websense

ESSENTIAL INFORMATION PROTECTION™

Transparent Identification of Users

Websense® Web Security
Websense Web Filter

v7

Transparent Identification of Users

10240 Sorrento Valley Rd., San Diego, CA 92121, USA

Published 2008

The products and/or methods of use described in this document are covered by U.S. Patent Numbers 5,983,270; 6,606,659; 6,947,985; 7,185,015; 7,194,464 and RE40,187 and other patents pending.

Every effort has been made to ensure the accuracy of this manual. However, Websense Inc., makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability and fitness for a particular purpose. Websense Inc., shall not be liable for any error or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. The information in this documentation is subject to change without notice.

Trademarks

Websense is a registered trademark of Websense, Inc., in the United States and certain international markets. Websense has numerous other unregistered trademarks in the United States and internationally. All other trademarks are the property of their respective owners.

Microsoft, Windows, Windows NT, Windows Server, and Active Directory are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Sun, Sun Java System, and all Sun Java System based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.

Mozilla and Firefox are registered trademarks of the Mozilla Foundation in the United States and/or other countries.

eDirectory and Novell Directory Services are a registered trademarks of Novell, Inc., in the U.S and other countries.

Adobe, Acrobat, and Acrobat Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Pentium is a registered trademark of Intel Corporation.

Red Hat is a registered trademark of Red Hat, Inc., in the United States and other countries. Linux is a trademark of Linus Torvalds, in the United States and other countries.

This product includes software distributed by the Apache Software Foundation (http://www.apache.org).
Copyright (c) 2000. The Apache Software Foundation. All rights reserved.

Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are the sole property of their respective manufacturers.

Contents

Chapter 1 **Introduction 1**

Transparent identification overview. 2

Websense transparent identification agents 2

Combining transparent identification agents. 2

Agent configuration settings 4

Transparent identification and remote connections 4

Using Websense Remote Filtering to identify remote users. 4

Deploying Remote Filtering 5

Chapter 2 **Websense DC Agent 7**

Computer polling 10

Components. 10

DC Agent 10

User Service 11

Filtering Service. 12

User machines 12

Files used in transparent identification with DC Agent. 13

Implementation. 14

Deploying DC Agent and related components 14

Configuring domain discovery 15

Chapter 3 **Websense Logon Agent 17**

Logon Agent user identification process 17

Components. 18

Logon Agent 18

LogonApp.exe 19

User Service 19

Filtering Service. 19

Files Used in Transparent Identification with Logon Agent 20

Implementation. 20

Deploying Logon Agent and related components. 21

Chapter 4 **Websense RADIUS Agent 23**

Processing RADIUS Traffic. 23

RADIUS Authentication 23

RADIUS Accounting 24

The RADIUS User Identification Process 24

Components. 27
RADIUS Agent 27
User Service 27
Filtering Service 27
RADIUS Client 28
RADIUS Server 28
Files 29
Implementation. 29
Deploying RADIUS Agent 30
Chapter 5 **Websense eDirectory Agent 31**
Server replication 31
eDirectory Agent user identification process 32
Components. 33
eDirectory Agent 33
Novell eDirectory. 34
User Service 34
Filtering Service 34
User Workstations 35
Files Used in Transparent Identification 35
Implementation. 36
Deploying eDirectory Agent 36
Default directory protocol 36
Enabling full queries 37
Chapter 6 **Transparent Identification Configuration Settings 39**
Websense Manager Settings. 39
Securing agent communication with Filtering Service 39
DC Agent 40
Logon Agent 41
RADIUS Agent. 41
eDirectory Agent. 42
Configuring an agent to ignore certain user names 42
Initialization parameters 43
DC Agent 44
Logon Agent 50
RADIUS Agent 51
eDirectory Agent. 53
Chapter 7 **FAQs 57**
DC Agent 57
Logon Agent. 59
RADIUS Agent 60
eDirectory Agent 62

Chapter 8

Troubleshooting 65

Troubleshooting tools. 65

Windows Services (Service Control Manager). 65

Windows Event Viewer 65

Websense log file 65

Websense ConsoleClient. 65

Websense TestLogServer 66

Websense RADIUS Agent Diagnostic Tool 67

Websense eDirectory Agent Diagnostic Tool. 68

Common Problems 69

Filtering Service cannot communicate with an agent 69

DC Agent 71

A user’s or machine’s network identity is obscured 71

DC Agent is not receiving domain controller information 71

Logon Agent 74

The user’s machine is not connected to the appropriate shared network location 74

NetBIOS for TCP/IP is disabled 74

The TCP/IP NetBIOS Helper service is not running on the client machine 75

The user profile stored on the client machine is corrupt 75

RADIUS Agent 75

VPN Usage 75

The VPN client is not successfully logged onto the VPN network 75

RADIUS Agent may impact a VPN connection 75

There is incorrect domain information in the VPN client 76

RADIUS Client/Server Configuration 76

The client cannot be filtered by the IP address assigned by RAS 76

RADIUS Agent fails to start 76

There are warnings or error messages in the Event Log 76

Websense User Identification 77

Remote users are not being filtered correctly 77

Users bypass a logon prompt to circumvent Websense filtering. 78

eDirectory Agent 79

Users are not being filtered correctly. 79

Users are not identified in a Cisco Wide Area Application Engine environment 80

Remote Filtering. 81

1 Introduction

With Websense software, you can define custom filtering policies for different individuals or groups within your organization.

- In any environment, you can assign policies to individual computers (identified by IP address), or networks (groups of computers with contiguous IP addresses).
- If your environment includes a directory service, you can configure Websense software to also filter **directory** clients: users, groups, and domains (organizational units).

To apply policies to users, groups, and other directory clients, Websense software must be able to identify the user making a request, given the originating IP address. Any of 3 methods can be used to identify users:

- An integration product can be configured to pass user information to Websense software.
- One or more Websense transparent identification agents can be used to retrieve user information.
- Websense software can prompt users for logon information when they open a browser.

This paper describes the deployment, configuration, and use of Websense transparent identification agents (second method above), including frequently asked questions and troubleshooting information.

Transparent identification overview

Transparent identification agents allow Websense software to apply user- and group-based filtering policies without prompting users for logon information.

> ✓ **Note**
> For deployments that include a transparent identification agent, Websense, Inc., recommends using anonymous authentication in your proxy server (if any). In rare cases, Basic or Integrated Windows Authentication may adversely affect access to Internet applications.
>
> If you integrate a NetCache appliance with Websense software, for transparent identification to work, NetCache must send user names to Websense software in WinNT, LDAP, or RADIUS format.

If a user cannot be identified transparently, and manual authentication is not enabled, Websense software filters based on computer or network policies, or the Default policy.

Websense transparent identification agents

There are 4 Websense transparent identification agents, which can be used alone or combined as noted in the next section.

- **Websense DC Agent:** Used with Windows-based directory services. Websense DC Agent is installed on any domain in the network on a Windows Server.
- **Websense Logon Agent:** Used with Windows-based client machines, plus either Windows Active Directory or Windows NT Directory.
- **Websense RADIUS Agent:** Can be used in conjunction with any supported directory service. Works together with a RADIUS client and RADIUS server to identify users logging on from remote locations.
- **Websense eDirectory Agent:** Designed specifically for use with Novell eDirectory. Detects users logged on to Novell eDirectory.

More information about agent files and their deployment can be found in the *Components* section for each transparent identification agent.

Combining transparent identification agents

Websense filtering software supports certain combinations of the agents within the same network or on the same machine. Generally, it is best to run one agent of a particular type on one machine. If your network configuration requires multiple agents, it is best to install them on separate machines. However, multiple agents on a single machine can work in some cases, as listed here.

Combination	Same machine?	Same network?	Configuration required
Multiple DC Agents	No	Yes	Ensure that all instances of DC Agent can communicate with Filtering Service.
Multiple RADIUS Agents	No	Yes	Configure each instance to communicate with Filtering Service.
Multiple eDirectory Agents	No	Yes	Configure each instance to communicate with Filtering Service.
Multiple Logon Agents	No	Yes	Configure each instance to communicate with Filtering Service.
DC Agent + RADIUS Agent	Yes	Yes	Install these agents in separate directories. Configure each agent to communicate with Filtering Service using a different communication port.
DC Agent + eDirectory Agent	No	No	Websense software does not support communication with both Windows and Novell directory services in the same deployment. However, you can have both agents installed, with only 1 active agent.
DC Agent + Logon Agent	Yes	Yes	Configure both agents to communicate with Filtering Service. By default, each agent uses a unique port, so port conflicts are not an issue unless these ports are changed.
eDirectory Agent + Logon Agent	No	No	Websense software does not support communication with both Windows and Novell directory services in the same deployment. However, you can have both agents installed, with only 1 active agent.
RADIUS Agent + eDirectory Agent	Yes	Yes	Configure each agent to communicate with Filtering Service using a different communication port.
DC Agent + Logon Agent + RADIUS Agent	Yes	Yes	Though this combination is rarely required, it is supported. Install each agent in a separate directory. Configure all agents to communicate with Filtering Service using different communication ports.

See your *Installation Guide* for detailed installation instructions.

Refer to the Websense Manager Help for instructions on configuring Websense transparent identification agents and implementing manual authentication.

Note

Instructions for configuring an integration product to handle user identification are provided in the *Installation Guide* and its supplements. Note that not all integration partner products can be configured to provide user names to Websense software.

Agent configuration settings

When you configure transparent identification agents in Websense Manager, the settings apply to all instances of the agent. Some settings (marked with an asterisk [“*”] in Websense Manager) are **also** stored in a separate configuration (.ini) file that is specific to each agent.

If you have multiple instances of an agent, you can use the configuration file to implement unique settings for each agent instance. Settings specified in the configuration file override the global settings in Websense Manager. For complete instructions on configuring an individual agent instance, see the *User Identification* topic in the Websense Manager Help.

Note

Websense software uses the **Character Set** option (under **Advanced Directory Settings** on the **Settings > Directory** services page) to determine how it encodes LDAP information.

Transparent identification and remote connections

Websense software can use various methods to transparently identify users logging on to your network from remote locations:

- With **Remote Filtering**, client software installed on the remote machine communicates with the Remote Filtering Server to ensure that filtering policies are applied correctly.
- If remote users log on directly to Windows domains in your network, you can use **DC Agent** to identify them.
- If your environment includes a RADIUS server, you can use **RADIUS Agent** to identify remote users.

Using Websense Remote Filtering to identify remote users

Remote Filtering allows Websense software to filter HTTP traffic for clients outside a network firewall.

When you deploy Remote Filtering Server and Remote Filtering Client, Websense software can identify any remote user logging on to a cached domain using a domain account. If a user logs on using only a local account, Websense software applies the Default policy, and Internet activity is logged under the local user name.

Deploying Remote Filtering

The Remote Filtering Server machine should be located in the DMZ, within the outermost firewall, but outside the firewall protecting the internal network. The Remote Filtering Client runs on client machines that operate outside the network firewall. The remote clients communicate with Remote Filtering Server, which acts as a proxy to Filtering Service. This communication is authenticated and encrypted.

See the *Installation Guide* for instructions on installing the Remote Filtering Server and Remote Filtering Client.

If your network is very large, you may benefit from installing Remote Filtering Server on multiple machines. This provides ample space for files that are continually populated with user information, and the user identification process is faster.

- A single Filtering Service can communicate with multiple instances of Remote Filtering Server.
- Filtering Service and Remote Filtering Server must be installed on separate machines.
- If you have installed multiple instances of Filtering Service, each instance must be able to communicate with every Remote Filtering Server.

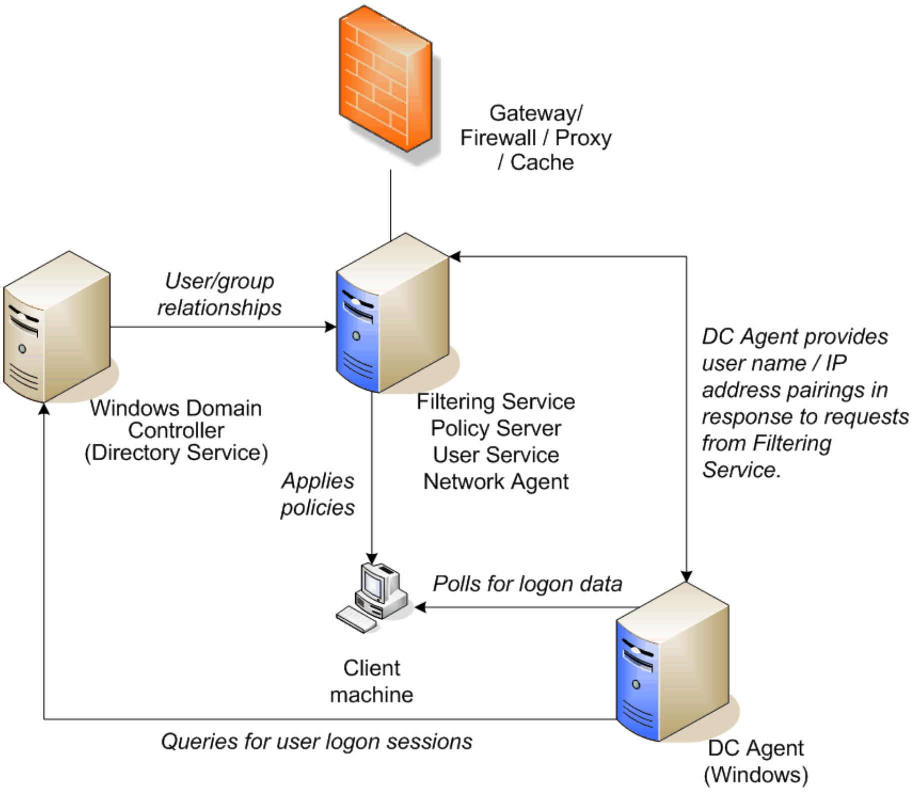
Important

Follow the recommendations in the *Deployment Guide* when setting up Remote Filtering.

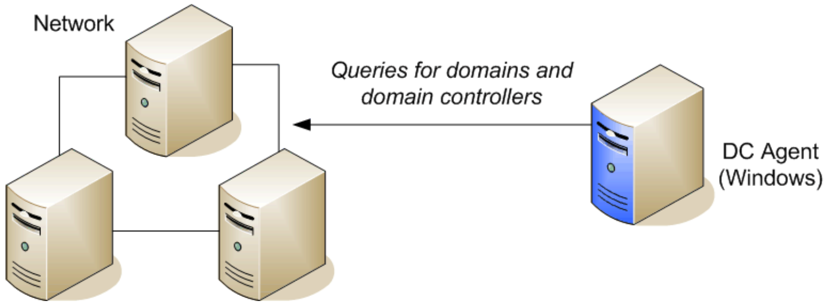
2 Websense DC Agent

Websense DC Agent makes it possible for Websense software to identify users in a Windows-based directory service without prompting them for logon information.

DC Agent and User Service gather user information and send it to Websense Filtering Service for use in applying policies. Several variables determine the speed of data transmission, including the size of your network and the amount of existing network traffic. See *Components*, page 10, for more information.



- **DC Agent detects domain controllers:** At startup, and (by default) every 24 hours thereafter, DC Agent identifies available domains and domain controllers in the network, saves the information to **dc_config.txt**, and sends the information to Filtering Service.



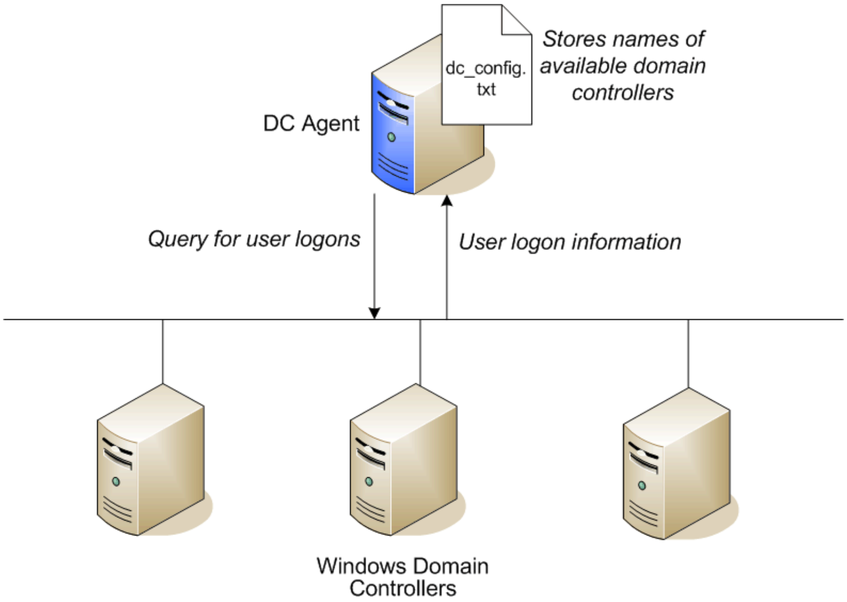
✓ **NetBIOS and Domain Discovery**

For automatic domain detection to occur, NetBIOS must be enabled on firewalls or routers connecting virtually or physically separate subnets or domains. In particular, TCP port 139 (used by NetBIOS) must be enabled. If NetBIOS is not enabled between domains or subnets, DC Agent cannot communicate with those domains or subnets. This can be true even if those domains or subnets are trusted by the domain where Filtering Service resides.

