

# VSGate 4 Administrator's Guide

## Description

[Overview](#)

[How It Works](#)

## Installation

[Installing on Windows](#)

[Installing on UNIX](#)

[What's Next](#)

## Basic Administration

[Authentication \(Local, GAP, RADIUS, etc\)](#)

[Managing Server Certificates](#)

[Configuring Session Requirements](#)

[Generating Reports](#)

## Advanced Administration

[Command Line and Startup Options](#)

[Server Script](#)

[Configuring Filters and Versions](#)

[Configuring SMART Tunnels](#)

[Configuring User Resources](#)

[Configuring WINS and Sync Services](#)

[System Configuration File \(vsgate.cfg\)](#)

[Tunnel Configurations and Port Numbers](#)

## File Formats

[Log File Format](#)

[Configuration Files](#)

## Troubleshooting

[Troubleshooting](#)

[Security Considerations](#)

[Remote VPN Performance](#)

## About InfoExpress

[Company Background](#)

Contact Information

Acknowledgements

## Overview

VTCP/Secure extends the corporate network to remote users over untrusted networks like the Internet. VTCP/Secure does this by creating a remote Virtual Private Network (VPN) which transparently encrypts and validates all data. The Bad Guys on the untrusted network cannot eavesdrop, modify, or intercept data exchanged with the corporate network.

The software combines encryption, strong authentication, and authorization to protect against potential intruders and eavesdroppers. Public key and symmetric key encryption transparently encrypt all communications over the untrusted network, without modifying applications. Strong authentication ensures that only legitimate users can access the secure network. Filters can be defined and assigned by administrators to control the resources available to each user.

SMART Tunnels offer greater ease of use and better performance when using VTCP/Secure for remote access over the Internet. The remote Virtual Private Network is created only when communications to the secure network is required. Only data exchanged with the corporate network is routed through the VPN, keeping Internet traffic on the Internet.

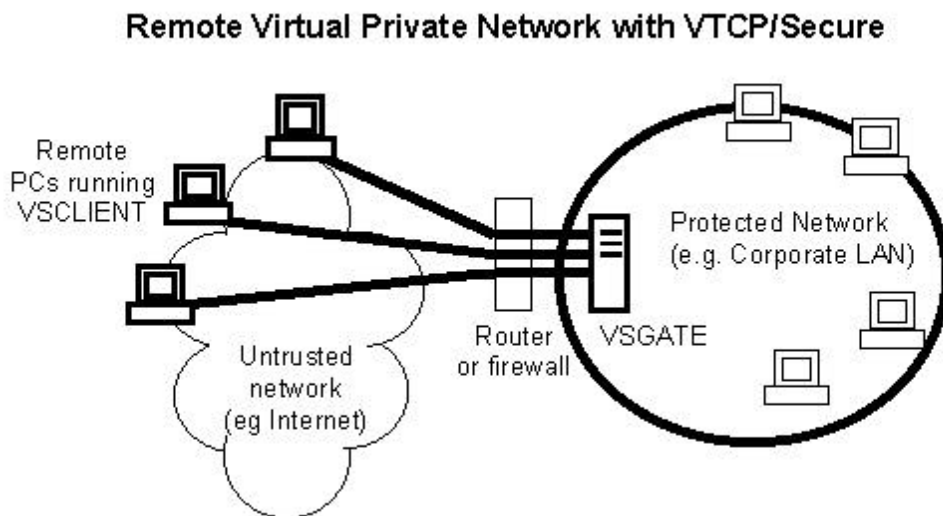
VTCP/Secure incorporates a number of standards based security and management features. The encryption, authentication and key exchange algorithms include DES 40, DES, Triple DES, and Diffie-Hellman.

The ability to scale is provided through redundancy, load balancing, and interfaces to authentication servers. VSGate provides interfaces to popular authentication, authorization, and accounting servers. VTCP/Secure also supports load balancing, redundancy, and replication in multi-gateway configurations.

## How It Works

VTCP/Secure consists of VSGate, VSClient, and VSAAdmin software. The VSGate software relays data between the remote PCs on the untrusted network and computers on the secure network. VSAAdmin is the management program for VSGate that lets administrators configure the options on VSGate.

VSClient runs on the user's remote PC. This program intercepts the application data and ensures that data exchanged over the untrusted network is encrypted and authenticated.



In the diagram above, the remote PCs are running VSClient which intercepts and redirects network data to the VSGate server. After VSGate executes the network operations on the protected network, data is returned to VSClient and subsequently to the applications.

All data transmitted over the untrusted network is encrypted and authenticated. When a session first starts up, VSClient establishes a TCP/IP connection to VSGate, then uses a key exchange protocol to mutually agree on a session key. The key exchange protocol is secure against active and passive cryptographic attacks.

To authenticate the user, VSGate requests credentials from the user such as a one time password or a response to a challenge. After the user has been authenticated, VSClient and VSGate will automatically route, encrypt, authenticate, and authorize data exchanged between them.

## Installing on Windows

### For Windows Servers...

If you are running VSGate for Windows, the computer running the VSGate software should meet the requirements listed below.

- The operating system must be Windows NT Server 4.0.
- The WINS Client on the NT server must be disabled and the server must not be a member of an NT domain.
- TCP/IP networking and DNS services must be working properly.
- To support assigned IP addresses and retroactive binds, a pool of consecutive IP addresses must be added to the NIC. Refer to your Windows documentation for details on adding multiple IP addresses to a single NIC.
- Server should be in a physically secure location.

### Installing the Software

The following procedure configures VSGate for Windows with one user.

- Obtain the VSGate for Windows software.
- Run the self-extracting VSGate installation software and specify a secure installation directory if possible.
- Follow the on-screen directions.
- If you have installed VSGate on Windows NT as a service, use the Services tool to verify that "VTCP/SECURE MULTIPLEXOR" has been installed as an NT service.

### Post Installation Steps for WINS Support

The installation software provides the option of supporting Windows Name Services (WINS) for remote clients. WINS is used to provide Windows file sharing and print services. Note that WINS is not the same as Domain Name Services which is used to resolve host names for TCP/IP networking.

If you will be using Windows file sharing or print services without WINS, you will need to configure an LMHOSTS file for each remote PC that will be running VSClient. Instructions for configuring LMHOSTS is provided by Microsoft as part of their Windows documentation.

If you require WINS support for the remote PCs, the computer running VSGate must not be running Microsoft WINS Client services because it conflicts with services provided by VSGate. Note that disabling the Microsoft WINS Client means that the Windows NT server cannot be used to share files or perform authentication services other than through Netbeui.

To disable Microsoft WINS services on the VSGate computer, follow the steps below. These instructions apply to Windows NT 4.0 server. If you are using Windows NT 3.51, please refer to your NT 3.51 documentation. If you are using Windows 95, WINS support for remote clients is not supported.

- Go to the "Control Panel -> Network -> Bindings" control.
- Show the bindings for "all protocols".
- Select "WINS Client (TCP/IP)" binding.

- Click the Disable button.

### Configuring Local Passwords

This section describes how to configure a sample account using the built-in one time password authentication service. Refer to the section [Configuring Authentication and Authorization](#) if you wish to interface to an authentication server (e.g. RADIUS).

- Run the VSAdmin program.
- Select **Manage Logins and Passwords**.
- Select **Show/Add/Edit Users**.
- Add a new user by typing the new user's name. Answer y to confirm the new user.
- Enter the new user's password. A good way to select a password is to use nonsense phrase of a few words which is lower case, easy to remember, and hard to guess (e.g. "chair glowed sparrows").
- Specify the maximum logins allowed between password changes. 9999 is a good number for "permanent" passwords.
- Select the Access Filters that should be assigned to this user. In the future, you may wish to create your own Access Filter definitions, but for now select the netops filter to allow unrestricted access to the network.
- Hit Enter and save the new user. If the previous steps have been performed properly, the new user should be listed.

At this point, you may wish to launch the VSGate server and read the [What's Next](#) section.

## Installing on UNIX

### Server Requirements

If you are running VSGate for a UNIX platform, the computer running the VSGate software should meet the requirements listed below.

- Operating System: Red Hat Linux 4.2, 5.x, Solaris 2.5, 2.6, HPUX, and AIX.
- Programs providing NETBIOS and WINS services must be disabled, such as Samba.
- To support assigned IP addresses and retroactive binds, a pool of consecutive IP addresses must be added to the NIC. Refer to your UNIX documentation for details on adding multiple IP addresses to a single NIC.
- Server should be in a physically secure location.
- TCP/IP networking and DNS services must be working properly.
- A name for the host. Verify by typing: `hostname`

If the `hostname` command does not return anything or displays an error, you need to give a host name to this computer. You can set the host name by adding a line like the second one below to the file: `/etc/hosts`

```
127.0.0.1      localhost
204.0.0.100   gate.acme.com
```

### Preparing the Host

The following installation procedure configures VSGate for UNIX with one user.

- Log in as root and create an account called `vtcpuser`. Although the account can have any name, calling it `vtcpuser` makes it easier to follow these instructions. Be sure to create a home directory for `vtcpuser`.
- Make the account's home directory private by typing the line below. Although VSGate never stores user passwords, it does contain sensitive information such as the private and secret keys for authentication servers, the server certificates, and synchronization services.

```
chmod 700 ~vtcpuser
```

- To take full advantage of the Retroactive bind feature on VSCClient, the UNIX host should run with `-o` option and must use assigned IP addresses for each user session.

### Installing the Software

- Download or transfer the appropriate version of the VSGate installation software to your system in a directory that can be accessible from the newly created `vtcpuser` account. Permissions for the VSGate installation file should also be modified to be readable by the `vtcpuser` account.
- **LOG IN AS VTCPUSER NOW!**
- Copy the VSGate installation file to the `vtcpuser` home directory and unpack it with the commands below.

- ```
cd  
cp <VSGATE_installation_file> vsgate.tar  
tar -xvf vsgate.tar
```
- Verify that the software has been properly installed by running the VSAdmin program (e.g. `./vsadmin`). If you see a warning message at the top of the screen, reinstall the software by carefully following the instructions on this page. Do not skip any of the steps, such as logging in as `vtcpuser`. If the problem persists, [contact InfoExpress](#) for assistance.

## Running VSGate

- **LOG IN AS ROOT NOW!**
- VSGate can run as a daemon that listens for incoming VSClient sessions on the port number that you specify. For each VSClient session, VSGate will create a new instance that assumes the identity of `vtcpuser`.

VSGate does not know the name of the `vtcpuser` account, so you must specify it as part of the command line. Because VSGate changes its identity “on the fly”, it must be running as root when invoked as a daemon.

VSGate can be run in the background or in the foreground.

### a. Running VSGate in the Background

Running VSGate in the background is the traditional way of running daemons on UNIX systems. This is recommended if VSGate will be invoked from a startup script when the system first boots or from the command line. The `-o` option causes bind operations to be performed as root. The syntax for running VSGate in this mode is:

```
vsgate -o -d vtcpuser [port]
```

The `vtcpuser` parameter should be the account that VSGate was installed in. It is very important that this be correct or VSGate will be unable to locate its configuration files. The `[port]` specifies which TCP port number will be used for incoming VSClient connections. If this is not specified, VSGate will use port 11160 by default.

### b. Running VSGate in the Foreground

VSGate can also run as a foreground daemon. This is the recommended way of running VSGate if invoked from `inittab`, which restarts VSGate if it stops running for any reason. The syntax for running VSGate in this mode is:

```
vsgate -o -df vtcpuser [port]
```

The arguments for this mode are the same as those for running VSGate in the background with the exception of the `-df` that is used instead of the `-d`. Refer to your UNIX documentation for more information on invoking programs from the `inittab` file.

- Verify that the software has been successfully installed by typing the line below.

```
telnet localhost 11160
```

If VSGate is running properly, you should see a message similar to the one below. If you do not see a prompt similar to the one below, reinstall the software or [contact InfoExpress](#) for assistance. Exit by typing “quit” followed by `<Ctrl-J>`. Sometimes nothing will be displayed on the screen while typing quit depending on which telnet program is being used.

```
+ONLINE SESSION/1.2/1 des passive
```



If the above command worked, you may wish to add the appropriate commands to your UNIX startup files. Otherwise, VSGate must be manually restarted whenever the server reboots.

### Configuring VSGate Services

VSGate can support Windows Name Services (WINS) for remote clients, synchronization services with other gateways, and load balancing services. For more information on configuring these services, refer to the section on [Configuring VSGate Services](#).

### Configuring Local Passwords

This section describes how to configure a sample account using the built-in one time password authentication service. Refer to the section [Configuring Authentication and Authorization](#) if you wish to interface to an authentication server (e.g. RADIUS).

