the client address used in the server connection of the specified service SSL offload:

```
ssl-proxy (config-ssl-proxy)# nat client NP1
ssl-proxy (config-ssl-proxy)#
```

This example shows how to enable a NAT server address for the server connection of the specified service SSL offload:

```
ssl-proxy (config-ssl-proxy)# nat server
ssl-proxy (config-ssl-proxy)#
```

Related Commands

show ssl-proxy service

ssl-proxy ssl ratelimit

To prohibit new connections during overload conditions, use the **ssl-proxyy ssl ratelimit** command. Use the **no** form of this command to allow new connections as long as memory is available.

ssl-proxyy ssl ratelimit

no ssl-proxyy ssl ratelimit

Syntax Description

This command has no arguments or keywords.

Defaults

This command has no default settings.

Command Modes

Global configuration

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Examples

This example shows how to prohibit new connections during overload conditions:

```
ssl-proxy (config)# ssl-proxy ssl ratelimit
ssl-proxy (config)#
```

This example shows how to allow new connections during overload conditions as long as memory is available:

```
ssl-proxy (config)# no ssl-proxy ssl ratelimit
ssl-proxy (config)#
```

ssl-proxy vlan

To enter the proxy-VLAN configuration submode, use the **ssl-proxy vlan** command. In proxy-VLAN configuration submode, you can configure a VLAN for the SSL Services Module.

ssl-proxy vlan *vlan*

Syntax Description

vlan	VLAN ID; valid values are from 1 to 1005.

Defaults

This command has no default settings.

Command Modes

Global configuration mode

Command History

Release	Modification
Cisco IOS Release 12.1(13)E and SSL Services Module Release 1.1(1)	Support for this command was introduced on the Catalyst 6500 series switches.

Usage Guidelines

VLAN 1 is not supported by the CSM.

Extended range VLANs are not supported by the SSL Services Module.

Each proxy-VLAN configuration submode command is entered on its own line.

Table B-6 lists the commands available in proxy-VLAN configuration submode.

Table B-6 Proxy-service Configuration Submode Command Descriptions

Syntax	Description
admin	Configures the VLAN to be an administration VLAN.
exit	Exits from the proxy-VLAN configuration submode.
gateway *prefix* [**drop** | **forward**]	Configures the VLAN with a gateway to the Internet.
help	Provides a description of the interactive help system.
ipaddr *prefix mask*	Configures the VLAN with an IP address and a subnet mask.
no	Negates a command or set its defaults.
route {*prefix mask*} {**gateway** *prefix*}	Configures a gateway for the SSL Services Module to reach a nondirect connected subnetwork.

You must remove the administration VLAN status of the current administration VLAN before you can configure a different administration VLAN.

An administration VLAN is used for communication with the certificate agent (PKI) and the management station (SNMP).

When configuring the gateway, the **drop** option allows the SSL Services Module to drop a packet if a virtual service cannot be found relating to the packet.

When configuring the gateway, the **forward** option allows the SSL Services Module to forward a packet to the gateway of the specified VLAN, if a virtual service cannot be found relating to the packet.

Examples

This example shows how to enter the proxy-VLAN configuration submode:

```
ssl-proxy (config)# ssl-proxy vlan 6
ssl-proxy (config-vlan)#
```

These examples show how to set a specified command to its default value:

```
ssl-proxy (config-vlan)# default admin
ssl-proxy (config-vlan)# default gateway
ssl-proxy (config-vlan)# default ipaddr
ssl-proxy (config-vlan)# default route
```

This example shows how to configure the specified VLAN with a gateway:

```
ssl-proxy (config-vlan)# gateway 209.0.207.5
ssl-proxy (config-vlan)#
```

This example shows how to configure the specified VLAN with an IP address and subnet mask:

```
ssl-proxy (config-vlan)# ipaddr 208.59.100.18 255.0.0.0
ssl-proxy (config-vlan)#
```

This example shows how to configure a gateway for the SSL Services Module to reach a nondirect connected subnetwork:

```
ssl-proxy (config-vlan)# route 210.0.207.0 255.0.0.0 gateway 209.0.207.6
ssl-proxy (config-vlan)#
```

Related Commands

show ssl-proxy vlan

APPENDIX C

System Messages

This appendix provides the list of system log messages supported in the SSL Services Module.

