V.10

Cerberus, LLC

[CERBERUS FTP SERVER 10.0]

User manual for Cerberus FTP Server 10.0. Detailed steps and help on configuring Cerberus FTP Server.

Contents

INTRODUCTION

DESCRIPTION

Cerberus FTP Server provides a secure and reliable file transfer solution for the demanding IT professional or the casual file sharer. Supporting SFTP, FTP/S, and HTTP/S, Cerberus is able to authenticate against Active Directory and LDAP, run as a Windows service, has native x64 support, includes a robust set of integrity and security features and offers an easy-to-use manager for controlling user access to files and file operations.

GUIDE

For additional help and troubleshooting information, take a look at the Cerberus FTP Server FAQ.

You can also access the most recent help documentation online.

MINIMUM SYSTEM REQUIREMENTS

This section describes the minimum hardware and software requirements to install and run Cerberus FTP Server.

HARDWARE REQUIREMENTS

- 2GHz x86 or x64 processor
- 2 GB RAM (4 GB or higher recommended
- WXGA (1280 x 768) or higher-resolution monitor

OPERATING SYSTEMS

Note: The latest Service Packs for your operating system are required in all cases.

CERBERUS FTP SERVER 10.0

- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008 and R2
- Windows Server 2012 and R2
- Window Server 2016

The latest Service Packs for your operating system are **highly** recommended.

Close all other programs (recommended) before installing Cerberus FTP Server and make sure that you install it logged in as Administrator or a member of the Administrators group if you are installing it on a Windows Server system.

1.   Download the latest version of Cerberus FTP Server from http://www.cerberusftp.com/download/

We recommend the 64-bit version of Cerberus FTP Server.

2.   Double click or run the **CerberusInstall64.exe** self-extracting installer.

3. You will see the initial setup screen. To continue you will need to check the box to confirm your acceptance of the licensing agreement. Select the "**I agree to the License terms and conditions**" checkbox and click **Install.**
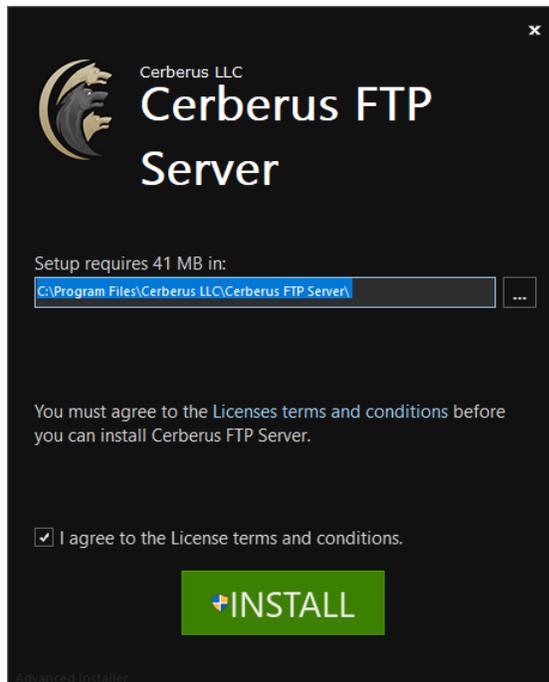
3. Wait for the installer to finish.

4. Click **Finish** or press the **Run** button to launch the Cerberus FTP Server Administration Tool.

UPGRADING AN EXISTING INSTALLATION

There are two methods for upgrading an existing installation of Cerberus FTP Server. You can use the built-in auto-updater, or you can download the latest installer and run it to manually upgrade your installation. Both methods are discussed below.

When upgrading using either method, the installer first stops the Cerberus FTP Server service, uninstalls the existing Cerberus FTP Server installation, then installs the latest release. The uninstallation only affects the actual program files. The server configuration and user settings are never removed.

An upgrade usually takes about 5 minutes, and almost never requires a reboot. The Cerberus FTP Server service will be unavailable during the upgrade.

NOTE: We always recommend making a backup of your users and settings from the Cerberus **Tools** menu before upgrading. Select the **Backup Users and Settings** option to create a zip file of all of your Cerberus settings and users.

METHOD 1: USING THE AUTO-UPDATER

The Cerberus FTP Server automatic updater will check for a new release of Cerberus FTP Server, and allow you to download and run the updater to upgrade your installation. To check for an update and, optionally, install it:

1. Select the **Help** menu option from the main menu.
2. Select the **Check for Update** menu option.
3. You will see the **Upgrade Check** dialog. It will list the current version installed, including whether you have the 32-bit or 64-bit version installed, and any updates that are available.
4. If no updates are available, the Current Version and Latest Version will be the same, and the **Update** button will be disabled. If this is the case, you have the latest release and can press the **Close** button to end the update process.
5. If a new release is available, the **Update** button will become enabled and a list of changes since your current version was released will be listed in the release notes list box.
6. Press the **Update** button to automatically download the latest release and begin the update process.
7. Once the download has completed, the Cerberus UI will close and shutdown and the installer will automatically launch. You should select the default options for any questions in the installer. The installer will automatically remove your existing installation (users and settings are never removed during an uninstallation) and then install the latest release. All of your existing users and settings will be preserved.
8. Finish the installation and you are done. The latest version of Cerberus FTP Server should now be installed and running.

You can manually download the latest installer and use it to upgrade your installation if you cannot use the auto-updater. The installer will upgrade an existing installation to the latest release. To download and run the latest installer:

1. Go to the Cerberus Download page
2. Download the latest 32-bit or 64-bit installer.
3. Shutdown the Cerberus FTP Server UI. Go to the File menu and select the Exit menu option. You should also shut down the Cerberus FTP Server Window Service. The installer will normally be able to shut down the service, but on rare occasions, the automatic shutdown will not work. Shutting down the Cerberus Windows Service before installation ensures a restart will not be necessary after the installation completes.
4. Launch the installer. You should select the default options for any questions in the installer. The installer will automatically remove your existing installation and install the latest release. All of your existing users and settings will be preserved.
5. Finish the installation and you are done. The latest version of Cerberus FTP Server should now be installed and running.

GETTING STARTED - INITIAL SETUP WIZARD

THE WIZARD

The Getting Started Wizard will appear when you start Cerberus FTP Server for the first time. The wizard is designed to walk you through the basic steps of configuring the server to allow clients to connect. At the end of the Getting Started Wizard, your server should be ready to accept connections from FTP, FTPS, SSH SFTP, and HTTP clients.

STEP 1 - LICENSING

The Licensing page allows the administrator to select the licensing option most appropriate for their intended use of Cerberus FTP Server. There are two options:

- Selecting "**As a Company, Government entity, or Educational institution"** enables a 25 day trial period of the Enterprise edition of Cerberus FTP Server. During the trial period, the server will perform and function as the Enterprise edition. Cerberus FTP Server reverts to the Home edition after the evaluation period expires and a message indicating that the server is unregistered will be added to the server welcome message for each connection. At any time, including after the trial period has expired or even if "For Personal Use" was selected at startup, Cerberus may be turned into the full commercial Personal, Standard, Professional, or Enterprise edition by entering a valid registration code into the license dialog.

- Selecting the **"For Personal, Home Use Only"** option immediately causes Cerberus to function as the Home edition. This license is only permitted for at home, personal use of the FTP server. The Home edition is limited to at most 5 simultaneous FTP or FTPS connections. A message indicating that the server is Cerberus FTP Server Home edition will also appear in the FTP welcome message whenever a client connects to the server. In all other respects, Cerberus FTP Server Home edition is functionally equivalent to the licensed Personal edition.
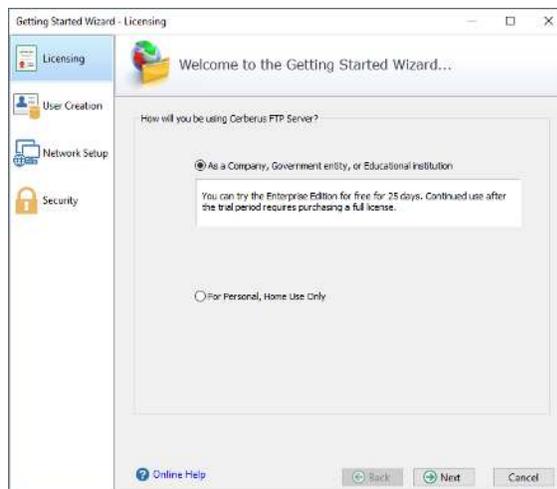
Licensing Page of the Getting Started Wizard

The User Creation page will allow you to automatically create a simple user account with access to a directory on the local machine. You can use this account to test out your initial connection to the server. You can turn off the creation of the user account by un-checking the "Create an Initial User?" checkbox.
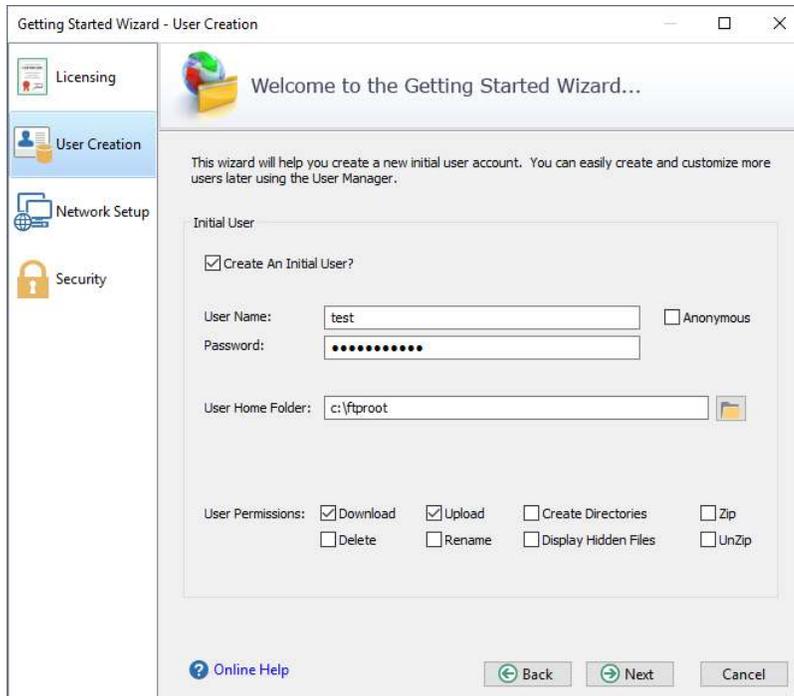
You can create an *anonymous* user that will be created under the User Manager.

The *anonymous* user will have download and upload-only access to the "C:\ftproot" directory as their root drive. This directory will be created if it does not already exist.

Please note, creating an anonymous user allow anyone to connect to your FTP server without specifying a password. Using the default settings, anyone can view and download any file from your "C:\ftproot" directory and any subdirectories of that directory.

You can further customize the newly added user, or create and manage additional users, through the User Manager after the "Getting Started" wizard has finished.

The Network Setup page detects basic network settings and tries to provide advice on any changes that may need to be made because of the computer's network configuration.
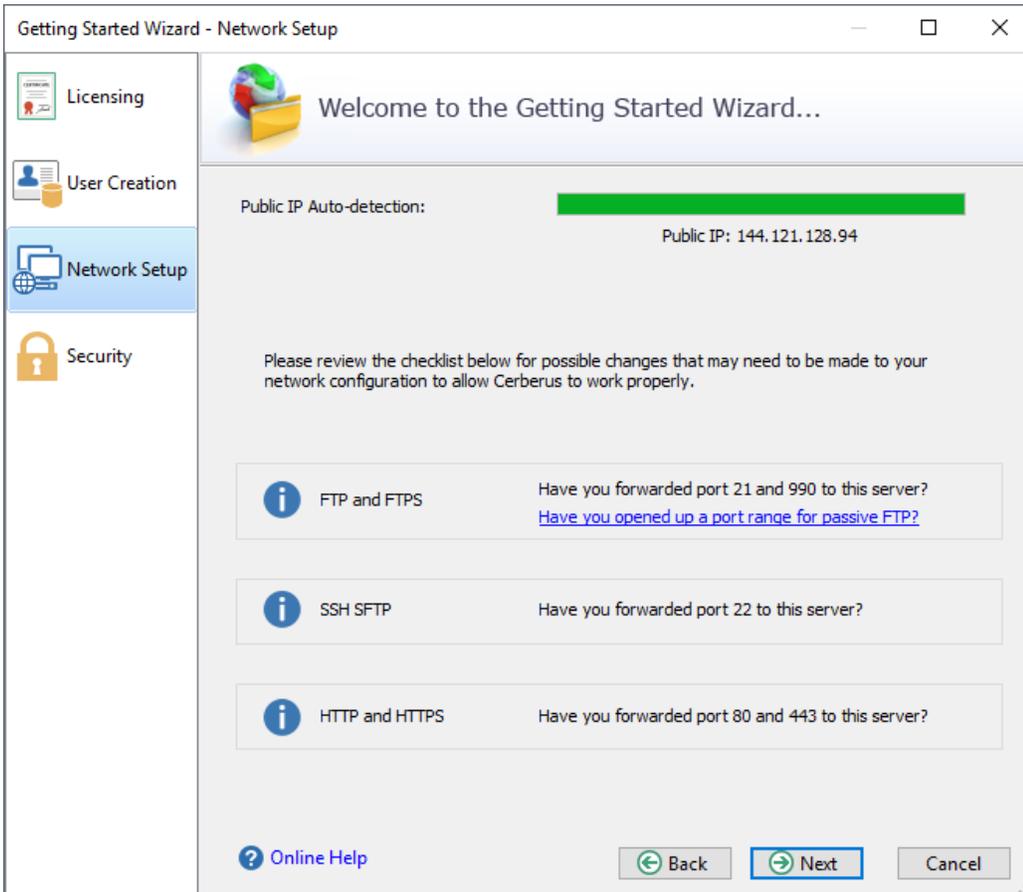
PUBLIC IP AUTO-DETECTION FOR PASSIVE MODE FTP

The most complex task in configuring basic FTP access to your server is preparing the machine to accept FTP data connections. Unlike SSH SFTP or HTTP/S protocols, FTP is complicated by the need for two connections for each client session. The first connection is established when the client initially connects and is used to exchange commands and status between the FTP server and the client. A second connection is created every time a directory listing or file transfer takes place. Whenever a directory listing or file transfer is requested, the FTP server has to respond with an IP address and port that the client can connect over to establish the secondary data connection. To aid the server in determining what IP address to give to the client, the server can be configured to automatically detect the IP address of the server on the internet and use this IP address when sending the client connection instructions.
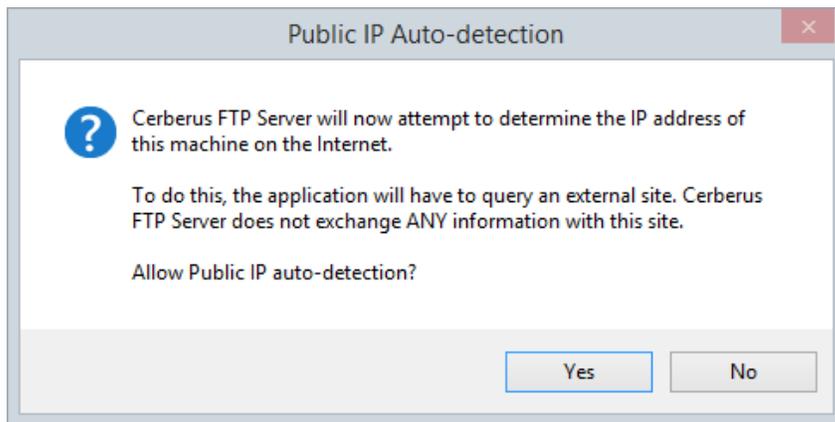
After clicking the Next button on the Network Setup page a dialog prompt will ask whether you want to allow Cerberus to automatically attempt to detect your public IP address. We normally recommend you answer **Yes** here. Answering yes will instruct Cerberus to automatically attempt to detect and use the correct external IP address when clients request passive FTP data connections.

STEP 4 – SECURITY

The last page of the Getting Started Wizard will allow the administrator to configure a few basic server security settings.
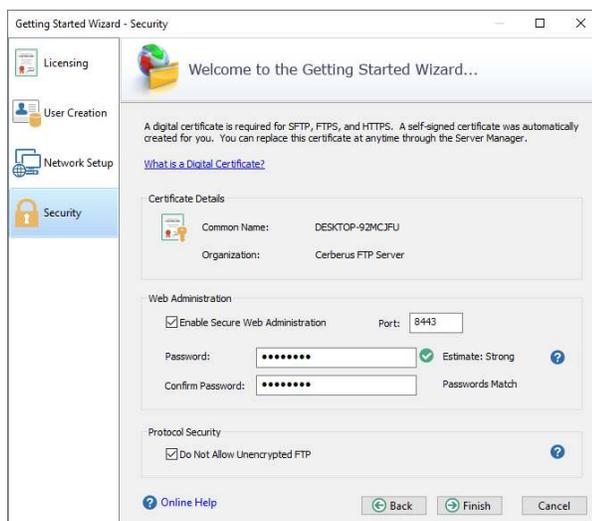
Cerberus FTP Server fully supports TLSv1/SSLv3 encryption over FTP (FTPS), HTTPS, and SSH SFTP.

To enable FTPS, HTTPS, and SSH SFTP support, a digital certificate must be generated for the server. This digital certificate contains the necessary security data to allow the server to establish encrypted connections with clients.

Cerberus FTP Server will automatically generate a new, self-signed certificate for you the first time you run the Getting Started Wizard. You can replace the certificate at any time through the Security page of the Server Manager.

Web Administration Password

You also have the option to configure a web administration and remote API access password on the Security Wizard page. You should set a strong password here even if you are not using web administration. Please note that the password strength estimation meter is only meant as a guide. It will flag obviously poor passwords but there is no official weighting system and this meter should only be utilized as a loose guide to improving your password.

Protocol Security

The last option allows you to configure the server to only accept encrypted FTP connections. Normal FTP has no encryption and therefore allows passwords and data to be transmitted unencrypted over a network.

Fortunately, it is possible to establish a normal unencrypted FTP connection and then "upgrade" the connection to secure encryption through special FTP commands (this enhanced protocol is called FTPES). This type of connection depends on the client issuing FTP commands instructing the server to establish encryption before accepting login credentials. However, the client can also continue as a normal FTP connection without enabling encryption. This situation allows for unencrypted connections and presents a security issue for servers.

If you wish to allow FTPES secure connections, but not FTP, then you must instruct the server to require encryption before allowing a connection to proceed.

Checking this option does exactly that. It requires the client to upgrade the connection to use encryption before allowing login.

Final Steps

Click the Finish button to complete the Getting Started Wizard. Your server is now ready to accept local network FTP/S, SSH SFTP, or HTTP/S web client connections. Please take a look at the next section for any changes that might need to be made to your firewall or router to allow connection from outside of your local network to reach your server.

GETTING STARTED - NETWORK SETUP

BASIC SETUP SO USERS CAN CONNECT FROM THE INTERNET

FTP connections within your local network usually work without any problems. However, when you want the FTP server to be available outside of your local network, additional steps are often necessary to make the server visible to the outside world. The following steps are usually required to allow Cerberus FTP Server to be accessed from the Internet:

STEP 1 - CONTROL CONNECTION

The control connection port Cerberus FTP Server is listening on needs to be forwarded from your router to the machine hosting Cerberus. The default port that Cerberus listens on is port 21. Consult your router documentation for instructions on how to setup port forwarding. Finishing this step will allow Internet users to establish a connection with your server. The next step is making sure **passive mode** is configured so that directory listings and file transfers work.

STEP 2 - PASSIVE MODE

To allow passive mode to work properly, you must forward the passive range of ports from your router to the machine running Cerberus. See "My IP address begins with 192.168.xxx.xxx. Is there anything special I have to do for people to see my FTP Server on the Internet?" for detailed instructions on how to make sure passive mode is set up properly. If you don't perform this step, users may be able to log in but directory listings may hang and timeout.

STEP 3 - FIREWALL

Make sure any firewalls you are running are allowing connections on port 21. Cerberus will automatically attempt to add itself to the Windows Firewall Exception list (you will be prompted to allow this). However, you may still have to manually add an exception to allow port 21 connections into your computer.

HOW MANY TYPES OF FTP ARE THERE?

There are three types of FTP connections possible (Cerberus FTP Server supports all three):

FTP: Plain, unencrypted FTP that defaults over port 21. Most web browsers support basic FTP.

FTPS: Implicit SSL/TLS encrypted FTP that works just like HTTPS. Security is enabled with SSL as soon as the connection starts. The default FTPS port is 990. This protocol was the first version of encrypted FTP available, and while considered deprecated, is still widely used. None of the major web browsers support FTPS.

FTPES: Explicit FTP over SSL/TLS. This starts out as plain FTP over port 21, but through special FTP commands is upgraded to TLS/SSL encryption. This upgrade usually occurs before the user credentials are sent over the connection. FTPES is a somewhat newer form of encrypted FTP (although still over a decade old), and is considered the preferred way to establish encrypted connections because it can be more firewall friendly. None of the major web browsers support FTPES.

CONTROLLING WHAT TYPES OF FTP ARE ALLOWED

You can control the types of FTP connections allowed at both the user level, and at the listener level.

RESTRICTING FTP CONNECTIONS AT THE USER LEVEL

For a user or group account, the **Require Secure Control** and **Require Secure Data** constraints are meant to enforce that the connection is encrypted using either FTPS or FTPES. If Require Secure Control is checked, FTP over port 21 will be denied login if the user attempts to authenticate without upgrading the connection to use encryption. If the FTP connection is upgraded to use encryption (upgraded to FTPES), then the user will be allowed to send login credentials and attempt to login. Cerberus requires an FTP listener to allow FTP or FTPES connections.

FTPS connections are always encrypted, and connections that come through on an FTPS listener will always be allowed to attempt to login.

The user and group constraints **Allow FTP** and **Allow FTPS** is meant to control what protocol a user can login over. If **Allow FTP** is selected for a user, then both FTP and FTPES connections will be allowed to attempt to login over an FTP listener. This can be further restricted to only allowing FTPES connections by selecting the **Require Secure Control** and **Require Secure Data** constraints for the user.

You can create combinations of these options to allow exactly the type of protocol and security settings that you prefer.

To allow any protocol, as long as it is secure, leave **Allow FTP** and **Allow FTPS** checked, and make sure **Require Secure Control** and **Require Secure Data** are checked.