- Run the VSAdmin program.
- Select **Manage Logins and Passwords**.
- Select **Show/Add/Edit Users**.
- Add a new user by typing the new user's name. Answer y to confirm the new user.
- Enter the new user's password. A good way to select a password is to use nonsense phrase of a few words which is lower case, easy to remember, and hard to guess (e.g. "chair glowed sparrows").
- Specify the maximum logins allowed between password changes. 9999 is a good number for "permanent" passwords.
- Select the Access Filters that should be assigned to this user. In the future, you may wish to create your own Access Filter definitions, but for now select the netops filter to allow unrestricted access to the network.
- Hit Enter and save the new user. If the previous steps have been performed properly, the new user should be listed.

At this point, you may wish to launch the VSGate server and read the [What's Next](#) section.

## What's Next?

Congratulations! You have just installed the VSGate server. There are several possible “next steps” depending on what your needs are. The options most likely to be of interest are listed below.

- Configure the routers or firewalls to allow access to the VSGate server. The VSClient software needs to connect to VSGate directly or indirectly over a **TCP** connection
- Install VSClient and connect to VSGate. Make sure you have started the VSGate service or application if you are running VSGate for Windows. You will need to know the address and **port number** of the VSGate server to establish a connection from VSClient. Later on, you may wish to create scripts for users to log in automatically.
- The Server Installation instructions configure “passive” privacy and perfect forward secrecy (PFS). Stronger security can be configured with minimal effort. The sections on **Configuring Authentication and Authorization**, **Session Requirements**, **Server Certificates**, and **Access Filters** will be of special interest to security administrators.
- Configure SMART Tunnels so users only tunnel the data which needs to go to the secure network. Information on configuring SMART Tunnels can be found in the section **Configuring SMART Tunnels**.
- Configure redundancy, replication, load balancing, or WINS services for remote clients. Refer to the section on **Configuring VSGate Services** for details on setting up these services.

## Authentication (Local, GAP, RADIUS, etc)

Users logging into VSGate can be authenticated through the local authentication service or through a separate authentication server. To interface to an authentication server, one of the authentication protocols shown below must be selected and configured.

### Selecting Authentication Mode

The authentication mode can be specified from the **SELECT AUTHENTICATION SERVICE** from VSAdmin. This command lets you specify local authentication or authentication through a protocol which interfaces to your authentication server.

Regardless of the authentication service which has been selected, users can log in only when authentication is enabled. When authentication is disabled, remote users will not be able to log in. However, users who are already logged in are unaffected until they need to authenticate again.

### Authentication Services and Protocols

VSGate supports the following authentication services and protocols. In general, smaller sites and pilot deployments may wish to use the local authentication service built into VSGate. Larger sites will generally prefer to use one of the protocols below to manage remote user authentication through a third party authentication server.

Local Authentication

RADIUS Authentication

TACACS+ Authentication

GAP Authentication (LDAP, PKI, etc)

## Local Authentication

The local authentication service built into VSGate supports one time passwords. The local authentication service does not store the user's secret password anywhere on VSGate or on VSClient.

### Configuration

This section describes how to add users to the local authentication server.

Using the VSAdmin program:

- Select **LOCAL** as the **AUTHENTICATION SERVICE**.
- Select **MANAGE LOCAL PASSWORDS**.
- Use the **SHOW/ADD/EDIT USERS** command to add new users or change the settings for existing users. This command lets you modify the **local password settings**.
- Use the **DELETE USERS** command to delete local accounts.
- Use the **DO/DON'T ALLOW USERS TO CHANGE PASSWORDS** commands to allow or prevent users from changing their own passwords. More information on this is provided below.
- Use the **NUMBER OF LOGINS BETWEEN ... CHANGES** command to specify how many times users are allowed to log in between password changes. This command is applicable only if users are allowed to change their own passwords.

### Allowing Users to Change Their Passwords

The local authentication service lets users change their own passwords. Users are also prompted to change their passwords when their current passwords are close to expiration.

To change the password, a user enters the word "changepass" when prompted for the user name or the password. VSGate then prompts the user for the current and desired password.

This method of changing passwords is supported only when using the local authentication built into VSGate. Using other authentication services may require different methods for changing user passwords.

## Local Password Settings

Local authentication parameters include the following:

- User name.
- Secret password used to generate one time passwords.
- Number of times the password may be used before expiration.
- The names of the Access Filters accessible by the user.

## RADIUS Authentication

VSGate can use the RADIUS protocol to authenticate users and authorize access to resources. The RADIUS protocol is supported by many third party authentication servers and is an Internet standard RFC.

To use RADIUS, you must configure VSGate and the RADIUS server. This section will discuss how to do both.

### Configuring VSGate

To configure VSGate to use RADIUS to authenticate and authorize users, run the VSAdmin program and follow the steps below.

- Select **RADIUS** as the **AUTHENTICATION SERVICE**.
- Select **MANAGE RADIUS CONFIGURATION**.
- Select **SHOW/ADD/EDIT RADIUS SERVER** to show the RADIUS servers. Edit the settings to reflect the RADIUS server parameters.
- Token users: See the RADIUS Configuration Tips below.

### Configuring the RADIUS Server

VSGate sends authentication, authorization, and accounting messages to the RADIUS server. To recognize this information, the RADIUS server must recognize VSGate and use the same encryption key. The attributes below will need to be defined on the RADIUS server with the exception of the Filter-Id attribute which is probably already present.

The parameters below on the RADIUS server must be compatible with those on VSGate. Refer to your RADIUS server documentation on how to set these values.

#### System Settings

NAS ID of VSGate  
Encryption Key

#### User Authentication

User Name  
Password

#### Attributes Assigned to Users

##### **Framed-IP-Address (Attribute 8)**

This attribute assigns a specific IP address to the user. This attribute is useful if a user must run server services on the remote PC that require well known ports, or if users must have their own IP address for any other reason. Note that the IP address must be available on the host that VSGate is running on. This is usually accomplished by assigning multiple IP addresses to the same network adapter on the host.

The "wildcard" IP address 255.255.255.254 indicates that VSGate should select any available IP address. The wildcard IP address range available to VSGate is set from the session settings screen in VSAdmin.

The value of this attribute is usually entered as a dotted decimal address. For example, the following attribute-value pair will force VSGate to assign the IP address 192.123.123.5 to a user:

Framed-IP-Address: 192.123.123.5

A range of IP addresses can be specified by specifying the Framed-IP-Address attribute twice, where the second attribute-value denotes the higher end of the range, inclusively. Wildcards are not permitted when specifying a range.

Frame-IP-Address: 192.123.123.1

Frame-IP-Address: 192.123.123.10

### Filter-Id (Attribute 11)

This attribute specifies the filter groups assigned to the user. The groups must match the Group Filters defined on VSGate. Note that the functionality of this attribute overlaps with the VSGATE-Filter described below.

Multiple groups can be assigned to a user by delimiting each group with a comma. For example, the groups netops and webservice can be assigned to a user as follows:

Filter-Id: netops,webservice

### VSGATE-Filter (Attribute 224)

This attribute specifies a filter entry associated with the user. Filter entries defined on the RADIUS server specify the resources accessible to a user. The format is the same as filter entries in Global Filters and Group Filters defined on VSGate, except that a group name is not required. For more information on filter entries and their syntax, refer to the section, **MANAGING FILTERS**.

More than one VSGATE-Filter attribute can be defined, each containing a separate filter entry. A set of filters that would restrict SMTP to host 201.1.2.3 (TCP port 25) while permitting unrestricted POP mail (TCP port 110) is shown below.

VSGATE-Filter: allow 201.1.2.3/32 tcp 25

VSGATE-Filter: allow any tcp 110

### VSGATE-Share (Attribute 225)

This attribute specifies the Windows shares that should be downloaded to VSClient to automatically map drives and printers. If this feature is enabled on VSClient, the remote PC will automatically map the specified drives and shares when the user logs in.

Multiple instances of this attribute may be defined, each specifying a different share. For example, to map several drives and printers from the Windows shares on NTSERVER, the following instances of the VSGATE-Share attribute could be defined.

VSGATE-Share: \\NTSERVER\\DRIVEC

VSGATE-Share: \\NTSERVER\\DRIVED

VSGATE-Share: \\NTSERVER\\LASERPRN

## RADIUS Configuration Tips

- The Filter-Id and the VSGATE-Filter provides authorization at different places. Although the Filter-Id specifies the groups that are associated with the user, the filter entries are configured on VSGate. In contrast, the VSGATE-Filter attribute defines filter entries on the RADIUS server itself so no filter entries or groups need to be created on VSGate.

- If both the Filter-Id and the VSGATE-Filter are used, VSGate authorizes access to resources by checking the VSGATE-Filter entries before checking the Filter-Id entries. If both filter methods only contain “allow” entries, the order is not important.
- A different prompt can be configured for the user name or password by editing the authconf file. Search for the “auth-raduserprompt” and “auth-radpassprompt” attributes and their description in the authconf file.
- Users requiring WINS, NT Domain logons, and NT file sharing will require access to UDP ports 137 and 138 on the local machine, and TCP on port 139 to the file server. Filter entries for these services can be added to the Global or Group Filters on VSGate, or to the RADIUS server under the VSGATE-Filter attribute.
- VSGate supports the vendor specific attribute to avoid conflicts with other RADIUS clients. If this feature is supported on your RADIUS server, you may wish to embed the attributes in the vendor specific attribute using InfoExpress's Vendor ID, 2939. Note: Even if this is used, the vendor specific attributes should not overlap with non-vendor specific attributes.
- VSGate looks up the attribute numbers for VSGATE-Filter and VSGATE-Share in the file, `vtcpconf/authconf`. The attribute numbers can be changed from their defaults (shown below) by changing the settings in this file.

```
auth-radfilterattr=224  
auth-radshareattr=225
```



## TACACS+ Authentication

VSGate can use the TACACS+ protocol to authenticate users. The TACACS+ protocol is a widely supported by many third party authentication servers and is a de-facto standard for authenticating users.

Setting up TACACS+ requires configuring both VSGate and the TACACS+ authentication server. This section will discuss how to do both.

### Configuring VSGate

This section describes how to configure VSGate to interface to an authentication server which supports the TACACS+ protocol.

Using the VSAdmin program:

- Select **TACACS+** as the **AUTHENTICATION SERVICE**.
- Select **MANAGE TACACS+ CONFIGURATION**.
- Select **SHOW/ADD/EDIT TACACS+ SERVER** to show the TACACS+ servers and to edit the settings to interface with the server. Usually this consists of adding the address of the TACACS+ server and its password. Leave the password blank if there is none.
- Select **DELETE TACACS+ SERVER** to erase a TACACS+ server entry.

### Configuring the TACACS+ Server

VSGate sends authentication, authorization, and accounting messages to the TACACS+ server. To recognize this information, the TACACS+ server must use the same TACACS+ password, recognize VSGate as an authentication client, and must authorize the appropriate users and services.

The settings below must be configured on the TACACS+ server. Refer to your TACACS+ server documentation on how to set these values. Note: Double quotes shown below are for clarity only and are not part of the attribute values.

#### TACACS+ Server Settings

TACACS+ client host name  
TACACS+ password

#### User Authentication

User name  
User password

#### Attributes Assigned to Users

##### Mandatory Attributes

The following attributes must be set before VSGate will allow the remote user to access the protected network. To change which attributes are mandatory ("service" and "protocol" by default), read the section below on Configuration Tips.

```
service=ppp
protocol=ip
```

## Address

This attribute assigns a specific IP address to the user. This attribute is useful if a user must run server services on the remote PC that require well known ports, or if users must have their own IP address for any other reason. Note that the IP address must be available on the host that VSGate is running on. This is usually accomplished by assigning multiple IP addresses to the same network adapter on the host.

The “wildcard” IP address 255.255.255.254 indicates that VSGate should select any available IP address. The wildcard IP address range is specified by the session-addrange attribute in the authconf configuration file. This file is described in the [System Configuration Files](#) section of this document.

The TACACS+ attribute associated with the Address is “addr” by default. For example, the attribute-value pair below will force VSGate to assign the IP address 192.123.123.5 to a user.

```
addr=192.123.123.5
```

A range of IP addresses can be defined by using two IP addresses for the addr attribute. The second address denotes the higher end of the range, inclusively. The example below assigns an IP address in the range 192.123.123.1 to 192.123.123.10, inclusive. Wildcards are not permitted when specifying a range.

```
addr=192.123.123.1 192.123.123.10
```

## Filter Group

This attribute specifies the filter groups assigned to the user. The groups must match the Group Filters defined on VSGate. Note that this attribute overlaps with the functionality provided by the Filter Entry described below.

