

NetCache™ 5.4 Security Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: <http://www.netapp.com>

Part number 210-00370
December 2002

Copyright and trademark information

Copyright information

Copyright © 1994–2002 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

Portions copyright © 1998–2001 The OpenSSL Project. All rights reserved.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980–1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California and Carnegie Mellon University is subject to the following license and disclaimer:

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

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of the software were created by Netscape Communications Corp.

The contents of those portions are subject to the Netscape Public License Version 1.0 (the “License”); you may not use those portions except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/NPL/>.

Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is Mozilla Communicator client code, released March 31, 1998.

The Initial Developer of the Original Code is Netscape Communications Corp. Portions created by Netscape are Copyright © 1998 Netscape Communications Corp. All rights reserved.

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use and purchase of this product do not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the European Union and Japan, and a trademark of Network Appliance, Inc. in the United States. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. SnapManager is a registered trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. FilerView and SecureShare are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp—the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, DataFabric, MultiStore, NearStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapMirror, SnapRestore, SnapVault, Vfiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries.

Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries.

RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

	Preface	vii
Chapter 1	Authenticating NetCache Users	1
	About LDAP and RADIUS.	16
	LDAP	17
	RADIUS	22
	About NTLM and Kerberos	24
	Authenticating with NTLM.	27
	Authenticating with Kerberos	30
	Joining the domain using DNS	34
	Joining the domain without using DNS	36
	Joining the domain using WINS	41
	Joining the domain without using WINS.	44
Chapter 2	Access Control	47
	About access controls.	49
	ACL building blocks	52
	Regular expressions.	61
	About setting restrictions	67
	ACL optimization.	69
	About controlling access	72
	About controlling access to streaming media	76
Chapter 3	HTTPS	79
	About HTTPS.	81
	Authentication	83
	Controlling access	85
	HTTPS acceleration rules.	87

	Planning your deployment	89
	HTTPS deployment scenarios	90
Chapter 4	WebWasher DynaBLocator Content Filtering	93
	About DynaBLocator.	94
	About the DynaBLocator URL database	96
	DynaBLocator configuration overview.	98
	Basic DynaBLocator options	100
	DynaBLocator URL database updates	101
	DynaBLocator blocking with ACLs	102
	About testing and troubleshooting DynaBLocator	105
	About rolling out DynaBLocator service to your users	106
Chapter 5	SmartFilter Content Filtering	107
	About SmartFilter.	108
	About SmartFilter control files	110
	SmartFilter configuration overview	112
	Basic SmartFilter options.	114
	SmartFilter control file updates.	115
	SmartFilter blocking with ACLs	116
	About testing and troubleshooting SmartFilter	119
	About rolling out SmartFilter service to your users	120
	Index	121

Preface

About this guide

This guide describes how to configure and optimize appliance security. It provides information for the following supported NetCache™ platforms: C1100, C1105, C1200, C2100, C3100, and C6100.

For information about how to upgrade or service the system hardware, see the *Field Service Guide* for your NetCache appliance model.

Audience

This guide is for system administrators who are familiar with computer networks, Web technology, and HTTP service. This guide assumes that customers planning to use the NetCache appliance as a news cache have a basic understanding of NNTP and news service. This guide also assumes that customers planning to use the NetCache appliance as a streaming media cache have a basic understanding of RTSP, MMS, and streaming media service. This guide does not cover basic system administration topics such as IP addressing, routing, and network topology. It emphasizes the characteristics of the NetCache appliance hardware and the NetCache software.

Command and keyboard conventions

You can enter NetCache appliance commands on either the system console or from any client computer that can access the NetCache appliance through Telnet. When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, “Ctrl-D” means pressing the “Control” and “D” keys simultaneously.

This guide uses the term “type” to mean pressing one or more keys on the keyboard. It uses the term “enter” to mean pressing one or more keys and then Enter. Also, this guide uses the term “Enter” to refer to the key that generates a carriage return, although the key is named “Return” on some keyboards.

This guide uses capitalization and some abbreviations to refer to the keys on the keyboard. (The keys on your keyboard might not be labeled exactly as they are in this guide.)

Formatting conventions

The following table describes typographical conventions used in this guide.

Formatting convention	Type of information
<i>Italic type</i>	Words or characters that require special attention. Placeholders for information you must supply. For example, if the guide says to enter the <code>arp -d hostname</code> command, you enter the characters “arp -d” followed by the actual name of the host. Book titles in cross-references.
Monospaced font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
Bold monospaced font	Words or characters you type. What you type is always shown in lowercase letters, unless it must be typed in uppercase letters to work properly.

About this chapter	This chapter describes NetCache client authentication and the supported methods for authenticating administrators and users of NetCache services. It also explains how you can join a NetCache appliance to a Microsoft Windows 2000 or Microsoft Windows NT domain if you want to join the NetCache to these domains without using DNS or WINS.
Topics in this chapter	This chapter provides information about the following topics: ◆ Section A, “About authentication,” on page 3 ◆ Section B, “Authenticating with the NetCache appliance,” on page 6 ◆ Section C, “Authenticating with the NetCache user database,” on page 10 ◆ Section D, “Authentication,” on page 12 ◆ Section E, “Authenticating users with LDAP or RADIUS servers,” on page 15 ◆ Section F, “Authenticating with NTLM and Kerberos,” on page 23 ◆ Section G, “Joining a Microsoft Windows 2000 domain,” on page 33 ◆ Section H, “Joining a Microsoft Windows NT domain,” on page 40
Procedures and option descriptions	See the Appliance Manager online help for configuration procedures and option descriptions.
Location of configuration options	Options for configuring authentication are in the following location in the Appliance Manager: Setup tab > Authentication
About using the CLI for configuration	This chapter provides conceptual and background information on features of the Appliance Manager that enable you to configure authentication on the appliance. For descriptions of CLI configuration commands, see the <i>Command Line Reference</i>. To list the authentication configuration commands from the CLI and obtain information about them, enter the following command: <pre>show config.auth.?</pre>

To get help for a specific authentication configuration option, enter the following command:

help config.auth.*option_name*

Section A: About authentication

About this section

This section introduces authentication services provided by the NetCache appliance. It gives a brief description of authentication and general information that is useful when planning NetCache client authentication and access to NetCache services.

What is authentication?

Authentication is the process of determining a user's identity so that the user becomes known to the system. NetCache allows you to require authentication for Web requests from clients using certain protocols, such as HTTP, HTTPS, FTP, NNTP, MMS, RTSP, and so on. NetCache also allows you to specify how that authentication is to be performed—for example, against the local NetCache user database, against a RADIUS database, or using the Microsoft Windows NTLM protocol server.

In addition to verifying a user's identity, authentication can be used in conjunction with access control to enforce local security domains, allocate resources, control bandwidth usage, and keep records of traffic and filter requests based on content.

Clear-text authentication

Clear-text authentication is performed using a base64-encoded user name and password information pair intended to identify the user. The pair is verified against information stored in a database or on a server.

Base64-encoding is intended to be easily decoded. For this reason, it is referred to as clear text.

You can configure NetCache to provide clear-text authentication for users against any of the following databases or servers: the NetCache user database, an LDAP server, a RADIUS server, and a Microsoft Windows domain controller.

Caching credentials

For clear-text authentication, after NetCache authenticates a user against an external server, such as LDAP or RADIUS or clear-text NTLM, each cached credential remains in the cache for one hour, by default. You can configure this caching Time-To-Live (TTL) value, increasing or decreasing it. Using the

Authentication Server Cache TTL option on the Authentication - General page in the Appliance Manager, you can modify the frequency of cache interaction with an authentication server to authorize users.

For true NTLM, NetCache allows you to enable caching of NTLM challenges for reuse to reduce the processing load in the Microsoft Windows domain controller. To minimize the security risks that could result from caching the NTLM challenge, NetCache expires the challenge after a period of time, gradually phasing in a new challenge. Security precautions are added to the challenge; the client's IP address is used to render the challenge unique making replay attacks more difficult.

Here are the two main benefits of caching credentials on the appliance:

- ◆ It speeds up NetCache response time. Because the appliance does not need to engage the external database yet again in the authentication process, it can respond to user requests more quickly.
- ◆ It reduces the workload on the authentication server, which, when overloaded, can become a bottleneck.

The only case in which NetCache fetches the credentials from the authentication database for a user whose credentials are presently cached is when a user passes credentials that do not match those cached for an account.

Planning client authentication

NetCache is permissive by default, which means access to all protocols (HTTP, FTP, and so on) is allowed and authentication is not required (enabled) for any protocols. When you are planning client authentication, consider

- ◆ For which protocols you want to require authentication
Consider whether you want to require users to supply a user name and password when requesting URLs over any or all of the following protocols: HTTP, HTTP-based tunneling (SSL), FTP, Gopher, NNTP, MMS, and RTSP.
- ◆ Which supported method, or methods, of authenticating users best suits your environment
- ◆ Which user database is to be queried first when authenticating users, if you support multiple authentication checking methods

Methods for authenticating users

NetCache supports various methods for authenticating users and provides options for regulating access to NetCache services. Supported methods for authenticating users are described in the following table.

Method	Description
NetCache user database	The native NetCache user database provides local authentication for users and administrators. (clear-text authentication)
LDAP authentication server	Lightweight Directory Access Protocol (LDAP) databases are commonly used as employee directory databases. You can enable NetCache to retrieve user and group data from an existing LDAP server to perform clear-text authentication.
RADIUS authentication server	Remote Authentication Dial-in User Service (RADIUS) was originally used to authenticate people logging in to the network through a modem to remote points of presence (POPs). You can enable NetCache to retrieve user data from an existing RADIUS server to perform clear-text authentication. Note_____ RADIUS does not support groups.
NTLM (NT LAN Manager)	NTLM supports NT domain access to the Microsoft Windows authentication environment. NTLM (in true mode) performs authentication using an encrypted challenge-and-response sequence between NetCache and a Windows domain controller. NTLM can be used for clear-text authentication when used with a browser other than Microsoft Internet Explorer.
Kerberos (Windows 2000)	Kerberos version 5 is a new authentication protocol for Windows 2000 domain access. It is an industry standard developed by Massachusetts Institute of Technology (MIT) that offers mutual authentication to client and server. The Kerberos standard is described in Request for Comments (RFC) 1510. Kerberos authentication uses a private key cryptography referred to as a shared secret key distribution model in which NetCache validates tickets presented by the client (user). The key is considered symmetrical because both client (NetCache) and server (Windows 2000 DC running Kerberos) must prove that they know the shared secret; one party encrypts a message with the key, and the other party decrypts it.

Section B: Authenticating with the NetCache appliance

About this section This section provides an overview of the NetCache authentication process and explains some of the processes entailed in authenticating a user.

When a connection authenticates When a client attempts to authenticate with the NetCache appliance, NetCache carries out the authentication sequence shown in the following table (this verification process also entails authorization through access control).

Note
In HTTP, only true NTLM and Kerberos are authenticated per connection. Clear-text authentication requests are authenticated per request. (Other protocols might work differently.)

Stage	Description
1	Checks protocol authentication settings (authenticates the user).
2	Checks ACL permission settings for the applicable group in the NetCache user database.
3	Checks the request-type (protocol specific) ACL.
4	Checks the accelerator ACL.
5	Checks the global ACL.
6	Verifies whether SmartFilter or WebWasher DynaBLocator denies the request.
7	Checks group permissions settings.
8	Allows or disallows the request, based on the results of the authentication process.

When does NetCache authenticate a user?

NetCache authenticates a user for the first time under these circumstances:

- ◆ When the user makes a request using a protocol, such as HTTP, for which authentication is enabled
- ◆ When ACLs specifying authentication are applied, forcing authentication

If a user makes a request using a protocol for which authentication is not enabled, NetCache does not attempt to authenticate the user, even if authentication is enabled for other protocols. However, if the same user later uses a protocol that requires authentication, NetCache validates that user.

For example, consider two user sessions through a appliance that is configured as follows:

- ◆ FTP authentication is enabled.
- ◆ LDAP authentication is configured and enabled.
- ◆ HTTP authentication is not enabled.

Example 1: If a user makes an FTP request, NetCache prompts the user for a user name and password, then authenticates the user to the LDAP server. Only after the user is authenticated does NetCache service the request.

Example 2: If a user makes an HTTP request, NetCache does not authenticate the user; NetCache just services the request. However, if the same user then makes an FTP request, NetCache prompts the user for a user name and password before servicing the request. For example, if authentication is enabled for HTTP, any HTTP client requests must be authenticated.

Types of authentication

The following table shows two primary categories of authentication.

Authentication type	How it works
Clear text	If a user makes a request that requires clear-text authentication, NetCache requests a user name and password and authenticates the user to the appropriate authentication server.
NTLM (for Microsoft Windows 2000)	NetCache undertakes an NTLM authentication handshake.

Authentication type	How it works
Kerberos (for Microsoft Windows 2000)	NetCache participates in the exchange of encrypted information based on a key (shared secret) used to authenticate both parties.

Authentication checking order

NetCache allows you to specify the order in which it should check authentication resources for clear-text authentication if your environment includes more than one clear-text authentication method. You set the Authentication Checking Order list on the Authentication - General page to specify the clear-text authentication resources that you want the appliance to use.

This process does not apply to use of the NTLM or Kerberos protocols in the Microsoft Windows 2000 environment. They are not included in the checking order for authentication methods.

Note

You should always include “local” in the authentication checking order, preferably first. Neglecting to do so can cause various administrative lockouts and delays.

NetCache carries out the process shown in the following table.

Stage	Description
1	NetCache prompts each user for a user name and password when the user attempts a request using a client protocol for which authentication is enabled (or when ACLs specifying authentication are applied, forcing authentication).
2	NetCache attempts to authenticate the user against the first method in the list.
3	If the first authentication attempt results in failure, NetCache tries the next authentication method in the list, and so on, until it succeeds in authenticating the user or has exhausted all possible authentication methods.

Note

The authentication order does not apply to Kerberos or true NTLM. Rather, it is used for clear-text authentication only.

A user is authenticated against the various databases following the preconfigured order. When authentication succeeds (that is, a match is found), any groups that a user belongs to are determined.

Section C: Authenticating with the NetCache user database

About this section

This section describes the NetCache user database, which you can use to authenticate users. In addition, this section describes how to add users to the database, create groups and add users to them, and create administrator accounts.

About the NetCache user database

The NetCache user database contains local user and group information that provides NetCache with instructions for handling client requests.

You can use the NetCache user database in many different ways, depending on the requirements of your organization. For example,

- ◆ If it is used to contain user information, the NetCache user database can provide Web protocol client authentication (by protocol) for users.
- ◆ If it is used to contain system administrator user account information, it provides administrative access into the appliance if other configured methods of authentication in your environment are unavailable due to system or network failure problems.
- ◆ Because it can contain group information, it can be used to assign user access permissions based on group membership.

Note

In a load-balanced NetCache environment that includes many appliances, for the NetCache local database to be effective as an authentication resource, its contents must be synchronized across all appliances. In this case, you must maintain the user database on each appliance separately. If the databases on cooperating appliances become unsynchronized, some users might be denied access to the Internet. To synchronize the content of the NetCache local database across all NetCache appliances, you can use central configuration management or Data Fabric Manager.

Authenticating administrators

System administrators cannot access the Appliance Manager unless the NetCache user database contains at least one user name and password. NetCache provides a default administrator account and a default NetCache group. The account user name is `admin` and the password is `NetCache`.

For the default system administrator, the default group is named “NetCache.” Because the administrator user belongs to the NetCache group, all permissions assigned to the NetCache group apply to the default administrator account.

The default admin user account is configured with permissions to use all protocol services, as well as permissions to access the Appliance Manager and change the NetCache configuration.

Note

The administrator user account user name and password are case-sensitive.

Using the NetCache options for enabling protocol authentication for users, you can create profiles with different levels of access for NetCache administrators. For example, you can allow some administrators to monitor the status of NetCache, but prevent them from making configuration changes. You can apply different ACLs to different administrator accounts that you create.

Note

Do not rely on an external database to provide authentication for administrative users. If these settings are provided for NetCache through an LDAP or RADIUS authentication server, or the NTLM or Kerberos protocol, you risk administrative lockout in the event of authentication server or network downtime.

