WEBSENSE™ Enterprise

v6.1

# Transparent Identification of Users in Websense Enterprise v6.1

# Transparent Identification of Users in Websense Enterprise v6.1

©1996–2005, Websense Inc.
All rights reserved.
10240 Sorrento Valley Rd., San Diego, CA 92121, USA

Published November 18, 2005
Printed in the United States of America

### <Part Number>

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 and Websense Enterprise are registered trademarks 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 NT, Windows 2000, Windows 2003, Windows XP, Internet Explorer, and Active Directory are trademarks or registered trademarks of Microsoft Corporation.

Solaris is a registered trademark of Sun Microsystems, Inc., in the United States and other countries. Sun, Sun ONE and all Sun ONE based trademarks and logos are trademarks of Sun Microsystems, Inc.

Netscape is a registered trademark of Netscape Communications Corporation in the U.S. and other countries. Netscape Navigator and Netscape Communicator are also trademarks of Netscape Communications Corporation and may be registered outside the U.S.

The following is a registered trademark of Novell, Inc., in the United States and other countries: Novell Directory Services.

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.

### WinPcap

# Contents

CHAPTER 1

# Introduction

In order to provide filtering specific to users and groups, as well as optional internet usage reporting, Websense must identify users in your network. Websense can identify users in one of three ways:

1. Your integration partner product (firewall or cache product) sends user names to Websense.
2. Websense identifies users transparently.
3. Users are prompted to authenticate manually (by responding to a logon prompt from a web browser).

This paper explores transparent identification (method 2 above) in greater detail, describing the identification process and functionality of the components involved. Frequently-asked questions and troubleshooting information specific to transparent identification are provided to help facilitate your implementation of this convenient feature.

The transparent identification feature allows Websense to filter user internet requests without prompting users to log on via a web browser. You have several options for implementing transparent identification in your network.  In all cases, the Websense User Service must be configured appropriately.

> ✓ **NOTE**
>
> If your network uses RAS servers or thin clients (such as a terminal server or Citrix Server) without RADIUS authentication, Websense, Inc. recommends that you do not implement the transparent identification feature. In such an environment, Websense may not be able to identify individual users. Contact Websense Technical Support for information about related workarounds, or refer to Websense Knowledge Base article 156.

## The Websense Transparent Identification Agents

Websense currently provides four optional components for identifying users transparently in various environments. All four enable Websense to filter based on policies assigned to users or groups housed in a directory service. Theses optional components can be used alone, or combined, with certain limitations. Limitations are noted later in this section.

- **Websense DC Agent:** Can be used with a Windows-based directory service. The Websense DC Agent is installed on a Windows Server in the network. DC Agent does not need to reside in any particular domain.
- **Websense Logon Agent:** Can be used with Windows-based client machines, plus either Windows Active Directory or an NTLM-based directory service.
- **Websense RADIUS Agent:** Can be used in conjunction with either Windows- or LDAP-based directory services. Works together with a RADIUS client and RADIUS server to identify users logging in from remote locations.

- **Websense eDirectory Agent:** Designed specifically for use with Novell eDirectory. Detects users logged on to Novell eDirectory.

Websense Enterprise v6.1 also provides optional components for filtering and identifying remote users. Remote Filtering Server and Remote Filtering Client work together with your transparent identification agent to identify users logging on from outside the network. See *Chapter 6 Remote Filtering* for information.

Each component involved in transparent identification is vital to the process. Understanding the pieces and how they work together can help you to implement and troubleshoot this feature. See the relevant *Components* section for information about the physical structure of transparent identification with DC Agent, Logon Agent, RADIUS Agent, or eDirectory Agent.

## Combining Transparent Identification Agents

Websense, Inc. supports certain combinations of the agent components within the same network, or on the same machine. Generally, it is recommended 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, you can configure Websense to work with multiple agents on a single machine in some cases.

Supported combinations are 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 all agents to communicate with Filtering Service. Also see Websense Knowledge Base article #1186. |
| 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 | See Websense Knowledge Base article #1115. |
| DC Agent + eDirectory Agent | No | No | Websense does not support communication with both Windows and Novell directory services in the same deployment. (However, you can have both Agents installed, with only one 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. |
| RADIUS Agent + Logon Agent | Yes | Yes | Configure all agents to communicate with Filtering Service.  See Websense Knowledge Base article #1115. |
| eDirectory Agent + Logon Agent | No | No | Websense does not support communication with both Windows and Novell directory services in the same deployment. However, you can have both agents installed, with only one active agent. |

| Combination | Same machine? | Same network? | Configuration required |
|---|---|---|---|
| RADIUS Agent + eDirectory Agent | Yes | Yes | Configure both agents to communicate with Filtering Service. |
| DC Agent + Logon Agent + RADIUS Agent | Yes | Yes | Though this combination is rarely required, it is functionally supported.<br>Configure all agents to communicate with Filtering Service.  See Websense Knowledge Base article #1115. |

This paper does not cover detailed installation instructions for installing the Websense transparent identification agents. Please see your installation guide for installation instructions. The *Websense Enterprise Administrator's Guide* includes instructions for implementing both transparent identification and manual authentication in Websense.

Instructions for configuring an integration partner product to handle user identification are provided in your installation guide. Note that not all integration partner products can be configured to provide user names to Websense.

# Agent Configuration Settings in v6.1

The transparent identification agent configuration settings in Websense Manager are global, and apply to all instances of the agent you have installed. Some configuration settings are *also* stored in the .ini file for that agent. These settings are marked with an asterisk (*) in Websense Manager.

If you have multiple instances of DC Agent, eDirectory Agent or RADIUS Agent, you can configure one instance independently of the others, if need be. Unique settings specified in the .ini file for a particular agent instance override comparable global settings in Websense Manager. For complete instructions on configuring an agent instance differently, please see the *User Identification* chapter of the *Websense Enterprise Administrator's Guide*.

See *Appendix A Transparent Identification Configuration Settings* for definitions of agent configuration settings in Websense Manager and in .ini files.

✓ **NOTE**

By default, Websense encodes LDAP directory server configuration information based on the character set specified under **Character Set** in the **Advanced Directory Settings** dialog box. In Websense Enterprise v6.1, UTF-8 is the default character set. You can change this to MBCS, if need be.

# Remote Filtering

Remote Filtering, new in Websense Enterprise v6.1, enables Websense to filter clients outside a network firewall. Filtering of remote clients works similarly to internal filtering. However, Websense Enteprise v6.1 only filters HTTP

Transparent identification of remote users is not supported in Websense Enterprise v6.1. The Websense transparent identification agents identify users based on IP addresses, whereas Remote Filtering Client associates user names with Media Access Control (MAC) addresses. Since the transparent identification agents do not recognize MAC addresses, an alternative method of user identification is required.

The Websense Remote Filtering Server identifies users based on user names and the MAC addresses of the machines where those users log on. In this way, remote users are identified so user- or group-specific policies can be applied. However, remote users *must* log on to cached domains in your network in order to be identified and filtered. These users also must be identified in your directory service, added to Websense as directory objects.

For remote clients, Websense uses the last portion of the Media Access Control (MAC) address to recognize users. It is possible that the last quadrant of a MAC address overlaps with another IP address. In this case, Websense applies the policy assigned to that IP address.

In Websense Enterprise v6.1, manual authentication of remote users is not supported. If remote users cannot be identified, or there is no policy assigned to a corresponding user or group object in your directory service, the remote user is filtered by the Global policy.

## Deployment of Remote Filtering Server

The Websense Remote Filtering Server resides on a machine within the outermost firewall, but in the DMZ outside the firewall protecting the internal network. The Websense Remote Filtering Client runs on client machines 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 your installation guide for instructions on installing Remote Filtering Server and Remote Filtering Client.

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

In most cases, you only need one Filtering Service that communicates with every instance of Remote Filtering Server. Filtering Service and Remote Filtering Server must be installed on separate machines. If you have installed multiple Filtering Services for load-balancing purposes, each Filtering Service must be able to communicate with every Remote Filtering Server.

**IMPORTANT**

Be sure to follow the recommendations in the *Websense Enterprise Deployment Guide* when setting up Remote Filtering.

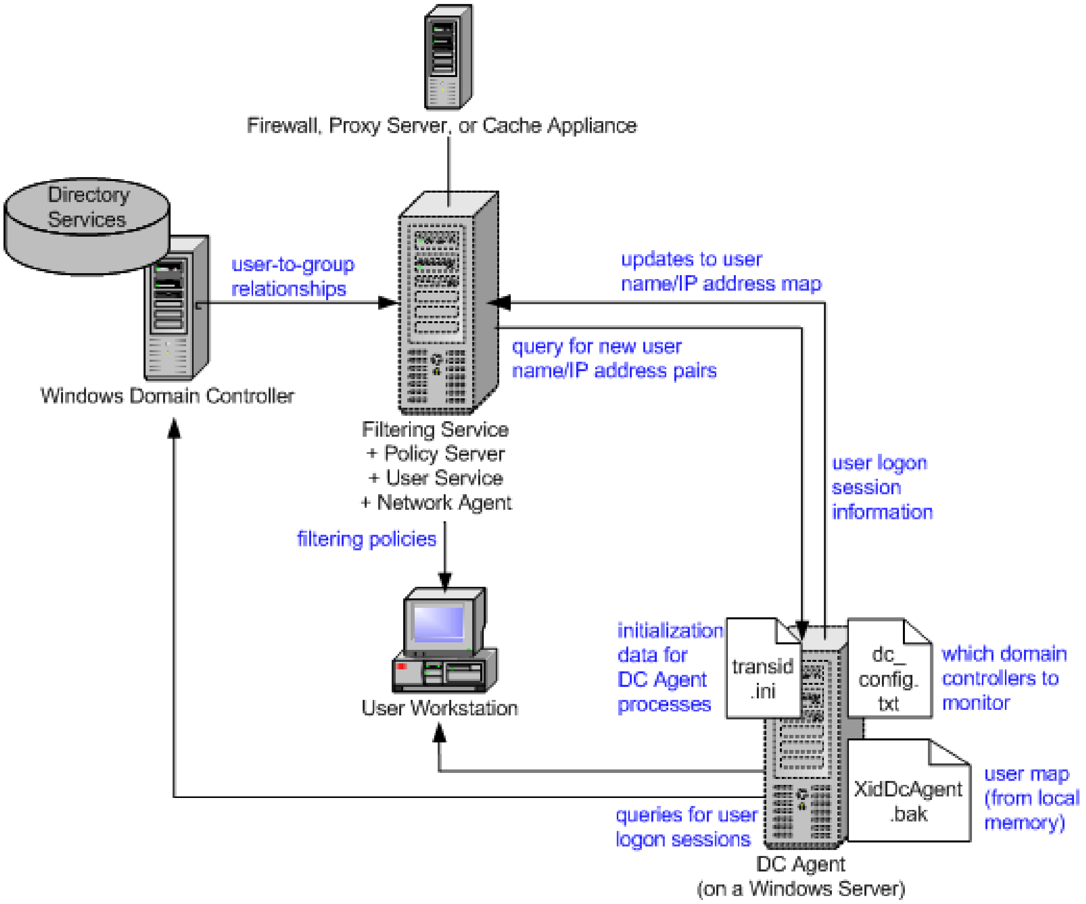
CHAPTER 2

# The Websense DC Agent

The Websense DC Agent is the backbone of the transparent user identification process with a Windows-based directory service. DC Agent allows Websense to identify users in a Windows-based directory service, without prompting users to manually authenticate. DC Agent communicates with the Websense User Service and Websense Filtering Service to provide up-to-date user logon session information to Websense for use in filtering.

## The DC Agent User Identification Process

The process by which DC Agent and User Service gather network user data and send it to the Websense Filtering Service is described here. There are several variables that determine the speed of data transmission, like the size of your network, or the amount of existing network traffic. The *Components* section on page 9 provides further information about each component involved in the process, and about how data is transmitted between components.

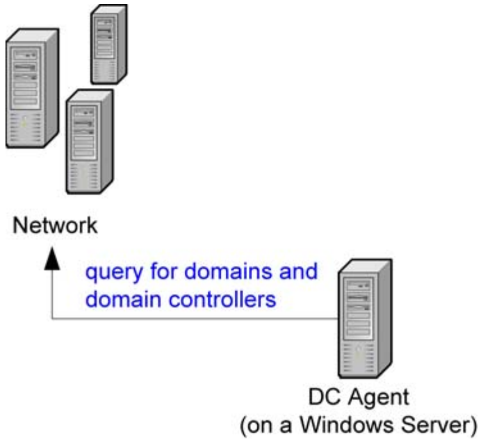


Firewall, Proxy Server, or Cache Appliance
Directory Services
Windows Domain Controller
user-to-group relationships
updates to user name/IP address map
query for new user name/IP address pairs
Filtering Service + Policy Server + User Service + Network Agent
user logon session information
filtering policies
User Workstation
initialization data for DC Agent processes
transid .ini
dc_ config. txt
which domain controllers to monitor
XidDcAgent .bak
user map (from local memory)
queries for user logon sessions
DC Agent (on a Windows Server)

*Components of transparent identification process with DC Agent*

1. **DC Agent detects domain controllers:** Upon startup, and every 24 hours thereafter by default, DC Agent identifies available domains and domain controllers in the network. DC Agent writes domain information to a file called **dc_config.txt**, and sends this information to User Service and Filtering Service.

   See the `UseUserService` parameter in *Appendix A Transparent Identification Configuration Settings* for details on how DC Agent communicates with domain controllers.

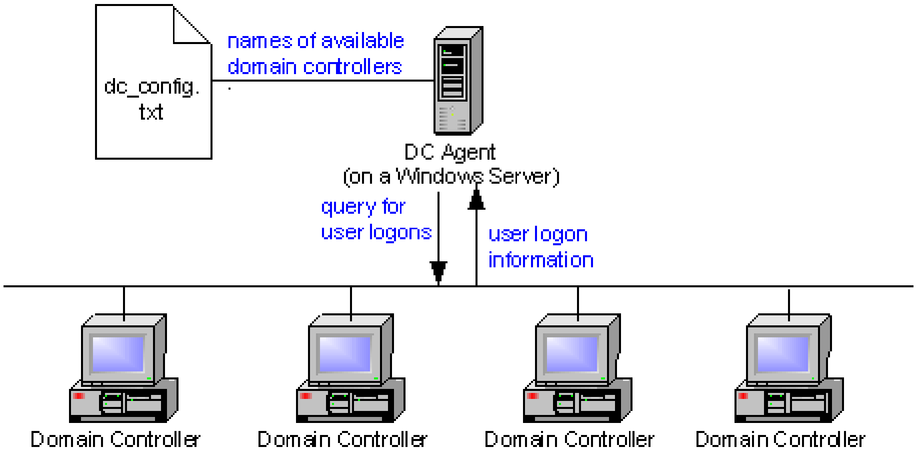


*DC Agent gets domain controller information*

2. **DC Agent obtains logon session information:** DC Agent queries each domain controller for user logon sessions every 10 seconds by default, obtaining the user name and workstation name for each logon session. This query interval is configurable via Websense Manager: Choose **Server > Settings**, and go to the **DC Agent** tab. The **Query Interval** setting is under **Domain Controller Settings**.

**NOTE**

DC Agent only identifies user logon sessions initiated while it is running. If DC Agent is not running at the time a user logs on to a domain controller, it cannot record that logon session. In this case, the user may be filtered by the workstation or network policy (if one exists), or by the default **Global** policy. See page 57 for more information on troubleshooting filtering problems.



*DC Agent gets logon session information from domain controllers*

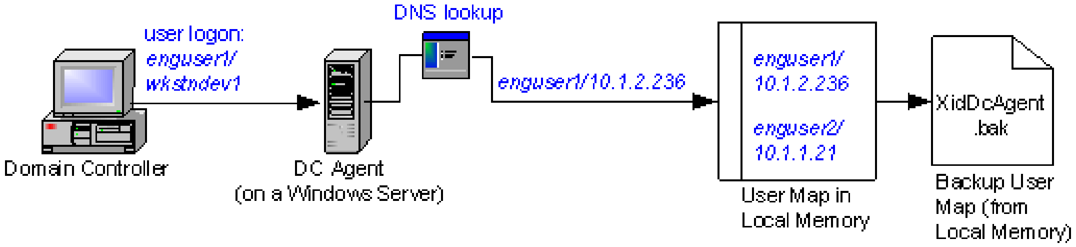
By default, DC Agent also queries workstations. (This query can be optionally disabled using Websense Manager: Choose **Server > Settings**, go to the **DC Agent** tab, and uncheck **Enable workstation polling**.) The first workstation query occurs if a user makes an internet request from a workstation before the next domain controller query occurs. After 15 minutes, DC Agent re-queries the workstation to verify that its user is still active. If no internet request is made between domain controller queries, DC Agent continues only querying domain controllers. The resulting user name/IP address pair is stored in DC Agent's user map, independently of entries resulting from domain controller query. Thereafter, DC Agent queries the workstation every 15 minutes by default.

1. **DC Agent may get user information directly from a workstation:** If DC Agent sees multiple logons to a workstation, DC Agent contacts the workstation directly to obtain the logon information. To determine which user is truly logged on, DC Agent first checks the registry key **WinLogon**. If it can't read this key, DC Agent reads the user's Security Identifier (SID) from the registry. DC Agent sends the SID to the appropriate domain controller, which returns the corresponding logon information.

✓ **NOTE**

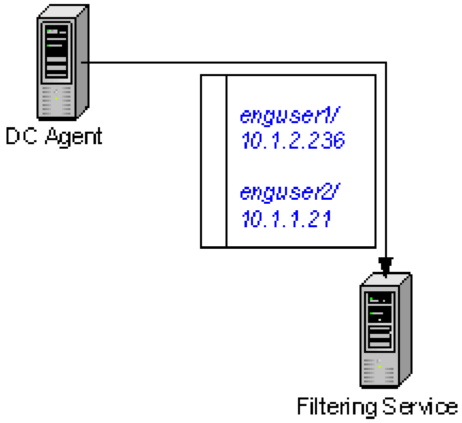
If the DC Agent setting **Enable workstation polling** in Websense Manager is disabled, DC Agent does not contact workstations in this manner.

2. **DC Agent records user name/IP address pairs:** For each logon session identified, DC Agent performs DNS lookup to resolve the workstation name to an IP address, and records the resulting user name/IP address pair to a user name-to-IP-address "map" in local memory. Data from this map is also periodically written to the **XidDCAgent.journal** file, on the DC Agent machine's hard disk. When **XidDcAgent.journal** reaches 1MB in size, **XidDcAgent.journal** is cleared, and the backup file **XidDcAgent.bak** is created. Thereafter, DC Agent continues to write new entries to **XidDcAgent.journal**, clearing the contents and writing to **XidDcAgent.bak** each time **XidDcAgent.journal** reaches 1MB.