The TACACS+ attribute associated with the Filter Group is “iacl” by default. Multiple groups can be assigned to a user by delimiting each group with a comma. For example, the groups “netops” and “webservice” can be assigned to a user as follows:

```
iacl=netops,webservice
```

## Filter Entries

These attributes specify one or more filter entries associated with the user. Filter entries define the resources accessible to a user. The format is the same as the filter entries used in Group Filters and Global Filters on VSGate, except that a group name is not required. For more information on filter entries and their syntax, refer to the section, [MANAGING FILTERS](#).

The TACACS+ attributes associated with the Filter Entries are “iacl#1” through “iacl#30” by default. More than one Filter Entry can be defined, each containing a separate filter entry. A set of filters that would restrict SMTP to host 201.1.2.3 (TCP port 25) while permitting unrestricted POP mail (TCP port 110) is shown below.

```
iacl#1=allow 201.1.2.3/32 tcp 25
iacl#2=allow any tcp 110
```

## Share Attribute

This attribute (“cmd-arg” by default) specifies the Windows shares that should be downloaded to VSClient to automatically map drives and printers. If this feature is enabled on VSClient, the remote PC will automatically map the specified drives and shares when the user logs in.

The TACACS+ attribute associated with the Share Attribute is "cmd-arg" by default. Multiple shares can be specified for download by delimiting each share with a comma. For example, to map a drive called DRVC and a printer called PCL from the Windows server called "NTSRV", the following attribute-value would be used:

```
cmd-arg=\\NTSRV\DRVC,\\NTSRV\PCL
```

### TACACS+ Configuration Tips

- A different prompt can be configured for the user name or password by editing the authconf file. Search for the "auth-tacuserprompt" and "auth-tacpassprompt" attributes and their description in the authconf file.
- Users requiring WINS, NT Domain logons, and NT file sharing will require access to UDP ports 137 and 138 on the local machine, and TCP on port 139 to the file server. Filter entries for these services can be added to the Global or Group Filters on VSGate, or to the RADIUS server under the VSGATE-Filter attribute.
- VSGate looks up the attribute names and values to send to the TACACS+ server from the file: `vtcpconf/authconf`. The attributes sent to the TACACS+ authentication server can be changed by setting the attributes in the authconf file.

```
auth-tacauthval1="service=ppp"  
auth-tacauthval2="protocol=ip"  
auth-tacshareattr="cmd-arg"  
auth-tacfilterattr="iacl"
```

The auth-tacauthval1 and auth-tacauthval2 settings define the mandatory TACACS+ attribute-value pairs that are required for authorization. The auth-tacshareattr setting defines TACACS+ attribute associated with the Share Attribute.

The auth-tacfilterattr setting defines the TACACS+ attribute associated with Filter Groups and Filter Entries. Note that Filter rules are distinguished from the Filter Groups by the hash symbol (#) used when defining Filter rules.

## GAP Authentication (PKI and LDAP)

VSGate can authenticate users and authorize access to resources to LDAP servers and PKI systems through a Generic Authentication Protocol (GAP) server provided by InfoExpress. GAP provides facilities similar to RADIUS except that it is designed to support public key systems.

GAP requires that you configure both VSGate and the GAP Server. This section will discuss how to configure VSGate and provides an overview on configuring the GAP Server.

### Configuring VSGate

To configure VSGate to use GAP, run the VSAdmin program and follow the steps below.

- Select **GAP** as the **AUTHENTICATION SERVICE**.
- Select **MANAGE GAP CONFIGURATION**.
- Select **SHOW/ADD/EDIT GAP SERVER** to show the GAP servers. Edit the settings to reflect the GAP server parameters. Note that the GAP configuration on VSGate does not specify what authentication servers (e.g. Entrust) are supported. The GAP client (VSGate) relies on the GAP server to transparently handle the details.

### Configuring the GAP Server

VSGate receives authentication and authorization information from the GAP server. The GAP server automatically maps VSGate authentication and authorization requests to a the native format understood by third party directory and authentication services such as LDAP and PKI. This means that each time the GAP server is upgraded, VSGate can support new PKI and directory services without modifying its configuration or software.

The GAP server must use the same encryption key as VSGate. Refer to the GAP Server Admin Guide on how to configure the GAP server and which directory servers and authentication protocols are currently supported. By default, the GAP server listens on port 11157 which is compatible with other VSGate services. This allows the GAP server to reside on the same machine as VSGate. This is not a requirement, however, and may not necessarily be the optimal configuration.

### Gap Server Attributes

The return attributes from the GAP server determine how VSGate behaves. This behavior is modeled after other authentication protocols like RADIUS and TACACS+, where each attribute consists of a number and its value.

The GAP server's configuration file specifies which attributes should be returned to VSGate. This file specifies which LDAP directories and fields contain the attributes to be returned to VSGate based on the user's identity. VSGate supports the following attributes, which are specified as strings:

#### Attribute 8: Assigned IP Address

This attribute assigns that the user should be assigned an IP address. If the IP address is set to 255.255.255.254, VSGate should select any available IP address, where the IP address range is set from the session settings screen in VSAdmin. This is the most common use of this attribute when used.

If the IP address is not the special IP address above, VSGate attempts to assign the specified IP address to the user. If two IP addresses are provided and separated with a space, VSGate will try to select an IP address within the given range. Because these modes are less flexible, they are not commonly used.

The value of this attribute is entered as a dotted decimal address. For example, the following value will force VSGate to assign a unique IP address to the user from the range defined by VSAdmin.

```
255.255.255.254
```

### Attribute 11: Filter Group

This attribute specifies one or more filter group(s) assigned to the user. The groups must match the Group Filters defined on VSGate. Note that the functionality of this attribute overlaps with the VSGATE-Filter described below.

Multiple groups can be assigned to a user by delimiting each group with a comma. For example, the groups netops and webservice can be assigned to a user as follows:

```
netops,webservice
```

### Attribute 224: Access Filter List

This attribute specifies one or more filter entries associated with the user. Filter entries defined on the RADIUS server specify the resources accessible to a user. The format is the same as filter entries in Global Filters and Group Filters defined on VSGate, except that the group name is not used. For more information on filter entries and their syntax, refer to the section, **MANAGING FILTERS**.

Each filter entry can occupy a single line. The following example shows a pair of filters that restricts SMTP to host 201.1.2.3 (TCP port 25) while permitting unrestricted POP mail (TCP port 110). These are separated by a new line.

```
allow 201.1.2.3/32 tcp 25
allow any tcp 110
```

### Attribute 225: Mount Share

This attribute specifies the Windows shares that should be downloaded to VSClient to automatically map drives and printers. If this feature is enabled on VSClient, the remote PC will automatically map the specified drives and shares when the user logs in.

Each share directive is specified on a separate line. For example, to map several drives and printers from the Windows shares on NTSERVER, the following value could be assigned to attribute 225. Each entry is separated by a new line.

```
\\NTSERVER\\DRIVEC
\\NTSERVER\\DRIVED
\\NTSERVER\\LASERPRN
```

## DSS Authentication

VSGate can use the DSS protocol to authenticate users. The DSS protocol is supported by the DSS authentication server from Axent. DSS authentication requires configuring both VSGate and the DSS server.

### Configuring VSGate

This section describes how to configure VSGate to interface to a DSS server.

Using the VSAdmin program:

- Select **DSS** as the **AUTHENTICATION SERVICE**.
- Select **MANAGE DSS CONFIGURATION**.
- Select **SHOW/ADD/EDIT DSS SERVER** to show the DSS servers and to edit the settings to interface with the server. Usually this consists of adding the address of the DSS server and configuring the various agent parameters.
- Select **DELETE DSS SERVER** to erase a DSS server entry.

### Configuring the DSS Server

VSGate sends authentication messages to the DSS server. To recognize this information, the DSS server must use the same agent password, recognize VSGate as an authentication client, and must authorize the appropriate users and services. A separate application note is available from Axent describing how to best configure the DSS to work with VSGate.

## Configuring Session Requirements

### Bypassing User Authentication

The **SELECT LOGIN REQUIREMENTS** menu lets you bypass user authentication if your application requires encryption but not end user authentication.

- **REQUIRE USER LOGINS** - Users must authenticate themselves before they are allowed to access the system. If the USER command is not in the script, the login will be terminated.
- **DO NOT REQUIRE USER LOGINS** - If the USER command is not in the client login script, the user will be granted default privileges into the protected network.

When user logins are not required, it is especially advisable to distribute server certificates rather than using the built in ones.

### Selecting the Encryption Algorithm

The **SELECT LOGIN REQUIREMENTS** menu lets you specify the type of encryption and key exchange algorithms which are required for users to log in. The algorithms selected ends up being the more secure of the algorithms requested between VSClient and VSGate.

The encryption algorithm commands available from VSAdmin are described below.

- **REQUIRE AT LEAST DES40** - Users can log in using any DES algorithm.
- **REQUIRE AT LEAST DES** - Users can log in only if using DES or better.
- **REQUIRE TRIPLE DES** - Users can log in only if using Triple DES with 3 keys (US only)

### Specifying the Default IP Address

- The **SELECT LOGIN REQUIREMENTS** menu lets you specify the default IP address in the event that none are assigned to the user. This address will be used for all users logged into the gateway if the assigned IP address option is not set below. Using a single IP address means that users will not be able to run server applications that listen on specific ports. This limitation exists because all users would attempt to use the same IP address to listen on the same port number on the gateway.

### Address Range Options

- The **SELECT LOGIN REQUIREMENTS** menu lets you specify address ranges that users should be assigned from, in the event that they are assigned an IP address. **All IP addresses in the range must be manually configured using the operating system's native method for creating multiple virtual IP addresses for a NIC.** On Windows NT, multiple IP addresses are assigned to a NIC from the Network Control Panel. On UNIX systems, multiple IP addresses are usually configured using the ifconfig command.

### Selecting the Session Privacy Algorithm

The **SELECT LOGIN REQUIREMENTS** menu lets you specify the key exchange algorithms to use to ensure session privacy. Note that active and perfect privacy options require a Default Certificate and its associated keyfile. Refer to the section on [Managing Server Certificates](#) for more information.

Passive privacy uses private keys which are generated "on the fly" from internal parameters so no server certificate is required unless the key length needs to be changed. Passive privacy is selected from **REQUIRE PASSIVE OR BETTER**.

Active privacy offers better security by using a server certificate to protect against man-in-the-middle and other active attacks. Active privacy is selected from **REQUIRE ACTIVE OR BETTER**.

Perfect privacy combines passive and active privacy to provide Perfect Forward Secrecy (PFS) and protection against active attacks. PFS is the ability for past sessions to remain secret even if the private keys are disclosed in the future. Perfect privacy is selected from **REQUIRE PERFECT OR BETTER**.

The session privacy options configured on VSGate are negotiated with VSClient. The actual key exchange algorithm used will meet the security requirements of VSGate and VSClient. For example, if VSClient requires passive privacy (default) and VSGate requires active privacy, the key exchange protocol will be perfect privacy which utilizes both modes.

### Specifying Timeout Values

The **Idle Timeout** specifies how long a period of inactivity before logging the user off the system. If an idle timeout occurs, logging out will cause all of the user's current TCP/IP connections to be closed. This option can be used with the suspend timeout option.

The **Suspend Timeout** specifies how long a period of inactivity before suspending the current session. When the current session is suspended, the user cannot send data to the protected network. Data can still be received during a suspended session over an existing connection although no new connections can be created. A suspended session can be resumed by re-entering the user's password.

The **Key Update Interval** specifies the period between session key updates. A new session key will be created periodically at the specified interval to provide stronger security even when using smaller key lengths.



## Managing Server Certificates

### Overview

VTCP/Secure uses public key cryptography to create the session keys used to encrypt data sent over untrusted networks. For the best security, a server certificate and its corresponding keyfile should be used by VSGate and VSClient to mutually agree upon a session key.

The keyfile is a file which contains the public key parameters in the server certificate. The keyfile is usually distributed to users with the installation software or through an e-mail message. VSClient and VSGate also have built-in keyfiles which can provide passive privacy, they do not offer active attack protection provided by external keyfiles.

When using server certificates, VTCP/Secure can use passive, active, or perfect privacy. Active privacy protects against active attacks such as man-in-the-middle. Perfect privacy provides perfect forward secrecy (PFS) in addition to protection against active attacks. Passive privacy is recommended only in cases where active attacks like man-in-the-middle are not a consideration.

The following sections describe how to create and maintain server certificates.

### Creating a Server Certificate

VTCP/Secure uses server certificates to enhance the security of the system. The keyfile associated with a server certificate is used by VSClient to validate the identity of the server and to create a new encryption key for each session. All privacy options are supported when using server certificates, although Active and Perfect privacy are recommended (see section above).