Error Message `STE-2-IPC_HEALTH_PROBE: [chars]`

Explanation This message indicates that the system did not receive a health probe response from the specified modules.

Recommended Action No action is required. The system resets itself automatically. If you continue to see this message after the system resets itself, contact your Cisco technical support representative.

Error Message `STE-2-IPC_HEALTH_PROBE_HEAD: The following modules failed to respond to a health probe.`

Explanation This message indicates that the system did not receive a health probe response from the specified modules.

Recommended Action No action is required. The system resets itself automatically. If you continue to see this message after the system resets itself, contact your Cisco technical support representative.

Error Message `STE-2-IPC_HEALTH_PROBE_TAIL: Declaring the module dead.`

Explanation This message indicates that the system did not receive a health probe response from the specified modules.

Recommended Action No action is required. The system resets itself automatically. If you continue to see this message after the system resets itself, contact your Cisco technical support representative.

Error Message `STE-3-APP_IPC_STATUS_FAILED: Module (APP) got a response with status failed.`

Explanation This message indicates that the module could not process the inter-process communications (IPC) message.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, try rebooting the module to eliminate the problem.

Error Message `STE-3-CRYPTO_IPC_FAILED: Failed to send IPC message to SSL Processor: [chars] [dec]`

Explanation This message indicates that the cryptographic module encountered an error when sending an IPC message to one or more SSL processors.

Recommended Action Cancel and reenter the command. If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information.

Error Message `STE-3-FDU_IPC_BUFFER_ALLOC_FAILED: Module (FDU) failed to get a buffer to send a IPC message.`

Explanation This message indicates that the system failed to allocate a buffer to send IPC messages.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, reboot the module.

Error Message `STE-3-IPC_BUFFER_ALLOC_FAILED: Module (IPC) failed to get a buffer to send a IPC message.`

Explanation This message indicates that the module is in a transient state or that a command failed.

Recommended Action If this message is related to the CLI, reenter the command. If this situation affects the functionality of the module, contact your Cisco technical support representative.

Error Message `STE-3-IPC_INVALID_MID: IPC received a message with a invalid destination module id [dec]`

Explanation This message indicates that a source module ID is not registered to receive IPC messages.

Recommended Action If this situation affects the functionality of the module, contact your Cisco technical support representative.

Error Message `STE-3-IPC_INVALID_TYPE: IPC received a message with a invalid type [dec]`

Explanation This message indicates that the system might have received a message that was not intended for it.

Recommended Action If this situation affects the functionality of the module, contact your Cisco technical support representative.

Error Message `STE-3-IPC_NULL_RECEIVE_METHOD: IPC module received a message with NULL callback.`

Explanation This message indicates that IPC received a message that does not have a valid callback set for it.

Recommended Action If this situation affects the functionality of the module, contact your Cisco technical support representative.

Error Message `STE-3-IPC_NULL_RECEIVE_QUEUE: IPC module received a message with method QUEUE but queue is NULL.`

Explanation This message indicates that IPC received a message that does not have a valid queue set for it.

Recommended Action If this situation affects the functionality of the module, contact your Cisco technical support representative.

Error Message `STE-3-IPC_SEND_FOR_DATE_FAILED: Module (IPC) failed to send a IPC message to get date and time.`

Explanation This message indicates that the daughter card is unable to synchronize with the clock on the supervisor engine because of a failure in the control channel. This situation sometimes occurs during bootup.

Recommended Action Set the clock manually by entering the **set clock** command.

Error Message `STE-3-PKI_CERT_INSTALL_FAILED: Failed to install a certificate chain, trustpoint: [chars], proxy service: [chars], index: [dec]`

Explanation This message indicates that the public key infrastructure (PKI) module failed to install a certificate chain for the specified proxy service. This error might be due to an unsupported key type or size.