Authenticating users and groups of users

If you choose to authenticate using the NetCache user database, then user names, passwords, group memberships, and protocol permissions are assigned using the authentication user and group options in the Appliance Manager. Using these options, you can do the following:

- ◆ You can individually add users.
Authenticated users who are not members of any groups are assigned default access permissions. There is only one set of default access permissions. Permissions cannot be modified on a per-user basis.
- ◆ You can create groups, specify group memberships for users, and set access permissions for the group.
- ◆ You can allow a user to be a member of one group or multiple groups. You can set and modify group access permissions on a per-group basis.

Grouping users is common practice and makes administering access to protocols less complicated than administering access by individual user. For example, NetCache can be configured to authorize access to FTP for users in the Engineering group while prohibiting access to FTP for the Marketing group.

Section D: Authentication

About this section

This section provides information about authenticating users, including how to navigate to the Appliance Manager pages that allow you to configure authentication options, how to test which type of authentication (clear-text, or Microsoft Windows NTLM or Kerberos) the NetCache appliance is using, and where authentication information is logged. This section also describes how proxy authentication for streaming clients is performed.

Where to configure authentication

Authentication options are located on the Authentication pages of the Appliance Manager. To access the Authentication pages, select Setup tab > Authentication.

The options on the Authentication pages enable you to

- ◆ Enable or disable requirement of client authentication per supported protocol
- ◆ Create groups, add users, and edit user and group information for the local NetCache user database
- ◆ For clear-text authentication, configure the authentication checking order that NetCache is to follow in attempting to authenticate users if your environment supports more than one method (for example, the local NetCache user database, LDAP, and NTLM)
- ◆ Set default protocol access permissions for authenticated users who are not members of any group in the NetCache user database
- ◆ Specify the authentication realm to be used for clear-text authentication and specify the length of time that user credentials are held in the cache
- ◆ For clear-text authentication, enable or disable support for transparent proxy authentication
- ◆ Specify the origin servers to which to forward proxy authentication credentials
- ◆ Configure an LDAP server or a RADIUS server for NetCache authentication, if you use these databases
- ◆ Enable and configure NTLM and Kerberos protocol support for the Microsoft Windows authentication environment

Namespace issues and other concerns

Using multiple authentication methods (databases) entails complexity, which can introduce problems. For example, sometimes a company transitions from one database to another, or sometimes the appliance is configured above multiple authentication domains that use different techniques. In the latter example, duplicate user names and group names can occur. Take these considerations into account, for example, if your company is transitioning from use of an LDAP server to exclusive use of the NetCache local user database for authentication or if you are using a combination of the two methods.

Testing the authentication server

To test whether NetCache is authenticating users using a clear-text authentication server, use your browser to access a Web site you prefer. If NetCache is configured correctly, you will be prompted for user name and password information.

Where authentication server information is logged

NetCache logs general information pertaining to authentication in the Messages log file. For example, if repeated authentication failures occur, NetCache logs a simple, explicit message noting that there were many authentication failures. You can use information such as this to assess if there might be a communication problem between the NetCache and the authentication server rather than a true failed attempt to authenticate the user.

NetCache logs this event as a connectivity error. Notification of this event in the log can be of importance to you as forewarning that a link is down between the NetCache appliance and the server.

Proxy authentication for streaming clients

The Authenticate Client Requests Using These Protocols option on the Authentication - General page in the Appliance Manager includes an Authenticate (MMS and RTSP) Requests item. This item allows you to enable proxy authentication for streaming clients. RTSP clients (RealPlayer, Quicktime Player) use Basic authentication only. MMS clients can use NTLM authentication.

If a streaming server requires client authentication, NetCache establishes a connection to the streaming server and forwards the authentication information given by the client. NetCache waits until the streaming server verifies the authentication request before delivering the stream to the client.

NetCache forwards the authentication information for *every* client, even if a client's request can be resolved by splitting an existing stream already authenticated for another client or by serving a stream from the cache.

If NetCache subsequently delivers a split stream to the client, NetCache closes the connection to the streaming server after the client has been authenticated. VOD (video on demand) connections are not terminated.

Section E: Authenticating users with LDAP or RADIUS servers

About this section

This section describes using an LDAP or a RADIUS server to authenticate users. Configuring one of these supported clear-text methods for authentication enables NetCache to retrieve user information from a preexisting database, which significantly reduces the workload of maintaining the NetCache user database.

Topics in this section

This section contains the following topics:

- ◆ [“About LDAP and RADIUS”](#) on page 16
- ◆ [“LDAP”](#) on page 17
- ◆ [“RADIUS”](#) on page 22

About LDAP and RADIUS

If you are using an LDAP or RADIUS server

If you have installed and configured an LDAP or RADIUS server, you can enable and use the service to authenticate users for NetCache. LDAP and RADIUS provide alternative clear-text authentication methods to the local NetCache user database. Additionally, many enterprises already have user databases established on these servers and prefer to continue using them. You can point NetCache to the RADIUS or LDAP server so that NetCache uses that server's database for authentication.

Using an existing LDAP or RADIUS server to authenticate users for NetCache reduces the workload of maintaining a user database for NetCache. This approach eliminates the need to add users individually to the NetCache user database and replicate additions and changes across discrete appliances to keep them synchronized.

Note

Always maintain at least a single administrator user account on the local NetCache user database as a back door into the appliance. Also, you should always include “local” in the authentication checking order, preferably first. Neglecting to do so can cause various administrative lockouts and delays.

LDAP

About LDAP

The Lightweight Directory Access Protocol (LDAP) allows for databases to be designed according to arbitrary, extensible schemas (ways of specifying data). There is no official schema regulating data definition for purposes of authentication. Therefore, LDAP databases are varied in design, and they can contain any kind of data.

LDAP and NetCache authentication: LDAP databases are commonly used as central repositories for employee directories. For example, LDAP can easily implement a directory of all employees in a company, including their phone numbers, mail stop numbers, addresses, passwords, HR information, as well as the management structure of the company.

When a client (user) begins to use a NetCache appliance configured to require LDAP authentication, NetCache carries out the process shown in the following table.

Stage	Description
1	The user is prompted for a user name and password at the outset of the session. Typically, the password is transmitted from the browser to the appliance through a clear-text credential (also referred to as a basic credential).
2	NetCache searches the LDAP directory for the user account.
3	NetCache sends the password to the LDAP server for verification.
4	NetCache searches the directory for the user’s group memberships.
5	Given the user and group information, the appliance consults the ACLs to determine if any apply.
6	If the ACLs grant the user's request, the data is returned to the user, and the log is updated accordingly (whether the request is allowed or denied).

About LDAP database contents: An LDAP database contains objects, each referred to by its own name distinguishing it from other objects in the database, and each object can contain attributes and values. The object’s name is referred to as a distinguished name (DN). The DN is normally shown on the first line of each object.

About the LDAP search: The tree analogy is commonly used to describe LDAP databases. In the case of NetCache requests, LDAP searches the directory for user authentication information. The location in the LDAP tree where the search begins is called the base. The base applies only to the DN. (This value is set in the Base Distinguished Name option on the Authentication - LDAP page.)

Requiring password renewal

If a NetCache user’s LDAP password expires and requires manual renewal, you can use the Expired Password URL option on the Authentication - General page to specify a URL that displays the requirement to reset an expired password.

Preparing to configure

Before you begin to configure LDAP authentication in NetCache, verify the following information:

- ◆ The LDAP server IP address is correct.
- ◆ Your LDAP server and the appliance are positioned on your network so that NetCache can open a TCP connection.
- ◆ The server port to which NetCache connects to the server; the LDAP default port number is 389.
- ◆ A user name and password are configured in the LDAP database.

NetCache requirements specific to LDAP

NetCache must be configured with specific LDAP server information, which is listed in the following table. (You can set this information through the Setup > Authentication > LDAP page in the Appliance Manager.)

LDAP parameter	Description
Base Distinguished Name	This parameter specifies the directory on the LDAP server that NetCache is to search. For example: o=domain.com

LDAP parameter	Description
NetCache LDAP Password	Password used to gain access to the bind distinguished name. This password authorizes the NetCache appliance to access and query the LDAP directory.
Bind Distinguished Name	This parameter specifies the bind DN that NetCache uses to identify itself to the LDAP server. For example: <code>cn=netcache, o=domain.com</code>
Protocol version	This parameter specifies which version of the LDAP protocol will be used by the appliance in communicating with the server.
User ID (UID) Field in the LDAP Directory	This parameter specifies which field in the LDAP directory contains the name that is entered by the user. For example: <code>uid</code>
Group ID (GID) Field in the LDAP Directory	This parameter specifies which field in the LDAP directory contains group membership information. For example: <code>gid</code>

LDAP parameter	Description
GroupOfUniqueNames	If the LDAP users are included as members of a GroupOfUniqueNames object in the LDAP directory, this option allows you to assign NetCache group membership to them based on a specified attribute in that GroupOfUniqueNames object. If the GroupOfUniqueNames object in the LDAP directory includes the following attributes, you can set the value of this option to “cn” to assign the group to “Coders”: DN: cn=Coders, ou=Groups, o=xyz_corp cn: Coders objectclass: GroupOfUniqueNames uniquemember: uid=diller, ou=development, o=xyz_corp uniquemember: uid=williams, ou=sales, o=xyz_corp NetCache will retrieve the value of the “cn” attribute from the GroupOfUniqueNames object and assign it as a group value to all the users listed as members in the GroupOfUniqueNames object. NetCache can then apply ACL rules, SmartFilter settings, or protocol associations to this group.
GroupOfUniqueNames Member	

LDAP user configuration example

The following example provides a guideline for configuring an LDAP user file. This file is created and resides on the LDAP server. The example assumes that you have an existing LDAP server configuration and a working knowledge of UNIX.

Note

The attribute values (directories, user names, domain names, and so on) used in this example are invalid entries. Replace these entries with data specific to your configuration.

Excerpt from an LDAP user file:

```
dn: o=domain.com
objectClass: organization
o: domain.com
```

```
dn: cn=tristan, o=domain.com
objectClass: inetOrgPerson
cn: tristan
sn: GDog
mail: tristan@domain.com
uid: tristan
userPassword: pass1
```

```
dn: cn=enrico, o=domain.com
objectClass: inetOrgPerson
cn: enrico
sn: DElephant
mail: enrico@domain.com
uid: enrico
userPassword: pass2
```

RADIUS

About RADIUS

Remote Authentication Dial-in User Service (RADIUS) was originally used to authenticate people logging in to the network through a modem to remote points of presence (POPs). If your environment includes a RADIUS server, you can configure NetCache to authenticate users and allow access to services using the RADIUS server.

For RADIUS, the NetCache client hop uses clear-text authentication. For the server hop, NetCache authenticates through an MD5 hash comparison of a shared secret between the RADIUS server and NetCache.

Preparing to configure

Before you begin to configure RADIUS authentication in NetCache, verify the following:

- ◆ The RADIUS server IP address is correct.
- ◆ Your RADIUS server and the appliance are positioned on your network so that NetCache can open a UDP connection.
- ◆ For the server port to which NetCache connects to the server: the RADIUS default port number is 1812.
- ◆ NetCache and RADIUS are configured with a shared secret.

Section F: Authenticating with NTLM and Kerberos

About this section

This section pertains to using Microsoft Windows NTLM and Kerberos protocol servers to authenticate NetCache users. It gives some background information on the NTLM and Kerberos protocols.

Topics in this section

This section contains the following topics:

- ◆ [“About NTLM and Kerberos”](#) on page 24
- ◆ [“Authenticating with NTLM”](#) on page 27
- ◆ [“Authenticating with Kerberos”](#) on page 30

About NTLM and Kerberos

NTLM and Kerberos protocols

You can configure your environment to provide client authentication for Microsoft domains using the following protocols.

Authentication method	Microsoft environment
NTLM (true NTLM)	Microsoft Windows NT or Microsoft Windows 2000 domain
Kerberos	Microsoft Windows 2000 domain

How you join the NetCache appliance to a Microsoft domain determines the authentication methods available to you.

Joining the Windows NT domain as an NT system: You can join the appliance to a domain using either of the methods described in “[Joining a Microsoft Windows NT domain](#)” on page 40.

In this case, the NetCache is viewed as an NT system. It can use NTLM for authentication.

Joining the Windows 2000 domain as 2000 system or an NT system:

You can join the appliance to a Microsoft Windows 2000 domain using either of the methods described in “[Joining a Microsoft Windows 2000 domain](#)” on page 33.

In this case, whether the Microsoft Windows 2000 domain is native mode or mixed mode and the method you use to join the NetCache to the domain determines how the NetCache is viewed as a system. For an explanation of the two Microsoft Windows 2000 domains, see “[About Windows 2000 domains](#)” on page 25.

Here is how the NetCache behaves when it joins the Windows 2000 domain in either of its two modes:

- ◆ In native mode, you must join the NetCache to the Microsoft Windows 2000 domain as a Windows 2000 system. In native mode, the NetCache can use either NTLM or Kerberos for authentication.
- ◆ In mixed mode, you can join the NetCache to the Microsoft Windows 2000 domain in either of the following ways:

- ❖ As a Windows 2000 system, using one of the methods described in [“Joining a Microsoft Windows 2000 domain”](#) on page 33. In this case, the NetCache can use either NTLM or Kerberos for authentication.
- ❖ As an NT system, using one of the methods described in [“Joining a Microsoft Windows NT domain”](#) on page 40. In this case, the NetCache can use NTLM for authentication.

About Windows 2000 domains

As mentioned previously, there are two kinds of Microsoft Windows 2000 domains: native mode and mixed mode. Here are their defining characteristics:

- ◆ A native-mode Microsoft Windows 2000 domain
 - ❖ Contains only Microsoft Windows 2000 (or greater) and Windows XP hosts and devices. (This domain excludes all devices that are not designed for a Windows 2000 domain environment.)
 - ❖ Supports NTLM and Kerberos authentication methods.
- ◆ A mixed-mode Microsoft Windows 2000 domain
 - ❖ Contains Microsoft Windows 2000 hosts, Microsoft Windows XP hosts, and also NT4 hosts.
 - ❖ Supports NTLM and Kerberos authentication methods, depending on how the NetCache joined the domain. For details, see the following sections:
 - [“Joining the Windows NT domain as an NT system”](#) on page 24.
 - [“Joining the Windows 2000 domain as 2000 system or an NT system”](#) on page 24.

Promptless authentication

Microsoft has created provisions for Microsoft Internet Explorer users to be authenticated without requiring the user to input a user name and password every time a browser session is started.

In this documentation, this process is referred to as promptless authentication because the user is not prompted from the browser (yet again) for this information. Rather, promptless authentication depends on the tendency of Microsoft Windows to maintain a persistent password state about the current user.

Note

Microsoft Internet Explorer 6 (MSIE 6) does not support authentication of a client to a proxy server (such as the NetCache appliance) using Kerberos.

How promptless authentication works: When a user signs on to an NT domain, the user name and password are kept locally for later use. When an application encounters the need to prove the identity of the user to another machine, the application might request that the user name and password be used to encrypt an arbitrary challenge string provided by the other machine. The encrypted string is then sent to the machine sending the challenge; that machine might then pass it to the Primary Domain Controller (PDC). The PDC will then encrypt the challenge with the password and compare the result with the challenge-response sent by the client.

Authenticating with NTLM

The NTLM protocol

NT LAN Manager (NTLM) performs authentication using a multiple-exchange, encrypted challenge-and-response sequence between NetCache and a Microsoft Windows domain controller. You can enable NTLM for either Microsoft Windows NT or Microsoft Windows 2000 domain access to the Microsoft Windows authentication environment.

For true NTLM requests—those coming from Microsoft Internet Explorer and not clear-text NTLM requests coming from another browser, such as Netscape—each authentication exchange is different. A unique challenge is posed to the browser on each connection, and the NetCache appliance does not discover the user's password.

The NTLM protocol handshake and log entries: The NTLM challenge-response handshake entails three requests and responses. These exchanges occur at the beginning of each browser-to-NetCache connection, before the data is delivered.

As it does for all authentication processes, the log file reflects the NTLM handshake process, showing three GET requests. Because they are similar, these entries can confuse log viewers.

Each intermediate GET request from the client is logged as generating a 403 authorization response from the appliance until the final OK response occurs. The appliance sends a 407 authorization message, which is noted as a 403 authorization message.