If NetBIOS port 139 is not enabled, deploy additional DC Agents in virtually or physically remote domains.

If you cannot (or prefer not to) enable port 139, you can disable NetBIOS usage. See *UseNetBIOS*, page 49.

- **DC Agent obtains logon session information:** DC Agent queries each domain controller for user logon sessions, obtaining the user and computer name.

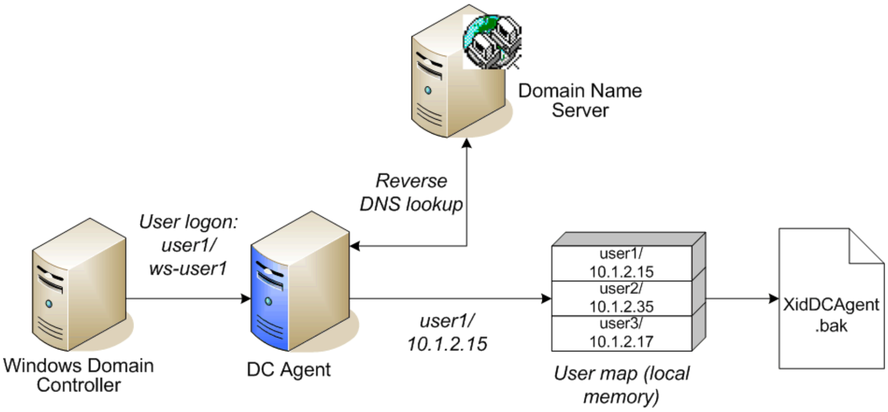

dc_config.txt
Stores names of available domain controllers
DC Agent
Query for user logons
User logon information
Windows Domain Controllers

By default, the query occurs every 10 seconds. This interval can be configured in Websense Manager (go to **Settings** > **User Identification**, and then click a DC Agent instance in the Transparent Identification Agents list).

✓ **Note**

If DC Agent is not running when a user logs on to a domain controller (because the DC Agent machine was restarted, for example), the logon session is not recorded. In this case, the user may be filtered by the computer or network policy (if it exists), or by the **Default** policy.

- **DC Agent records user name/IP address pairs:** For each logon session, DC Agent performs a DNS lookup to resolve the computer name to an IP address, and then stores the user name/IP address pair in its user map in local memory. It periodically writes a copy of the user map to **XidDcAgent.bak**.



- **DC Agent sends user information to Filtering Service:** DC Agent provides user names and IP addresses to Filtering Service each time its user map is updated.
  - The agent sends only those new user name/IP address pairs recorded since the last query.
  - Filtering Service adds new user name/IP address pairs to its copy of the user map in local memory.

  No confidential information (such as user passwords) is transmitted.

- **Filtering Service gets group information for logged-on users:** Filtering Service queries User Service to get group information for users in its copy of the user map. User Service queries the directory service for this group information, and sends the information to Filtering Service.

  User Service also provides user, group, domain, and organizational unit information from the directory service to Websense Manager when you add Directory clients.

- **Websense software applies policies to logged-on users:** Filtering Service uses the information from DC Agent and User Service to ensure that the correct policies are applied to Directory clients.

Filtering Service does not check the policy every time an Internet request is made; policy data is cached for 3 hours by the server, unless updates are saved in Websense Manager. For more information, see the Websense Manager Help.

DC Agent can be used in conjunction with Logon Agent. In this configuration, user logon information provided by Logon Agent takes precedence over information from DC Agent. DC Agent communicates a logon session to Filtering Service only in the unlikely event that Logon Agent has missed one.

Computer polling

In addition to polling domain controllers for logon information, DC Agent also polls client machines (computers) by default. This helps to verify which user is logged on to a machine.

When Filtering Service receives a request from a client machine, and the logged-on user does not appear in its user map, Filtering Service prompts DC Agent to poll the client machine.

DC Agent uses WMI (Windows Management Instruction) for computer polling. If you use computer polling, configure the Windows Firewall on client machines to allow communication on port 135.

✓ **Note**

To disable computer polling, go to the **Settings > User Identification** page in Websense Manager and click on a DC Agent instance. Under Computer Polling, clear the **Enable computer polling** check box.

Websense, Inc., recommends leaving this feature enabled.

DC Agent stores the resulting user name/IP address pair in its user map and provides the information to Filtering Service. At a pre-defined interval, DC Agent uses computer polling to verify that users are still logged on.

- The **User entry timeout** setting (1 hour, by default) determines how long an entry remains in the user map.
- The **User map verification interval** (15 minutes, by default) determines how often DC Agent attempts to verify users are still logged on.

Components

Transparent identification with Websense DC Agent involves the following Websense components and supporting configuration files.

DC Agent

Websense DC Agent monitors domain controllers and client machines for user logon information, and then provides the information to Filtering Service for use in applying filtering policies.

DC Agent can be installed on a single machine, and can discover domains outside of its own domain. Multiple DC Agent instances can also be used. This may provided some benefit in larger networks. For details, see *Deploying DC Agent and related components*, page 14.

You can configure DC Agent and Filtering Service to use an authenticated connection for communication (see *Securing agent communication with Filtering Service*, page 39).

The DC Agent executable (**XidDcAgent.exe**) is installed in the Websense **bin** directory (C:\Program Files\Websense\bin, by default). This program runs as a Windows service (Websense DC Agent), and initiates the processes that enable DC Agent to identify domains and monitor logon sessions.

DC Agent stores domain controller information in a file called **dc_config.txt**. New domain information is written to **dc_config.txt** at agent startup, and every 24 hours thereafter (by default).

The **dc_config.txt** file contains:

- names of the available domains and domain controllers in the network
- whether DC Agent monitors each domain controllers

You can configure which domain controllers are monitored by assigning each a value of **on** (monitor, the default) or **off** (do not monitor) to each entry in the file:

```
[Domain1]
DCA=on
DCB=off
[Domain2]
DC1=on
DC2=off
```

Note

If you change the setting applied to a domain controller, you must restart the Websense DC Agent service for the change to take effect.

Additional DC Agent settings can be configured in the **transid.ini** file. (A synopsis of each parameter in the **transid.ini** file is provided in *Transparent Identification Configuration Settings*, page 39.)

DC Agent writes its user-name-to-IP address map to a file called **XidDcAgent.bak**.

User Service

User Service works with DC Agent to provide an up-to-date list of domains in the network and users in each domain. User Service interacts with the directory service to get group, domain, and organizational unit information for logged-on users.

Filtering Service

Filtering Service receives user logon session information from DC Agent as users log on to the network. Filtering Service gets user data in the form of user name/IP address pairs (originating from DC Agent's map in local memory).

When Filtering Service receives the IP address of a machine making an Internet request, it matches the address with the user name provided by DC Agent. Filtering Service then uses the appropriate policy to filter the request.

Filtering Service and DC Agent can be installed on the same machine, or on different machines.

Websense software can be configured to prompt users for their logon information if DC Agent is unable to identify them (manual authentication). When manual authentication is used, a user who does not provide a valid account name and password is blocked from Internet access.

If a user cannot be identified transparently, and manual authentication is not enabled, then Websense software filters requests based on computer or network policies, or on the **Default** policy.

Note

Filtering Service, User Service, and DC Agent must be able to communicate successfully. Problems with communication between components can cause user identification errors. See *DC Agent*, page 71, for common problems and related solutions.

User machines

When DC Agent is used for transparent identification, filtered users must be able to log on to a Windows domain from their machines. Client machines do not necessarily need to be running Windows. See the *Installation Guide* for a list of supported operating systems.

The IP address of each computer is a key element in applying Websense filtering policies. If DC Agent cannot identify a machine by IP address, Internet requests made from that machine are filtered according to the **Default** policy.

For each logon session detected by a domain controller, DC Agent performs a reverse DNS lookup to convert the user's machine name to an IP address, and then stores the IP address in its user map.

You can set DC Agent to use NetBIOS to get IP Addresses if the DNS lookup fails. See the **UseNetBIOS** description in *Transparent Identification Configuration Settings*, page 39,for details.

Files used in transparent identification with DC Agent

Most of the files involved in the transparent identification process are created automatically during installation. The table below includes a brief synopsis of each file and its primary functions.

Filename	Location and Purpose	Functionality
XidDcAgent.exe	\Websense\bin\ (runs as a Windows service named **Websense DC Agent**)	Is the core of DC Agent. ◆ Automatically discovers domains at startup and at 24-hour intervals, by default. ◆ Sends new entries to Filtering Service, when queried. ◆ Allows communication of transparent identification configuration from Websense Manager to DC Agent. ◆ Uses port 30600 by default.
transid.ini	\Websense\bin\	Contains initialization parameters for **XidDcAgent.exe**. See page 40 for parameter descriptions. Changes to this file override settings configured in Websense Manager.
dc_config.txt	\Websense\bin\	Specifies which domains and domain controllers DC Agent should monitor.
XidDcAgent.bak	\Websense\bin\	Serves as a backup copy of the DC Agent user map. Read on agent startup.
ignore.txt	\Websense\bin\	Contains list of user names, machines, and user/machine pairs for DC Agent to ignore. See *Configuring an agent to ignore certain user names*, page 42.

Implementation

This section includes basic setup information, as well as options for increasing the performance of the user identification process.

Detailed instructions for configuring Websense software to communicate with DC Agent can be found in the *User Identification* section of the Websense Manager Help.

To enable transparent user identification with DC Agent:

- Install DC Agent and User Service. DC Agent can be installed with other Websense software components or alone.
- Use Websense Manager to configure Websense software to communicate with DC Agent.
- Use Websense Manager to identify directory clients to filter.

✓ **Note**

If you are using DC Agent only for Internet usage *logging*, and not for filtering, you can skip the previous step.

You can also configure Websense software to prompt users for logon information if transparent identification fails or is not available (for example, due to a network connection problem between the agent and the directory service). See the Websense Manager Help for details.

For information about securing communications between DC Agent and Filtering Service, see *Securing agent communication with Filtering Service*, page 39.

Deploying DC Agent and related components

If your network is very large (10,000+ users or 30+ domain controllers), you may benefit from installing DC Agent on multiple machines, particularly if you have different domains in separate subnets. This way, you have ample space for files that are continually populated with user information, and the user identification process is faster.

In most cases, you need only 1 Filtering Service that communicates with every instance of DC Agent in your network. If you have installed multiple Filtering Services for load-balancing purposes, each Filtering Service must be able to communicate with every DC Agent.

Typically, User Service is installed on the same machine as Policy Server. User Service can be installed separately, as long as there is 1 instance of User Service for each instance of Policy Server.

DC Agent uses Transmission Control Protocol (TCP) to transmit data. When user data is sent to Filtering Service, roughly 80 bytes is transmitted per user name/IP address pair. On average, DC Agent uses 2.8 MB RAM, but this varies based on the number of logon sessions. The table below shows average quantities of data transferred per day, by network size.

250 users	30 KB
2000 users	240 KB
10,000 users	1200 KB

To use multiple DC Agent instances for load balancing in a large network:

1. Install DC Agent on multiple servers, and configure each DC Agent instance to monitor a portion of the network.
2. Edit the **dc_config.txt** file for each DC Agent instance to query only 1 domain, or only a selected domain controllers in a domain. To do this:
   a. Navigate to the Websense bin directory on the DC Agent machine (C:\Program Files\Websense\bin, by default).
   b. Make a backup copy of **dc_config.txt** in another location.
   c. Open the original **dc_config.txt** file in a text editor and locate the name of each domain that the selected DC Agent instance monitors.
   d. Under each domain name, set any domain controller entry that the selected DC Agent should not monitor to **off**. For example, to configure a DC Agent instance to monitor only DC1 and DC2 in Domain2, edit the file as follows:

      ```
      [Domain2]
      DC1=on
      DC2=on
      DC3=off
      DC4=off
      ```

   e. Save and close the file, and then restart the Websense DC Agent service.

See *Files used in transparent identification with DC Agent*, page 13 for more information about **dc_config.txt** and other files used in the transparent identification process.

Configuring domain discovery

By default, DC Agent automatically identifies the domains in its subnet every 24 hours, adding new domains and domain controllers to the **dc_config.txt** file.

You can configure this behavior in the **transid.ini** file:

1. On the DC Agent machine, navigate to the Websense bin directory (C:\Program Files\Websense\bin, by default).
2. Make a backup copy of **transid.ini** in another location.
3. Open the original **transid.ini** file in a text editor.
4. To change the domain detection interval, modify the line:

   ```
   DiscoverInterval=86400
   ```

   - The default value is 86400 seconds (equals 24 hours).

- The minimum permitted interval is 3600 seconds (1 hour).
- To disable domain detection altogether, set **DiscoverInterval** equal to **0**.

5. Save and close the file.
6. Restart the Websense DC Agent service.

See *Transparent Identification Configuration Settings*, page 39, for more information about the **DiscoverInterval** parameter.

3 Websense Logon Agent

Using Websense Logon Agent maximizes accuracy in identifying users as they log on to the network. While DC Agent identifies users by periodically querying domain controllers and client machines, Logon Agent identifies users in real-time manner, as they log on to domains. This eliminates the possibility of missing a user logon due to a query timing issue.

Logon Agent (also called the **Authentication Server**) works with the Websense logon application (**LogonApp.exe**) to identify users as they log on to Windows domains in your network. Logon Agent then provides up-to-date logon session information to Filtering Service.

The logon application is activated via a logon script (a text file with a **.bat** or **.cmd** extension) that resides in the same directory as **LogonApp.exe**. Customize the default script installed with your Websense software to meet your needs.

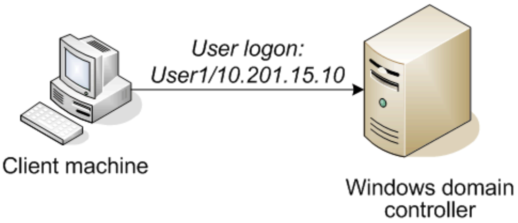
See the *Installation Guide* for information about configuring logon and logout scripts based on the directory service in your network.

Logon Agent user identification process

After the Logon Agent service is installed on a server machine, and the logon application is deployed to client machines, the 2 components work together to detect users as they log on to your network.

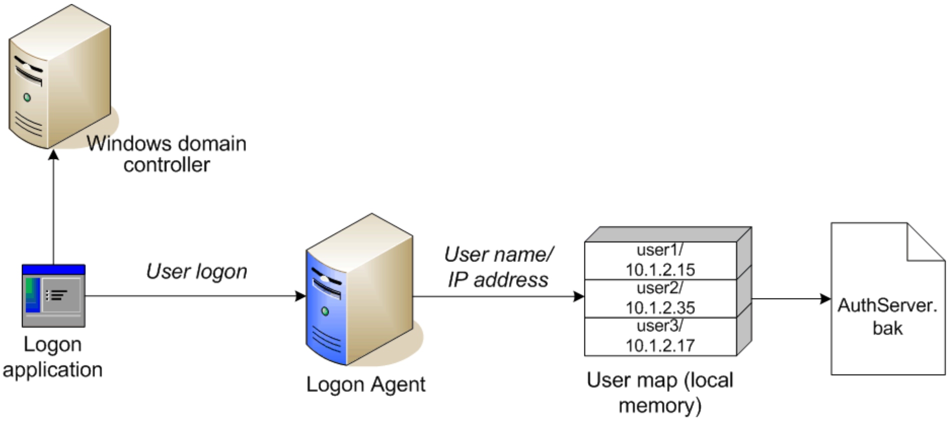
The user identification process works as follows.

1. When users log on to the network, a network logon script invokes the Websense logon application (**LogonApp.exe**).



2. The logon application contacts Logon Agent via HTTP.

3. Logon Agent sends an NTLM authentication challenge, and the logon application provides a user name, hashed password, and IP address to Logon Agent.
4. Logon Agent verifies the user name/password combination from the logon application by establishing a session with the domain controller. (Logon Agent contacts User Service to determine which domain controller is the logon source.)
5. After verifying the user name/IP address pair, Logon Agent provides the information to Filtering Service and adds an entry to its user map in local memory. The user map is periodically saved to a backup file, **AuthServer.bak**.



6. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. Filtering Service is not sent confidential information (such as user passwords).

If you Logon Agent in conjunction with DC Agent, Logon Agent takes precedence. DC Agent communicates a logon session to Filtering Service only in the unlikely event that Logon Agent has missed one.

Components

Transparent identification with Websense Logon Agent uses the following components.

Logon Agent

The Websense Logon Agent (Authentication Server) can be installed on Windows or Linux, and works with the logon application installed on the Windows client.

Logon Agent can communicate with Windows Active Directory or Windows NT Directory, and uses information sent by the logon application to authenticate user logon sessions from all Windows domains in your network. The agent stores authenticated user name/IP address pairs in a user map in local memory. The user map is periodically saved to a backup file, **AuthServer.bak**.

Multiple Logon Agent instances can be used if required; this may benefit larger networks (see *Deploying Logon Agent and related components*, page 21).

Filtering Service uses the information provided by Logon Agent to apply filtering policies to logged-on users.

LogonApp.exe

The logon application (**LogonApp.exe**) runs on Windows client machines, and sends user logon information to Logon Agent for authentication. The application sends user data either when logon sessions first occur, or at a specified interval (default), depending on the application's operation mode.

- In **persistent mode** (default), the logon application sends logon information to Logon Agent at a specific interval (configured using the **Query interval (persistent mode)** setting in Websense Manager).
- In **nonpersistent mode**, the logon application sends logon information to Logon Agent only once for each logon. The entry remains in the user map for a specific interval (configured using the **User entry expiration (nonpersistent mode)** setting in Websense Manager).

See *Transparent Identification Configuration Settings*, page 39, for information about configuring persistent and nonpersistent mode.

User Service

User Service provides domain controller names and IP addresses to Logon Agent so that the agent can authenticate users logged on to domains. User Service also interacts with your directory service to get group information for logged-on users.

Filtering Service

Filtering Service translates logon session data provided by Logon Agent so that Policy Server can apply the appropriate filtering policies to users, groups, and organizational units.

Filtering Service receives user logon session information from Logon Agent as users log on to domain controllers or machines. Filtering Service gets user data as user name/IP address pairs. When Filtering Service receives the IP address of a machine making an Internet request, it consults its user map to match the address with a user name, allowing users to be identified transparently. Filtering Service then filters users according to policies assigned to those users or groups.

Websense software can be configured to prompt users to manually authenticate if it cannot obtain user information via Logon Agent. When manual authentication is enabled, users who cannot provide a valid user name and password are blocked from Internet access.

If a user cannot be identified transparently, and manual authentication is not enabled, Websense software filters requests using computer or network policies, or the **Default** policy.

✓ **Note**

Filtering Service, User Service, and Logon Agent must be able to communicate successfully. Problems with communication between components can cause user identification errors.

Files Used in Transparent Identification with Logon Agent

Filename	Location and Purpose	Functionality
AuthServer.exe	Websense\bin\ or /opt/Websense/ Runs as the Websense Logon Agent service.	The Logon Agent executable: ◆ Sends new entries to Filtering Service. ◆ Receives configuration information from Websense Manager. ◆ Uses port 30602 by default.
LogonApp.exe	Stored in a shared network location (recommended), and activated on client machines by a logon script	Captures user logon sessions as they occur. Runs on Windows client machines.
[logonscript].bat	Resides in the same shared network location as LogonApp.exe	Invokes LogonApp.exe, which runs on client machines and captures logon sessions.
AuthServer.bak	Websense\bin\ or /opt/Websense/	Backup copy of the Logon Agent user name/IP address map. Read at startup.
AuthServer.ini	Websense\bin\ or /opt/Websense/	Contains one initialization parameter for Logon Agent.

Implementation

Logon Agent can be used with a Windows NT Directory or Windows Active Directory, and can run on Windows or Linux machines. The logon application runs only on Windows client machines.

For information about securing communications between Logon Agent and Filtering Service, see *Securing agent communication with Filtering Service*, page 39.

Deploying Logon Agent and related components

Logon Agent needs to be installed on only 1 machine in the network. However, if your network is very large (10,000+ users or 30+ domain controllers), you may benefit from installing Logon Agent on multiple machines, particularly if you have different domains in separate subnets. This way, you have ample space for files that are continually populated with user information, and the user identification process is faster.

In most cases, you need only 1 Filtering Service to communicate with every instance of Logon Agent in your network. If you have installed multiple Filtering Services for load-balancing purposes, each Filtering Service must be able to communicate with every Logon Agent.

Typically, User Service is installed on the same machine as Policy Server. User Service can be installed separately, as long as there is 1 instance of User Service for each instance of Policy Server.

4 Websense RADIUS Agent

Websense RADIUS Agent works together with the RADIUS server and RADIUS clients in your network to process and track Remote Access Dial-In User Service (RADIUS) traffic.

Websense RADIUS Agent enables Websense filtering software to transparently identify users who access your network using a dial-up, Virtual Private Network (VPN), Digital Subscriber Line (DSL), or other remote connection (depending on your configuration).

Processing RADIUS Traffic

RADIUS Agent acts as a proxy that forwards RADIUS messages between a RADIUS client and a RADIUS server (or multiple clients and servers, depending on the network configuration). RADIUS Agent does not authenticate users directly. Instead, the Agent identifies remote users authenticated by a RADIUS server and associates them with IP addresses, so Websense software can filter those users.

When properly configured, RADIUS Agent captures and processes RADIUS protocol packets of the following types:

- **Access-Request:** Sent by a RADIUS client to request authorization for a network access connection attempt.
- **Access-Accept:** Sent by a RADIUS server in response to an Access-Request message; tells the RADIUS client that the attempted connection is authorized and authenticated.
- **Access-Reject:** Sent by a RADIUS server in response to an Access-Request message; tells the RADIUS client that the attempted connection is rejected.
- **Accounting-Stop-Request:** Sent by a RADIUS client to tell the RADIUS server to stop tracking activity for a specific user.

RADIUS Authentication

Each RADIUS message packet contains attributes that describe the connection attempt, such as user name, password, and IP address of an access server. Websense RADIUS Agent stores user name-to-IP-address pairings in a user map, and provides this information to Websense Filtering Service.

RADIUS Accounting

If your RADIUS client supports accounting (user logon tracking), and accounting is enabled, RADIUS Agent is able to extract more details about user logon sessions from the RADIUS messages it receives.

For example, if there is no static IP address for an authenticated remote user, a dynamic IP address is assigned to that user. RADIUS Agent receives the dynamic IP address via an accounting request from the RADIUS client, and then records the resulting user name/IP address entry in its user map.

Stop accounting requests tell the RADIUS server to stop tracking logon activity for a particular user. The stop accounting request process is as follows:

1. RADIUS Agent receives a RADIUS stop accounting message.
2. RADIUS Agent extracts the user name and IP address from the request, and tells the RADIUS Agent service to remove the matching entry from its map.

The RADIUS User Identification Process

Websense RADIUS Agent works together with the RADIUS server and RADIUS clients in your network to process and track Remote Access Dial-In User Service (RADIUS) protocol traffic. This enables you to assign particular filtering policies to users or groups of users who access your network remotely, as well as to local users.

✓ **Note**

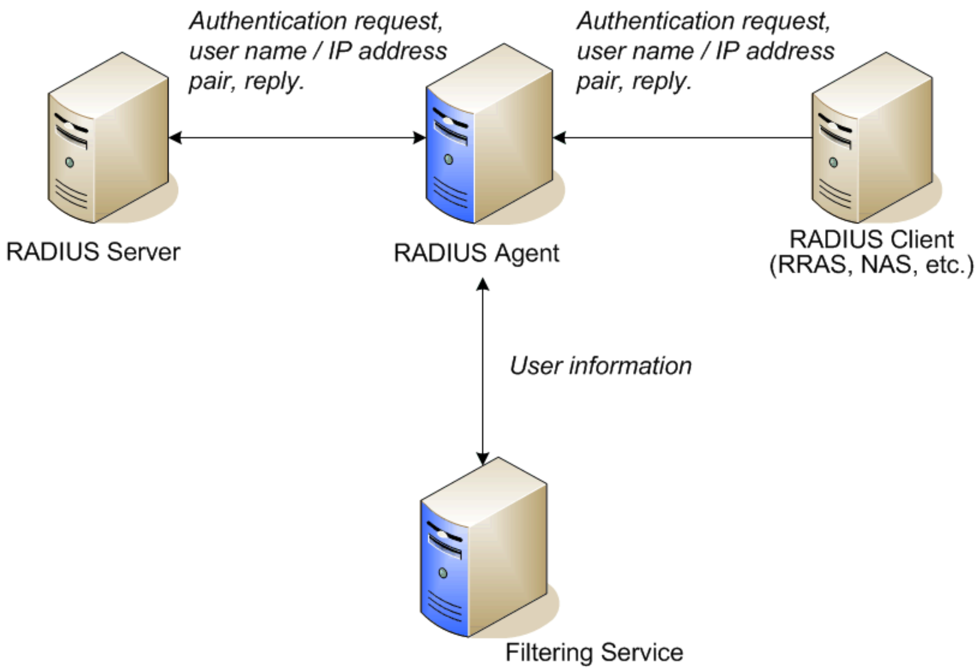
Websense, Inc., recommends installing RADIUS Agent on a machine separate from the RADIUS server machine. This prevents port and IP address conflicts between RADIUS Agent and the RADIUS server.

Without Websense RADIUS Agent, remote users are authenticated by a RADIUS client (typically, an RAS server, VPN server, or firewall).

The authentication process without RADIUS Agent is as follows:

1. A user logs on to the network from a remote machine.
2. The RADIUS client receives an authentication request for that user.
3. The RADIUS client contacts the RADIUS server via the default RADIUS ports (1645 for authentication, and 1646 for accounting), and sends the user name and password to the RADIUS server.
4. The RADIUS server validates the user name/password combination by checking it against the directory service, and then responds to the RADIUS client.

With Websense RADIUS Agent in place in your network, the user authentication process allows the agent to process and transmit remote authentication requests and provide user information to Filtering Service for use in filtering.


Authentication request, user name / IP address pair, reply.
Authentication request, user name / IP address pair, reply.
RADIUS Server
RADIUS Agent
RADIUS Client (RRAS, NAS, etc.)
User information
Filtering Service

The transparent identification process is as follows:

1. RADIUS Agent detects listens on port 1645 (the RADIUS authentication port) for authentication requests and detects users logging on to domains, or logging on to the RADIUS server directly.

✓ **Note**

If you are using RADIUS authentication in a specific Windows domain, run the Websense RADIUS Agent service as a domain user, or as the default System account on a machine in that domain.

2. When a remote user logs on to the network, the RADIUS client receives an authentication request and contacts the RADIUS Agent machine via port 1645.

3. RADIUS Agent extracts the authentication request ID (a unique identifier), user name, and originating IP address and stores the data in a user name-to-IP-address map in local memory, and in the **RadiusAgent.bak** file.

✓ **Note**

If RADIUS Agent receives a new request from an IP address already included in its user map, it **replaces** the existing pair with the new pair.

4. After extracting the required information, RADIUS Agent forwards the authentication request to the RADIUS server.
5. The RADIUS server checks the user name and password entered against the corresponding account in the directory service, and then sends a response to RADIUS Agent indicating the status of the authentication request.

✓ **Note**

To configure the amount of time RADIUS Agent waits for a response from the RADIUS server before ending a query attempt. modify the **Timeout** parameter in the RADIUS configuration file (**wsradius.ini**).

For more details, see *Transparent Identification Configuration Settings*, page 39.

6. RADIUS Agent evaluates the response from the RADIUS server. If the RADIUS message received is an authentication **rejection**, RADIUS Agent removes the corresponding entry from its user map.

   If the RADIUS packet received is an authentication **acceptance**, RADIUS Agent gets copies the corresponding entry to its main user map (a listing of full domain/user name/IP address entries).
7. RADIUS Agent forwards the authentication response to the RADIUS client.
8. RADIUS Agent sends user names and IP addresses to Filtering Service each time its user map is updated, using port 30800. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. No confidential information (such as user passwords) is transmitted.

✓ **Note**

If you configure RADIUS Agent to require authentication, the RADIUS Agent service checks the password provided by Filtering Service against the password you specified via the **User Identification** settings in Websense Manager. See *Securing agent communication with Filtering Service*, page 39, for more information.

9. Filtering Service queries User Service to get group information for user names in its copy of the user map. User Service queries the directory service for group information corresponding to those users, and sends the information to Filtering Service.

10. Filtering Service applies filtering policies to logged-on users. For more information about applying policies to directory clients, see the Websense Manager Help.

Components

Transparent identification with Websense RADIUS Agent uses the following components.

RADIUS Agent

RADIUS Agent is installed on a Windows 2003 Server or Red Hat Enterprise 3, 4, or 5 machine. It runs as a service on Windows, and as a daemon on Linux. One instance of Websense RADIUS Agent can support multiple RADIUS clients. Multiple RADIUS Agents can also be used; this may benefit larger networks. For details, see the *Deployment Guide*.

By default, RADIUS Agent listens for authentication requests on the RADIUS authentication port. Filtering Service uses the information provided by RADIUS Agent to apply filtering policies to remote users logged on to the network.

RADIUS Agent extracts the authentication request ID (a unique identifier), user name, and originating IP address. The Agent stores this data in a user name-to-IP-address map in local memory and in the **RadiusAgent.bak** file.

IP addresses, rather than user names, are the key element in tracking logon sessions, because it is possible for the same user to log on to the network via different machines or from varying locations. In cases where users share an IP address (as with Windows Terminal Services), Websense software cannot always identify particular users for filtering purposes. In this case, users are filtered by computer or network policies, or by the **Default** policy.

User Service

User Service interacts with your directory service to get group information corresponding to logged-on users. It provides this information to Filtering Service.

Filtering Service

Filtering Service receives user logon information from RADIUS Agent as users log on to the network. At each transmission, only the record of logon sessions established since the last transmission is sent back to the server. This includes new users logged on to existing remote machines and new users logged on to new remote machines.

Filtering Service receives user data in the form of user name/IP address pairs (originating from RADIUS Agent’s map in local memory). When Filtering Service gets the IP address of a machine making an internet request, the server matches the address with the corresponding user name provided by RADIUS Agent, allowing users to be identified transparently whenever they make internet requests. Filtering Service then filters users according to policies assigned to those users or groups.

Filtering Service is the destination for the user information RADIUS Agent gleans from authentication requests. When you are troubleshooting user identification problems, be sure to determine whether Filtering Service is getting the latest and most accurate user data.

Websense software can be configured to prompt users to manually authenticate if it cannot obtain user information via RADIUS Agent. With manual authentication, if a user does not provide a valid user name and password, he or she is blocked from internet access.

If a user cannot be identified transparently, and manual authentication is not enabled, then Websense software filters requests based on workstation or network policies, or on the **Default** policy, depending on your configuration.

✓ **Note**

Filtering Service and RADIUS Agent must be able to communicate successfully. Problems with communication between components can cause user identification errors. See *Common Problems*, page 69 for information.

RADIUS Client

Typically, the RADIUS client is a Network Access Service (NAS) or remote access server, which acts as the point of contact for remote user logons. The client receives authentication requests as users log on, and sends authentication requests to RADIUS Agent for processing.

The RADIUS client sends authentication requests to the port specified in Websense Manager (go to **Settings > User Identification** and click a RADIUS Agent instance to view and configure this setting).

These port values are also stored as **AuthInPort** and **AccInPort** in the RADIUS Agent **wsradius.ini** file. See *Transparent Identification Configuration Settings*, page 39, for configuration parameter descriptions.

Important

The RADIUS client and server must be configured to communicate via RADIUS Agent. Follow the configuration guidelines in the Websense Manager Help.

RADIUS Server

The RADIUS server is typically a service that performs internet authentication, such as the Microsoft Internet Authentication Service (IAS).

The RADIUS server performs the actual user authentication function. The RADIUS server receives authentication requests from Websense RADIUS Agent, and checks the user name and password entered against the corresponding account in the directory service. Finally, the RADIUS server sends a response to RADIUS Agent indicating the status of the authentication request.

Files

Most of the files involved in the remote user transparent identification process are created automatically during RADIUS Agent installation. The table below includes a brief synopsis of each file and its primary functions.