*DC Agent saves user name/IP address pairs to its local map*

3. **Filtering Service gets user information:** DC Agent sends user names and IP addresses to Filtering Service each time its user map is updated. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. Note that there are no security risks in this data transfer, as no confidential information (such as user passwords) is transmitted.



*DC Agent sends user name/IP address pairs to Filtering Service*

4. DC Agent sends only new user name/IP address pairs recorded since the last query. At pre-defined intervals, DC Agent attempts to validate that users are still logged on to specific workstations. The DC Agent settings **User Map Verification Interval** and **User Entry Timeout** determine when DC Agent performs user validation, and how user names are expired from the map. To access these settings in Websense Manager, choose **Server > Settings**, and go to the **DC Agent** tab.

✓ **NOTE**

If you have configured DC Agent to require authentication, the DC Agent service checks the password provided by Filtering Service against the password you specified via the **User Identification** settings in Websense Manager. See page *Authenticated Connections*, page 14 for information about authentication between DC Agent and Filtering Service.

5. **Filtering Service gets group information for users logged on:** 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. Directory objects (users and groups) are then made available to Websense Manager, which allows configuration of filtering policies based on those users and groups.
6. If you have configured filtering policies based on directory objects, Filtering Service uses this configuration in determining which filtering policies to apply to users making internet requests. Filtering Service does not determine the filtering policy every time an internet request is made; policy determinations are cached for three hours by the server. For more information about filtering based on directory object policies, please see the *User Identification* and *Filtering Basics* chapters in the *Websense Enterprise Administrator's Guide.*
7. **Websense filters users making internet requests:** Filtering Service uses the policy configuration established via Websense Manager (and saved by Policy Server) to apply filtering policies to logged-on users.

# Components

These descriptions will help you understand the role of each piece involved in transparent identification with the Websense DC Agent.

## *DC Agent*

In Websense v4.4, DC Agent (Domain Controller Agent) was the backbone of transparent user identification. In Websense Enterprise v5.x, DC Agent still plays a central role, but also works together with User Service to provide user logon information to the Websense servers. The DC Agent program is installed on a Windows 2000/2003 Server machine, and runs as a Windows service. DC Agent can be installed on one machine, and can "discover" domains outside of its own domain. Multiple DC Agents can also be used; this may benefit larger networks. For details, see *Deployment of DC Agent and Related Components*, page 13.

DC Agent identifies available domains and domain controllers in the network, and then monitors the domain controllers and associated client machines (workstations) for user logon sessions. Filtering Service uses the information provided by DC Agent to apply internet filtering policies to users logged on to the network.

✓ **NETBIOS AND DOMAIN DISCOVERY**

In order 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 and/or subnets, then DC Agent cannot communicate with those domains or subnets. This can occasionally 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.

There is an option to disable NetBIOS usage, if you do not want to enable port 139. See the UseNetBIOS description under *Initialization Parameters*, page 77 for details.

A program called **XidDcAgent.exe** is installed by default on the DC Agent machine, to the directory **\Websense\bin\**. This program runs as a Windows service, and initiates the processes that enable DC Agent to identify domains and monitor logon sessions. DC Agent stores domain information to the hard disk of the server where it is installed, in a file called **dc_config.txt**. New domain information is recorded to **dc_config.txt** upon 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 should monitor the domain controllers listed

Domain controller monitoring is determined by a value of on (default value) or off corresponding to each domain controller entry. Typical entries look like the following:

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

✓ **NOTE**

If you change the monitoring behavior setting for any domain controller, you must restart the DC Agent service in order for changes to take effect. For information about changing this setting, see page 13.

You can change the domain detection interval via a setting in the **transid.ini** file, as described under *Domain Discovery* on page 15. (A synopsis of each parameter in the **transid.ini** file is provided in *Appendix A Transparent Identification Configuration Settings* .)

DC Agent obtains logon session information from the domain controllers it monitors and resolves client workstation names to IP addresses. The Agent stores the resulting user name/IP address pairs in a user map in local memory, and in the file **XidDcAgent.journal**. 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 a network domain via multiple workstations.

DC Agent also writes its user-name-to-IP address map to a file called **XidDcAgent.bak**, stored on the local disk. Each time **XidDcAgent.journal** reaches 1MB in size, DC Agent backs up the contents to **XidDcAgent.bak**, on its hard disk. The process by which DC Agent identifies and records network domain and user information is detailed on page 5.

## *User Service*

User Service works together with DC Agent to provide an up-to-date list of domains in the network and users in each domain. User Service relays user logon information from DC Agent to Filtering Service. User Service also interacts with your directory service to get group information corresponding to logged-on users.

User Service mediates communication of user logon information between Filtering Service and DC Agent. User Service can run on Windows, Linux, or Solaris. In Solaris or Linux environments, you can still filter based on user or group policies, and take advantage of the transparent identification feature. User Service translates logon session data provided by DC Agent so the Websense Filtering Service can apply the appropriate internet filtering policies.

## *Filtering Service*

Filtering Service can be installed on a server running Windows, Linux or Solaris. (See your installation guide for operating system requirements.) This does not necessarily need to be the same machine where DC Agent is installed, but the two machines must be able to communicate. This also enables Filtering Service to communicate with a Windows directory service in your network, and consequently filter users based on polices assigned to groups in the directory service.

Filtering Service receives user logon session information from DC Agent as users log on to the network. User data is transmitted to Filtering Service 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, the server matches the address with the corresponding user name provided by DC Agent, allowing users to be identified transparently. Filtering Service then filters users according to policies assigned to those users or groups.

Websense can be configured to prompt users to manually authenticate if it cannot obtain user information via DC Agent and User Service. (This can occur if a user logs on to a workstation instead of a domain, for example.) With manual authentication, if a user does not provide a valid user name and password, he/she is blocked from internet access entirely.

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

✓ **NOTE**

It is crucial that Filtering Service, User Service and DC Agent can communicate successfully; problems with communication between components can cause user identification errors. See *DC Agent*, page 57 for information about common problems and related solutions.

## *User Workstations*

Because DC Agent is installed in a Windows domain, users must be able to log on to a Windows domain via their workstations. Workstations do not necessarily need to be running Windows. See your installation guide for a list of operating systems supported by Websense.

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

DC Agent obtains workstation and logon session information indirectly, via the domain controller managing those workstations. For each logon session detected by a domain controller, DC Agent performs DNS lookup by default to convert the user's workstation name into an IP address, and stores the IP address in its user map. The settings `User Map Verification Interval` and `User Entry Timeout` determine how user names are expired from the user map. To access these settings in Websense Manager, choose **Server** > **Settings**, and go to the **DC Agent** tab.

You can optionally set DC Agent to use NetBIOS to get IP Addresses, if DNS lookup fails to identify a workstation. See the `UseNetBIOS` description in *Appendix A Transparent Identification Configuration Settings* for details.

## *Files Used in Transparent Identification with DC Agent*

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

| Filename | Location and Purpose | Functionality |
|---|---|---|
| **XidDcAgent.exe** | \Websense\bin\ (on DC Agent machine; runs as a Windows service named **Websense DCAgent**) | Is the core of the Websense DC Agent.<br>By default, automatically discovers domains at startup and at 24-hour intervals.<br>Sends new entries to User Service and Filtering Service, when queried.<br>Allows communication of transparent identification configuration from Websense Manager to DC Agent.<br>Uses port 30600 by default. |
| **transid.ini** | \Websense\bin\ (on DC Agent machine) | Contains initialization parameters for XidDcAgent.exe, which performs DC Agent processes. See page 73 for a description of each parameter.<br>*Implementation*, page 13 describes situations where you may want to edit this file. |
| **dc_config.txt** | \Websense\bin\ (on DC Agent machine) | Contains information regarding which domains and domain controllers DC Agent should monitor.<br>The list is updated at the interval determined by the `DiscoverInterval` parameter (see *Appendix A Transparent Identification Configuration Settings* ). |
| **XidDcAgent.journal** | \Websense\bin\<br>(on DC Agent machine) | Houses DC Agent's user name-to-IP address map (also in local memory).<br>Updated when logon information is added to or removed from the Agent's map. Read upon startup. |
| **XidDcAgent.bak** | \Websense\bin\<br>(on DC Agent machine) | Houses backup copy (on hard disk) of DC Agent's user name-to-IP address map.<br>Updated whenever **XidDcAgent.journal** exceeds 1MB. Read upon startup. |
| **ignore.txt** | \Websense\bin\<br>(on DC Agent machine) | Contains list of user names for DC Agent to ignore, and the workstations on which to ignore those names (see *Configuring DC Agent to Ignore Particular User Names*, page 15). |

# Implementation

Deployment of the transparent identification feature involves some important installation and setup considerations, discussed in this section. In addition to the standard transparent identification setup procedure, there are configuration options for increasing security and performance of the user identification process. These options are also described here.

For detailed instructions on implementing transparent identification in Websense, please refer to the *User Identification* chapter in the *Websense Enterprise Administrator's Guide*. To summarize those instructions, the following is required to enable transparent identification of users:

- Install DC Agent and User Service. DC Agent can be installed during Websense Enterprise setup, either with other Websense components, or alone.
- Configure User Service to communicate with DC Agent via Websense Manager.
- Add the directory objects you want to filter individually via Websense Manager.

> ✓ **NOTE**
>
> If you are using DC Agent only to perform internet usage *logging*, you do not need to add directory objects (users, groups, workstations or networks) to be filtered.

- If you want Websense to prompt users to authenticate if it is not able to obtain user information from DC Agent and User Service, see *Manual Authentication* in the *Websense Enterprise Administrator's Guide*.

## Deployment of DC Agent and Related Components

With Websense versions 4.4 and later, DC Agent only needs to be installed on one Windows Server machine in the network. However, 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 will have ample space for files that are continually populated with user information, and the user identification process will be faster.

In most cases, you only need one 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 one instance of User Service for each instance of Policy Server.

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

**Average Daily Bytes Tranferred by Number of Users**

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

For large networks, it also may be beneficial to balance the network load supported by DC Agent. To do this:

1. Install DC Agent on multiple Windows servers, and configure each DC Agent to monitor only a portion of the network.
2. Edit the **dc_config.txt** file on each DC Agent machine to set a given DC Agent to monitor only one domain, or only one or a few domain controllers in a given domain.
3. To set each DC Agent to monitor only a portion of the overall network traffic, open each **dc_config.txt** file in a text editor, and edit the settings so that only the domain controllers to be monitored by that DC Agent are set to on. For example, the entries telling DC Agent which domain controllers to monitor in Domain2 could look like the following:

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

   where `[Domain2]` is a domain name, and DCN is a domain controller name.

   See *Files Used in Transparent Identification with DC Agent*, page 12 for further details about **dc_config.txt** and other files used in the transparent identification process.

## Authenticated Connections

You can configure DC Agent and User Service to require an authenticated connection before data is transmitted between them. If you elect to use an authenticated connection, DC Agent requires authentication from User Service.

When DC Agent is configured to require authentication from User Service, DC Agent validates User Service against a password you specify, before it sends data to User Service.

After installing DC Agent, do the following to activate an authenticated connection between DC Agent and User Service:

1. Choose **Server > Settings**. The **Settings** dialog box appears.
2. Select **User Identification** at the left. Installed agents appear in a list under Identify users with these transparent identification agents.
3. Select the DC Agent you installed, and then click **Edit**. The **Edit Transparent Identification Agent** dialog box appears.
4. Check **Enable Authentication**.
5. Specify a password for the authenticated connection. The password is case-sensitive.
6. Click **OK**.
7. Go to the **DC Agent** tab.
8. Under **DC Agent Settings**, ensure that **Enable Authentication** is checked, and that the **Password** matches the one you just entered in the **Edit Transparent Identification Agent** dialog box.
9. Click **OK** to close the **Settings** dialog box.
10. Click **Done** to save your changes.

Further details on configuring authenticated connections are included in the *Websense Enterprise Administrator's Guide*, under *Installing the Websense DC Agent*.

## Domain Discovery

By default, DC Agent performs auto-discovery of domains in its subnet every 24 hours, as described under *Components*, page 9. If desired, you can change the domain detection interval. To modify the domain discovery behavior of DC Agent:

1. On the DC Agent machine, go to the Websense\bin\ directory, and open the file **transid.ini** in a text editor.
2. To change the domain detection interval, modify the line:

   ```
   DiscoverInterval=[N]
   ```

   where N is the desired time interval, in seconds. The default value is 86400 (or 24 hours). This value must be greater than 3600 seconds (or 1 hour).
3. To disable domain detection altogether, set `DiscoverInterval` equal to 0.
4. After making the desired changes, restart the Websense DC Agent service in order for changes to take effect.

See *Appendix A Transparent Identification Configuration Settings* for further details about the `DiscoverInterval` parameter.

## Configuring DC Agent to Ignore Particular User Names

Some Windows services contact domain controllers from user workstations, causing the original users logged on from those workstations to be misidentified. Windows XP, for example, automatically sets up two internal user names (Local Service and Network Service) for various internal processes to use for communication with domain controllers. Running Systems Management Server (SMS) on a workstation can also cause confusion with user identification.

A sample scenario follows: While **domainA/user1** is logged on to the network and is being filtered by a Websense policy assigned to **user1**, a service starts up on that user's machine. The service assumes the identity **domainA/ServiceName** in order to contact the domain controller. This can cause filtering problems, because Websense interprets **domainA/ServiceName** as a new logon session, separate from the one established by **user1**. Because there is no specific policy assigned to the user **ServiceName**, Websense begins filtering this user according to the workstation or network policy (if assigned), or by the Global policy.

Similar behavior occurs when a standard Windows 2000 service contacts a domain controller with a user name made up of the workstation name followed by a dollar sign (wkstn$). DC Agent interprets the service as a new user, for which no policy has been assigned. A wkstn$ user name can also result from a shared resource on the user’s workstation, such as a drive mapped to a domain controller. By default, Websense v5.x is configured to ignore user names containing dollar signs ($), via the IgnoreDollarSign initialization parameter (see *Appendix A Transparent Identification Configuration Settings* ).

To prevent or correct these problems, configure DC Agent to ignore logon names that are not associated with actual users, as follows.

1. Stop the DC Agent service, using the **Services** applet accessible via **Administrative Tools** (*Windows 2000/2003*).
2. In the DC Agent installation directory, open or create a file named **ignore.txt** in a text editor.

> ✓ **NOTE**
>
> You can optionally set a size limit for the ignore list, using the MaxIgnoreListSize initialization parameter. See *Appendix A Transparent Identification Configuration Settings* for details.

3. Add entries to this file according to your needs. There are three types of entries:
4. **User names:** Type a one-line entry for each user name to be ignored. Websense will ignore user names listed this way, regardless of the associated workstation.
5. **User name/workstation pairs**: Type two words on one line, separated by a comma, for each user name/workstation pair to be ignored. Websense will ignore a user name only if it comes from the corresponding workstation.
6. **Workstation names**: Type an asterisk (*), separated by a comma, followed by the workstation name. The workstation name can be the machine name, the IP address, or a range of IP addresses.

   In the following example:

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

   - The user name **johnsmith** will be ignored for ALL workstations.
   - The user name **admin** will only be ignored if it came from workstation **WKSTA-NAME**.
   - The workstations **WKSTA-NAME** and **10.209.34.56** will be ignored for all logons.
   - The range of IP addresses from **10.203.34.1** to **10.203.34.255** will be ignored for all logons.
7. Save and close the file when all entries are complete.
8. Restart the DC Agent service to activate the changes.

DC Agent will ignore the specified user names.

CHAPTER 3

# The Websense Logon Agent

The Websense Logon Agent provides maximal accuracy in identifying users as they log on to the network via Windows client machines. Logon Agent's associated logon application captures logon sessions as users log on to Windows domains in your network. Logon Agent communicates with the Websense User Service to provide up-to-date user logon session information to Websense for use in filtering.

Whereas DC Agent identifies users by periodically querying domain controllers and workstations, Logon Agent identifies users in a real-time manner, as they log on to domains. This enables the Websense Filtering Service to accurately filter internet access based on policies assigned to particular users or groups.

The logon application (client) portion of Logon Agent is called LogonApp.exe. LogonApp.exe is activated via a logon script. The script is most easily stored in a text file with a .bat or .cmd extension in the directory where LogonApp.exe resides.

There are sample procedures for configuring a logon script, depending on the directory service in your network. Please see your installation guide for logon script configuration instructions for Windows NT or Active Directory environments.

## The Logon Agent User Identification Process

Once the Websense Logon Agent service is installed on a server machine, and the logon application is deployed successfully via logon script or other means, the Agent works to detect users as they log on to your network.

The identification process with Logon Agent is unique in that it relies on self-identification of the client machine. Typically, a logon script in a shared network location evokes a process on client machines called LogonApp.exe. This logon application provides user logon information to the Logon Agent (XID Authentication Server) service.

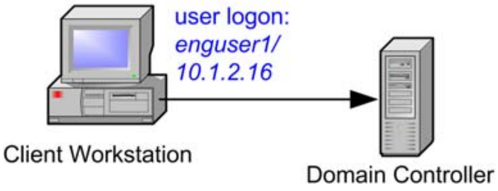
LogonApp.exe has two operation modes: persistent mode and non-persistent mode. When the application is invoked from a command, the [PERSIST/NOPERSIST] parameter determines which mode is used. See *Logon Agent*, page 19 for more details about how these modes work.

- *Persistent mode:* LogonApp.exe runs as a background task on the client machine, and periodically sends user name/password pairs to Logon Agent. This interval is determined by the **Query Interval (persistent mode)** setting in Websense Manager. If a logout script is configured to register when the user logs out, LogonApp.exe sends the logout information to the Agent at that time.
- *Non-persistent mode:* LogonApp.exe contacts Logon Agent once when a user logs on. The user logon session is stored in Logon Agent's local memory. Logon Agent's resulting user map is subject to the **Entry Lifetime (non-persistent mode)** setting in Websense Manager.

See *Appendix A Transparent Identification Configuration Settings* for configuration setting descriptions.

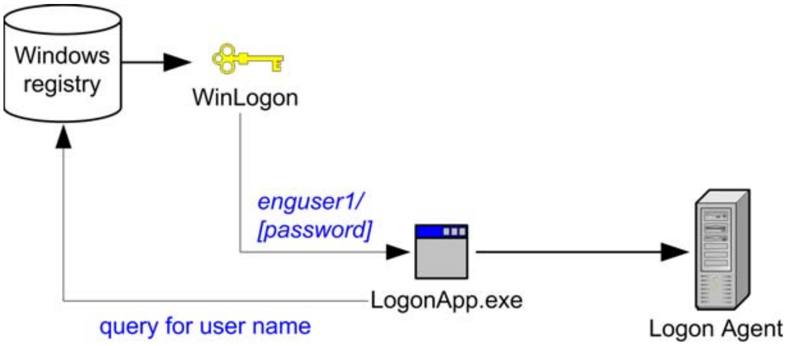
The user identification process works as follows.