The Web access log entries for the NTLM handshake: The NTLM handshake process log entries are written to the Web access log. These log entries contain information similar in content to that shown in the following example:

```
1019544846.91 0.079 10.53.50.76 TCP_HIT_ACCESS_DENIED/403 542
GET http://web.mycompany.com/~bohr/particle.txt - - " "
AUTH_FAIL

1019544846.91 0.001 10.53.50.76 TCP_HIT_ACCESS_DENIED/403 619
GET http://web.mycomany.com/~bohr/particle.txt - - " " -

1019544846.92 0.030 10.53.50.76 TCP_MISS_PRIVATE/200 388 GET
http://web.mycompany.com/~bohr/particle.txt administrator
DIRECT/10.10.20.171 "text/plain" -
```

Reprocessing NTLM authentication failures

If the Domain Name Service (DNS) is not properly configured or if other Microsoft Windows domain configuration problems exist, the authentication process can result in failure causing the appliance to return an authentication failure message to Microsoft IE.

In response to that error, the Microsoft Internet Explorer browser might prompt the user for a user name and password to transmit to NetCache to persist in authentication. In this case, NetCache performs clear-text authentication, following the specified checking order sequence.

Clear-text NTLM requests

For clear-text NTLM requests (for example, those coming from a Netscape browser), the appliance converts the clear-text authentication information from the browser into NTLM format.

Requiring renewal for an expired NTLM password

If a NetCache user's NTLM password expires and requires manual renewal, you can use the Expired Password URL option on the Authentication - General page to specify a URL that displays to the user the requirement to reset an expired password.

NTLM and origin servers

Early versions of Microsoft Internet Explorer prior to MSIE 5.5sp1 and MSIE 6, both of which in part fix the problem, cannot perform NTLM authentication with origin servers if, as clients, they are aware that a proxy server is involved in the process.

One solution to this problem is to use transparency so that the Microsoft Internet Explorer browser is unaware of the involvement of the NetCache.

Testing NTLM authentication

To test whether NetCache is authenticating users using NTLM, use your browser to access a Web site you prefer.

If NetCache is configured correctly and you are using:

- ◆ Microsoft Internet Explorer
NTLM authenticates Microsoft Internet Explorer users to the Windows domain authentication database without requiring the user to enter user name and password information. In this document, this process is referred to as promptless authentication.
- ◆ Netscape or another client browser

NTLM authenticates these users to the Windows domain, but they are required to reenter their user name and password information.

Authenticating with Kerberos

The Kerberos protocol

The Kerberos protocol supports Microsoft Windows 2000 access to the Microsoft Windows authentication environment. Kerberos is the authentication method used with Microsoft's Active Directory.

Note

The Microsoft Windows 2000 Active Directory configuration and administration architecture allows for remote administration of systems and transaction of tasks that entail communications with centralized servers, including authentication.

Kerberos authentication is based on a shared-secret key distribution model in which NetCache validates tickets presented by the client (user). The ticket, which verifies the client's identity, is issued by the Kerberos Key Distribution Center (KDC). When a Kerberos client sends a request to NetCache and NetCache requests authentication, the client obtains a ticket from the Kerberos server and presents it to NetCache.

Kerberos keeps a database of its clients and their keys. The private key is a large number known only to Kerberos and the client. Private keys are negotiated at registration. Because Kerberos knows these keys, it can create messages that convince a server that a client is really who it claims to be.

Kerberos and backup protocols

If Kerberos authentication is enabled, then the NetCache does the following:

1. NetCache first attempts Kerberos authentication of a requesting client against the Microsoft Windows 2000 Active Directory user database. If the appliance challenges the requesting client for Kerberos authentication, the client browser will automatically send a Kerberos ticket.
2. If the requesting client browser does not support Kerberos authentication, or if Kerberos authentication fails for some other reason, NetCache attempts to authenticate the user against a clear-text database, in response to the browser's request.

For the clear-text method to use, the NetCache follows the order specified in the Authentication Checking Order option on the Authentication - General page.

Note

To support authentication of clients within a Microsoft Windows 2000 domain with browsers that do not support Kerberos authentication (such as Netscape), Network Appliance recommends enabling both Kerberos and NTLM authentication. For details on enabling NTLM, see Section , “[Authenticating with NTLM](#),” on page 27. For details on enabling Kerberos, see Section , “[Authenticating with Kerberos](#),” on page 30.

Configuring your browsers for Kerberos authentication

NetCache supports Kerberos authentication of users using client browsers within a Microsoft Windows 2000 domain of a specific type and configuration. Network Appliance recommends that you do not use Kerberos for proxy authentication.

To use Kerberos for authentication, the following conditions must be met:

- ◆ The client browser must support Kerberos exchange. Currently, only Microsoft Internet Explorer is Kerberos-capable.

Note

Microsoft Internet Explorer 6 (MSIE 6) does not support authentication of a client to a proxy server (such as the appliance) using Kerberos.

- ◆ The client browser must be configured to access the appliance as a proxy server in nontransparent mode only.
- ◆ The client browser’s proxy server setting must specify this appliance by its host name rather than its IP address. The host name of the appliance must be the same as the Microsoft Domain Name Services (MS-DNS) entry, and it must also match the machine-account name that the NetCache appliance used when it joined the domain.

For example, on a client Microsoft Internet Explorer browser, you would use the following menu selections to specify the appliance host name, `netcache.domain.com` instead of its equivalent IP address:

Tools > Internet Options > Connections > LAN settings > Use a proxy server menu

Testing Kerberos authentication

To test whether NetCache is authenticating users using Kerberos, complete the following steps.

Step	Action
1	Check to ensure that authentication is required for HTTP.
2	Use the Appliance Manager Setup > Authentication pages to temporarily disable any LDAP, RADIUS, or NTLM support that you have enabled on this appliance. Keep Kerberos support enabled.
3	Use Microsoft Internet Explorer to access the Web site you prefer. ◆ Make sure that the client browser is configured to access the appliance in nontransparent mode. ◆ Make sure that the Microsoft Internet Explorer version supports Kerberos authentication. If NetCache is configured correctly, Kerberos authenticates Microsoft Internet Explorer users against the Active Directory database without requiring the user to enter user name and password information.

Section G: Joining a Microsoft Windows 2000 domain

About this section

This section describes the two ways in which you can join a NetCache appliance to a Microsoft Windows 2000 domain, also known as an Active Directory (AD) domain, in order to authenticate clients using either the Kerberos or NTLM protocols.

Before you use the Authentication - NTLM and Kerberos page in the Appliance Manager to enable the protocols, you must include the appliance in the Microsoft Windows domain.

You can use the Domain Name Service (DNS) to discover services within the domain or you can bypass DNS and directly connect to the domain controller in order to authenticate to the domain.

Preliminary notes on Windows 2000 machine accounts

Microsoft Windows 2000 hosts create their own machine accounts using the credentials of an Administrator user account. The Microsoft Windows 2000 machine account must match exactly its DNS name. Be aware that in a Microsoft Windows 2000 mixed-mode domain, the host cannot have a NetBIOS name that is different from its DNS name.

Topics in this section

This section contains the following topics:

- ◆ [“Joining the domain using DNS”](#) on page 34
- ◆ [“Joining the domain without using DNS”](#) on page 36

Joining the domain using DNS

About DNS

In Microsoft Windows 2000, domains are named according to the Internet's domain name system, and a Domain Name Service (DNS) server is commonly used to resolve these names into their IP addresses. A DNS server contains a database of names specified in user-friendly standard domain name dot format. A system such as the NetCache appliance might send a host name to the DNS server, and the DNS server would resolve the name to its mapped host IP address and service.

A Windows 2000 domain, or Active Directory (AD) domain, integrates tightly with DNS. When a Windows 2000 server exports a service, it advertises that service in DNS by creating a Service Location Resource (SRV) record that indicates the type of service it provides and lists its host name.

All hosts that must access Windows 2000 domain services must have access to the DNS server storing these records. For example, if a NetCache appliance must communicate with an AD server or a Kerberos server, it must be able to access the AD DNS server. If it must circumvent the DNS server, the NetCache appliance must have access to the information stored in the SRV records. This process is explained in greater detail in [“Circumventing DNS”](#) on page 36.

Requirements if using DNS

Joining the appliance to a Microsoft Windows 2000 domain using DNS is simple. You configure the appliance to use the AD DNS server, and the NetCache will discover the appropriate domain controller through DNS. You can join the NetCache appliance to the domain using either the Appliance Manager or the CLI (see [“Procedures and option descriptions”](#) on page 1).

To join a Microsoft Windows 2000 domain using DNS, you must provide the following information.

Information	Required to...	Specified from ...
Name of the Windows domain Note_____ Specify the full DNS name of the Windows 2000 domain, for example, domain.com.	Identify the domain you want the appliance to join	The Appliance Manager's Setup > Authentication > NTLM and Kerberos page: ◆ Join Domain tab ◆ Windows Domain Name option (full DNS domain name) CLI: windows_setup
Windows 2000 machine name for this NetCache appliance Note_____ This is the name configured for this NetCache appliance in the target domain's Active Directory (AD).	Identify this NetCache appliance to the domain	The Appliance Manager: ◆ Join Domain tab ◆ Windows Machine Name option (appliance name configured in AD) CLI: windows_setup
Windows administrator credentials: User name and password	Authenticate the NetCache appliance (administrator) to the Windows domain	The Appliance Manager: ◆ Join Domain tab ◆ Windows Administrator Credentials option (user name and password) CLI: windows_setup

Joining the domain without using DNS

Circumventing DNS You can configure the NetCache appliance for deployment in enterprise environments that require the appliance to authenticate to Microsoft Windows 2000 without using DNS as a name server. For this type of deployment, the appliance does not use the Server Message Block (SMB) over port 139 to maintain its machine account within the Microsoft Windows 2000 domain. (However, it can use the SMB port to authenticate clients. SMB is used for communication between hosts.) Instead, the appliance uses LDAP and Kerberos to maintain its machine account, and it is for this reason that the appliance must know their location.

Given these circumstances, the appliance must be able to locate the Microsoft Windows 2000 Kerberos and LDAP services. Recall that Kerberos is used to authenticate hosts and users (in this case, the NetCache), and LDAP is used to provide generalized access to domain information.

The DNS server stores the locations of these services in special records called Service Location Resource (SRV) records. These records allow a network to register its location and services with DNS, which then queries the records for the location (IP address) of a service when it is required.

This section describes how you can override use of DNS in joining the appliance to the Microsoft Windows 2000 domain by specifying the IP addresses and other information for Microsoft Windows 2000 domain account setup.

Requirements if not using DNS Joining the NetCache appliance to a Microsoft Windows 2000 domain without using DNS requires that you identify a set of hosts that export services required to join the domain and authenticate the appliance to it.

Windows 2000 servers can export any of four types of services, shown in the following table. For each of these services, special records are stored in DNS that a Windows 2000 host uses to locate the servers exporting the services it requires. Because DNS is not available, the NetCache must specify information about the hosts as shown in the following table.

Host type	Service record
Active Directory	LDAP

Host type	Service record
Kerberos Key Distribution Center	KDC
Kerberos Admin	kpasswd
SMB	DC

Note

Most often all of these services are exported by the same host.

Kerberos

The NetCache appliance authenticates itself to the Microsoft Windows 2000 domain using Kerberos authentication. Kerberos is based on a ticket-granting service, which runs on a Kerberos Key Distribution Center (KDC). The KDC provides session keys for authentication, authenticates the client and server, and grants tickets which assert the user's identity and include other security data. Among the information you must provide are addresses to the KDC and Kerberos password servers to be used.

You can join the appliance to the domain using either the Appliance Manager or the CLI (see "[Procedures and option descriptions](#)" on page 1).

To join a Microsoft Windows 2000 domain without using DNS, you must provide the following information.

Information	Required to...	Specified from...
Windows 2000 Domain Controller	Gain access to a host providing SMB services to be used for communication between hosts in the Windows 2000 domain	The Appliance Manager's Setup > Authentication > NTLM and Kerberos page: ◆ Domain Controllers tab ◆ Windows 2000 Domain Controllers option (name and IP address) CLI: <pre>config.auth.windows.dc</pre>

Information	Required to...	Specified from...
Windows 2000 LDAP server	Gain access to a host providing LDAP services ◆ To obtain generalized access to domain information ◆ To configure the appliance's server account	The Appliance Manager: ◆ Domain Controllers tab ◆ Windows 2000 LDAP servers option (host name and IP address) CLI: <pre>config.auth.windows.ldap</pre>
Windows 2000 kpasswd servers	Securely set the NetCache appliance's secret key for Kerberos	The Appliance Manager: ◆ Domain Controllers tab ◆ Windows 2000 kpasswd servers option (IP address, and, optionally, port number) CLI: <pre>config.auth.windows kpasswd</pre>
Windows 2000 KDCs	Get Kerberos tickets to authenticate to the LDAP domain controller	The Appliance Manager: ◆ Domain Controllers tab ◆ Windows 2000 KDCs option (host name and IP address) CLI: <pre>config.auth.windows.kdc</pre>

Information	Required to...	Specified from...
Name of the Windows domain, in its full DNS form, such as domain.com	Identify the domain you want to join Used for many purposes.	The Appliance Manager: ◆ Join Domain tab ◆ Windows Domain Name option (Windows 2000 domain name) CLI: windows_setup
Windows 2000 machine name for this appliance Note _____ This is the name configured for this appliance in the target domain's Active Directory (AD).	Create the appliance's server account	The Appliance Manager: ◆ Join Domain tab ◆ Windows Machine Name option (appliance name configured in AD) CLI: windows_setup
Windows administrator credentials: User name and password	Authenticate the appliance (administrator) to the Windows domain	The Appliance Manager: ◆ Join Domain tab ◆ Windows Administrator Credentials option CLI: windows_setup

Section H: Joining a Microsoft Windows NT domain

About this section

This section describes the two ways in which you can join a NetCache appliance to a Microsoft Windows NT domain. You can deploy the appliance in enterprise environments that rely on use of Microsoft Windows Internet Name Service (WINS) to authenticate the appliance to a Microsoft Windows NT domain. Alternatively, you can deploy the appliance in environments that require the cache to authenticate to a Microsoft Windows NT domain without using WINS to resolve the NetCache name and authenticate to the domain.

Topics in this section

This section contains the following topics:

- ◆ [“Joining the domain using WINS”](#) on page 41
- ◆ [“Joining the domain without using WINS”](#) on page 44

Preliminary notes on Windows NT machine accounts

Before you join a appliance to an NT domain, you must manually create a machine account for the cache. An NT machine account uses an initial password that is not difficult to discern. The appliance changes this password as part of the domain join process.

Joining the domain using WINS

About WINS

Proprietary to Microsoft and similar to DNS, WINS provides computer-name to IP-address mapping for NetBIOS-based computers and services. WINS is a name-resolution server used by computers and in Microsoft Windows NT domains to locate computers and services.

Requirements if using WINS

To join a Microsoft Windows NT domain using WINS, you provide the following information.

Information	Required to...	Specified from...
NT4 preferred domain controllers Note _____The list you specify should include at least two of the DCs whose IP addresses WINS returns. (Your list does not need to overlap entirely with the returned list.)_____	Order by preference the list of DCs that WINS returns to the appliance	The Appliance Manager's Setup > Authentication > NTLM and Kerberos page: ◆ Domain Controllers tab ◆ NT4 Preferred DCs option CLI: <code>config.auth.windows.prefdc</code>
Name of the Windows NT domain	Identify the domain you want this appliance to join	The Appliance Manager: ◆ Join Domain tab ◆ Windows Domain Name option (full domain name) CLI: <code>windows_setup</code>

Information	Required to...	Specified from...
Windows NT machine name for this appliance Note_____ This is the name of the NetCache server account configured on the Windows primary domain controller (PDC). _____	Identify this appliance to the domain	The Appliance Manager: ◆ Join Domain tab ◆ Windows Machine Name option (name specified on the PDC) CLI: <code>windows_setup</code>
If required, the address of the WINS server	Locate the Windows NT domain controller, when it is not on the local subnet	The Appliance Manager: ◆ Join Domain tab ◆ WINS Server IP Addresses CLI: <code>windows_setup</code>

To enable NTLM authentication in a Microsoft Windows NT domain, you must create a NetCache server account on the Microsoft Windows NT primary domain controller (PDC) for NTLM access.

You use the Microsoft Windows NT Server Manager to create a machine account for an appliance in the Microsoft Windows NT domain. To create a machine account, complete the following steps.

Step	Action
1	At the Windows NT Primary Domain Controller (PDC), select Start > Programs > Administrative Tools > Server Manager.
2	Click Computer > Add to domain.
3	Enter the machine account name for the appliance, then click OK.
4	Join the domain.

Step	Action
5	Use the Authentication - NTLM and Kerberos page to enable and configure the protocols.