This will allow connecting over implicit FTPS listeners on port 990, and explicit FTPES connections over FTP listeners on port 21 (as long as the connection gets upgraded to TLS/SSL encryption before the user attempts to login).

RESTRICTING FTP CONNECTIONS AT THE LISTENER LEVEL

In addition to the fine-grain control, administrators have at the user level, broader restrictions can be enforced at the listener level. FTP listeners also have the **Require Secure Control** and **Require Secure Data** settings. These settings are checked first, before a user even attempts to login. If the **Require Secure Control** and **Require Secure Data** options are specified for an FTP listener, then only secure FTPES connections will be allowed. These settings are enforced before the individual user settings are checked.
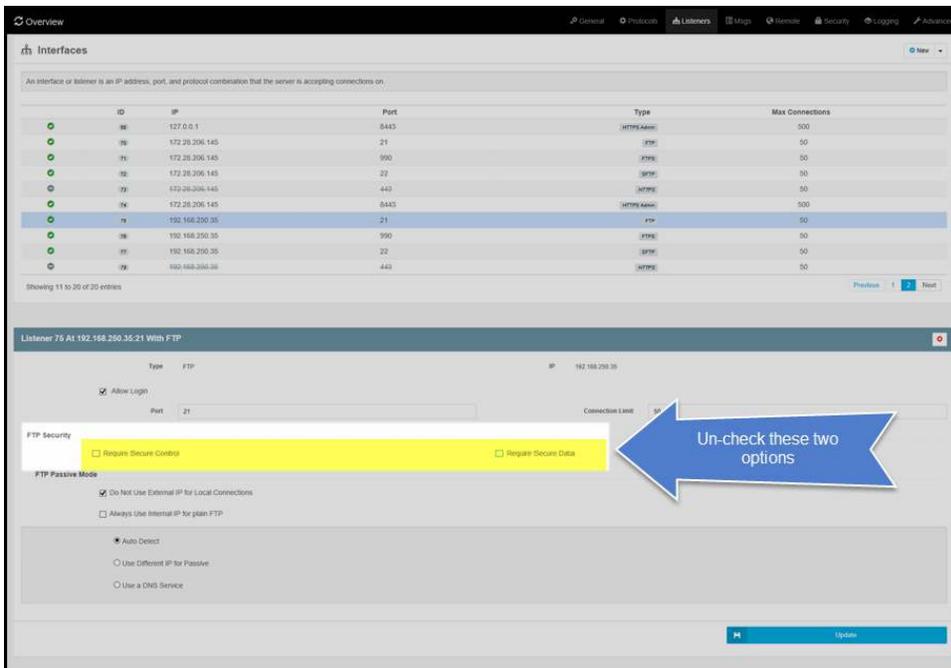
SSH2 SFTP Setup

About SSH SFTP Support in Cerberus FTP Server

Cerberus FTP Server Professional edition and higher supports the SSH2 File Transfer Protocol, also known as SFTP. SFTP is a network protocol that provides secure and reliable file access, file transfer, and file management functionality. Features of the protocol include resuming interrupted file transfers, directory listings, getting and setting file attributes, and remote file removal.

There are currently 6 different versions of the SFTP protocol, with versions 3 - 6 being in common use by modern SFTP clients. Cerberus supports SFTP version 3, 4, 5, and 6 clients.

Cerberus also supports SSH public key authentication.

Supported SSH2 Key Exchange Methods

Cerberus supports both Diffie-Hellman and Elliptic Curve Diffie-Hellman (ECDH) SSH2 key exchange methods. The following exchange methods are supported:

- diffie-hellman-group1-sha1
- diffie-hellman-group14-sha1
- diffie-hellman-group-exchange-sha1
- diffie-hellman-group-exchange-sha256
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521

Supported SSH2 Ciphers

The following SSH ciphers are supported:

- des **(disabled by default)**
- 3des-cbc
- aes256-cbc
- aes192-cbc
- aes128-cbc
- aes256-ctr
- aes192-ctr
- aes128-ctr

The following SSH MAC algorithms are supported:

- hmac-sha1
- hmac-sha1-96
- hmac-sha2-256,
- hmac-sha2-256-96
- hmac-sha2-512
- hmac-sha2-512-96
- hmac-ripemd160 (disabled in FIPS mode)
- hmac-ripemd160@openssh.com (disabled in FIPS mode)

ADDING AN SSH2 SFTP LISTENER

You must first have at least one SFTP listener for Cerberus to be able to accept SFTP connections. Cerberus FTP Server will automatically add and enable SFTP listeners on each available IP address the first time it is run so you normally do not need to add an SFTP listener. However, if you've previously removed an SFTP listener you can add a new one from the Interfaces page of the Server Manager.

To add a new SFTP listener:

1. Open the Server Manager
2. Select the **Interfaces** page
3. Select the "plus" icon next to the interface list box to add a new interface. The "Add New Listener" dialog box will appear to ask for the interface details (interface IP, type, and port combination)
4. Select the IP address that you want to listen for connections on
5. Select the **SSH SFTP** interface type
6. Enter the port you wish to listen on (the default for SSH2 SFTP is 22). Cerberus will automatically pre-populate the port with the default port for the type of listener you are adding
7. Press the **Add** button to add the listener
8. The listener should now be added to the Interfaces list. Press **Ok** to close the Server Manager and save your changes.

ALLOWING SSH2 SFTP CONNECTIONS THROUGH A FIREWALL

SFTP connections use port 22 by default. You may need to allow that port through your firewall to the machine running Cerberus FTP Server. You may also need to make sure your router is forwarding incoming connections on that port to the machine running Cerberus FTP Server.

ENABLING OR DISABLING EXISTING SFTP LISTENERS

In addition to adding and deleting interfaces, Cerberus allows an administrator to disable or enable an existing interface. This feature can be used to temporarily disable a listener or to re-enable a listener that has become disabled because of a port conflict or trial license expiration.

ALLOWING EXTERNAL ACCESS TO YOUR SERVER

Depending upon your connection to the Internet, you may need to configure your router or firewall before users outside of your local network can see your FTP server. Communication with an FTP server is done through two connections, a control connection, and a data connection. Ensuring these connections can be established are the two areas where special attention is usually needed.

THE CONTROL CONNECTION

The control connection is always the first connection established with an FTP server. The control connection's purpose is to allow clients to connect and to send commands to the server (and receive server responses). Port **21** is considered the default control connection port, and this is the default port that Cerberus FTP Server will configure your IP interfaces to listen on for new connections. Using the default port is not mandatory - the administrator is free to change the interface to use any free port on the system as the listening port. However, if the administrator is running a software-based firewall, the administrator must be certain that [incoming] connections are not blocked on the port chosen for the control connection. If the port that Cerberus is listening on is blocked, no one will be able to see or connect to the FTP server.

THE DATA CONNECTION

The second type of connection is called the data connection. This is the connection that an FTP server uses to exchange file listings and transfer files on. When an FTP client uses the control connection to instruct Cerberus FTP Server to send a file listing or transfer a file, the actual data exchange takes place on the data connection. The data connection is usually where most of the confusion and problems arise for FTP server administrators.

There are two different ways a data connection can be established between an FTP client and an FTP server. The first is commonly called **active** FTP. In this mode, an FTP client sends the IP address and port that the client is currently listening for data connections on to the FTP server. The client accomplishes this by sending the server a *PORT* command over the control connection. Using the address and port from the *PORT* command, the FTP Server then connects to the client and sends the file or file listing. When using **active** FTP, the administrator has to make sure that port 20 on the machine that Cerberus FTP Server is running on is open for outgoing connections. The reason for this is because when using **active** FTP, the server always establishes connections from port 20. Most firewalls allow outgoing connections automatically, so manually opening up port 20 for outgoing connections is usually not necessary.

The other way to establish a data connection between client and server is to use **passive** FTP. **Passive** mode was introduced to get around common problems with client firewalls. Instead of the FTP server connecting to the FTP client, the client connects to the FTP server using a port previously communicated using the *PASV* command. When a client issues the *PASV* command, the FTP server responds with a port that the server is currently listening on for data communication. Problems occur with **passive** FTP when the firewall that Cerberus FTP Server is running on is blocking the selected ports. To get around this problem, the administrator is required to open up the range of ports that

Cerberus has reserved for **passive** FTP connections. You can configure what range of ports Cerberus FTP Server uses for **passive** FTP mode by looking under the 'Advanced' tab of the Server manager.

Failures during *LIST, NLST, MLST, RETR*, or *STOR* operations can usually be attributed to problems with the data connection.

COMMON NETWORK CONFIGURATIONS

A PC running Cerberus FTP Server with access to the Internet often fits into one of two configurations:

CONFIGURATION 1: YOUR COMPUTER IS CONNECTED DIRECTLY TO THE INTERNET

This is the simplest network configuration you can have and usually requires little or no configuration to Cerberus FTP Server to allow full access. This configuration is most common with dial-up, DSL, cable modem, and other broadband users. However, machines connected to the Internet directly often employ a software firewall to provide some protection against unwanted intrusion attempts. While some firewall software can automatically detect an FTP server and properly configure itself, the administrator usually has to manually configure the firewall. See the explanation above about the control and data connection for common ports that have to be allowed through a firewall.

CONFIGURATION 2: YOUR COMPUTER IS CONNECTED TO A ROUTER, AND THE ROUTER IS CONNECTED TO THE INTERNET

Routers usually act as firewalls, so the same problems that can occur in Configuration 1 can occur here. Follow the advice in Configuration 1 to resolve firewall problems.

In addition to the firewall problems that can occur in this network configuration, there is now the problem that the IP address you are using on your machine is not the IP address that the Internet sees for your machine. Other users on the Internet usually see your router's IP address instead of your PC's private address. Routers are devices on your network, just like your PC, and they have their own IP address, and that is the IP address the router tells other computers is your address when you go out on the Internet. When a user attempts to connect to the FTP server, they need to use the Internet-facing IP address of the router (the router is where the connection is really happening), not the private address of the computer Cerberus FTP Server is running on. When the router receives the connection attempt it is then able to forward the connection to your computer.

The first thing to check in this configuration is that your router is sending all of the FTP traffic to the computer Cerberus FTP Server is running on. Most routers have a web-based configuration utility that you can use to configure **Port Forwarding**. Specifically, you will want to make sure you forward the control and possible data connection ports to the computer running Cerberus FTP Server.

There is one more problem that crops up in this network configuration. To properly allow **passive** transfer mode, the administrator will have to make sure Cerberus is giving out the router address in response to PASV requests. You can automatically enable this by making sure "WAN IP Autodetection" is enabled in the 'General' tab of the Server Manager. Alternately, you can enter the IP address of the

router manually for each interface in the "Use different IP for PASV mode" IP box under the Server manager's 'Interfaces' tab.

While more complicated network configurations are possible, most users will fall into one of the above configurations.

UNDERSTANDING THE SUMMARY VIEW

The Summary View provides the administrator with a one-page overview of the server's configuration and any potential security issues that may be present.

Every time a configuration change is made the server scans the current Cerberus configuration at startup to look for any potential security issues that might result from the current system configuration. System warnings and messages are displayed in the System Messages list and each protocol type is given an overall security status indicator.
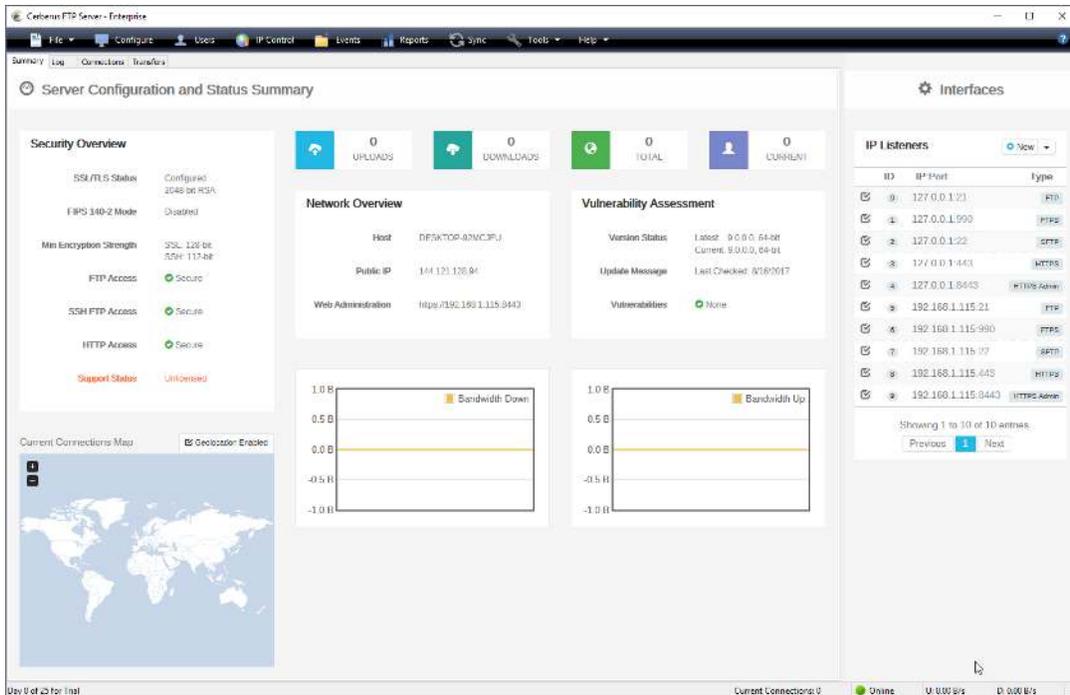
Cerberus FTP Server Summary View

The possible status for each protocol type is:

Secure	All listeners currently active for this protocol type are configured to accept only encrypted connections.
Not Secure	Some or all listeners currently active for this protocol type are configured to allow unencrypted connections.
Disabled	There are no listeners currently active on the server for this protocol.

Common System Messages

There are generally two types of system messages displayed in the System Messages list - general messages and security messages.

Anytime a protocol is listed as Not Secure there will be a system security message detailing the reason. Common system messages, their explanation, and resolution, if applicable, are detailed below.

"FTP Listener X can allow unencrypted control or data connections"

Explanation: Normal FTP has no encryption and therefore allows passwords and data to be transmitted in the clear over a network. To address this security issue, two secure forms of FTP were developed called implicit FTPS and explicit FTPES. Implicit FTPS is very similar to HTTPS and takes place on a completely separate port from typical FTP. Interfaces of this type are always encrypted and considered secure. Explicit FTPES, however, starts on a normal unencrypted FTP connection and is then "upgraded" to a secure connection through special FTP commands. This type of connection depends on the client issuing commands instructing the server to enable encryption. However, the client can also continue as a normal FTP connection without enabling encryption. This situation allows for unencrypted connections and presents a security issue for servers.

Resolution: To resolve this issue and still allow FTP access there are two possible solutions. One is to remove all FTP listeners and only enable FTPS listeners. FTPS listeners only accept encrypted communications and are considered secure.

If you wish to also allow FTPES secure connections then you must instruct the server to require encryption before allowing a connection to proceed. To require the FTP listener to require encryption, go to the Interfaces page of the Server Manager and for each FTP interface, select the Require Secure Control and Require Secure Data options.

For more detailed information, please take a look at our information page describing the different forms of FTP and secure FTP.

"HTTP Listener X only accepts unencrypted connections"

Explanation: Connections of type HTTP are always unencrypted and are therefore very susceptible to inspection on a network. System administrators are encouraged to disable HTTP listeners in favor of secure HTTPS listeners.

Resolution: To resolve this issue the system administrator must disable any HTTP listeners in the system, or set the redirect to HTTPS flag on the HTTP listener to make sure the connection is immediately redirected to HTTPS. HTTPS listeners will not trigger a security issue.

"HIPAA Non-compliance: One or more listeners allows non-encrypted traffic"

Explanation: HIPAA requires all data to be encrypted before being sent over a network. You have an active listener that allows data to be transmitted without encryption.

An FTP listener without the Require Secure Control and Require Secure Data settings will trigger this warning. An HTTP listener that is not configured to redirect to HTTPS will also result in a warning. Allowing SSH SFTP to use no encryption (configured from the **Advanced** section on the **Security** page of the Server Manager) will also result in a warning.

Resolution: To resolve this issue the system administrator must disable any HTTP listeners in the system (or redirect them to HTTP), configure FTP listeners to require encryption and make sure SSH SFTP listeners are not allowed to use any encryption for connections.

"FXP IS ENABLED AND COULD LEAVE THE SERVER VULNERABLE TO AN FTP BOUNCE ATTACK"

Explanation: FTP bounce attack is an exploit of the FTP protocol whereby an attacker is able to use the PORT command to request access to ports indirectly through the use of the victim machine as a middle man for the request.

Resolution: Go to the **Advanced** page of the Server Manager and check the option to Deny FXP Transfers.

"SERVER IS CONFIGURED TO ALLOW FTP DATA CONNECTION TO RESERVED PORTS"

Explanation: You will receive this warning if you have configured Cerberus to allow FTP data connections to ports less than 1025. Ports 1 through 1024 are intended for system services, so those ports are called reserved ports. FTP should normally not be allowed to establish data connections within that port range.

Resolution: Go to the **Advanced** page of the Server Manager and check the option to Deny Reserved Ports.

"YOU SHOULD SET A REMOTE ACCESS PASSWORD"

Explanation: Web administration and SOAP API remote access use an admin password to control or deny access to the server.

Resolution: Go to the Remote page of the Server Manager and set an admin password.

ABOUT CERBERUS FTP SERVER AUTHENTICATION

Cerberus FTP Server can manage user accounts from three different sources. The first is the default Cerberus FTP Server user database. The Cerberus default user database is displayed in the User List box on the Users page of the User Manager. The accounts within the default database are users created just for Cerberus FTP Server. The directions on this page are for adding a user to this default database.

You may also use Cerberus FTP Server to authenticate Active Directory users when the machine hosting Cerberus is part of a domain (or the local NT account database), even if the computer Cerberus FTP Server is installed on is not the domain controller. See the page Active Directory Authentication for more information on how to configure Cerberus to allow authentication of Active Directory domain users.

Finally, users can also be authenticated against an LDAP service. See the section on configuring Cerberus for LDAP authentication for more information.

NOTE: Active Directory and LDAP authentication are only available in the Professional and Enterprise editions of Cerberus FTP Server.

ADDING A NEW USER

Users can be added and modified in the Cerberus FTP Server user database by opening up the **User Manager** and selecting the **Users** tab.
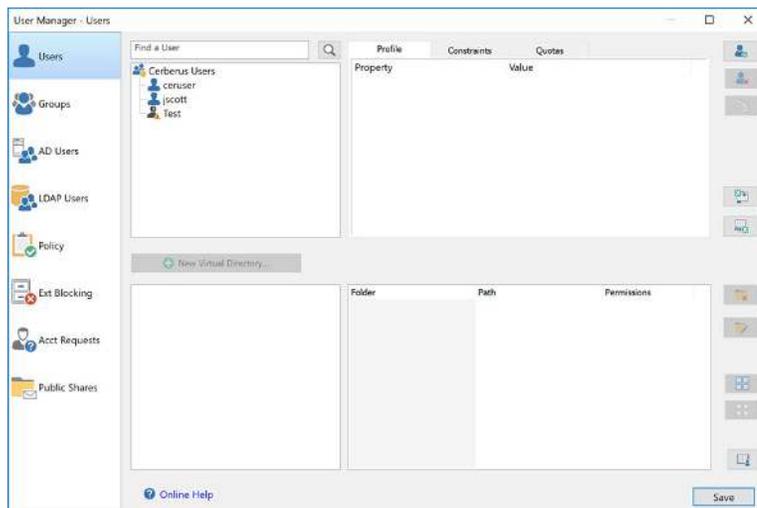
To add a user, select the **New** button from the button group along the right side of the page. A new user will appear under the user list box. The newly created user will already be in rename mode, so simply type in the new name of the user. All usernames must be unique and are case insensitive. Once you

have entered the new username, press enter to commit the change. The user can then be configured by clicking on the user's name in the user list box.

A list of configurable properties for that user will appear in the list box to the right of the user. Those properties are:

Password	The password for the user. **Note**: The Password always displays as 7 (*) characters.
Group	A Cerberus FTP Server Group that this user belongs to.
Is Anonymous	If checked, the user password is ignored and the user can be logged in using any password.
Is Simple Directories	In simple directory mode, the administrator can only assign one directory to represent the virtual directory for a user. See below for an explanation of this setting.
Is Disabled	Determines whether the account can log in or not. A disabled account cannot log in into the server.
Simultaneous Logins	The maximum number of connections this user can make to the server at the same time.
Require Secure Control	(Applies to FTP only) If enabled, this user can only login to the server using a secure TLS/SSL encrypted connection.
Require Secure Data	(Applies to FTP only) If enabled, file transfers will only be allowed over secure TLS/SSL encrypted connections.
Disable After Date	If a date is set here then the account will become disabled after the date specified. **Note**: The granularity of the timer is 30 minutes. The account will be disabled within 30 minutes of the time set.
Allow Protocols to Login	Controls which protocols a user is allowed to login with. If a protocol is not checked then the user will not be allowed to login using that protocol.
SSH Authentication	Determines the authentication requirements for logging into an SFTP interface. Valid options are: ● **Password Only**: Require only a password for authentication. ● **Public Key Only**: Require only a valid public key for authentication

	● **Public Key and Password**: Require both a valid public key and a valid password for authenticating a user
Maximum Upload File Size	This field can be used to limit the maximum size of an uploaded file. This value defaults to unlimited. The file size is specified in bytes. Specify 0 or any non-positive value to reset the maximum file size to unlimited.
Allowed IP Addresses	A comma-separated list of IP addresses that this user can login from. If no IP addresses are specified then no per-user IP address filtering is enforced. IP addresses can be specified as a single IP, a range of IP addresses separated by a dash, e.g. 192.168.0.100 - 192.168.0.150, or a CIDR-formatted IP address range. Multiple formats can be combined, with each single IP or range separated by a comma. Note, global IP address blacklists or whitelists are always enforced first, regardless of this setting.

CONFIGURING A USER FOR SSH PUBLIC KEY AUTHENTICATION

The procedure for configuring a user for SSH Public Key Authentication in Cerberus FTP Server is:

1. Open the Cerberus FTP Server **User Manager**. The default page is the **Users** tab.
2. Select the User from the **Cerberus User Accounts** list that you wish to configure for Public Key Authentication.
3. Double-click on the **SSH Authentication Method** property for the selected user. The **Change SSH Authentication Requirements** dialog will appear.