Recommended Action Check the configuration and state of the key pair associated with the trustpoint assigned to the specified proxy service. Correct the key type or size, and reenroll the certificate. Remove the trustpoint assigned to the proxy service, and reassign it. If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information.

Error Message `STE-3-PKI_CERT_ROLLOVER_FAILED: The process of rolling over the certificate without the sudden loss of services has failed for the proxy service [chars], trustpoint [chars]`

Explanation This message indicates that the rollover process cannot be completed because of an error that was encountered when installing the new certificate. This error might be due to an unsupported key type or size.

Recommended Action Check the current configuration and state of the key pair associated with the trustpoint assigned for the proxy service. Correct the key type or size, and reenroll the certificate. Remove the trustpoint assigned to the service, and reassign it. Enter the **show ssl-proxy service** command to display information about keys and certificates associated with the proxy service.

Error Message `STE-3-PKI_INVALID_IPC_MSG: Invalid PKI IPC messages: [chars]`

Explanation This message indicates that the public key infrastructure module received an invalid IPC message.

Recommended Action If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information.

Error Message `STE-3-PKI_IPC_FAILED: Failed to send IPC message to SSL Processor: [chars] [chars] [dec]`

Explanation This message indicates that the public key infrastructure module encountered an error when the module sent an IPC message to one or more SSL processors.

Recommended Action Remove the certificate that is assigned to the proxy services. Reassign the certificate to trigger IPC again. If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information.

Error Message `STE-3-PKI_KEY_INSTALL_FAILED: Failed to install a key pair: [chars], trustpoint: [chars], proxy service: [chars], index: [dec]`

Explanation This message indicates that the Public Key Infrastructure module failed to install a key pair for the specified proxy service.

Recommended Action Check that the key pair of the trust point assigned to the proxy service is in the IOS key chain by entering the **show crypto key mypub rsa** command. Remove the certificate that was assigned to the proxy service. Reassign the certificate to reinstall it. If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information

Error Message `STE-3-PKI_MISCONFIGURED_KEY_TYPE: Trustpoint [chars] key type [chars] does not match type for SSL proxy service.`

Explanation This message indicates that the key type of the trust point must be the same as what was configured for the SSL proxy service.

Recommended Action Regenerate a key pair of the same type configured for the SSL proxy service. Enroll for a new certificate.

Error Message `STE-3-PKI_MISMATCHED_CERT_KEY_TYPE: Certificate key type [chars] does not match type for SSL proxy service [chars].`

Explanation This message indicates that the specified key type of the certificate must be the same as what was configured for the SSL proxy service.

Recommended Action Regenerate a key pair of the same type configured for the SSL proxy service. Enroll for a new certificate.

Error Message `STE-3-PKI_OP_FAILURE: [chars] [chars] [dec]`

Explanation This message indicates that a public key infrastructure operation failed. The failure might have occurred because of a lack of resources.

Recommended Action If this message recurs, copy the error message exactly as it appears on the console or in the system log, contact your Cisco technical support representative, and provide the representative with the gathered information.

Error Message `STE-3-PKI_UNSUPPORTED_KEY_ALGORITHM: Algorithm of key pair [chars] is unsupported.`

Explanation This message indicates that the key algorithm is unsupported. The supported key type is RSA.

Recommended Action Regenerate a key pair of the supported type.

Error Message `STE-3-PKI_UNSUPPORTED_KEY_SIZE: Trustpoint [chars] key size is not supported. Supported sizes are: 512, 678, 1024, 1536, 2048-bit`

Explanation This message indicates that the trust point key size is not supported.

Recommended Action Regenerate a key pair of supported size for the trust point. Enroll for a new certificate.

Error Message `STE-3-PKI_UNSUPPORTED_KEY_TYPE: Trustpoint [chars] key type [chars] is unsupported.`

Explanation This message indicates that the specified key type is unsupported. Supported key types are RSA key pairs and general purpose key pairs.

Recommended Action Regenerate a key pair of a supported type for the trust point. Enroll for a new certificate.