1. **Users log on to the network:** A user logs on to a network domain via a Windows client machine.



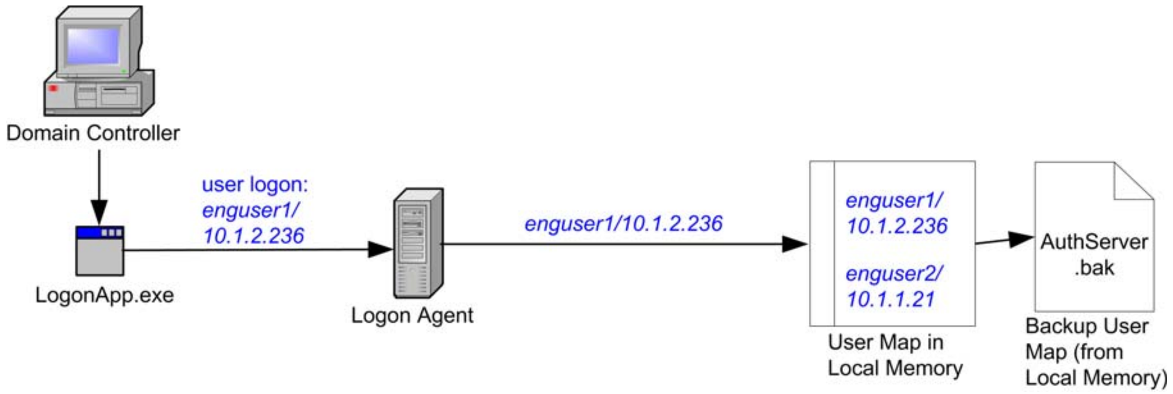
*User logs on*

2. **LogonApp.exe starts up:** A network logon script stored in a shared network location evokes the Websense logon application (LogonApp.exe).
3. **LogonApp.exe contacts Logon Agent:** LogonApp.exe connects to Logon Agent (XID Authentication Server service) via HTTP.
4. **Logon Agent gets user names and IP addresses:** When Logon Agent sends an NTLM authentication challenge, LogonApp.exe extracts the user name from the Windows registry on the client machine, and sends the user name, hashed password, and client machine IP address to Logon Agent.



*Logon Agent gets user information*

5. **Logon Agent verifies user logons:** Logon Agent verifies the user name/password combination from LogonApp.exe by establishing a session with the domain controller. (Logon Agent contacts the Websense User Service to determine which domain controller is the logon source.)
6. **Logon Agent records user name/IP address pairs:** After the user is authenticated and verified via NTLM, Logon Agent stores the user name/IP address pair in its user map in local memory, and in a file named **AuthServer.journal**. When this file reaches 1MB in size, Logon Agent creates the backup file **AuthServer.bak**. Thereafter, Logon Agent continues to write new entries to **AuthServer.journal**, clearing the contents and writing to **AuthServer.bak** each time **AuthServer.journal** reaches 1MB.



*Logon Agent records user name/IP address pairs*

7. **Filtering Service gets user information:** Logon Agent sends user names and IP addresses to Filtering Service and User Service each time its user map is updated. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. Note that there are no security risks in this data transfer, as no confidential information (such as user passwords) is transmitted.

## Components

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

### *Logon Agent*

The Logon Agent service (the Websense XID Authentication Server) is optionally installed with Websense Enterprise. Logon Agent can get user logon sessions from all Windows domains in your network, even if installed on just one machine. Multiple Logon Agents can also be used; this may benefit larger networks. For details, see *Logon Agent can be used with a Windows NTLM-based directory service or with Active Directory, which is LDAP-based. However, the associated logon application runs only on Windows-based client machines. The Agent runs on Windows, Linux, or Solaris, and works together with the Websense User Service and Websense Filtering Service.*, page 22.

Logon Agent authenticates users as they log on to domains in the network, or as LogonApp.exe sends user information to the Agent. (This depends on the mode of operation; see below.) Filtering Service uses the information provided by Logon Agent to apply internet filtering policies to logged-on users.

Logon Agent obtains logon session information from its associated logon application, LogonApp.exe, and stores the resulting user name/IP address pairs in a “user map” in local memory, and in the file **AuthServer.journal**. 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 a network domain via multiple workstations. Each time **AuthServer.journal** reaches 1MB in size, Logon Agent backs up the contents to **AuthServer.bak**, on its hard disk.

The Logon Agent settings **Query Interval (persistent mode)** and **Entry Lifetime (non-persistent mode)** determine how long a user entry (user name/IP address pair) remains in Logon Agent’s user map before expiring. The timeout behavior depends on whether Logon Agent is running in persistent (default) or non-persistent mode (see page 22). To access these settings, choose **Server** > **Settings**, select **User Identification**, and go to the **Logon Agent** tab. See *Appendix A Transparent Identification Configuration Settings* for configuration setting descriptions.

When LogonApp.exe is running in persistent mode, it sends user logon information to Logon Agent at the specified interval (every 15 minutes by default). The **Query Interval (persistent mode)** setting controls this interval. If a logout script is configured to record when users log out, LogonApp.exe also sends user logout information to Logon Agent, which removes corresponding user entries from its user map at that time.

When LogonApp.exe is running in non-persistent mode, logon information is sent to the Agent only once for each logon. The **Entry Lifetime (non-persistent mode)** setting is configured via Websense Manager, in the **Settings** dialog box.

## *LogonApp.exe*

Logon Agent works together with its associated application, LogonApp.exe. LogonApp.exe runs on Windows client machines, and sends user logon sessions to the Agent for authentication. LogonApp.exe sends data either when logon sessions first occur, or as they occur, depending on the application's operation mode. See page 17 for descriptions of these operation modes.

LogonApp.exe provides user logon session information to Logon Agent. When it is running in persistent mode, LogonApp.exe sends logon information to the Agent at a specified interval, determined by the **Query Interval (persistent mode)** configuration setting. **Query Interval (persistent mode)** is configured via the **Settings** dialog box in Websense Manager. See page 75 for information about this configuration setting.

## *User Service*

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

You can filter based on user or group policies and take advantage of the transparent identification feature in Windows, Linux, or Solaris environments. Websense can communicate with a directory service whether it runs on the same operating system as Websense or on a different operating system. Even if you are using a Windows NTLM-based directory service, you can have the Websense User Service running on Windows, Linux, or Solaris.

## *Filtering Service*

Filtering Service translates logon session data provided by Logon Agent so the Websense Policy Server can apply the appropriate internet filtering policies.

Filtering Service can be installed on a machine running Windows Server, Linux or Solaris. (See your installation guide for operating system requirements.) This does not necessarily need to be the same machine where Logon Agent is installed, but the two machines must be able to communicate. This also enables Filtering Service to communicate with a directory service in your network, and consequently filter users based on polices assigned to objects in the directory service.

Filtering Service receives user logon session information from Logon Agent as users log on to domain controllers or workstations. User data is transmitted to Filtering Service in the form of user name/IP address pairs (originating from Logon Agent's map in local memory). When Filtering Service receives the IP address of a machine making an internet request, the server matches the address with the corresponding user name provided by Logon 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.

Websense can be configured to prompt users to manually authenticate if it cannot obtain user information via Logon Agent and User Service. (This can occur if a user logs on to a workstation instead of a domain, for example.) With manual authentication, if a user does not provide a valid user name and password, he/she is blocked from internet access entirely.

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

✓ **NOTE**

It is crucial that Filtering Service, User Service and Logon Agent can communicate successfully; problems with communication between components can cause user identification errors. See *Logon Agent*, page 62 for information about common problems and related solutions.

## *Files Used in Transparent Identification with Logon Agent*

| Filename | Location and Purpose | Functionality |
|---|---|---|
| **AuthServer.exe** | \Websense\bin\ (on Logon Agent machine). When installed, runs as the XID Authentication Server service. | Is the core of the Websense Logon Agent.<br>Sends new entries to User Service and Filtering Service, when queried.<br>Allows communication of transparent identification configuration from Websense Manager to Logon Agent.<br>Uses port 30602 by default. |
| **LogonApp.exe** | Stored in a shared network location (recommended), and activated on client workstations by a logon script | Application associated with Logon Agent that captures user logon sessions as they occur.<br>Runs on Windows client machines. |
| **[logonscript].cmd** | Recommended to store in the same shared network location as LogonApp.exe | Invokes LogonApp.exe, which runs on client machines and captures logon sessions. |
| **AuthServer.journal** | \Websense\bin\ (on Logon Agent machine) | Houses Logon Agent’s user name-to-IP address map (also in local memory).<br>Updated when logon information is added to or removed from the Agent’s map. Read upon startup. |
| **AuthServer.bak** | \Websense\bin\ (on Logon Agent machine) | Houses backup copy of Logon Agent’s user name-to-IP address map on local hard disk.<br>Updated whenever **AuthServer.journal** exceeds 1MB. Read upon startup. |
| **AuthServer.ini** | \Websense\bin\ (on Logon Agent machine) | Contains initialization parameters for Logon Agent processes. See page 75 for a description of each parameter. |

# Implementation

Logon Agent can be used with a Windows NTLM-based directory service or with Active Directory, which is LDAP-based. However, the associated logon application runs only on Windows-based client machines. The Agent runs on Windows, Linux, or Solaris, and works together with the Websense User Service and Websense Filtering Service.

## Deployment of Logon Agent and Related Components

Logon Agent only needs to be installed on one 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 will have ample space for files that are continually populated with user information, and the user identification process will be faster.

In most cases, you only need one Filtering Service that communicates 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 one instance of User Service for each instance of Policy Server.

## Authenticated Connections

You can configure Logon Agent and User Service to require an authenticated connection before data is transmitted between them. If you elect to use an authenticated connection, Logon Agent requires authentication from User Service.

When Logon Agent is configured to require authentication from User Service, Logon Agent validates User Service against a password you specify, before it sends data to User Service.

After installing Logon Agent, do the following to activate an authenticated connection between Logon Agent and User Service:

1. Choose **Server > Settings**. The **Settings** dialog box appears.
2. Select User Identification at the left. Installed agents appear in a list under Identify users with these transparent identification agents.
3. Select the Logon Agent you installed, and then click **Edit**. The **Edit Transparent Identification Agent** dialog box appears.
4. Check Enable Authentication.
5. Specify a password for the authenticated connection. The password is case-sensitive.
6. Click **OK**.
7. Go to the **Logon Agent** tab.
8. Under **Logon Agent Settings**, ensure that **Enable Authentication** is checked, and that the **Password** matches the one you just entered in the **Edit Transparent Identification Agent** dialog box.
9. Click **OK** to close the **Settings** dialog box.
10. Click **Done** to save your changes.

Further details on configuring authenticated connections are included in the *Websense Enterprise Administrator's Guide*.

CHAPTER 4

# The Websense RADIUS Agent

The Webense RADIUS Agent allows you to integrate your Websense filtering policies with authentication provided by a RADIUS server. The Websense RADIUS Agent enables Websense Enterprise 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 can filter those users.

When properly configured, RADIUS Agent captures and processes all 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 user activity.

### RADIUS Authentication

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

The RADIUS authentication process is described in detail in the next section, on page 24.

### RADIUS Accounting

If your RADIUS client supports accounting (or user logon tracking), and accounting is enabled on the client, RADIUS Agent also processes accounting requests. The RADIUS client keeps track of who is logged onto the network via accounting messages transmitted between a RADIUS client and a RADIUS server. “Stop accounting” requests tell the RADIUS server to stop tracking logon activity for a particular user.

In some cases, RADIUS Agent can gather more accurate information via accounting requests. 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.

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

The Websense RADIUS Agent works together with the RADIUS server and RADIUS client 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.

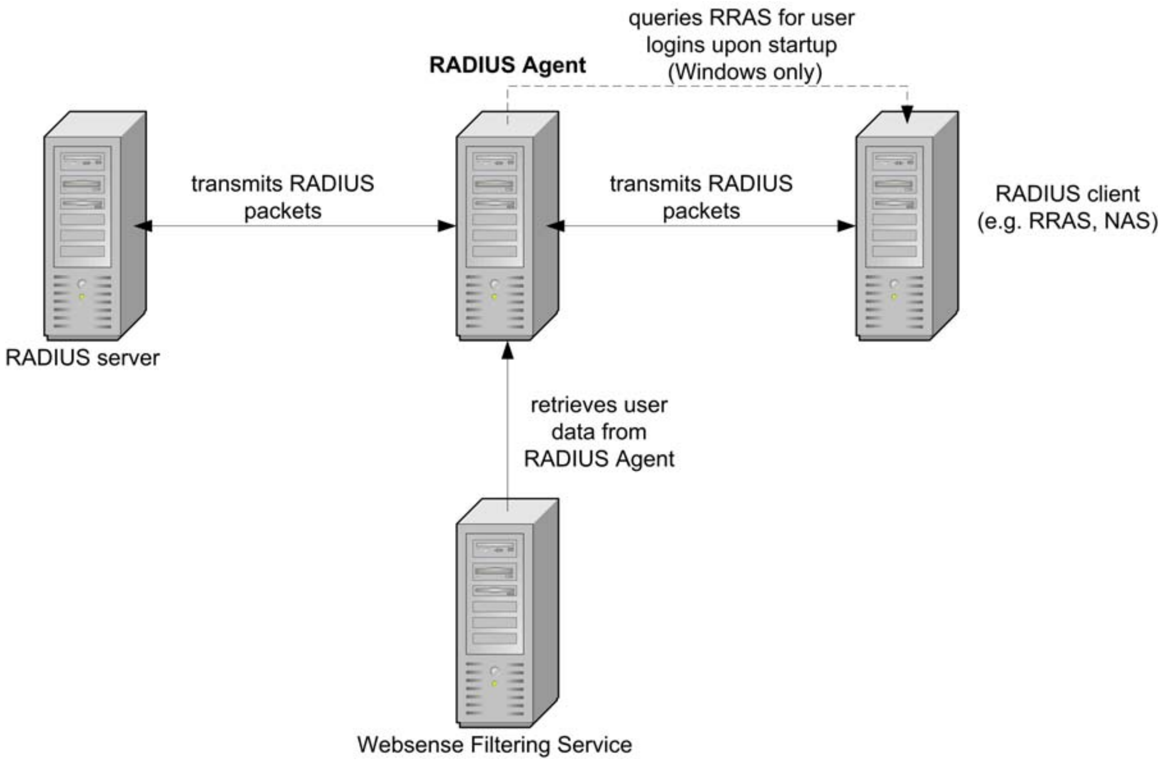
✓ **NOTE**

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* the Websense RADIUS Agent, remote users are authenticated by a RADIUS client – typically, an RAS server, VPN server, or firewall. The authentication process is as follows:

1. A user logs on to the network from a remote workstation.
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 back to the RADIUS client.

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



*Components of transparent identification process with RADIUS Agent*

The transparent identification process is as follows:

1. **RADIUS Agent detects remote user logons:** Once installed and configured appropriately, RADIUS Agent listens on port 1645 (the RADIUS authentication port) for authentication requests. RADIUS Agent detects users logging in to domains, or directly to the RADIUS server in your network.

   ✓ **NOTE**

   If you are using RADIUS authentication against a particular Windows domain in your network, Websense, Inc. recommends running the RADIUS Agent service as a domain user, or as the default System account on a machine belonging to that domain.

2. **The RADIUS client receives authentication requests:** 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 records user name/IP address pairs:** 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 file **wsradius.journal**. After **wsradius.journal** reaches 1 MB in size, this map is cleared, and the data is written to the **wsradius.bak** file, on the RADIUS Agent machine’s hard disk. Thereafter, RADIUS Agent continues to write new entries to **wsradius.journal**, clearing the contents and writing to **wsradius.bak** each time **wsradius.journal** reaches 1MB.

> ✓ **NOTE**
>
> If RADIUS Agent receives a new request having an IP address already included in a user name/IP entry in its map, it *replaces* the existing pairing in its map with the new one.

4. **RADIUS Agent transmits authentication requests:** After it has extracted the needed information, RADIUS Agent forwards the authentication request to the RADIUS server through an alternate port. This port is configurable via Websense Manager and can be any port not already in use.
5. **The RADIUS server authenticates users:** 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**
>
> You can configure the amount of time RADIUS Agent waits for a response from the RADIUS server before ending a query attempt. This time interval is controlled by the `Timeout` parameter in the RADIUS configuration file, installed to **Websense\bin\wsradius.ini** on the RADIUS Agent machine by default.
>
> For more details about parameters in **wsradius.ini**, see *Appendix A Transparent Identification Configuration Settings* .

6. **RADIUS Agent processes the RADIUS message:** RADIUS Agent evaluates the response from the RADIUS server. If the RADIUS message received is an authentication *rejection*, RADIUS Agent gets the ID, and removes the user name/IP address entry corresponding to the request ID from its internal map. If the RADIUS packet received is an authentication *acceptance*, RA gets the ID and copies the corresponding user name/IP address pair from its internal map to its main user map, which is a listing of full domain/user name/IP address entries.
7. **The RADIUS client receives authentication responses:** RADIUS Agent responds back to the RADIUS client (which sent the original authentication request), verifying that the user was authenticated.

8. **Filtering Service gets user information:** RADIUS Agent sends user names and IP addresses to Filtering Service each time its user map is updated. Data is sent over port 30800. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. Note that there are no security risks in this data transfer, as no confidential information (such as user passwords) is transmitted.

> ✓ **NOTE**
>
> If you configured RADIUS Agent during setup to require authentication, the RADIUS Agent service checks the password provided by User Service and Filtering Service against the password you specified during setup or via the **User Identification** settings in Websense Manager. See page 31 for information about authentication between RADIUS Agent and User Service.

9. **Filtering Service gets group information for users logged on:** 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. Directory objects (users and groups) are then made available to Websense Manager, which allows configuration of filtering policies based on those users and groups.
10. If you have configured filtering policies based on directory objects, Filtering Service uses this configuration in determining which filtering policies to apply to users making internet requests. Filtering Service does not determine the filtering policy every time an internet request is made; policy determinations are cached for three hours by the server.
11. **Websense filters users making internet requests:** Filtering Service uses the policy configuration established via Websense Manager to apply filtering policies to logged-on users. Filtering Service does not determine the filtering policy every time an internet request is made; policy determinations are cached for three hours by the server. For more information about filtering based on directory object policies, please see the *User Identification* and *Filtering Basics* chapters of the *Websense Enterprise Administrator's Guide.*

## Components

These descriptions will help you understand the role of each piece involved in transparent identification with the Websense RADIUS Agent.

### *RADIUS Agent*

RADIUS Agent is installed on a Windows 2000 or 2003 Server machine, and runs as a service on Windows, or a daemon on Linux/Solaris. One instance of the Websense RADIUS Agent can support multiple RADIUS clients. Multiple RADIUS Agents can also be used; this may benefit larger networks. For details, see Websense Knowledge Base article number 1186.