4. Select the **Public Key Only** or **Public Key and Password** radio option. The **Key Path** edit box and file selection button will become visible/enabled.
5. Select the folder button next to the **Key Path** edit box. A file selection dialog box will appear.
6. Select the public key file you wish to use for the selected user. Press the **Open** button to select the file.
7. Press the **OK** button on the Change **SSH Authentication Requirements** dialog to close and save the new SSH authentication settings.
8. Press the **Close** button on the **User Manager** to save the changes to the selected user.

THE VIRTUAL DIRECTORY SYSTEM

The virtual directory (VD) system allows the administrator to attach any directory or drive to the root. When a client requests the root directory from the server, the VDs you specify are sent to the client. The client can also navigate to any of the VD directories' subdirectories. The VD system takes care of all path translation. Security settings can be specified for each virtual directory. All subdirectories under the VD inherit the security settings of the VD. There are 2 modes that a user account can operate in with respect to the virtual file system. The two modes are simple and standard mode.

SIMPLE VIRTUAL DIRECTORY MODE

When a user account uses simple directory mode, the administrator can only assign one directory to represent the virtual directory for that user. Instead of that directory being seen as a subdirectory off of the root, the virtual directory selected will be the directory the user is placed in when they first log into the server. In other words, the directory selected as the virtual root directory will be the root directory.

STANDARD VIRTUAL DIRECTORY MODE

In standard mode (the **Simple Directories** option is unchecked), the administrator may add as many directories as virtual directories to a user account as desired. The directories selected will appear as subdirectories off of the root when the designated user logs into the server.

A VIRTUAL DIRECTORY MODE EXAMPLE

Let's take a user with one simple virtual directory called **ftproot** that maps to **C:\ftproot**.
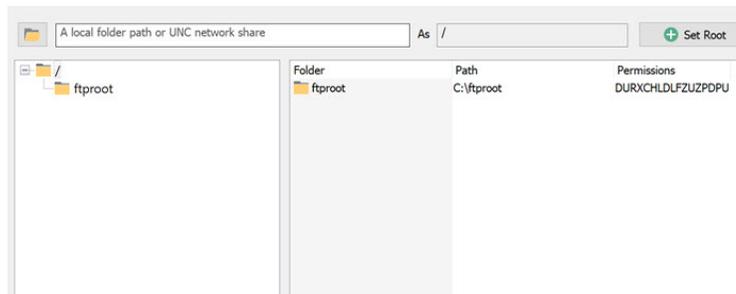
Virtual Directory Settings for a User

In **Simple Directory** mode, the remote root directory that the user sees, "**/**", is mapped directly to **C:\ftproot** on the server. The actual virtual directory name is ignored (you can think of it as always being named "**/**"). The user will see all files and folders in **C:\ftproot** listed in their root directory. They can upload and download files directly into the root directory and they will be uploaded or downloaded to **C:\ftproot** on the server.

When not in simple directory mode, the root directory "**/**" doesn't map to anything. Instead, the root directory "**/**" becomes a virtual file system that you can attach sub-directories to. When not in simple directory mode, you can add as many virtual directories to a user account as you like and the virtual directory name will become a sub-directory in the virtual root. However, you have to change to that sub-directory before you can upload or download anything. If you try to upload a file to the root folder "**/**" then the operation is invalid because the path "**/**" doesn't map directly to a folder on the server. You would need to specify the path **/ftproot** to upload or download files from the virtual directory **ftproot**.

VARIABLES THAT CAN APPEAR IN VIRTUAL DIRECTORY NAMES AND PATHS

The special variable **%USER%** can be present in a virtual directory name or path. When present, the **%USER%** variable is replaced by the user's username during login.

ADDING A VIRTUAL DIRECTORY TO A USER ACCOUNT

Each user can be assigned different virtual directories. A virtual directory is added to a user account by using the User Manager, pictured above. To add a virtual directory to a user, first:

1. Select the user in the "User List."
2. Next, click on the button labeled "**...**". This button is located below the "User List" list box, in the "Virtual Directory" group. Once you have clicked on the "..." button, a "**Browse for Folder**" dialog will appear.
3. Navigate to the directory you wish to add and press the "**OK**" button on the dialog box. The directory you selected should appear in the edit box to the right of the "**...**" button.
4. Finally, select the "**Add to Root**" (this button will be labeled "**Assign as Root**" in simple mode) button located to the right of the "**...**" button.
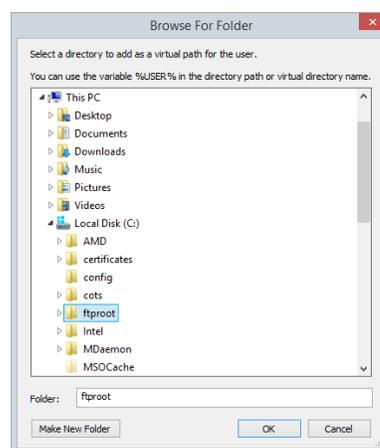
The directory should appear in the "Virtual Root directory" list box. To configure the newly added directory, click on the directory name in the list box. The directory's permission options should appear in the list box to the right of the directory list. Place a check beside any permission that you would like to grant to the virtual directory and all of that directories subdirectories.

VIRTUAL DIRECTORY PERMISSIONS

Each virtual directory that you add for a user can have a separate and distinct set of access permissions. The settings applied to a top-level virtual directory filter down to all of that root directory's subdirectories.

Permissions can only be assigned at the top, root level. To edit the permissions for a virtual directory:

1. Select the user in the Users page of the User Manager

2. Select the virtual directory from the user's virtual directory list

3. Check or uncheck the permissions associated with that directory from the list box that appears when you select the virtual directory.

	A local folder path or UNC network share	As	ftproot		⊕ Set Root

Permission	Status
✔ Download	Allow
✔ Upload	Allow
✔ Rename	Allow
✔ Delete	Allow
✔ Create Directory	Allow
✔ Display Hidden	Allow
✔ List Directories	Allow
✔ List Files	Allow
✔ Zip	Allow
✔ Unzip	Allow
✔ Public Download Share	Allow
✔ Public Upload Share	Allow
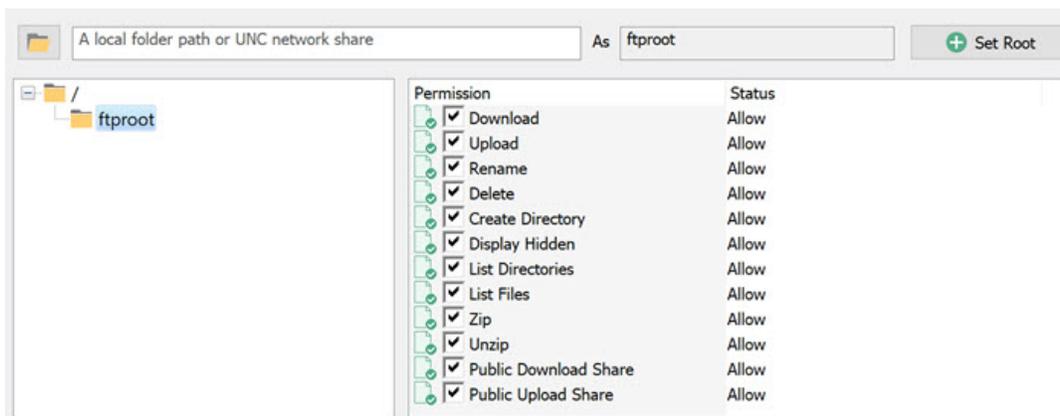
Permissions for virtual directories

ABOUT GROUPS

Using groups simplifies administration of multiple accounts by letting you assign permissions once to a group, instead of multiple times to each individual user. You can add Virtual Directories and basic user settings to a group and have users inherit those permissions. By default, when a user is assigned a group, that user inherits all of the group's settings. However, those settings can still be overridden for the user account.

When a user is a member of a group, the user's settings on the Users page will be grayed out, and the actual value displayed for each grayed setting is the value of the group that the user belongs to.

Virtual directories for the user account are a combination of the group's virtual directories, and any virtual directories you assign specifically to the user account.

Cerberus FTP Server User Manager- Groups page

You can always override the group settings for a user by right-clicking on that user in the User Manager, and then selecting the "Override Group" option to assign a value different from the group value. You can revert back to the group setting by right-clicking on the user and selecting "Default to Group".

A group can be added and modified in the Cerberus FTP Server database by opening up the User Manager and selecting the **Groups** tab. To add a group, select the **New** button to the right of the "Cerberus Group Accounts" group box. A new group will appear under the group list box. The newly created group will already be selected and in rename mode. All group names must be unique and are case insensitive. Once you have entered the new group name, press "enter" to commit the change. The group can then be configured by clicking on the group name in the group list box. A list of configurable properties for that group will appear in the list box to the right of the group.

Those properties are:

Is Anonymous	If checked, the password for any user that is part of this group is ignored and the user can be logged in using any password.
Is Simple Directories	In simple directory mode the administrator can only assign one directory to represent the virtual directory for a user that is a member of this group.
Is Disabled	Determines whether the account can login or not. A disabled account cannot login to the server.
Simultaneous Logins	The maximum number of connections this user can make to the server at the same time.
Can Change Password	Controls, whether a user's that, belong to the group can change their password through the HTTP/S web client or through SSH SFTP or FTP commands.
Require Secure Control	(Applies to FTP only) If enabled, members of this group can only login to the server using a secure TLS/SSL encrypted connection.
Require Secure Data	(Applies to FTP only) If enabled, members of this group can only initiate file transfers over secure TLS/SSL encrypted connections.
Disable After Date	If a date is set here then the group will become disabled after the date specified. All users that are members of this group will also become disabled. **Note**: The granularity of the timer is 30 minutes. The account will be disabled within 30 minutes of the time set.

Allow Protocols to Login	Controls which protocols a member of this group is allowed to login with. If a protocol is not checked then the user will not be allowed to login using that protocol.
SSH Authentication	Determines the SSH authentication requirements for users that are members of this group. Valid options are: ● **Password Only**: Require only a password for authentication. ● **Public Key Only**: Require only a valid public key for authentication ● **Public Key and Password**: Require both a valid public key and a valid password for authenticating a user
Maximum Upload File Size	This field can be used to limit the maximum size of an uploaded file. This value defaults to unlimited. The file size is specified in bytes. Specify 0 or any non-positive value to reset the maximum file size to unlimited.
Allowed IP Addresses	A comma-separated list of IP addresses that members of this group can login from. If no IP addresses are specified then no per-group IP address filtering is enforced. IP addresses can be specified as a single IP, a range of IP addresses separated by a dash, e.g. 192.168.0.100 - 192.168.0.150, or a CIDR-formatted IP address range. Multiple formats can be combined, with each single IP or range separated by a comma. Note, global IP address blacklists or whitelists are always enforced first, regardless of this setting.

USER POLICY SETTINGS

AUTHENTICATION ORDER

Cerberus FTP Server can authenticate against several different types of data sources. The current possible authentication sources include the **Native user system**, **Active Directory (AD)**, and **LDAP**. You can have multiple AD and LDAP servers configured and Cerberus will check each one and attempt to match a username and password. Cerberus will try each authentication source in order until a successful authentication occurs or until all sources fail authentication.

 The order that authentication sources are checked is determined by the Authentication Order list box. You can move authentication sources up and down in order depending upon your needs.

AUTHENTICATION REQUIREMENTS

The Disable Account and Password Storage Format options only apply to Cerberus Native accounts.

Disable Account After *x* Failed Attempts	The Native account becomes disabled if *x* number of consecutive failed login attempts. The counter is reset on a successful login.

Password Storage Format	This is the method Cerberus uses to store user account password information. Options are SHA1, SHA256, and SHA512. All options are salted and are performed using FIPS compliant crypto routines if the server is in FIPS mode.
Disable Account Last Login Exceeded	Native accounts become disabled if they exceed x number of days without a successful login.
Stop Authentication Chain if User Exists	If a user is found in an authentication source but the password is incorrect, don't proceed to check the other authentication sources. Just fail the authentication request.
Auto-Create Variable Directories	The variable **%USER%** can be used in virtual directory names and paths. This variable is evaluated to the account's name when the user logs in. Selecting this option ensures that virtual directory paths with the **%USER%** variable in them will be automatically created when the user account is evaluated during login.
Create Home Directory As User For AD	This setting influences how home directories are created for Active Directory users when the default virtual directory mapping mode in AD is set to **Global Home/%USER%** mode. Normally, Cerberus creates the home directory while under the service account. If this option is enabled, Cerberus will impersonate the AD user before creating the directory. This ensures the home directory is owned by the AD user instead of the service account.
Use UPN for Home Directory for AD	This setting influences how home directories are created for Active Directory users when the default virtual directory mapping mode in AD is set to Global Home/%USER% mode. If this option is checked, Cerberus will always use the AD user's UPN name as the home directory name, instead of the user's login name. AD users can usually use either their SAMAccount or their UPN name. Checking this option will ensure the user is always placed in the same home directory, regardless of whether they login with their SAMAccount or UPN name.

PASSWORD COMPLEXITY REQUIREMENTS

These settings only apply to Cerberus Native accounts.

Minimum Length	The password must be at least x characters long.
Require at Least *x* Letters	The password must contain at least x count of letters.
Require at Least *x* Numbers	The password must contain at least x count of numbers.
Require at Least *x* Special Characters	The password must contain at least x count of special characters (e.g.: %, $, #).

PASSWORD CHANGE POLICY

These settings only apply to Cerberus Native accounts.

Require Password Change Every *x* Days	The server will require that native account passwords be changed after this number of days. Not all protocols have standard support for password changing, and not all clients implement that support when it does exist. To overcome this limitation, you can disable password expiration checking for specific protocols. Note, marking a user account password as requiring a change on the next login requires the password change option to be checked. **Applies to FTP** - When checked, this policy is enforced for FTP/S account access. Note, FTP does not have a standard way of changing or prompting the user to change an account password. Cerberus supports a common extension that allows changing the user password using the SITE PSWD oldpassword newpassword command. However, using that command requires the user to be logged in. The protocol does not have a mechanism for informing the user of an expired password during login. As a result, there is no way to change an expired password via FTP once it has expired. The user will be unable to login via FTP. **Applies to SSH SFTP** - When checked, this policy is enforced for SSH SFTP account access. SSH has a standard method of allowing users to change their passwords, but many SFTP clients do not implement it. **Applies to HTTP** - When checked, this policy is enforced for HTTP/S account access. Cerberus handles the logic of making sure the user is properly prompted for changing an expired password during login, so this method is supported by all web browsers.

PASSWORD HISTORY

These settings only apply to Cerberus Native accounts.

Remember Last *x* Passwords	Cerberus will save a secure hash of the last specified number of passwords that the user has used.
Can't Reuse Last *x* Passwords	Cerberus will prevent a user from changing their password to any password used within the specified history count.

General Settings

Configuring General Settings

The general settings page contains options for connection timeout, network detection, login notifications, and auto-update settings.

General

The general settings page contains options for connection timeout and hiding the main Cerberus window.

Use idle connection timeout	Controls whether idle connections should be terminated after a period of inactivity. The **Idle Connection Timeout (seconds)** value controls how long a connection to the server can remain idle without being terminated.
Use HTTP/S web client idle session timeout (in seconds)	Controls how long (in seconds) a web client session can remain idle before the session becomes invalid and the user has to login again.

Minimize window to tray on startup	If selected, Cerberus FTP Server will start hidden when windows starts up. Only the tray icon will appear. You can restore the graphical interface by double-clicking on the Cerberus tray icon, or right-clicking on the tray icon and selecting "Show/Hide Server"

NETWORK

Controls general network settings.

Detect WAN IP at Startup	If enabled, Cerberus will attempt to detect the external address that Internet computers see for connecting to the network this machine is located on. This is usually the external router address. Enabling this option is important for ensuring passive connections work correctly.
Add to Windows Firewall Exception List	If selected, Cerberus FTP Server will attempt to add itself to the Windows Firewall Exception list. This setting is disabled on operating systems that do not support the Windows Firewall (Windows 2000 and below).
Detect IPv6 Addresses	If selected, Cerberus FTP Server will attempt to detect any IPv6 addresses that the system has initialized. You can leave this setting disabled if you are not using IPv6.
Bind to Localhost Address	If selected, the server will bind to the IPv4 loopback addresses 127.0.0.1, and (if IPv6 is enabled) the ::1 loopback address.

NOTIFICATION

Controls user login notification settings.

Display taskbar notification window on user login	If enabled, Cerberus will display a small notification window on the bottom-right corner of the desktop whenever a user attempts to login to the server.
Check for Updates	Controls how often the server will check for updates. Possible values are: Never, Daily, Weekly, or Monthly.

The Protocols page allows you to control individual settings that affect the security, functionality, and compatibility of the different secure file transfer protocols.

PASV Port Range

These settings control passive FTP options.

Start	First port in the port range to use for passive connections.
End	Last port to use for passive connections before wrapping back around to the Start port.
Randomize Passive Ports	A security option that when enabled causes the server to choose a cryptographically random, unused passive port from the passive port range. When this option is disabled the server selects a passive port from the passive port range incrementally.
Deny FXP Transfers	File eXchange Protocol (FXP) is a method of data transfer which uses the FTP protocol to transfer data from one remote server to another (inter-server) without routing this data through the client's connection. Conventional FTP involves a single server and a single client; all data transmission is done between these two. In the FXP session, a client maintains a standard FTP connection to two servers and can direct either server to connect to the other to initiate a data transfer.
Deny Reserved Ports	Do not allow passive or active port requests below port 1024.

Advanced FTP/S Settings

FTP Directory Listing Time Format

This setting determines the time zone format for the file list returned in response to the LIST and NLST commands. Most clients expect dates and times to be UTC format.

Universal Time (UTC)	The default, send file date/time in UTC format.
Local Time	Send file date/time in local time.
Advertise FTP MLST/MLSD	Allow the FTP server to advertise to clients that it supports the MLST/MLSD command (recommended).
Retrieve Owner/Group information for file listings	Includes the owner and group of each file in responses to the LIST and NLST command. NOTE: This will slow down file listings.

FTP MDTM Time Format

The FTP command, *MODIFICATION TIME (MDTM)*, can be used to determine when a file in the server file system was last modified. This command has existed in many FTP servers for many years, as an adjunct to the REST command for STREAM mode. As a result, this command is widely available.

This command is also frequently used in a non-standard fashion to set file modification times. Cerberus supports both the standard MDTM command for retrieving file times and the non-standard use for setting the date/time on a file.

NOTE: Setting dates and times requires FTP client support. There is often a setting that has to be enabled in many FTP clients before an uploaded or downloaded file will have its date/time set. Consult your FTP client documentation on how to enable this setting. Cerberus automatically supports setting a file date/time without any additional configuration.

Universal Time (UTC)	Most FTP clients expect the *MDTM* command to process date/time values in UTC format and this is the default. Selecting this option will cause Cerberus to interpret and send dates in UTC format.
Local Time	Interpret and send dates in local time (not RFC compliant).
Set Modification Time	When clients attempt to use the non-standard MDTM extension to set a date/time for a file, this setting determines whether the file modification time will be set.
Set Access Time	When clients attempt to use the non-standard MDTM extension to set a date/time for a file, this setting determines whether the file access time will be set.

FTP COMPRESSION

Cerberus FTP Server 5.0 and higher support MODE Z compression for FTP directory listings, uploads, and downloads.

Allow MODE Z Compression	The default, send file date/time in UTC format.
Disable Compression on Local Network	The benefits of compression on the local network can often time be outweighed by the time it takes to compress that data. It is recommended that compression be disabled for local network connections. (recommended)

FTP MISCELLANEOUS

These are FTP settings that don't fit anywhere else.

Allow FTP Renames to Overwrite Existing Files	When this option is enabled an FTP client can issue a rename command and overwrite an existing file.
Allow FTP TLS Upgrade	The FTP server will advertise and allow clients to upgrade plain FTP connections to encrypted FTP connections (FTPES) when this option is enabled (recommended).
Use Optimized File Sending	Uses the built-in Windows API for potentially faster file sending on Windows Server machines. This option only applies to plain FTP transfers. It provides no benefit for encrypted file transfers.

SSH Security Defaults	These are common security settings. Selecting a security profile from the button list list will immediately modify the server's SSH security settings to match that profile.
Ignore SSH Window Size	Some SFTP clients do not correctly request an increase in the SSH channel window size. Enabling this option will allow those connections to continue even after exceeding the available channel window space.
Require Encryption on SFTP	Although most clients won't request an unencrypted connection, the SSH protocol does allow it. Check this option to disallow unencrypted SSH connections. This option should always be enabled for production servers.
Mask Server Identification	If this option is checked, the server will use a generic identification string for the welcome message during SSH connections. The server will also omit the server header for HTTP/S connections.

ADVANCED SSH SETTINGS

Active Key Exchange	The SSH key exchange algorithms that the server will advertise as supported to SSH clients.
Active SSH SFTP ciphers	The cipher algorithms advertised by Cerberus to clients during secure connection negotiation for SSH2 SFTP. You can select the algorithms you want advertised using this list. A full Cerberus FTP Server Windows Service restart is necessary for changes to this list to take effect.
Active MAC	The HMAC algorithms advertised by Cerberus to clients during secure connection negotiation for SSH2 SFTP. You can select the algorithms you want advertised using this list. A full Cerberus FTP Server Windows Service restart is necessary for changes to this list to take effect.

HTTP/S SETTINGS

These are advanced settings for controlling HTTP/S web client defaults for all users.

| Temp Upload Directory | HTTP/S web client uploads are stored in this directory as they are uploaded. When the upload completes, the file is moved to its final destination.

If this edit box is left blank, the temporary upload directory defaults to the temporary files directory for the account running the Cerberus FTP Server Windows Service for native |
|---|---|

	accounts and LDAP accounts. For AD accounts, it defaults to the temporary folder for that AD user on the server machine. This field can be used to override the defaults for all account types.
Optional Headers to Include	Allows the administrator to determine if the listed HTTP headers should be sent to clients for HTTP/S web client connections.

CONFIGURING LOGGING SUPPORT

AUDITING

Cerberus FTP Server provides comprehensive logging of all file and user operations and provides both on-screen logging, file logging, and Syslog support. File-based logging can be managed through an XML configuration file that can control nearly all aspects of how log data is written to a file.