Joining the domain without using WINS

Circumventing WINS

You can configure the NetCache appliance for deployment in enterprise environments that require the appliance to authenticate to Microsoft Windows NT without using WINS as a name server. For example, an enterprise might want to bypass WINS because its traffic load is heavy or because WINS is considered less reliable than the enterprise can allow for.

To bypass WINS, the appliance must connect directly to the Primary Domain Controller (PDC) used to maintain the user and machine database for the domain. The appliance must also specify the location (IP address) of a Backup Domain Controller (BDC) in the domain to be used to authenticate users.

A Microsoft Windows NT domain can have only one PDC, which propagates its data to the BDCs, but it can have many BDCs.

To further ensure that WINS is successfully circumvented if the first PDC and BDCs specified are unavailable, you can specify multiple failover domain controllers to be used.

Requirements if not using WINS

To join a Microsoft Windows NT domain without using WINS, you must provide the following information

Information	Required to...	Specified from...
NT4 primary DC addresses	To locate the domain. ◆ The appliance then identifies itself to the domain using the PDC ◆ The appliance regularly (weekly) changes its user password on the PDC automatically to enhance security. ◆ The NetCache can also use the PDC to authenticate clients (users). However, in most cases it uses one of the BDCs, which are less busy.	The Appliance Manager's: Setup > Authentication > NTLM and Kerberos page: ◆ Domain Controllers tab ◆ NT4 Primary DC IP addresses
NT4 backup DC addresses	Authenticate users to the NT domain	The Appliance Manager: ◆ Domain Controllers tab ◆ NT4 Backup IP Addresses option
Name of the Windows domain	Identify the domain you want to join	Appliance Manager: ◆ Join Domain tab ◆ Windows Domain Name option CLI: windows_setup

Information	Required to...	Specified from...
Windows NT machine name for this NetCache Note_____ This is the name of the NetCache server account configured on the Windows NT primary domain controller (PDC). _____	Identify this appliance to the domain	The Appliance Manager: ◆ Join Domain tab ◆ Windows Machine Name option (name of the appliance on the NT PDC) CLI windows_setup

About this chapter	This chapter describes how to create rules and implement secure policies for controlling access to the NetCache by using the NetCache ACL feature. The NetCache is permissive by default; that is, ACLs are not enabled and no requests are denied.
Topics in this chapter	This chapter provides information about the following topics: ◆ “About access controls” on page 49 ◆ “ACL building blocks” on page 52 ◆ “Regular expressions” on page 61 ◆ “About setting restrictions” on page 67 ◆ “ACL optimization” on page 69 ◆ “About controlling access” on page 72 ◆ “About controlling access to streaming media” on page 76
Procedures and option descriptions	See the Appliance Manager online Help for configuration procedures and option descriptions.
Location of configuration options	Options for configuring the features discussed in this chapter are in the following location in the Appliance Manager: Setup tab > Access control > Access Control Lists
Controlling access using the CLI	This chapter describes options in the Appliance Manager to control access to the NetCache appliance, and to control access to content. For descriptions of CLI configuration commands, see the <i>Command Line Reference</i>. To get information about access control configuration options from the CLI, enter the following command: <pre>set config.acl.?</pre> The preceding command lists the available access control options. To get help for a specific access control option, enter the following command: <pre>help config.acl.optionname</pre>

To get information about SmartFilter configuration options from the CLI, enter the following command:

```
set config.smartfilter.?
```

To get help for a specific SmartFilter option, enter the following command:

```
help config.smartfilter.optionname
```

To get information about WebWasher DynaBLocator configuration options from the CLI, enter the following command:

```
set config.wwfilter.?
```

To get help for a specific WebWasher DynaBLocator option, enter the following command:

```
help config.wwfilter.optionname
```

About access controls

What access controls do

Access control settings prevent unauthorized clients from accessing the NetCache appliance, and restrict the access of authorized clients to services. The appliance supports different types of rules that impose a variety of restrictions. Sets of rules, or access control lists (ACLs), are used to control access to the Internet, and to impose restrictions inside the intranet domain.

Internet setting: In an Internet setting, client access controls prevent unauthorized users outside your organization from resolving their Web requests through your cache and wide-area bandwidth. You can also set rules to limit access for authorized users.

Intranet setting: In an intranet setting, client access controls can help enforce authentication and authorize exactly what users have access to.

Authorized access

When a user makes an authenticated request, NetCache checks the user against all groups in the NetCache user database, and attempts to match ACLs that are associated with those groups. If the request is not matched by user or group ACLs, then the request is checked against all other access control rules. If access is denied, a system message displays the rule and the name of the ACL that prevented access to NetCache services.

Note

See Chapter 1, “[Authenticating NetCache Users](#),” on page 1 for a description of NetCache client authentication and supported methods for authenticating administrators and users.

ACL rule checking

NetCache sequentially reads ACLs and checks for the first matching entry. When a match occurs, the action is executed. If no match occurs, no action is executed and rule checking continues.

The NetCache rule checking sequence is shown in the following table.

Stage	Description
1	Check global settings that control authentication for protocol requests.
2	Check ACL permission settings for the applicable group in the NetCache user database.
3	Check request-type (protocol specific) ACL.
4	Check Web accelerator ACL.
5	Check global ACL.
6	Verify whether the content filtering application (SmartFilter or WebWasher DynaBLocator) denies the request.
7	Check group permissions settings.
8	Allow the request.

Order of precedence

Types of rules and order of precedence are described in the following table.

Order	Type	Description
1	User and group	User and group rules explicitly allow, deny, or redirect requests. User and group ACLs have precedence over request-type, global, SmartFilter, and group access permissions.
2	Request-type (protocol specific)	Request-type rules control specific types of traffic. Access control rules are processed only for requests of a particular type, or protocol, such as HTTP. Request-type access controls have precedence over Global and SmartFilter access controls. If a request-type rule explicitly allows, denies, or redirects a request, NetCache follows this rule regardless of any rules configured in the Global Access Control option and any content category restrictions set in the SmartFilter options.
3	Accelerator	If a request originates from a Web-accelerator (reverse-proxy), this ACL is checked.

Order	Type	Description
4	Global	Global rules impose restrictions on all traffic by allowing you to specify variables, such as a user name and a URL. Global access controls have precedence over SmartFilter rules. If a global rule explicitly allows, denies, or redirects a request, NetCache follows this rule regardless of any content category restrictions set in the SmartFilter or DynaBLocator options.
5	Content filtering application (SmartFilter or DynaBLocator)	Restricts access to Web sites that contain objectionable content. Rules apply only to URLs that are not explicitly allowed, denied, or redirected by global or request-type options.
6	Group access permissions	Group access permissions specify default access to services for groups, and control rewrites or redirects for blocked requests. Default rules apply to requests that have no rule matches.

ACL building blocks

ACLs ACL entries allow or deny requests based on a number of variables, such as client IP addresses. ACLs are associated with specific users, groups of users, and types of requests, such as HTTP.

When ACL rules are formed, actions are combined with expressions that contain a variable-value pair. Variables and associated values form what is commonly referred to as a pattern-matching expression.

Syntax When applying actions (for example, allow, deny, redirect, rewrite, and auth) to matching expressions, use the `action expression` syntax, as shown in the following table.

Action	Expression
allow	client-ip 128.126.10.0
auth	group finance
deny	server-domain net
allow	time 9am - 5pm
deny	FTP

Actions Actions and expressions form rules that control requests for access to Web content and supported protocols. The following table lists supported actions, and describes how they govern client requests.

This action...	Specifies...
allow	Allow the request to proceed and disregard all the following rules.
auth	Require authentication and continue the rule checking sequence if already authenticated.

This action...	Specifies...
cache	Cache the stream. This action is used to create rules for controlling streaming media cacheability only.
context	This action is used to create rules for controlling HTTPS requests, and will direct traffic as defined by an SSL inbound or outbound context. For example: context "inctx" cache-port 443 context "outctx" server-port 443 and server-ip 10.24.155.13
deny	Deny the request immediately.
drop <ttl> where ttl is length of time in seconds. Default is 60 seconds.	Block packets from the source specified. For example, when used with the client-ip variable, packets from a matched client IP address are blocked at the IP level.
icap	Directs requests for ICAP services.
icapnow	This action is used for ICAP services that must be invoked early in an ACL evaluation process. For example, content-filtering services.
no-cache	Do not cache the stream. This action is used to create rules for controlling streaming media cacheability only.
no-log	Selectively log and filter log file information by field. If applied as a global ACL, all logs are filtered. Entries in log files are replaced with a "-". Note _____ Global authentication must be enabled. _____

This action...	Specifies...
proxy	Proxy session. Do not serve from cache. This action is used to create rules for controlling streaming, streaming server accelerator, HTTP, and Web accelerator only.
re-auth	Force reauthentication.
redirect <url>	Redirect the request immediately.
rewrite	Rewrite the URL, but continue the rule checking sequence. Note _____The ACL <code>rewrite</code> action does not support HTTP-based tunneling requests._____
rewrite-redirect	Rewrite the URL prefix, and send a redirect. This action is specific to HTTP, and supported for the <code>url-prefix</code> variable only. If target URL is absent, the Default ACL URL option is used to specify a default URL to redirect or rewrite requests to an internal Web page.
try-auth	Require authentication and continue the rule checking sequence even if authentication fails.

For information on setting restrictions to control client access to streaming media, see “[About controlling access to streaming media](#)” on page 76.

Variables

Variables are assigned values that form expressions, for example, *time 9am - 5pm*. The following table lists variables and provides a description and example of each.

Note

This appliance supports variables that specifically control streaming media. However, some streaming variables can only be used within the context of Streaming Bandwidth Management and Cacheability rules. These variables are designated with an asterisk (*) in the following table.

Variable/Description	Example
accel Web Accelerator requests	deny accel
attribute Attributes returned by ICAP services. Note An ICAP service must be invoked before an attribute check can occur.	icap(content-filter) not group marketing deny attribute content-filter news
any Any requests	deny any
auth Authenticated request	redirect "http://web.com/usage.html" not auth and not url "http://web.com/usage.html"
bitrate* Stream bit rate	allow bitrate 0-220kbps
cacheable* Cacheable content	deny not cacheable
cached* Cached streams	deny group sales and not (cached and bitrate 0k - 100k)
cache-ip IP address of this NetCache	allow cache-ip 206.80.5.0/24
cache-port Port number on this NetCache	deny cache-port 8080

Variable/Description	Example
client-domain Client domain name	deny client-domain .wisc.edu
client-ip Client IP address	deny client-ip 206.79.5.51
client-port Client port number	deny client-port 25
contains Substring search operator Note _____ This variable supports user, group, req-headers, server-name, client-domain, and server-domain variables.	deny url contains bomb
contains-case Case-sensitive substring search operator Note _____ This variable supports user, group, req-headers, server-name, client-domain, and server-domain variables.	deny url contains-case VIOlence
date Date and time	deny group sales and date "sun nov 6 08:49:37 2002"
day Day	allow day mon - fri

Variable/Description	Example
ftp FTP requests Note _____ Setting <code>auth ftp</code> is supported for FTP over HTTP requests only. It is not supported for FTP proxy requests. If <code>auth ftp</code> is used in a proxy configured environment, the appliance will allow FTP requests and bypass the remaining FTP ACLs.	deny ftp
get HTTP, FTP, NNTP, streaming requests	allow get and group readonly
gopher Gopher requests	deny gopher
group Multidomain support for groups, including NetCache, Windows, and LDAP. Note _____ A group name with no domain name matches any group with that name, regardless of the domain.	auth any allow group finance deny any This ACL allows access for members of group “finance” in all trusted domains
group\ Multidomain support for groups, including NetCache, Windows, and LDAP. Note _____ Using the “\” delimiter between the domain name and group name enables domain comparison.	auth any allow group “ITALY\globalgroup” deny group “EUROPE\globalgroup” deny any This ACL allows access for members of ITALY, and denies access for members of EUROPE. Globalgroup is a trusted domain.

Variable/Description	Example
http HTTP requests	deny url "http://www.corp.com"
live* Live streams	deny group sales and live and day monday - friday
post HTTP, FTP, NNTP requests	allow post and group readwrite
req-headers HTTP request headers	deny req-headers matches "User-Agent:.*Mozilla"
req-line HTTP request line Note _____The req-line variable supports method arguments (GET, POST, HEAD), and supports the regex contains and contains-case operators._____	deny req-line matches "GET.*http://badsite*" deny req-line contains POST
server-domain Server domain name	allow server-domain .mycorp.com
server-ip Server IP address	deny server-ip 206.79.5.0/24
server-name Server name in URL	allow server-name www.mine.com
server-port Server port number in URL	deny server-port 1 - 1024
smartfilter SmartFilter categories	deny smartfilter "chat" "sports" "games"

Variable/Description	Example
splittable* Splittable content	deny group sales and note splittable
time Time Note _____ The time variable syntax is restrictive, and is supported only when expressed as a range or as part of a regex matching expression. _____	allow time 9am - 5pm
tunnel Tunnel requests	allow tunnel
url URL string	deny url http://www.mine.com/
url-prefix Prefix of URL string	deny url-prefix http://www.mine
user User name of authenticated user	deny user bob
wwfilter WebWasher DynaBLocator categories	deny wwfilter "job search" deny wwfilter sects

Expressions

Expressions are single predicates, variable-value pairs, range matches, or regular expressions (regex). The following table provides an example of each expression type.

Type	Definition	Example
Single predicate	<i><variable></i>	FTP
Variable-value pair	<i><variable> <value></i>	client-ip 128.148.37.17

Type	Definition	Example
Range matching	<code><variable> <value> - <value></code>	server-port 1- 1024
Regular expression matching (regex)	<code><variable> matches <regex></code>	<code>“^http://[0-9.]+/redcodeworm.asp\?X+”</code>

Note

When you create rules, you must enclose values that contain white space or nonalphanumeric characters in double quotation marks.

Operators

Complex expressions are built using parentheses and boolean operators, such as `and` or `&&`, `or` or `||`, and `not`. For example:

- ◆ `<expr> and <expr>`
- ◆ `<expr> or <expr>`
- ◆ `(<expr>)`
- ◆ `not <expr>`

Caution

Using the operator `not` in combination with the action `allow` (for example, `allow not user user1`) does not result in an action to deny requests when the rule is applied.

For a variety of operators and expressions that can be used to create complex rules, see “[Pseudogrammar](#)” on page 64.

Regular expressions

About regular expressions

Regular expressions are familiar to those who have worked extensively with UNIX command-line tools. They form complex rules, or pattern descriptions which are used to match patterns against a string. Regular expressions are built by combining simple, small expressions, the simplest being a single character, such as “a”, that matches itself. Simple regular expressions can be combined to form strings, such as “hello”, which matches “hello” or “hihellothere”.

Matching

Regular expression matches compare a specified string against each pattern until a match occurs. When a match occurs, the function returns true. If no match occurs, the function is false. Rules are executed only when a pattern-matching expression is true.

Note

NetCache uses POSIX-standard extended regular expression syntax, fully described by IEEE POSIX 1003.2.

Performance and efficiency

Regular expressions can be used to perform a variety of restrictions. However, rule complexity and the number of matches that are checked before servicing each request can affect performance. For this reason, regular expressions, or expressions that contain the word “matches”, should be cautiously applied and used only when necessary.

Example: Values returned match any pattern that ends with netapp.com.

```
server-name matches "\.*\.netapp\.com$"
```

In comparison, an equivalent but more efficient expression produces the same result.

```
deny server-domain netapp.com
```

Optimizing performance

Complex regular expressions or expressions that perform extensive character matching consume the greatest amount of system resources, followed by simple expressions and exact matches. To optimize performance when expression

matching is necessary, be sure that you form expressions using the correct variables. Often, a complex expression can be simplified to an exact match. For example, the following sets of expressions achieve the same goal.

Example 1: In the following example, `url matches` can be replaced with the `server-name` variable to achieve the same result.

Simple expression: `deny url matches "[a-z]*://www.netapp\.com/.*"`
Exact match: `deny server-name www.netapp.com`

Example 2: In the following example, `url matches` can be replaced with the `server-domain` variable to achieve the same result.

Simple expression: `deny url matches "[a-z]*://[^/]*jobs\.(com | net)"`
Exact match: `deny server-domain "jobs.com" and`
`deny server-domain "jobs.net"`

Example 3: In the following example, `url matches` can be replaced with either `server-name` or `url-prefix` variables to achieve the same result.

Complex expression: `deny url matches "^http://chat\.yahoo\.com/(fishing | music | cooking)$"`
Exact match: `deny server-name chat.yahoo.com`

In the preceding example, the entire site is blocked.