There are two types of server certificates. One is used for the initial session and the other is used for subsequent sessions.

|                     |                                                                                                                                                                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default Certificate | The keyfile associated with a Default Certificate is used by VSClient when a user logs in the first time. When the Default Certificate changes, its corresponding keyfile which contains its parameters should be distributed with the installation software to future users. The Default Certificate is created from VSAdmin. |
| Active Certificate  | The Active Certificate is used for all sessions excluding the initial one. The Active Certificate can be updated periodically (e.g. once a week) for better security without requiring any special actions on the part of the users. The Active Certificate is created automatically when the system first starts up.          |

To create or to update a server certificate:

- Select **MANAGE SERVER CERTIFICATES** from the main screen followed by **CREATE CERTIFICATES**.
- Specify which type of certificate to create. When configuring a VSGate server for the first time, both certificates should be created for optimal security.
- If you wish to change the public key length or when configuring VSGate for the first time, enter a new public key length when prompted. This step can take several minutes to an hour or more depending on the new key length and the computer. Most workstations can finish this step in less than 10 minutes for a key length of 768 bits.
- If prompted for the private key length, hit **ENTER** to use the recommended value.
- Enter **y** to create or update the server certificates.
- If prompted to make a keyfile using the Default Certificate, enter **y** and then enter the name of the keyfile. The new keyfile should be distributed along with the installation software.

### Creating a Keyfile for Remote Users

After a default server certificate has been created, VSAdmin provides the option of creating a corresponding keyfile for distribution to new users.

You can also recreate a keyfile without creating a new default server certificate. To do this, select the **MAKE KEYFILE** option from the **MANAGE SERVER CERTIFICATES** screen. This will generate a keyfile corresponding to the existing default server certificate. This file must be distributed to new users to allow them to log in.

### Tips on Server Certificates and Keyfiles

- It is usually a good idea to update the Active Certificate periodically. Updating the Active Certificate requires no special actions from users because the changes are performed transparently.
- Each time the Default Certificate is changed, be sure to distribute the corresponding keyfile to potential users along with the software, passwords, etc. Although the keyfile does not contain secret information, limiting its distribution to legitimate users enhances security.

## Command Line and Startup Options

### Windows NT

On Windows, VSGate is installed as either a service or a standalone program whose default port number for the tunneling service is 11160. You can change the tunnel TCP port number by manually editing the `tcpport` field in the `vsgate.ini` file located in the `vtcpconf` directory. For the Windows version of VSGate, you must restart the service or application each time you reconfigure the gateway.

- |       |                                                                                                                                                                                                                                                                        |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23    | VSGate assumes that incoming connections are using the Telnet protocol. When an incoming connection is established, VSGate will initiate Telnet handshaking for incoming connections. Tunneling into port 23 does not require remote users to select a special script. |
| 443   | VSGate assumes that incoming connections will use SSL 3.0 encapsulation. This requires that remote users use one of the SSL scripts to connect to the gateway. Do not use the HTTPS script, which is for backwards compatibility with older gateways.                  |
| Other | VSGate assumes a direct TCP connection with no special requirements                                                                                                                                                                                                    |

### UNIX

On UNIX, VSGate is installed as a standalone daemon which uses the port number specified on the command line for the tunneling service. The daemon can be run in the foreground suitable for systems that support `inittab` or can be run in the background for systems that start VSGate from a script.

VSGate is usually started as root and must use the `-df` or `-d` option to specify which user directory contains the VSGate configuration files (see below). If no port number is specified in the `-df` or `-d` options, 11160 is used by default. To change the port number, you must kill the daemon and restart it, specifying a different port number on the command line. Run VSGate with the `-h` argument to view all available command line options.

Some of the most common command line options are described below.

- |                                 |                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-c &lt;n&gt;</code>       | Limits the number of incoming connections to <code>&lt;n&gt;</code> per second. New incoming connections above this value within a one second period will be dropped. If <code>&lt;n&gt;</code> is set to <code>-1</code> , all connections will be dropped. If <code>&lt;n&gt;</code> is 0 (default), the gateway does not limit the number of incoming connections. |
| <code>-o</code>                 | Lets remote users run applications that bind (listen) to ports below 1024. Some applications may require binding to ports below 1024. On UNIX systems, the <code>-o</code> option is required for such applications to run. When using this option, VSGate should be started from root.                                                                               |
| <code>-X</code>                 | Disable the WINS and Netbios Datagram proxies for this instance of VSGate. This option is useful when WINS is not required, or a different instance of VSGate is providing proxy services on the same system.                                                                                                                                                         |
| <code>-r &lt;port&gt;</code>    | Additional ports that VSGate should listen on for incoming connections. This option assumes that VSGate should listen on an incoming TCP connection. This option can be used more than once.                                                                                                                                                                          |
| <code>-rssl &lt;port&gt;</code> | Additional ports that VSGate should listen on using the SSL protocol. This option requires that the client connect to the gateway on the specified port using the SSL encapsulation option. VSGate can usually detect incoming connections that are using SSL encapsulation without the <code>ssl</code> option if the <b>server script</b> is enabled.               |

-rtelnet <port>

Additional ports that VSGate should listen on using the Telnet protocol. This option forces VSGate to negotiate Telnet options with the incoming connection to permit tunneling through telnet proxies.

-d <user> <port>

-df <user> <port>

Specifies which port VSGate should listen for incoming connections and what user it should assume when tunneling. VSGate also uses <user> to locate the configuration files which are placed in the <user> home directory. The `-df` option is used when running VSGate from `inittab`. The `-d` option is used when running VSGate from the command line or from an rc script. The `-df` option is used when running VSGate from `inittab`. Using either `-d` or `-df` option is mandatory when invoking VSGate.

## Managing Filters

VSGate authorizes data transfers and name services by using filters defined by the administrators. A filter consists of one or more entries that define the resources accessible through that filter. There are several types of filters:

- Global Filters configure the system and are not associated with individual users, although some global filters may affect all users. Global Filters are defined in the globconf file on VSGate.
- Group Filters specify resources available to specific users. Group filters are defined in the filtconf file on VSGate. Group filters are assigned to users from a field returned from the authentication server.
- Server Filters specify resources available to specific users. These are defined on the authentication servers that can pass back attributes to VSGate such as RADIUS or TACACS+.

Each type of filter contains one or more filter entries consisting of an action (typically allow or deny), and a set of conditions such as addresses (hosts, networks, and subnets), protocols (TCP and UDP), and port numbers. The “allow” filter entries permit execution of network operations, and “deny” filter entries prevent network operations from executing. An “ignore” filter entry causes the entry to be ignored and can be used as a placeholder.

A network operation requested by VSCient must match one of the “allow” filter entries before it can be executed. If a network operation does not match any “allow” filter entries or matches a “deny” filter entry first, the requested operation will be rejected and an error message will be displayed for the remote user.

Information on configuring the different types of filters is provided in the sections on [Editing Filters](#) and the [Format of Filter Entries](#).

Detailed information on each type of filter is provided in the table below. The order that a filter is processed is important because preceding filters take precedence over following ones. The table also lists filter types which indicates where the filter is actually defined (global filters are defined in the globconf file and group filter types are defined in the filtconf file).

| Order | Filter Name                    | Type   | Description                                                            |
|-------|--------------------------------|--------|------------------------------------------------------------------------|
| 1     | <a href="#">Source</a>         | Global | Filters by source address of remote computer.                          |
| 2     | <a href="#">Version</a>        | Global | Filters by version of client software.                                 |
| 3     | <a href="#">All</a>            | Global | Filters all users.                                                     |
| 4     | <a href="#">Implicit Group</a> | Group  | Special filter targeting single users (typically unused).              |
| 5     | <a href="#">Normal Group</a>   | Group  | Filters users assigned to the group.                                   |
| 6     | <a href="#">Server Filters</a> | Server | Filters returned from the authentication server for a particular user. |
| 7     | <a href="#">Default</a>        | Global | Used only if no Group Filters or Server Filters assigned to user.      |

## Editing Filters

### Editing Global and Group Filters

Global Filters and Group Filters are defined on the VSGate host. These filters are viewed or edited from the VSAdmin program using the operating system's default editor or an editor specified in by the user.

To edit the filter files, perform the following from VSAdmin:

- Select **MANAGE ACCESS FILTERS** from the main menu.
- Either:
  - a. Select **VIEW/EDIT GLOBAL FILTERS** to edit or view the Global Filters OR
  - b. Select **VIEW/EDIT GROUP FILTERS** to edit or view the Group Filters.
- At this point, you can select which editor to use:
  - o To view or edit the file using the default editor, type Enter,
  - o To specify a different editor, type the name of the editor you wish to use,
  - o To abort this command, type "n" .

### Editing Server Filters

Server Filter entries are defined on a third party authentication server. The actual mechanism for editing the filter entries depends on the authentication server although the format of the filter entry is the same regardless of the authentication protocol.

The section on [Server Filters](#) explains this in more detail.

### Viewing System and Group Filters

You can use VSAdmin to view locally defined Global Filters and Group Filters in an expanded format. The expanded format shows the numeric addresses and port numbers for the locally defined filters exactly as they will be read by VSGate. Note that this cannot be used for viewing Server Filters because those are stored on a remote authentication server.

Viewing filters in expanded format is useful for verifying that the filter file has been configured correctly. To show the locally defined filters in expanded format:

- Select **MANAGE ACCESS FILTERS**
- Select **SHOW FILTERS IN EXPANDED FORMAT**

### Uploading Filters

You can upload filter files to VSGate rather than editing them on the VSGate server itself. The locations of the filter files is given below.

UNIX (relative to the home directory of vtcpuser):

vtcpconf/filtconf  
vtcpconf/globconf

WINDOWS (relative to the directory that VSGate was installed in):

VTCPCONF/FILTCONF  
VTCPCONF/GLOBCONF

## Format of Filter Entries

### Format Overview

Global Filters, Group Filters, and Server Filters are comprised of one or more filter entries. Server filters are defined on the authentication server, whereas Global Filters and Group Filters are defined on the gateway. Each filter consists of one or more entries that specifies the filter's name (Global and Group Filters only), action, and resources that apply to the filter entry.

The name indicates which filter the entry belongs to. Because Server Filters are defined on the authentication server and are implicitly associated with a specific user, Server Filter entries do not have a name field.

The format of filter entries is described below. The format does not include the special **Version Filters**, which have a different format and are described in their own section.

```
filter-<name> <action> <crypt> <addr> <proto> <port>
```

A description of the fields in a filter entry is given below.

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>&lt;name&gt;</b>    | The name of the filter. This is always prefixed with the word "filter-" (Global and Group Filters only).                                                                                                                                                                                                                                                                                                                                                                         |
| <b>&lt;action&gt;</b>  | <p>The action to perform if a network request matches the entry. The keyword "ignore" causes this entry to be ignored. The keywords "allow" and "deny" indicate whether a matching network operation should be permitted or rejected.</p> <p>KEYWORDS: allow, deny, ignore</p>                                                                                                                                                                                                   |
| <b>&lt;crypt&gt;</b>   | <p>Specifies the required encryption algorithm associated with an entry. When &lt;action&gt; is allow, the encryption algorithm being used must be equal to or more secure than &lt;crypt&gt; for the entry to match. When &lt;action&gt; is deny, the encryption algorithm must be equal to or less secure than &lt;crypt&gt; for the entry to match. If this field is blank, encryption checking is not performed for this entry.</p> <p>KEYWORDS: des40, des, 3des, 3desk</p> |
| <b>&lt;address&gt;</b> | <p>Defines an address, network, or host name associated with an entry. An optional slash can be used to define a mask either in bits or in dotted decimal notation. The keyword "any" means all IP addresses.</p> <p>KEYWORDS: any, &lt;IP addr/mask&gt;, &lt;domain.name&gt;</p> <p>EXAMPLES: www.infoexpress.com<br/>198.151.234.0/24<br/>198.151.234.0/255.255.255.0<br/>any</p>                                                                                              |
| <b>&lt;proto&gt;</b>   | <p>The network protocol associated with an entry. If this field is blank, the protocol is assumed to be "any" which matches UDP or TCP requests.</p> <p>KEYWORDS: udp, tcp, any</p>                                                                                                                                                                                                                                                                                              |
| <b>&lt;port&gt;</b>    | <p>A list of port numbers, port ranges, and service names associated with an entry. Port number ranges are separated by the word "to" and the service name "any" means all ports are valid. The service name to port number mapping is handled through the getservbyname database call.</p> <p>KEYWORDS: any, to, &lt;portnum&gt;</p>                                                                                                                                            |

EXAMPLES: 80  
20 to 21  
telnet



Global Filters

Global Filters specify the system behavior of VTCP/Secure and are not assigned to individual users. Global Filters are checked prior to Group Filters and Server Filters. The behavior of Global Filters is pre-defined by VSGate and the filter definitions are stored in the file: `vtcpconf/globconf`. The Global Filters consist of the following:

- filter-SOURCE** If defined, this Global Filter enables source IP checking. This feature only allows remote PCs with IP addresses matching an “allow” entry in the SOURCE filter to log into VSGate. If this filter is not defined, the source checking feature is disabled. Note that this option only works for sessions whose connections to VSGate are not transferred through a circuit layer firewall or NAT device.
- filter-ALL** This Global Filter applies to network operations for all users. If the action of an ALL filter entry is “allow”, the network operation is performed if the entry matches the network operation. If the action of an ALL filter entry is “deny”, the network operation will fail. If there are no matches, the network operation is checked against either the Group Filters assigned to the user or the default filter if no Group Filters have been assigned.
- filter-DEFAULT** This Global Filter applies to users who have not been assigned at least one Group Filter. The DEFAULT filter is also a convenient way to define default behavior for all users without having to specifically assign a Group Filter to each them.
- filter-TUNNEL** This Global Filter is used with the SMART Tunnel setting on VSClient. Unlike other filters, the TUNNEL filter is used by VSClient and not VSGate. When VSClient logs in, the TUNNEL filter is downloaded to indicate which addresses and DNS names should be tunneled. Refer to the section on [Configuring SMART Tunnels](#) for more information on the TUNNEL filter.
- version-ALL** If defined, this Global Filter enables version checking of the client software distribution. The format of these filters differs from the other global filters and is described in the section describing [Version Filters](#).