LOG FILE LOCATION

Cerberus FTP Server logging is implemented through the **Apache Log4cxx** framework, a robust logging package modeled after the popular log4j Java logging package. The default configuration logs up to 5000 KB of data to a single file and then rolls over to a new log file. The past 10 log files are kept by default but log file size, naming, and history are all completely configurable through the log4j.xml file.

The log file is located at the following location:

ON WINDOWS VISTA, WINDOWS 2008 AND ABOVE

C:\ProgramData\Cerberus LLC\Cerberus FTP Server\log

ON WINDOWS 2003, XP, AND 2000

C:\Documents and Settings\All Users\Application Data\Cerberus LLC\Cerberus FTP Server\log

 You can also open the log file by simply clicking on the **Open Log File** link on the **Log** tab of the main user interface console as demonstrated below:
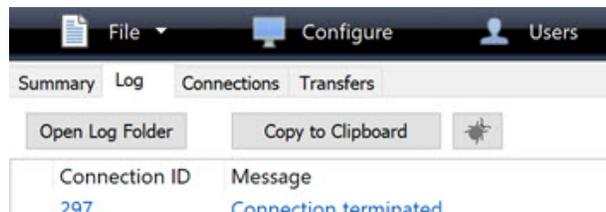
CONFIGURING LOGGING

The **log4j.xml** configuration file is one level above in the "Cerberus FTP Server" folder. An example log4j.xml file is below.

There is an example of a size-based log appenders which roll over after the log file reaches a certain maximum size and that limit the number of log files that are kept. These types of loggers are limited to at most 13 saved log files.

There is also daily log file appender example (with no maximum number of kept log file limits), and a Syslog log appender example.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<log4j:configuration xmlns:log4j='http://logging.apache.org/' debug="false">

        <appender name="FILE" class="org.apache.log4j.rolling.RollingFileAppender">
                <rollingPolicy
class="org.apache.log4j.rolling.FixedWindowRollingPolicy" >
                        <param name="activeFileName" value="log/server.log" />
                        <param name="fileNamePattern" value="log/server.%i.log" />
                        <param name="minIndex" value="1" />
                        <param name="maxIndex" value="5" />
                </rollingPolicy>
                <triggeringPolicy
class="org.apache.log4j.rolling.SizeBasedTriggeringPolicy">
                        <param name="maxFileSize" value="5000KB" />
                </triggeringPolicy>
                <layout class="org.apache.log4j.PatternLayout">
                        <param name="ConversionPattern"
                                value="[%d{yyyy-MM-dd HH:mm:ss}]:%7.7p [%6.6x] -
%m%n" />
                </layout>
        </appender>

        <appender name="ERROR_FILE"
class="org.apache.log4j.rolling.RollingFileAppender">
                <rollingPolicy
class="org.apache.log4j.rolling.FixedWindowRollingPolicy">
                        <param name="activeFileName" value="log/server_error.log"/>
                        <param name="fileNamePattern"
value="log/server_error.%i.log"/>
                </rollingPolicy>
                <triggeringPolicy
class="org.apache.log4j.rolling.SizeBasedTriggeringPolicy">
                        <param name="maxFileSize" value="5000KB"/>
                </triggeringPolicy>
                <layout class="org.apache.log4j.PatternLayout">
                        <param name="ConversionPattern" value="[%d{yyyy-MM-dd
HH:mm:ss}]:%7.7p [%6.6x]
                                - %m%n"/>
                </layout>
                <filter class="org.apache.log4j.varia.LevelRangeFilter">
                        <param name="LevelMin" value="ERROR" />
                </filter>
        </appender>


        <root>
                <level value="INFO" class="org.apache.log4j.xml.XLevel" />
```

```
            <appender-ref ref="FILE"/>
            <appender-ref ref="ERROR_FILE"/>
        </root>
</log4j:configuration>
```

Possible values for the **<level value="LEVEL" class="org.apache.log4j.xml.XLevel" />** tag's *level* parameter are:

- TRACE
- DEBUG
- INFO
- WARN
- ERROR

Example Syslog log4j.xml Configuration File

The below log file example shows a Syslog logger.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <log4j:configuration xmlns:log4j='http://logging.apache.org/' debug="false">


      <!-- Add a Syslog appender -->
      <appender name="syslog" class="org.apache.log4j.net.SyslogAppender">
            <param name="SyslogHost" value="127.0.0.1"/>
            <param name="Facility" value="USER"/>
            <param name="FacilityPrinting" value="true"/>
            <layout class="org.apache.log4j.PatternLayout">
                  <param name="ConversionPattern"
                        value="%t %5r %-5p %-21d{yyyyMMdd HH:mm:ss,SSS} %c{2} [%x] %m
%n"/>
            </layout>
      </appender>


      <root>
            <level value="INFO" class="org.apache.log4j.xml.XLevel" />
            <appender-ref ref="syslog"/>
      </root>
 </log4j:configuration>
```

Example Daily Rollover log4j.xml Configuration File

The below log file example shows a simple daily rollover logger.

```
 <?xml version="1.0" encoding="UTF-8" ?>
 <log4j:configuration xmlns:log4j='http://logging.apache.org/' debug="false">
```

```
        <!-- Add a Daily log file appender that will roll over to a new log file each
night -->
        <appender name="DAILY_ROLL"
class="org.apache.log4j.rolling.RollingFileAppender">
            <rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
                    <param name="FileNamePattern"
value="log/daily_server.%d{yyyy-MM-dd}.log"/>
            </rollingPolicy>
            <layout class="org.apache.log4j.PatternLayout">
                    <param name="ConversionPattern" value="[%d{yyyy-MM-dd
HH:mm:ss}]:%7.7p [%6.6x] - %m%n" />
            </layout>
        </appender>

        <root>
            <level value="INFO" class="org.apache.log4j.xml.XLevel" />
            <appender-ref ref="DAILY_ROLL"/>
        </root>
 </log4j:configuration>
```

Log File Location and Naming

You can change the location and name of the file created under the various log appenders using the appropriate field. For the default RollingFileAppender logger with a FixedWindowRollingPolicy you will need to change both the activeFileName and fileNamePattern parameters in the **log4j.xml** file. For the DailyRollingFileAppender you will just need to change the File parameter associated with the logger.

If using a relative log file path, the path is relative to the **C:\ProgramData\Cerberus LLC\Cerberus FTP Server** folder.

Screen Logging Settings

In addition to the file-based log, Cerberus also displays the current log output to the graphical user interface while the server is running. Options for the screen-based logging can be controlled through the Logging settings tab of the Server Manager.

Root Log Level

Controls the root log level for all log appenders. All log appenders inherit the root log level as their lowest threshold. The default level is INFO.

Log appenders can be set at a higher log level threshold than the root logger, but they cannot be set at a lower level. For example, if the Syslog appender is set to DEBUG, but the root log level is set to INFO, the Syslog will still only write out log information at the INFO level.

The DEBUG level is for troubleshooting, and the root log level should not be left at this level for regular production use because of the excessive logging produced.

SYSLOG SUPPORT

Cerberus FTP Server 5.0 and higher supports Syslog integration. Administrators can control Syslog settings from this page.

Enable Syslog logging	Enable syslog logging
Syslog Host	The address of the machine hosting the syslog server.
Syslog Facility	The syslog facility value that should be associated with the syslog events.

CONFIGURING INTERFACE SETTINGS

An interface or listener is simply an IP address, port, and protocol combination that the server is accepting connections on. For example, you can add an FTP listener on port 21 and attach it to an IP address. It can be an IPv4 or IPv6 address. The "Default" interfaces represent the settings that will be applied for newly detected interfaces. There are several different parameters that each interface can have:

There are seven types of listeners that you can add to an IP address:

FTP	Traditional FTP, default port 21
FTPS	Implicit FTP with TLS/SSL encryption, default port 990
SSH SFTP	SSH2 File Transfer Protocol, default port 22
HTTP	HTTP, default port 80
HTTPS	HTTP with TLS/SSL encryption, default port 443
HTTP Admin	HTTP for web administration, default port 8080
HTTPS Admin	HTTP with TLS/SSL encryption for web administration, default port 8443

The first two allow regular FTP as well as different forms of secure FTP while the SSH2 SFTP listener is for establishing connections over the SFTP protocol (a completely different protocol from FTP, despite the similar name). The HTTP and HTTPS listeners allow web client connections to the server using either the unsecure HTTP protocol or encrypted HTTPS protocol.

There are two types of secure FTP connections possible, FTPS and FTPES. FTPS is usually referred to as implicit FTP with TLS/SSL security. Its closest analog is HTTPS. It is basically the FTP protocol over a TLS/SSL secured connection. This form of secure FTP is deprecated but widely supported and still in use. This is what a Cerberus FTP Server **FTPS listener** is for and this type of listener typically listens on port 990. Note, the settings "Require Secure Control" and "Require Secure Data" are meaningless for this type of listener. Connections established to an FTPS listener can only be established securely.

FTPES, which is often referred to as **explicit FTP** with TLS/SSL security, is a modification of the FTP protocol that starts out over an insecure, normal FTP connection and is then upgraded to a secure connection through FTP command extensions during login. This is the preferred method of secure FTP because it allows SPI firewalls to know that there is FTP traffic occurring on the connection. You establish FTPES sessions using a normal Cerberus FTP Server **FTP listener**, typically over port 21. Both unencrypted FTP and explicit TLS/SSL connections can be established to this type of listener. You cannot establish an implicit FTPS connection over this type of listener.

ADDING A NEW INTERFACE LISTENER

Cerberus FTP Server supports adding multiple listening interfaces for a given IP address. This allows you to have Cerberus accepting connections from different protocols on multiple ports. The only requirement is that each listener be on a unique IP/port combination. You can add FTP, FTPS (for implicit secure FTP only), SSH2 SFTP, HTTP or HTTPS listeners.

Select the "plus" icon next to the interface list box to add a new interface. A new dialog box will appear to ask for the interface details (interface IP, type, and port combination). Selecting the "X" icon will prompt you to delete the selected interface listener.

INTERFACE SETTINGS

Listen Port	This setting is the port that this interface will listen on for the control connection.
Max Connections	The setting determines the maximum number of simultaneous connections that can connect to this interface listener.
Require Secure Control	(Applies to FTP only) If enabled, only secure control connection will be allowed. This is required to protect passwords from compromise on unsecured networks with FTP.
Require Secure Data	(Applies to FTP only) If enabled, only secure data connections will be allowed. All directory listings and file transfers will be required to be encrypted.
Don't Use External IP for Passive connections	If this option is checked, Cerberus will always use the internal IP address when the incoming connection originates on the local network.
Passive IP Options	**Auto Detect** - If WAN IP auto detection is enabled then use the WAN IP for the PASV command, otherwise use the interface's IP. **Specify PASV IP** - Allows the administrator to specify what IP address is returned in response to a PASV command **Use DNS service** - Allows use of DNS names like www.cerberusftp.com. The address specified will be examined at regular intervals and the IP address that represents that DNS name will be used in PASV commands.

Show Welcome Message	If checked, the server will send a welcome message during user login for FTP/S, SSH SFTP, and the HTTP/S web client (note, some FTP and SFTP clients won't display the welcome message).
Allow User Updates	(Applies to HTTP/S only) If checked, the user will be allowed to update his or her personal account information (first name, last name, email, or telephone number) through the HTTP/S web client.
Allow Web Account Requests	(Applies to HTTP/S only) If checked, users can request new accounts through the HTTP/s web client.
Company Name	(Applies to HTTP/S only) The company name to display in the web client page title
Logo Image	(Applies to HTTP/S only) The logo image to display in the web client header. This file's dimensions should be 230 by 70.
Login Image	(Applies to HTTP/S only) The image to display on the web client login page. This file's dimensions should be 70 by 70.
Default Web Directory List Count	(Applies to HTTP/S only) The default number of entries that appear in the web client file list.
Show Time zone on Dates	(Applies to HTTP/S only) Toggles displaying time zone information for files and directories in the web client
Display Local Time	(Applies to HTTP/S only) Toggles between displaying server local time or UTC time for files and directories in the web client
Configure CAPTCHA	(Applies to HTTP/S only) Configures Google reCaptcha for the web client login and web requests pages.
Redirect requests to HTTP/S listener	(Applies to HTTP only) Any requests that come in over this HTTP listener will be redirected to the same address using HTTPS.

The "Default" Interfaces

There is a Default interface for each type of listener (FTP, implicit FTPS, SFTP, HTTP, and HTTPS). When a new interface (IP address) is detected, that interface will receive an FTP, FTPS and SFTP listener and each of those listeners will be assigned the values of the appropriate "Default" interface at

the time of detection. For example, If the "Default FTP" interface was defined to be on port 21, then when a new interface is detected for the first time it will receive an FTP listener on port 21 with the values of the Default FTP interface. Those settings then become the settings for the newly detected interface. Note that the new interface's settings are not linked to the "Default" interface in any way. The "Default" interface simply represents the values that newly detected interfaces will be initialized with. Changing the values of the "Default" interface wouldn't change any values on existing or previously detected interfaces.

For example, when you first install Cerberus FTP Server, the "Default FTP" interface is set to port 21 (the default FTP listening port) and all interfaces detected during that first start will receive FTP listeners with that port value. If you later change the "Default FTP" interface settings then that change will have no effect on existing interfaces.

It is also worth noting that Cerberus remembers the settings for interfaces that were previously detected but might have changed. For servers that have dynamic addresses that constantly change or cycle between a range of addresses, Cerberus will "remember" the old values and apply those instead of the "Default" settings if that interface address is later detected again.

Un-checking the box next to each Default interface will disable automatic listener activation for that interface type when a new interface is detected.

INTERFACE STATUS CONTROLS

Interfaces can also be enabled or disabled from the main Cerberus FTP Server user interface:
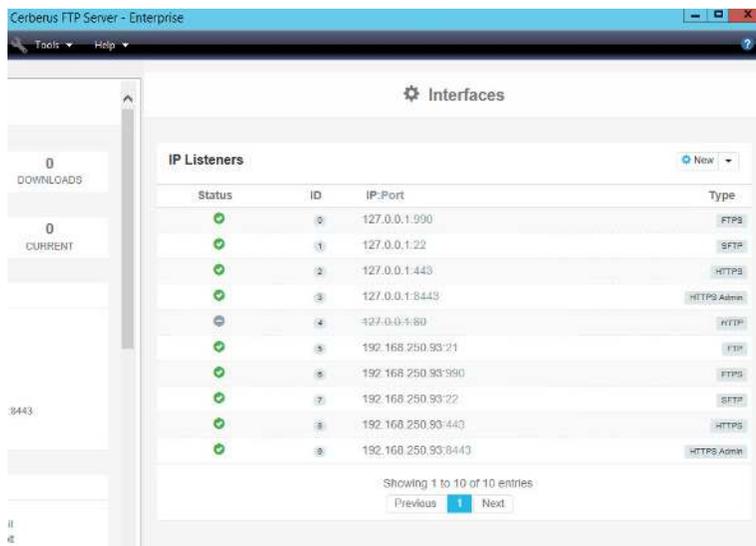
Enabling or disabling a listener

Select an interface and right-click. Click the Enable/Disable menu item to toggle enabling or disabling an interface. Disabled listeners will no longer accept connections.

The HTTP/S Web Client

Available in Cerberus FTP Server Enterprise edition, the **HTTP/S web client** capability allows any user with access to a common web browser to easily connect to the server to perform file operations (uploading, downloading, deleting, renaming, creating directories, and zipping and unzipping files and directories) using a desktop or mobile web browser.

You can also grant users the ability to generate a public link to any file, and email that link to someone from directly within the web client.

The web client is a native web application that requires no plug-ins or external tools to use. The web client relies on HTML and JavaScript for all of its functionality, and will run on any modern web browser.

Security

Administrators can also configure the server to allow or require users to authenticate using two factor authentication when connecting using the web client. We currently support HOTP-compliant one-time password applications, as well as direct integration with DUO Security.

Additional Web Client features include:
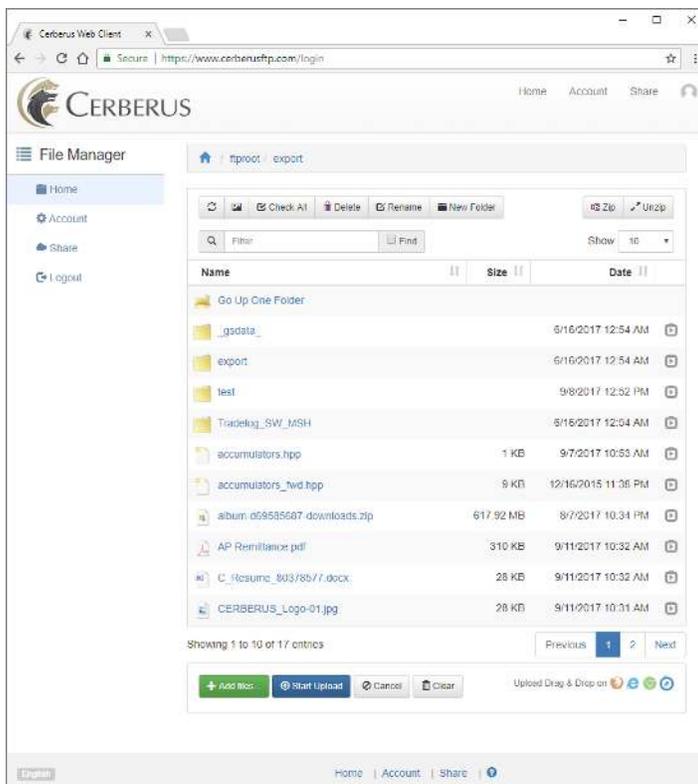
- Fully tested against IE 9.0+, the latest version of Firefox, Chrome, and Safari
- No file size limitations and can efficiently handle file uploads and downloads of any size
- Simultaneous file upload, and upload drag and drop support in Firefox, Safari, Chrome, and IE 10+
- Upload Resume support
- Confirmation dialogs for file deleting, zipping, and unzipping
- Right-click Cut/Copy/Paste support for moving and copying files and folders between other folders
- Web-based with no software or plugins to install for end users
- Allow users to request new accounts through the web
- Allow users to reset forgotten passwords using email link requests combined with security questions
- Configurable anti-bot security using reCAPTCHA for logins and account requests.
- Generate and email a public file link (with optional password protection) from within the web client
- Custom Theme support using the popular Bootstrap 3 framework
- Desktop and mobile Browser Optimized

PUBLIC FILE SHARING

Public file sharing allows a user to can take any file and generate a unique, time-limited, public link to that file and share it with anyone. The public link can also be password protected. The administrator has full control over public file sharing. The administrator has to grant explicit sharing privileges to a user's virtual folder before the user can share files publicly from that folder.
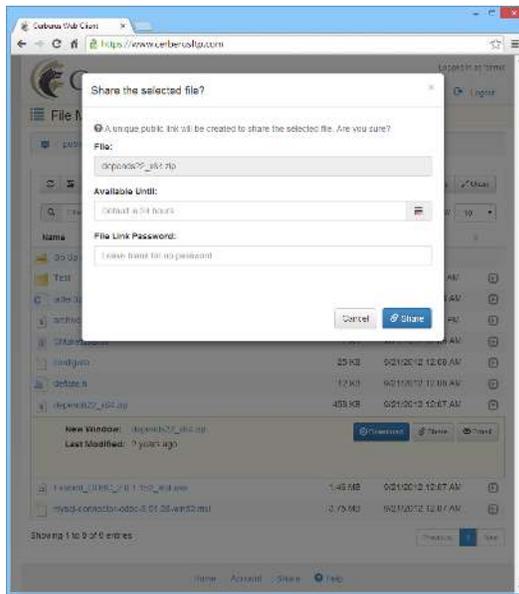

Cerberus FTP Server HTTP/S Web Client

A user can revoke access to the public link at any time through the Shares page of the web client.

EMAILING A LINK TO A PUBLIC FILE

In addition to public file sharing, users can also email a link to a public file directly from within the web client. Users just need to click the "Email" button on a file (and have the appropriate folder permissions granted by the administrator for sharing) to open a message dialog for emailing the file.
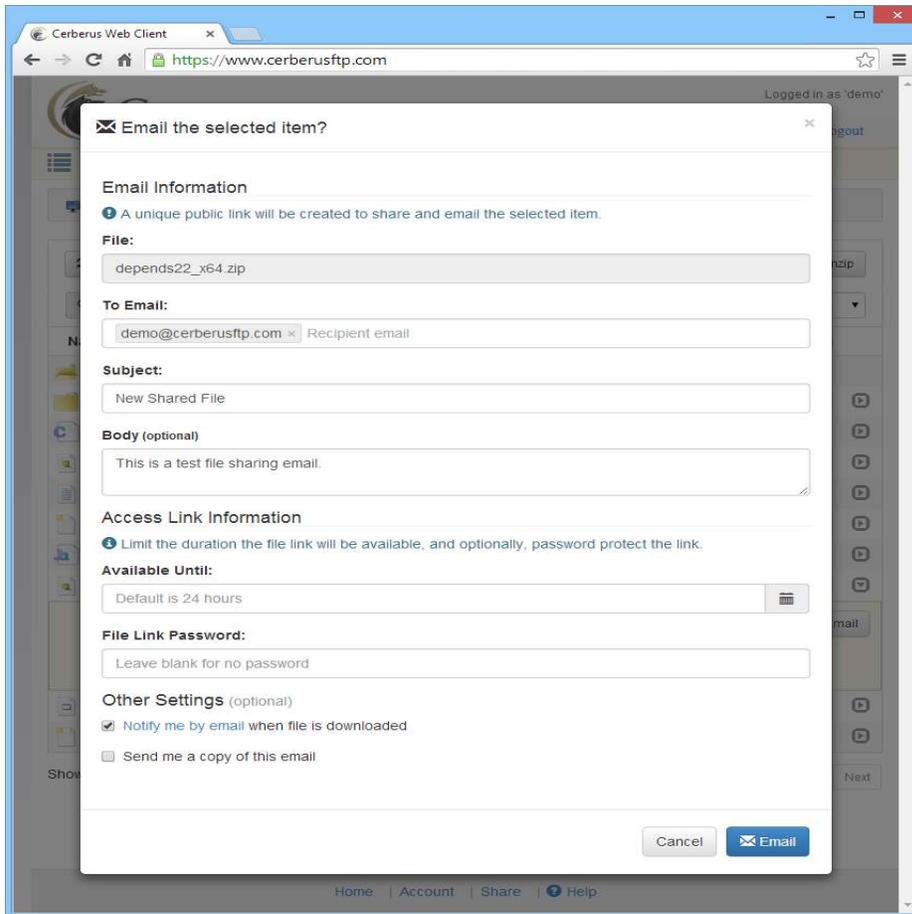
Cerberus FTP Server HTTP/S Web Client

Cerberus FTP Server will use the first SMTP server configured in the Event Manager's Targets page as the email server for sending emails.