Exact match: `deny url-prefix http://chat.yahoo.com/fishing`
`deny url-prefix http://chat.yahoo.com/music`
`deny url-prefix http://chat.yahoo.com/cooking`

Contains and contains-case operators

Regular expressions are commonly used to match simple substrings in URLs. Alternatively, you can use the ACL `contains` and `contains-case` operators. These operators provide a more efficient method for matching URL substrings.

For example, the following regex expression denies any URL matches that contain the string "sports".

`deny url matches ".*sports.*"`

In comparison, either of the following ACL rules produces the same result as the preceding regex matching expression.

```
deny url contains sports
```

```
deny url contains-case SporTS
```

The `contains` and `contains-case` operators support URL substring matching for the following ACL variables:

- ◆ `client-domain`
- ◆ `group`
- ◆ `req-headers`
- ◆ `req-line`
- ◆ `server-name`
- ◆ `server-domain`
- ◆ `user`

Characters

Characters that are commonly used to form regular expressions are shown in the following table.

Character	Description
.	Matches any single character.
*	Matches zero or more copies of the preceding expression.
[]	Matches any character within the brackets. A “-” character can be used inside brackets to specify range, for example [0-7] is the same as [01234567]. A “^” character can be used inside brackets to negate the characters specified. For example, if the first character within the brackets is a “^”, the call is to match any character except those specified within the brackets.
^	Matches the beginning of a line as the first character of a regular expression.
\$	Matches the end of a line as the last character of a regular expression.

Character	Description
{ }	Specifies how many times the previous pattern is allowed to match. For example {1,4} matches one to four occurrences of the letter A.
\	Escape. Used as part of the Ctrl-C escape sequence.
+	Matches one or more occurrence of the preceding regular expression. For example, [0,7]+ matches 1, 111, or 12345, but it does not match an empty string.
?	Matches zero or one occurrence of the preceding regular expression.
	Matches the preceding regular expression or the following regular expression.
" . . . "	Interprets every character enclosed within the quotation marks literally.
()	Groups together a series of regular expressions.

Note

For more information, see the UNIX man page on regular expression handling (man 5 regexp). In addition, see the man pages for the UNIX ed editor (man ed) for descriptions of regular expressions.

Pseudogrammar

The following table provides a variety of operators that can be used in creating complex expressions. Parsing is performed in a case-insensitive manner.

Operator	Example
AND	&&, and
OR	, or
NOT	!, not
BITRATE	[0-9]* kbps/Mbps
REGCMP	=~, matches

Operator	Example
TIME	[0-2] [0-9] (:[0-5] [0-9]) ? (am pm)
DAY	(mon tue ...), (monday tuesday ...)
SUBNET	XXX.XXX.XXX.XXX/YY, XXX.XXX.XXX.XXX/YYY.YYY.YYY.YYY
ADDR-VAR	client-ip, server-ip
PORT-VAR	client-port, server-port
VAR	cache-ip, cache-port, client-ip, client-domain, client-port, date, day, group, req-headers, server-ip, server-domain, server-name, server-port, time, url, user
ACTION	allow, deny, redirect, rewrite, auth, re-auth, try-auth
EXPN	VAR REGCMP <regexp>
EXPN	ADDR-VAR SUBNET ADDR-VAR <ip> ADDR-VAR <ip-wildcard>
EXPN	any auth ftp nntp http tunnel accel cacheable cached gopher live mms rtsp splittable streaming
EXPN	PORT-VAR <port> PORT-VAR [<port>] - [<port>]
EXPN	date <date> date [<date>] - [<date>]
EXPN	day DAY - DAY day DAY
EXPN	group <group-name>
EXPN	bitrate [BITRATE] - [BITRATE]
EXPN	server-name <name>
EXPN	smartfilter [<category-name>]
EXPN	wwfilter [<category-name>]
EXPN	time TIME - TIME
EXPN	url <url> url-prefix <prefix>
EXPN	user <user-name>

Operator	Example
EXPN	EXPN and EXPN
EXPN	EXPN or EXPN
EXPN	(EXPN)
EXPN	NOT EXPN
RULE	ACTION EXPN

About setting restrictions

Controlling access

NetCache enables you to create ACL settings that

- ◆ Control access to individual URLs or entire Web domains
- ◆ Restrict the IP addresses from which users can make Web requests
- ◆ Redirect requests for restricted objects to other URLs

Policies

The following set of rules is an example of ACL entries working together to form an access control policy:

```
deny url "http://www.corp.com"
```

```
allow time 9am - 5am
```

```
deny group marketing
```

```
deny ftp
```

In this example, the following access control policy is implemented:

- ◆ Deny access to a specified URL.
- ◆ Allow access to services during the specified time.
- ◆ Deny requests from members of the marketing group.
- ◆ Deny all FTP requests only. Requests for supported protocols other than FTP are allowed.

Ordering rules within a policy

When you form rules and create policies, it is important to consider the logical order in which ACLs are read. NetCache starts at the top on an ACL list and reads down until it finds a match. ACL rules must be ordered correctly to ensure that efficient rule matching occurs.

Example: NetCache checks global authentication before checking permissions that are applied to individual users or groups. If protocol requests are authenticated using a rule such as `auth any`, that rule must be placed above any rules that contain `user` or `group` variables. Otherwise, rule checking continues through the list, and authentication is not applied as intended.

The following policy illustrates correct order:

```
allow server-domain company.com
allow server-domain travel.com
allow client-ip 128.126.10.40
auth any
allow group execs
allow group humanresources and smartfilter job
deny any
```

This access control policy can be stated as follows:

Allow access to the company domain.

Allow access to travel.

Allow client IP-based access.

Authenticate all requests.

Allow access for members of the executive group

Allow access for members of the human resources group and allow this group access to specific Web sites.

Deny all requests.

Note

This example assumes that protocol authentication is not globally enabled on the Setup > Authentication General page.

Use caution when setting rules

The type of rule and the number of rules that NetCache must check before servicing each request can affect performance. For example, rules containing the `server-ip` variable prompt a cached DNS lookup that might be slower than using the `server-name` variable to specify server names.

In general, order your rules so that specific restrictions are listed before broad restrictions. Carefully plan the access restrictions you want to impose. Configure only the rules that are required for your environment, and where possible, collapse several rules into a single rule.

ACL optimization

Rule complexity

ACL rule complexity, matching, and the number of rules that are checked before servicing each request can affect performance. For this reason, rules and restrictive policies should be cautiously applied. Ordering rules, grouping rules, and using the correct variables to form rules are just a few techniques that ensure optimal performance when you implement secure policies for controlling access to NetCache services. These techniques include:

- ◆ Matching frequency
- ◆ Collapsing ACLS
- ◆ Using contiguous blocks
- ◆ Ordering rules
- ◆ Using contains and contains-case operators

Note

See, “[Contains and contains-case operators](#)” on page 62 for information about using these operators as an efficient alternative for matching URL substrings.

Matching frequency

To achieve optimum efficiency, order the most frequently matched rules at the top of an ACL list. You can use the Web access log files to determine the frequency at which a match occurs.

Collapsing ACL rules

Minimize the length of your ACL lists by collapsing many complex rules into one simple rule where possible. For example, the following ACL lists multiple directories that can be collapsed into one:

```
deny url-prefix http://www.company.com/monitoring
```

```
deny url-prefix http://www.company.com/administration
```

```
deny url-prefix http://www.company.com/security
```

The preceding ACL can be collapsed into the following simple rule:

```
deny server-domain company.com
```

Using contiguous blocks

Contiguous blocks are useful for optimizing performance when the same ACL action and variable occur in 10 or more rules. A contiguous list of at least 10 like rules is converted internally to a data structure that optimizes matching. For example:

```
allow client-ip 128.126.10.40
allow client-ip 128.126.10.41
allow client-ip 128.126.10.42
allow client-ip 128.126.10.43
allow client-ip 128.126.10.44
allow client-ip 128.126.10.45
allow client-ip 128.126.10.46
allow client-ip 128.126.10.47
allow client-ip 128.126.10.48
allow client-ip 128.126.10.49
allow client-ip 128.126.10.42
allow client-ip 128.126.10.43
deny any
```

NetCache supports contiguous blocks for the following variables:

```
server-name
client-name
url
client-ip
server-ip
```

Ordering rules

You can optimize performance by ordering rules according to their requirements for system resources. Network Appliance recommends that you list rules by order of impact on system resources, beginning with the rules that have the least impact on resources and ending with the rules that have the greatest impact. The following table illustrates this guideline.

Order	Description	Example
1	Absolute values directly obtainable from a request	client-ip, server-name, server-ip, url, req-header
2	Absolute values requiring an extra DNS cache lookup Caution_____ A low DNS cache hit ratio in combination with a slow DNS service can cause a serious bottleneck. If a bottleneck occurs, the impact on performance is significantly greater than any of the following types of rules.	client-name
3	Simple request patterns	url-prefix, server-domain
4	Simple regular expressions	server-name matches company
5	Content filtering application categories To optimize expressions that contain categories, use one rule for each category and form a contiguous block for two or more rules. SmartFilter examples: deny smartfilter “cults” deny smartfilter “entertainment” DynaBLocator examples: deny wwfilter “chat” deny wwfilter “sports”	cults, entertainment or chat, sports
6	Complex regular expressions	url-matches "foo([Bb][Aa][Rr])?.com"

Absolute values are significantly more efficient if they are sorted by variable.
Use contiguous blocks when the same variable occurs in 10 or more rules.

About controlling access

Controlling individual and group access

ACLs can be applied explicitly to authorized users, but they are also effective for administering access control to individuals as part of a group, such as finance.

Note—
You can create groups, add users, and assign group memberships using the Authentication feature of the Appliance Manager. See Chapter 1, “[Authenticating NetCache Users](#),” on page 1 for information about configuring the NetCache user database.

Controlling administrative access

ACLs can be applied to members of administrative groups that are assigned “Admin, Full” and “Admin, Read-Only” access permissions.

Controlling specific types of requests

To allow or deny requests for a particular protocol, for example, FTP, then carefully combine variables to limit that type of request.

Note—
Rules for controlling access to protocols can also be applied from protocol-specific pages in the Appliance Manager, for example, Setup tab > FTP > General.

Request-type rules

The following table lists examples of rules and explains how those rules govern requests.

Request	Rule	Explanation
Gopher	deny time 9am - 5pm	Denies Gopher requests between 9 a.m. and 5 p.m.
HTTP	deny smartfilter worthless	Denies HTTP requests for Web content in the SmartFilter “Worthless” category.
Tunnel	deny any	Denies all Tunnel requests.

Note

Access to protocols must be enabled before rules that control client requests apply. See Chapter 1, “[Authenticating NetCache Users](#),” on page 1 for information on enabling access to NetCache supported protocols.

Controlling access to Web content

You can allow or deny user or group access to Web objects or domains using variables and pattern-matching expressions that specify the URLs or servers to which you want to restrict access.

Global rules

The following table lists examples of rules and explains how those rules govern requests.

Rule	Explanation
<code>deny url "http://www.corp.com"</code>	Denies access to a specific URL.
<code>deny server-ip 10.10.10.11</code>	Denies access to a specific server.
<code>deny server-name matches "www\.bad\.com"</code>	Denies access to a specific server.

Note

Because requests can be made by specifying an IP address, the `server-ip` variable is more secure than specifying `server-name` for denying access to a server.

Controlling client access

You can allow or deny access, based on the IP addresses from which users make requests, by using variables and pattern-matching expressions that represent the IP addresses of a specific host, a subnet, or an entire network.

The following table lists examples of rules and explains how those rules govern requests.

Rule	Explanation
allow client-ip 192.168.1.4	Allows a specific client access to NetCache services.
deny client-ip 128.148.37	Denies a specific client subnet access to NetCache services.
deny client-domain corp.com	Denies a specific client domain access to NetCache services.
allow time 9am - 5pm	Allows access to NetCache services from 9 a.m. to 5 p.m.
deny group finance	Denies requests to access NetCache services from members of the finance group.

Redirecting requests

A redirect action will immediately send a redirect response to the client, causing the client's browser to attempt to fetch the new URL, usually through the cache. You can redirect NetCache-generated "access denied" messages to a URL explaining why the access was denied with instructions on how the user can resolve the problem.

Example: To redirect requests from server subnet 128.125.51 to an internal Web page, specify the following rule:

```
redirect "http://www.mycorp.com/restricted.html" server-ip  
128.125.51
```

Because redirect causes the client to issue a new request to the cache, a redirection loop may occur. To avoid a redirection loop, use the Global Access Control option on the Setup tab > Access Control > Access Control Lists page of the Appliance Manager to explicitly allow the redirection URL. For example:

```
allow url "http://www.mycorp.com/gambling.html"
```

Rewriting requests

A rewrite will internally rewrite a URL, causing NetCache to fetch a different object and return it to the client, if the request is allowed. You can also rewrite requests for restricted objects to other URLs by creating a pattern-matching expression or rule that represents the URL to which requests are directed.

Note

The ACL rewrite action does not support HTTP-based tunneling requests.

Example: To rewrite requests for block.com to safe.com, specify the following rule:

```
rewrite "http://www.safe.com/restrict.html" url-prefix  
http://www.block.com
```

Note

Rewrite should be used only when absolutely necessary, as relative links and inline images may not work. Use of redirect avoids this problem.

Optional syntax

Rewrite and redirect rules have the same `action expression` syntax. Optionally, however, the action can be followed by a URL that is the redirection target. The URL must be included in double quotation marks.

Example: To follow an action by a URL that is a redirection target for example, `<action> "URL" <expression>`, specify the following rule:

```
redirect "http://mycorp.com/no-gopher.html" gopher
```

Note

If a URL is not specified for a rewrite or redirect action, the Default Redirect/Rewrite URL setting in the Appliance Manager Access Control Options is used. To enter a default rewrite/redirect URL, use the following command line interface command: `set config.acl.default_url URL name`

About controlling access to streaming media

Streaming ACLs

Streaming ACL options enable you to set access controls for the streaming media features of your NetCache. Streaming bandwidth management, and cacheability rules perform ongoing regulation of streaming connections so that you can control

- ◆ Bandwidth by limiting access to bit rates
- ◆ What is cached by specifying the bit rates and content to be cached

Rules to control client access to streaming media can be created using all of the standard access control variables.

Example: To deny client 128.126.10.40 requests for streams, specify the following:

```
deny client-ip 128.126.10.40
```

Standard NetCache ACLs, however, cannot be used to manage bandwidth or cacheability because these parameters are not known until after an initial connection has been established. Bandwidth and cacheability rules are evaluated after connection setup and are periodically reevaluated throughout the duration of a streaming connection.

Note_____The access control settings that you commit on the Setup tab > Access Control > Access Control Lists page are reflected in the ACL settings options on the Setup tab > Streaming > General page.

Variables for controlling streaming media

The streaming bandwidth management and cacheability variables enable you to control the bit rates used to access streaming content.

The variables listed in the following table are restricted in their use because they apply only to streaming media.

Variable	Description
bitrate <bitrate-range>	Streaming content consumes bandwidth within bit-rate range.

Variable	Description
cacheable	Streaming content is cacheable. ◆ All live streams are not cacheable. ◆ All on-demand streams are cacheable.
cached	Streaming content is already cached.
live	Streaming requests for live content.
splittable	Streaming content is splittable.

Controlling streaming bandwidth

Streaming bandwidth rules enable you to enforce strict bandwidth policies on your network.

You cannot enter streaming bandwidth rules in standard ACLs. Bandwidth rules can only be entered in the Streaming Bandwidth Management Rules box on the Setup > Streaming > General page.

Controlling streaming cacheability

Streaming cacheability rules perform ongoing regulation of streaming connections so that you can control the content that the NetCache stores on disk.

You cannot enter streaming cacheability rules in standard ACLs. Cacheability rules can only be entered in the Streaming Cacheability Rules box on the Setup > Streaming > Caching Behavior page.

Note

For detailed information about `bitrate` and `cacheable` settings to control client access to streaming content, see the chapter on configuring NetCache for streaming media in the *Guide to Caching Protocols and Services*.

About this chapter

This chapter describes HTTPS service for caching, access control, and Web acceleration. The HTTPS implementation facilitates SSL transport for secure HTTP objects.