Error Message `STE-3-SSL_IPC_BUFFER_ALLOC_FAILED: Module (SSL) failed to get a buffer to send a IPC message.`

Explanation This message indicates that the system failed to allocate a buffer to send IPC messages.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, try rebooting the module to eliminate the problem.

Error Message `STE-3-SSL_IPC_SEND_FAILED: Module (SSL) failed to send a IPC message because of a lack of resources`

Explanation This message indicates that the system failed to allocate a buffer to send IPC messages.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, try rebooting the module to eliminate the problem.

Error Message `STE-3-TCP_IPC_BUFFER_ALLOC_FAILED: Module (TCP) failed to get a buffer to send a IPC message.`

Explanation This message indicates that the system failed to allocate a buffer to send IPC messages.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, try rebooting the module to eliminate the problem.

Error Message `STE-3-TCP_IPC_STATUS_FAILED: Module (TCP) got a response with status failed.`

Explanation This message indicates that the module could not process the IPC message.

Recommended Action If you see this message when entering a command, reenter the command. If you do not see this message when entering a command, try rebooting the module to eliminate the problem.

Error Message `STE-4-PKI_WEAK_KEY: Trustpoint [chars] key size is weak. Recommended sizes are: 1024, 1536 and 2048-bit`

Explanation This message indicates that the key size is either 512 bits or 768 bits. We recommend stronger keys.

Recommended Action Regenerate a stronger key pair for the trust point and enroll for a new certificate.

Error Message `STE-5-PKI_NO_ENTRY: No free key and certificate table entries. [dec] entries in use.`

Explanation This message indicates that all entries in the proxy service key and certificate table are now in use. New proxy services cannot be supported.

Recommended Action Enter the **show ssl-proxy stats pki** command to display the counters. If long-lived connections still remain after rollover, some entries might still be used by old certificates. Clear the connections and restart the service.

Error Message `STE-5-UPDOWN:ssl-proxy service [chars] changed state to [chars]`

Explanation This message indicates that the SSL proxy service state changed.

Recommended Action No action is required.

Error Message `STE-6-CRYPTO_SELFTEST_RUNNING: Cryptographic self-tests have started to run on the SSL Processor(s).`

Explanation This message indicates that the cryptographic algorithm test cases are running in the background with a time interval of 1 to 8 seconds. These self-tests are run on each cryptographic device in turn. Data traffic performance might be impacted.

Recommended Action Enter the **show ssl-proxy status crypto** command to display test results. These tests are for troubleshooting purposes only. You do not need to continually run these tests in the background.

Error Message `STE-6-CRYPTO_SELFTEST_STATS_CLEARED: Cryptographic self-tests statistics have been cleared.`

Explanation This message indicates that statistics for the cryptographic self-tests have been cleared.

Recommended Action No action is required.

Error Message `STE-6-CRYPTO_SELFTEST_STOPPED: Cryptographic self-tests have stopped to run on the SSL Processor(s).`

Explanation This message indicates that the cryptographic algorithm tests are no longer running on the SSL processor.

Recommended Action No action is required.

Error Message `STE-6-IPC_UNSUPPORTED_VERSION: Unsupported IPC Version number [dec]`

Explanation This message indicates that the system received an IPC message with an invalid version number. Only IPC version 1.0 is supported.

Recommended Action No action is required. IPC retries sending the message. If you continue to see this message, contact your Cisco technical support representative.

Error Message `STE-6-NVRAM_DOWNGRADE_NOT_READY`

Explanation This message indicates that the configuration will not be saved when you downgrade the SSL module software to an earlier version.

Recommended Action If you plan to downgrade the SSL module software to an earlier version, issue the **copy running-config startup-config** command one more time. This action will prepare the configuration for the image downgrade. If you do not plan to downgrade the image, no action is required.

Error Message `STE-6-NVRAM_DOWNGRADE_READY`

Explanation This message indicates that the configuration will be save when you downgrade the SSL module software to an earlier version.

Recommended Action No action is required.

Error Message `STE-6-PKI_CA_CERT_DELETE: [chars], Subject Name: [chars], Serial#: [chars], Index: [dec]`

Explanation This message indicates that a certificate authority certificate was deleted because no proxy services use it.