By default, RADIUS Agent listens on the RADIUS authentication port for authentication requests corresponding to remote user logons. Filtering Service uses the information provided by RADIUS Agent to apply internet 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 a file named **wsradius.journal**. Data from this map is also periodically written to the **wsradius.bak** file, on the RADIUS Agent machine's hard disk. Each time **wsradius.journal** reaches 1MB in size, RADIUS Agent backs up the contents to **wsradius.bak**.

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

## *User Service*

User Service works together with RADIUS Agent to provide an up-to-date list of users logged on to the local network from remote machines. Both Filtering Service and User Service receive updates on user logon sessions from RADIUS Agent, as logons are recorded to the Agent's user map. User Service also interacts with your directory service to get group information corresponding to logged-on users.

User Service interacts with RADIUS Agent much like Filtering Service does. However, the interaction between RADIUS Agent and Filtering Service is of central importance. Filtering Service is the end destination for the user information RADIUS Agent gleans from authentication requests. When troubleshooting user identification problems, it is most important to determine whether Filtering Service is getting the latest and most accurate user data.

User Service mediates communication of user logon information between Filtering Service and RADIUS Agent. User Service can run on Windows, Linux, or Solaris. In Solaris or Linux environments, you can still filter based on user or group policies, and take advantage of the transparent identification feature. User Service translates logon session data provided by RADIUS Agent so the Websense Policy Server can apply the appropriate internet filtering policies.

## *Filtering Service*

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

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.

Websense 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/she is blocked from internet access entirely.

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

✓ **NOTE**

It is crucial that Filtering Service and RADIUS Agent can communicate successfully. Problems with communication between components can cause user identification errors. See *Common Problems*, page 57 for information about common problems and related solutions.

## *RADIUS Client*

Typically, the RADIUS client is a server such as a Network Access Service (NAS) server, or a 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 under **Authentication Ports** in Websense Manager. (To access this setting, choose **Server > Settings**, and go to the **RADIUS Agent** tab.) The RADIUS client sends accounting requests to the port specified under **Accounting Ports**.

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

**IMPORTANT**

It is crucial that the RADIUS client and RADIUS server be configured to communicate via RADIUS Agent. Please follow the configuration guidelines in the *Websense Enterprise Administrator's Guide*.

## *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 the 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.

## *User Workstations*

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

## 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\ (installed by default on RADIUS Agent machine; runs as a service or daemon named **Websense RADIUS Agent**) | Is the core of the Websense RADIUS Agent.<br>Automatically sends new entries to Filtering Service, when queried.<br>Allows communication of transparent identification configuration from Websense Manager to RADIUS Agent. |
| **wsradius.ini** | \Websense\bin\ (on RADIUS Agent machine) | Contains initialization parameters for RADIUSAgent.exe, which performs RADIUS Agent processes. See page 76 for a description of each parameter. |
| **wsradius.journal** | \Websense\bin\ (on RADIUS Agent machine) | RADIUS Agent's user name-to-IP address map (also in local memory).<br>Updated when logon information is added to or removed from the Agent's map. Read upon startup. |
| **wsradius.bak** | \Websense\bin\ (on RADIUS Agent machine) | Houses backup copy (on hard disk) of RADIUS Agent's user name-to-IP address map.<br>Updated whenever **wsradius.journal** exceeds 1MB. Read upon startup. |
| **ignore.txt** | \Websense\bin\ (on RADIUS Agent machine, if created) | Contains list of user names for RADIUS Agent to ignore, and the workstations on which to ignore those names (see page 32). |

# Implementation

Deployment of the transparent identification feature involves some important installation and setup considerations, discussed in this section. In addition to the standard transparent identification setup procedure, there are configuration options for increasing security and performance of the user identification process. These options are also described here.

For detailed instructions on implementing transparent identification in Websense, please refer to the *User Identification* chapter in the *Websense Enterprise Administrator's Guide*. To summarize, the following steps are required to enable transparent identification of remote users:

- Install RADIUS Agent on the desired machine. (This can be a machine running Windows 2000 Server +SP2, Windows 2003, Linux Red Hat version 8.0 or 9.0, or Solaris version 2.7, 8 or 9 plus patches.)
- RADIUS Agent can be installed as part of the main Websense Enterprise setup. Alternatively, you can install RADIUS Agent separately, by choosing the **Custom** installation option and selecting the RADIUS Agent component.
- Configure Filtering Service to communicate with RADIUS Agent.
- Configure RADIUS Agent to forward authentication requests from client machines to the RADIUS server.
- Configure the RADIUS client to communicate with the Websense RADIUS Agent instead of directly with your RADIUS server. The RADIUS client uses RADIUS Agent as the source of authentication requests.
- Configure the RADIUS server to use the Websense RADIUS Agent as a proxy.
- Add the directory objects you want to filter individually via Websense Manager.
- Please refer to the *Websense Enterprise Administrator's Guide* for instructions on configuring your RADIUS environment.

## Deployment of RADIUS Transparent Identification Components

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

In most cases, you only need one 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.

## Authenticated Connections

You can configure RADIUS Agent and User Service to require an authenticated connection before data is transmitted between them. If you elect to use an authenticated connection, RADIUS Agent requires authentication from User Service.

When RADIUS Agent is configured to require authentication from User Service, RADIUS Agent validates User Service against a password you specify, before it sends data to User Service.

1. During installation of RADIUS Agent, choose to use an **Authenticated Connection** between RADIUS Agent and User Service, and specify a password to be used for the connection.
2. After RADIUS Agent installation is complete, enter the connection password via Websense Manager:

a. Choose **Server** > **Settings**. The **Settings** dialog box appears.

b. Select User Identification at the left. Installed agents appear in a list under Identify users with these transparent identification agents.

c. Select the RADIUS Agent you installed, and then click **Edit**. The **Edit Transparent Identification Agent** dialog box appears.

d. Ensure that **Enable Authentication** is checked.

e. Ensure that the password you specified for the authenticated connection is entered here. The password is case-sensitive.

f. Click **OK**.

g. Go to the **RADIUS Agent** tab.

h. Under **RADIUS Agent Settings**, ensure that **Enable Authentication** is checked, and that the **Password** matches the one you just entered in the **Edit Transparent Identification Agent** dialog box.

i. Click **OK** to close the **Settings** dialog box.

j. Click **Done** to save your changes.

This password is global for all instances of RADIUS Agent. If you want to configure one instance of RADIUS Agent differently, specify a unique password by modifying the password parameter in the **wsradius.ini** file, as shown:

```
password=[PASSWORD]
```

where **[**PASSWORD] is a 4-to-16-character password of your choice. This password is case-sensitive.

## Configuring RADIUS Agent to Ignore Particular User Names

If desired, you can configure RADIUS Agent to ignore logon names that are not associated with actual users. Some Windows 200x and XP services contact domain controllers using the workstation identities of active Websense users in your network.

Windows XP automatically sets up two internal user names (Local Service and Network Service) for various internal processes to use for communication with domain controllers. You can configure Websense to ignore such user names.

Follow these instructions to set up a list of user names, or user name/workstation pairs, for the Websense RADIUS Agent to ignore.

1. Stop the Websense RADIUS Agent service.
2. In the RADIUS Agent installation directory, create a text file named **ignore.txt**.
3. Type a one-line entry for each user name to be ignored. Websense will ignore user names listed this way, regardless of the associated workstation. Do not use wildcard characters, such as “*”.

   In the following example:

   ```
   johnsmith
   aperez, enggroup1
   ```

   The user name **johnsmith** will be ignored for ALL workstations. The user name **aperez** will be ignored only for the domain **enggroup1**.
4. Save and close the file when all entries are complete.
5. Restart the Websense RADIUS Agent service.

RADIUS Agent will ignore the user names you have specified, and Websense will not consider these names in filtering.

# CHAPTER 5 The Websense eDirectory Agent

The Websense eDirectory Agent works together with Novell eDirectory to transparently identify users so Websense can filter them according to particular policies assigned to 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 used can be changed; see *Default Directory Protocol*, page 39 for details.)

The Websense eDirectory Agent associates each authenticated user with an IP address, and records user name-to-IP-address pairings to a "user map." With the help of the Websense User Service, eDirectory Agent supplies this information to the Websense Filtering Service.

✓ **WEBSENSE V5.1 USERS**

With Websense v5.1, use of eDirectory Agent requires a patch. If you are currently a Websense v5.1 customer and have downloaded the Websense eDirectory Agent, this patch is available from http://www.mywebsense.com. If you do not have the patch, please contact Technical Support for assistance.

- **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, one or more attributes with this name. For each successful logon, Novell eDirectory server adds one *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. The 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 the 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.

The eDirectory Agent uses the *networkAddress* property of a user object to associate IP addresses with logged-on users. Since *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.

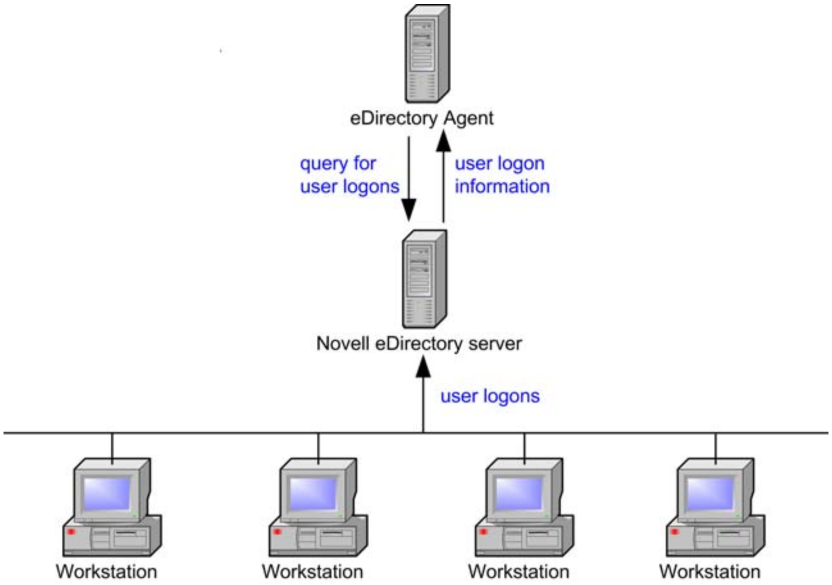
eDirectory Agent must be configured to connect to each machine running an Novell eDirectory replica. This can be done during or after initial installation of eDirectory Agent. See the *Websense Enterprise Administrator's Guide* for details on configuring this communication.

## The eDirectory Agent User Identification Process

The transparent identification process with eDirectory Agent is as follows.

1. **Novell eDirectory authenticates users:** Users logging on to Novell eDirectory are compared to user accounts in the directory service, and authenticated if a match is found.
2. **eDirectory Agent detects user logons:** Once installed and configured appropriately, eDirectory Agent retrieves information from Novell eDirectory about users logging on to the directory service, using Lightweight Directory Access Protocol (LDAP). eDirectory Agent only detects users logging on directly to Novell eDirectory server.

   eDirectory Agent queries Novell eDirectory for user logons at regular intervals (30,000 milliseconds, or 30 seconds, by default). This interval is determined by the `PollInterval` parameter in **wsedir.ini**. See page page 76 for details on initialization parameters.



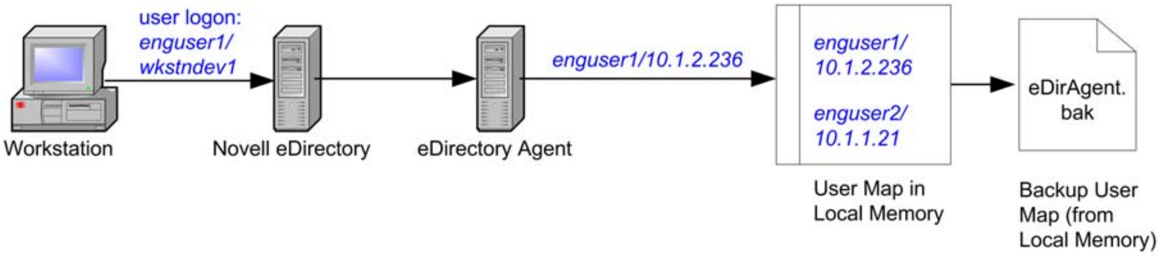
*eDirectory Agent gets user logons from Novell eDirectory*

3. 3.**eDirectory Agent records user name/IP address pairs:** 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 file **eDirAgent.journal**. When **eDirAgent.journal** reaches 1 MB in size, this map is cleared, and data is backed up to the **eDirAgent.bak** file, on the

eDirectory Agent machine's hard disk. Thereafter, eDirectory Agent continues to write new entries to **eDirAgent.journal**, clearing the contents and writing to **eDirAgent.bak** each time **eDirAgent.journal** reaches 1MB.

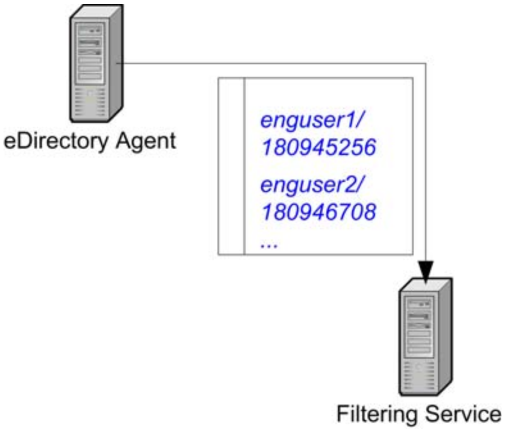
✓ **NOTE**

If eDirectory Agent receives a new request having an IP address already included in a user name/IP entry in its map, it *replaces* the existing pairing in its map with the new one.



*eDirectory Agent stores user name/IP address pairs in its local map*

4. **Filtering Service gets user information:** eDirectory Agent sends user names and IP addresses for logon sessions whenever they are initiated. Data is sent over port 30700. Filtering Service records user name/IP address pairs to its own copy of the user map in local memory. Note that there are no security risks in this data transfer, as no confidential information (such as user passwords) is transmitted.



*eDirectory Agent sends user name/IP address pairs to Filtering Service*

✓ **NOTE**

If you configured eDirectory Agent during setup to require authentication, the eDirectory Agent service checks the password provided by User Service against the password you specified during setup or via the **User Identification** settings in Websense Manager. See page 40 for information about authentication between eDirectory Agent and User Service.

5. **Filtering Service gets group information for users logged on:** 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.

   If you have configured filtering policies based on directory objects, Filtering Service uses this configuration in determining which filtering policies to apply to users making internet requests. Filtering Service does not determine the filtering policy every time an internet request is made; policy determinations are cached for three hours by the server.

6. **Websense filters users making internet requests:** Filtering Service uses the policy configuration established via Websense Manager to apply filtering policies to logged-on users. Filtering Service does not determine the filtering policy every time an internet request is made; policy determinations are cached for three hours by the server. For more information about filtering based on directory object policies, please see the *User Identification* and *Filtering Basics* chapters of the *Websense Enterprise Administrator's Guide*.

## Components

These descriptions will help you understand the role of each piece involved in transparent identification with the Websense eDirectory Agent.

### *eDirectory Agent*

The 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**. Each time the local map in **eDirAgent.journal** reaches 1MB in size, eDirectory Agent backs up the contents to **eDirAgent.bak**.

With the help of the Websense User Service, eDirectory Agent supplies this information to the 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 the Websense eDirectory Agent.

One instance of the Websense eDirectory Agent can support one 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* chapter of the *Websense Enterprise Administrator’s Guide* for more information on eDirectory Agent deployment.

## *User Service*

User Service works together with eDirectory Agent to provide an up-to-date list of users logged on to Novell eDirectory server. Both Filtering Service and User Service receive updates on user logon sessions from eDirectory Agent, as logons are recorded to the Agent's user map. User Service also interacts with Novell eDirectory to get group information corresponding to logged-on users.

User Service interacts with eDirectory Agent much like Filtering Service does. However, the interaction between eDirectory Agent and Filtering Service is of central importance. Filtering Service is the end destination for information about users authenticated by Novell eDirectory. When troubleshooting user identification problems, it is most important to determine whether Filtering Service is getting the latest and most accurate user data.

User Service mediates communication of user logon information between Filtering Service and eDirectory Agent. User Service can run on Windows, Linux, or Solaris. In Solaris or Linux environments, you can still filter based on user or group policies, and take advantage of the transparent identification feature. User Service translates logon session data provided by eDirectory Agent so the Websense Policy Server can apply the appropriate internet filtering policies.

## *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 a new user logged on to an existing machine, or a new user logged on to a new machine.

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, the server 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.

Websense 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/she is blocked from internet access entirely.

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

✓ **NOTE**

It is crucial that Filtering Service and eDirectory Agent can communicate successfully; problems with communication between components can cause user identification errors. See page 68 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 **Global** policy.

## *Files Used in Transparent Identification*

All but one 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\ (installed by default on eDirectory Agent machine; runs as a service or daemon named **Websense eDirectory Agent**) | Automatically sends new entries to Filtering Service, when queried.<br>Allows communication of transparent identification configuration from Websense Manager to eDirectory Agent. |
| **wsedir.ini** | \Websense\bin\<br>(on eDirectory Agent machine) | Contains initialization parameters for eDirectoryAgent.exe, which performs eDirectory Agent processes. See page page 76 for a description of each parameter. |
| **eDirAgent.journal** | \Websense\bin\<br>(on eDirectory Agent machine) | Stores user-name-to-IP address pairs (from local memory).<br>Updated when logon information is added to or removed from the Agent's map. Read upon startup. |
| **eDirAgent.bak** | \Websense\bin\<br>(on eDirectory Agent machine) | Houses backup copy (on hard disk) of eDirectory Agent's user name-to-IP address map.<br>Updated whenever **eDirAgent.journal** exceeds 1MB. Read upon startup. |
| **ignore.txt** | \Websense\bin\<br>(on eDirectory Agent machine, if created) | Contains list of user names for eDirectory Agent to ignore, and the workstations on which to ignore those names (see page 41). |

# Implementation

Deployment of the transparent identification feature involves some important installation and setup considerations, discussed in this section. In addition to the standard transparent identification setup procedure, there are configuration options for increasing security and performance of the user identification process.

## Deployment of eDirectory Agent

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

In most cases, you only need one 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.

Websense, Inc., does *not* support using eDirectory Agent together with DC Agent.

## Default Directory Protocol

The Websense eDirectory Agent can use Netware Core Protocol (NCP) or Lightweight Directory Access Protocol (LDAP) to get user logon information from Novell eDirectory, depending on your configuration. By default, in Websense v6.1, eDirectory Agent on Windows uses NCP. Meanwhile, eDirectory Agent on Linux or Solaris must use LDAP.