Topics in this chapter

This chapter provides information about the following topics:

- ◆ [“About HTTPS”](#) on page 81
- ◆ [“Authentication”](#) on page 83
- ◆ [“Controlling access”](#) on page 85
- ◆ [“HTTPS acceleration rules”](#) on page 87
- ◆ [“Planning your deployment”](#) on page 89
- ◆ [“HTTPS deployment scenarios”](#) on page 90

Procedures and option descriptions

See the Appliance Manager online Help for configuration procedures and option descriptions.

Configuration overview

Options for configuring HTTPS are located in the Appliance Manager. Configure the HTTPS feature in the order shown in the following table.

Step	Action
1	Install the HTTPS license. Setup tab > System > Licenses
2	Download certificate and key file information to the appliance. Setup tab > Authentication > SSL Contexts
3	Define an SSL context. Configure a set of parameters used for establishing a secure SSL connection. Setup tab > Authentication > SSL Contexts

Step	Action
4	Enable or disable HTTPS services. Before you enable HTTPS for any configuration, you must configure protocol support and client access to the appliance using HTTP options. Setup tab > HTTPS General
5	Create access control rules. ACL functionality for HTTPS is equivalent to HTTP ACL functionality. Some variables, however, are not supported for both HTTPS inbound and HTTPS outbound traffic. For more information, see “Authentication” on page 83. Setup tab > HTTPS > General
6	Create Web accelerator rules. Web acceleration deployments might require rules to direct HTTPS traffic. Setup tab > HTTPS > Web Server Acceleration
7	Set transparency options. Network Appliance assumes that transparency is enabled for all forward proxy deployments that support HTTPS. Setup tab > Transparency
8	Create header and caching rules. Specify what the appliance will and will not cache by creating and editing explicit caching rules. Setup tab > HTTP > Header and Caching Behavior

Note

Options for HTTP function in the same way as their HTTPS counterparts; however, connections accepted on an HTTPS port are governed only by the HTTPS configuration options. In the absence of options for controlling HTTPS traffic, rules set to control HTTP govern HTTPS objects.

About HTTPS

What is HTTPS?	HTTPS is a solution for handling nonadministrative SSL traffic. HTTPS uses the SSL protocol securely encrypt HTTP requests and responses.
SSL	SSL is an industry standard protocol for ensuring confidentiality. It is built into all major browsers and Web servers. SSL ensures confidentiality by using key cryptography algorithms in combination with a certificate signed by a Certificate Authority (CA).
Key encryption	Public key encryption is a technique that uses a pair of asymmetric keys to encrypt and decrypt messages. Key pairs consist of a public key and a private key. The public key is distributed, but the private key is kept secret. Data that is encrypted with the public key can be decrypted only with the private key. Conversely, data encrypted with the private key can be decrypted only with the public key. This asymmetry is the property that makes public key cryptography useful. For example, a client uses a server's public key to encrypt a message before sending it to the server, and the server uses a private key to decrypt the message.
Certificates	Certificates are used for the secure transfer of public keys. CA-signed certificates are signed by a mutually trusted authority for secure Web services, such as Verisign, Inc. Signed certificates provide your browser with a way to verify identity. For example, a signed message is encrypted with a private key. Devices that receive the message decrypt it with the public key. The public key is used to verify that a message is from a particular party. CA-signed certificates are the most secure method of configuring SSL for devices on the Internet.
Root certificates	A root certificate is a self-signed certificate from a CA. Root certificates are explicitly trusted and are used to authenticate other certificates. The appliance is installed with standard certificates for many CAs.

Key and certificate management

Private key files and certificates are stored in PEM (Privacy Enhanced Mail) format on the appliance's internal file system after being encrypted with a password. The password used to encrypt the private key is machine-generated on each appliance and hidden on disk to ensure that only a Data ONTAP™ process can access it. The appliance administrator will not have access to the password that is used to encrypt the private keys.

Downloading key files and certificates

HTTPS options on the Authentication - SSL Contexts page enable you to download private key, user certificate, and root certificate files from an HTTP URL to the cache. On each download page, you have the option to enter a password. This password enables the appliance to decrypt the private key that is downloaded. It must be same password that was used to encrypt the key on the host device. Once downloaded, private keys are encrypted and hidden on disk.

Note

To ensure a secure transfer, Network Appliance recommends that you place the appliance and the host device that contains the keys on a private network. Files can then be securely transferred before the appliance is returned to a public domain.

Supported encryption methods

The appliance supports the following types of encryption:

- ◆ Public key encryption
 - RSA algorithm with modulus sizes of 512, 768, 1,024, and 2,048 bits
- ◆ Data encryption
 - DES algorithm with 40- and 56-bit keys, using CBC (Cipher Block Chaining) mode
 - 3DES algorithm with 168-bit keys, using CBC and EDE (Encrypt-Decrypt-Encrypt) mode
 - RC4 algorithm with 40-, 56-, and 128-bit keys
 - RC2 algorithm with 40-bit keys, using CBC mode

Authentication

Enabling HTTPS authentication

Authentication is the process of determining a user's identity so that the user becomes known to the system. The appliance authenticates Web requests from clients using HTTPS when authentication for HTTP is enabled.

You can select and configure the type of authentication you want to use on the Authentication pages under the Setup tab in the Appliance Manager.

In addition to verifying a user's identity, authentication can be used in conjunction with access control to enforce local security domains, allocate resources, control bandwidth usage, and record traffic and filter requests based on content.

Note

See Chapter 1, “[Authenticating NetCache Users](#),” on page 1 for a description of client authentication and supported methods for authenticating administrators and users.

SSL context

All HTTPS sessions require an SSL context. SSL secures HTTP transmissions to appliance port 443 by providing an encrypted exchange between the appliance and a client.

An SSL context is a set of parameters that is required for establishing a secure SSL connection. The SSL context provides secure information, for example, encryption algorithms, certificates, keys, and other options, such as those that specify the number of SSL sessions that can be cached and how long the sessions are cached.

An SSL context is evaluated as part of a connection request for either inbound or outbound traffic.

Inbound: This SSL context is set to handle inbound connections and is evaluated for each connection request that is accepted by the appliance on an HTTPS port.

Outbound: This SSL context is set to handle outbound connections and is evaluated for each request initiated by the appliance to the origin server.

Note

Each SSL context must be associated with a context rule that specifies connections to which the context is applied. If HTTPS is enabled for outbound traffic, you must set an outbound context rule. Similarly, if HTTPS is enabled for inbound traffic, you must set an inbound context rule. For more information, see [“HTTPS acceleration rules”](#) on page 87.

Configuring an SSL context

An SSL context is defined using the following parameters:

Name

Certificate file

Private key file

Direction (applied to inbound or outbound connections)

Certificate authentication (enabled or disabled)

Supported protocol (SSL version 2, SSL version 3, TLS version 1)

Strength of data encryption (weak or strong)

Ciphers to use for this connection

Number of SSL sessions that may be cached for SSL resumption

Session timeout, to specify which sessions are not reused

Root certificates used to authenticate incoming certificates

For detailed parameter descriptions, see the Appliance Manager online Help.

Controlling access

ACLs

There are various ways to control access to the appliance by creating sets of rules, or ACLs. You can create ACL rules that control SSL inbound and outbound contexts, as well as rules that control HTTPS requests. To create ACL rules to control HTTPS access, use options on the HTTPS - General page.

Using ACLs to regulate access

Authentication is used to protect access to sensitive data, but implementing ACLs can further restrict access. ACLs are evaluated as part of a connection request, and they instruct the appliance to allow or deny access to requested content.

ACLs are associated with groups of users and types of requests, such as HTTPS. For example, the following rule denies client requests for access to the Web site specified.

```
deny url "https://www.usbank.com"
```

Controlling HTTPS inbound and outbound requests

The appliance functions as an SSL server for inbound connections, and as an SSL client for outbound connections. HTTPS inbound and outbound context ACLs are required for the appliance to determine which connections use SSL, and which contexts are used for those SSL connections.

Inbound: An inbound context specifies that settings are applied to HTTPS connections that the appliance receives.

You can specify ACL rules to control inbound HTTPS requests. Inbound configuration options are evaluated when a connection is accepted on an HTTPS port. No decryption has occurred by this time, so only variables that are unrelated to HTTP traffic are supported for this context, for example, client-ip, server-ip, and time.

Outbound: An outbound context specifies that settings are applied to HTTPS connections that the appliance sends to a secure server.

You can specify ACL rules that apply to outbound HTTPS requests. Outbound configuration options are evaluated before the appliance makes a connection to an origin server. Outbound context supports all ACL variables that are supported for HTTP.

For a feature description and information specific to writing ACL rules, see Chapter 2, “[Access Control](#),” on page 47 and the Appliance Manager online Help.

HTTPS acceleration rules

SSL context association

When using HTTPS in a reverse proxy configuration, each SSL connection to the appliance must be associated with a particular SSL context. These rules instruct the appliance on how to direct traffic in association with SSL context settings. You can create context rules using options on the New Web Server HTTPS Acceleration Rule page (Setup tab > Web Accelerator > New Web Server HTTPS Acceleration Rule).

Parameters include the following:

Web accelerator IP address

Port number

Host name

Path name

Web server IP address

Port number

Host name

Path name

You must enter the acceleration rules using the preceding eight parameters to form the following syntax:

```
132.10.10.10 80 front.org /fpath 132.10.10.15 81 back.org /bpath
```

The asterisk (*) wildcard can be used for any parameter. If you substitute an asterisk, the value you specify is “any”.

Note

You must associate an SSL context with each inbound connection. To do this, ensure that each connection that matches an acceleration rule, also matches an HTTPS inbound context rule.

In an accelerated environment, cache misses are forwarded to the accelerated server identified in the rule. If an outbound context rule matches the Web server parameter settings in the acceleration rule, the cache communicates with the server using HTTPS. Otherwise, plain HTTP is used.

Explicit rules for cacheable objects

You can specify what the appliance does and does not cache by creating and editing explicit caching rules. Rules set for HTTP function in the same way as their HTTPS counterparts; however, connections accepted on an HTTPS port are governed only by the HTTPS configuration options. In the absence of rules for controlling HTTPS traffic, rules set to control HTTP govern HTTPS objects.

Explicit caching rules determine which objects the appliance will or will not cache. You can create and edit explicit caching rules and configure additional options using the Explicit Caching Rules tab on the HTTP - Header and Caching Behavior page.

Creating rules for HTTP object handling enable you to

- ◆ Specify types of content to be proxied and cached
- ◆ Configure the appliance to drop or maintain specified client connections
- ◆ Specify how the appliance will respond to canceled requests and reload requests
- ◆ Specify object refresh rates
- ◆ Control user access to HTTP services
- ◆ Configure the appliance to not provide the client IP address to the Web server
- ◆ Suppress error messages that refer to the presence of a proxy-cache server to clients

For parameter descriptions, instructions for forming rules, and examples, see the Appliance Manager online Help.

Planning your deployment

Web acceleration

An HTTPS-enabled Web accelerator works to offload secure processing (encryption and decryption) from an origin server. In this environment, the appliance is exposed to clients, and connections are made over a private network to an HTTP server.

Example: An appliance configured to support HTTPS Web site acceleration is installed with the certificate and private key of the accelerated server. Traffic from the HTTPS client to the cache is encrypted. Also, traffic from the cache to the accelerated server can be encrypted, although in many cases, HTTP is sufficient.

Forward proxy

An HTTPS forward proxy handles client requests for the origin server. In this environment, client browsers use HTTPS to communicate with the appliance, and the appliance uses HTTPS to communicate with the origin server.

Example: A site with an existing SSL application between the branch office and the central office can deploy an appliance at the branch office to offload traffic from the central office servers. This appliance has a copy of the central office's certificate and private keys, and is therefore able to terminate client SSL connections and decrypt traffic. The appliance acts as the central office server, and encrypts traffic from the appliance to the central office origin server, as well.

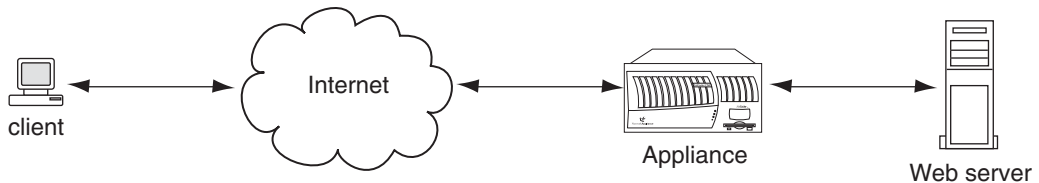
HTTPS deployment scenarios

Configuring the HTTPS service

HTTPS service can be configured to manage inbound and outbound SSL contexts in forward proxy and reverse proxy environments. The following scenarios provide examples of settings for a variety of HTTPS deployments.

Web acceleration

The following two examples illustrate the appliance deployed as a reverse proxy, or Web accelerator. In a Web accelerated environment, secure traffic is offloaded from the Web server to the appliance.



Web acceleration example 1

The accelerated environment supports HTTPS traffic between the client and the cache, and HTTPS traffic between the cache and the origin server.

The following SSL context parameters are assumed:

SSL context: (Setup tab > Authentication > SSL Contexts)

Inbound context name = inctx

Outbound context name = outctx

Appliance name = ncache.com

Accelerated server name = server.com

Accelerated server IP address = 10.24.155.13

The following HTTPS parameter settings are required:

HTTPS: (Setup tab > HTTPS > General)

HTTPS = enabled

```
HTTPS inbound context acl = context "inctx" cache-port 443
```

```
HTTPS outbound context acl = context "outctx" server-port 443 and  
server-ip 10.24.155.13
```

Acceleration rule: (Setup tab > HTTPS > Web Server Acceleration)

```
HTTPS Web acceleration = enabled
```

```
HTTPS acceleration rule = * 443 ncache.com * 10.24.155.13 443  
server.com *
```

Web acceleration example 2

The accelerated environment supports HTTPS traffic between the client and the cache, and HTTP traffic between the cache and the origin server.

The following SSL context parameters are assumed:

SSL context: (Setup tab > Authentication > SSL Contexts)

```
Inbound context name = inctx
```

```
Outbound context name = outctx
```

```
Appliance name = ncache.com
```

```
Accelerated server name = server.com
```

```
Accelerated server IP address = 10.24.155.13
```

The following HTTPS parameter settings are required:

HTTPS: (Setup tab > HTTPS > General)

```
HTTPS = enabled
```

```
HTTPS inbound context acl = context "inctx" cache-port 443
```

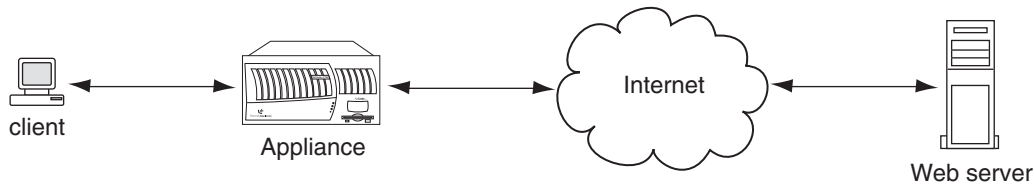
Acceleration rule: (Setup tab > HTTPS > Web Server Acceleration)

```
HTTPS Web acceleration = enabled
```

```
HTTPS acceleration rule = * 443 ncache.com * 10.24.155.13 80  
server.com *
```

Forward proxy

The following example illustrates the appliance deployed as a forward proxy. In a forward proxy environment, the appliance handles requests from client Web browsers to an origin server.



Forward proxy example

In an HTTPS forward proxy configuration, client browsers use HTTPS to communicate with the appliance, and the appliance uses HTTPS to communicate with the origin server.

The following SSL context parameters are assumed:

SSL context: (Setup tab > Authentication > SSL Contexts)

Inbound context name = inctx

Outbound context name = outctx

Web server name = server.com

Web IP address = 10.24.155.13

The following HTTPS parameter settings are required:

HTTPS: (Setup tab > HTTPS > General)

HTTPS = enabled

HTTPS inbound context acl = context "inctx" server-port 443 and server-ip 10.24.155.13

HTTPS outbound context acl = context "outctx" server-port 443 and server-ip 10.24.155.13

About this chapter	This chapter describes the WebWasher DynaBLocator content filtering software. Network Appliance has developed a partnership with the WebWasher corporation to provide DynaBLocator as part of its NetCache features.
Topics in this chapter	This chapter provides information about the following topics: ◆ “About DynaBLocator” on page 94 ◆ “About the DynaBLocator URL database” on page 96 ◆ “DynaBLocator configuration overview” on page 98 ◆ “Basic DynaBLocator options” on page 100 ◆ “DynaBLocator URL database updates” on page 101 ◆ “DynaBLocator blocking with ACLs” on page 102 ◆ “About testing and troubleshooting DynaBLocator” on page 105 ◆ “About rolling out DynaBLocator service to your users” on page 106
Procedures and option descriptions	See the Appliance Manager online Help for configuration procedures and option descriptions.
Location of configuration options	Options for configuring the features discussed in this chapter are in the following location in the Appliance Manager: Setup tab > Access Control > WebWasher DynaBLocator

About DynaBLocator

What is DynaBLocator?