ADDING AN HTTP/S LISTENER

The Cerberus FTP Server web client can be accessed by adding an HTTP or HTTPS listener to Cerberus FTP Server's listener list. You can add a new HTTP/S listener from the **Interfaces** page of the **Server Manager**.

To add a new HTTP or HTTPS listener:

1. Open the Server Manager
2. Select the **Interfaces** page
3. Select the "plus" icon next to the interface list box to add a new interface. The "Add New Listener" dialog box will appear to ask for the interface details (interface IP, type, and port combination)
4. Select the IP address that you want to listen for connections on
5. Select the interface type (HTTP or HTTPS for web client access)
6. Enter the port you wish to listen on. Cerberus will automatically pre-populate the port with the default port for the type of listener you are adding
7. Press the **Add** button to add the listener

The listener should now be added to the Interfaces list. Press **Ok** to close the Server Manager and save you changes.

WEB CLIENT CUSTOMIZATIONS

The HTTP/S web client can be customized in several ways. Options for changing the default settings are discussed in the following sections.

CHANGING THE COMPANY LOGO AND LOGIN IMAGE

You can change the company logo displayed on the web client by specifying your own logo file.

1. Go to the **Interfaces** page of the **Server Manager** (pictured above)
2. Select the HTTP/S interface you wish to change (not the default interface)
3. Press the file selection button across from the Logo Image edit box
4. Select the image file you wish to use and press Ok. The preferred image size is 230 x 70.

The login image displayed on the login page is also customizable using the same procedure as for the company logo. The preferred login image size is 70 x 70 pixels.

The image format for both logos should be one that is supported by all web browsers. We recommend PNG, GIF, or JPEG.

Changing the Login Welcome Message

If you select the Show Welcome Message option for the HTTP/S listener then the server welcome message is displayed next to the login credentials box when a client logs in on that listener. This message can be customized from the Messages page of the Server Manager.

Custom Web Client Themes

The HTTPS web client comes installed with several themes, but administrators can easily adapt and add their own. The web client was redesigned in version 7.0 to use the popular Bootstrap 3 framework. You can develop your own custom CSS theme file and drop it in:

C:\Program Files\Cerberus LLC\Cerberus FTP Server\webadmin\client\custom

Then, restart the Cerberus FTP Server Windows Service to have it automatically detect and make available the new theme.

A theme file is simply a css file that contains your own custom overrides of the default Bootstrap 3 theme. Any files your CSS file references should be relative to the custom folder. Cerberus will detect the new CSS file during startup and make it available as a theme (the theme name is based on the file name) on the Accounts page of the web client.

FURTHER WEB CLIENT CUSTOMIZATIONS

The HTTP/S web client can be further customized by modifying the underlying template files. However, any changes made to those template files will be overwritten whenever Cerberus FTP Server is upgraded. We are working on ways to allow more permanent and lasting changes to the web client. The relevant template files are in:

C:\Program Files\Cerberus LLC\Cerberus FTP Server\webadmin

and

C:\Program Files\Cerberus LLC\Cerberus FTP Server\webadmin\client

The client-index.tpl file is probably the best place to start for modifying the overall look of the web client. The template files are cached in memory in Cerberus after the first time they are read, so a restart of the underlying Cerberus FTP Server Windows Service is required before any changes to these files will take effect.

WEB ACCOUNT REQUESTS

ALLOWING USERS TO REQUEST ACCOUNTS THROUGH THE WEB

Starting with Cerberus FTP Server 5.0 Enterprise edition, users can now request new accounts through the HTTP/S Web Client. A "Request a New Account" link will appear on the login page if the administrator decides to allow web account requests.
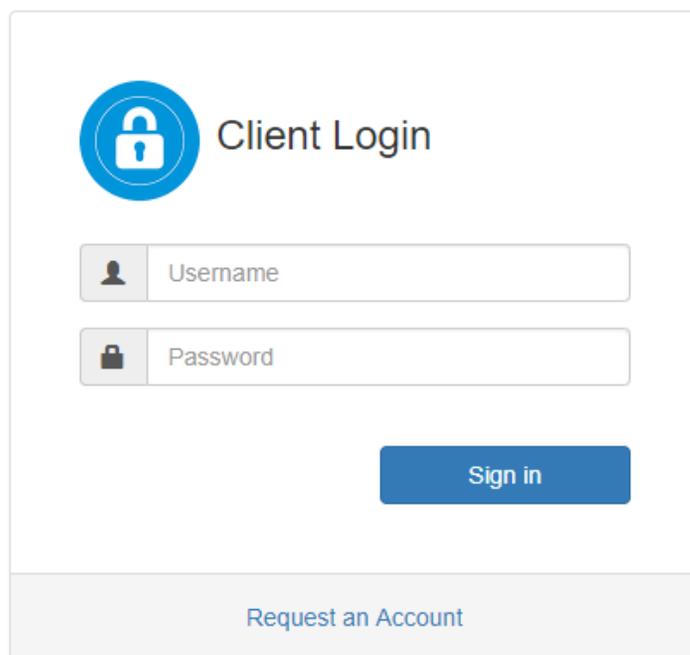
HTTP/S Login Page with "Request a New Account" Link

REQUESTING A NEW ACCOUNT

The account request page allows a user to submit a request for a new account to the Cerberus FTP Server system administrator. The user can set a password for the account (subject to password policy rules) at the time of the account request. This relieves the administrator from having to set a new password for the user and from having to securely distribute that password.

Event Rules can be enabled on the server to automatically email the administrator whenever a new account request is made.

Enabling or Disabling Account Requests

The link can be enabled or disabled for any HTTP or HTTPS listener by selecting that listener in the Interfaces page of the Server Manager.

Administrators can view pending account requests through both the **Account Requests** page of the **User Manager** in the Cerberus GUI, or through the Account Requests administrator web page. Accounts can be approved or denied through either method by selecting an account and using the **Approve** or **Delete** button.

Approved accounts are automatically created and activated in the **Users** page of the **User Manager** and can be further customized there.

The Cerberus FTP Server Account Request Page of the User Manager

CONFIGURING SECURITY SETTINGS

The security settings page allows the administrator to configure all aspects of Cerberus FTP Server SSL/TLS and SSH security. To enable TLS/SSL connections between FTP and HTTP clients and the server, you need a server certificate and a private key.

DIGITAL CERTIFICATE SUPPORT

Cerberus FTP Server supports RSA, DSA and Elliptical Curve (EC) keys. Support for elliptical curve ciphers with FTPS and HTTPS is available with Cerberus FTP Server 6.0 and higher.

There are generally two options for obtaining a digital certificate (with private key):

1. You can generate your own self-signed certificate using the Cerberus **Create Cert** button.

2. You can obtain a certificate from a recognized certificate authority (CA)

Which option is more appropriate really depends upon your goals. If you just want to make sure that client and server connections are securely encrypted, then a self-signed certificate is all you need. Self-signed certificates have the benefit of being easily created through Cerberus and are completely free.

If your goal is to make sure that your clients can verify that the server they are connecting to is legitimate, and to ensure that clients do not see any warning messages about being "unable to verify the server", then you will need to use a certificate signed by a trusted CA. You will have to contact one of the recognized CAs (such as Comodo, Thawte, Verisign, GoDaddy) and request a server certificate. Take a look at our help page on generating Certificate Signing Requests for more information about using a CA-signed certificate.

A note about secure connections: Cerberus supports FTPS, FTPES, SFTP, and HTTPS encryption. To establish a secure connection you must connect to the server with a client that supports one of those secure methods. For secure FTPES, FTPS, or SFTP, this will require a dedicated FTP client, not a web browser. *No* web browsers natively support any type of secure FTP.

We have documentation available that walks you step-by-step through the process of using a self-signed certificate or importing a certificate from a third party certificate authority.

ABOUT CERTIFICATE AUTHORITIES

You only need to worry about setting up and validating against a certificate authority if you (the server) want to authenticate the certificates coming from your FTPS and HTTPS clients. If you are not concerned with verifying your FTPS and HTTPS clients using certificates, then you can safely ignore all of the certificate authority configuration information. Just select the **No verification** setting (the

default). **Note:** Client certificate verification is completely separate from SSH SFTP public key authentication. SSH SFTP public key authentication is configured on a per user basis.
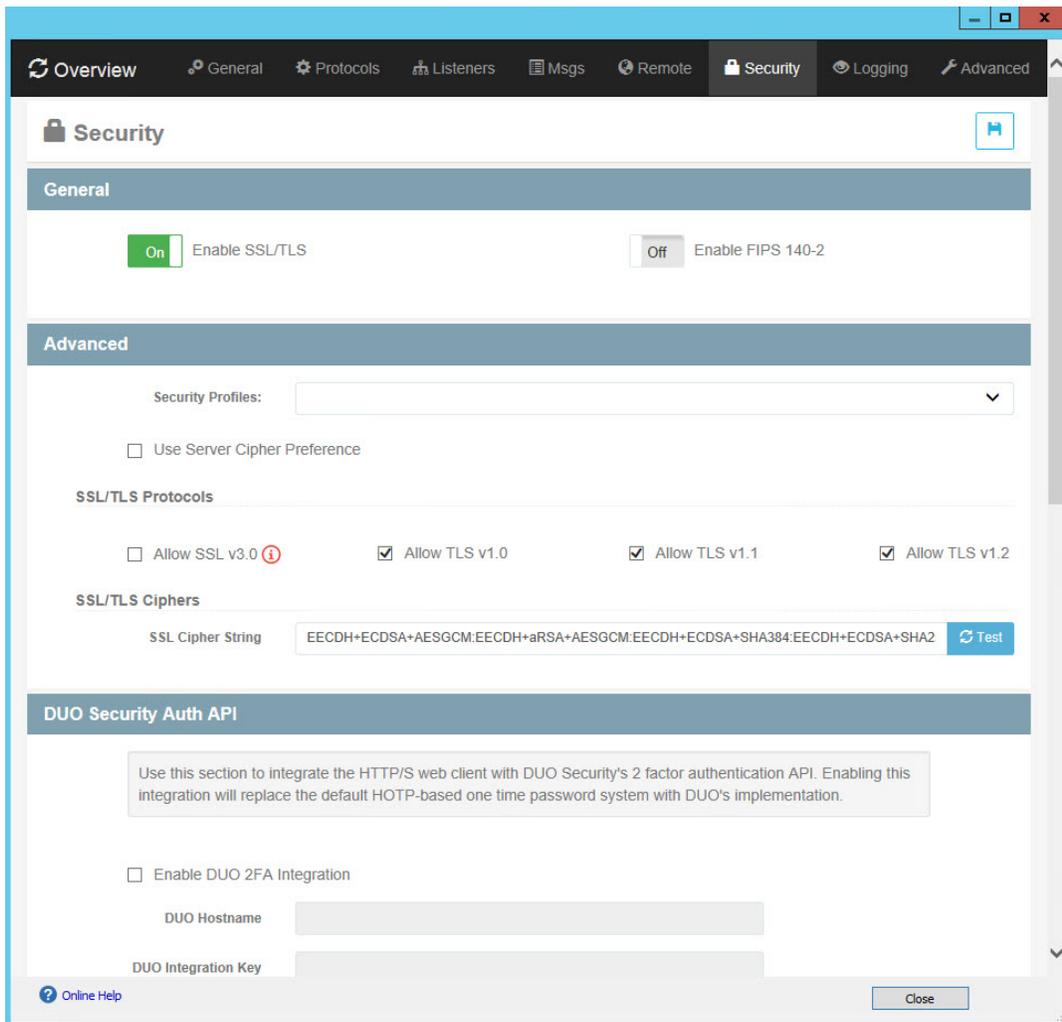
Security settings page of the Server Manager

TLS/SSL Security

Cerberus uses the settings here for all secure connections.

Security Options

These are basic TLS/SSL settings applicable to secure client FTPS, HTTPS, and SSH connections and encrypted HTTPS SOAP messages.

Enable Explicit TLS/SSL	This must be enabled to allow secure access to the server. NOTE: A certificate and private key must be available before TLS/SSL encryption will be available.
Enable FIPS 140-2 Mode	Enable the FIPS 140-2 certified encryption module for Cerberus FTP Server. Selecting this option enables encryption using only FIPS 140-2 certified algorithms. *Only available in the Professional and Enterprise edition*.
Public Certificate	The full path to your public certificate. The public certificate is exchanged with the client during TLS/SSL encryption and is examined by the client to verify the server. Supported key types include RSA, DSA, and Elliptical Curve keys.
Private Key	This is the server's private key. The private key is used to encrypt messages to the client. The client can use the server's public key to decrypt messages encrypted with the server's private key. The private key is not sent to the client. If your public and private key are in the same file then set this path to be the same as the Public Certificate. *NOTE*: The public and private key can be in the same file. If your public and private key are in the same file then set this path to the same path as your Public Certificate path. Cerberus understands both DER and PEM encoded certificate formats.
Needs Key Password	Check this option if the digital certificate is encrypted.
Password	The key password used to decrypt your digital certificate.
CA File	A file containing a PEM-encoded list of Certificate Authorities with which to verify client certificates against. Cerberus FTP Server will also use this file to load and send the entire certificate chain for the server certificate when a client connects. Many CAs call this a CA bundle file.

Create Cert	Cerberus will generate a Self-Signed Certificate that will allow encrypted connections.
Create CSR	Generates a server private key file, and a CSR file. The server private key is kept on the server, and the CSR is submitted to a Certificate Authority. See this help section on generating a CSR for more information.
Verify	Cerberus will attempt to verify that the certificate at the Public and Private key path is recognized and readable with the given password.

CLIENT CERTIFICATE VERIFICATION

Cerberus FTP Server can be configured to require FTPS and HTTPS clients to verify themselves using digital certificates. When given a CA file, Cerberus will verify that the client certificate is signed and valid for the given certificate authorities. Cerberus will also make sure the certificate hasn't been revoked if a CRL is specified. **This feature is only available in Cerberus FTP Server Professional and Enterprise edition and currently only applies to FTPS, FTPES, and HTTPS connections.**

No Verification	This is the default option. Cerberus will not require nor will it verify digital certificates
Verify Certificate	Cerberus will attempt to verify that the certificate presented by the client is signed and valid. It will compare the certificate against the certificate authorities present in the CA Certificates File. Any FTPS or HTTPS connection attempts without a valid certificate will be denied when this option is selected.
CRL File	A file containing a PEM or DER-encoded list of key serial numbers that have been revoked. Note, the CRL must have been signed by the CA certificate.

Security Profiles	These are common security settings. Selecting a security profile from the dropdown list will immediately modify the server's security settings to match that profile.
Server Cipher Preference	During SSL/TLS session negotiation, the connecting client sends an ordered list of cipher suites to the server. The first suite in the list is the one most preferred by the client. Normally, the server honors the client preference by selecting the suite most preferred by the client among the list of suites that both the client and server support. If this option is selected, the server selects the suite that the server itself most prefers among those that both the client and server support. This can be used to, for example, enforce that the strongest cipher that both the server and client support be used for the connection.
Verify Client Certificate Common Name	Cerberus can be configured to provide additional post-verification client certificate checking. Specifically, you can require the certificate common name to match the FTP username. This option can be accessed by pressing the **Advanced** button. Check the option to enable certificate common name to FTP username checking.
SSL Cipher String	The ciphers that Cerberus uses during secure connection negotiation for TLS/SSL can be controlled through a text string. This option can be accessed by pressing the **Advanced** button. An example string: **ALL:!LOW:!EXP:!ADH:@STRENGTH** The string follows the same cipher string format as the OpenSSL ciphers string.
Allow SSLv3, TLS1.0, TLS1.1, TLS1.2	These 4 settings allow you to enable or disable support for SSLv3.0, TLSv1.0, TLSv.1.1, and TLSv1.2 respectively.

DSA Certificates and Ephemeral Diffie-Hellman Keys

Cerberus FTP Server 4.0.3 and higher includes support for **DSA certificates**. Unlike RSA certificates, DSA certificates cannot be used for key exchange (a necessary part of establishing an SSL or SSH connection), and additional pieces of information, known as Diffie-Hellman (DH) parameters, are required to allow key exchange using DSA.

DH parameters are computationally very expensive to generate, and it isn't feasible (or necessary) to generate those parameters in real-time. Cerberus FTP Server includes DH parameters for 512, 1024, 2048, and 4096 bit keys. The parameters were pre-generated using strong sources of pseudo-random entropy, and are used during DH key exchange to generate new, temporary keys for each SSL session.

Cerberus looks for the DH parameter files in the **C:\ProgramData\Cerberus LLC\Cerberus FTP Server\certificates** directory. You can freely replace the included parameter files with your own, pre-generated versions if you desire. If the existing files are deleted, Cerberus will attempt to re-create the missing files during startup by generating new ones. This can take a *very* long time, and Cerberus will appear to hang during startup while the files are generated. Deleting the existing DH parameter files is **not recommended**.

Elliptic Curve SSH Support

Cerberus FTP Server 4.0.9 and higher support Elliptic Curve Diffie-Hellman (ECDH) key agreement, Elliptic Curve Digital Signature Algorithm (ECDSA), and elliptic curve public keys for SSH SFTP as specified in RFC 5656. Only the required NIST curves at 256, 384, and 521 bits with uncompressed points are currently supported. Please see this page for more information on elliptic curve cryptography support.

CONFIGURING REMOTE SETTINGS

The remote settings page allows the administrator to configure web administration access and remote Application Programming Interface (API) access to Cerberus FTP Server. Cerberus allows remote access to the server administrator via a web browser-based interface and via the normal Cerberus FTP Server Graphical User Interface (GUI) when running in Windows Service mode.

For software developers, Cerberus exposes several APIs for controlling all aspects of the server using the SOAP web services.

Remote settings page of the Server Manager

SOAP Administration Settings

The remote access settings control HTTP and HTTPS web administration, as well as SOAP API access to Cerberus FTP Server.

When Cerberus is running as a Windows Service, the GUI connects to and communicates with, the Cerberus Windows Service through a remote access API called SOAP. The Cerberus Windows Service listens for SOAP connections on the Port specified under the Remote Settings page. That port must be available for Cerberus to listen on, or the GUI will be unable to connect to the service.

HTTP Port	The port that the SOAP service and web administration pages will be served from.
Use Secure HTTP (HTTPS)	Select this option to allow only secure HTTPS connections for the web administration and SOAP access. A restart of the underlying Cerberus FTP Server Windows Service is required after changing this parameter.
Allow Remote SOAP Access	Enable SOAP-based remote access. SOAP is an API for connecting programmatically to the server. When this setting is enabled, applications can make SOAP calls to the server from outside the local machine (subject to authentication). **NOTE:** Local SOAP access is always enabled. The Cerberus UI requires SOAP access to enable communication between the UI and the underlying Cerberus Windows Service.

Advanced SOAP Settings

You can control what SSL protocols are supported, as well as what ciphers to allow for SOAP-based SSL connections. Changes to these settings require a service restart.

Administrator Accounts

There is always a primary admin account, with full permissions to all server functions. The primary admin account is highlighted in green lettering in the administrator list.

Primary Admin Username	The username used to access the web administration page. This username is also used for basic authentication when using the SOAP web services API to access the server.

| Primary Admin Password | The password used to access the web administration page. This password is also used for basic authentication when using the SOAP web services API to access the server.

NOTE: This is also the username and password used when accessing Cerberus as a Windows Service from the Cerberus GUI. Normally, administrators won't be prompted for this password and the GUI will automatically connect to the service whenever it is started. |
| --- | --- |

The administrator can also control the server through web administration. The web administration feature has nearly the same capabilities as the desktop user interface. Most server functions can be controlled through web administration. We will be adding more features with every minor release until web administration mirrors the local graphical user interface.

WEB ADMINISTRATION AND SOAP ACCESS NO LONGER SHARE THE SAME PROTOCOL AND PORT FOR ACCESS.

Previous releases of Cerberus FTP Server used the same HTTP/S engine to provide web administration and SOAP API access.

In version 9.0 we switched web administration to the custom HTTPS engine we've been using for years for our HTTPS web client. The new HTTPS engine is fast, flexible, and secure.

SOAP remote administration runs on the same port and in the same way that it always has. Nothing has changed for SOAP access.

Web administration will simply require adding a new HTTPS Admin or HTTP Admin listener on any IP and port you wish to enable web administration on again. It will have to be on a different port than SOAP access now.

To add a new HTTPS web admin listener:

1. Open the **Server Manager.**
2. Select the **Interfaces** page.
3. Press the plus button to add a new listener. The **Add New Listener** dialog will appear.
4. Select the IP address you want the web administration listener to be on.
5. Select either **HTTPS Admin** or **HTTP Admin** as the listener type to add a new listener on the selected IP address using either HTTPS or HTTP.
6. Press the **Add** button to add the new listener.
7. Press the **Save** button on the Server Manager to save the new addition and activate the listener.

SECONDARY WEB ADMINISTRATION ACCOUNTS

You can assign additional web administration users, and limit their access to different aspects of the server like user management, reporting, ect.

Secondary web administration users can be managed on the Remote page.

Note: Secondary web administration users cannot access the SOAP API. Only the primary admin user can use the SOAP API at this time.

Cerberus FTP Server 6.0 Enterprise edition and above support collecting and reporting detailed session and file access statistics using an ODBC-compliant database. A database connection must be configured in Cerberus before the server will begin collecting statistics. The reporting database connection will also be used by the Reporting Manager for generating reports.

The following databases are currently supported (others may work with appropriate ODBC driver installed):

- Microsoft SQL Server 2012 LocalDB
- Microsoft SQL Server 2012 Express
- Microsoft SQL Server 2012 Enterprise
- MySQL Server 5.2 and higher

Database Configuration page for Statistics Collection

INSTALLING MICROSOFT SQL SERVER 2012 LOCALDB

The quickest and easiest database option is Microsoft Server 2012 LocalDB. LocalDB is a light-weight, embedded database option from Microsoft that is suitable for local, low-utilization database traffic. It has a relatively small footprint, and installs quickly. You will also need the Microsoft SQL Server 2012 Native Client ODBC driver for connecting to LocalDB. Links to both products are below.