File name	Location and Purpose	Functionality
RADIUSAgent.exe	\Websense\bin\ or /opt/Websense/	The Websense RADIUS Agent executable. Automatically sends new entries to Filtering Service, when queried. Allows communication of transparent identification configuration from Websense Manager to RADIUS Agent.
wsradius.ini	\Websense\bin\ or /opt/Websense/	Contains RADIUS Agent initialization parameters. See page 41 for parameter descriptions.
RadiusAgent.bak	\Websense\bin\ or /opt/Websense/	Backup copy of RADIUS Agent's user name-to-IP address map. Read at startup.
ignore.txt (optional)	\Websense\bin\ or /opt/Websense/	Contains list of users, machines, and user/machine pairs for RADIUS Agent to ignore. See *Configuring an agent to ignore certain user names*, page 42.

Implementation

For detailed instructions on implementing transparent identification with Websense software, see *User Identification* in the Websense Manager Help. To summarize:

- Install RADIUS Agent on the desired machine. (This can be a machine running Windows 2003 Server or Red Hat Enterprise version 3, 4, or 5 [AS, ES, or WS].)
- Configure Filtering Service to communicate with RADIUS Agent. (For information about securing communication between the agent and Filtering Service, see *Securing agent communication with Filtering Service*, page 39.)
- Configure the RADIUS client to communicate with Websense RADIUS Agent instead of directly with the RADIUS server. The RADIUS client uses RADIUS Agent as the source of responses to authentication requests.
- Configure RADIUS Agent to forward authentication requests from client machines to the RADIUS server.

- Configure the RADIUS server to use Websense RADIUS Agent as a proxy.
- Use Websense Manager to add the directory clients you want to filter.

Deploying RADIUS Agent

RADIUS Agent needs to be installed on only 1 machine in the network. However, if your network is very large, you may benefit from installing RADIUS Agent on multiple machines. This way, you have ample space for files that are continually populated with user information, and the user identification process is faster.

In most cases, you need only 1 Filtering Service that communicates with every instance of RADIUS Agent in your network. If you have installed multiple Filtering Services for load-balancing purposes, each Filtering Service must be able to communicate with every RADIUS Agent.

5 Websense eDirectory Agent

eDirectory Agent works with Novell eDirectory to transparently identify users so that Websense software can filter them according to policies assigned to particular users or groups. eDirectory Agent does not authenticate users directly. Instead, the agent uses Netware Core Protocol (NCP) to gather user logon session information from Novell eDirectory, which authenticates users logging on to the network. (The query protocol can be changed; see *Default directory protocol*, page 36.)

Websense eDirectory Agent associates each authenticated user with an IP address and records user name-to-IP-address pairings to a user map. eDirectory Agent supplies this information to Websense Filtering Service.

- **User name:** The name by which the user is identified and authenticated in the network. eDirectory Agent correlates the Novell eDirectory Common Name (*cn*) attribute to a user logging in. The *cn* acts as a unique identifier of an object within the Novell eDirectory structure.
- **IP address:** The IP address of a logged-on user. eDirectory correlates the Novell attribute *networkAddress* with the user. It is possible for each user to have zero, 1, or more attributes with this name. For each successful logon, Novell eDirectory server adds 1 *networkAddress* entry to a user's attribute profile. If the *networkAddress* attribute is not present for a user, it means the user is not logged on to Novell eDirectory. Websense eDirectory Agent scans all the *networkAddress* attributes of a user and adds corresponding user name/IP address entries to its user map.

Server replication

Novell eDirectory server can be configured to support several replicas of the directory service on separate machines. The replicas are synchronized copies of the directory that are stored in different locations on the network. Replication increases the availability, robustness, and fail-safety of Novell eDirectory.

There are two schemes by which Novell server performs replication between machines running eDirectory server replicas: fast and slow. Fast replication occurs every 10 seconds, and slow replication every five minutes. When a user logs on to a particular eDirectory replica, the data for this user is first updated on the machine running this replica. It takes time for user logon data to propagate to all replicas.

Websense eDirectory Agent uses the *networkAddress* property of a user object to associate IP addresses with logged-on users. Because the *networkAddress* property is synchronized during the slow replication process, there is potentially a five-minute gap between the logon event and the update of user data on all machines containing replicas.

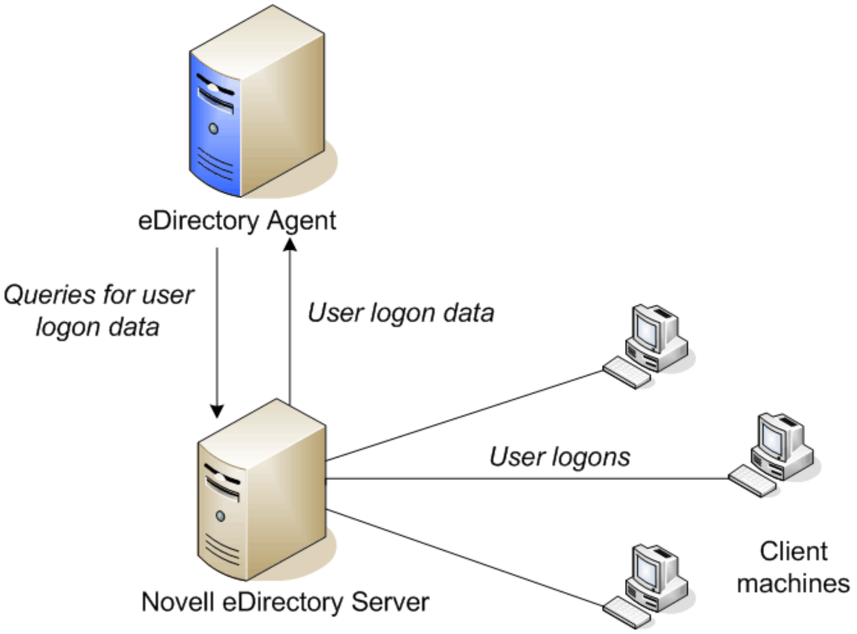
eDirectory Agent must be configured to connect to each machine running an Novell eDirectory replica. See the Websense Manager Help for details on configuring this communication.

eDirectory Agent user identification process

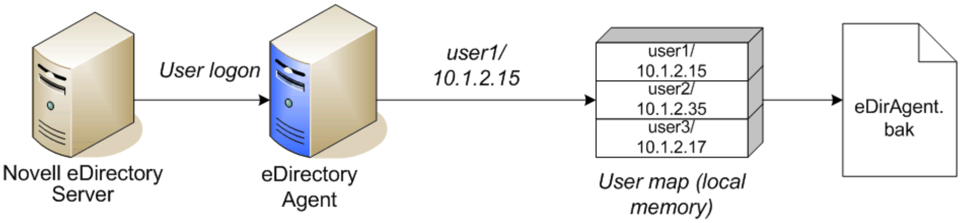
The transparent identification process with eDirectory Agent is as follows.

1. Novell eDirectory authenticates users as they log on.
2. eDirectory Agent retrieves information from Novell eDirectory about logged-on users. The agent queries the directory service or user logons at regular intervals (30,000 milliseconds, or 30 seconds, by default).

   The agent detects only users logging on directly to Novell eDirectory server.



3. eDirectory Agent stores the user name, domain name, and originating IP address from each logon session in a user name-to-IP-address map in local memory, and in the **eDirAgent.bak** file.



✓ **Note**

If eDirectory Agent receives a new request from an IP address already included in its map, it **replaces** the existing pairing with the new pair.

4. eDirectory Agent sends user names and IP addresses to Filtering Service using port 30700. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. No confidential information (such as user passwords) is transmitted.

✓ **Note**

You can configure eDirectory Agent to require authentication from Filtering Service (see *Securing agent communication with Filtering Service*, page 39).

5. Filtering Service queries User Service for group information for user names in its user map. User Service queries Novell eDirectory for group information corresponding to those users, and sends the information to Filtering Service.
6. Filtering Service applies policies to the logged-on users. For more information about applying policies to directory clients, see the Websense Manager Help.

Components

These descriptions clarify the role of each piece involved in transparent identification with Websense eDirectory Agent.

eDirectory Agent

Websense eDirectory Agent queries Novell eDirectory for user logon session information at a given interval. eDirectory Agent associates each authenticated user with an IP address, and records user name-to-IP-address pairings to a local user map. This user map is also written to a backup file named **eDirAgent.bak**.

eDirectory Agent supplies this information to Websense Filtering Service for use in filtering internet requests.

Novell eDirectory

Novell eDirectory houses your organization’s user accounts, and provides user authentication to Websense via Websense eDirectory Agent.

One instance of Websense eDirectory Agent can support 1 Novell eDirectory master, plus any number of Novell eDirectory replicas. eDirectory Agent must be able to communicate with each machine running a replica of the directory service. This ensures that the Agent gets the latest logon information as quickly as possible, and does not need to wait for eDirectory replication to occur. See the *User Identification* topic in the Websense Manager Help for more information on deployment.

User Service

Filtering Service queries User Service to get group information for user names in its copy of the user map. User Service queries Novell eDirectory for group information corresponding to those users, and sends the information to Filtering Service. Directory objects (users and groups) are then made available to Websense Manager, which allows configuration of filtering policies based on those users and groups.

Filtering Service

Filtering Service receives user logon information from eDirectory Agent as users log on to the network. At each transmission, only the record of logon sessions established since the last transmission is sent back to the server. This includes new users logged on to existing machines and new users logged on to new machines.

Filtering Service receives user data in the form of user name/IP address pairs (originating from eDirectory Agent’s map in local memory). When Filtering Service gets the IP address of a machine making an internet request, it matches the address with the corresponding user name provided by eDirectory Agent, allowing users to be identified transparently whenever they make internet requests. Filtering Service then filters users according to policies assigned to those users or groups.

The interaction between eDirectory Agent and Filtering Service is of central importance. Filtering Service is the destination for information about users authenticated by Novell eDirectory. When you are troubleshooting user identification problems, be sure to determine whether Filtering Service is getting the latest and most accurate user data.

Websense software can be configured to prompt users to manually authenticate if it cannot obtain user information via eDirectory Agent. With manual authentication, if a user does not provide a valid user name and password, he or she is blocked from Internet access.

If a user cannot be identified transparently, and manual authentication is not enabled, Websense software filters requests based on workstation or network policies, or on the **Default** policy, depending on your configuration settings.

✓ **Note**

Filtering Service and eDirectory Agent must be able to communicate successfully. Problems with communication between components can cause user identification errors. See page 79 for information about common problems and related solutions.

User Workstations

User workstations act as network entry points for users and reflect Websense filtering policies back to users. The IP address is a key element in transparent user identification. If eDirectory Agent cannot identify a remote workstation by its IP address, internet requests made from that workstation are filtered according to the **Default** policy.

Files Used in Transparent Identification

All but 1 of the files involved in the transparent identification process are created automatically during eDirectory Agent installation. The table below includes a brief synopsis of each file and its primary functions.

File name	Location and Purpose	Functionality
eDirectoryAgent.exe	\Websense\bin\ or /opt/Websense/ (runs as a service or daemon named **Websense eDirectory Agent**)	The eDirectory Agent executable. Collects user logon information from Novell eDirectory Server. Sends user logon data to Filtering Service.
wsedir.ini	\Websense\bin\ or opt/Websense/	Contains eDirectory Agent initialization parameters. See page *eDirectory Agent*, page 42, for parameter descriptions.
eDirAgent.bak	\Websense\bin\ or /opt/Websense/	Backup copy of eDirectory Agent's user name-to-IP address map. Read at startup.
ignore.txt (optional)	\Websense\bin\ /opt/Websense	Contains list of user names, machines, and user/machine pairs for eDirectory Agent to ignore. See *Configuring an agent to ignore certain user names*, page 42.

Implementation

In addition to standard setup, options for increasing the security and performance of the user identification process are described here.

For instructions to secure communication between the agent and Filtering Service, see *Securing agent communication with Filtering Service*, page 39.

Deploying eDirectory Agent

eDirectory Agent needs to be installed on only 1 machine in the network. However, if your network is very large, you may benefit from installing the agent on multiple machines. This way, you have ample space for files that are continually populated with user information, and the user identification process is faster.

In most cases, you need only 1 Filtering Service that communicates with every instance of eDirectory Agent. If you have installed multiple Filtering Services for load-balancing purposes, each Filtering Service must be able to communicate with every eDirectory Agent.

✓ **Note**

eDirectory Agent can **not** be used in combination with DC Agent.

Default directory protocol

Websense eDirectory Agent can use Netware Core Protocol (NCP)—the Windows default—or Lightweight Directory Access Protocol (LDAP)—required on Linux—to retrieve user logon information from Novell eDirectory.

In Windows environments, NCP generally provides a more efficient query method. If your network supports LDAP, however, you can configure eDirectory Agent to use LDAP:

1. Ensure that you have at least 1 Novell eDirectory replica containing all directory objects you want to monitor and filter in your network.
2. Stop the Websense eDirectory Agent service.
3. Go to the eDirectory Agent directory (C:\Program Files\Websense\bin, by default), and locate the **wsedir.ini** file.
4. Open the file in a text editor.
5. Modify this **QueryMethod** entry as follows:

   ```
   QueryMethod=0
   ```

   Here, **0** enables LDAP queries. (1, the default, enables **NCP** queries.)
6. Save and close the file.
7. Restart the Websense eDirectory Agent service.

eDirectory Agent now uses LDAP to query the directory service.

Enabling full queries

In small networks, you can configure Websense eDirectory Agent to query the eDirectory Server for all logged-on users at regular intervals. This allows the agent to detect both newly logged-on users and users who have logged off since the last query, and to update its local user map accordingly.

Important

Configuring eDirectory Agent to use full queries is not recommended for larger networks, because the length of time required to return query results depends on the number of logged on users. The more logged-on users there are, the higher the performance impact.

When you enable full queries for eDirectory Agent, the **User entry timeout** interval is not used, because users who have logged off are identified by the query. By default, the query is performed every 30 seconds.

Enabling this feature increases eDirectory Agent processing time in 2 ways:

- Time needed to retrieve the names of logged-on users each time a query is performed
- Time required to process user name information, remove obsolete entries from the local user map, and add new entries based on the most recent query

eDirectory Agent examines the entire local user map after each query, rather than identifying only new logons. The time required for this process depends on the number of users returned by each query. The query process can therefore affect both eDirectory Agent and Novell eDirectory Server response times.

To enable full queries:

1. On the eDirectory Agent machine, navigate to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).
2. Locate the file **wsedir.ini** and make a backup copy in another directory.
3. Open **wsedir.ini** in a text editor.
4. Locate the following entry:

   ```
   QueryMethod=<N>
   ```

   Make a note of the *QueryMethod* value, in case you want to revert to the default setting later.
5. Update the **QueryMethod** value as follows:
   - If the current value is **0** (communicate with the directory via LDAP), change the value to **2**.
   - If the current value is **1** (communicate with the directory via NCP), change the value to **3**.

Note

If changing this query value slows system performance, return the QueryMethod entry to its previous value.

6. If the default query interval (30 seconds) is not appropriate for your environment, edit the *PollInterval* value appropriately.

   Note that the interval time is set in milliseconds.

7. Save and close the file.
8. Restart the Websense eDirectory Agent service.

6 Transparent Identification Configuration Settings

When your Websense software deployment includes a transparent identification agent, initial agent configuration is performed in Websense Manager via the **Settings > User Identification** page (see *Websense Manager Settings*, page 39).

If your deployment includes multiple agent instances, or if your environment requires additional configuration options not included in Websense Manager, you can edit each agent's initialization (INI) file to modify its behavior (see *Initialization parameters*, page 43).

Websense Manager Settings

Transparent identification agent configuration settings related to agent processes are defined here. For complete configuration instructions for each agent type, see *User Identification* in the Websense Manager Help.

To access these settings in Websense Manager, select the **Settings** tab of the left navigation pane, and then click **User Identification**.

Securing agent communication with Filtering Service

You can configure any transparent identification agent to use an authenticated connection for communication with Filtering Service. When authenticated communication is enabled, the agent requires a password from Filtering Service before sending data.

Use Websense Manager to enable authenticated communication:

1. Go to the **Settings > User Identification** page. Installed agents appear in the **Transparent Identification Agents** list.
2. Click an agent instance (identified by machine name or IP address) in the list.
3. Under Basic Agent Configuration, mark **Enable authentication**.
4. Specify a **Password** for the authenticated connection. The password is case-sensitive.
5. Click **OK** to return to the User Identification page.
6. Click **OK** again to cache your changes. Changes are not implemented until you click **Save All**.

See the Websense Manager Help for more information about configuring authenticated connections.

DC Agent

Refer to the Websense Manager Help for detailed configuration instructions.

Domain Controller Polling

The **Enable domain controller polling** check box determines whether DC Agent queries domain controllers for user logon information.

- Domain controller polling is enabled by default.
- Disabling the process is strongly discouraged.

When domain controller polling is enabled, DC Agent queries the domain controllers specified in its **dc_config.txt** file.