DynaBLocator software, included with NetCache, enables you to deny your users access to Web sites that contain objectionable content. DynaBLocator identifies Web sites whose HTTP content matches DynaBLocator content categories and includes lists of those sites in its URL database. The sites that DynaBLocator actually blocks depends on the categories you select when you set up content blocking for your organization.

DynaBLocator includes 58 categories for Web sites that can be blocked, including categories for sites related to pornography and nudity, educational, societal, and religious organizations, extremist activities, criminal activities, games and gambling, entertainment and culture, information and communication, drugs, lifestyle, finance and investing, medicine, and more. For a full list of supported DynaBLocator content categories, see the Appliance Manager online Help by selecting Setup tab > Access Control > WebWasher DynaBLocator.

The DynaBLocator URL database is continually updated through the following methods:

- ◆ Searches of the Internet for URLs with suspicious images
- ◆ Searches of the Internet in which a linguistic-semantic method is used to categorize Web sites based on their textual content

Requirement to use DynaBLocator

If your appliance has less than 512 MB of memory, which is the case with the C1100 and C720S2 models, the Optimize Memory option on the WebWasher - DynaBLocator page in the Appliance Manager must be enabled. The Optimize Memory option is enabled by default. DynaBLocator requires an appliance with a minimum of 512 MB of memory unless the Optimize Memory option is enabled.

Only one content filtering method can be enabled

DynaBLocator is one of two content filtering methods that NetCache provides. You can enable only one method at a time. Therefore, if you enable DynaBLocator, you cannot enable SmartFilter at the same time.

Obtaining a DynaBLocator subscription

To use the DynaBLocator feature on NetCache, you must first purchase a subscription to the DynaBLocator URL database. See the NOW Web site (<http://now.netapp.com>) for details. You will receive a customer ID that enables you to download the DynaBLocator URL database.

Evaluating DynaBLocator: If you want to evaluate DynaBLocator before determining whether to purchase a subscription, you can do so by filling out the DynaBLocator evaluation form on the NOW Web site at <http://now.netapp.com/>. Then follow the instructions in “[DynaBLocator configuration overview](#)” on page 98 to set up this NetCache appliance for DynaBLocator content blocking service.

About the DynaBLocator URL database

About this section

The DynaBLocator URL database includes lists of Web sites that contain HTTP content that match the DynaBLocator categories. This section describes the DynaBLocator URL database, discusses how you obtain the URL database initially, and makes recommendations for how often you need to update the URL database that is on your appliance.

Types of entries in the DynaBLocator URL database

The DynaBLocator URL database contains a list of URLs and IP addresses that match the DynaBLocator categories. This variety in entry type enables you to block access to Web content at various levels, as the following table shows.

Entry type	Example	The entry blocks...
Site	www.aabbcc.com	The entire Web page and all of its content
URL	www.aabbcc.com/pics/	A specific directory on a Web page
Path	www.aabbcc.com/pics/001.jpg	A specific object (picture or document) in a directory on the Web page
IP address	1.1.1.1	Access to the Web server with the IP address 1.1.1.1

Entire sites are not blocked because of content that is in a specific directory. For example, the following entry does not block the entire site www.abcinc.com:

www.abcinc.com/jobs/jobs.html

Obtaining the DynaBLocator URL database

After you subscribe to DynaBLocator or when you request a DynaBLocator evaluation, you will receive a customer ID for the Web site from which you can download the DynaBLocator URL database. *See the* NOW Web site at <http://now.netapp.com/> for details.

Frequency of URL database updates

The DynaBLocator URL database is updated and released by the WebWasher corporation twice a day. Network Appliance recommends that you update the URL database at the frequency at which the WebWasher corporation updates and releases it.

How updates are provided

After the initial download of the DynaBLocator URL database, only the changes to the URL database are downloaded to your NetCache appliance.

How Web sites can be added to the URL database

If you find a Web site that needs to be added to the URL database, send your request to the WebWasher corporation at the following e-mail address:

dynablocator-feedback@webwasher.com

DynaBLocator configuration overview

Requirements to use DynaBLocator

To use the DynaBLocator feature, you must do the following:

1. Obtain a licensed customer ID to allow you to download the URL database.
Purchase a WebWasher DynaBLocator subscription, as described in [“Obtaining a DynaBLocator subscription”](#) on page 95. Alternatively, you can evaluate DynaBLocator.
2. Set up the DynaBLocator feature on this NetCache appliance, as described in [“Basic DynaBLocator options”](#) on page 100.
3. Regularly obtain new (updated) versions of the DynaBLocator URL database.

Network Appliance recommends that you set up NetCache to automatically obtain an updated URL database to ensure that the most recent list of potentially objectionable sites is always available. See [“DynaBLocator URL database updates”](#) on page 101 for information about obtaining a new URL database manually and automatically.

Ways of setting up DynaBLocator content blocking in NetCache

You can set up DynaBLocator content blocking in two ways, as shown in the following table.

Method of configuring blocking for DynaBLocator	When you would use this method...
Selecting DynaBLocator categories on the WebWasher - DynaBLocator pages in the Appliance Manager	To block one or more categories of content as a global setting for your users
Setting access control rules for DynaBLocator	To make exceptions to the policies that you have set up for entire content categories. For example, you might ◆ Exclude URLs from DynaBLocator categories you have blocked ◆ Override blocking of content for specific users

Method of configuring blocking for DynaBLocator	When you would use this method...
Setting category restrictions by authentication group	To block a category that you have not blocked on the Access Control - WebWasher - DynaBLocator page (that is, for all users) for a specific group of authenticated users

Although you could use ACLs to block entire categories of content, the easiest way to block categories is to block the content categories on the WebWasher - DynaBLocator pages. Then use ACLs to set exceptions to your DynaBLocator category settings. See “[DynaBLocator blocking with ACLs](#)” on page 102 for more information.

Note

If you decide to use only ACL rules for DynaBLocator blocking, you must also set up some basic DynaBLocator options, as described in “[Basic DynaBLocator options](#)” on page 100.

Precedence of access control settings

NetCache access control list options have precedence over DynaBLocator options. NetCache checks all ACL rules that are set before it checks content category restrictions that are set in the DynaBLocator options.

Using NetCache ACLs is helpful, for example, if a DynaBLocator content category restricts a URL that you want one or more of your users to be able to access. You can explicitly allow the URL by using NetCache access control options. All other URLs in the DynaBLocator content category continue to be restricted, but the users you specified in the ACL can access the URL that you explicitly allowed for them.

Settings for user authentication groups are checked after ACLs and before category settings on the WebWasher - DynaBLocator page.

Basic DynaBLocator options

Location of options and procedures

Whether you plan to set up your DynaBLocator category blocking on the WebWasher - DynaBLocator pages or through ACLs, you must set certain basic options on the WebWasher - DynaBLocator page in the Appliance Manager.

Setting up basic DynaBLocator options

For instructions on how to set up basic DynaBLocator options, see the Appliance Manager online Help.

DynaBLocator information in the Web access log

If you want information about DynaBLocator to be recorded in the Web access log, you must specify the following headers when you configure logging for the Web access log:

x-wwfilter-categories

x-wwfilter-result

During testing and troubleshooting, it is helpful to review information in the Web access log. A DynaBLocator log entry for a particular Web site shows all DynaBLocator categories associated with a specific URL.

The setting of the Optimize Memory option on the WebWasher - DynaBLocator page in the Appliance Manager affects the logging of DynaBLocator entries in the Web access log. If the Optimize Memory option is enabled and you have explicitly allowed categories through ACLs or have blocked any categories, the Web access log does not show any entries for DynaBLocator.

To configure custom headers for the Web access log, select Setup tab > HTTP > Logging.

DynaBLocator URL database updates

Ways you can update the URL database

You can update the DynaBLocator URL database automatically or manually. To ensure that your URL database is always up-to-date, Network Appliance recommends that you set up DynaBLocator for automatic updates. Bandwidth consumption resulting from downloading URL database updates should not be a concern. The reason is that only the changes to the URL database are downloaded.

For instructions on how to set up automatic updating of the DynaBLocator URL database, see the Appliance Manager online Help.

For instructions on how to obtain the updated DynaBLocator URL database manually, see the Appliance Manager online Help.

How often to update the URL database

Network Appliance recommends that you update the URL database twice each day—the same frequency at which the WebWasher corporation updates the file.

DynaBLocator blocking with ACLs

About this section

This section contains a few examples of how you might use ACL rules with DynaBLocator.

Before setting ACLs for DynaBLocator

Before setting ACLs for DynaBLocator, you must configure basic DynaBLocator options, as described in “[Basic DynaBLocator options](#)” on page 100. If DynaBLocator is not enabled, NetCache ignores DynaBLocator ACLs.

Ways of configuring ACLs for DynaBLocator

Methods of configuring ACLs for DynaBLocator are shown in the following table.

Method	Description
Create standard HTTP ACL rules on the Access Controls pages (manually entered ACL rules)	◆ This method enables you to use a full suite of building blocks for creating ACLs. For example, you can allow some users access to specific URLs and deny other users; you can deny or allow access to specific URLs. ◆ This method requires sound knowledge of ACL building blocks and syntax.
Use the ACL wizard for DynaBLocator	◆ You can allow some URLs in DynaBLocator categories that you have blocked globally. The ACL wizard can be used to allow only URLs containing phrases that you specify. ◆ This method is a simple way of creating ACLs because it does not require knowledge of ACL building blocks and syntax. ◆ The Exceptions to Full Category Blocking option on the WebWasher - DynaBLocator page is the ACL wizard.

DynaBLocator restrictions are checked after ACLs

NetCache invokes restrictions associated with DynaBLocator in the following order:

1. Standard ACLs (that is, ACLs set on the Access Controls pages) or ACL rules set through the ACL wizard (that is, the Exceptions to Full Category Blocking option on the WebWasher - DynaBLocator page).

NetCache evaluates the settings in the Exceptions to Full Category Blocking option after the standard ACLs.

2. Any category blocking that you might have set for user authentication groups.
3. Category restrictions set on the WebWasher - DynaBLocator page.

Examples of when you might use an ACL

The benefit of NetCache invoking ACL rules before checking DynaBLocator category restrictions set on the WebWasher - DynaBLocator page is that you can set ACL rules to allow exceptions to the categories for which you have blocked access. For example, after you enable DynaBLocator and specify content categories to be restricted, you could create an ACL rule to

- ◆ Allow access to a specific URL that DynaBLocator would block because it matches a category that you have blocked

For instructions on how to exclude a URL from DynaBLocator blocking, see the Appliance Manager online Help

- ◆ Allow a specific user access to a content category that you have blocked. (This is not possible to set up with the ACL wizard.)

For instructions on how to allow a specific user access to a blocked DynaBLocator category, see the Appliance Manager online Help.

Examples of ACL rules with the DynaBLocator ACL wizard

Assume that you have blocked the category Software and Hardware Vendors/Distributors, but you wanted to allow access to the Network Appliance Web site. You could enter any of the following in the Exceptions to Full Category Blocking option on the WebWasher - DynaBLocator page to allow access to the Network Appliance Web site:

`http://www.netapp.com`
`www.netapp.com`
`netapp.com`

If you enter a phrase, it must be an exact match. The phrase “net app” does not return `http://www.netapp.com` because of the embedded white space.

Examples of ACL rules with standard ACLs

The following examples show ACL rules that exempt specific URLs from blocking.

Example 1: The following rule shows how to specify either a URL string or a regular expression of a set of URLs to be exempted from blocking:

```
allow url-prefix http://www.corp.com
```

This rule explicitly allows access to a URL blocked by DynaBLocator for all users of the NetCache service.

Example 2: The following rule shows how to specify an IP address or range of IP addresses, in addition to a regular expression of a set of URLs, to be exempted from DynaBLocator blocking:

```
allow client-ip 128.125.51 and url-prefix http://www.corp.com
```

This rule explicitly allows access to a URL only for specified clients.

See Chapter 2, “[Access Control](#),” on page 47 for details about ACLs.

About testing and troubleshooting DynaBLocator

Confirming DynaBLocator category blocking

After blocking a DynaBLocator category (on the WebWasher - DynaBLocator page or through an ACL rule), you might want to confirm that blocking is in effect for the category you specified.

For instructions on how to test your DynaBLocator setup, see the Appliance Manager online Help.

Use the Web access log to help test and troubleshoot

If you have configured the Web access log to include DynaBLocator headers, the Web access log will show the status of a URL request and its corresponding DynaBLocator category. Therefore, if the category is shown in the log along with the URL, the URL is present in the DynaBLocator URL database.

About rolling out DynaBLocator service to your users

Informing users about content blocking software

Before you start using DynaBLocator on your network, you might want to inform your users that you are adding a content blocking service to the network and identify the type of content that you plan to block. You can also create a Web page that informs users why they are unable to reach a Web site. You can use the “Redirect Blocked Requests to This URL” option on the WebWasher - DynaBLocator page to identify the page to which NetCache is to redirect users when requests are blocked. See “[Basic DynaBLocator options](#)” on page 100 for more information.

About this chapter

This chapter describes the SmartFilter content filtering software. Network Appliance has developed a partnership with Secure Computing to provide SmartFilter as part of its NetCache features.

Topics in this chapter

This chapter provides information about the following topics:

- ◆ [“About SmartFilter”](#) on page 108
- ◆ [“About SmartFilter control files”](#) on page 110
- ◆ [“SmartFilter configuration overview”](#) on page 112
- ◆ [“Basic SmartFilter options”](#) on page 114
- ◆ [“SmartFilter control file updates”](#) on page 115
- ◆ [“SmartFilter blocking with ACLs”](#) on page 116
- ◆ [“About testing and troubleshooting SmartFilter”](#) on page 119
- ◆ [“About rolling out SmartFilter service to your users”](#) on page 120

Procedures and option descriptions

See the Appliance Manager online Help for configuration procedures and option descriptions.

Location of configuration options

Options for configuring the features discussed in this chapter are in the following location in the Appliance Manager:

Setup tab > Access Control > SmartFilter

About SmartFilter

What is SmartFilter?

SmartFilter software, included with NetCache, enables you to deny your users access to Web sites that contain objectionable content. SmartFilter identifies Web sites whose HTTP content matches SmartFilter content categories and includes lists of those sites in its *control files*. The sites that SmartFilter actually blocks depends on the categories you select when you set up content blocking for your organization.

SmartFilter categories represent content categories considered objectionable in the context of a standard business or educational environment. SmartFilter categories include chat, criminal skills, drugs, entertainment, gambling, games, hate speech, humor, investments, politics, sex, sports, and more. For a full list of supported SmartFilter content categories for each SmartFilter version, see the Appliance Manager online Help by selecting Setup tab > Access Control > SmartFilter.

Automated tools continuously search the Internet for new sites and pages that meet the criteria for exclusion as designated by SmartFilter categories. Secure Computing control list technicians typically review all potential candidates for exclusion before they are added to the control lists.

Only one content filtering method can be enabled

SmartFilter is one of two content filtering methods that NetCache provides. You can only enable one method at a time. Therefore, if you enable SmartFilter, you cannot enable WebWasher DynaBLocator at the same time.

Obtaining a SmartFilter subscription

To use the SmartFilter feature on NetCache, you must first purchase a subscription to the SmartFilter control list by contacting ibg@netapp.com or your sales representative. You will receive a user name and password and an FTP address from which you can download the SmartFilter control files. Be sure to indicate the version of SmartFilter that you want to use.

Evaluating SmartFilter: If you want to evaluate SmartFilter before determining whether to purchase a subscription, you can do so by filling out the SmartFilter evaluation form on the NOW Web site at <http://now.netapp.com/>. After you complete the evaluation form, you will receive instructions about how

to obtain the SmartFilter control files. Then follow the instructions in [“SmartFilter configuration overview”](#) on page 112 to set up this NetCache appliance for SmartFilter content blocking service.

About SmartFilter control files

About this section SmartFilter control files include lists of Web sites that contain HTTP content that match the SmartFilter categories. This section describes the SmartFilter control files, discusses how you obtain control files initially, and makes recommendations for how often you need to update the control files.