Recommended Action No action is required. A record of this deletion can be archived for reference or auditing.

Error Message `STE-6-PKI_CA_CERT_INSTALL: [chars], Subject Name: [chars], Serial#: [chars], Index: [dec]`

Explanation This message indicates that a certificate authority certificate was installed for use by proxy services.

Recommended Action No action is required. A record of this certificate authority certificate can be archived for reference or auditing.

Error Message `STE-6-PKI_CERT_HIST_CLEARED: [dec] certificate history records have been cleared from memory.`

Explanation This message indicates that the specified number of certificate history records were cleared from the system memory.

Recommended Action No action is required.

Error Message `STE-6-PKI_CERT_HIST_DISABLED: Certificate history of proxy services has been disabled.`

Explanation This message indicates that the proxy service certificate history function was disabled. Certificate installation and deletion records will be cleared from memory. No new history records will be written into memory.

Recommended Action No action is required.

Error Message `STE-6-PKI_CERT_HIST_ENABLED: Proxy Service Certificate History has been enabled.`

Explanation This message indicates that the proxy service certificate history function was enabled. Certificate installation and deletion records will be written into memory.

Recommended Action Enter the **show ssl-proxy certificate-history** command to display certificate history records. Save the output of this command to a file for archiving.

Error Message `STE-6-PKI_CERT_HIST_RECORD_THRESHOLD: [dec] certificate history records have been logged to memory\n. Maximum of [dec] can be logged before the oldest ones are overwritten.`

Explanation This message indicates that there is maximum number of certificate history records that can be saved to memory. The maximum number will be reached soon. Older records will be overwritten.

Recommended Action Enter the **show ssl-proxy certificate-history** command to display certificate history records. To prevent the loss of older records, save the output of this command to a file for archiving.

Error Message `STE-6-PKI_CERT_ROLLOVER_BEGIN: The process of rolling over the certificate without the sudden loss of services has begun for the proxy service: [chars], trustpoint: [chars]`

Explanation This message indicates that the key pair, the certificate, or the trustpoint assigned to the specified proxy service has been modified. Until the new certificate is received, the old certificate will be used.

Recommended Action Finish the rollover process by enrolling or importing the modified trustpoint. Enter the **show ssl-proxy service** command to display information about certificates, key pairs, and trustpoints associated with the specified proxy service.

Error Message `STE-6-PKI_CERT_ROLLOVER_END: The process of rolling over the certificate without the sudden loss of services has ended for the proxy service: [chars], trustpoint: [chars]`

Explanation This message indicates that a new certificate has been received for the specified proxy service. The old certificate will be deleted when all connections using it are finished.

Recommended Action No action is required. Enter the **show ssl-proxy service** command to display more information about new and old certificates.

Error Message `STE-6_PKI_SERVER_CERT_DELETE: Proxy: [chars], Trustpoint [chars], Key [chars], Serial#: [chars], Index: [dec]`

Explanation This message indicates that a certificate was deleted for a proxy service.

Recommended Action No action is required. A record of this deletion can be archived for reference or auditing.

Error Message `STE-6-PKI_SERVER_CERT_INSTALL: Proxy: [chars], Trustpoint: [chars], Key: [chars], Serial#: [chars], Index: [dec]`

Explanation This message indicates that a certificate was installed for a proxy service.

Recommended Action No action is required. A record of this certificate can be archived for reference or auditing.

Error Message `STE-6-PKI_TEST_CERT_INSTALL: Test key and certificate was installed into NVRAM in a PKCS#12 file.`

Explanation This message indicates that a PKCS12 file, containing a key pair and a certificate chain that can be used for testing purposes, was copied from memory into the NVRAM device.

Recommended Action No action is required.

Error Message `STE-7-IPC_REQUEST_RESPONSE_MISMATCH: IPC module received a message where the request and response do not match.`

Explanation This message indicates that IPC received a message that does not have a corresponding valid request.

Recommended Action If this situation is impacting the functionality of the module, contact your Cisco technical support representative.