To configure domain controller polling behavior:

- Specify a **Query interval** (10 seconds, by default) to determine how often DC Agent queries domain controllers.

  Decreasing the query interval may enhance accuracy in capturing logon sessions, but also increases overall network traffic. Increasing the query interval decreases network traffic but may also delay or prevent the capture of some logon sessions.
- Specify a **User entry timeout** value (24 hours, by default) to determine how long entries resulting from domain controller polling remain in DC Agent's user map.

Domain controller polling uses both DNS lookup and NetBIOS communications. If for security reasons you do not want to allow NetBIOS communications, there are 2 options:

- Disable domain controller polling (not recommended).
- Use the **UseNetBIOS** parameter to disable DC Agent NetBIOS communications, but keep domain controller polling active. (See *UseNetBIOS*.)

Computer Polling

The **Enable computer polling** check box determines whether DC Agent contacts client machines to verify which users are logged on.

- Computer polling is enabled by default.
- Disabling this process is not recommended.

To configure computer polling behavior:

- Specify a **User map verification interval** (15 minutes, by default) to determine how often DC Agent contacts client machines to verify which users are logged on.

  Websense, Inc., recommends using the default value. Decreasing this interval may provide greater user map accuracy, but increases network traffic. Increasing the interval decreases network traffic, but also may decrease user map accuracy.

DC Agent compares the query results with the user name/IP address pairs in the user map it sends to Filtering Service.

- Specify a **User entry timeout** value (1 hour, by default) to determine how often DC Agent refreshes entries obtained through computer polling in its user map.

  Any user map entries that DC Agent cannot match to currently logged-on users are removed after this timeout period.

  Websense, Inc., recommends using the default query value. Increasing this interval may lessen user map accuracy, because the map would potentially retain old user names for a longer time.

  Do not decrease this interval to less than the **User map verification interval** value.

Logon Agent

Refer to the Websense Manager Help for detailed configuration instructions.

User Logon Information

Specify a **Query interval (persistent mode)** value (15 minutes, by default) to determine how often the logon application on the client machine (**LogonApp.exe**) sends logon information to Logon Agent.

- In persistent mode (the default), the logon application sends logon information to Logon Agent periodically, at the interval set here. Decreasing this query interval may provide greater accuracy in identifying and filtering users, but also increases overall network traffic.
- Note that when you change this value, the change does not take effect until the previous interval period has elapsed. For example, if you change the interval from 15 minutes to 5 minutes, the current 15-minute interval must end before the query starts occurring every 5 minutes.

Specify a **User entry expiration (nonpersistent mode)** value (24 hours, by default), to determine how long a user entry (user name/IP address pair) remains in the Logon Agent’s user map.

This entry applies only when the logon application is not running in persistent mode. In nonpersistent mode, logon information is sent to the Logon Agent only once for each logon.

RADIUS Agent

Refer to the Websense Manager Help for detailed configuration instructions.

Authentication Ports

Specify which port is used for communication **Between RADIUS Agent and RADIUS server** (1645, by default).

Also specify which port is used for communication **Between RADIUS clients and RADIUS Agent** (12345, by default).

Accounting Ports

If RADIUS accounting is enabled in your environment:

- Specify which port is used for communication **Between RADIUS Agent and RADIUS server** (1646, by default).
- Also specify which port is used for communication **Between RADIUS clients and RADIUS Agent** (12346, by default).

eDirectory Agent

Refer to the Websense Manager Help for detailed configuration instructions.

eDirectory Server

Specify a **User entry timeout** value (24 hours, by default), to determine how often eDirectory Agent refreshes its user map.

Websense, Inc., recommends using the default timeout value.

Configuring an agent to ignore certain user names

The method that some Windows services use to contact domain controllers from user machines can cause the users logged on to those machines to be misidentified. For example, problems can be caused by:

- The internal user names (Local Service and Network Service) that Windows XP assigns for processes to use for communication with domain controllers.
- Running Systems Management Server (SMS) on a client machine.
- Windows 2000 services may contact the domain controller with a user name consisting of the machine name followed by a dollar sign (host$).

  (To address this last issue, by default, Websense software is configured to ignore user names containing dollar signs.)

In each of these situations, when **domainA/user1** logs on to the network, Websense software enforces the policy assigned to **user1**. Then a service on the user's machine connects to the domain controller with a name like **domainA/ServiceName**. The transparent identification agent interprets **domainA/ServiceName** as a new logon session, separate from the session established by **user1**. Because there is no specific policy assigned to the user **ServiceName**, Websense begins filtering this user according to the computer or network policy, or the **Default** policy.

To prevent or work around possible misidentification, configure your transparent identification agent to ignore logon names that are not associated with actual users.

1. Use the Windows Services dialog box to stop the agent service (Websense DC Agent, Websense Logon Agent, Websense RADIUS Agent, or Websense eDirectory Agent).
2. Navigate to the Websense **bin** directory (C:\Program Files\Websense\bin, by default).
3. Use a text editor to either create or open **ignore.txt**.

✓ **Note**

To set a size limit for the ignore list, use the **MaxIgnoreListSize** initialization parameter. See *Transparent Identification Configuration Settings*, page 39.

4. Populate the file as follows. Place each entry on a separate line.
   - Add each **user name** that should be ignored on its own line. Websense software ignores these users, regardless of which machine they use.
   - To add a **user name/machine pair**, enter the user name, followed by a comma, and then the machine host name or IP address (ypark,YPARK-WS1). In this case, Websense software ignores the specified user only on the specified machine.
   - To add a **machine**, enter an asterisk (*), followed by a comma, followed by the machine host name, IP address, or IP address range.

   The following example shows correctly formatted entries:

   ```
   johnsmith
   admin,WKSTA-NAME
   *, WKSTB-NAME
   *, 10.209.34.56
   *, 10.203.34.1-10.203.34.255
   ```

   In this example, the user name **johnsmith** is ignored on all machines, the user name **admin** is ignored only when associated with machine **WKSTA-NAME**, and the machines **WKSTB-NAME** and **10.209.34.56** are ignored for all logons. In addition, logons on all machines in the network range **10.203.34.1** to **10.203.34.255** are ignored.
5. When you are finished making changes, save and close the file.
6. Restart the transparent identification agent service.

The agent ignores the specified user names and machines.

Initialization parameters

This section describes configurable initialization parameters for each transparent identification agent.

Before making changes to the initialization files, please consider that the default values are designed to maximize accuracy and efficiency in most environments. In most cases, Websense, Inc., recommends leaving the default values as they are.

DC Agent

After configuring DC Agent behavior in Websense Manager, you can customize the behavior of a specific DC Agent instance in **transid.ini**, the agent's initialization file. This file resides in the Websense **bin** directory (C:\Program Files\Websense\bin, by default).

- Some DC Agent settings can only be configured via Websense Manager.
- Some settings can only be configured via the initialization file.

Settings that can be configured either via Websense Manager or the initialization file are marked by an asterisk (*) in this document and in Websense Manager.

Most of these parameters do not appear in the **transid.ini** file by default. The only parameter that must have a value in the file is **port** (set by default to 30600).

All parameters and values described here are case-sensitive.

AllDollarSign

Prompts DC Agent to ignore logon sessions from any user names that contain a dollar sign character ($).

Default	True
Options	True, False
Required	No
Synopsis	Ensures that DC Agent drops all entries containing dollar signs from its user map, without performing any additional verification. This option is a more powerful version of *IgnoreDollarSign*.

DiagServerPort*

The port on which the Websense ConsoleClient listens for data from DC Agent. (Equivalent to **Diagnostic port** in Websense Manager.)

Default	30601
Options	Integers between 1024 and 65535
Required	No
Synopsis	Avoid changing the value of this parameter unless prompted to do so by Websense Technical Support.

DiscoverInterval

Interval at which the domain auto-discovery process runs, in seconds. The default is 86400 seconds, or 24 hours.

Default	86400
Options	Integer greater than 3600, or 0 to disable
Required	No
Synopsis	DC Agent automatically detects new domains or domain controllers added to the network. By default, detected domain names are recorded to the **dc_config.txt** file at startup, and every 24 hours thereafter. Increasing the domain discovery interval may delay discovery of a new domain or domain controller. Decreasing the interval increases network traffic, because the process runs more frequently.

IgnoreDollarSign

Enables DC Agent to ignore logons from user names containing dollar signs ($).

Default	True
Options	True, False
Required	No
Synopsis	Used to prevent a problem involving Windows 2000 services that use a machine name followed by a dollar sign (**wkstn$**) as a user name when contacting the domain controller. DC Agent interprets the service as a new user to whom no policy has been assigned. When this parameter is set to **True**, if DC Agent detects a **user$** entry in its map, it compares the user name without the dollar sign to the source machine's name. If these match, DC Agent ignores the logon session entirely, because it knows the logon did not originate from an actual user. If the user name and machine name do not match, DC Agent attempts to get the name of the actual user logged on from the source machine. If it obtains a user name, DC Agent pairs that with the IP address of the source machine, and records these together in its map. If DC Agent cannot obtain an actual user name, it simply records the **user$** entry in its map. This process minimizes the number of false user names DC Agent stores in its map and sends to Filtering Service. When the parameter is set to **False**, if DC Agent detects a **user$** entry in its map, the agent attempts to replace it with an actual user name from the source machine. If DC Agent does not obtain an actual user name, it records the **user$** entry in its map.

IgnoreLocalLogins

Determines whether DC Agent registers local (non-domain) user logons to local client machines.

Default	False
Options	True, False
Required	No
Synopsis	By default, DC Agent detects users logging on to domains and to local machines. If for some reason you want DC Agent to register logons only to domain controllers, and ignore local logons, set this value to True. See also *AllDollarSign*.

IgnoreRepeats

Determines whether DC Agent re-records user logon sessions that it already recorded at the time of the previous query.

Default	True
Options	True, False
Required	No
Synopsis	By default, DC Agent ignores a user logon to a domain controller, if it already registered that logon after the previous domain controller query. Websense, Inc., recommends leaving this default setting as is. In most cases, there is no benefit to duplicating recognition of an earlier logon session.

IPCleanInterval

Interval at which DC Agent checks its cache for stale machine name/IP address pairs, in seconds.

Default	600 [seconds = 10 minutes]
Options	Between 300 and 3600 seconds.
Required	No
Synopsis	Determines how often DC Agent checks the machine name/IP address pairs in its cache for entries older than the *IPCleanLifetime* period. Entries older than this time period are removed from the cache. This parameter typically does not need to be changed.

IPCleanLifetime

The amount of time a machine name/IP address pair remains in DC Agent's cache before it is removed, in seconds.

Default	7200 [seconds = 2 hours]
Options	Integer greater than 3600, or 0 to disable
Required	No
Synopsis	As DC Agent receives logon session information, it stores machine name/IP address pairs in its local memory cache. This reduces the number of times DC Agent must perform DNS lookups for each active client machine, because it already has the IP address.

MaxIgnoreListSize

The maximum number of entries (user names, user name/machine name pairs, and machine names) in DC Agent's **ignore.txt** file.

Default	70000
Options	Integer 5000 or greater
Required	No
Synopsis	If you use an **ignore.txt** file to configure DC Agent to ignore particular users or client machines, this parameter sets an upper limit on the number of entries in the file. See *Configuring an agent to ignore certain user names*, page 42, for details.

password*

The password DC Agent uses to authenticate connections from other Websense components. (Equivalent to **Password** in Websense Manager.)

Default	N/A
Options	Strings between 4 and 16 characters in length
Required	No
Synopsis	Allows you to specify a password for authenticated connections between DC Agent and other Websense services. The password is case-sensitive. See *Securing agent communication with Filtering Service*, page 39, for information about configuring authenticated DC Agent connections.

port*

The port over which Filtering Service connects to DC Agent. (Equivalent to **TCP port** in Websense Manager.)

Default	30600
Options	Integers between 1024 and 65535
Required	Yes
Synopsis	The port value is originally set during DC Agent installation and written to the **transid.ini** file.

QueryInterval*

The interval at which DC Agent queries domain controllers, in seconds. (Equivalent to **Query Interval** in Websense Manager.)

Default	10 [seconds]
Options	Between 5 and 90 seconds
Required	No
Synopsis	Determines how often DC Agent queries the domain controllers specified in its **dc_config.txt** file. ◆ Decreasing the query interval (to less than 10 seconds) may enhance accuracy in capturing logon sessions, but increases network traffic. Greater accuracy may be needed especially with Windows XP logon sessions, which are typically shorter than 15 seconds. ◆ Increasing the query interval decreases network traffic, but may also delay or prevent the capture of some logon sessions. Use extreme caution when modifying this parameter; an incorrect value can overload network traffic.

StartDelay

Time period by which to delay DC Agent service initialization to allow diagnostic routines to start first.

Default	0 [seconds]
Options	Between 0 and 120 seconds
Required	No
Synopsis	Used primarily by Websense Technical Support. Allows the ConsoleClient diagnostic tool to connect to DC Agent while the service is running, but before its processes are activated. Use extreme caution when modifying this parameter.

UseFileTrace

Whether to enable diagnostic file tracing for DC Agent.

Default	False
Options	True, False
Required	No
Synopsis	When this parameter is set to **True**, DC Agent writes diagnostic information to the **xid_trace.txt** file, in the Websense **bin** directory. This parameter must be enabled for the *VerifyTracing* parameter to have any effect.

UseNetBIOS

Whether to use NetBIOS to perform domain controller machine name lookups.

Default	True
Options	True, False
Required	No
Synopsis	By default, DC Agent first uses DNS lookup to identify domain controllers by name and IP address. If this fails, DC Agent uses NetBIOS calls to identify domain controllers. Set this parameter to **False** to cause DC Agent to rely solely on DNS, and not use NetBIOS at all.

UseUserService

Whether to use Websense User Service or Windows networking calls to communicate with domain controllers.

Default	True
Options	True, False
Required	No
Synopsis	By default, DC Agent uses User Service for communications with domain controllers in the network. To close the ports required for User Service to facilitate communications between DC Agent and domain controllers, set this value to **False**. In this case, DC Agent uses Windows networking calls for communications instead.

VerifyTracing

Whether to enable diagnostic tracing of computer polling routines.

Default	False
Options	True, False
Required	No
Synopsis	When this parameter is enabled, DC Agent writes diagnostic information about its computer polling processes to the **xid_trace.txt** file, in the Websense **bin** directory. The *UseFileTrace* parameter also must be set to True.

VerifyUserDomain

Whether to make sure that a user exists in a particular domain as indicated by domain controller polling results.

Default	True
Options	True, False
Required	No
Synopsis	When this parameter is enabled, DC Agent checks the existence of a user account against the domain where a user logon session is detected. When this parameter is set to **False**, DC Agent may not update its user map right away if a user account is moved from one domain to another.

Logon Agent

The Logon Agent initialization file (**AuthServer.ini**) is optional, and may contain only one parameter. Other Logon Agent configuration is performed exclusively in Websense Manager.

UserServerWaitTime

This parameter ensures that Websense User Service is running before Logon Agent starts.

Default	1 [second]
Options	0 or greater
Required	No
Synopsis	Logon Agent cannot communicate data to Filtering Service if User Service is not running. When this parameter is set to 0, Logon Agent starts even if User Service is down.

RADIUS Agent

After configuring RADIUS Agent behavior in Websense Manager, you can customize the behavior of a specific RADIUS Agent instance in **wsradius.ini**, the agent's initialization file. This file resides in the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).

- Some RADIUS Agent settings can only be configured via Websense Manager.
- Some settings can only be configured via the initialization file.

Some parameters can be modified either via Websense Manager or via **wsradius.ini**; these parameters are marked with an asterisk (*).

The parameters and values described here are case-sensitive.

AccInPort*

Port over which RADIUS Agent accepts accounting requests from RADIUS clients.

Default	12346
Options	1024 through 65535
Required	No
Synopsis	If your RADIUS environment is configured to support RADIUS accounting (user tracking), RADIUS Agent receives accounting requests from client machines over this port.

AccOutPort*

Port over which the RADIUS server listens for RADIUS accounting messages.

Default	1646
Options	1024 through 65535
Required	No
Synopsis	If your RADIUS environment supports RADIUS accounting, the RADIUS server receives accounting messages from client machines over this port.

AuthInPort*

Port over which RADIUS Agent accepts authentication requests from RADIUS clients.

Default	12345
Options	1024 through 65535
Required	No
Synopsis	Used to configure the port on which RADIUS Agent receives authentication requests from the RADIUS client as users log on to the network.

AuthOutPort*

Port on which the RADIUS server listens for authentication requests.

Default	1645
Options	1024 through 65535
Required	No
Synopsis	RADIUS Agent processes the authentication requests it receives from the RADIUS client, and then forwards them to the RADIUS server over this port.