SmartFilter 3.0.1 requirement The SmartFilter control file version 3.0.1 requires a NetCache appliance with a minimum of 512 MB of memory. Therefore, it cannot be used on the NetCache C1100 and C720S2 appliances.

The SmartFilter control file The SmartFilter control file (sfcontrol) contains a list of URLs that match SmartFilter categories. Users can access sites for categories that you have blocked if they include an IP address rather than a host name in the URL. For example, if you block the Nudity category, a user who includes the IP address that maps to www.playboy.com in the URL can access the site. NetCache provides an option called SmartFilter IP Address Blocking, that blocks access to *all* content by IP address, regardless of category

Types of entries in a SmartFilter control list The sfcontrol file contains the types of entries that are shown in the following table. This variety in entry type enables you to block access to Web content at various levels, as the following table shows.

Entry type	Example	The entry blocks...
Site	www.aabbcc.com	The entire Web page and all of its content
URL	www.aabbcc.com/pics/	A specific directory on a Web page
Path	www.aabbcc.com/pics/001.jpg	A specific object (picture or document) in a directory on the Web page

Entire sites are not blocked because of content that is in a specific directory. For example, the following entry does not block the entire site www.abcinc.com:

www.abcinc.com/jobs/jobs.html

Determining whether a particular site is in a control list

SmartFilter control lists are in a format that only the SmartFilter software can use. You cannot view them. However, you can use the SmartFilterWhere tool on the Secure Computing Web site to check whether a specific site is contained in the current SmartFilter control list. To access the SmartFilterWhere tool, go to <http://www.smartfilterwhere.com>.

Obtaining SmartFilter control files

After you purchase SmartFilter or when you request a SmartFilter evaluation, you will receive a user name, password, and FTP address that you need to use to obtain control files. Go to the NOW Web site at <http://now.netapp.com/> to request a user name, password, and FTP address to use for SmartFilter.

Expiration of SmartFilter control files

Because content on the Web changes rapidly, the SmartFilter control files expire after one month. Starting one week before the files expire, NetCache sends an e-mail message to Autosupport recipients every day warning that the files are about to expire.

Caution

When the SmartFilter control files expire, NetCache blocks all Web requests because it is unable to determine which requests to allow or deny. For this reason, be sure to replace your SmartFilter control files before they expire.

SmartFilter configuration overview

Requirements to use SmartFilter

To use the SmartFilter feature, you must do the following:

1. Determine the version of SmartFilter that you want to use.
2. Obtain a licensed user name and password and the location of the control files.

Purchase a SmartFilter subscription, as described in [“Obtaining a SmartFilter subscription”](#) on page 108. Alternatively, you can evaluate SmartFilter for three weeks. See [“Evaluating SmartFilter”](#) on page 108.

3. Set up the SmartFilter feature on this NetCache appliance, as described in [“Basic SmartFilter options”](#) on page 114.
4. Regularly obtain new (updated) versions of SmartFilter control files.

Network Appliance recommends that you set up NetCache to automatically obtain updated control lists to ensure that the most recent list of potentially objectionable sites is always available. See [“SmartFilter control file updates”](#) on page 115 for information about obtaining new control files manually and automatically.

Ways of setting up SmartFilter content blocking in NetCache

You can set up SmartFilter content blocking as shown in the following table.

Method of configuring blocking for SmartFilter	When you would use this method...
Selecting SmartFilter categories on the SmartFilter pages in the Appliance Manager.	To block one or more categories of content as a global setting for your users
Setting access control rules for SmartFilter	To make exceptions to the policies that you have set up for entire content categories. For example, you might ◆ Exclude URLs from SmartFilter categories you have blocked ◆ Override blocking of content for specific users

Method of configuring blocking for SmartFilter	When you would use this method...
Setting category restrictions by authentication group	To block a category that you have not blocked on the Access Control - SmartFilter page (that is, for all users) for a specific group of authenticated users

Although you could use ACLs to block entire categories of content, the easiest way to block categories is to block the content categories on the SmartFilter pages. Then use ACLs to set exceptions to your SmartFilter category settings. See “[SmartFilter blocking with ACLs](#)” on page 116 for more information.

Note

If you decide to use only ACL rules for SmartFilter blocking, you must also set up some basic SmartFilter options, as described in “[Basic SmartFilter options](#)” on page 114.

Precedence of access control settings

NetCache access control list options have precedence over SmartFilter options. NetCache checks all ACL rules that are set before it checks content category restrictions that are set in the SmartFilter options.

Using NetCache ACLs is helpful, for example, if a SmartFilter content category restricts a URL that you want one or more of your users to be able to access. You can explicitly allow the URL by using NetCache access control options. All other URLs in the SmartFilter content category continue to be restricted, but the users you specified in the ACL can access the URL that you explicitly allowed for them.

Basic SmartFilter options

Location of options and procedures

Whether you plan to set up your SmartFilter category blocking on the SmartFilter pages or through ACLs, you must set certain basic options on the Access Control - SmartFilter page in the Appliance Manager.

After setting up your basic options, you must also obtain SmartFilter control files, as described in [“SmartFilter control file updates”](#) on page 115.

The options displayed vary according to SmartFilter version

The options displayed on the SmartFilter page vary according to the version of SmartFilter that is selected. If you change the selected version, you must click Commit Changes at the bottom of the page before the options associated with the newly selected version are displayed.

Setting up basic SmartFilter options

For instructions on how to set up basic SmartFilter options, see the Appliance Manager online Help.

SmartFilter information in the log

If you want information about SmartFilter to be recorded in the Web access log, you must specify the following headers when you configure logging for the Web access log:

x-smartfilter-categories

x-smartfilter-result

During testing and troubleshooting, it is helpful to review information in the Web access log. A SmartFilter log entry for a particular Web site shows all SmartFilter categories associated with a specific URL.

To configure custom headers for the Web access log, select Setup tab > HTTP > Logging.

SmartFilter control file updates

Ways you can update your control files

You can update your control files in either of the following ways:

- ◆ Automatically, by using the options on the Install from URL tab on the SmartFilter page
- ◆ For instructions on how to set up automatic updating of SmartFilter control files, see the Appliance Manager online Help.
- ◆ Manually, by using the options on the Install from Your Computer tab on the SmartFilter page
- ◆ For instructions on how to obtain updated SmartFilter control files manually, see the Appliance Manager online Help.

To access the SmartFilter page in the Appliance Manager, select Setup tab > Access Control > SmartFilter.

How often to update control files

Whether you decide to update control files manually or automatically, Network Appliance recommends you update the control file twice weekly.

SmartFilter blocking with ACLs

About this section

This section contains a few examples of how you might use ACL rules with SmartFilter.

Before setting ACLs for SmartFilter

Before setting ACLs for SmartFilter, you must configure basic SmartFilter options, as described in “[Basic SmartFilter options](#)” on page 114. If SmartFilter is not enabled, NetCache ignores SmartFilter ACLs. Additionally, if you specify a category in your ACL that is not applicable to the SmartFilter version selected on the SmartFilter page, NetCache will ignore the rule.

Ways of configuring ACLs for SmartFilter

Methods of configuring ACLs for SmartFilter are shown in the following table.

Method	Description
Create standard HTTP ACL rules on the Access Controls pages (manually entered ACL rules)	◆ This method enables you to use a full suite of building blocks for creating ACLs. For example, you can allow some users access to specific URLs and deny other users; you can deny or allow access to specific URLs. ◆ This method requires sound knowledge of ACL building blocks and syntax.
Use the ACL wizard for SmartFilter	◆ You can allow some URLs in SmartFilter categories that you have blocked globally. The ACL wizard can be used to allow only URLs containing phrases that you specify. ◆ This method is a simple way of creating ACLs because it does not require knowledge of ACL building blocks and syntax. ◆ The Exceptions to Full Category Blocking option on the SmartFilter page is the ACL wizard.

SmartFilter restrictions are checked after ACLs

NetCache invokes restrictions associated with SmartFilter in the following order:

1. Standard ACLs (that is, ACLs set on the Access Controls pages) or ACL rules set through the ACL wizard (that is, the Exceptions to Full Category Blocking option on the SmartFilter page).

NetCache evaluates the settings in the Exceptions to Full Category Blocking option after the standard ACLs.

2. Any category blocking that you might have set for user authentication groups.
3. Category restrictions set on the SmartFilter page.

Examples of when you might use a SmartFilter ACL

The benefit of NetCache invoking ACL rules before checking SmartFilter category restrictions is that you can set ACL rules to allow exceptions to the categories for which you have blocked access. For example, after you enable SmartFilter and specify content categories to be restricted, you could create an ACL rule to

- ◆ Allow access to a specific URL that SmartFilter would block because it matches a category that you have blocked
For instructions on how to exclude a URL from SmartFilter blocking, see the Appliance Manager online Help.
- ◆ Allow a specific user access to a content category that you have blocked. (This is not possible to set up with the ACL wizard.)
For instructions on how to allow a specific user access to a blocked SmartFilter category, see the Appliance Manager online Help.

Examples of ACL rules with the SmartFilter ACL wizard

Assume that you have blocked the category Travel, but you wanted to allow access to the Web site <http://www.earthwatch.org>. You could enter any of the following in the Exceptions to Full Category Blocking option on the SmartFilter page to allow access to the <http://www.earthwatch.org> Web site:

<http://www.earthwatch.org>
www.earthwatch.org
[earthwatch.org](http://www.earthwatch.org)

If you enter a phrase, it must be an exact match. The phrase “earth watch” does not return <http://www.earthwatch.org> because of the embedded white space. However, the phrase “earthwatch” returns <http://www.earthwatch.org>.

Examples of standard ACL rules with the SmartFilter ACL wizard

The following examples show ACL rules that exempt specific URLs from SmartFilter blocking.

Example 1: The following rule shows how to specify either a URL string or a regular expression of a set of URLs to be exempted from SmartFilter blocking:

```
allow url-prefix http://www.corp.com
```

This rule explicitly allows access to a URL blocked by SmartFilter for all users of the NetCache service.

Example 2: The following rule shows how to specify an IP address or range of IP addresses, in addition to a regular expression of a set of URLs, to be exempted from SmartFilter blocking:

```
allow client-ip 128.125.51 and url-prefix http://www.corp.com
```

This rule explicitly allows access to a URL only for specified clients.

See Chapter 2, “[Access Control](#),” on page 47 for details about ACLs.

About testing and troubleshooting SmartFilter

Confirming SmartFilter category blocking

After blocking a SmartFilter category (on the SmartFilter page or through an ACL rule), you might want to confirm that blocking is in effect for the category you specified.

For instructions on how to test your SmartFilter setup, see the Appliance Manager online Help.

About rolling out SmartFilter service to your users

Informing users about content blocking software

Before you start using SmartFilter on your network, you might want to inform your users that you are adding a content blocking service to the network and identify the type of content that you plan to block. You can also create a Web page that informs users why they are unable to reach a Web site. You can use the “Redirect Blocked Requests to This URL” option on the SmartFilter page to identify the page to which NetCache is to redirect users when requests are blocked. See “[Basic SmartFilter options](#)” on page 114 for more information.

Index

A

access controls

- access control lists 49
 - building 52
- ACL rule checking 49
- action expression syntax 52
- authorized access 49
- client access 73
- commonly used characters 63
- description of 49
- DynaBLocator
 - configuration overview 98
 - evaluating 95
 - exempting URLs from restriction 104
 - precedence of ACL 99
 - URL database 96
- expression matching 61
- expression types 59
- operators 60
- optional syntax 75
- redirecting requests 74
- regular expressions 61
- rewriting requests 74
- rules 52
 - examples 73
 - forming 67
 - guidelines for setting 68
 - specific requests 72
- SmartFilter 108
 - configuration overview 112
 - control files 110
 - evaluating 108
 - exempting URLs from restriction 118
 - precedence of ACL checking over 113
 - purchasing subscription 108
 - requirements for 112
- streaming bandwidth 77
- streaming cacheability 77
- streaming media 76
- variables 54
- WebWasher
 - configuration overview 98

- evaluating 95
- exempting URLs from restriction 104
- precedence of ACL 99
- URL database 96

ACL wizard

- with DynaBLocator 102
- with SmartFilter 116

authentication

- about 3
- configuring 12
 - options 12
- expired password URL 28
- FTP 4
- Gopher 4
- groups 11
- HTTP 4
- HTTPS 83
- Kerberos 15, 30
 - backup protocols 30
 - configuring browsers 31
 - testing 31

LDAP 15

- password expiration 18

logs 13

MMS 4

NetCache native 10

NetCache user database 6, 10

NNTP 4

NTLM 15, 18

- testing 28

RADIUS 15

RTSP 4

server information, logging of 13

SSL 4

streaming media cache 13

testing of 13

users 11

- authentication methods 4

video on demand 14

when NetCache authenticates to server 7

authentication servers 3

- Kerberos 1, 5

- LDAP 1, 5
- NTLM 1
- preparing to configure 18, 22
- RADIUS 1, 5
- setting up 18, 22
- testing 13

C

- certificates, SSL 81
- control files, SmartFilter 110

D

- DynaBLocator
 - ACL wizard with 102
 - blocking with ACLs 102
 - prerequisites 102
 - when to use 103
 - configuration overview 98
 - configuring
 - basic options 100
 - content that can be restricted 94
 - description of 94
 - evaluating 95
 - exempting a user from restriction 103
 - exempting URLs from restriction 99, 103
 - log entries for 100
 - precedence of ACL checking 99
 - URL database 96, 101
 - automatic updates, setting up 101
 - entries in 96
 - frequency of updates 97
 - how often to update 101
 - obtaining 96
 - updating 101
 - URL database contents 96
 - ways of blocking 98
 - selecting categories 98
 - using ACLs 98
- DynaBLocatorr
 - exempting URLs from restriction 103

E

- encryption, HTTPS 82

H

HTTPS

- ACLs 85
- authentication 83
- deployment planning 89
- deployment scenarios 90
- encryption 82
- inbound connections 85
- key files and certificates 82
- outbound connections 85
- SSL context 83

K

- Kerberos 15, 30
 - authentication server 5
 - backup protocols 30
 - configuring browsers for 31
 - testing authentication 31
- key encryption, public and private 81

L

LDAP

- authentication server 5
- RADIUS 15
- requirements 18
- setting up the LDAP server port 18

N

- NTLM 15, 18
 - testing authentication 28

P

- password
 - NTLM password renewal 28

R

RADIUS

- authentication servers 5
- root certificates 81

S

sfcontrol file, obtaining 111

SmartFilter

- ACL wizard with 116

- blocking with ACLs 116

 - prerequisites 116

 - when to use 117

- configuration overview 112

- configuring

 - ACLs with 116

 - basic options 114

 - control file updates 115

- content that can be restricted 108

- control files 110

 - determining sites in 111

 - expiration 111

 - how often to update 115

 - loading 115

 - obtaining 108

 - types of entries in 110

 - updating 115

- control files, obtaining 111

- description of 108

- evaluating 108

- exempting URLs from restriction 113, 117

- introducing to users 120

- log entries for 114, 119

- precedence of ACL checking 113

- purchasing subscription 108

- requirements for 112

- restricting user access to URL categories 108

- testing blocking 119

- troubleshooting 119

- ways of blocking 112

 - selecting categories 112

 - using ACLs 112

SSL

- and HTTPS 81

- certificates 81

- defined 81

- key encryption 81

- root certificates 81

- SSL context 83

- streaming media cache

 - authentication 13

U

URL database

- DynaBlocator 96

- WebWasher 96

URL database (DynaBLocator)

- frequency of updates 97

- obtaining 96

- types of entries in 96

URLs, restricting categories of 108

user authentication 11

user database

- authenticating with NetCache 10

user groups

- changing permissions 12

V

video on demand, authentication 14

W

Web cache

- as a Web accelerator 121

WebWasher

- blocking with ACLs 102

 - prerequisites 102

 - when to use 103

- configuration overview 98

- configuring

 - basic options 100

 - URL database updates 101

- content that can be restricted 94

- description of 94

- evaluating 95

- exempting a user from restriction 103

- exempting URLs from restriction 99, 103

- introducing to users 106

- log entries for 100

- precedence of ACL checking 99

- testing blocking 105

- troubleshooting 105

- URL database 96

 - automatic updates, setting up 101

 - frequency of updates 97

 - how often to update 101

loading 101
obtaining 96
URL database contents 96

ways of blocking 98
selecting categories 98
using ACLs 98