If you are running eDirectory Agent on Windows, but still would like the Agent to use LDAP to query Novell eDirectory, you can set the Agent to use LDAP instead of NCP. Generally, NCP provides a more efficient query mechanism. However, if your environment supports LDAP (for example, you have rules on a firewall that allow LDAP but not certain other protocols), then you may want to continue using LDAP.

To set the Websense eDirectory Agent on Windows to use LDAP instead of NCP, do the following.

1. Ensure that you have at least one Novell eDirectory replica containing all directory objects you want to monitor and filter in your network.
2. Stop the Websense eDirectory Agent service, using the **Services** applet accessible via **Administrative Tools** (*Windows 200x*).
3. Go to the eDirectory Agent installation directory, and locate the text file named **wsedir.ini**.
4. Open the file in a text editor (preferably one that supports line-wrapping).

5. Modify this line as indicated:

   ```
   QueryMethod=0
   ```

   where 0 sets the Agent to use LDAP to query Novell eDirectory. (The default value of 1 tells Novell eDirectory Agent to use NCP.)
6. Save and close the file when all entries are complete.
7. Restart the Websense eDirectory Agent service.

eDirectory Agent will use LDAP to query Novell eDirectory for user logon information.

## Authenticated Connections

You can configure eDirectory Agent and User Service to require an authenticated connection before data is transmitted between them. If you elect to use an authenticated connection, eDirectory Agent requires authentication from User Service.

When eDirectory Agent is configured to require authentication from User Service, eDirectory Agent validates User Service against a password you specify, before it sends data to User Service.

1. During installation of eDirectory Agent, choose to use an **Authenticated Connection** between eDirectory Agent and User Service, and specify a password to be used for the connection.
2. After eDirectory Agent installation is complete, enter the connection password via Websense Manager:
   a. Choose **Server > Settings**. The **Settings** dialog box appears.
   b. Select User Identification at the left. Installed agents appear in a list under Identify users with these transparent identification agents.
   c. Select the eDirectory Agent you installed, and then click **Edit**. The **Edit Transparent Identification Agent** dialog box appears.
   d. Ensure that **Enable Authentication** is checked.
   e. Ensure that the password you specified for the authenticated connection is entered here. The password is case-sensitive.
   f. Click **OK**.
   g. Go to the **eDirectory Agent** tab.
   h. Under **eDirectory Agent Settings**, ensure that **Enable Authentication** is checked, and that the **Password** matches the one you just entered in the **Edit Transparent Identification Agent** dialog box.
   i. Click **OK** to close the **Settings** dialog box.
   j. Click **Done** to save your changes.

This password is global for all instances of eDirectory Agent. If you want to configure one instance of the Agent differently, specify a unique password by modifying the password parameter in the **wsedir.ini** file, as shown:

```
password=[PASSWORD]
```

where [PASSWORD] is a 4-to-16-character password of your choice. This password is case-sensitive.

## Configuring eDirectory Agent to Ignore Particular User Names

If desired, you can configure eDirectory Agent to ignore logon names that are not associated with actual users. Some Windows 200x and XP services contact domain controllers using the workstation identities of active Websense users in your network.

Windows XP automatically sets up two internal user names (Local Service and Network Service) for various internal processes to use for communication with domain controllers. You can configure Websense to ignore such user names.

Follow these instructions to set up a list of user names, or user name/workstation pairs, for the Websense eDirectory Agent to ignore.

1. Stop the Websense eDirectory Agent service.
2. In the eDirectory Agent installation directory, create a text file named **ignore.txt**.
3. Type a one-line entry for each user name to be ignored. Websense will ignore user names listed this way, regardless of the associated workstation. Do not use wildcard characters, such as “*”.

   In the following example:

   ```
   johnsmith
   aperez, enggroup1
   ```

   The user name **johnsmith** will be ignored for ALL workstations. The user name **aperez** will be ignored only for the domain **enggroup1**.
4. Save and close the file when all entries are complete.
5. Restart the Websense eDirectory Agent service.

eDirectory Agent will ignore the user names you have specified, and Websense will not consider these names in filtering.

CHAPTER 6

# FAQs

This section answers common questions received by Websense Marketing and Technical Support personnel regarding the transparent identification feature. These answers may facilitate the decision process if you are currently using Websense v5.5 or earlier, and are looking to upgrade to v6.1. Questions are grouped by agent.

## DC Agent

Questions about DC Agent include the following:

- Does my version of Websense support DC Agent?
- How do I identify the version of DC Agent?
- Is transmission of user logon session data reliable?
- When are users expired from the user map?
- What happens if Websense, or the network, shuts down unexpectedly?
- Is transparent identification secure?
- Which ports listen for data transmitted by DC Agent?
- Can DC Agent be installed and run on a machine other than a domain controller?
- If I add a new domain controller to the network, will DC Agent find it?
- Does the DC Agent service require special rights to run properly?

### *Q: Does my version of Websense support DC Agent?*

**A:** Websense versions 4.3.x, 4.4.x, 5.1 and 5.2 are compatible with DC Agent v1.0 and above. Websense versions 5.5 and 5.5.2 work with DC Agent v2.x. Websense Enterprise v6.1 works with DC Agent version 6.1.

To upgrade DC Agent on the same machine, install DC Agent over the existing copy. The new version of DC Agent will automatically use the settings specified in your existing ignore.txt file, if you have one.

If you are installing the new version of DC Agent on a different machine than before, you must copy any existing ignore.txt file from the old DC Agent's hard disk, and place it on the new DC Agent machine, under **\Websense\bin\.**

### *Q: How do I identify the version of DC Agent?*

**A:** On the machine where DC Agent is installed, go to the **Websense\bin\** directory. Right-click the file **XidDcAgent.exe**, and then choose **Properties**. Go to the **Version** tab to view the version number for DC Agent.

### *Q: Is transmission of user logon session data reliable?*

**A:** Yes, data transmission between DC Agent and Filtering Service is completely 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 one or the other service is not running.

In order for the capture of logon session data to be accurate, it is necessary for users to log on to domains, and not to their local machines. If a user logs on locally, that logon session is not recorded on any domain controller, and DC Agent cannot capture logon session data. Therefore, DC Agent can only be implemented in an environment using Windows domains.

### *Q: When are users expired from the user map?*

**A:** By default, user name/IP address pairs that DC Agent cannot verify against users currently logged on to workstations expire every hour. You can change this interval by editing the **Workstation Settings: User Entry Timeout** value in Websense Manager. (Choose **Server > Settings**, and go to the **DC Agent** tab.) Increasing this interval may lessen user map accuracy, as the map would potentially retain old user names for a longer time. Decreasing this interval to less than the **User Map Verification Interval** value may cause problems with the expiration process. User names may be removed from the user map before they can be verified.

User name/IP address pairs that DC Agent obtains via domain controller query remain in the map for longer (24 hours by default). This interval is configurable via **Domain Controller Settings: User Entry Timeout** in Websense Manager.

Every 10 seconds, DC Agent checks for users newly logged on to the network, and writes these user name/IP address entries to the **XidDcAgent.journal** file. This data is also backed up to the **XidDcAgent.bak** file every 10 minutes, by default.

### *Q: What happens if Websense, or the network, shuts down unexpectedly?*

**A:** Logon session data is preserved in the **XidDcAgent.bak** file on the DC Agent machine. If the DC Agent machine shuts down, it records the latest logon session information to this file. DC Agent reads this file upon restart, and then resumes monitoring where it left off before the shutdown.

In a worst-case scenario (such as a power outage resulting in abnormal network shutdown), Filtering Service continues to use the existing user map if it cannot communicate with DC Agent. The information in the **XidDcAgent.bak** file is 10 minutes older than the time of shutdown, at the most. Any users whose logon information was not captured may need to log on again to network domains in order to be identified. User data is also saved in the **XidDcAgent.journal** file.

It is possible that **XidDcAgent.journal** had not yet reached 1MB, and was not persisted to **XidDcAgent.bak**. In this case, you can still retrieve user data from **XidDcAgent.journal**.

### *Q: Is transparent identification secure?*

**A:** Transparent identification is a secure method of user authentication, 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, you can also configure an authenticated connection between DC Agent and User Service. See page 14 for details.

If desired, you can disable NetBIOS usage – for example, if you prefer to close the port used by NetBIOS traffic. In this case, DC Agent relies on DNS lookup to identify internet request sources. See the UseNetBIOS initialization parameter in *Appendix A Transparent Identification Configuration Settings* for details.

### *Q: Which ports listen for data transmitted by DC Agent?*

**A:** The port over which DC Agent transmits data is the port you specify during installation of DC Agent (30600 by default). Filtering Service connects over this port when requesting user information from DC Agent. This port can be changed by editing the transid.ini file. See *Appendix A Transparent Identification Configuration Settings* for details about the port parameter. See page 15 for help with editing **transid.ini**.

### *Q: Can DC Agent be installed and run on a machine other than a domain controller?*

**A:** With Websense versions 4.4 and above, DC Agent can be run on any Windows NT or 2000 Server machine. With Websense version 5.2 and later, DC Agent can optionally be run on a Windows 2003 Server machine. With Websense v6.1, DC Agent must be run on Windows 2000 or later (not NT).

### *Q: If I add a new domain controller to the network, will DC Agent find it?*

**A:** DC Agent identifies new and existing domain controllers every 24 hours by default, *and* upon startup of the DC Agent service. This 24-hour interval is configurable via the DiscoverInterval parameter; see *Appendix A Transparent Identification Configuration Settings* for details. DC Agent stores domain information to the hard disk of the server where it is installed, in a file called dc_config.txt.

### *Q: Does the DC Agent service require special rights to run properly?*

**A:** 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 using administrative rights. See page page 59 for details about troubleshooting DC Agent's domain detection behavior and configuring specific rights for the DC Agent service.

# Logon Agent

Questions about Logon Agent include the following:

- Does my version of Websense support 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 workstation to the network, does Logon Agent detect it?
- When do users expire from the user map?

### *Q: Does my version of Websense support Logon Agent?*

**A:** The Websense Logon Agent was introduced with Websense Enterprise v5.5. If you are running an earlier version of Websense, you will need to upgrade to v5.5 or v6.1.

### *Q: What is the advantage of using Logon Agent?*

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

### *Q: Why would I use Logon Agent in combination with DC Agent?*

**A:** In most cases, using either DC Agent or Logon Agent is sufficient. However, there are many network environment variables that may affect which agent or agents you choose to use. If you have had reliability problems in the past with DC Agent, you can use Logon Agent in conjunction with DC Agent to enhance reliability. In this case, if DC Agent does not 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.

DC Agent also captures activity by system users (services) that may not have individual accounts associated with particular workstations. You can prevent this by adding any system user names to DC Agent's **ignore.txt** file (see *Configuring DC Agent to Ignore Particular User Names*, page 15).

### *Q: Can I install both Logon Agent and eDirectory Agent on the same machine?*

**A:** No. Websense does not support communication with both Novell eDirectory and an NTLM-based directory service, or Active Directory, at the same time. Optionally, you can have both agents installated, with only one active agent.

### *Q: Can I install multiple instances of Logon Agent on the same machine?*

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

### *Q: If I add a new workstation to the network, does Logon Agent detect it?*

**A:** The logon script that is configured to push out LogonApp.exe to client machines should activate LogonApp.exe on the new workstation. When a user logs on to that workstation, LogonApp.exe detects the logon session, and sends that information to Logon Agent.

### *Q: When do users expire from the user map?*

**A:** When 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. You can change this behavior by modifying the Entry Lifetime (non-persistent mode) value in Websense Manager. (Choose Server > Settings, select User Identification, and go to the Logon Agent tab.)

## RADIUS Agent

Questions about RADIUS Agent include the following:

- Will users at client workstations 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 users expire from the user map?
- Can RADIUS Agent ignore certain users or workstations like DC Agent does?
- How much bandwidth does RADIUS Agent occupy?
- What happens if Websense, or the network, shuts down unexpectedly?

### *Q: Will users at client workstations be identified regardless of logon domain?*

**A:** As long as the RADIUS server can authenticate the user as a domain user, filtering policies assigned to that user should be applied properly.

### *Q: Can one RADIUS Agent communicate with multiple RADIUS servers, and vice versa?*

**A**: 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.

Websense, Inc., does not support running multiple instances of RADIUS Agent on one machine. See *Can I install multiple instances of RADIUS Agent on the same machine?* on page 47 for more details.

### *Q: Can one RADIUS Agent communicate with multiple Filtering Services, and vice versa?*

**A:** One instance of RADIUS Agent can communicate with multiple Filtering Services.

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.

### *Q: Can I install both DC Agent and RADIUS Agent on the same machine?*

**A:** Websense recommends running one instance of an Agent on a particular machine. However, you can install DC Agent and RADIUS Agent on the same machine.

Both Agents are automatically installed to the Websense\bin\ directory. The port numbers each Agent uses to communicate with the Websense Filtering Service are also made unique by default. By default, DC Agent uses port 30600; RADIUS Agent uses port 30800.

✓ **NOTE**

For details about running both DC Agent and RADIUS Agent on one machine with Websense Enterprise v5.1, see Websense Knowledge Base article number 1115.

You must also add the Agents to Websense Manager, and configure each Agent to talk with Filtering Service, via Websense Manager. See the *User Identification* chapter of the *Websense Enterprise Administrator's Guide* for complete instructions.

### *Q: Can I install multiple instances of RADIUS Agent on the same machine?*

**A:** No, you cannot install multiple instances of the same agent on the same machine. If multiple instances of the same agent are needed in the same deployment of Websense Enterprise v5.2 or later, those instances *must* be installed on separate machines.

Installation instructions for each agent are provided in the installation guides available from the Websense product documentation web page: http://www.websense.com/support/documentation/.

### *Q: Can I install both eDirectory Agent and RADIUS Agent on the same machine?*

**A:** Websense recommends running one instance of an agent on a particular machine. However, you can install eDirectory Agent and RADIUS Agent on the same machine.

In Websense Enterprise v5.2 and v5.5, both agents are automatically installed to the \bin directory. The port numbers each agent uses to communicate with the Websense Filtering Service are also made unique by default. By default, eDirectory Agent uses port 30700; RADIUS Agent uses port 30800.

You must also add the agents to Websense Manager, and configure each agent to talk with Filtering Service, via Websense Manager. See the *User Identification* chapter of the *Websense Enterprise Administrator's Guide* for complete instructions.

### *Q: When do users expire from the user map?*

**A:** User name/IP address pairs in the map created by RADIUS Agent expire every 24 hours by default. You can change this behavior by modifying the **User Entry Timeout** value in Websense Manager. (Choose **Server** > **Settings**, select **User Identification**, and go to the **RADIUS Agent** tab.)

### *Q: Can RADIUS Agent ignore certain users or workstations like DC Agent does?*

**A:** Yes, RADIUS Agent can be configured to ignore particular logon names.

Follow these instructions to set up a list of user names, or user name/workstation pairs, for the Websense RADIUS Agent to ignore.

1. Stop the Websense RADIUS Agent service.
2. In the RADIUS Agent installation directory, create a text file named **ignore.txt**.
3. Type a one-line entry for each user name to be ignored. Websense will ignore user names listed this way, regardless of the associated workstation. Do not use wildcard characters, such as “*”.

   In the following example:

   ```
   johnsmith
   aperez, enggroup1
   ```

   The user name **`johnsmith`** will be ignored for ALL workstations. The user name **`aperez`** will be ignored only for the domain **`enggroup1`**.
4. Save and close the file when all entries are complete.
5. Restart the Websense RADIUS Agent service.

RADIUS Agent will ignore the user names you have specified, and Websense will not consider these names in filtering.

### *Q: How much bandwidth does RADIUS Agent occupy?*

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

For 10,000 RADIUS users, RADIUS Agent is working without problems all platforms, and with memory usage between ~15MB (for Windows) and 25MB (for Solaris).

### *Q: What happens if Websense, or the network, shuts down unexpectedly?*

A: Logon session data is preserved in the **wsradius.bak** file on the RADIUS Agent machine. If the RADIUS Agent machine shuts down, it records the latest logon session information to this file. RADIUS Agent reads this file upon restart, and then resumes monitoring where it left off before the shutdown.

In a worst-case scenario (such as a power outage resulting in abnormal network shutdown), Filtering Service continues to use the existing user map if it cannot communicate with RADIUS Agent. The information in the **wsradius.bak** file is 10 minutes older than the time of shutdown, at the most.

It is possible that **wsradius.journal** had not yet reached 1MB, and was not persisted to **wsradius.bak**. In this case, you can still retrieve user data from **wsradius.journal**.

# eDirectory Agent

Questions about eDirectory Agent include the following:

- 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 physical machine?
- Can I install both DC Agent and eDirectory Agent on the same physical machine?
- Can I install both eDirectory Agent and RADIUS Agent on the same physical machine?
- When do users expire from the user map?
- Can eDirectory Agent ignore particular user or workstation names like DC Agent does?
- How long do entries remain in eDirectory Agent's user map? Are any of the DC Agent parameters (such as EntryLifetime) relevant to eDirectory Agent?
- If the Websense administrator password is changed, do I need to reinstall eDirectory Agent to reflect the change?
- Are any Windows registry entries added with installation of eDirectory Agent?
- How many users can eDirectory Agent handle?
- What happens if Websense, or the network, shuts down unexpectedly?

### *Q: Can one instance of eDirectory Agent communicate with multiple Novell eDirectory servers, and vice versa?*

**A:** Yes, one instance of eDirectory Agent can communicate with multiple eDirectory servers. Technically, this means that one instance of the Websense eDirectory Agent supports one Novell eDirectory master, plus any number of Novell eDirectory "replicas" running on separate machines. To enable this, you must do the following:

Add the IP addresses of all additional eDirectory servers in Websense Manager. To do this: Choose **Server > Settings**, select **User Identification** at the left, and then go to the **eDirectory Agent** tab.

Ensure that all instances of Novell eDirectory server share the same user account and root context. These are specified during initial setup of the Websense eDirectory Agent, and allow eDirectory Agent to access Novell eDirectory server. If all of your Novell eDirectory servers do not share the same user account or root context, you may not get accurate user information from all of the servers. See the *Websense Enterprise Administrator's Guide* for eDirectory Agent setup instructions.

It is always true that multiple instances of eDirectory Agent can communicate with one Novell eDirectory server.

### *Q: Can one instance of eDirectory Agent communicate with multiple Filtering Services?*

**A:** Yes, one instance of eDirectory Agent can communicate with multiple instances of the Websense Filtering Service. You can configure one instance of Filtering Service to communicate with multiple eDirectory Agents on different machines. If you decide to let Filtering Service communicate with multiple eDirectory Agents installed on the *same* physical machine, please refer to the next question.

### *Q: Can I install multiple instances of eDirectory Agent on the same physical machine?*

**A:** It is not recommended to run 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.

### *Q: Can I install both DC Agent and eDirectory Agent on the same physical machine?*

**A:** It is very rare that both a Windows-based directory service and Novell eDirectory server reside in the same network. Websense, Inc. does not support such a scenario, because Websense Manager does not currently allow configuration of two different types of directory service for the same installation.