DebugLevel

Determines the detail level of the RADIUS Agent diagnostic activity. (See definition for *DebugMode*.)

Default	0
Options	0, 1, 2, 3
Required	No
Synopsis	Specifies the level of log file detail provided for debugging purposes, from none (0) to high (3). Any value outside the range of 0-3 is interpreted as 0. Diagnostic output with a detail level of 3 includes all RADIUS transactions involved in a user logon.

DebugMode

Controls the RADIUS Agent diagnostic activity.

Default	Off
Options	On, Off
Required	No
Synopsis	Enables or disables RADIUS Agent’s built-in diagnostic (logging and debugging) capabilities.

LogFile

Output file for RADIUS Agent diagnostic messages.

Default	N/A
Options	Any string of characters valid for your operating system
Required	No
Synopsis	If you have enabled *DebugMode*, specify a name for the text file in which RADIUS Agent stores diagnostic (log) output.

RADIUSHost*

IP address of the RADIUS server machine.

Default	None
Options	Valid IP address in the format 123.123.123.123
Required	Yes
Synopsis	RADIUS Agent forwards authentication and accounting requests to the RADIUS server, and must therefore know the location of the RADIUS server machine.

RRASHost*

IP address of a machine running Microsoft RRAS.

Default	N/A
Options	Valid IP address in the format 123.123.123.123
Required	No
Synopsis	*(Windows)* If Microsoft RRAS is in use, Websense software queries the machine running RRAS for user logon sessions. If no IP address is entered, no query occurs.

Timeout

Amount of time to wait for a response from the RADIUS server.

Default	1000 [milliseconds = 1 second]
Options	Integers greater than 500
Required	Yes
Synopsis	RADIUS Agent waits for a response to an authentication request from the RADIUS server for a specified amount of time before ending a query attempt.

eDirectory Agent

After configuring eDirectory Agent behavior in Websense Manager, you can customize the behavior of a specific eDirectory Agent instance in **wsedir.ini**, the agent’s initialization file. This file resides in the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).

- Some eDirectory Agent settings can only be configured via Websense Manager.
- Some settings can only be configured via the initialization file.

Some parameters can be modified either via Websense Manager or via **wsedir.ini**; these parameters are marked with an asterisk (*).

The parameters and values described here are case-sensitive.

DebugLevel

Determines the detail level of the eDirectory Agent diagnostic activity. (See definition for *DebugMode*.)

Default	0
Options	0, 1, 2, 3
Required	No
Synopsis	Specifies the level of log file detail provided for debugging purposes, from none (0) to high (3). Any value outside the range of 0-3 is interpreted as 0.

DebugMode

Controls the eDirectory Agent diagnostic activity.

Default	Off
Options	On, Off
Required	No
Synopsis	Enables or disables eDirectory Agent's built-in diagnostic (logging and debugging) capabilities. This can be a valuable tool for troubleshooting user identification problems, and determining whether eDirectory Agent is identifying Novell eDirectory users correctly.

DN*

Novell eDirectory server administrator name.

Default	None
Options	Any valid distinguished user name
Required	Yes
Synopsis	The distinguished name of a user with administrative rights in Novell eDirectory server. Novell eDirectory requires an authenticated name to issue LDAP requests. Should match the account specified on the **Settings** > **Directory Services** page in Websense Manager.

LogFile

Output file for eDirectory Agent diagnostic messages.

Default	N/A
Options	Any string of characters valid for your operating system
Required	No
Synopsis	If you have enabled *DebugMode*, specify a name for the text file where eDirectory Agent sends diagnostic (log) output.

password*

Novell eDirectory server administrator password.

Default	N/A
Options	Any string of characters
Required	Yes
Synopsis	The password for the Novell eDirectory server administrator account specified via *DN**. Should match the password specified on the **Settings > Directory Services** page in Websense Manager.

PollInterval

Interval at which to query Novell eDirectory for user logon sessions.

Default	30000 [milliseconds = 30 seconds]
Options	Any number of milliseconds
Required	Yes
Synopsis	Determines how long eDirectory Agent waits between Novell eDirectory server queries. ◆ A higher query frequency increases accuracy in identifying users but increases network traffic. ◆ A lower frequency may decrease immediacy in identifying users, but also decreases network traffic.

QueryMethod

Method (NCP or LDAP) used to query Novell eDirectory for user logon sessions, and whether each query is a full query are enabled (see *Enabling full queries*, page 37).

Default	1 [NCP]
Options	0, 1, 2, 3 [LDAP, NCP, LDAP + full queries, NCP + full queries]
Required	Yes
Synopsis	Determines whether eDirectory Agent uses NCP or LDAP to communicate with Novell eDirectory servers. Also determines whether eDirectory Agent performs a **full query** each time it polls the Novell eDirectory server. Enabling full queries is not recommended in larger networks, because the length of time required to return query results depends on the number of logged on users. The more logged-on users there are, the higher the performance impact.

SearchBase*

Novell eDirectory server root context.

Default	N/A
Options	Any string of characters
Required	Yes
Synopsis	The DN (distinguished name) of your Novell eDirectory root context. This value should match the root context specified on the **Settings > Directory Services** page in Websense Manager.

Server

IP addresses or names of machines running Novell eDirectory.

Default	N/A
Options	A valid IP address or host name
Required	Yes
Synopsis	Specify the identity of any machine running Novell eDirectory so that eDirectory Agent can query the directory service. If you are running multiple instances of Novell eDirectory, place each server entry on a separate line.

7 FAQs

DC Agent

- *Why isn't user logon data being transmitted?*
- *When are entries removed from the user map?*
- *What happens if Websense software, or the network, shuts down unexpectedly?*
- *Is transparent identification secure?*
- *Which ports listen for data transmitted by DC Agent?*
- *Can DC Agent run on a machine other than a domain controller?*
- *If I add a new domain controller, will DC Agent find it?*
- *Does the DC Agent service require special rights to run properly?*

Why isn't user logon data being transmitted?

For the agent to accurately capture logon session data, users must log on to domains, and not to their local machines. If a user logs on locally, the logon session is not recorded on any domain controller, and DC Agent cannot capture the logon sessions.

Data transmission between DC Agent and Filtering Service is reliable. There is no possibility of data being re-routed or lost along the way. If transmission fails at any point, it is probably because a service is not running. Use the Services dialog box (Windows) or the **WebsenseAdmin status** command (Linux) to check service status.

When are entries removed from the user map?

User name/IP address pairs that DC Agent obtains by querying the domain controller remain in the map for approximately 24 hours, by default. This interval has been randomized to prevent performance spikes, so the interval may vary by up to 20%.

Use Websense Manager to configure this interval.

User name/IP address pairs that DC Agent cannot verify using computer polling expire every hour, by default.

Use Websense Manager to configure this interval. Be aware that increasing the interval may lessen user map accuracy. Also, do not specify an interval smaller than the **User Map Verification Interval** to avoid removing user names from the map before they can be verified.

What happens if Websense software, or the network, shuts down unexpectedly?

DC Agent saves logon session data in the **XidDcAgent.bak** file periodically. The saved data will never be more than 10 minutes old at the time of shutdown. The agent reads the file at start, and resumes monitoring where it left off. Any users whose logon information was not captured may need to log on again to network domains to be identified.

If a power or network outage prevents DC Agent from communicating with other Websense software components, Filtering Service continues to use its existing user map to apply filtering.

Is transparent identification secure?

Yes, transparent identification is secure, because:

- Logon session data is translated to user name/IP address value pairs, and those pairs are sent over the network, without passwords. No other critical or proprietary information is transmitted.
- Transparent identification components use existing Windows networking calls to contact domain controllers and Filtering Service.

For increased security, configure an authenticated connection between DC Agent and Filtering Service. See *Securing agent communication with Filtering Service*, page 39, for details.

You can also disable NetBIOS usage (for example, to close the port used by NetBIOS traffic). In this case, DC Agent relies on DNS lookup to identify internet request sources. See *UseNetBIOS* for details.

Which ports listen for data transmitted by DC Agent?

By default, 30600. Filtering Service connects over this port when requesting user information from DC Agent. You can change this port in Websense Manager, or by editing the **transid.ini** file.

Can DC Agent run on a machine other than a domain controller?

DC Agent can be run on any Windows 2003 Server machine.

If I add a new domain controller, will DC Agent find it?

Yes. DC Agent identifies new and existing domain controllers every 24 hours by default, **and** at agent startup. This 24-hour interval is configurable via the *DiscoverInterval* parameter. DC Agent stores domain information in a file called **dc_config.txt**.

Does the DC Agent service require special rights to run properly?

If your network includes domains that are trusted by other domains, or allows NetBIOS traffic between virtually or physically separate subnets, then the DC Agent service must be configured to run with administrative rights. See page page 71 for details about troubleshooting DC Agent's domain detection behavior and configuring specific rights for the DC Agent service.

Logon Agent

- *What is the advantage of using Logon Agent?*
- *Why would I use Logon Agent in combination with DC Agent?*
- *Can I install both Logon Agent and eDirectory Agent on the same machine?*
- *Can I install multiple instances of Logon Agent on the same machine?*
- *If I add a new client machine to the network, does Logon Agent detect it?*
- *When do users expire from the user map?*

What is the advantage of using Logon Agent?

Logon Agent maximizes accuracy in identifying users as they log on to the network. While DC Agent identifies users by periodically querying domain controllers and client machines; Logon Agent identifies users in real-time, as they log on to domains. This eliminates the possibility of missing a user logon due to a query timing issue.

Why would I use Logon Agent in combination with DC Agent?

In most cases, using either agent is sufficient. However, there are many network environment variables that may affect which agent or combination of agents you choose. When Logon Agent works in conjunction with DC Agent, if DC Agent cannot capture a logon session (because of a timing issue or because a user logs on to a workstation outside its recognized domains), Logon Agent's associated logon application still captures that session.

Can I install both Logon Agent and eDirectory Agent on the same machine?

Websense software does not support communication with both Novell eDirectory and Windows NT Directory or Windows Active Directory at the same time. However, you can have both agents installed, with only 1 active agent.

Can I install multiple instances of Logon Agent on the same machine?

No. However, you can run multiple instances of Logon Agent within the network. Each instance must be able to communicate with Websense Filtering Service.

If I add a new client machine to the network, does Logon Agent detect it?

The logon script should activate the logon application (**LogonApp.exe**) on the new client machine. When a user logs on to that machine, the logon application detects the logon session, and sends that information to Logon Agent.

When do users expire from the user map?

When the logon application (**LogonApp.exe**) is **not** running in persistent mode, user name/IP address pairs in the map created by Logon Agent expire every 24 hours, by default. This interval, however, has been randomized to prevent performance spikes. Individual user entries expire after 24 hours, give or take 0-20% of that time period.

To change this interval in Websense Manager, go to **Settings** > **User Identification** and click on a Logon Agent instance. Then modify the **User entry expiration (nonpersistent mode)** value.

RADIUS Agent

- *Will users on client machines be identified regardless of logon domain?*
- *Can one RADIUS Agent communicate with multiple RADIUS servers, and vice-versa?*
- *Can one RADIUS Agent communicate with multiple Filtering Services, and vice-versa?*
- *Can I install both DC Agent and RADIUS Agent on the same machine?*
- *Can I install multiple instances of RADIUS Agent on the same machine?*
- *Can I install both eDirectory Agent and RADIUS Agent on the same machine?*
- *When do user map entries expire?*
- *Can RADIUS Agent ignore specific users and machines?*
- *How many resources does RADIUS Agent use?*
- *What happens if Websense software, or the network, shuts down unexpectedly?*

Will users on client machines be identified regardless of logon domain?

As long as the RADIUS server can authenticate the user as a domain user, filtering policies assigned to that user are applied properly.

Can one RADIUS Agent communicate with multiple RADIUS servers, and vice-versa?

One instance of RADIUS Agent can communicate with only one RADIUS server. However, one RADIUS server can communicate with multiple instances of RADIUS Agent.

If you have multiple RADIUS servers installed for backup purposes, you may want to install multiple instances of RADIUS Agent in your network. Each instance must be configured to communicate with one RADIUS server.

Do **not** install multiple RADIUS Agents on the same machine.

Can one RADIUS Agent communicate with multiple Filtering Services, and vice-versa?

One instance of RADIUS Agent can communicate with multiple Filtering Service instances.

You can configure one Filtering Service to communicate with multiple instances of RADIUS Agent on different machines. Do **not** install multiple RADIUS Agents on the same machine.

Can I install both DC Agent and RADIUS Agent on the same machine?

Websense, Inc., recommends running only one instance of a particular agent on any machine. DC Agent and RADIUS Agent can, however, run on the same machine.

Both agents are automatically installed in the Websense **bin** directory (C:\Program Files\Websense\bin, by default). Each agent uses a unique port number to communicate with Filtering Service. By default, DC Agent uses port 30600; RADIUS Agent uses port 30800.

Can I install multiple instances of RADIUS Agent on the same machine?

No. If multiple instances of the same agent are needed, those instances **must** be installed on separate machines.

Installation instructions for each agent are provided in the installation guides available from www.websense.com.

Can I install both eDirectory Agent and RADIUS Agent on the same machine?

Websense, Inc., recommends running one instance of an agent on any machine. eDirectory Agent and RADIUS Agent can, however, run on the same machine.

Both agents are automatically installed in the Websense **bin** directory. Each agent uses a unique port to communicate with Filtering Service. By default, eDirectory Agent uses port 30700; RADIUS Agent uses port 30800.

When do user map entries expire?

User name/IP address pairs in the map created by RADIUS Agent expire every 24 hours by default. However, this interval has been randomized to prevent performance spikes. Individual user entries expire after 24 hours, give or take 0-20% of that time period.

To change this interval in Websense Manager, go to **Settings** > **User Identification** and click a RADIUS Agent instance. Edit the **User entry timeout** value (under RADIUS server).

Can RADIUS Agent ignore specific users and machines?

Yes, RADIUS Agent can be configured to ignore particular logon names. See *Configuring an agent to ignore certain user names*, page 42.

How many resources does RADIUS Agent use?

Tests show that Radius Agent can handle 40-50 requests per second, and that it uses approximately 3% of the CPU on faster machines (1500MHz) and about 25% of the CPU time on slower machines (400-500MHz).

For 10,000 RADIUS users, RADIUS Agent operates with memory usage between approximately 15MB (for Windows) and 25MB (for Linux).

What happens if Websense software, or the network, shuts down unexpectedly?

RADIUS Agent saves logon session data in the **RadiusAgent.bak** file periodically. The saved data will never be more than 10 minutes old at the time of shutdown. The agent reads the file at start, and resumes monitoring where it left off.

If a power or network outage prevents RADIUS Agent from communicating with other Websense software components, Filtering Service continues to use its existing user map to apply filtering.

eDirectory Agent

- *Can one instance of eDirectory Agent communicate with multiple Novell eDirectory servers, and vice-versa?*
- *Can one instance of eDirectory Agent communicate with multiple Filtering Services?*
- *Can I install multiple instances of eDirectory Agent on the same machine?*
- *Can I install both DC Agent and eDirectory Agent on the same machine?*
- *Can I install both eDirectory Agent and RADIUS Agent on the same machine?*
- *When do user map entries expire?*
- *Can eDirectory Agent ignore specific users and machines?*
- *How long do entries remain in eDirectory Agent's user map?*
- *Are any Windows registry entries created for eDirectory Agent?*
- *How many users can eDirectory Agent handle?*
- *What happens if Websense software, or the network, shuts down unexpectedly?*

Can one instance of eDirectory Agent communicate with multiple Novell eDirectory servers, and vice-versa?

Yes, one instance of eDirectory Agent can communicate with multiple eDirectory servers. Technically, this means that one instance of eDirectory Agent supports one Novell eDirectory master, plus any number of Novell eDirectory replicas running on separate machines. To enable this:

- Add the IP addresses of all additional eDirectory servers in Websense Manager. (Go to the **Settings > User Identification** page, click on an eDirectory Agent instance, and then add the servers to the eDirectory Replicas list at the bottom of the page.)

- Ensure that all instances of Novell eDirectory server share the same user account and root context in order to ensure accurate user information from all of the servers.

See the Websense Manager Help for details.

Also, multiple instances of eDirectory Agent can communicate with one Novell eDirectory server.

Can one instance of eDirectory Agent communicate with multiple Filtering Services?

Yes. You can configure one instance of Filtering Service to communicate with multiple eDirectory Agents on different machines. If you let Filtering Service communicate with multiple eDirectory Agents installed on the **same** physical machine, see the next question.

Can I install multiple instances of eDirectory Agent on the same machine?