To download and install LocalDB:

If you are running the 64-bit version of Cerberus FTP Server, make sure you download and install the x64 version of SQL LocalDB and the corresponding x64 SQL Server 2012 Native Client driver.

FOR THE 64-BIT VERSION OF CERBERUS FTP SERVER:

Microsoft SQL Server 2012 LocalDB x64 (for 64-bit Windows)

SQL Server 2012 Native Client (64-bit)

FOR THE 32-BIT VERSION OF CERBERUS FTP SERVER:

Microsoft SQL Server 2012 LocalDB x86 (for 32-bit Windows)

SQL Server 2012 Native Client (32-bit)

After installing SQL Server 2012 LocalDB and the SQL Server 2012 Native Client, you can go to the Server Manager's Reporting page and select the **SQL Server Native Client 11.0** driver.

SELECTING A DATABASE

If you are setting up a new database connection for the first time you will need to enable statistics collection and select a database ODBC driver. You can accomplish these tasks using the steps below:

1. Open the **Server Manager** and go to the **Reporting** page.
2. Open the Drivers select box and select the ODBC database driver appropriate for your database type (i.e. ODBC Driver 13.1 for SQL Server for a remote connection to Microsoft SQL Server).

For Microsoft SQL Server installations other than LocalDB, we recommend downloading and installing the **Microsoft ODBC Driver 13.1 for SQL Server**. Some reporting features may not work with the default SQL Server ODBC driver installed on most machines.

3. The connection parameters available for your driver type will now appear and must be filled in.
4. After filling in the ODBC driver connection parameters, press the Connect button to test your connection.
5. If there are no errors after pressing the **Connect** button, press the **Create Tables** button to create the necessary database tables for Cerberus to write to the database.
6. If **Create Tables** was successful then you are finished setting up your connection.
7. Check the "**Enable Statistics Collection**" checkbox.

The **Connect** button will test that Cerberus can establish a connection to the database, and run a quick search for the necessary Cerberus statistics tables. If this is the first time connecting to the database, and the tables do not exist, click the **Create Tables** button to allow Cerberus to try to create the necessary tables on the database.

If you are using a database that requires a username and password, the user account must have permission to create a database, and tables in the database. Statistics collection and reporting will not work if the user account does not have create database and create table permissions.

Once you have verified a working database connection, and that the database and tables exist, select the **Enable Statistics Collection** checkbox to enable statistics collection.

DATABASE BACKUP AND RESTORE

The Backup and Restore buttons are currently only supported when connecting to Microsoft SQL Server databases. The buttons will be disabled when any other database type is selected. The buttons allow downloading a copy of a local database, and later restoring it. Note, the database will be saved to the local machine where the database is running. If you click the Backup button for a remote database, the database will be saved to the selected path *on the remote server running the database*. The same goes for restoring a database. The database must be on the local machine.

Non-SQL Server databases should be backed up using whatever backup system is native to that database.

CONFIGURING ADVANCED SETTINGS

The advanced settings page contains options for passive mode, running as a Windows Service, network buffers, and power management.

Advanced page of the Server Manager

SEND AND RECEIVE BUFFERS

These settings control the size of the buffers used for data transfers. Cerberus will read and write packets of this size for send and receive operations.

Socket Send	Specifies the total per-socket buffer space reserved for sends. This value is in bytes.
Socket Receive	Specifies the total per-socket buffer space reserved for receives. This value is in bytes.

FTP Send	The size of the send buffer used for reading file data and writing data to the network for unencrypted FTP transfers, in bytes
FTP Receive	The size of the receive buffer for reading network data and writing data to files for unencrypted FTP transfers, in bytes.

ADVANCED WINDOWS SETTINGS

These settings are only available on Windows NT and higher.

Install as a Windows Service	If enabled, installs Cerberus FTP Server as a Windows Service. After selecting this option and pressing "Ok" to close the Server Manager the user may be prompted for an account with permissions to add a service to the system. **IMPORTANT:** Remote access using SOAP must be enabled for the administrator to be able to access the GUI when running as a Windows Service. If you haven't already enabled Remote access then selecting *Install as Windows* Service will automatically enable it and prompt you for a remote access password.
Use optimized file sending	Uses the built-in Windows API for fast file sending.
Respond to power management events	If enabled, Cerberus will attempt to gracefully shutdown and startup in response to power suspend and resume events. May allow more graceful recovery from suspending and resuming the system.
Shutdown Server when Windows is shutting down	Detects operating system shutdown or restarts and tries to gracefully terminate all connections and ensure all server settings are saved.

THE IP MANAGER

GENERAL SETTINGS

The Cerberus FTP Server IP Manager allows an administrator to selectively allow or deny access to the FTP server based upon IP address. The IP manager functions in one of two policy modes, either denying any IP addresses listed from logging into Cerberus FTP Server (functioning as a Blacklist), or only allowing IP addresses listed to log in (a Whitelist). The policy mode is controlled by a radio button at the bottom of the **General** tab page.

General page of the IP Manager

The IP list shows the IP address or IP address range and how long that address or address range is blocked for. Possible options for block time are "Forever" (Blacklist mode), "Never" (Whitelist mode), or a date/time value.

If a date/time value is present, the IP address or IP address range is blocked from connecting until that date/time has elapsed (Blacklist or Whitelist mode). You can change how long an IP address entry is blocked for by right-clicking on that IP entry and selecting "Change Time" from the menu that appears.

ADDING A SINGLE IP ADDRESS TO THE IP MANAGER POLICY

IP addresses can be managed individually, or whole ranges of addresses can be affected by the current policy. To add a single address to the current policy, make sure the "Assign a range of addresses" checkbox is unselected. Then, enter the IP address you wish to add to the first IP address box. Finally, click the "Add" button immediately below the IP address box.

Adding a range of IP addresses to the IP manager policy

To add a range of addresses, first ensure the "Assign a range of addresses" checkbox is selected. Then, enter the beginning IP address in the "IP From" box and the ending IP address in the "IP To" box. The range will be interpreted as a contiguous range of addresses to block or allow. Finally, click the **Add** button immediately below the IP address box.

CIDR Support

You can also enter a range of IP addresses in CIDR notation using the CIDR edit box. You can enter one CIDR range or multiple CIDR ranges. To enter multiple CIDR ranges, separate each CIDR range with a space or comma. The CIDR address will be converted to a contiguous range and added to the IP Manager list.

Deleting a IP addresses from the current policy

To delete either an IP address or range of IP addresses from the current policy, select the item from the "IP Addresses" list view box. Once selected, press the Delete button. You can also select and delete multiple items at once from the IP manager by ctrl or shift-clicking multiple items in the list box. NOTE: You can also delete an IP address or a range of IP addresses by right-clicking on the selected IP and selecting "Delete" from the menu that appears.

Searching for an IP Address

You can use the "Find" button at the top of the IP list box to search for an IP address in the list box. The "Find" button will select the first IP address or range of IP addresses containing the IP address you are searching for.

The "Auto-Blocking" page

The other use for the IP manager is the ability to configure an auto-blocking policy for the FTP server. Administrators can use the auto-blocking policy to help prevent DoS (Denial of Service) and brute force password guessing. If the auto-blocking policy is enabled, a user that continually fails to log into the server will be blocked from trying after a certain number of failed attempts. The number of failed attempts and the length of time the IP address will be blocked from attempting to log in can be configured from the "Auto-Blocking" page.

When **Enable Auto-Blocking** is enabled a failed attempt is logged whenever a user enters an incorrect password or tries to login with an invalid username. If **Enable DoS Protection** is selected then any attempt to connect to the server will be counted towards auto-blocking, even if the connection doesn't attempt to authenticate. This can help prevent DoS attacks that try to tie up connections and overwhelm the server. DoS Protection can also be useful for services continuously probing the server with garbage data attempting to find security vulnerabilities. However, a successful login from an IP address resets the "Failed login attempts" counter to zero for the IP address.

The number of failed login attempts can be configured from the **Pre-Blocked Settings** frame. The **Time before login counter reset** edit control can be used to set the amount of time that must elapse before the **Failed login attempt** counter is reset.

The length of time an address is blocked can be configured using the **Auto-Block Timeout** setting. Select the Forever radio button to block a flagged IP address indefinitely, or select the "Block for X minutes" radio button to set the length of time the address is blocked. Once an address is blocked, the timeout period must elapse before the address is allowed to log in again.

IP addresses that have recently failed logins, but have not yet exceeded the **Failed login attempt** threshold, are displayed in the **IP Addresses being "watched"** list view. You can freely delete an address from the list view. Deleting the address has the effect of resetting the **Failed Login attempt** counter for that address to zero.

Auto-blocking page of the IP Manager

IMMEDIATELY BAN THESE USERS

Certain usernames are often tried by automated bots. You can configure Cerberus to automatically block the IP of any connection that attempts to login using one of these banned usernames.

DIFFERENCES IN AUTO-BLOCKING BETWEEN BLACKLIST MODE AND WHITELIST MODE

How auto-blocking works differs depending upon whether the IP manager is functioning in Blacklist or Whitelist mode. If the IP manager is functioning as a Blacklist (denying addresses listed in the IP

manager), then whenever a connection exceeds the failed login attempt threshold, that connection's IP address is added to the deny list.

Auto-blocking works differently for Whitelist mode (allowing only addresses listed to login to the server). In Whitelist mode, whenever a failed login attempt exceeds the failed login threshold, the IP address is either removed from the IP manager's list of allowed IP addresses (if auto-blocking is set to block failed logins forever) or blocked for the Auto-Block Timeout period. The exception is if the IP address is part of a range of IP addresses. If an IP address is part of a range of allowed IP addresses, that range is not deleted.

The Event Manager

About Event Rules

Available in **Cerberus FTP Server Enterprise edition**, the Event Manager allows an administrator to configure email notification, perform file operation or batch file actions, and carry out certain server operations based off of server events.

Event rules are based on the simple premise that an event occurs that triggers an action. There are several different rule types, and for each rule type there is a corresponding event that can trigger that rule.

You can further restrict a rule by specifying additional conditions on the event that must exist before the rule's actions are taken.

For example, suppose you have a folder into which customers can upload files. You can set up an event rule that monitors that folder, and when someone uploads a file into that folder, the rule moves the file to another folder, and then sends an e-mail to an administrator informing them that a file has been moved.

You can also set up a rule that only moves particular files. For example, you can configure the rule to move only the files that end in .zip, or you can route particular files to different folders.

An event rule consists of a triggering event (e.g. a File Transfer), any optional conditions affecting that event (e.g. uploaded by a specific user), and the resulting actions that are carried out (e.g. moving the file, or sending an email to an administrator). You can modify your rules any time in the event manager.

The Event Targets page

The Event Targets page allows an administrator to add email servers, executable files, and HTTP endpoints as event targets. Many of the actions you can invoke as part of an event rule, or scheduled task, require an event target. For example, the "Email someone" action requires an email server, and the "Launch an executable" action requires the file path to the executable file. Those event targets can be defined here.

There are also certain server actions that can require an SMTP server, like public file sharing, or password expiration notification. You will first need to add at least one SMTP server here before the server can carry out those operations.
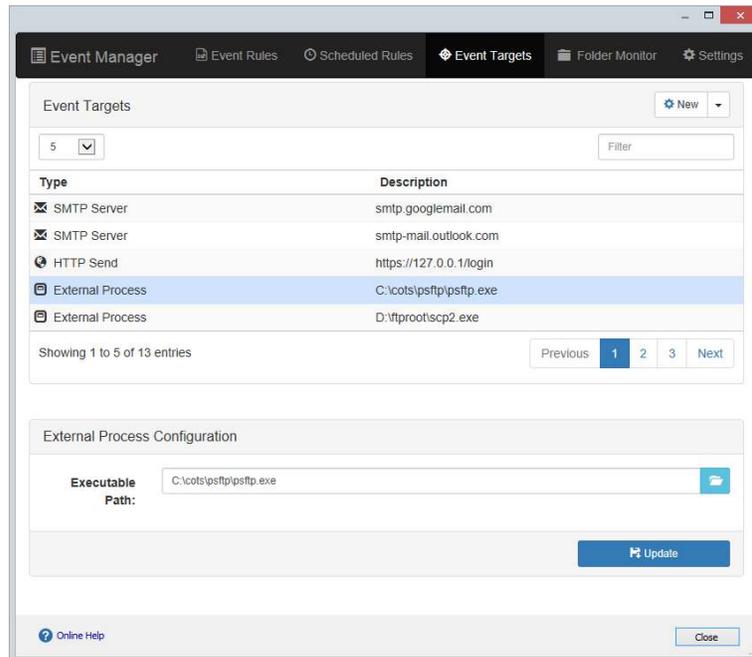
There are three different type of event targets you can add for use in event rules and scheduled tasks.

SMTP Server Targets

You can add SMTP servers using the SMTP Server Target box. Cerberus currently supports the SMTP protocol, including SMTP with SSL encryption and STARTTLS. If your server requires it, SMTP server credentials can be configured by selecting the **SMTP Authentication** checkbox.

Executable Targets

Cerberus can be configured to launch an .exe, .bat, or .com file as an action for any event. Just select a file path and press the "**Update**" button to make an executable target available for selection when adding and editing rules. Command line options for the executable are specified on a per action basis from the rule editing page.

HTTP Post Targets

This options allows you to specify a URL that will receive an HTTP or HTTPS POST containing all of the rule's variables. Variables are included in a POST request using **application/x-www-form-urlencoded** encoding.

Adding a New Event Target

Press the **New** button at the top of the Event Targets page. A dialog will prompt you for the type of target you wish to add.

Modifying an existing event target.

Select the event target in the Targets list. An edit section for that target will appear below the event targets list. Press the **Update** button after making your changes to save those settings to the server.

The Rules page

The Rules page provides an overview of all of the rules you have added. From this page you can Add, Delete, Clone, or Enable and Disable a rule.

You can enable or disable a rule from this page. Whenever a rule is disabled, that rule is no longer checked whenever the system generates an event that would normally trigger the rule.

Selecting a rule from the Event Rules table will open up a summary of the rule for editing.

Editing a Rule

Select a rule from the rules list to open it for editing.

Rule Editing in the Event Manager

A rule is defined by the type of event that triggers it. Each rule has a single event type associated with it. When that event occurs, any rules associated with that event type are triggered. The following rule event types are available:

File Transfer Event	This event is triggered whenever a user uploads a file to the server, or downloads a file from the server through an authenticated Cerberus account. This event is not generated for public share file downloads. There is a separate event for public file downloads.
IP Blocked Event	This event is triggered whenever the server adds an IP address to the block list.
User Account Blocked Event	This event is triggered whenever a user account is locked out because of a policy violation (too many failed login attempts).
User Disable Date Elapsed	This event is triggered whenever a user account is disabled because the disable date for the account has elapsed, or because the account has exceeded the last login time threshold.
Account Password Expiring Event	This event is triggered when an account password is set to expire. The number of days before expiration that this event is sent is based upon the password expiration policy settings.
New Account Request Event	This event is triggered when a new account request is submitted through the HTTP/S web client.
Login Event	This event is triggered whenever a user attempts to login to the server.
Logoff Event	This event is triggered whenever a user attempts to login to the server.
Directory Created Event	This event is triggered whenever a user creates a directory on the server.
File Deleted Event	This event is triggered whenever a user deletes a file or folder on the server.

File Move/Copy Event	This event is triggered when a file or directory is moved or copied by a user.
Upgrade Available Event	This event is triggered whenever the server detects that a new version of Cerberus FTP Server is available.
Public File Share Event	This event is triggered whenever a public file share link is generated for a file by a user. A public file share link is generated whenever a user uses the Share or Email button in the HTTPS web client to generate a new public link.
Public File Download Event	This event is triggered whenever a publicly shared file is downloaded from the server. This event will not be generated for a file download by an authenticated (logged in) Cerberus user.
Backup Server Synchronized	This event is triggered after the server attempts to synchronize settings to a backup server.

ADDING A NEW RULE OR EDITING AN EXISTING RULE

TO ADD A NEW RULE:

1. Go to the **Event Rules** page of the Event Manager
2. Click the **New** button. The **Add a New Rule** dialog will appear.
3. Select the **Rule Type** for your new rule option. The rule type will determine what server event triggers this rule.
4. Enter a name for your rule in the **Rule Name** edit box.
5. Press the **Add New Rule** button on the Add A New Rule dialog to save and add the new Event Rule. The event rule will be selected and ready for editing on the Event Rules page.

TO EDIT AN EXISTING RULE:

1. Go to the **Event Rules** page of the Event Manager
2. Select the name of the existing rule you wish to edit from the event rules table. The event rule should appear and be ready for editing.

CHANGING THE NAME OF A RULE

You can change the name of an existing rule by selecting it in the rules table. You can then modify the Rule Name under the Rule Summary section. After entering the new Rule Name, press the Update button attached to the Rule Name text field.

You can add a new condition to an event rule by pressing the Create button in the Event Conditions header. The new condition section will appear below the header.

A rule's actions are carried out whenever that rule's event trigger happens. For example, a Login Event rule will be triggered whenever a user logs into the server. Conditions (also called filters) can be placed on rules to further modify if an event matches a rule. For example, a Login Event rule can have a filter placed on it that requires the username of the user logging in to match a specific name, or be in a list of names, before the rule's actions are invoked. There are three modes that influence how conditions or filters are applied.

RULE MATCHING MODES

The three rule matching modes are:

Match All Events	This rule will always be triggered whenever the rule's event occurs.
Match If Any Filters Match	This rule will be triggered whenever the rule's event occurs and if **any** of the conditions listed are fulfilled
Match If All Filters Match	This rule will only be triggered whenever the rule's event occurs and if **all** of the conditions listed are fulfilled

RULE VARIABLES

Each event type has specific variables that can be used as part of a condition or action. A rule condition consists of a variable, a comparison operation to perform on that variable, and a set of values to compare the variable to. For example, an IP Blocked event has an **{{IP}}** variable associated with it that contains the IP address that was blocked. You can use the variable in a condition to help decide if the event should trigger the rule.

You can determine what rule variables are available for each event type by looking in the **Rule Variables** combo box.

COMPARISON OPERATIONS

A condition is basically a comparison operation of an event variable to a set of values. The comparison operations you can perform are detailed below:

- \> (Greater than or Equal To)
- ≥ (Greater than)
- < (Less than)
- ≤ (Less than or Equal To)
- = (Equal To)
- != (Not Equal To)

- Contains
- Does Not Contain
- Starts with
- Ends with
- Regular Express match

Once a comparison operation is selected, you can enter the **values** to compare to. There is a text field labeled "**Values"** below the comparisons select control that you use to enter values to compare the rule variable to. Multiple values can be entered by separating the values with a comma. Each value is checked, and if any are a match then the condition is considered fulfilled (or true).

Press the **plus (+) button** next to the Values text box to add the new rule condition to the event rule.

The new event condition will appear at the bottom of the Event Conditions section.

DELETING AN EVENT CONDITION

You can delete an existing event condition by pressing the red X button next to the event condition.

RULE ACTIONS

Rule actions are the operations the administrator wishes the server to carry out in response to server events that match their rule conditions. Event actions can be of two types:

1. Normal top-level actions that get executed sequentially, or
2. Failure actions that get executed whenever the event action they are associated with fails

Actions are normally executed one after the other, in a sequential order. Failure actions are always associated with a top-level action, and only get executed if the action they are associated with fails. The failure action is executed right after the action it is associated with.

Each top-level action has a "**Stop on Failure**" option. If the "Stop on Failure" option is checked, no further actions will be executed for the event rule if the action fails (other than any failure action associated with the top-level action).

ADDING RULE ACTIONS

When an event matches all of the conditions of a rule then the rule actions are carried out. The current rule actions allow an administrator to:

- Send an email message detailing the event that occurred
- Send an email session report of all user activity when a user logs off
- Launch an external process
- Perform a file copy, move, delete or directory create or delete operation
- Perform a user or group delete or disable
- Add a configurable delay before the next action is invoked
- Create a backup file of the server configuration

Each action can have optional parameters such as the email name and address to send a message to, or the path from and path to for a file move or copy operation. In addition, rule variables can be specified as parameters for the external processes command line or file operation parameters. You can use a rule variable as a parameter and when the rule is actually triggered, the variable's value will be substituted for the variable. You specify variables by enclosing the variable in double brackets, i.e. **{{U}}**.

These instructions assume you have selected a rule for editing from the rules tables.

1. Go to the **ACTIONS** header section of the event rule.
2. Press the **Create** button in the ACTIONS header. The new action section will appear below the ACTIONs header.
3. Select an action from the Action drop-down list (i.e., Email someone)
4. Select any secondary actions associated with that action (i.e, an email server for emailing someone)
5. New fields will appear below the Actions drop-down lists based on the action and secondary action selected
6. Fill in the details for that action (i.e., an email address)
7. If you for the rule action list to stop executing if this action fails then select the "**Stop on Failure**" option for the action.
8. Press the **plus (+) button** to add the new action to the rule

The new event action will be added to the bottom of the Actions section. New actions will be added to the bottom of the list, and will be executed in the order they appear in the list.

Editing an Existing Rule Action

You can edit an existing rule action by selecting the **Action** button to the left of the event action. Selecting the Action button will bring up a menu of available operations you can perform on the event action.

Select the Edit Action button from the menu that appears to have the action selected in the Actions section.

DELETING AN EXISTING RULE ACTION

You can delete an existing rule action by selecting the **Action** button to the left of the event action.

Select the **Delete** button from the menu that appears to have the action deleted from the event rule.

CHANGING THE ORDER AN ACTION IS EXECUTED

You change the existing execution order of event actions by selecting the **Action** button of the event action you wish to change.

Select the **Move Action Up** or **Move Action Down** to swap positions with the action above or below the selected action.

CREATING A FAILURE ACTION

Each action can have a failure action associated with it. Failure actions are additional actions that only get executed whenever the action they are associated with fails. For example, you can add an "Email Someone" failure action to an action to email the administrator whenever the top-level action the failure actions is associated with fails. Or, you can try the action a second time as your failure action.

The same action options are available as failure actions as are available for top-level actions.

To create a failure action, create a new action as you normally would for a top-level action. Use the Move Up or Move Down action options to place the new failure action below the top-level action you wish it to be associated with.

Once the action you wish to associate as failure action is below the top-level action, select the "Assign as Failure Action" option from the Actions button next to the failure action. You will now see the action become indented under the top-level action, and the text "if fail then" appear in front of the failure action.

REMOVING A FAILURE ACTION

Removing a failure action just requires pressing the Action button associated with the failure action's top-level action, and then selecting the "**Detach Failure Action**" option from the menu that appears.