### *Q: Can I install both eDirectory Agent and RADIUS Agent on the same physical machine?*

**A:** Websense recommends running one instance of an Agent on a particular machine. However, you can install eDirectory Agent and RADIUS Agent on the same machine.

Both agents are automatically installed to the \bin directory. The port numbers each agent uses to communicate with the Websense Filtering Service are also made unique by default. By default, eDirectory Agent uses port 30700; RADIUS Agent uses port 30800.

You must also add the agents to Websense Manager, and configure each agent to talk with Filtering Service, via Websense Manager. See the *User Identification* chapter of the *Websense Enterprise Administrator's Guide* for complete instructions.

### *Q: When do users expire from the user map?*

**A:** User name/IP address pairs in the map created by eDirectory Agent expire every 24 hours by default. You can change this behavior by modifying the **User Entry Timeout** value in Websense Manager. (Choose **Server** > **Settings**, select **User Identification**, and go to the **eDirectory Agent** tab.)

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**. See page 91 for details on Agent initialization parameters.

### *Q: Can eDirectory Agent ignore particular user or workstation names like DC Agent does?*

**A:** Yes, eDirectory Agent can be configured to ignore particular logon names.

Follow these instructions to set up a list of user names, or user name/workstation pairs, for the Websense eDirectory Agent to ignore.

1.Stop the Websense eDirectory Agent service.

2. In the eDirectory Agent installation directory, create a text file named **ignore.txt**.

3.Type a one-line entry for each user name to be ignored. Websense will ignore user names listed this way, regardless of the associated workstation. Do not use wildcard characters, such as "*".

In the following example:

```
johnsmith
aperez, enggroup1
```

The user name **`johnsmith`** will be ignored for ALL workstations. The user name **`aperez`** will be ignored only for the domain **`enggroup1`**.

4.Save and close the file when all entries are complete.

5.Restart the Websense eDirectory Agent service.

eDirectory Agent will ignore the user names you have specified, and Websense will not consider these names in filtering.

### *Q: How long do entries remain in eDirectory Agent's user map?*

**A:** User name/IP address entries remain in the user map for 30 seconds by default. Every 30 seconds, the Agent queries Novell eDirectory server to refresh the user map. Only new entries since the last query are added to the map. You can adjust the query interval by modifying the `PollInterval` value in the **wsedir.ini** file. To do this, go to the **Websense\bin\** directory on the eDirectory Agent machine, and edit **wsedir.ini** using a text editor. See *Appendix A Transparent Identification Configuration Settings* for information on possible values for `PollInterval`.

If a user logs on locally, eDirectory Agent adds an IP address and an empty user name placeholder to the user map. Every 30 seconds, eDirectory Agent sends updates to Filtering Service. In a worst-case scenario, it takes one minute at the most for Filtering Service to discover that a user is logged on locally, and apply the appropriate filtering policy to that user.

The eDirectory Agent setting **User Entry Timeout** determines the lifetime of user/name IP address pairs in the Agent's user map. By default, the timeout is 24 hours. To access this setting in Websense Manager, choose **Server** > **Settings**, select **User Identification**, and go to the **eDirectory Agent** tab.

### *Q: If the Websense administrator password is changed, do I need to reinstall eDirectory Agent to reflect the change?*

**A:** No, you do not need to uninstall and reinstall eDirectory Agent to reflect a password change. You can simply re-run the eDirectory Agent installation program, accepting all the default settings. Accepting the default setting for the administrator password reverts the password to whatever you specified during initial installation. When the installation program is finished, restart the eDirectory Agent service or daemon.

### *Q: Are any Windows registry entries added with installation of eDirectory Agent?*

**A:** No specific registry entries are added during eDirectory Agent installation, except for the standard keys required to run the Agent as a Windows service. These standard keys are stored in the registry under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**.

### *Q: How many users can eDirectory Agent handle?*

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

### *Q: What happens if Websense, or the network, shuts down unexpectedly?*

**A:** Logon session data is preserved in the **eDirAgent.bak** file on the eDirectory Agent machine. If the eDirectory Agent machine shuts down, it records the latest logon session information to this file. eDirectory Agent reads this file upon restart, and then resumes monitoring where it left off before the shutdown.

In a worst-case scenario (such as a power outage resulting in abnormal network shutdown), Filtering Service continues to use the existing user map if it cannot communicate with eDirectory Agent. The information in the **eDirAgent.bak** file is 10 minutes older than the time of shutdown, at the most.

It is possible that **eDirAgent.journal** had not yet reached 1MB, and was not persisted to **eDirAgent.bak**. In this case, you can still retrieve user data from **eDirAgent.journal**.

# CHAPTER 7 Troubleshooting

Troubleshooting tools and methods related to transparent identification are outlined in this section. Common causes of user identification problems, and how to determine and solve these problems, begin ons page 57.

Problems with transparent identification of users almost always result in filtering errors, so if a user is not being filtered properly, it is likely that either Filtering Service or DC Agent, or both, are not identifying that user correctly.

## Troubleshooting tools

Various built-in Windows networking tools, in addition to the Websense troubleshooting tools, can help you in resolving any issues that may arise.

### Windows Services (or Service Control Manager)

Transparent identification agents, User Service, and Filtering Service run as Windows services, so they are accessible from the Windows Services manager. From the Windows Control Panel, select Administrative Tools (Windows 2000/2003 only), and then select Services to open the Services manager.

### Windows Event Viewer

Because it records error messages pertaining to Windows events such as service activities, the Event Viewer can help you identify network or service errors that may be causing user identification problems.

To access the Event Viewer: Choose **Start** > **Programs** > **Administrative Tools** > **Event Viewer**. In the Event Viewer, click **Application Log**.

### Websense Log

In Solaris and Linux environments, Websense writes errors to the **Websense.log** file in the **Websense\bin\** directory. This error record is comparable to the Windows Event Log.

### Websense ConsoleClient

This tool is automatically installed to the **Websense\bin\** directory on the Filtering Service machine, and includes several facilities classified as Tracing and Printself modules. ConsoleClient runs locally or on a remote Windows Server machine, and helps Websense Technical Support obtain diagnostic information for troubleshooting purposes. The Tracing and Printself modules produce data that can either be saved to a static text file, or viewed real-time from a terminal window via an application called Consolecout.

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, it is recommended that you first check all network connections, and then check the Windows Event Viewer and/or the Websense Log for related error messages. It is frequently not necessary to use ConsoleClient at all.

The transparent identification agents store user name-to-IP address correspondences to a “user map” in local memory. Analyzing the user name associations an Agent has recorded can help you to determine whether users and workstations are being identified correctly.

To obtain user map output from ConsoleClient:

1. On the Filtering Service machine, open a command window.
2. At the command prompt, type the following:

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

> ✓ **NOTE**
>
> For RADIUS Agent, this port number must match the `DiagServerPort` value in the **wsradius.ini** file (30801 by default).

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

Other potential causes of user identification problems are described next.

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* on page 57. In most cases, it is possible to solve problems using the procedures provided in this paper.

## Websense TestLogServer

TestLogServer is a purely diagnostic form of the Websense Log Server. You can use TestLogServer to see if a user is being properly identified. TestLogServer must be run on a machine that is not currently running the Websense Log Server.

To perform a user identification test via TestLogServer:

1. Stop the Websense Log Server service if it is running on the same machine where you will be running TestLogServer.
2. Go to a command prompt.
3. Navigate to the Websense installation directory (typically, **\Program Files\Websense\bin\** or **\Program Files\Websense Enterprise\EIM\bin\**, depending on the version of Websense.
4. At the prompt, type the following command:

   ```
   testlogserver
   ```

   and press Enter.

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

   To send the output to a text file, add the parameter `-file filename.txt`. For example: `testlogserver -file log.txt` sends the data to the file **log.txt** in the installation directory.
5. 5.Ensure that Websense is passing traffic to the machine where you are running TestLogServer.
   a. Start Websense Manager (Start > Programs > Websense > Websense Manager).

   b. Choose **Server > Settings**. The **Settings** dialog box appears.
   c. Select **Logging** at the left.
   d. In the **Server** field, enter the IP address of the machine running testlogserver.exe (this does not necessarily have to be the Websense machine).
   e. Click **OK**.

6. TestLogServer should be sending all traffic to your display, as well as to the **log.txt** file if the `-file` parameter was enabled.
7. Have a user try to access a site.
8. Check the log file for this site and user name.
9. Press CTRL-C to stop TestLogServer.

If the user name *is* passed, then the problem most likely has to do with the way the policies are configured. Verify that policies and category sets are configured as desired in Websense Manager.

If something other than the user name is passed, or the wrong user name is passed, see the troubleshooting items in this section for additional, possible causes of the problem.

If nothing is passed for the user name, check to see if the transparent identification agent and Websense are set up properly. The *Websense Enterprise Administrator's Guide* includes installation and configuration details for each of the transparent identification agents.

## Websense RADIUS Agent Diagnostic Tool

RADIUS Agent has built-in diagnostic capabilities that you can use to determine whether RADIUS Agent is identifying users correctly.

To activate RADIUS Agent logging and debugging:

1. Stop the RADIUS Agent service.
2. On the RADIUS Agent machine, go to the RADIUS Agent installation directory (**Websense\bin**\).
3. Open the file **wsradius.ini** in a text editor.
4. Locate the section named [RADIUSAgent].
5. Change the line

   ```
   DebugMode=Off
   ```

   to

   ```
   DebugMode=On
   ```

   This enables logging and debugging.
6. Modify the line

   ```
   DebugLevel=3
   ```

   where `3` is the level of message verbosity (1 indicates less detail, 3 indicates the most detail).
7. Modify the line

   ```
   LogFile=[filename.txt]
   ```

   where `[filename.txt]` is the log output file. By default, log output is sent to the RADIUS Agent console. If you are running the Agent in console mode, you can optionally keep this default value.
8. Save and close the **wsradius.ini** file.

9. Start the RADIUS Agent service.
10. Go to the RADIUS Agent installation directory, and open the RADIUS Agent log file.

    A successfully identified logon session looks like this:

    ```
    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=QUALITY\administrator
    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=QUALITY\administrator
    ```

    This indicates that RADIUS Agent detected the user QUALITY\administrator logging into a machine with IP address 10.202.11.7, added this user name/IP address pair to its user map, and received an authentication response from the RADIUS server for this user.

## Websense eDirectory Agent Diagnostic Tool

eDirectory Agent also has built-in diagnostic capabilities that you can use to determine whether eDirectory Agent is identifying users correctly.

To activate eDirectory Agent logging and debugging:

1. 1.Stop the eDirectory Agent service.
2. 2.On the eDirectory Agent machine, go to the eDirectory Agent installation directory (**Websense\bin**\).
3. 3.Open the file **wsedir.ini** in a text editor.
4. 4.Locate the section named [eDirAgent].
5. 5.Change the line
6. DebugMode=Off
7. to
8. DebugMode=On
9. This enables logging and debugging.
10. Modify the line

    ```
    DebugLevel=3
    ```

    where `3` is the level of message verbosity (1 indicates less detail, 3 indicates the most detail).
11. Modify the line

    ```
    LogFile=[filename.txt]
    ```

    where `[filename.txt]` is the log output file.Save and close the **wsedir.ini** file.
12. Start the eDirectory Agent service.
13. 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
```

This indicates that the user Admin logged onto Novell eDirectory server and was identified correctly by eDirectory Agent.

# Common Problems

Most frequently, problems with transparent identification involve users not being filtered according to the policies assigned to them. Filtering errors occur when Filtering Service is not able to identify a user. In the context of transparent user identification, this is due to one of the following causes:

- The transparent identification agent and Filtering Service are not communicating data.
- A user's network identity is obscured. In this case, Filtering Service can't correlate the user with a user or group object in the directory service, and therefore can't apply filtering policies assigned to that object.
- (*DC Agent only:*) DC Agent is not receiving domain controller information.
- (*Logon Agent only:*) The logon script that invokes LogonApp.exe on a client machine does not run properly, for one of several possible reasons.
- (*Logon Agent only:*) NetBIOS or a NetBIOS service is disabled on the client machines.
- *(RADIUS Agent or eDirectory Agent only:)* A user is not authenticated by the RADIUS server or by Novell eDirectory server.
- *(Remote Filtering only:)* 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, first check the following:

- Filtering policies are configured appropriately and assigned to the correct users in the Websense Manager.
- Client workstations are connecting to the internet via the correct gateway.

Additional items to verify are described here in greater detail. Common problems are grouped by type of transparent identification agent. Use the troubleshooting tools described in the previous section and the procedures following each possible cause to identify and solve problems that may occur.

## DC Agent

Possible causes for transparent identification problems with DC Agent are listed in order of likelihood, so it's a good idea to start at the beginning of this section and work through the troubleshooting steps until you isolate and solve the problem.

### *DC Agent and Filtering Service are not communicating data*

If Websense is (a) filtering based on workstation or network policies, or on the Global policy, even after directory object policies have been assigned, or (b) logging user names incompletely or incorrectly, then user name/IP address pairs may not have reached Filtering Service. This occurs when DC Agent and Filtering Service fail to communicate.

Follow these steps to determine whether DC Agent and Filtering Service are able to exchange data, and subsequently resolve the problem.

1. Check for event messages in the Windows Application Event Log (Windows) or the Websense Log (Solaris/Linux) regarding successful or unsuccessful connections between DC Agent and Filtering Service. The authentication password for connections between DC Agent and Filtering Service may not be configured correctly, or the connection between DC Agent and Filtering Service may not be active, preventing successful authentication.

   a. If connection messages show success, then it is likely that DC Agent was not running prior to user logon. Verify that the DC Agent service is running, and then have users log on again.

      If messages indicate unsuccessful connections, check to see if the error messages reference "authentication failed." This indicates that there is a problem with the password for connections between DC Agent and User Service.

2. Verify that all DC Agent machines are alive on the network and can be contacted. User name/IP address pairs may fail to be transmitted because DC Agent is not running when a user logs on. To verify the activity of DC Agent:

   a. Make sure that the DC Agent service is running. To do this, check the service's status via Windows Services. (From the Windows Control Panel, select **Administrative Tools**, then **Services** to open the **Windows Services** window.)

   b. Ensure that connections from DC Agent to Filtering Service can be made via the port established during installation (30600 by default). To do this, you can use the telnet command to connect to the DC Agent machine on this port, as shown:

      ```
      telnet [IP address] 30600
      ```

      where `[IP address]` is the IP address of the DC Agent machine.

      `Telnet` allows you to log on as a network user to a remote computer (in this case, to the DC Agent machine).

      Alternatively, on Solaris/Linux or Windows you can run netstat -na from a command prompt. The netstat command returns a network status summary, which will show whether or not the connection to DC Agent has been established successfully.

3. Check the configuration in Websense Manager to verify that DC Agents are entered appropriately.

   a. Choose **Server** > **Settings**.

   b. Select **User Identification** at the left.

   c. Select the IP address or machine name of the DC Agent that is not authenticating properly, and click **Edit** to open the **Edit Transparent Identification Agent** dialog box.

   d. Make sure the **Enable Authentication** option is checked. Then re-enter the password.

   e. In Websense Manager, choose **File** > **Save Changes**, and then verify that the desired filtering policy is applied correctly.

4. If you have enabled an authenticated connection between DC Agent and Filtering Service, verify the password parameter in the **transid.ini** file, on the DC Agent machine.

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

   b. Locate the line containing the password parameter. Make sure the parameter is spelled with a lower-case "p," and that the password specified matches the one you entered during installation of DC Agent.

   c. Save the file, and then restart the DC Agent machine to activate any changes.

5. Check to see if Filtering Service is busy downloading the Master Database. Filtering Service cannot transmit data to or receive data from User Service or DC Agent while the database download process is running on the server. If Filtering Service attempts to get data from User Service, and a scheduled database download interrupts this action, data transmission fails.
   a. Check for event messages in the Windows Application Event Log or the Websense Log regarding successful or unsuccessful connections between User Service and Filtering Service.
   b. If the Master Database download process was running, then a Windows Event Viewer message references "data request failed." This problem should resolve itself after the database has finished downloading.

## *A user's or workstation's network identity is obscured*

When the identity of a user making an internet request is unclear, Filtering Service cannot apply the appropriate filtering policies. This occurs because incorrect user name/IP address pairs are associated with workstations, due to service or network user name confusion.

Some Windows services contact domain controllers from user workstations, causing the original user logged on from that workstation to be misidentified. Windows XP, for example, automatically sets up two internal user names (Local Service and Network Service) for various internal processes to use for communication with domain controllers.

DC Agent may record a service user name in place of a real user name. See *Configuring DC Agent to Ignore Particular User Names*, page 15 for how to configure DC Agent to ignore service user names.

Alternatively, you can correct this user identity problem by configuring DC Agent to ignore logon names that are not associated with actual users. See *Implementation*, page 13 for instructions.

## *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. Follow these steps to determine whether this is the problem:
  a. Go to the **\Websense\bin\** directory on the DC Agent machine, and open the file **dc_config.txt** in a text editor.
  b. Verify that all of your domain controllers are listed, and are set to "on."
  c. If domain controllers are missing from the list, type their names, and assign the value "on" to those machine names. For example:

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

  where `[Domain2]` is a domain name, and `DCN` is a domain controller name.

- DC Agent and/or User Service may be configured to use an anonymous account. To a domain controller, an anonymous account is equivalent to a Windows Guest account. If a domain controller has been set not to give the list of user logon sessions to an anonymous user, then when DC Agent attempts to get logon sessions, it is not allowed to download the list.

  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://msdn.microsoft.com/library/default.asp?url=/library/en-us/netmgmt/netmgmt/netsessionenum.asp), which may fail depending on your Local Security Policy or Trust Relationship configuration.

  User Service uses `NetUserGetGroups` (see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/netmgmt/netmgmt/netusergetgroups.asp), which requires domain administrative rights.

  To determine whether this is the problem, check the Windows Event Viewer.

  a. Open the Windows Event Viewer as described on page 53.

  b. If the DC Agent service does not have sufficient user rights, you may see the message ERROR_ACCESS_DENIED - 5 in the Windows Event Viewer.

  c. To correct the problem, go to the DC Agent machine, and then follow these steps.

  d. In each domain, create a user with a general name such as **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 this user.
     - Assign the same password to this user 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, as they must be entered later.

  e. Open Windows Services on each Websense DC Agent machine.

     *Windows 2000/2003:* Choose **Start** > **Programs** > **Administrative Tools**, and then double-click **Services**.

  f. Select the **Websense DC Agent** entry, and then click the **Stop** button.

  g. Double-click the **Websense DC Agent** entry to display the **Services** dialog box.

  h. On the **Log On** tab, select the **This account** option.

  i. In the text box, enter the user name for the account created earlier. Some environments require the format [domain\user name]. For example: **DomainName\wsDCAgent**.

  j. Enter and confirm the Windows password for this account.

  k. Click **OK** to close the dialog box.

  l. Select the **Websense DC Agent** entry in the **Services** dialog box, and then click the **Start** button to restart the DC Agent service.

     DC Agent now runs using an account with sufficient rights to access domain controllers. Repeat this procedure for User Service, if necessary.

- DC Agent may not be able identify the domain controllers in a particular domain.

  To determine whether DC Agent cannot identify certain domain controllers:

  a. Open the Windows Event Viewer (instructions are on page 53).

  b. If DC Agent cannot locate a Master Browser with a list of domain controllers for a particular domain, you will 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 and/or domain controller machines in other subnets. It is recommended to install a separate DC Agent in each subnet to avoid problems gathering logon information from domain controllers.

- DC Agent may not be able to contact a remote domain controller. This can occur when the domain controller machine has been shut down or restarted.

  To determine whether a network problem is preventing DC Agent from contacting a domain controller:

  a. Open the Windows Event Viewer as (instructions are on page 53).

  b. Open Windows Services on the Websense DC Agent machine.S

     *Windows 2000/2003:* Choose **Start** > **Programs** > **Administrative Tools**, and then double-click **Services**.

  c. If DC Agent cannot contact a domain controller, you may see the message ERROR_BAD_NETPATH – 53 in the Event Viewer.

  To troubleshoot the remote access problem:

  d. 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.

  e. Ensure that NetBIOS is bound to the network adapter on the DC Agent machine:

     - Search the registry for the key **RPC_Binding_Order**, and verify that NetBIOS is listed as a bound protocol.
     - Browse to the key **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ControlNetwork**.

       Search for the key **Ndi**. Under **Ndi**, locate the **Interfaces** key. The **Interfaces** key has two values: **Upper Range** and **Lower Range**. Ensure that at least one of these values has a Data item named **NetBIOS**.
     - Use the Windows Services applet to verify that Printing and Network services are running on the DC Agent machine.

  f. Ensure that remote administration is enabled on the remote domain controller machine.

     - On the remote domain controller machine, go to the Windows Control Panel.
     - Open the **Passwords** dialog box.
     - Go to the Remote Administration tab, and verify that Enable Remote Administration is checked.
     - If the domain controller or DC Agent machine is running *Windows 9x*, ensure that Remote Administration Services is installed on each computer.

  g. Make sure the computers are communicating via the same network protocol (for example, TCP/IP). For more details about connecting to a remote Windows 9x registry, see the following Microsoft Knowledge Base article:

     http://support.microsoft.com/default.aspx?scid=kb;EN-US;q193463

# Logon Agent

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

These possible causes of failure to capture a logon session will help you to troubleshoot the problem.

## *Logon Agent and Filtering Service are not communicating data*

If Websense is:

a) filtering based on workstation or network policies, or on the Global policy, even after directory object policies have been assigned; or

b) logging user names incompletely or incorrectly;

user name/IP address pairs may not have reached Filtering Service. This occurs when Logon Agent and Filtering Service fail to communicate.

Follow these steps to determine whether Logon Agent and Filtering Service are able to exchange data, and subsequently resolve the problem.

1. Check for event messages in the Windows Application Event Log (Windows) or the Websense Log (Solaris/Linux) regarding successful or unsuccessful connections between Logon Agent and Filtering Service. The authentication password for connections between Logon Agent and Filtering Service may not be configured correctly, or the connection between these components may not be active, preventing successful authentication.
    a. If connection messages show success, then it is likely that Logon Agent was not running prior to user logon. Verify that the Logon Agent service is running, and then have users log on again.

       If messages indicate unsuccessful connections, check to see if the error messages reference "authentication failed." This indicates that there is a problem with the password for connections between Logon Agent and User Service.
2. Verify that all Logon Agent machines are alive on the network and can be contacted. User name/IP address pairs may fail to be transmitted because Logon Agent is not running when a user logs on. To verify the activity of Logon Agent:
    a. Make sure that the Logon Agent service is running. To do this, check the service's status via Windows Services. (From the Windows Control Panel, select **Administrative Tools**, then **Services** to open the **Windows Services** window.)
    b. Ensure that connections from Logon Agent to Filtering Service can be made via the port established during installation (30602 by default). To do this, you can use the telnet command to connect to the Logon Agent machine on this port, as shown:

       ```
       telnet [IP address] 30602
       ```

       where **`[IP address]`** is the IP address of the Logon Agent machine.

       `Telnet` allows you to log on as a network user to a remote computer (in this case, to the Logon Agent machine).

       Alternatively, on Solaris/Linux you can run netstat -nr from a command prompt. The netstat command returns a network status summary, which will show whether or not the connection to Logon Agent has been established successfully.

3. Check the configuration in Websense Manager to verify that Logon Agents are entered appropriately.
    a. Choose **Server** > **Settings**.
    b. Select **User Identification** at the left. Installed agents appear in a list under Identify users with these transparent identification agents.
    c. Select the IP address or machine name of the DC Agent that is not authenticating properly. and
    d. Click **Edit** to open the **Edit Transparent Identification Agent** dialog box.
    e. Make sure the **Enable Authentication** option is checked. Then re-enter the password.
    f. Click **OK**.
    g. Go to the **Logon Agent** tab.
    h. Under **Logon Agent Settings**, ensure that **Enable Authentication** is checked, and that the **Password** matches the one you just entered in the **Edit Transparent Identification Agent** dialog box.
    i. Click **OK** to close the **Settings** dialog box.
    j. Click **Done** to save your changes.
    k. In Websense Manager, click the **Save Changes** button, and then verify that the desired filtering policy is applied correctly.
4. If you have multiple instances of Logon Agent, and one has a unique authentication password specified in its .ini file: Verify the password parameter in the AuthServer.ini file, on the machine where this instance of Logon Agent is installed.
    a. On the Logon Agent machine, go to the **Websense\bin\** directory, and open the file **AuthServer.ini**.
    b. Locate the line containing the password parameter. Make sure to use a lower-case “p,” and that the password specified matches the one you entered during installation of logon Agent.
    c. Save the file, and then restart the Logon Agent machine to activate any changes.
5. Check the configuration in Websense Manager to verify that Logon Agents are entered appropriately.
    a. Choose **Server** > **Settings**.
    b. Select **User Identification** at the left.
    c. Select the IP address or machine name of the Logon Agent that is not authenticating properly, and then click **Edit** to open the **Edit Transparent Identification Agent** dialog box.
    d. Make sure the **Enable Authentication** option is checked. Then re-enter the password.
    e. In Websense Manager, choose **File** > **Save Changes**, and then verify that the desired filtering policy is applied correctly.

6. Check to see if Filtering Service is busy downloading the Master Database. Filtering Service cannot communicate with User Service or Logon Agent while the database download process is running on the server. If Filtering Service attempts to get data from User Service, and a scheduled database download interrupts this action, data transmission fails.
    a. Check for event messages in the Windows Application Event Log or the Websense Log regarding successful or unsuccessful connections between User Service and Filtering Service.
    b. If the Master Database download process was running, then a Windows Event Viewer message references “data request failed.” This problem should resolve itself after the database has finished downloading.

### *The user's workstation 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, run the following command from a Windows command prompt:

```
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. In Windows 98, TCP/IP NetBIOS is enabled by default.

If NetBIOS is disabled on client machines, LogonApp.exe may not be able to run, causing users to be filtered by the Global policy by default. (If manual authentication is enabled, users will be prompted to log on via a browser window.)

If NetBIOS is disabled on the Logon Agent machine, Logon Agent may not be able to communicate with domain controllers. In this case, users would also be filtered by the Global policy or prompted to authenticate manually.

### *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 that will 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 in order for the logon script to run. A user profile can become corrupt due to Windows factors external to Websense.

To restore a corrupt user profile:

1. Log on as the workstation's local administrator.
2. Remove the user profile from C:\Documents and Settings\.
3. Log off and then log on again.

Windows recreates the profile automatically, and the logon script should execute.

# RADIUS Agent

There are several possible causes for transparent identification problems with RADIUS Agent, because the RADIUS environment is complex and involves many networking variables. Possible problems and related solutions are grouped by area of RADIUS configuration. Use the suggested solutions to isolate and solve user identification problems.

## 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. You can also set which actions are logged via the **Properties** panel.

### *RADIUS Agent may impact a VPN connection*

Since RADIUS Agent sits between a VPN client and VPN server, it may occur that RADIUS Agent blocks VPN traffic. In this case, you must remove the Agent. Simply stopping the Agent is not enough. The Agent must be torn down 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*

Please 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 Cannnot bind to port: 98 (Linux/Solaris).

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 very helpful in determining the cause of VPN connection or authentication problems, especially when distinguishing whether the problem lies in RADIUS Agent or VPN setup.

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

When using some RADIUS servers (Microsoft IAS for example), RADIUS Accounting must be enabled so 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. Consult 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 the Websense RADIUS Agent as a proxy. This involves adding RADIUS Agent as a client to the RADIUS server. Please refer to 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 will result 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. Please 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, please consult your respective product documentation.

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

It is recommended to 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 55 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, 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/Solaris).

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 Enterprise Administrator's Guide.

## *Users bypass a logon prompt to circumvent Websense filtering*

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

You can run TestLogServer to check whether the user is logged on locally. See page page 53 for instructions.

Alternatively, a workstation rule can be set up for filtering local users based on the workstations where they log on. Refer to the *Websense Enterprise Administrator's Guide* for instructions on configuring filtering policies for workstations or other objects.

## *A user name is duplicated in an Active Directory father domain and child domain*

If you have an Active Directory tree, users can have duplicated accounts in father and child domains - for example, testuser@father.com and testuser@child.father.com.

If those accounts share the same password, RADIUS Agent may not be able to identify the user correctly.

Make sure the VPN client has the correct domain information set before users log on.

# eDirectory Agent

Possible causes for transparent identification problems with the Websense eDirectory Agent are listed here.

## *Users are not being filtered correctly*

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

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

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

  You can run TestLogServer to check whether the user is logged on locally. See page page 53 for instructions.

  Alternatively, a workstation rule can be set up for filtering local users based on the workstations where they log on. Refer to the *Websense Enterprise Administrator's Guide* for instructions on configuring filtering policies for workstations or 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 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 **Global** policy.

  If these root context values are different, a user can log onto two different trees or branches in Novell eDirectory server, and still be identified by eDirectory Agent. However, when the Websense User Service determines the filtering policy for this user, it uses the root context specified in Websense Manager to retrieve information. User 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 or Solaris, and the Novell Modular Authentication Service (NMAS) is running when it should not be.

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

## *Users are not being identified in a Novell eDirectory/Cisco Content Engine environment*

In an environment where Websense is integrated with Cisco Content Engine v5.3.1.5 or higher, eDirectory Agent may be unable to identify users unless particular setup guidelines are followed. In such an environment, you will need to do the following:

- Install and run these Websense services on the same machine as Cisco 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. The file wsedir.ini is located in the Websense installation directory (\**Websense**\**bin**\\ by default).

  To specify Novell eDirectory replicas:

  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 the 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 **wsedir.bak** file from the machine running Websense, Cisco Content Engine and eDirectory Agent. By default, **wsedir.bak** is in the \**Websense**\**bin**\\ directory.
- Run any Websense Reporting Tools services on a machine *separate* from Cisco Content Engine and Websense.

See the *User Identification* chapter of the *Websense Enterprise Administrator's Guide* for complete instructions on configuring Websense to communicate with eDirectory Agent.

# Remote Filtering

In many cases, if a remote client is not filtered according to the policy assigned to it, the client was not identified by Websense. In a remote filtering situation, this frequently means the client cannot connect to Remote Filtering Server. This section assists you with troubleshooting communication problems between the client and the Server, which prevent Websense from identifying the client.

## *Remote users are not being filtered according to policies assigned to them*

There are several possible reasons for filtering problems with a remote client:

- 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 the proper IP addresses have been configured for internal and external communication.
    - Ensure 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 *only* cover those causes related to user identification. To troubleshoot general remote filtering problems, see Websense Knowledge Base article number **4732**.

1. Ensure that the Remote Filtering Server is running.
    a. Check that the machine is running.
    b. Check that the Remote Filtering Server service is running.

       *Windows:* Check the Services Control Panel to ensure the the **Websense Remote Filtering Service** is running.

       *Linux or Solaris:*

       - Go to the **/opt/Websense** directory.
       - From a command prompt, run: **./WebsenseAdmin status**.
       - The Remote Filtering Server service should be running. If not, run **./WebsenseAdmin start**.
2. Ensure that Remote Filtering Server is not installed on the same machine as Filtering Service.

   Installing these components on the same machine will cause a serious drain on resources on the machine. Filtering will become very slow at first and then fail and allow all requests.
3. Check that the connections are working properly.
    a. Check that the remote machines are able to communicate with the server. The ping command can be used to verify this connection.
    b. Check that that server machine is properly communicating with the network. Try to ping other machines on the local network.

4. Check that communication are properly configured for the server and the clients.

Remote Filtering Client instances must be able to connect to Remote Filtering Server, from both inside and outside the internet gateway or network firewall.

a. On the Remote Filtering Server machine, open the **securewispproxy.ini** file in a text editor.
This file is located in the directory where Remote Filtering Server was installed.
In Windows, this file is located in the **\Program Files\Websense\bin** directory.
In Linux and Solaris, this file is located in **/opt/Websense/bin** directory.

b. Under **Proxy Server parameters**, make note of these settings:

- **ProxyIP—**needs to match the IP address for NIC used for internal communications.
- **ProxyPort—**the port used on the server for external communications. The default setting is 80.
- **ProxyPublicAddress—**the IP address or hostname used for external access of the machine running Remote Filtering Server.

c. Under the **HeartBeat Server Parameters**, make note of the **HeartBeatPort** setting. This port is used for internal communications by Remote Filtering Client machines that have been moved to the network inside the firewall. The default setting is 8800.

d. Open a command prompt.

e. Run **ipconfig** on the server machine to get the IP addresses for each network interface card (NIC) in the server machine.
This is the same command in Windows, Linux and Solaris.

f. Check that the IP address values match the values found in the securewispproxy.ini file.

g. The values need to be checked on the client machines.
Please contact Technical Support for assistance. The technician will need the information gathered in the previous steps to verify that communications are properly set up.

# APPENDIX A Transparent Identification Configuration Settings

## Websense Manager Settings

Transparent identification agent configuration settings related to agent processes are defined here. For complete configuration instructions for an agent, please refer to the *User Identification* chapter of the *Websense Enterprise Administrator's Guide*.

To access these settings in Websense Manager, choose Server > Settings, select User Identification, and go to the appropriate agent tab.

### DC Agent

#### Domain Controller Settings: Enable Domain Controller Polling

Enables DC Agent's domain controller query process. This is enabled by default. Disabling this process is not recommended.

When enabled, DC Agent queries the domain controllers specified in its dc_config.txt file for user logon sessions.

To enable this setting, go to the **DC Agent** tab and check **Enable domain controller polling** under **Domain Controller Settings**. Specify the following parameters:

- **Query interval**: Specify the interval at which DC Agent queries domain controllers. Decreasing the query interval may provide greater accuracy in capturing logon sessions, but it also increases overall network traffic. Increasing the query interval decreases network traffic, but may also delay or prevent the capture of some logon sessions. 10 seconds (default) is an ideal interval for most cases.
- **User Entry Timeout**: Specify how long entries resulting from domain controller polling remain in DC Agent's user map.

Domain controller polling process uses both DNS lookup and NetBIOS communications. If for security reasons you do not want to allow NetBIOS communications, you can disable this setting. Alternatively, you can disable only NetBIOS communications so domain controller polling can remain active. See the UseNetBIOS .ini parameter description in this appendix for details.

#### Domain Controller Settings: Query Interval

See the definition for the `QueryInterval .ini` parameter, in this appendix.

### Domain Controller Settings: User Entry Timeout

The amount of time an entry obtained through domain controller polling remains in DC Agent’s user map.

**Default Value**

24 hours.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

DC Agent removes from its map user name/IP address entries that are older than this timeout period, and that DC Agent cannot verify against users currently logged on to domain controllers. Increasing this interval may lessen user map accuracy, as the map would potentially retain old user names for a longer time.

### Workstation Settings: Enable Workstation Polling

Enables DC Agent’s domain controller query process. This is enabled by default. Unless *all* users in your network log on to domain controllers, disabling this process is not recommended.

### Workstation Settings: User Map Verification Interval

The interval at which DC Agent contacts workstations to verify which users are logged on.

**Default Value**

15 minutes.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

DC Agent compares the query results with the user name/IP address pairs in the user map it sends to Filtering Service. It is generally recommended to accept the default value (15 minutes). 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.

### Workstation Settings: User Entry Timeout

The interval at which DC Agent refreshes entries obtained through workstation polling its user map.

**Default Value**

1 hour.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

DC Agent removes from its map user name/IP address entries that are older than this timeout period, and that DC Agent cannot verify against users currently logged on to workstations in the network. It is recommended to accept the default query value (1 hour). Increasing this interval may lessen user map accuracy, as the map would potentially retain old user names for a longer time. Decreasing this interval to less than the **User Map Verification Interval** value may cause problems with the expiration process. User names may be removed from the user map before they can be verified.

## Logon Agent

### Query Interval (persistent mode)

The interval at which LogonApp.exe (running on client machines) sends logon information to Logon Agent.

**Default Value**

15 minutes.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

In persistent mode, the logon application communicates logon information periodically, according to this interval setting. Decreasing this query interval may provide greater accuracy in identifying and filtering users, but also increases overall network traffic.

Note that if you change this value at any time, the change will 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.

### Entry Lifetime (non-persistent mode)

How long a user entry (user name/IP address pair) remains in Logon Agent’s user map before expiring.

**Default Value**

24 hours.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

How long a user entry (user name/IP address pair) remains in Logon Agent’s user map before expiring. This applies only when the application is running in non-persistent mode. In this mode, logon information is sent to the Agent only once for each logon.

## RADIUS Agent

### Authentication Ports

See the definitions for the `AuthInPort` and `AuthOutPort .ini` parameters, in this appendix.

### Accounting Ports

See the definitions for the `AccInPort` and `AccOutPort .ini` parameters, in this appendix.

## eDirectory Agent

### User Entry Timeout

The interval at which eDirectory Agent refreshes its user map.

**Default Value**

24 hours.

**Allowed Values**

Any.

**Required**

No.

**Synopsis**

eDirectory Agent refreshes its user map periodically, against logon sessions received. It is recommended to accept the default timeout value (24 hours).

# Initialization Parameters

The initialization parameters for each transparent identification agent are defined here, grouped by agent. Some values are editable. However, the default values are designed to maximize accuracy and efficiency in most environments. In many cases Websense, Inc. recommends leaving the default values as they are. Details are included in each definition.