Sample Global Filter File

Network operations requested by VSClient are first checked against the SOURCE filter (if defined) and the ALL filter. VSGate also checks the DEFAULT filter if no Group Filters or Server Filters were assigned to the user.

If client software distribution is “Venus”, no message is displayed. If the client software version is “Pluto”, users receive a nag message but are allowed to continue. Any client software distributions not matching “Pluto” or “Venus” are rejected with a warning.

If the network operation matches an “allow” entry, checking stops and the network operation is executed. If the network operation matches a “deny” entry, checking is stopped and the network operation fails with a message displayed to the remote user. If there are no matching allow or deny entries, the network operation is blocked and the user will see a rejection message.

The TUNNEL filter is special because it is downloaded to VSClient and not processed by VSGate. If VSClient is configured to use SMART tunnels, network operations destined for hosts on 222.1.100.0 and 222.1.101.0 or name resolutions with the suffix `acme.com` will be tunneled. Name resolutions containing periods that are not in `acme.com` will not be tunneled using the wildcard domain, `.anydomain`. More information on SMART Tunnels is available in the section [Configuring SMART Tunnels](#).

```
filter-ALL          ignore

version-ALL         allow    "**/**/**/**/**/Venus"
version-ALL         allow    "vcmac/**/**/**/**/Pluto"      "Upgrade to Mac Venus"
version-ALL         allow    "vcwin/**/**/**/**/Pluto"       "Upgrade to Win Venus"
version-ALL         deny     "**/**/**/**/**/**/**"          "Invalid client"
```

|                |       |                |
|----------------|-------|----------------|
| filter-DEFAULT | deny  | any            |
| filter-TUNNEL  | allow | acme.com       |
| filter-TUNNEL  | allow | 222.1.100.0/24 |
| filter-TUNNEL  | allow | 222.1.101.0/24 |
| filter-TUNNEL  | deny  | .anydomain     |

## Group Filters

### Group Filters

A Group Filter consists of one or more filter entries defined on VSGate. A Group Filter can have any name consisting of alphanumeric characters, hyphens (-), and underscore (\_) characters. The group name in a Group Filter definition must be preceded by the prefix, "filter-". Group Filter definitions are stored in the file: `vtcpconf/filtconf`

The following Group Filter allows POP mail retrieval from any host, but only permits SMTP mail to be sent out on the central mail server called mail.acme.com.

```
filter-mailservice  allow      mail.acme.com  tcp  25
filter-mailservice  allow      any             tcp  110
```

Groups are assigned to individual users to selectively provide access to resources. The assignment of groups to individual users is performed by the authentication service. Refer to the authentication service you are using for details on assigning groups to users.

### Implicit Group Filters

Each user is implicitly assigned to an implicit Group Filter called **user-*<username>***. This filter provides a way to assign filters to individual users without having to reconfigure the authentication server. The Implicit Group Filter is checked just before the regular Group Filters.

For example, the user called *smith* who needs to use telnet to host 123.123.123.123 would be given the following filter entry:

```
filter-user-smith allow 123.123.123.123/32 tcp 23
```

### Group Filters vs Server Filters

Some authentication services such as RADIUS and TACACS+ support Server Filters that are defined on the authentication server instead of on VSGate. If you are using such an authentication service, you can configure Server Filters on the authentication server instead of using Group Filters on VSGate. This has the advantage of centralizing the access control functions.

Another mode is to use Group Filters and Server Filters at the same time. Using both may be useful when there are multiple sites, each with their own VSGate that authenticates through a central authentication server. In this case, the system administrator at each site may wish to have the ability to define Group Filters which limit the resources available to users.

When a user is assigned to one or more groups and has Server Filters defined on the authentication server, the Group Filters associated with the user are checked after checking the Server Filters. Refer to the section that discusses the authentication service you are using for more information on defining Server Filters.

### Sample Group Filter File

The file below shows five Group Filters. The netops group provides access to any service as long as the encryption protocol is Triple DES. The engineering group requires Triple DES as well but are limited to web, FTP, telnet, and NNTP services. The mail group provides POP mail from anywhere, so long as outbound mail is sent through the mail server. The webservice group allows access to anywhere. The msnet group lets clients access the WINS and NETBIOS services.

```
filter-netops      allow 3desk  any          any    any
filter-mailservice allow      mail.acme.com tcp    25    # smtp
```

|                    |       |       |     |     |            |        |
|--------------------|-------|-------|-----|-----|------------|--------|
| filter-mailservice | allow |       | any | tcp | 110        | # pop3 |
| filter-webservice  | allow |       | any | tcp | 80         | # Web  |
| filter-engineering | allow | 3desk | any | tcp | 80         | # Web  |
| filter-engineering | allow | 3desk | any | tcp | 20 21      | # FTP  |
| filter-engineering | allow | 3desk | any | any |            | telnet |
| filter-engineering | allow | 3desk | any | any |            | nntp   |
| filter-msnet       | allow |       | any | any | 137 to 139 |        |

## Server Filters

A Server Filter consists of one or more filter entries defined on an external authentication server. VSGate determines what the filter entries are for a Server Filter by using the authentication protocol (e.g. RADIUS) to extract the information from the authentication server.

Note that the actual configuration of the filter entries depends on the actual server. The format and the attributes to assign the filter entries to is explained in the section describing the specific authentication service interface.

The following protocols support the use of Server Filters that can be stored on the authentication server:

TACACS+

RADIUS

## Version Filters

VSGate can limit which versions of VSCClient are permitted to access the system. Beginning with version 3.3, VSCClient supports an optional script command, "version", that tells VSGate information about itself. Based on the version command information or the lack of it (if version was not provided), VSGate can log, reject, accept, or notify the user with a message.

The version information received by VSGate is processed through a list of version filter entries stored in the global configuration file. The syntax for a version filter entry is as follows:

```
version-ALL    <action>      "expression"    ["response"]
```

The <action> is either "allow" or "deny" to indicate whether VSCClient should continue or stop if "expression" matches the version of VSCClient. If "expression" matches, the specified action is performed immediately and the "response" message, if any, is displayed to the remote user. The format of an expression is 7 fields delimited with a forward slash (/). In other words, the expression format is "blah/blah/blah/blah/blah/blah/blah". The response field is the message displayed to the user in the event that this rule matches the version command.

There are several types of wildcards and special words. A field within expression that contains an asterisk (\*) is a wildcard field and will match anything. A backslash (\) at the end of the line means that the line is continued. If the "expression" consists of the single word, "none", the expression matches only VSClients that have either disabled the version command or do not support it.

**Note:** A client matching "none" will display the response message only if the user successfully logs in. If the action field is deny, the message will not be displayed and the user session will be silently rejected.

The seven fields are as follows:

|                             |                                           |
|-----------------------------|-------------------------------------------|
| Field 1, VSCClient name:    | One of: vcwin, vcmac, vcunix              |
| Field 2, VSCClient version: | Defined in vtcp.ini by the "version" tag. |
| Field 3, VSCClient info:    | Defined in vtcp.ini by the "verinfo" tag. |
| Field 4, Reserved:          | Set to *.                                 |
| Field 5, Operating System:  | One of: win31, win95, winnt, macos, unix  |
| Field 6, Reserved:          | Set to *.                                 |
| Field 7, User Defined:      | Defined in vtcp.ini by the "userdef" tag. |

The User Defined field is particularly useful because administrators can distribute custom versions of vtcp.ini that contain a distribution release number specifically for their deployment. This field can then be used to determine the remote user's exact configuration and to tailor response messages based on this information.

Additionally, use of the User Defined field can identify and prevent end users from installing the standard VSCClient distribution from InfoExpress, which may not have been approved by the company's IT organization.

A sample set of version rules for VSCClient is shown below. The rules below indicate that any Windows version of VSCClient that supports the version command should be transparently permitted access. Versions that do not use the version command will display a message after the user logs in requesting that the user upgrade their software. However, end users with the customer version REV1 should be denied access with a message that upgrading is imperative.

```
version-ALL allow "vcwin/**/*.*/**/*"

version-ALL allow "none" \
    "Please update your VSClient software to " \
    "version 3.3 or higher. Thank you."

Version-ALL deny  "**/**/*.*/**/REV1" \
    "Access denied for REV1 users. Please contact Acme " \
    "for further details."
```

Note the use of the backslash to make the entries more readable.

## Configuring SMART Tunnels

### Overview

SMART Tunnels only tunnel the data destined for the secure network and bypass the tunnel when sending data not meant for the secure network. With SMART tunnelling, data destined for the Internet sites or a customer's LAN would automatically bypass the tunnel. Routing information for SMART Tunnels can be dynamically updated based on DNS resolutions or automatically downloaded from VSGate each time the user logs in.

Another use of SMART Tunnels is to automatically log in when an application on the remote PC needs to communicate to the secure network. This behaviour occurs when VSCClient is configured to use SMART Tunnels and to create tunnels On-Demand.

The static routes on VSGate that are downloaded to VSCClient are defined in the Global Filter called TUNNEL. The TUNNEL filter is configured by the administrator using the VSAdmin program and lists allow (tunnel) and deny (don't tunnel) entries.

If you are unfamiliar with how filters work in VSGate or how to edit Global Filters, you may wish to read the section on [Managing Filters](#). However, be sure to note that the TUNNEL filter is downloaded to VSCClient and is not used by VSGate.

### SMART Tunnel Configuration on VSGate

SMART Tunnels are configured from VSAdmin by editing a special Global Filter called TUNNEL. The TUNNEL entries are automatically downloaded to VSCClient. The interpretation of the TUNNEL entries depends on the VSCClient settings under SMART tunneling.

- Create “allow” entries which contain the domain suffixes, networks, and addresses which require tunneling. For example, if the domain lookups which require tunneling are **acme.com** and **acme.net** and the networks are **253.53.53.0/255.255.255.0** and **253.53.54.0/255.255.255.0**, create the entries below. Note that address entries cannot be deny.

```
filter-TUNNEL    allow    acme.com
filter-TUNNEL    allow    acme.net
filter-TUNNEL    allow    253.53.53.0/255.255.255.0
filter-TUNNEL    allow    253.53.54.0/255.255.255.0
```

- Create “deny” entries for domain suffixes which should **not** be tunneled. To bypass tunnels for any name with a period in it which does not end with acme.com, you can add a single entry below with the wildcard name, “**.anydomain**” or “**anydomain**” where the former matches any name with a period in it.

```
filter-TUNNEL    deny     .anydomain
```

A list of “deny” domain suffix entries can also be used instead of a wildcard domain suffix. In general, this technique isn't recommended because new top level domains appear every day.

That's it! At this point, the VSGate configuration is set up to provide the necessary information for clients to determine what to do. The next section describes how VSCClient behaves depending on its tunnel settings.

### Using Static and Dynamic Routes on VSCClient

If the Use Static Routes is set, VSCClient will use the TUNNEL address entries to determine which IP addresses to tunnel. VSCClient uses the domain suffix entries to determine which name resolutions should be tunneled. Domain suffixes support both allow and deny entries. If a domain name does not match any of the allow or deny entries, it is resolved through the tunnel.