Detaching a failure option from a top-level action will make the failure action a normal top-level action again. You can then move it around, re-assign it as a failure action of another top-level action, or delete it.

EVENT TASKS

Event tasks are similar to event rules. However, rather than being triggered whenever an event like a file upload or directory creation occurs, event tasks are time-based, and occur on an admin-defined schedule.

Administrators can configure event tasks to occur once, or to repeat every minute, hour, day, week, weekday, month, or year.

You can create and edit event tasks on the Event Tasks page of the Event Manager.

ADDING A SCHEDULE TO AN EVENT TASK

Schedules can be added to event tasks in a similar way that event conditions are added to event rules.

1. Specify a Start Date for the scheduled task. If you do not specify a Start Date then the task will be executed immediately.
2. Select how often you want the task to repeat. You can select a period and frequency. For example, every 5 hours.
3. Press the **plus (+) button** to add the schedule to the task.

ADDING AND EDITING EVENT TASK ACTIONS

Please see the section on **adding and editing actions** in the **Event Rules help** section. The process is identical for scheduled tasks.

FOLDER MONITORS

The Folder Monitor page allows you to configure the server to monitor a top level folder and subfolders for setting up a file retention policy.

You can configure a directory and subdirectories to be monitored for files older than a specified time period. The directory will be checked at an administrator-defined interval, and files older than the specified age will be deleted.

Folder Monitor for configuring file retention policies

OTHER EVENT SETTINGS

The event settings page allows the administrator to configure settings like the email template logo, whether to include server information in event emails, and other global event settings.
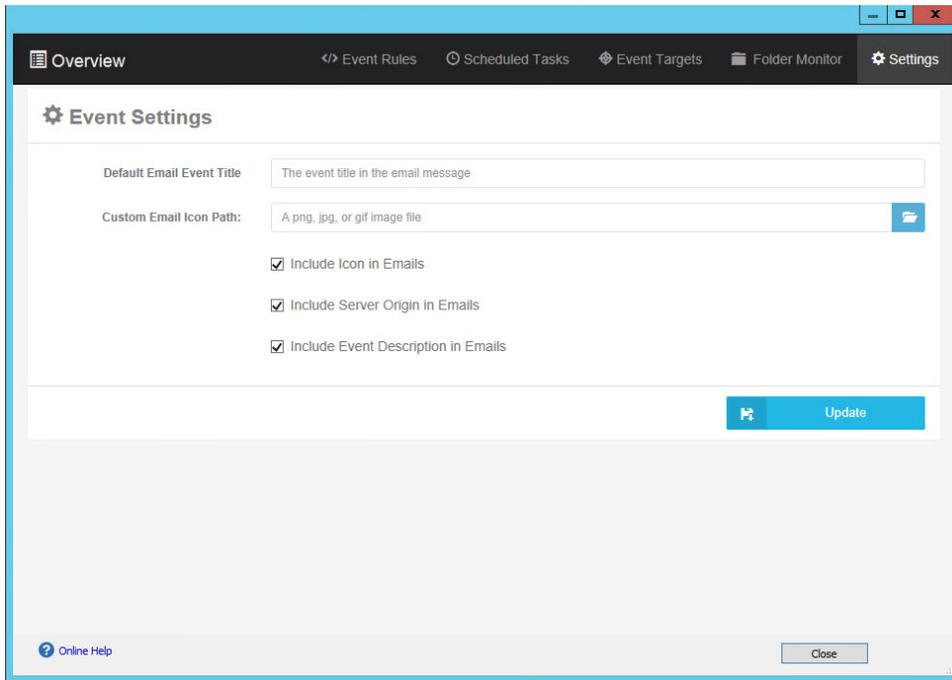
Default Email Event Title	The email heading title at the top of each event notification email.
Custom Email Icon Path	Allows the administrator to include their own icon logo with event notification emails, instead of the default logo icon.
Include Icon in Emails	Determines whether or not the default or customer email icon path is included with each event notification email.
Include Server Origin in Emails	Determines whether or not the server version and machine name are included with each event notification email.
Include Event Description in Emails	Determines whether or not the basic event description is included with each event notification email.

Administrators can use the statistics and reporting feature to generate detailed reports of client activity based on user names, dates ranges, and file access.

The Report Manager Overview Page

Generating a Report

Administrators can use the statistics and reporting feature in Cerberus FTP Server Enterprise edition to generate detailed reports of client activity based on user names, dates ranges, and file access. In addition to client activity, the administrator can also generate native account reports indicating account creation and last login dates.

NOTE: Using the Report Manager requires that a report database be configured.

Cerberus FTP Server Report Manager

An administrator can generate three types of reports:

- Login Sessions
- File Access
- User Account Status

FILE ACCESS REPORTS

File access reporting can by filtered by file name, date and time, and user name. The file reports contain important information about a file transfer, including:

- The full local path of the file
- The type of file operation performed on the file (upload, download, rename, delete, copy)
- The user that accessed the file
- The IP address the user accessed the file from
- The date and time the access occurred

- The protocol used to access the file (FTP, FTPS, SSH SFTP, HTTP, or HTTPS)
- The type of encryption used, if any, to perform the file operation

The administrator can also use the Include feature to decide what type of file activity to include in the report (downloads, uploads, file renames/moves, public file shares, etc.).

Cerberus FTP Server Professional is able to authenticate users against LDAP directory services. The **Lightweight Directory Access Protocol**, or **LDAP**, is an application protocol for querying and modifying directory services running over TCP/IP.

Administrators can easily integrate Cerberus and LDAP or LDAPS (LDAP over SSL). All you need are a few parameters describing the LDAP service.

What do I need to use LDAP Authentication?

An LDAP service and some information about the server hosting the LDAP service:

Server	This parameter is the FQDN or IP address of the LDAP server to search.
Port	The network port of the LDAP server.
Enable SSL	This checkbox determines whether the connection to the LDAP server is encrypted. The LDAP server must support encryption for this to work. Port 389 is the default port for unencrypted LDAP and port 636 is the default LDAPS port.
Base DN	The distinguished name to use as the search base.
User DN	The FDN of an account with read privileges to the LDAP server.
Password	The password for the User DN account. This password is encrypted when saved.
User DN attribute	The name of the uid attribute for a user in the directory.

By default, all LDAP users are assigned the same virtual directories and permissions. These defaults are configured under the **Default Group and Virtual Directory Mapping for LDAP Users** section of the LDAP Users page. However, if you wish to customize the directory and permission mappings for individual LDAP users then you can do so using the **Customize** button.
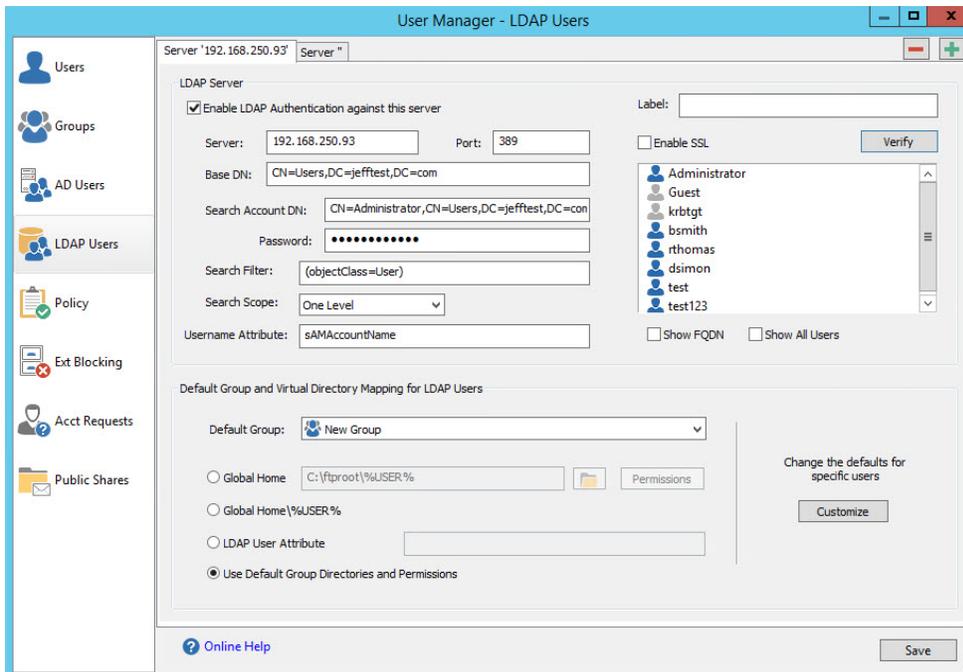
The **Customize** button allows you to override the default settings for a user by mapping individual LDAP users to Cerberus groups. The mapped LDAP users will receive the settings and virtual directories from the mapped group, instead of the defaults.

DEFAULT VIRTUAL DIRECTORY MAPPING FOR LDAP USERS

The *Default Virtual Directory Mapping* modes work as follows:

Global Home	Every LDAP account will use the directory specified under the "Global Home" edit box as the FTP root. This is the simplest option, and every LDAP user is assigned this one directory as their root folder. The Cerberus permissions on this folder can be restricted through the **Permissions** button to the right of the Global Home edit box.
Global Home\%USER%	Every LDAP account will use a subdirectory off of the "Global Home" directory that is the same as the account's name. This directory will be created automatically, if it doesn't exist, when the user logs in. The Cerberus permissions on this folder can be restricted through the **Permissions** button to the right of the Global Home edit box.

LDAP User Attribute	Every LDAP account will use the directory attribute defined here to determine what virtual directories to add to their account.
	This attribute can have multiple values, and each value will be added as a separate virtual directory.
	The default value will be a valid Windows directory path. By default, the last directory of the file path will be used for the virtual directory name, and the user will have full permissions to the directory path.
	The value can be customized into 3 semicolon separate components to customize the added virtual directory path into a full directory path, a virtual directory name, and a permissions set for the virtual directory.
	For example, the value for the attribute could be:
	C:\ftproot\user\andrew;home;2047
	The first part is the directory path, the second is the directory name, and the third is a bit mask indicating the permissions the user has for that virtual directory.
	The directory permissions field for a virtual directory is a simple bit mask. Permissions have the following values:

Permission	Value
DOWNLOAD	1
UPLOAD	2
RENAME	4
DELETE	8
CREATEDIR	16
LIST DIRECTORIES	32
LIST FILES	64
DISPLAY HIDDEN FILES	128
ZIP	256
UNZIP	512
SHARE	1024

	Just add the values up to achieve the desired permissions. e.g., Download, Upload, Rename, and Delete permissions would be (1 + 2 + 4 + 8) = **15**. Granting all permissions would be **2047**.
Use Default Group Directories and Permissions	The specified Cerberus Group will be used to determine what directories and what settings to apply to the LDAP user when they login, including any security requirements associated with the group.

OTHER LDAP DIALOG OPTIONS

The LDAP Accounts list box that enumerates LDAP accounts is only meant as an aid in determining if your LDAP connection is configured correctly. If you can get a successful listing of user accounts, then those accounts should be accessible to Cerberus during authentication. Some additional display options are detailed below:

Show FQDN	Display the fully qualified domain name of each enumerated object. **Note:**This setting has no effect on actual LDAP authentication.
Show All Users	If this option is checked, every account will be retrieved and enumerated in the LDAP Accounts list box. This can take a very long time if there are a large number of users. **Note:** This setting has no effect on actual LDAP authentication.

SETTING UP ACTIVE DIRECTORY AUTHENTICATION USING LDAP

The following steps detail the procedure for enabling LDAP Authentication to verify credentials against Active Directory. The steps are similar for connecting to other LDAP servers, such as OpenLDAP or ApacheDS.

1. Change the LDAP Server and Port attribute in the User Manager, LDAP Users tab to the host name and port number of the Active Directory:

   - e.g., Server: hostname.domain.com **or** an IP address:192.168.0.100

   - Port: 389 is the default for unencrypted LDAP connections. Port 636 is the default for LDAPS encrypted connections.

2. Change the Base DN to the proper base for the Active Directory.

   Simply specifying the base suffix will not work in this attribute. For Active Directory, it would usually be the cn=Users plus base suffix. e.g.: for domain *corp.cerberusllc.com* :

   CN=Users,DC=corp,DC=cerberusllc,DC=com

   or for local domain *corp.cerberusllc.local* :

   CN=Users,DC=corp,DC=cerberusllc,DC=local

3. Change the DN for the **User DN** bind attribute to a user with the right to read the Active Directory.

    Anonymous access to Active Directory is not allowed, so a bind account is needed. This is simply an account for Active Directory that has read ability on the attribute to which the user will authenticate. An example might be **cn=administrator,CN=Users,DC=corp,DC=cerberusllc,DC=local**. Enter the password for the user account. Note: This password will be encrypted in memory and before being saved to disk.

4. Change the **User DN Attribute**.

    This attribute is the one that the LDAP module will search for in Active Directory and attempt to match against the supplied FTP username. It is often the UID attribute on many LDAP servers. For example, if users login using their Common Name, the value of this attribute would be **cn**. For Active Directory, the login name is usually mapped to **sAMAccountName** as it is the attribute in Active Directory most like UID. For Active Directory, it is usually best to specify **sAMAccountName**.

5. Change the **Search Filter**.

    This string is an LDAP search string used to locate and filter the account in Active Directory. This filter can be used to make sure only certain types of objects are checked for authentication.

SEARCH FILTER EXAMPLES:

```
(objectClass=User)
```

The above filter will include only search entities that have the object class **User**.

```
(memberof:1.2.840.113556.1.4.1941:=cn=FTPUsers,CN=Users,dc=corp,dc=cerberusllc,DC=local)
```

The above filter will include all users that are member of the group **FTPUsers**. **Do not** attempt to add the uid search attribute here. Cerberus will automatically append an attribute filter to select the correct account based on the User DN Attribute, e.g., if the User DN Attribute is **sAMAccountName**, Cerberus will automatically create a string like

```
(&(objectClass=User)(sAMAccountName=ftpUser))
```

where *ftpUser* is the name of the user that attempted login.

6. Set the **Search Scope**.

    This setting controls how deep into the directory to search for users. This setting combined with the Base DN and Search Filter determines which users are matched for authentication. **One Level** is usually the best setting for typical Active Directory configurations.

7. Verify that the settings are correct by clicking the **Connect** button. You should see the user DNs from Active Directory that are able to log in to Cerberus FTP Server. Note: Unless "Use FQDN" is checked, only the value of the **User DN Attribute** will be displayed in the LDAP user list. It is this value that will be compared against the FTP username to determine an account match.

8. Select a Cerberus FTP Group to represent the virtual directories and permissions for LDAP users. Note that the "isAnonymous" setting on the group is ignored. The group cannot be anonymous.

Cerberus FTP Server is now configured for authentication against an LDAP server (Active Directory, in this case).

LDAP User to Cerberus Group Mapping

You can customize the directory and permission mappings for individual LDAP users through the **LDAP Directory Mapping** tab. Customizing an LDAP account is accomplished by mapping an LDAP user account to a Cerberus group account. This mapping will override the default Cerberus Group and directory mapping, specified on the LDAP Users page, for the mapped LDAP account.
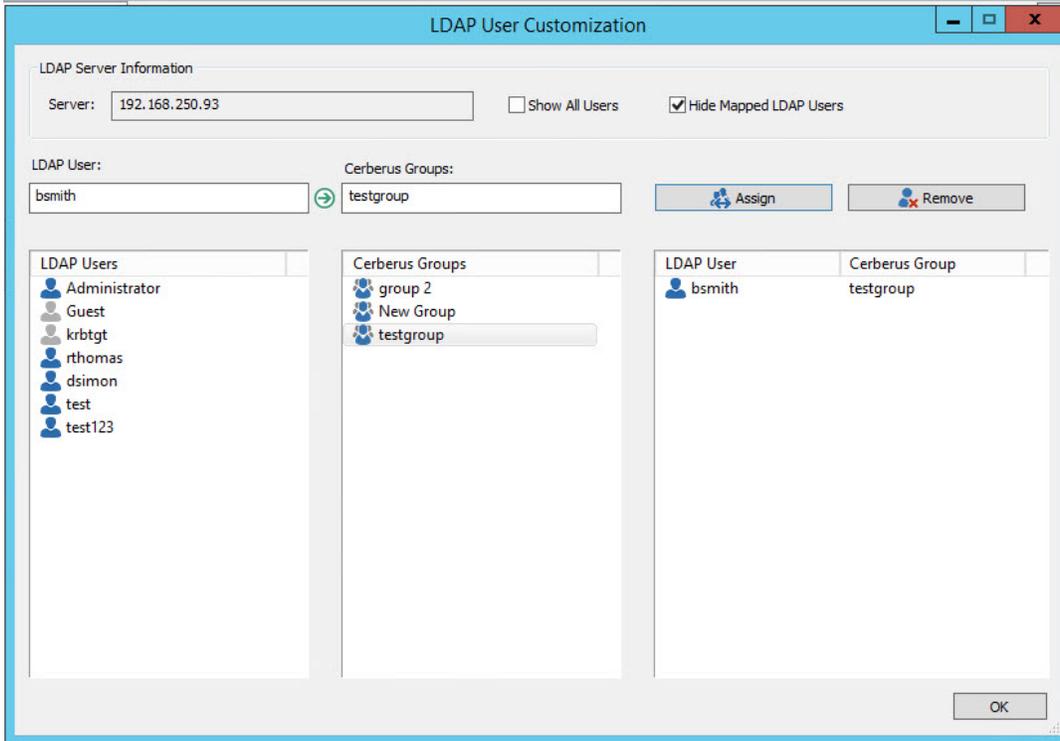
Configuration page for LDAP User to Cerberus Group Mapping

Creating an LDAP User to Cerberus Group Mapping

Mappings between an LDAP User and a Cerberus Group can be achieved by first selecting an LDAP user. Then, select an LDAP user (or simply typing the name of the LDAP user in the edit box) and then select a Cerberus Group. Select the **Assign** button and a mapping entry will be placed in the mapping list box to indicate the LDAP user will now have the same constraints and virtual directory mappings as the selected Cerberus Group.

Removing an LDAP mapping

To remove a mapping, simply select the mapped entry and press the **Remove** button.

ABOUT ACTIVE DIRECTORY INTEGRATION

Cerberus FTP Server Professional and Enterprise editions are able to authenticate users on a Windows domain (or the local NT account database), even if the computer Cerberus FTP Server is installed on is not the domain controller. The domain may be a pre-Windows 2000 domain (NT4), a domain configured to use Active Directory, or the local system account database (use "." as the domain for authenticating against local machine accounts). However, the machine Cerberus FTP Server is running on must be a member of the domain you wish to authenticate users against.

Configuring Cerberus to use Active Directory authentication simply requires enabling Active Directory authentication and telling the server the name of the domain to authenticate against. The rest of the configuration is automatic. Users are able to FTP into the server using the same username and password they use to log into their workstations on the domain. For the purpose of access to files and folders, the FTP user has the same access as the Active Directory user with the same name. All operations on the server by the user are carried out while impersonating the Active Directory user.

Important Security Consideration: There is an exception to impersonation for Active Directory authentication when using SFTP and **Public Key only** SSH authentication. The Active Directory user can still be authenticated with Public Key only authentication, but the Active Directory user cannot be impersonated. Only **Password** or **Public Key and Password** SSH authentication methods support AD user impersonation.

To allow Active Directory authentication, you will need to check the **Enable Windows Authentication for this Domain** checkbox under the User Manager's **AD Users** tab. Once checked, Cerberus will attempt to authenticate users from the domain listed in the **Domain** edit box.
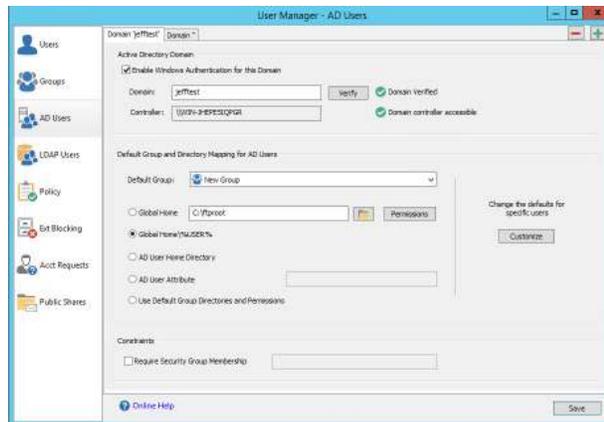
Active Directory Authentication page

Active Directory accounts are always configured for simple directory mode (See Adding a New User for more information about simple mode) if any mode other than **Cerberus Group** is selected for the *Default Virtual Directory Mapping* mode.

The *Default Virtual Directory Mapping* modes work as follows:

Global Home	Every AD account will use the directory specified under the "Global Home" edit box as the FTP root. This is the simplest option, and every AD user is assigned this one directory as their root folder. The Cerberus permissions on this folder can be restricted through the **Permissions** button to the right of the Global Home edit box. NTFS permissions for the AD user still apply.
Global Home\%USER%	Every AD account will use a subdirectory off of the "Global Home" directory that is the same as the account's name. This directory will be created automatically, if it doesn't exist, when the user logs in. The Cerberus permissions on this folder can be restricted through the **Permissions** button to the right of the Global Home edit box. NTFS permissions for the AD user still apply.
AD User Home Directory	Every AD account will use that account's home directory as the FTP root. The Cerberus **permissions** on this folder can be restricted through the Permissions button to the right of the Global Home edit box. NTFS permissions for the AD user still apply.
AD Directory Attribute	Every AD account will use the directory attribute defined here to determine what virtual directories to add to their account. This attribute can have multiple values, and each value will be added as a separate virtual directory. The default value will be a valid Windows directory path. By default, the last directory of the file path will be used for the virtual directory name, and the user will have full permissions to the directory path. The value can be customized into 3 semicolon separate components to customize the added virtual directory path into a full directory path, a virtual directory name, and a permissions set for the virtual directory.

For example, the value for the attribute could be:

C:\ftproot\user\andrew;home;2047

The first part is the directory path, the second is the directory name, and the third is a bit mask indicating the permissions the user has for that virtual directory.

The directory permissions field for a virtual directory is a simple bit mask. Permissions have the following values:

Permission	Value
DOWNLOAD	1
UPLOAD	2
RENAME	4
DELETE	8
CREATEDIR	16
LIST DIRECTORIES	32
LIST FILES	64
DISPLAY HIDDEN FILES	128
ZIP	256
UNZIP	512
SHARE	1024

Just add the values up to achieve the desired permissions. e.g., Download, Upload, Rename, and Delete permissions would be (1 + 2 + 4 + 8) = **15**.