## DC Agent

**IMPORTANT**

As of Websense Enterprise v5.5, many configuration parameters are available in Websense Manager, rather than only in an .ini file.

Other parameters are configurable in either place. These parameters are marked here with an asterisk (*). They are also marked with an asterisk (*) in Websense Manager. Such UI settings can be overridden by the related parameter in an .ini file. Please see the *User Identification* chapter of the *Websense Enterprise Administrator's Guide* for full configuration instructions.

The DC Agent initialization file (**transid.ini**) contains several parameters that determine how and when various DC Agent processes run. Only the port parameter always requires a value. All parameters and values described here are case-sensitive.

### AllDollarSign

Enables DC Agent to ignore logon sessions from any user names containing dollar signs ($). Available only with DC Agent versions 2.0.1 and higher. See page 43 for how to determine the version of DC Agent.

**Default Value**

True.

**Allowed Values**

True, False.

**Required**

No.

**Synopsis**

`AllDollarSign` ensures that DC Agent drops all entries containing dollar signs from its user map, without performing any additional verification of such names. This option is a more powerful version of `IgnoreDollarSign`. `AllDollarSign` was created due to occasional cases where the functionality of `IgnoreDollarSign` was not sufficient. (See the `IgnoreDollarSign` synopsis in this Appendix for more information.)

### *DiagServerPort

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

#### Default Value

30601.

#### Allowed Values

Between 1024 and 65535.

#### Required

No.

#### Synopsis

The Websense ConsoleClient tool is installed by default to the **\Websense\bin\** directory on the Websense Server machine. ConsoleClient can help you troubleshoot problems that may occur during the transparent identification process. See *Websense ConsoleClient* , page 53 for more information. Websense Technical Support can assist you with using this tool.

The `DiagServerPort` value rarely needs to be changed. However, if you do want to change it, Websense Technical Support can help you do this.

### DiscoverInterval

Interval at which the domain auto-discovery process runs.

#### Default Value

86400 seconds (or 24 hours).

#### Allowed Values

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 upon 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, as the process runs more frequently.

### IgnoreDollarSign

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

#### Default Value

True.

#### Allowed Values

True, False.

#### Required

No.

#### Synopsis

This parameter is used to prevent a problem involving Windows service user names. The problem occurs when a standard Windows 2000 service contacts a domain controller with a user name comprised of the workstation name followed by a dollar sign (for example, **wkstn$**). DC Agent interprets the service as a new user, for which no policy has been assigned. DC Agent can also potentially misinterpret a user name if a user has a shared resource mapped to a domain controller.

*When `IgnoreDollarSign` 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 workstation'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 workstation name do not match, DC Agent attempts to get the name of the actual user logged on from the source workstation. If it obtains a user name, DC Agent pairs that with the IP address of the source workstation, 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 Websense Server.

*When `IgnoreDollarSign` is set to false:* If it detects a user$ entry in its map, DC Agent attempts to replace it with an actual user name from the source workstation. 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 Value**

False.

**Allowed Values**

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, you can set this value to `True`. For example, you might want administrators logging on locally not to be detected by DC Agent.

### IgnoreRepeats

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

**Default Value**

True.

**Allowed Values**

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 workstation name/IP address pairs.

**Default Value**

600 seconds (or 10 minutes).

**Allowed Values**

Between 30 and 3600 seconds.

**Required**

No.

**Synopsis**

Every 10 minutes, DC Agent checks the workstation name/IP address pairs in its cache for entries older than IPCleanLifetime specifies. (The local memory cache on the DC Agent machine stores workstation-to-IP addresses mappings.) Entries older than the time period determined by the `IPCleanLifetime` interval are removed from the cache. The `IPCleanInterval` value typically does not need to be changed.

### IPCleanLifetime

The amount of time a workstation name/IP address pair remains in DC Agent’s cache before it is removed.

**Default Value**

7200 seconds (or 2 hours).

**Allowed Values**

Greater than 3600, or 0 to disable.

**Required**

No.

**Synopsis**

As DC Agent receives logon session information and converts workstation names to IP addresses, it stores the workstation name/IP address pairs in its local memory cache. This reduces the number of times DC Agent must perform DNS lookup for each active workstation, because it already has the IP address information for the current 2-hour period.

### MaxIgnoreListSize

The maximum number of entries in DC Agent’s optional **ignore.txt** file.

**Default Value**

70000 (user names, user name/workstation pairs, or workstation names)

**Allowed Values**

5000 or greater

**Required**

No.

**Synopsis**

You can optionally configure DC Agent to ignore particular users or workstations.  See *Configuring DC Agent to Ignore Particular User Names*, page 15 for details. The **ignore.txt** file can contain user names, workstation names, and/or user name/workstation pairs. This parameter sets a limit on the number of possible entries in this file. Typically, you do not need to change the default value. However, if you would like to further limit the number of users or workstations for DC Agent to ignore, you can decrease this value. The limit can be no lower than 5000.

### *password

The password DC Agent uses to authenticate the User Service when it contacts DC Agent. (Equivalent to **Password** in Websense Manager.)

**Default Value:**

N/A.

**Allowed Values**

Strings containing 4-16 characters.

**Required**

No.

**Synopsis**

The `password` parameter allows you to specify a password for authenticated connections to DC Agent from User Service. If specified, DC Agent uses the password to authenticate User Service when the service attempts to connect to DC Agent. If the password provided by User Service does not match this password value, then DC Agent does not transmit data to User Service. The password must be between 4 and 16 characters in length, and is case-sensitive. See Authenticated Connections on page *Authenticated Connections*, page 14 for information about configuring DC Agent and User Service to use an authenticated connection.

### *port

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

#### Default Value

30600.

#### Allowed Values

Between 1024 and 65535.

#### Required

Yes.

#### Synopsis

The port value is originally set during installation of DC Agent. The DC Agent installation program prompts for a port value, and then writes the value entered to the **transid.ini** file (see *Files Used in Transparent Identification with DC Agent*, page 12). The port you specify during installation is the port over which Filtering Service connects to DC Agent.

### *QueryInterval

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

#### Default Value

10 seconds.

#### Allowed Values

Between 5 and 90 seconds.

#### Required

No.

#### Synopsis

DC Agent periodically queries the domain controllers specified in its **dc_config.txt** file. Each query retrieves user logon session information from these domain controllers. Decreasing the query interval (to less than 10 seconds) may provide greater accuracy in capturing logon sessions, but it also increases overall network traffic. Greater accuracy is needed especially in cases as with Windows XP logon sessions, which are typically shorter than 15 seconds. 10 seconds has been determined to be an ideal interval for most cases. 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.

DC Agent uses DNS to translate workstation names to IP addresses, and then writes the resulting user name/IP address pairs to local memory on the DC Agent machine. (This data is also copied to the **XidDcAgent.bak** file on the DC Agent machine's hard disk.) Note that for user names specified in an

**ignore.txt** file, DC Agent does not record user name/IP address pairs to its map. See page 15 for information about creating an **ignore.txt** file.

### StartDelay

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

#### Default Value

0 seconds.

#### Allowed Values

Between 0 and 120 seconds.

#### Required

No.

#### Synopsis

The `StartDelay` parameter relates to use of the Websense ConsoleClient tool to troubleshoot DC Agent problems. This tool is used primarily by Websense Technical Support. Please contact Technical Support for assistance in using this tool.

To have ConsoleClient connect to DC Agent while the service is running, but before its processes are activated, `StartDelay` can be set to a particular value, to allow time for ConsoleClient to connect. This parameter is usually modified by Websense Technical Support personnel only. Use extreme caution when modifying this parameter; an incorrect value can overload network traffic.

### UseFileTrace

Whether to enable diagnostic file tracing for DC Agent.

#### Default Value

False.

#### Allowed Values

True, False.

#### Required

No.

#### Synopsis

Setting this parameter to `True` tells DC Agent to write diagnostic information about files DC Agent uses to a file named **xid_trace.txt**, in the **/bin** directory. Note that `UseFileTrace` must be *enabled* for the **VerifyTracing** parameter to have any effect. If **UseFileTrace** is set to `False`, `VerifyTracing` will also behave as though it is set to `False`.

### UseNetBIOS

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

#### Default Value

True.

#### Allowed Values

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, then DC Agent uses NetBIOS calls to identify domain controllers. Optionally, you can set `UseNetBIOS` to `False`. This causes DC Agent to rely solely on DNS, and not attempt to use NetBIOS at all. Typically, it is sufficient to leave the default value as is is.

### UseUserService

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

#### Default Value

True.

#### Allowed Values

True, False.

#### Required

No.

#### Synopsis

By default, in Websense versions 5.2 and later, DC Agent uses the Websense User service for communications with domain controllers in the network. For example, DC Agent identifies domain controllers in the network, and then queries those domain controllers for user logon sessions.

Typically, this parameter can be left as is. However, if you do not want to open the ports required for User Service to facilitate communications between DC Agent and domain controllers, you can set this value to False. For example, if you have a firewall placed between DC Agent and certain domain controllers, you might want to close certain communication ports. In this case, you can tell DC Agent to use Windows networking calls for communications instead, by setting `UseUserService` to `False`.

### VerifyTracing

Whether to enable diagnostic tracing of workstation polling routines.

#### Default Value

False.

#### Allowed Values

True, False.

#### Required

No.

#### Synopsis

If this parameter is enabled, DC Agent writes diagnostic information about its workstation polling processes to a file names **xid_trace.txt**, in the **/bin** directory. Note that in order for DC Agent to trace workstation polling, 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 Value

True.

#### Allowed Values

True, False.

#### Required

No.

#### Synopsis

If this parameter is enabled, DC Agent checks the existence of a user account against the domain where a user logon session is detected. If `VerifyUserDomain` 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**) contains only one parameter. In Websense Enterprise versions 5.5 and 6.1, parameters controlling how Logon Agent processes run can be managed directly from Websense Manager. Please see the *Websense Enterprise Administrator's Guide* or online Help for descriptions of these parameters, and how to modify them.

### UserServerWaitTime

This parameter ensure that the Websense User Service is running before Logon Agent starts. Logon Agent cannot communicate data to Filtering Service if User Service is not running. By default, this value is set to 1 second. If this value is set to 0, Logon Agent will start even if User Service is down. This value should not need to be changed.

## RADIUS Agent

The RADIUS Agent initialization file (**wsradius.ini**) contains several parameters that determine how RADIUS Agent communicates with other RADIUS components, and how various RADIUS Agent processes run. All parameters and values described here are case-sensitive.

In Websense Enterprise versions 5.5 and 6.1, you can modify many of the default RADIUS Agent parameters after installation, via Websense Manager. Please see the *Websense Enterprise Administrator's Guide* for instructions. Some parameters can be modified either via Manager or in **wsradius.ini**; these parameters are marked with an asterisk (*).

### *AccInPort

Port over which RADIUS Agent accepts accounting requests. (Equivalent to **Accounting Ports/ From RADIUS clients to RADIUS Agent** in Websense Manager.)

#### Default Value

12346.

#### Allowed Values

Any valid, available port number.

#### Required

No.

#### Synopsis

If your RADIUS environment is configured to support RADIUS accounting, the Websense RADIUS Agent receives accounting requests from client machines over this port.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### *AccOutPort

Port over which the RADIUS server listens for RADIUS accounting messages. (Equivalent to **Accounting Ports/From RADIUS Agent RADIUS server** in Websense Manager.)

**Default Value**

1646.

**Allowed Values**

Any valid, available port number.

**Required**

No.

**Synopsis**

If your RADIUS environment is configured to support RADIUS accounting, the RADIUS server receives accounting messages from client machines over this port.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### *AuthInPort

Port over which RADIUS Agent accepts authentication requests. (Equivalent to **Authentication Ports/From RADIUS clients to RADIUS Agent** in Websense Manager.)

**Default Value**

12345.

**Allowed Values**

Any valid, available port number.

**Required**

Yes.

**Synopsis**

The Webense RADIUS Agent receives authentication requests from the RADIUS client, as users log on to the network. RADIUS Agent must be configured to listen over a particular port number designated for RADIUS authentication requests.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### *AuthOutPort

Port on which the RADIUS server listens for authentication requests. (Equivalent to **Authentication Ports/From RADIUS Agent to RADIUS server** in Websense Manager.)

**Default Value**

1645.

**Allowed Values**

Any valid, available port number.

**Required**

Yes.

**Synopsis**

The Webense RADIUS Agent processes the authentication requests it receives from the RADIUS client, and then forwards them to the RADIUS server over this port.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### DebugLevel

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

**Default Value**

0.

**Allowed Values**

0, 1, 2, 3.

**Required**

No.

**Synopsis**

This is a number between 0 and 3 specifying the level of detail for log messages, where 0=least detail, and 3=most detail. 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. See page 55 for more details on using the built-in RADIUS Agent diagnostic tool.

### DebugMode

Controls the RADIUS Agent diagnostic activity.

**Default Value**

Off.

**Allowed Values**

On, Off (case-sensitive).

**Required**

No.

**Synopsis**

Enables or disables RADIUS Agent's built-in diagnostic (logging and debugging) capabilities. This can be a valuable tool for troubleshooting user identification problems, and determining whether RADIUS Agent is identifying remote users correctly. You can enable diagnostic activity during RADIUS Agent setup, or later on if desired. See page 55 for instructions on modifying the `DebugMode` value after installation.

### LogFile

Output file for RADIUS Agent diagnostic messages.

**Default Value**

None.

**Allowed Values**

Any string of characters valid in your operating system.

**Required**

No.

**Synopsis**

If you have set `DebugMode=On`, specify a filename for the text file where RADIUS Agent should send diagnostic (log) output.

### *RADIUSHost

Machine name or IP address of the RADIUS server machine. (Equivalent to RADIUS server in Websense Manager.)

#### Default Value

None.

#### Allowed Values

Valid IP address in the format N.N.N.N.

#### Required

Yes.

#### Synopsis

The Websense RADIUS Agent forwards authentication and accounting requests to the RADIUS server machine. RADIUS Agent must know the location of the RADIUS server in order to communicate with it.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### *RRASHost

IP address of a machine running Microsoft RRAS. (Equivalent to **RRAS Machine** in Websense Manager.)

#### Default Value

“”

#### Allowed Values

A valid IP address in the format N.N.N.N.

#### Required

No.

#### Synopsis

*(Windows only.)* If Microsoft RRAS is in use, Websense queries the machine running Microsoft RRAS for user logon sessions. Websense needs the IP address of this machine in order to perform the queries. An empty value means that no query occurs.

This parameter is not present in **wsradius.ini** by default, but it can be added – for example, to configure a particular instance of RADIUS Agent differently.

### Timeout

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

#### Default Value

1000 milliseconds (1 second).

#### Allowed Values

Numbers 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. Lowering this value to decrease the timeout interval may decrease accuracy in identifying users. A value of 500 (.5 seconds) is the lower limit, designed to maximize success in capturing responses from the RADIUS server.

## eDirectory Agent

The eDirectory Agent initialization file (**wsedir.ini**) contains several parameters that determine how the Websense eDirectory Agent communicates with the Websense Filtering Service, and how various eDirectory Agent processes run. All parameters and values described here are case-sensitive.

You can modify many of the default eDirectory Agent parameters after installation, via Websense Manager. Please see the *Websense Enterprise Administrator's Guide* for instructions. Some parameters can be modified either via Manager or in **wsedir.ini**; these parameters are marked with an asterisk (*).

### DebugLevel

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

**Default Value**

0.

**Allowed Values**

0, 1, 2, 3.

**Required**

No.

**Synopsis**

This is a number between 0 and 3 specifying the level of detail for log messages, where 0=least detail, and 3=most detail. Any value outside the range of 0-3 is interpreted as 0. Diagnostic output with a detail level of 3 includes all transactions involved in a user logon. See page 56 for more details on using the built-in eDirectory Agent diagnostic tool.

### DebugMode

Controls the eDirectory Agent diagnostic activity.

**Default Value**

Off.

**Allowed Values**

On, Off (case-sensitive).

**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. You can enable diagnostic activity during eDirectory Agent setup, or later on if desired. See page 55 for instructions on modifying the `DebugMode` value after installation.

### *DN

Novell eDirectory server administrator name. (Equivalent to **eDirectory Administrator Full Qualified Domain Name** in Websense Manager).

#### Default Value

None.

#### Allowed Values

Any string of characters.

#### Required

Yes.

#### Synopsis

This is the distinguished name of a user with administrative rights in Novell eDirectory server. Novell eDirectory requires an authenticated name in order to issue LDAP requests. The name should match the DN specified in Websense Manager, via **Server** > **Settings** > **Directory Service**.

This parameter is not present in **wsedir.ini** by default, but it can be added – for example, to configure a particular instance of eDirectory Agent differently.

### LogFile

Output file for eDirectory Agent diagnostic messages.

#### Default Value

None.

#### Allowed Values

Any string of characters valid in your operating system.

#### Required

No.

#### Synopsis

If you have set `DebugMode=On`, specify a filename for the text file where eDirectory Agent should send diagnostic (log) output.

### *password

Novell eDirectory server administrators password. (Equivalent to **eDirectory Administrator Password** in Websense Manager.)

#### Default Value

None.

#### Allowed Values

Any string of characters.

#### Required

Yes.

#### Synopsis

This is the password for your Novell eDirectory server administrator account. This password should match the one specified in Websense Manager, via **Server** > **Settings** > **Directory Service**.

This parameter is not present in **wsedir.ini** by default, but it can be added – for example, to configure a particular instance of eDirectory Agent differently.

### PollInterval

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

#### Default Value

30000 milliseconds (or 30 seconds).

#### Allowed Values

Any number of milliseconds.

#### Required

Yes.

#### Synopsis

You can opt to have eDirectory Agent wait longer than the default interval between queries of Novell eDirectory server, or query the server more frequently. 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.

### *SearchBase

Novell eDirectory server root context. (Equivalent to **eDirectory Server Search Base** in Websense Manager.)

#### Default Value

None.

#### Allowed Values

Any string of characters.

#### Required

Yes.

#### Synopsis

This is the Distinguished Name, or DN, of your Novell eDirectory root context. This root context should match the one specified in Websense Manager, via **Server > Settings > Directory Service**.

This parameter is not present in **wsedir.ini** by default, but it can be added – for example, to configure a particular instance of eDirectory Agent differently.

### Server

IP addresses or names of machines running Novell eDirectory.

#### Default Value

None.

#### Allowed Values

A valid IP address in the format N.N.N.N, or a string of alpha-numeric characters.

#### Required

Yes.

#### Synopsis

The Websense eDirectory Agent needs to know the identity of any machine running Novell eDirectory in order to query the directory service. If you are running multiple instances of Novell eDirectory, you can specify multiple servers, one line at a time. You can add, remove, or modify servers after installation by editing the **wsedir.ini** file.