If the Use Dynamic Routes is set, VSCClient will also tunnel the IP addresses that have been returned from tunneled domain resolutions. This provides the ability to tunnel IP addresses without configuring static allow entries. This method requires that users enter domain names to access services instead of IP addresses.

Dynamic routes are most useful to handle the requirements of certain organizations. For example, dynamic routes are useful when organizations have internal network addresses that overlap with Internet or extranet addresses. Another case is when an organization has too many private subnets to be included in static route entries.

If VSCClient will only use dynamic routes and not static routes, the TUNNEL filter on VSGate only needs to include the domain name suffix entries, and can skip address entries. If address entries are not required, be sure to add a dummy IP address (e.g. 10.0.0.1/32) because VSCClient assumes everything should be tunneled if there are no address entries.

In the example below, traffic to any host with a suffix of acme.com or acme.net would automatically be tunneled, as well as to hosts whose names did not contain a period.

```
filter-TUNNEL    allow  acme.com
filter-TUNNEL    allow  acme.net
filter-TUNNEL    allow  10.0.0.1/255.255.255.255
filter-TUNNEL    deny   .anydomain
```

### TUNNEL Filter Differences

The TUNNEL filter behaves differently than other Global Filters because it is interpreted by VSCClient and not VSGate. These differences are described below.

- Only the **<name>**, **<action>**, and **<addr>** fields may be used in a TUNNEL filter entry. Each entry must have these three fields and no others.
- When a name is placed in the address field, it is used as a suffix to check for matching host name lookups that should or should not be tunneled. For example, placing **acme.com** in the address field would cause name resolutions for **mail.acme.com** and **acme.com** to match.
- Unlike other filters, names in the address field of a TUNNEL filter entry are not converted to IP addresses. Instead, they are used to compare name resolutions for fully qualified domain names.
- Unlike other filters, the TUNNEL filters checks for "allow" entries (do tunnel) before "deny" entries (don't tunnel) regardless of the order of the entries. Deny entries are only supported for domain suffixes, not IP addresses.

## Configuring User Resources

VSGate can be configured to automatically download resources. In particular, shares can be downloaded to VSClient to map them on the remote PC after the user has logged in. The downloaded shares will automatically be mapped if VSClient has not disabled this feature in the Map Shares option. This feature is also platform dependent so check the readme.txt file to see if it is supported for your version of VSClient.

To configure downloadable shares on VSGate, run VSAdmin and select **MANAGE FILTERS AND USER RESOURCES** → **VIEW/EDIT USER RESOURCES**. This will provide you with the option to edit the userconf on VSGate which contains the downloadable shares. The format of a share to download to the client is:

```
share-<group>=<sharename>
```

Where:

<group> is the group name which will cause users to receive the specified <sharename>.

For example, the following lines will cause VSGate to download the three shares below to VSClients whose users are assigned to the netops group.

```
share-netops=\\NTSERVER\\DRIVEC  
share-netops=\\NTSERVER\\DRIVED  
share-netops=\\NTSERVER\\LASERPRN
```

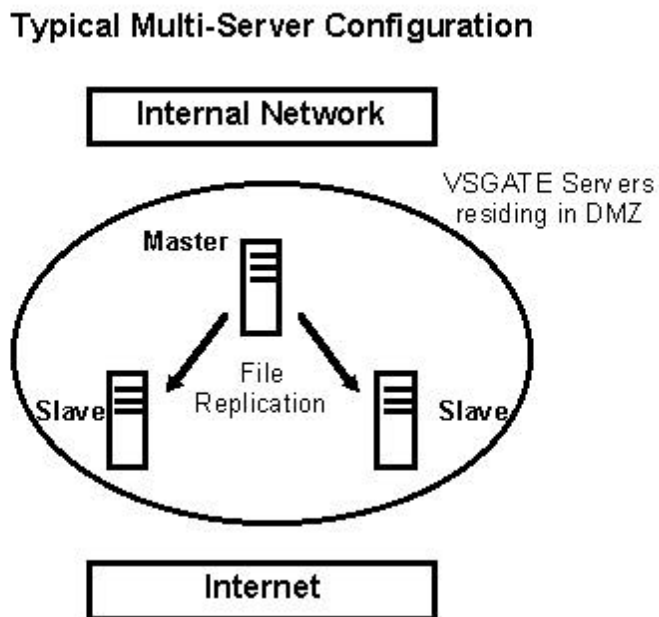
User resources can also be configured on certain authentication servers. Both **RADIUS** and **TACACS+** authentication servers are capable of defining downloadable shares on the servers.

## Configuring WINS and Sync Services

VSGate can provide WINS, load balancing, redundancy, and synchronization services. These services are configured in the muxconf file by setting the appropriate attribute values. The load balancing, redundancy, and synchronization services use files that are replicated from a single master server. The WINS service can be provided in a single gateway system as well as multi-VSGate systems.

### Multi-Gateway System

A diagram of a typical multi-gateway configuration is shown below.



In the above configuration, the master VSGate copies files to slave VSGates to keep all VSGate configurations in sync. This process only provides replication and synchronization services and no other relationship is implied between master and slave gateways. Whether a VSGate is a slave or a master, the tunneling services provided to VSClients for each type of gateway is identical.

### Load Balancing and Redundancy

There are several different ways to balance the load and provide redundancy among multiple gateways:

- Assigning multiple IP addresses to the same host name on the DNS server. VSCClient will take the list of IP addresses and attempt to connect to the first available one starting at the first IP address in the list. You can use the random option for the scripting command, connect, to select a random IP address to start with. If the first IP address fails, the client will attempt to connect to the other IP addresses in the list.
- Specifying two or more hosts or IP addresses with the script command, connect, on VSCClient. The client will attempt to connect in the order specified unless the random option is specified in which case a random one from the list will be used. If the first IP address selected fails, the client will attempt to connect to the other IP addresses in the list.

- Using the load balancing script in VSClient. This script can be configured to specify the host address of a VSGate that is providing the load balancing service that specifies which VSGate to log into. The load balancing service automatically determines which VSGate servers are running from the list specified in muxconf.

If the latter method is used, set the **mux-tunnel** attribute to specify the address and port number of each gateway providing tunneling services. Use a separate mux-tunnel attribute value line for each instance of VSGate that is providing services.

Example:        mux-tunnel=10.0.0.100  
                 mux-tunnel=10.0.0.101  
                 mux-tunnel=10.0.0.102

### File Synchronization

The muxconf file includes attributes specifying the addresses of the master gateway(s). Non-master gateways will read this file to determine which master gateway they should connect to in order to download the latest configuration files. In a sense, each gateway is a slave to the master because a slave gateway can appear to be the master for another gateway.

The relevant attributes in the muxconf file for configuring file synchronization are:

- sync-serverN**    Specifies the address of the gateway to synchronize files with. Files specified for replication in the muxconf file should not be directly modified on the slave gateway (with the possible exception of muxconf itself) because they will be overwritten. In most cases, even muxconf can be replicated and does not need to be edited on the slave gateway. N can be either 1 or 2.
- sync-key**        Specifies the encryption key that will be used to encrypt files transferred between gateways. This attribute and the sync-server attribute must both be set in order for the synchronization service to run. If both have been configured, the gateway will automatically transfer files from the specified sync server as required. Note that each gateway can function as a master and a slave, because all gateways configured with these two parameters also provide synchronization services to other gateways by default.
- sync-file**        Up to 128 files can be replicated from the gateway specified by the sync-server attribute. This service only replicates files in the vtcpconf directory - files in directories above and adjacent to vtcpconf will not be replicated. Note that replicated files will be deleted if the corresponding files in the sync-server are deleted.

Example:        The following example assumes that the master VSGate is at address 10.0.0.100 and most of the configuration files should be synchronized including the active, default, and past server certificates.

```
sync-server1=10.0.0.100
sync-server1=10.0.0.101
sync-key=HelloMyDarling
sync-file=muxconf
sync-file=vsgate.cfg
sync-file=authconf
sync-file=filtconf
sync-file=globconf
sync-file=keyconf
sync-file=keyconf.0
...
sync-file=keyconf.def
```

### WINS Service

WINS is Microsoft's Windows Name Service that provides name resolution for Windows file sharing and printer services. Without WINS, a LMHOSTS file must be used to convert Windows computer names to IP addresses. Note that WINS is unrelated to Internet domain name services (DNS) although both perform similar functions.

VSGate can provide Microsoft WINS services for remote PCs running VSClient. The attributes in the muxconf file are used to configure WINS services on VSGate. When VSGate is providing WINS services for the remote PCs, any WINS services configured on the local computer must be disabled.

If VSGate is installed on Windows NT, the native WINS services can be disabled as follows: Network Control Panel → Bindings → All Protocols → WINS Client → Disable. Note that disabling the WINS Client on Windows NT will also disable Windows NT's ability to share files and authenticate with other servers over NETBIOS. Services related to NETBEUI will continue to function, however.

If VSGate is installed on UNIX, you will need to kill any WINS server such as Samba in order to provide WINS services to VSClient. Note that you will need to find an alternate mechanism for providing WINS and NETBIOS file sharing services on your UNIX system if you disable it.

After disabling the native WINS services on the host, you can configure the following parameters in the muxconf file to enable the VSGate WINS services that will be provided to the remote PCs running VSClient. The VSGate parameters relating to WINS are listed below.

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>mux-winsproxy</code>   | Specifies the gateway providing the WINS proxy service. Usually, this is the local gateway so the attribute is set to the IP loopback address (127.0.0.1) by default. If the <code>mux-localaddr</code> attribute is set to something other than 0.0.0.0, change this attribute to forward WINS and NBDG packets (see below) to the specified local address.                                                                                         |
| <code>mux-localaddr</code>   | (UNIX only) Specifies the local machine's IP address that the gateway should use for its WINS and NBDG proxies. If this is set to anything besides 0.0.0.0, multiple instances of the WINS and NBDG proxy can run on the same machine if each instance uses a different IP address on the gateway. This is useful for running separate instances of the gateway to support multiple NT domains that do not have trust relationships with each other. |
| <code>mux-winsserverN</code> | Specifies the WINS server that the WINS proxy will send name resolution requests to. WINS requests from the client are forwarded to the WINS proxy which then forwards the requests to the WINS server. You can specify up to 2 WINS servers for redundancy. N should be either 1 or 2.                                                                                                                                                              |
| <code>mux-nbdgserverN</code> | Specifies the NT Domain Controller that the NBDG (NETBIOS datagram) proxy will send authentication requests to. NETBIOS datagrams are forwarded to the NBDG proxy which then forwards the requests to the NT Domain Controller. You can specify up to 2 NT Domain Controllers for redundancy. N should be either 1 or 2.                                                                                                                             |

**Example:** The following example assumes that there is a WINS server at address 10.0.0.101 and a backup WINS server at 10.0.0.102.

```
mux-winsproxy=127.0.0.1
mux-winsserver1=10.0.0.101
mux-winsserver2=10.0.0.102
```

## Server Script

VSGate supports a server script that is run prior to establishing an encrypted tunnel. The server script is stored in the file, `vtcpconf/server.scr`. Some of the potential uses for the server script include:

- Automatically detecting incoming tunnels that need to traverse content inspecting SSL proxies
- Automatically detecting incoming tunnels that need to traverse HTTP proxies
- Dividing remote users into different categories by distributing a different keyword to each type of user. The scripting language can then be used as a lightweight mechanism to enforce access to VSGate.
- Prevent hackers from seeing the `+ONLINE SESSION` prompt by creating a server script that does not start a login session until the VSClient script sends an administrator defined keyword to the server.

The standard server script adds a 5 second delay before starting VSGate. Beginning with version 4.2, VSGate can use this delay to automatically detect most inbound tunnels that are using true SSL encapsulation. Automatic detection preserves compatibility with older clients that cannot perform true SSL encapsulation. However, automatic detection may fail if the connection between the client and gateway takes longer than 5 seconds to transfer data.

VSGate can also be configured to require inbound SSL tunnels on a particular port. If all VSClient versions are 4.2 or higher, then users can select the script to use true SSL tunneling. Requiring an inbound SSL tunnel is also appropriate if older clients (< 4.2) will not be connecting to the gateway using the SSL port. If either requirement is met, the 5 second delay can be eliminated through one of the methods below.

- Configuring VSClient scripts to send `"direct"` after connecting to the gateway.
- Reducing the timeout value in `server.scr`.
- Deleting or renaming `server.scr`.

The best choice depends on the benefits of the server script for your network. If the benefits are not substantial, the most prudent choice would be to keep the script, but reduce its timeout to a short period. If the benefits are significant, you may wish to distribute VSClient scripts to users that reduce the delay by sending a `"direct"` command.

Details on the scripting language are documented in the `server.scr` file.