Granting all permissions would be **2047**.

Use Default Group Directories and Permissions	The specified Cerberus Group will be used to determine what directories and what settings to apply to the AD user when they login, including any security requirements associated with the group.

Optionally, you can also configure a Security Group for FTP users. This will cause Cerberus FTP Server to check that the Active Directory user is a member of the listed Active Directory Global security group before allowing login. If selected, only members of the security group will be allowed to login.

AUTHENTICATING AGAINST MORE THAN ONE ACTIVE DIRECTORY DOMAIN

Cerberus FTP Server can be configured to authenticate against multiple domains. Select the **AD Users** page of the User Manager and select the ✚ icon in the top right corner. This will add a new domain tab to the AD User page. This new domain tab can be configured the same way as the default Active Directory domain tab.

UNDERSTANDING WINDOWS AUTHENTICATION

Active Directory user authentication is intended for experienced system administrators that understand the NT security model. Novice users, or users wishing to avoid the details of Windows security, should leave Windows Authentication disabled and stick with native Cerberus FTP Server users.

THE "GUEST" ACCOUNT

In Windows, the **Guest** account lets people log on to a computer when they don't have a personal account defined on the computer, in the computer's domain, or in any of the domains that the computer's domain trusts. Like the Administrator account, the **Guest** account is a built-in account with a fixed SID; although you can rename the account, it can't--by default--be deleted. Unlike the Administrator account, the **Guest** account doesn't require a password for logon, which is why it's disabled by default. A **Guest** account re-enabled by mistake would pose a significant security hole.

To help guards against this potential security hole, an administrator cannot enable Cerberus FTP Server's Windows authentication integration if the **Guest** account is enabled.

ACTIVE DIRECTORY USER TO CERBERUS GROUP MAPPING

By default, all AD users are assigned the same virtual directories and permissions. These defaults are configured on the Domain tab of the AD Users page. However, if you wish to customize the directory and permission mappings for individual AD users then you can do so using the **Customize** button. You can select individual AD accounts and map them to Cerberus group accounts, or, you can map AD group accounts to Cerberus group accounts. Configuring an AD user to group mapping will override the default Cerberus Group and directory mapping specified for all AD users.
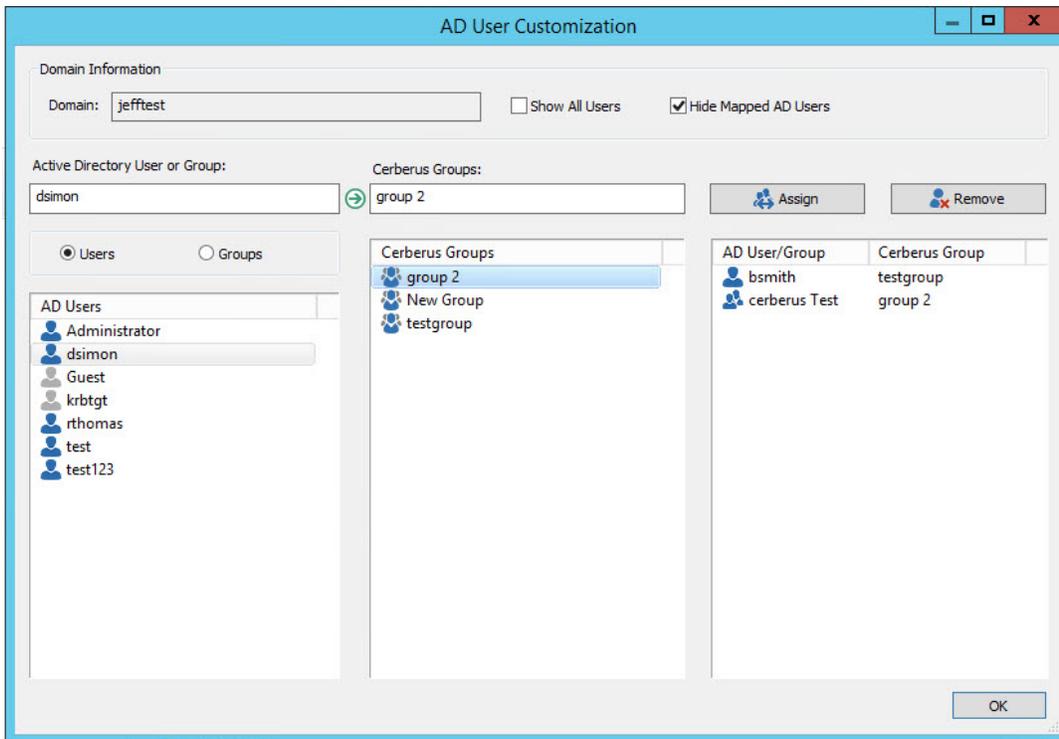
Configuration page for AD User to Cerberus Group Mapping

CREATING AN AD USER TO CERBERUS GROUP MAPPING

Mappings between an AD User and a Cerberus Group can be achieved by first selecting an AD domain. Then, select an AD user from the AD Users list box (or simply type the name of the AD user in the edit box) and then select a Cerberus Group. Select the **Assign** button and a mapping entry will be placed in the mapping list box to indicate the AD user will now have the same constraints and virtual directory mappings as the selected Cerberus Group.

CREATING AN AD GROUP TO CERBERUS GROUP MAPPING

Customizing each individual AD User to a Cerberus group can be a time-consuming task if you have many users, especially if you can divide up large groups of users into just a few groups.

To make maintaining large numbers of users easier, you can use the new AD group to Cerberus group mapping capability. On the AD User **Customization** page, you can map AD groups to Cerberus groups.

When an AD user logs into Cerberus, the server will check the **direct** AD group memberships for that AD user and see if there are any AD group to Cerberus group mappings. If a mapping is founds, the

virtual directories for that Cerberus group will added to the virtual root for the AD user. Only the virtual directories from the Cerberus groups are added to the AD user. No other constraints are transferred.

Note: The Default Group and Default Virtual Directory mappings are still applied to the user when AD group to Cerberus group mappings are present, unlike AD user to Cerberus user mappings.

REMOVING AN AD MAPPING

To remove a mapping, simply select the mapped entry and press the **Remove** button.

THE REGISTRATION DIALOG BOX

Using Cerberus FTP Server for commercial use past the 25 day evaluation period requires a license key. Once you have purchased and received a license key, you need to enter the license key details in the registration dialog box.
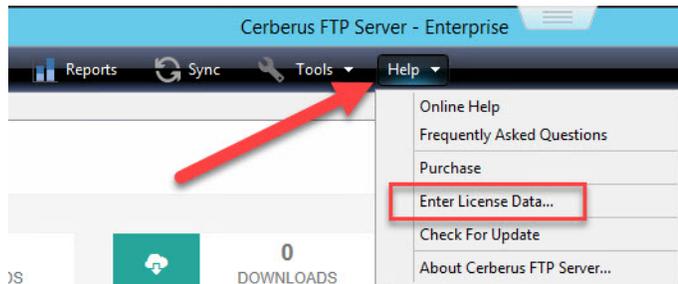
To open the registration dialog box, go to the **Help** menu and select the **Enter License Data...** menu item. A box similar to the one below will prompt you to enter your registration code. Open your license email and copy everything starting at and including "-----BEGIN REGISTRATION-----" all the way until and including "-----END REGISTRATION-----". Paste the copied text into the large edit box.
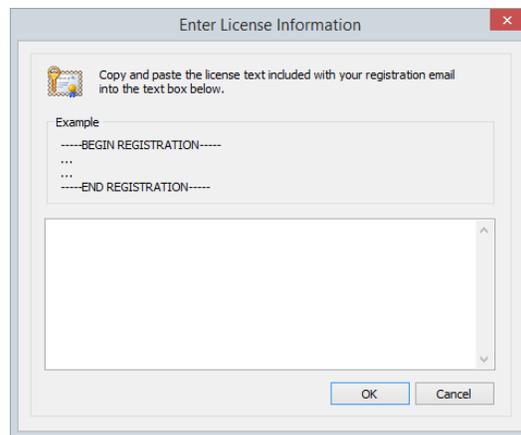
Press the OK button. Another dialog box will appear, after you press enter, to inform you of correct or incorrect registration information. Please note that a service restart is required after entering a new license key. Cerberus will prompt you to restart after successfully entering a new license key.

Once you have successfully registered Cerberus FTP Server, the "About" dialog box will display the registration contact name, company name, purchase date, and for how long the license entitles the user to free upgrades.

Cerberus FTP Server Professional and Enterprise editions support automatically replicating users and settings from a primary or master server to other running Cerberus FTP Server machines. This capability allows administrators to maintain active backups of a main server in case of failure, or to ensure a cluster of servers contain identical configurations while only having to manage one machine.

The Synchronization Manager is used from the machine you want to use as the primary server. The Manager allows an administrator to designate one or more running Cerberus instances for syncing. With the exception of machine-specific configuration information, the other servers become exact copies of the primary server. Each server that is being synced to will have its users and settings replaced by the users and settings on the primary server.

The replication process can be configured to occur at regular intervals to ensure that all of your synced servers are kept current with the primary server.

The server instances must all be running the same version, and have unique license keys.

Cerberus Synchronization Manager

BACKUP SERVER REQUIREMENTS

To add a backup server to the synchronization list, that backup server must be running the same version of Cerberus FTP Server as the primary server and have a valid, unique license key. All users, groups, and other settings will be synchronized to the backup servers, **except**:

- License keys
- SOAP and remote/web administration settings
- Server certificate, private key, CA, and CRL security settings
- Client SSH public key files

AVAILABLE SETTINGS

BACKUP SERVER

Server	The hostname or IP address of a backup server
Port	The remote administration port of the backup server to connect to.
Secure Connection	Instructs this server to connect using TLS/SSL security to the backup server. This setting must always be enabled.
Username	The remote administration account username on the remote server.
Password	The remote administration account password on the remote server. This value will be encrypted before being saved to disk.

SYNCHRONIZATION SETTINGS

These are basic server synchronization settings. You can enable and set server synchronization intervals using these settings.

Enable Server Synchronization	Checking this setting will enable automatically replicating this server's users and settings to the added backup servers. This replication will occur at the sync interval, in minutes.
Sync Interval	How often, in minutes, to synchronize this server's setting to the backup servers.

SERVER CERTIFICATES

WHAT IS A SERVER CERTIFICATE?

The most common use of a digital certificate is to verify that a user (or server) sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

There are generally two options for obtaining a digital certificate (and the accompanying private key).

1. You can generate your own self-signed certificate using the Cerberus FTP Server Getting Started Wizard.

2. You can obtain a certificate from a recognized Certificate Authority

Which is more appropriate really depends upon your goals. If you just want to make sure that client and server connections are securely encrypted then a self-signed certificate is all you need. It has the benefit of being easily created through Cerberus and completely free.

If your goal is to make sure that your clients can verify that the server they are connecting to is legitimate and to ensure they don't see any warning messages about being "unable to verify the server" then using a certificate signed by a trusted certificate authority is required. You will have to contact one of the recognized Certificate Authorities such as Comodo, Thawte, Verisign or one of the many other recognized Certificate Authorities and request a server certificate (for a price).

CAN I JUST USE A SELF-SIGNED CERTIFICATE?

Yes, but your users will not be able to easily verify your server's identity. If you are using Cerberus FTP Server exclusively on your own private network, or are just looking to test Cerberus FTP Server out before deploying it on the Internet, a self-signed certificate is more than adequate. You can always change your certificate later to one signed by a recognized Certificate Authority.

MORE INFORMATION

- Creating a Certificate Signing Request

- Creating a Self-Signed Certificate

- Importing a 3rd Party Certificate

- Exporting a certificate from the Windows Certificate Store for use by Cerberus FTP Server

CREATING A CERTIFICATE SIGNING REQUEST

The first step in requesting a certificate from a Certificate Authority (CA) is usually creating what is called a Certificate Signing Request (CSR). Cerberus FTP Server 5.0 and higher allow you to easily create a CSR using a simple wizard. You can start the CSR Wizard by opening the **Tools** menu and selecting the **Generate a CSR** menu item.

Certificate Signing Request Wizard

Fill in all of the required fields for the CSR and then press the **Generate** button. After you select the Generate button a directory selection dialog will appear to allow you to specify a directory to save the private key and certificate signing request.

Make sure you save both the private key file, and the CSR file. You will need both of these files.

You will submit the CSR file to your CA and keep the private key file. Once your CA has approved your CSR they will issue you a signed public certificate file. This signed public certificate file from your CA and the private key file, created during your certificate signing request, together represent your server public and private key pair.

The CA will usually provide several different format options for the signed public certificate. The preferred format is a PEM-formatted certificate (the same format Apache web server uses). PEM is also called a Base64 encoded DER certificate. You can tell if a certificate is in this format by opening it in a text editor, and looking for the beginning and ending lines "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----".

ASSIGNING YOUR CERTIFICATE AND PRIVATE KEY IN CERBERUS FTP SERVER

The final step involves assigning the signed public certificate file and private key file as your public key pair in the Security page of the Server Manager.

1. Open the Server Manager by selecting the **Configure** -> **Server Manager** item from the main menu.

2. Select the **Security** page.

3. Under the Server Key Pair group, Click the file selection button next to the **Certificate** edit control.

4. A file open dialog will appear that will allow you to select the public certificate provided from your certificate authority.

5. Under the Server Key Pair group, Click the file selection button next to the **Private Key** edit control.

6. A file open dialog will appear that will allow you to select the server's private key. This file was generated when you first created your CSR.

7. Most CAs provide a CA bundle file that contains all of the intermediate CA certificates leading up to your signed certificate. If your CA provides a CA bundle file, download and assign that file to the **CA File** field.

DIGITAL CERTIFICATE SUPPORT

There are generally two options for obtaining a digital certificate (and a private key).

1. You can generate your own certificate using the Cerberus **Create Cert** button.

2. You can obtain a certificate from a recognized Certificate Authority

Which option is more appropriate really depends upon your goals. If you just want to make sure that client and server connections are securely encrypted then a self-signed certificate is all you need. It has the benefit of being easily created through Cerberus and completely free.

CREATING A SELF-SIGNED CERTIFICATE

If you just want to be sure that connections are security encrypted then a self-signed certificate is sufficient for your organization.

STEPS TO CREATE A SELF-SIGNED CERTIFICATE:

1. Open the Server Manager by selecting the **Configure -> Server Manager** item from the main menu.

2. Select the **Security** page.

Security settings page of the Server Manager

3. Click the **Create Cert** button

4. A **Create a Self-Signed Certificate** dialog will appear that asks for certificate details. The organization details that you use will be displayed to the FTP client user when they securely connect to your server. The Key Type should normally be RSA for maximum client compatibility. They Key Length value controls how strong the generated keys are and should normally be set to 2048. The default validity period for the certificate is 1095 days (3 years). Press the **Generate** button to create the certificate.

Create a Self-Signed Certificate dialog

5. A self-signed certificate will be created and Cerberus will be automatically configured to use it.

6. Click **Ok** to close the Server Manager. If no certificate was previously being used then Cerberus will configure itself immediately to use the new certificate. You may need to restart the FTP server service if you were overwriting a previous certificate.

USING A CERTIFICATE CREATED BY A 3RD PARTY CERTIFICATE AUTHORITY

If your goal is to make sure that your clients can verify that the server they are connecting to is legitimate, and to ensure users don't see any warning messages about being "unable to verify the server", then you must use a certificate signed by a trusted certificate authority. You will have to contact one of the recognized Certificate Authorities such as Comodo, Thawte, Verisign or one of the many other recognized Certificate Authorities and request a server certificate (for a price).

STEPS TO IMPORT A 3RD PARTY CERTIFICATE:

1. Ensure that you have a digital certificate and private key in a format that Cerberus FTP Server understands. First, you will need to generate a new certificate (either by purchasing one from a public Certificate Authority, or you can install a Certificate Authority in your domain). You need to have a public certificate and a private key along with the passphrase for the private key.

2. Open the Server Manager by selecting the **Configure** -> **Server Manager** item from the main menu.

3. Select the **Security** page.

4. Under the Server Key Pair group, Click the **Certificate ...** button.

5. A file open dialog will appear that will allow you to select the public certificate provided from your certificate authority.

6. Under the Server Key Pair group, Click the **Private Key...** button.

7. A file open dialog will appear that will allow you to select the server's private key. If your public and private key are in the same file then set this path to be the same as the Certificate file path. *NOTE*: Cerberus understands both DER and PEM encoded certificate formats.

8. **Needs Key Password** - Check this option if the digital certificate is encrypted.

9. **Password** - If the digital certificate is encrypted then this is the password used to decrypt your digital certificate. The password is the same password you used to create the certificate request with your 3rd party certificate authority.

10. Click the **Verify** button to verify that Cerberus FTP Server can read the certificate and private key. If there are no errors then the certificate is valid and can be used by Cerberus.

11. Click **Ok** to close the Server Manager. If no certificate was previously being used or the certificate file path changed then Cerberus will configure itself immediately to use the new certificate.

CLUSTERING

FAILOVER CLUSTERING AND LOAD BALANCING

Cerberus FTP Server does not natively support clustering. However, using Active Directory or LDAP authentication, and a hardware or software load balancer (such as Microsoft NLB), you can achieve simple load balancing and failover with Active Directory or LDAP authenticated accounts.

To achieve Active Directory or LDAP-based load balancing, each Cerberus FTP Server machine is configured to point to the same AD or LDAP database, and requests can be load balanced to any of the available servers in that fashion. Many of our customers use such an arrangement for achieving simple failover and load balancing support.

Cerberus FTP Server Professional and Enterprise editions can now be configured to automatically synchronize all user accounts and settings to one or more other Cerberus servers. This new capability allows native Cerberus accounts, as well as customizations to Active Directory and LDAP authentication, to be easily synchronized across several Cerberus instances. Combining the new synchronization manager with shared storage between the Cerberus FTP Server machines allows for multiple active backup and failover servers.

LOAD BALANCING EXCEPTIONS

HTTP/S web client traffic cannot be load balanced using a simple connection balancer. The HTTP/S session database is local to each Cerberus machine, and any load balancer will have to ensure that all of the connections coming from a single IP are routed to the same Cerberus machine. We are working on a solution that will bring full clustering support to Cerberus FTP Server in the near future.

Web Services control

The Cerberus FTP Server Graphical User Interface (GUI) and underlying Windows Service use a distributed remote protocol called SOAP for communication. The primary function of the SOAP API is to allow communication between these two services. However, we've made the API available so that anyone can use it to programmatically control the server.

Please note: The SOAP API can change between releases. We do try to maintain backwards compatibility, but sometimes we have to make breaking changes in the interests of improving the API. Always refer to the actual WSDL included with the Cerberus distribution you are using for the latest definitions.

Available Features

Programmers can now access most of Cerberus FTP Server's common functions through a new Web Services interface. These services use SOAP 1.2 over HTTP or HTTPS and do not require a separate HTTP server. Cerberus FTP Server's implementation of Web Services includes a built-in, lightweight HTTP stack.

The following functionality is available through the Web Services API:

- Listing the current Cerberus FTP Server user and group accounts

- Adding new users or groups and modifying existing users and groups

- Deleting users or groups

- Retrieving user or group information

- Adding new virtual directories or modifying existing directories for a given user or group

- Deleting a virtual directory for a given user or group

- Getting the server's current started or stopped status

- Stopping or Starting the server

- Retrieving server statistics

- Retrieving and modifying interface details

- List, terminating, and blocking active connections

- Retrieving and saving configuration information

Refer to the included **Cerberus.wsdl** file for specifics on the Web Services interface to these functions. You can view an example Cerberus.wsdl online here.

EXAMPLE SOAP APPLICATIONS

We have two example applications available that use the SOAP API. There is an example .NET project available here:

.NET SOAP Example Application

A newer, simpler WCF-based client application is available for download here:

WCF SOAP Client Example

ACCESS URL

Make sure you enable SOAP access from the Remote settings page on the Server Manager. You can access the SOAP service WSDL on your local machine at the URL http://localhost:10001/wsdl/Cerberus.wsdl.

Make sure you have **Enable Web Administration** selected to view the actual WSDL. If Web Administration is not enabled you will still be able to use the WSDL to develop SOAP services but you won't be able to use the built-in web server to view the WSDL using the URL link. The WSDL is located in the installation directory where Cerberus is installed.

SECURITY CONSIDERATIONS

By default, Cerberus FTP Server's Web Services access is turned off. Before allowing Web Services access to Cerberus FTP Server, you should be well aware of the security implication that this entails. While it is the user's responsibility to be knowledgeable of Web Services and the risks associated with using them, here are some reminders:

- Make sure the port you are running the Web service on is properly locked down. If you are only using Web Services to communicate between programs on the same machine, the port Cerberus is running the Web Services on shouldn't be accessible from outside of the local machine.

- When using Web Services, remember that anyone with access to the port that the Web Services is running on can send service requests to Cerberus FTP Server. This can represent a serious security risk. Make sure you set a strong Remote access password.

- HTTP, the backbone of Cerberus FTP Server's Web Services, transmits information as unencrypted text. Anything you send over HTTP has the potential to be intercepted and read. Cerberus also has the option of using SSL/TLS support for Web Services over HTTPS. Using HTTPS instead of HTTP significantly increased the security of any data transmitted.

Cerberus FTP Server uses the gSOAP toolkit to implement Web Services. You can find out more about gSOAP at the gSOAP home page.

FTP COMMANDS SUPPORTED

The following FTP commands are supported by Cerberus FTP Server:

- ABOR
- ACCT
- ADAT
- ALLO
- APPE
- AUTH
- CCC
- CDUP
- CLNT
- CSID
- CWD
- DELE
- EPSV
- EPRT
- FEAT
- HASH
- HELP
- LANG
- LIST
- MDTM
- MFMT
- MFCT
- MKD
- MODE
- MLSD
- MLST
- MLSD
- NLST
- NOOP
- OPTS
- P@SV
- PASS
- PASV
- PBSZ
- PWD
- PORT
- PROT
- QUIT
- REIN
- RETR
- REST
- RMD
- RMDA
- RNFR
- RNFT
- SITE
- SIZE
- STOR
- STOU
- STRU
- SYST
- TYPE
- USER
- XCRC
- XCUP
- XPWD
- XMD5
- XMKD
- XSHA1
- XSHA256
- XSHA512
- XRMD