Websense, Inc., does not recommend running more than one instance of eDirectory Agent on one machine. Running multiple agents on a single machine could result in IP address and port conflicts, causing problems with user identification and filtering.

Can I install both DC Agent and eDirectory Agent on the same machine?

No. It is rare that both a Windows-based directory service and Novell eDirectory server reside in the same network, and Websense Manager does not currently allow configuration of two different types of directory service for the same installation.

Can I install both eDirectory Agent and RADIUS Agent on the same machine?

Websense, Inc., recommends running one instance of a particular Agent on a particular machine. However, eDirectory Agent and RADIUS Agent can run on the same machine.

Both agents are automatically installed in the Websense **bin** directory. Each agent uses a unique port to communicate with Filtering Service. By default, eDirectory Agent uses port 30700; RADIUS Agent uses port 30800.

When do user map entries expire?

User name/IP address pairs in the map created by eDirectory Agent expire every 24 hours by default. However, this interval has been randomized to prevent performance spikes. Individual user entries expire after 24 hours, give or take 0-20% of that time period.

To change this interval in Websense Manager, go to **Settings** > **User Identification** and click on an eDirectory Agent instance. Modify the **User entry timeout** value (under eDirectory Server).

Additionally, when a user logs out, that user name is removed from the user map when eDirectory Agent performs its next query of Novell eDirectory server. This query interval is 30 seconds by default, and is determined by the *PollInterval* parameter in **wsedir.ini**.

Can eDirectory Agent ignore specific users and machines?

Yes, eDirectory Agent can be configured to ignore particular logon names. See *Configuring an agent to ignore certain user names*, page 42.

How long do entries remain in eDirectory Agent's user map?

The user map is updated every 30 seconds, by default, when the agent queries Novell eDirectory server for new logons. Use the *PollInterval* parameter to adjust the frequency of the query.

If a user logs on to the local machine (rather than the network), eDirectory Agent adds an IP address and an empty user name placeholder to the user map.

The **User entry timeout** setting in Websense Manager (go to **Settings** > **User Identification**, and then click an eDirectory Agent instance) determines the lifetime of user/name IP address pairs in the map. By default, the timeout is 24 hours.

Are any Windows registry entries created for eDirectory Agent?

The only registry entries added during eDirectory Agent installation are the standard keys required to run the agent as a Windows service. These keys are stored under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**.

How many users can eDirectory Agent handle?

Tests show that eDirectory Agent can process about 500 user authentication requests per second. Therefore, processing 10,000 requests takes approximately 20 seconds.

What happens if Websense software, or the network, shuts down unexpectedly?

eDirectory Agent saves logon session data in the **eDirAgent.bak** file periodically. The saved data will never be more than 10 minutes old at the time of shutdown. The agent reads the file at start, and resumes monitoring where it left off.

If a power or network outage prevents eDirectory Agent from communicating with other Websense software components, Filtering Service continues to use its existing user map to apply filtering.

8 Troubleshooting

Troubleshooting tools

Windows Services (Service Control Manager)

Transparent identification agents, User Service, and Filtering Service run as Windows services, and therefore appear in the Windows Services dialog box. To access the Services dialog box, go to **Start >Programs** > **Administrative Tools** > **Services**.

Use the Services dialog box to check current service status, and to stop, start, or restart services.

Windows Event Viewer

The Windows Event Viewer records errors, warnings, and informational alerts related to service activities, and can be useful in uncovering network or service issues. To access the Event Viewer, go to **Start** > **Programs** > **Administrative Tools** > **Event Viewer**, and then double-click the **Application** log.

Websense log file

On all platforms, Websense software writes errors to the **Websense.log** file in the Websense **bin** directory (/opt/Websense/bin, or C:\Program Files\Websense\bin, by default). This information in this file is comparable to that found in the Windows Event Viewer.

Websense ConsoleClient

Websense software includes a diagnostic tool to help Websense Technical Support obtain troubleshooting information. The tool is installed in the Websense **bin** directory (/opt/Websense/bin or C:\Program Files\Websense\bin, by default). The tool's Tracing and Printself modules produce data that can be saved to a text file or viewed real-time from a terminal window.

Websense Technical Support can use data obtained via ConsoleClient to determine the source of any problems with the transparent identification process, if other troubleshooting methods have not revealed a root cause. For example, if a user is not being filtered properly, but the user name and IP address have been recorded by the agent and User Service, ConsoleClient can be used to gather data that may reveal the cause.

If user identification problems occur, first check all network connections, and then check the Windows Event Viewer or the Websense log file for related error messages. It may not be necessary to use ConsoleClient at all.

The transparent identification agents store user name-to-IP address correspondences to a user map in local memory. You can analyze the user name/IP address pairs to determine whether users and machines are being identified correctly.

To obtain user map output from ConsoleClient:

1. Open a command prompt or command shell on the Filtering Service machine and navigate to the Websense **bin** directory.
2. Enter the following command:

   ```
   consoleClient [IP address of agent machine] [port number]
   ```

✓ **Note**

For RADIUS Agent, use the server **Diagnostic port** value (30801, by default).

3. Send the output of the previous command to a file to see the user name-to-IP address map.

Websense Technical Support can assist you in using ConsoleClient at a more detailed level. Some known problems that can hinder the transparent identification process are described under *Common Problems*, page 69.

Websense TestLogServer

TestLogServer is a diagnostic form of Websense Log Server that can be used to see if a user is being properly identified. TestLogServer cannot be run on a machine that is currently running Websense Log Server.

To use TestLogServer to verify user identification:

1. If you are performing the test on the Log Server machine, stop the Websense Log Server service.
2. Open a command prompt or shell.
3. Navigate to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).
4. Enter the following command:

   ```
   testlogserver
   ```

   This command runs a process on port 55805 that listens to data being sent from Filtering Service.

By default, TestLogServer output is send to the console. To send output to a file, add the **-file** parameter:

```
testlogserver -file filename.txt
```

5. To ensure that Websense software is passing traffic to the TestLogServer machine:
    a. Open Websense Manager and go to the **Settings > Logging** page.
    b. If necessary, replace the **Log Server IP address or name** value with the IP address of the TestLogServer machine.
    c. Click **OK** to cache the change, and then click **Save All**.
6. Ask a user to access a site.
7. Check the log file for this site and user name.
8. When you have collected sufficient data, press **CTRL+C** to stop TestLogServer.
9. Restart the Websense Log Server service, if necessary, or update Websense Manager to resume sending traffic to the Log Server machine.

If correct user information appears in the TestLogServer output, there may be a policy configuration problem. Use the Websense Manager **Check Policy** and **Test Filtering** tools to troubleshoot the problem.

If the output shows something other than a user name, or the wrong user name, refer to the *Common Problems* section for more troubleshooting tips.

If no user name appears, refer to the Websense Manager Help to verify that the transparent identification agent and related Websense software components are configured correctly.

Websense RADIUS Agent Diagnostic Tool

To activate RADIUS Agent logging and debugging:

1. Stop the Websense RADIUS Agent service.
2. Navigate to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).
3. Open the file **wsradius.ini** in a text editor.
4. Locate the [RADIUSAgent] section.
5. To enable logging and debugging, edit the **DebugMode** entry to read:

    ```
    DebugMode=On
    ```

6. Modify the **DebugLevel** entry to read:

    ```
    DebugLevel=3
    ```

    Level 3 provides the highest level of debugging detail.
7. Modify the **LogFile** entry to read:

    ```
    LogFile=RadiusLog.txt
    ```

    This causes log output to be sent to a file called **RadiusLog.txt**. You can enter a different file name, if you prefer, or leave the entry blank to send debugging information to the console.

8. Save and close the **wsradius.ini** file.
9. Start the Websense RADIUS Agent service.
10. Open the RADIUS Agent log file (stored by default in the **bin** directory).

    When a user is identified correctly, the log record resembles the following:

    ```
    15-01-2004 14:31:28 Received request from 10.202.11.14
    15-01-2004 14:31:28 WsRadiusPacket::Parse
    15-01-2004 14:31:28 code=Accounting-Request, id=15,
    size=294;
    15-01-2004 14:31:28 name=TEST\admin
    15-01-2004 14:31:28 Framed-IP-Address=10.202.11.7 (aca0b07)
    15-01-2004 14:31:28 Accounting status type=1
    15-01-2004 14:31:28 Forwarding request to RADIUS server
    15-01-2004 14:31:28 Received reply from RADIUS server
    15-01-2004 14:31:28 Forward response to client 10.202.11.14
    15-01-2004 14:31:28 WsRadiusPacket::Parse
    15-01-2004 14:31:28 code=Accounting-Response, id=15,
       size=20;
    15-01-2004 14:31:28
    WsProxyThreadAcc::ProcessPackets(2e8e710,
       2e8eb24)
    15-01-2004 14:31:28 Adding entry to user map: ip=aca0b07,
       user=TEST\admin
    ```

    This indicates that RADIUS Agent detected the user **TEST\admin** logging into IP address **10.202.11.7**, received an authentication response from the RADIUS server, and added the entry to its user map.

Websense eDirectory Agent Diagnostic Tool

To activate eDirectory Agent logging and debugging:

1. Stop the Websense eDirectory Agent service.
2. On the eDirectory Agent machine, navigate to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default).
3. Open the file **wsedir.ini** in a text editor.
4. Locate the **[eDirAgent]** section.
5. Modify the **DebugMode** entry to read:

    ```
       DebugMode=On
    ```

6. Modify the **DebugLevel** entry to read:

    ```
    DebugLevel=3
    ```

    Level 3 provides the highest level of debugging detail.

7. Modify the **LogFile** entry to read:

    ```
    LogFile=eDirLog.txt
    ```

    This causes log output to be sent to a file called **eDirLog.txt**. You can enter a different file name, if you prefer, or leave the entry blank to send debugging information to the console.

8. Start the Websense eDirectory Agent service.

9. Go to the eDirectory Agent installation directory and open the eDirectory Agent log file.

   A successfully identified logon session looks like this:

   ```
   22-11-2004 11:15:09 Adding user WS\admin (10.1.0.76) to
      user map
   ```

   The user **admin** logged onto Novell eDirectory server and was identified correctly by eDirectory Agent.

Common Problems

If users are not being filtered by the correct policy, there may be a problem with transparent user identification. The cause may be:

- The transparent identification agent and Filtering Service are not communicating.
- A user's network identity is obscured, preventing Filtering Service from applying policies assigned to that directory client.
- DC Agent is not receiving domain controller information.
- (*Logon Agent*) The logon script that invokes **LogonApp.exe** on a client machine does not run properly.
- (*Logon Agent*) NetBIOS or a NetBIOS service is disabled on the client machines.
- A user is not authenticated by the RADIUS server or by Novell eDirectory server.
- The client cannot connect to Remote Filtering Server.

If a user can successfully log on to the network but is not being filtered properly by Websense software, first check that:

- Policies are configured appropriately and assigned to the correct users in Websense Manager.

  You can use the **Check Policy** and **Test Filtering** tools (accessed via the Toolbox in the right shortcut pane) to see which policy currently applies to the user, and how a request from the user for a specific site is filtered.
- Client machines connect to the Internet via the correct gateway.

Filtering Service cannot communicate with an agent

If Filtering Service is not receiving user name / IP address pairs from a transparent identification agent:

- Users may be filtered by computer or network policies, or the **Default** policy, even after directory policies have been assigned.
- User names may be logged incompletely or incorrectly.

To determine whether a transparent identification agent can exchange data with Filtering Service:

1. Check for errors and warnings Event Viewer (Windows) or the **websense.log** file (Windows or Linux) relating to successful or unsuccessful connections between the agent and Filtering Service.
   - If connection messages show **success**, the agent was probably not running when the user logged on. Verify that the agent service is running, and then have users log on again.
   - If connections were **unsuccessful**, check to see if "authentication failed." This indicates a problem with the password for connections between the agent (see Step 5).
2. Check to see that all transparent identification agent machines are running and connected to the network, and that the agent has started successfully.
3. Verify that it is possible to connect to the agent machine on the agent's communications port, configured in Websense Manager. The defaults are 30600 (DC Agent), 30602 (Logon Agent), 30700 (eDirectory Agent), and 30800 (RADIUS Agent).

   To do this, open a command prompt and use the **telnet** command:

   ```
   telnet [IP address] [port]
   ```

   For example:

   ```
   telnet 10.201.77.15 30700
   ```

   A blinking cursor indicates a successful connection.
4. If the agent is running on Linux, use the **netstat** command to check the machine's network connections:

   ```
   netstat –an > netstat.txt
   ```

   This example sends the output of the **netstat** command to a text file.

   Make sure you see an entry for **15880** (the Filtering Service communication port), and that the connection is set to listen.
5. Use the **Settings** > **User Identification** page in Websense Manager to make sure that the agent is configured correctly.
   a. In the Transparent Identification Agents list, click the IP address or host name of the agent instance that is not authenticating correctly.
   b. If you are using an authenticated connection with Filtering Service, make sure the **Enable Authentication** option is checked, and then re-enter the password. (If you have manually added a password to the agent's INI file, see Step 6.)
   c. Click **OK** to return to the User Identification page, and then click **OK** again to cache your changes. Changes are not implemented until you click **Save All**.
6. If you have enabled an authenticated connection between the agent and Filtering Service in the agent's INI file:
   a. On the agent machine, go to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default), and open the INI file (see *Initialization parameters*, page 43).
   b. Locate the **password** parameter, and then make sure that:
      - The parameter is spelled with a lower-case "p".

- The password is correct.

c. Save the file, and then restart the agent service.

7. Filtering Service cannot transmit data to or receive data from User Service or a transparent identification agent while the database download process is running.
   - Check the Health Alert Summary on the Status > Today page or check the Status > Today > Database Download page in Websense Manager to see if a download is currently in process on any Filtering Service machine.
   - Check for errors or warnings in the Event Viewer (Windows) or the **websense.log** file (Windows/Linux) regarding successful or unsuccessful connections between User Service and Filtering Service.

     If the Master Database download process was running, an Event Viewer message states that "data request failed." This problem should resolve itself after the database has finished downloading.
8. If you are using Logon Agent, make sure that the logon application can communicate with Logon Agent. To do this, open a command prompt on a client machine where the logon application is installed, and then run the following command:

   ```
   logonapp.exe http://<Logon Agent IP address>:15880
   ```

   This establishes that there is a connection between the logon application and the Logon Agent machine.

DC Agent

A user's or machine's network identity is obscured

Filtering Service cannot correctly apply user-based policies when the identity of a user making an Internet request is uncertain. This uncertainty can occur incorrect user name/IP address pairs are associated with machines (due, for example, to a service on the user's machine contacting the domain controller with its own user name).

See *Configuring an agent to ignore certain user names*, page 42, to configure DC Agent to ignore logon names not associated with actual users.

If a user name is missing from the DC Agent user map, run **net use %logonserver%**. You do not need to log off and log back on. This command initiates a logon session with the machine hosting the domain controller. You can add the missing user name, and then run ConsoleClient to verify that the user name appears in the map.

You can also run the **set** command, which will show which domain the user is authenticated against, and then you can compare that domain to the setting in the **dc_config.txt** file.

DC Agent is not receiving domain controller information

DC Agent can misidentify users if it is unable to get data from domain controllers, resulting in incorrect filtering behavior. This can happen in the cases described here.

- NetBIOS is not enabled between DC Agent and domain controllers. First, verify that DC Agent has a NetBIOS connection to each domain controller.
- DC Agent may not be detecting all domain controllers in the network. (See *Check domain controller entries in dc_config.txt*, page 72.)
- DC Agent may not be able identify the domain controllers in a particular domain. (See *Check for browser server errors*, page 72.)
- DC Agent and User Service may be configured to use an anonymous account. (See *Configure DC Agent and User Service permissions*, page 72.)
- DC Agent may not be able to contact a remote domain controller that has been shut down or restarted. (See *Remote domain controller connections issues*, page 73.)

Check domain controller entries in dc_config.txt

To verify that DC Agent has detected all domain controllers in the network:

1. On the DC Agent machine, navigate to the Websense **bin** directory (C:\Program Files\Websense\bin, by default) and open the file **dc_config.txt** in a text editor.
2. Verify that all of the domains in your network are listed, and that all active domain controllers are assigned a value of **on**. For example:

   ```
   [Domain1]
   DC1A=on
   DC2A=on

   [Domain2]
   DC1B=on
   DC2B=on
   DC3B=off
   ```

   Here, the names in brackets ([Domain1], [Domain2]) are domain names, and the **DC** entries are domain controller names.

Check for browser server errors

To determine whether DC Agent cannot identify certain domain controllers:

1. Open the Windows Event Viewer (instructions are on page 65).
2. If DC Agent cannot locate a Master Browser with a list of domain controllers for a particular domain, you see the message ERROR_NO_BROWSER_SERVERS_FOUND – 6118 in the Event Viewer.