## Generating Reports

VTCP/Secure provides a report generator called VSMON. VSMON reads the log files in the vtcplog directory and generates reports based on these logs. To see a menu of reports which VSMON can generate, run vsmon from the command line in UNIX, or double click the VSMON Icon in Windows. If you are recording logs in yyyyymmdd format, vsmon should be run with the -ymd option.

Reports are provided for user activity, login activity, security alerts, and more. Information in reports contains statistics on traffic, hosts connected to, warnings, etc.

VSMON requires PERL 4.0 or higher on UNIX platforms. The Windows version of VSGate comes with a version of PERL.

## Tunnel Configurations and Ports

### Overview

VTCP/Secure uses TCP to tunnel data. VTCP/Secure can tunnel through multiple hops, NAT routers, circuit layer firewalls, and IP networks with overlapping IP address spaces.

The services and corresponding port numbers below are useful when configuring VSGate. The port numbers listed below are the default values for the services, although many can be changed by the network administrator.

| <u>VSGate Service</u>  | <u>Proto</u> | <u>Port</u> | <u>Type of Service</u> |
|------------------------|--------------|-------------|------------------------|
| Tunnel Service         | TCP          | 11160       | <b>External</b>        |
| Load Balancing         | TCP          | 11159       | <b>External*</b>       |
| Synchronization        | TCP          | 11158       | Internal*              |
| Generic Authentication | TCP          | 11157       | Internal*              |
| WINS Proxy             | UDP          | 137         | Application*           |
| Keep Alive             | UDP          | 11160       | Internal*              |

\* Optional - depends on configuration.

### External Services

External services are provided over the public network. The ports used external services must be externally accessible if their services are required. The tunnel service is mandatory because this provides the actual VPN tunneling, although the port number can be changed. The load balancing service is optional, and is required only when clients use the load balancing script instead of the “random” keyword in the connect command.

### Application Services

Application services support specific applications running on VSClient and should not be accessible from the public network. Access to application services should be controlled through the Global, Group, or Server Filters. The only application service is the WINS proxy which provides WINS and Netbios Datagram support for remote clients. These services should not be accessible from the public network.

### Internal Services

Internal services are used by VSGate only and are documented for completeness. Ports used by VSGate internal services should not be accessible from the public network.



## System Configuration File (vsgate.cfg)

Most sites can be configured without modifying the settings in the system configuration file. However, this file can be used to fine tune site configurations by redefining services, mapping port numbers, and modifying other settings related to VSGate.

The system configuration file can be edited from the main menu of VSAdmin by selecting **SYSTEM INFORMATION**.

**complevel** <level>

This command specifies the compression level which will be used by the server if the client requests compression. Setting the level to 0 disables data compression, setting to 1 minimizes CPU usage, and setting to 9 maximizes compression. By default, the compression level is set to 5 which is a reasonable tradeoff between CPU usage and effective throughput. Setting the compression level to higher levels increases the effective throughput and also increases the CPU usage.

In most cases, there is no benefit to setting the compression level to 0 even if the server is overloaded. Disabling compression causes data to be transmitted more slowly and increases the time to complete an arbitrary task. The increased time to complete the task will often increase the total CPU usage. If reducing CPU usage is important, try setting the compression level to 1.

Examples:

|             |                                                   |
|-------------|---------------------------------------------------|
| complevel 0 | Disable data compression                          |
| complevel 1 | Enable data compression, minimizing CPU usage.    |
| complevel 9 | Enable data compression with maximum performance. |

**dns** 0.0.0.0

**dns** <address>

*This command is only supported on VSGate for UNIX.*

The first form of this command forces name resolution to occur locally through the `gethostbyname()` and `gethostbyaddr()` call on the gateway, if no DNS servers are specified in the `muxconf` file. The normal mode of operation will forward DNS requests to the DNS servers defined on the system when there are no DNS servers specified in `muxconf`. This behavior is useful in networks where NIS or other name resolution services are being used and where a DNS server may not be available. This command has no effect if DNS servers are specified in `muxconf` because `muxconf` settings override this command.

The second form of this command specifies the address of the DNS server which should be used for name resolution. This command is usually used when the gateway software should use a set of DNS servers which are different from the ones configured for the host. If multiple DNS servers should be used for name resolution, this command can be repeated with the addresses of different DNS servers.

**escape** <escape list>

Some protocols which may be used to create an indirect connection between the VSClient and VSGate may not be able to pass certain characters. Characters which cannot be transmitted need to be "escaped" (not sent) by substituting them with an acceptable character sequence prior to transmission and restoring the escaped characters when received at the other end of the tunnel.

The escape command specifies which characters to escape. Characters in the escape list should be separated by a space and

can be specified as decimal numbers (0 to 255), hex numbers (0x0 to 0xff), or control characters by preceding the letter with a caret symbol (^A, ^B, ^C, etc). A range may be specified by using the word "to" between two escape characters (21 to 31).

Example: escape 0 to 31 127 to 159 255

**ignoresetbuf 1**

Setting this command will cause VSGate to always return success whenever an application running on the remote PC sets the send or receive buffer size, regardless of the parameters passed to this routine.

**interactive 1**

The interactive command adjusts the response time to the client. When interactive is set to 1, response time improves but CPU usage increases. Unless the server load is a significant factor, this command is recommended, especially if users are running terminal applications like Telnet or 3270 emulators.

**keyrecover <user> password <passwd>**  
**keyrecover <user> plain**

The keyrecover command will cause key recovery packets to be sent each time the session encryption key changes for the specified users. If this feature is enabled, the key recovery packets can be used to decrypt the encrypted data for the specified users if the password is known. By default, this feature is turned off because it weakens the security of the sessions that match the user name. It should only be enabled if a government agency serves the appropriate warrants to your organization.

Two types of key recovery packets are available. Using "password" embeds the current session key, which is itself encrypted with a cryptographic hash of the specified password. Using "plain" embeds an 8 character ASCII message which is itself encrypted with the session key.

The user name can either be a specific user, an asterisk (\*) to indicate that all user sessions should send key recovery packets. Contact InfoExpress for details regarding the KRP packet formats.

**log <user> [session] [warning] [connect] [data]**  
**log global ymd**  
**log syslog**

The first command indicates what information should be logged and during which session. If <user> is set to an asterisk (\*), the log applies to all users. If <user> is set to a user, the log only applies to the specified user. The options following the <user> field indicate the type of activity to log.

The second command forces all files to be logged into year/month/day file names. The ymd file naming convention creates log files that have a four digit year, two digit month, and two digit day as part of their names. The actual format is "vtcplog.YYYYMMDD" (e.g. vtcplog.19980704 for a log created on July 4, 1998). Note that the ymd command affects all logs.

The third command causes logs to be saved to the system logging service. On UNIX systems, this is the syslog facility and on NT, this is NT Event Manager. Enabling system logging does not disable logging to local files.

The session option logs the session start and end time, and the user's total traffic between VSGate and VSClient. The session option can be used to determine who is logged in. This option is enabled by default.

The warning option logs all warnings regardless of the level of logging. Without this option, warning messages are logged only when their corresponding level is logged (e.g. session warnings, connect warnings, data warnings).

The connect option records the IP addresses, ports, and traffic generated for each service accessed by the remote user. The connect option provides accounting information on a per connection basis. This option is disabled by default.

The data option records the IP address, port, and data size for each individual transmission sent and received during a session. This option can be used to track the user's activity in a high degree of detail. This option is disabled by default.

The example below logs the session, connect, and data activity for user smith. Only session and connect information is logged for other users.

Example:      `log smith session connect data`  
              `log * session connect`  
              `log global ymd`

### `logperiod <secs>`

This command specifies the interval between session statistics logs. The session statistics log shows how many bytes were sent and received by the user. By default, the session statistics log is written to the log file every 30 minutes (1800 seconds).

The session statistics log can also act as a "keep alive" by periodically updating the log file even if there is no other activity for the user which is being recorded. This may be helpful for tracking sessions which have logs which span more than one day, and therefore write log entries to more than one log file.

### `mapaddr <proto> <host> <port> <newhost> <newport>`

This command is used to redirect connections established by remote users to different hosts and services. Although this command can be used for many things, its primary uses include (1) creating virtual hosts and services and mapping them to physical hosts and services, (2) redirecting connections to a specific host to another one, (3) redirecting connections to a particular service to a different service.

For example, this command can create a virtual mail host called mail.vpn and direct connections to mail.vpn to a host on the LAN. Or this command can map connections to any SMTP host (port 25) to a specific SMTP server on the LAN.

**<proto>** indicates the protocol to redirect. This can be **udp**, **tcp**, or **all** (for TCP and UDP).

**<host>** can be a (1) virtual host name which is created on the fly (e.g. mail.vpn), (2) an IP address of a real host to redirect, or (3) a wildcard ("\*") which means that all hosts will be redirected. Note that this cannot be a real host name.

**<port>** can be (1) a port number to redirect, or (2) a wildcard ("\*") which means all ports should be redirected.

**<newhost>** can be (1) a new host to redirect the traffic to, or (2) a wildcard ("\*") which means that traffic should go to the host specified by the remote application.

**<newport>** can be (1) the new port number that traffic is redirected to or (2) a wildcard ("\*") which means that traffic will be sent to the port specified by the remote application.

Examples:

|                                                     |                                                       |
|-----------------------------------------------------|-------------------------------------------------------|
| <code>mapaddr tcp * 110 pop3.acme.com 110</code>    | Redirect POP3 traffic to pop3.acme.com                |
| <code>mapaddr tcp * 80 www.acme.com 80</code>       | Redirect Web traffic to www.acme.com                  |
| <code>mapaddr tcp mail.vpn * mail.acme.com *</code> | Create virtual host mail.vpn aliased to mail.acme.com |

### `mapport <reqport> <newports> [verbose|quiet|silent]`

This command maps requests to listen on a fixed port number to an alternate port number. Whenever a Windows application requests a listen on a port matching <reqport>, VSGate will allocate one of the <newports> instead. This lets multiple users simultaneously run applications such as FTP servers when logged into the same host.

Use of this command is not necessary in most cases because very few applications need to listen on a fixed port number. However, there are two situations which may warrant the use of this command:

- o Certain client applications may need to listen on a privileged port. These ports would normally be denied due to security reasons, but the `mapport` command lets VSGate know that it should transparently map requests for these ports to other non-privileged port numbers.
- o Certain client applications (such as X-Windows) issue a listen to a fixed port number which cannot be reconfigured. Without remapping, only a single such application could run at any time because each port only supports a single listen at a time. By remapping the requested port number to a range of port numbers, many such applications can be run at the same time.

The parameters passed to the `mapport` command are described below.

|                               |                                                                                                                                                                                                                             |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;reqport&gt;</code>  | Requested port from application. Can be specified as a decimal number (23), hex number (0x17), or service name.                                                                                                             |
| <code>&lt;newports&gt;</code> | List of ports to map the requested port to. Port numbers can be specified as a decimal number or hex number separated by spaces. A range of newports can be defined by using the keyword "to" between any two port numbers. |
| <code>[verbose/silent]</code> | Indicates how the user should be notified when a port has been remapped.<br>verbose      Display message.<br>silent      No indication.                                                                                     |

Example: `mapport 21 8000 to 8099 verbose`

Lets the remote user run an FTP server on their remote PC which accepts incoming connections on a port in the range 8000 to 8099.

**`nettimeout <seconds>`**

This command specifies how long a loss of contact with VSCClient is allowed before VSGate automatically disconnects the session. Prior to this timeout, VSCClient will automatically attempt to communicate with VSGate even if there is no user data. By default, the timeout period is 60 seconds.

**`randaddr 1`**

When addresses are assigned to users from the range specified in `vsadmin` under session settings, the addresses are assigned to users consecutively starting from the lowest address available. Whether an address is used is determined by performing a bind operation on an address by the gateway. When large number of concurrent users (> 1000 concurrent) are present on operating systems that perform binds slowly (Solaris 2.5 and 2.6), this can consume a significant amount of CPU. Setting this value to 1 performs binds against random address over the entire range of addresses assigned, resulting in far fewer binds to find an unused address.

**`tcpport <portnum>`**

*Note: This command is only supported on VSGate for Windows.*

When running as a standalone application, this command specifies the TCP port number that the VSGate server should listen for incoming connections on. This command is not used when running VSGate as a service.

**`udplimit <pktspersec>`**

This command specifies the maximum number of allowable UDP packets per second for a given socket. By default, VSGate does not limit the number of UDP packets per second.

**`userconf on/off`**

If the configuration file is located in the /etc directory, this option can enable or disable the ability to read additional configuration settings in a configuration file located in the vtcpuser home directory. This option is ignored if there is no configuration file in the /etc directory in the first place.