   If your network includes multiple subnets, DC Agent may have problems communicating with Master Browser or domain controller machines in other subnets. Websense recommends installing a separate DC Agent in each subnet to avoid problems gathering logon information from domain controllers.

Configure DC Agent and User Service permissions

To a domain controller, an anonymous account is equivalent to a Windows Guest account. If DC Agent is configured to use an anonymous account, and the domain controller has been set not to give the list of user logon sessions to an anonymous user, then DC Agent is unable to retrieve logon information.

Websense, Inc., recommends running DC Agent and User Service with domain administrative rights. Certain networking calls that these services use may fail if the services have insufficient rights.

DC Agent uses the **NetSessionEnum** call (see http://msdn2.microsoft.com/library), which may fail depending on your Local Security Policy or Trust Relationship configuration.

User Service uses **NetUserGetGroups**, which requires domain administrative rights.

To troubleshoot the problem:

1. Open the Windows Event Viewer on the DC Agent machine, and look for the message "ERROR_ACCESS_DENIED - 5".

   If this message appears, DC Agent does not have sufficient permissions to access the domain controller. Continue with this procedure.

2. In each domain, create a user with a descriptive name (like **wsDCAgent**). You can use an existing account, but setting up a new account is preferable so that the password can be set not to expire.

   - Assign domain administrative privileges to the new user account.
   - Assign the same password to the account in all domains.
   - Set the password never to expire. This account has no function other than to provide a security context for accessing directory objects.

   Remember the user name and password you establish for this account; they must be entered later.

3. Open the Windows Services dialog box on the DC Agent machine (**Start** > **Programs** > **Administrative Tools** > **Services**.)

4. Double-click the **Websense DC Agent** entry, and then click **Stop**.

5. On the **Log On** tab of the Properties dialog box, select the **This account** radio button, and then enter the user name and password for the account that you created in Step 2.

   Some environments require that the account name be entered in the format "domain\user name" (for example, **Domain1\wsDCAgent**).

6. On the **General** tab, click **Start** to restart the DC Agent service, and then click **OK** to close the dialog box.

DC Agent now runs using an account with sufficient rights to access domain controllers.

Repeat the procedure for User Service, if necessary.

Remote domain controller connections issues

To determine whether a network problem is preventing DC Agent from contacting a domain controller, check the Windows Event Viewer on the DC Agent machine for a message like:

```
ERROR_BAD_NETPATH - 53
```

To troubleshoot the remote access problem:

1. Ensure that the Remote Registry Service is started on the DC Agent machine, and that the related application is shared on the domain controller machine.
2. To find out if NetBIOS is bound to the network adapter on the DC Agent machine:
    a. Right-click **My Network Places**, and then select **Properties**.
    b. On the General tab of the Local Area Connection Properties dialog box, select **Internet Protocol (TCP/IP)**, and the click **Properties**.
    c. Click **Advanced**, and then select the **WINS** tab of the Advanced TCP/IP Settings dialog box.
    d. Under NetBIOS setting, verify that one of the first 2 radio buttons is selected.
    e. If you made changes, click **OK** 3 times to close the various Properties dialog boxes.
3. Use the Windows Services dialog box to verify that Printing and Network services are running on the DC Agent machine.
4. Make sure that remote administration is enabled on the remote domain controller machine.

Logon Agent

If Logon Agent cannot get a user name/IP address pair from a client machine, Websense software does not apply the appropriate user or group policy. In most cases, this occurs because there is some problem getting information from the machine where the request originated. As a result, a user may not be identified even after logging on to the network.

The user's machine is not connected to the appropriate shared network location

The client machine must be connected to the shared drive on the domain controller where **LogonApp.exe** and the logon script are stored.

To determine if a client machine has access to the domain controller, open a Windows command prompt and enter the following:

```
net view /domain:<domain name>
```

NetBIOS for TCP/IP is disabled

NetBIOS for TCP/IP must be enabled on client machines and on the machine running Logon Agent. If NetBIOs is disabled:

- On client machines, **LogonApp.exe** may not be able to run.
- On the Logon Agent machine, Logon Agent may not be able to communicate with domain controllers.

In either case, users are filtered by the computer, network, or **Default** policy, or prompted for manual authentication.

The TCP/IP NetBIOS Helper service is not running on the client machine

The TCP/IP NetBIOS Helper service must be running on each client machine needing to be identified by Logon Agent. This service runs on Windows 2000, Windows XP, Windows 2003, and Windows NT.

If this service is not running, **LogonApp.exe** cannot be properly deployed on client machines, and therefore cannot capture logon sessions.

The user profile stored on the client machine is corrupt

The Windows user profile on the client machine must be intact for the logon script to run. A user profile can become corrupt due to Windows factors external to Websense software.

To restore a corrupt user profile:

1. Log on as the machine’s local administrator.
2. Back up the files in the user’s **My Documents** folder (if possible). This folder and all of its contents will be deleted when you remove the corrupted profile.
3. Remove the corrupted user profile from **C:\Documents and Settings**.
4. Log off and then log on again.

Windows recreates the profile automatically, and the logon script executes.

RADIUS Agent

VPN Usage

The VPN client is not successfully logged onto the VPN network

To verify that RADIUS server is authenticating clients, check the RADIUS server’s log file for the user name in question.

For Microsoft IAS, go to the IAS management console and see **Remote Access Logging** to find out where the log file is. (Set which actions are logged via the **Properties** panel).

RADIUS Agent may impact a VPN connection

Because RADIUS Agent sits between a VPN client and VPN server, RADIUS Agent may block VPN traffic. In this case, you must remove the Agent. Simply stopping the

Agent is not sufficient; the Agent must be removed from the link between the RADIUS client and server.

To ensure that RADIUS Agent is removed:

- On the VPN client (in most cases these are RAS servers), configure the client to communicate directly with the server. In most cases, this involves setting the IP of RADIUS server and changing the port number from 12345 to 1812.
- On the RADIUS server, simply remove RADIUS Agent as a client.

There is incorrect domain information in the VPN client

Make sure the VPN client has the correct domain information set before users log on to the network. Active Directory, for example, may contain both parent and child domains.

RADIUS Client/Server Configuration

The client cannot be filtered by the IP address assigned by RAS

A client might be successfully authenticated by the RADIUS server, but not filtered correctly. If the client cannot be filtered by IP address (the new IP address assigned by RAS for the corporate network) even before RADIUS Agent receives the user information, there may be something wrong with your VPN setup.

RADIUS Agent fails to start

If RADIUS Agent does not start, check your RADIUS Agent logs for the message 'Cannot bind to port: 10048' (Windows) or 'Cannot bind to port: 98' (Linux).

The usual cause is that another application (for example, a second instance of RADIUS Agent, or the RADIUS server) is currently running on the RADIUS Agent machine and using the same port RADIUS Agent is defined to use. Ensure that each RADIUS application on the RADIUS Agent machine uses a different port.

There are warnings or error messages in the Event Log

The Event Log for the RADIUS server can be helpful in determining the cause of VPN connection or authentication problems, and in distinguishing whether the problem lies in RADIUS Agent or VPN setup.

RADIUS Accounting is not enabled on the RADIUS server when it should be

With some RADIUS servers (Microsoft IAS for example), RADIUS Accounting must be enabled so that RADIUS Agent can get the IP address of the RADIUS client.

The RADIUS server should include the attributes **User-Name** and **Framed-IP-Address** in authentication and accounting messages. RADIUS Agent uses the values of these attributes to interpret and store user name/IP address pairs. If your RADIUS server does not generate this information by default, configure it to do so. See your RADIUS server documentation for instructions.

RADIUS Agent has not been added as a client to the RADIUS Server

Configure your RADIUS server to use Websense RADIUS Agent as a proxy. This involves adding RADIUS Agent as a client to the RADIUS server. See your RADIUS server documentation for instructions on configuring a proxy. If you have multiple RADIUS servers, each server must be configured separately. Failure to configure RADIUS Agent as a proxy results in a RADIUS connection failure, even before RADIUS Agent can function.

Is RADIUS Authentication for Windows domain users in use?

If you require the RADIUS server to authenticate Windows domain users, the RADIUS server may need to reside in the same Windows domain as these users. See your RADIUS server documentation for information on domain user authentication.

Is Livingston RADIUS server in use?

Lucent RADIUS Server must be configured to use Password Authentication Protocol (PAP), and the RRAS server must be configured to accept only PAP requests. For instructions, see your respective product documentation.

Is Microsoft Routing and Remote Access Server (RRAS) in use?

Run RADIUS Agent with administrative rights on an RRAS server. This ensures that when it is restarted, RADIUS Agent can retrieve all currently logged-on users from the RRAS server. In most cases, domain administrative rights are sufficient.

To verify that RADIUS Agent is retrieving all currently logged-on users, check the RADIUS Agent log file for the following entry:

```
WsRadiusApp::StartAgent()
WsRRASInspector::Inspect(127.0.0.1, 151ff24)
Adding RRAS entry to user map: ip=C0A8030C,
user=SOFIA\radiustest
```

(See *Websense RADIUS Agent Diagnostic Tool*, page 67 for instructions on enabling RADIUS Agent logging and debugging.)

Websense User Identification

Remote users are not being filtered correctly

If remote users are not being filtered by Websense software, or are not being filtered by particular policies assigned to them, check your RADIUS Agent logs for the message **Error receiving from server: 10060** (Windows) or **Error receiving from server: 0** (Linux).

This usually occurs when the RADIUS server does not recognize RADIUS Agent as a client (source of RADIUS requests). Ensure that your RADIUS server is configured as described in the Websense Manager Help.

Users bypass a logon prompt to circumvent Websense filtering

If a user logs on to a RADIUS server as a local user, the user is identified as RADIUS_SERVER_HOST\username. Because there is no way to assign a policy to a user unless the user belongs either to a domain or to an LDAP container, this local user is filtered only by the **Default** policy. The **Default** policy is enforced if no user name/IP address pairing is captured and no other filtering settings take precedence.

Run TestLogServer to check whether the user is logged on locally. See page 65 for instructions.

Alternatively, you can apply policies to computers or network ranges, rather than only to users and groups. See the Websense Manager Help for more information about working with computer and network clients.

eDirectory Agent

Users are not being filtered correctly

This happens when Filtering Service does not get the appropriate user information from eDirectory Agent. Possible causes and solutions are described here.

- Users are not logging onto Novell eDirectory server. Users might be bypassing a logon prompt, or logging on to a different domain and circumventing the Websense filter.

  If a user does not log onto Novell eDirectory server, there is no way for Websense software to capture the user name/IP address pair, and apply a user-specific policy to that user. In this case, Websense software applies a workstation or network policy (if one exists), or the **Default** policy.

  Run TestLogServer to check whether the user is logged on locally. See page page 65 for instructions.

  Alternatively, a workstation rule can be set up for filtering local users based on the workstations where they log on. See the Websense Manager Help for instructions on configuring filtering policies for workstations and other objects.

- The root context set in the **wsedir.ini** file is different from the one set for eDirectory Agent in Websense Manager. In this case, although the user can be identified, Websense software may not be able to apply the correct filtering policy. The user may be filtered by a workstation or network policy (if applicable), or by the **Default** policy.

  If these root context values are different, a user can log on to two different trees or branches in Novell eDirectory server, and still be identified by eDirectory Agent. However, when Websense Filtering Service determines the filtering policy for this user, it uses the root context specified in Websense Manager to retrieve information. Filtering Service cannot determine the appropriate filtering policy for a user logging into a Novell eDirectory tree or branch outside the specified root context.

  Ensure that you are using the same user and the same root context in both the .ini file and Websense Manager.

  To verify the root context value in **wsedir.ini**:

  a. On the eDirectory Agent machine, go to the **Websense\bin** directory, and open the file **wsedir.ini**.

  b. Verify the line

  ```
  SearchBase=[DN]
  ```

  where DN is the Distinguished Name of the eDirectory root context.

  c. Save the file, and then restart eDirectory Agent to activate the changes.

- eDirectory Agent is running on Linux, and the Novell Modular Authentication Service (NMAS) is running when it should not be.

  In order for eDirectory Agent to work properly on Linux, NMAS must be disabled in Novell eDirectory server. See your Novell documentation for instructions.

Users are not identified in a Cisco Wide Area Application Engine environment

In an environment where Websense software is integrated with Cisco Wide Area Application Engine (formerly Cisco Content Engine) v5.3.1.5 or higher, eDirectory Agent may be unable to identify users unless particular setup guidelines are followed. You need to:

- Install and run these Websense services on the same machine as Cisco WAE (Content Engine):
    - Websense eDirectory Agent
    - Websense User Service
    - Websense Filtering Service
    - Websense Policy Server
- Ensure that all Novell eDirectory replicas are added to the **wsedir.ini** file on the same machine. This file is located in the Websense installation directory (**\Websense\bin** by default).
    a. Stop the eDirectory Agent service.
    b. On the eDirectory Agent machine, go to the eDirectory Agent installation directory.
    c. Open the file **wsedir.ini** in a text editor.
    d. Locate the section named `[eDirAgent]`.
    e. For each new instance of eDirectory server, add the line

       ```
       Server=[X]:port
       ```

       where X is the IP address or name of the machine running eDirectory or an eDirectory server replica, and port is the port over which the eDirectory server connects to Websense eDirectory Agent. Be sure to use a valid port number.
    f. For any instance of eDirectory server that no longer exists, remove the line

       ```
       Server=[X]:port
       ```
    g. Save and close the file.
    h. Start eDirectory Agent.
- Delete the **eDirAgent.bak** file from the machine running Websense, Cisco WAE (Content Engine) and eDirectory Agent. By default, **eDirAgent.bak** is in the **\Websense\bin** directory.
- Run any Websense Reporting Tools services on a machine *separate* from Cisco WAE (Content Engine) and Websense software.

See the *User Identification* topic in the Websense Manager Help for complete instructions on configuring Websense to communicate with eDirectory Agent.

Remote Filtering

Remote filtering problems can result when:

- The Remote Filtering Server is down.
- The client cannot connect to Remote Filtering Server.
- Network Agent is filtering responses to remote filtering requests.
- Remote Filtering Server and Filtering Service are installed on the same machine.
- DHCP is enabled on the Remote Filtering Server machine.
- If the Remote Filtering Server machine is running with Windows Server 2003, Service Pack 1 is not installed.
- Communications are not properly configured:
  - Ensure that the proper IP addresses have been configured for internal and external communication.
  - Ensure that the proper ports have been configured for internal and external communication.
- Passphrases do not match.
- The load balancer is not forwarding packets to the server.

The steps below cover only those causes related to user identification. To troubleshoot general remote filtering problems, see the Websense Knowledge Base.

1. Verify that the Remote Filtering Server machine is running and connected to the network.
2. Check that the Remote Filtering Server service is running.
   - *Windows:* Check the Windows Services dialog box to find the status of the **Websense Remote Filtering Service**.
   - *Linux:* Go to the **/opt/Websense** directory and enter the following command:

     ```
     ./WebsenseAdmin status
     ```
3. Ensure that Remote Filtering Server is not installed on the same machine as Filtering Service.

   Installing these components on the same machine causes a serious drain on resources on the machine. Filtering is slow at first, and then fails and allows all requests.
4. Check that the connections are working properly.
   a. Use the **ping** command to see if remote machines are able to communicate with the server.
   b. Check that the server machine is properly communicating with the network. Try to ping other machines on the local network.

Remote Filtering Client instances must be able to connect to Remote Filtering Server, from both inside and outside the Internet gateway or network firewall. To verify that communication between the server and the clients is configured correctly:

1. On the Remote Filtering Server machine, navigate to the Websense **bin** directory (C:\Program Files\Websense\bin or /opt/Websense/bin, by default) and open the **securewispproxy.ini** file in a text editor.
2. Make a note of the following values:
   - **ProxyIP** (the IP address of the NIC used for internal communications)
   - **ProxyPort** (the port used on the server for external communications; 80, by default)
   - **ProxyPublicAddress** (the IP address or host name used for external access to the Remote Filtering Server machine)
   - **HeartBeatPort** (the port used for internal communications by Remote Filtering Client machines that have moved inside the network firewall; 8800, by default)
3. Open a command prompt on the Remote Filter Server machine, and then enter the **ipconfig** (Windows) or **ifconfig** (Linux) command to get the IP addresses for each network card in the machine.
4. Check that the IP address values match the values found in the **securewispproxy.ini** file.

   The values need to be checked on the client machines. Contact your Websense Technical Support provider for assistance. The technician needs the information gathered in the previous steps to verify that communications are properly set up.