**wppcompat <level>**

This command indicates the level of compatibility that the gateway should provide to clients. Level 0 means that compatibility is restricted to 3.1 clients and higher. Level 1 provides compatibility with 3.0 and older clients in most situations while providing an extended number of sockets to newer clients. Level 2 provides the greatest compatibility with 3.0 and older clients, but supports much fewer concurrent sockets in newer clients.

## Log File Format

Log files record user activity for up to a week. Each log file consists of the log entries for users accessing the system in a single day. All log files are located in the directory, `~/vtcplog`. The log file without the suffix (`vtcplog`) is the current day's log file. Log files with suffixes (`vtcplog.0`, `vtcplog.1`, etc) correspond to the previous day's logs, the logs which are two days old, etc.

### Types of Log Entries

There are three types of entries written to the log file. Of the ones listed below, the data logs generate the most traffic if enabled, and the session logs generate the least amount of traffic.

**Session logs** These entries record session information. Session information consists of the user start and end times and session statistics logs which record the traffic generated by the user. The session statistics logs are written periodically based on the interval specified by the `logperiod` parameter in the system configuration file.

**Connect logs** These entries record connection information specifying when and what (IP address, port, etc) resources the user has connected to. The last entry for a network operation will log the number of bytes sent and received by the user for the connection.

**Data logs** These entries record the traffic generated from each transfer of data initiated by the client. This log generates a lot of information and is normally turned off.

### Format of Log Entries

Each of the above logs has a common format which resembles the log entries produced by Web servers. This format is described below.

**<ip addr> <session ID> <user> <date/time> "<log specific data>" [- <total bytes>]**

<ip addr> is the IP address of the remote client. If the connection to VSGate is coming through a firewall plug, the address is the firewall's IP address.

<session ID> is the session ID associated with the user. This value is guaranteed to be unique for the duration of the user's session. This value can be used in conjunction with the `event_id` field contained with the <log specific data> to create a statistically unique value for the session.

<user> is the user's login name and <date/time> are the time that this log was written.

<log specific data> contains details about the log. The <log specific data> field can be broken into several more fields. The first part of the text within the log specific data begins with "GET /vsgate?..." The portion after the ? character contains various attributes describing the log entry.

Some of the more important attribute and their values contained within the "<log specific data>" portion of the log entry are listed below.

- proto** This indicates the protocol associated with the log.  
Usually one of the following: `tcp`, `udp`, `session`, `server` (VSGate for Win32 only)
- event** This indicates the event which triggered the log.  
Usually one of the following: `start` and `end` (`proto=session`), `trace`, `warning`, `data`
- status** Displays human readable text explaining errors or warnings.

op      Specifies the network operation associated with the log. This entry is used when proto=tcp or proto=udp.

## Configuration Files

Most configuration files can be edited from the VSAdmin program. However, it may sometimes be convenient to modify certain files directly, such as the access filter files.

### Directory Tree

Note that all files are relative to the installation directory. The location of the installation directory depends on which version of VSGate is running. On UNIX, the installation directory is the home directory of the VSGate account (~vtcpuser). On Windows, the installation directory is the current directory when launching VSGate.

Because all configuration files used by VSGate reside in the installation directory, this directory should be inaccessible to non-administrative personnel. The private key used by server certificates and the TACACS+ password are both stored on the VSGate server, although end user passwords are not stored.

The directory structure is shown below. Asterisks (\*) indicate files that are often edited without using the VSAdmin program.

Installation directory:

|              |                                                                                  |
|--------------|----------------------------------------------------------------------------------|
| vtcpconf/    |                                                                                  |
| authconf     | Authentication service settings (local, TACACS+).                                |
| authconf.x   | VSGate reads this prior to authconf for the same attributes.                     |
| * filtconf   | Group Filter configuration file.                                                 |
| * globconf   | Global Filter configuration file.                                                |
| * muxconf    | Multi-server configuration file. Changes to this file require restarting VSGate. |
| * muxconf.x  | VSGate reads this prior to muxconf for the same attributes.                      |
| * userconf   | Resources to download and configure remote PC.                                   |
| keyconf      | Active public key parameter file downloaded to all active users.                 |
| keyconf.#    | Previously active public key parameter files (# is backup number).               |
| keyconf.def  | Default public key parameter file provided to new users as bootstrap.            |
| * server.scr | Server script that is run prior to session negotiation.                          |
| * vsgate.cfg | System configuration file (settings not related to security).                    |
| vsgate.ini   | Configuration file for Windows VSGate when run as a service.                     |
| vsgate.prd   | Product license file.                                                            |
| vtcpauth/    |                                                                                  |
| user1        | Local user authentication files. Ignored for TACACS+, etc.                       |
| user2        |                                                                                  |
| ...          |                                                                                  |
| vtcplog/     |                                                                                  |
| * vtcplog    | Log files for user logins. Format similar to HTTPD logs.                         |
| * vtcplog.0  |                                                                                  |
| * vtcplog.1  |                                                                                  |
| vtcpbin/     |                                                                                  |
| * scripts    | Report scripts and other utilities.                                              |



## **AUTHCONF**

This file contains settings for the authentication services and session security settings. These include authentication server passwords and login security requirements. The file itself contains comments describing its format.

## KEYCONF

### KEYCONF.#

### KEYCONF.DEF

These are the server certificates used to create secure sessions between VSCClient and VSGate. Keyconf contains the Active Certificate parameters, including the private key. The keyconf files with numeric extensions are older certificates which can still be used for sessions, but will also cause the active key file to be downloaded automatically. The keyconf.def file is the Default Certificate used for initial logins.

**All KEYCONF files in the vtcpconf directory must be kept secret!** These files should never leave the server or be revealed to others because they contain the private keys. Users requiring a keyfile for the server should be given the keyfile created through the VSAdmin program (see the section, [Managing Server Certificates](#)).

## **MUXCONF**

This file contains information on setting up services such as load balancing, replication, and WINS services. These services are useful for configuring multiple gateways, allowing clients to log into NT, and supporting minimal client configuration changes. The file itself contains comments describing its format.

## **USERCONF**

This file contains resources and configuration information which is downloaded to VSCClient. In particular, the shares that the remote PC should mount are stored in this file. The file itself contains comments describing its format.

## VSGATE.INI

This file contains settings for the Windows version of VSGate when installed as an NT service. The location of the file is:  
`vtcpconf/vsgate.ini`

Administrator configurable settings include:

`tcpport` Set this to specify the port that VSGate should listen for incoming connections.

`lifetime` Maximum number of seconds before restarting a new process.

`lifetime` Maximum number of sessions before restarting a new process.

## VSGATE.PRD

The product license file indicates that the current software has been registered or is under evaluation. Without a product license file, session times are limited to a relatively short period of time. The product license file must be copied to the installation directory as: `vtcpconf/vsgate.prd`

## Troubleshooting

If you encounter difficulties running VTCP/Secure, you may find some of the suggestions below useful in troubleshooting the problem. Some suggestions are also included for making the most of VTCP/Secure.

### **VSCient fails to connect to VSGate or immediately disconnects**

This is usually caused because there is no connectivity between VSCient and VSGate. The easiest way to determine whether the problem is due to lack of connectivity or improper client configuration is to run the telnet application at the remote PC to connect to the port number VSGate is listening on.

The telnet program that comes with Windows 95 and Windows NT can be used to verify this. Be sure to enter the port number or telnet will use port 23 instead of 11160 (or whatever other port VSGate is listening on).

```
telnet <ip address> <port>
(e.g. telnet 1.2.3.4 11160)
```

Lack of connectivity to VSGate may be due to a firewall or proxy sitting between the remote user and VSGate. If a remote user is connecting to VSGate from a remote site, be sure that the firewall or proxy server lets VSCient go through.

On UNIX systems, be sure that you've told VSGate which user's directory contains the configuration files. This is passed to VSGate in the `-d` command line option (e.g. `vsgate -o -d smith 11160`). If this information is not provided, VSGate will search for configuration files in the directory belonging to the user who started VSGate, which is often root. A sure sign that this has occurred is if you see "none none" when telnetting into the VSGate port 11160 (e.g. "+ONLINE SESSION... none none").

### **Session terminates after a short period of time**

One possibility is that VSCient is traversing one or more firewalls that are configured to only permit access to VSGate for a certain period of time. One of the firewalls may be configured to limit the time that a connection may be established. In such cases, the TCP connection will be terminated automatically, resulting in a terminated session. If this is the case, you will notice that the session always terminates after a fixed period of time after the session starts up.

Another possibility is that the connectivity between VSCient and VSGate is very poor. This is especially common when VSCient is communicating to a VSGate located in a different country. If there is a long period where connectivity is lost between VSGate and VSCient (e.g. > 60 seconds), the session will terminate automatically by default. This behavior ensures that connections will be terminated when a dial-up connection is dropped without having time to shut down the connection in an orderly fashion, but has the side effect of terminating sessions running over poor connections.

If you suspect a poor connection between VSGate and VSCient, you can manually disable or alter the network timeout on VSCient and VSGate. On VSGate, the network timeout is specified by the `nettimeout` attribute in the `vsgate.cfg` file. On VSCient, the network timeout value is specified by the `nettimeout` in the `vtcp.ini` file. An alternative to the default settings is to set `nettimeout` to 0 on VSGate to disable it and set `nettimeout` to 600 on VSCient to increase it to 10 minutes (600 seconds).

### **Application reports name resolution or DNS failure**

This occurs when domain name services have not been configured on VSGate. If the problem is because the DNS has not been configured on VSGate, configure name services on VSGate host.

### **Application unable to open a port**

The port number may need to be remapped to a different port number. See the `mapport` command in the system configuration (`vsgate.cfg`) file for more information.

### Log into VSGate Securely

Log into the VSGate system using VSClient to protect your telnet session.

If the problem you are seeing does not fit into one of the above categories, try the InfoExpress Web site or **contact** us with a description of the problem. Please include both the name and version of the following software: VSGate, VSClient, PC operating system, and server operating system.



## Security Considerations

This section discusses the general security considerations when using the remote VPN. Understanding these requirements is important in order to determine which security options to use when configuring VTCP/Secure.

### Types of Attack

**Passive Attack.** In a passive attack, the Bad Guy watches the data sent between the remote PC and the secure network, but makes no attempt to alter it.

**Active Attack.** In an active attack, the Bad Guy can insert, modify, and delete messages between the remote PC and the secure network. Hijacking connections, spoofing IP addresses, and man-in-the-middle are all examples of active attacks. This type of attack is technically more difficult than a passive attack, especially over a LAN.

**Insider Attack.** In an insider attack, the Bad Guy is an internal user or someone who has obtained authentication credentials from a legitimate user. One of the most common ways is through “social engineering” such as calling up the password administrator and pretending to be someone else, or by calling up the user and pretending to be an administrator.

**Denial of Service.** In this type of attack, the Bad Guy attempts to disrupt services. For example, the Bad Guy could flood the company's T-1 line to the Internet so that Internet connectivity is effectively lost.

### Basic Security

#### Authentication

Assessing the user's identity is the lynchpin of strong security. The user's identity is the key to granting access to the network, determining which resources can be accessed, and logging what actions have been performed. Because security hinges on the user's identity, authentication should be made as strong as possible without making the system too difficult for remote users.

The most common forms of authentication are passwords, hardware tokens, and software tokens. All of these mechanisms are cryptographically secure, provided that the authentication credentials are not divulged and the authentication hardware or software is not compromised.

In the real world, however, credentials can be divulged and the hardware can be compromised (stolen or borrowed). Passwords are most vulnerable to being given out. If the password is not given out, remote authentication is usually good even if the remote user's PC is stolen.

Hardware tokens are harder to compromise than passwords because both the token and the PIN/password are required to log in. If either component is missing, the system not allow a user to log in. Hardware tokens are not susceptible to password guessing attacks even if the token is stolen because the token locks up after a small number of invalid PINs have been entered.

Software tokens offer similar security as hardware tokens as long as the user's PC is not accessible to the bad guys. If the token database is “plaintext aware” (i.e. the validity of a PIN can be determined by examining the token database), the soft token is vulnerable to brute force attacks if the copy protection mechanism is broken and the remote user's PC is stolen.

#### Encryption

Encryption ensures that data is private only if the keys exchanged remain private. VTCP/Secure ensures the confidentiality of the keys by using public key algorithms to generate a private key that is only known to the gateway and the client. The private

information is never stored on the remote PC, but is stored on the gateway because the security there is likely to be much better.

The use of public key algorithms makes it possible to protect against passive and active attacks without storing any confidential information on the remote user's PC.

### **Data Integrity**

Information exchanged between the remote PC and the corporate network must be tamper proof. VTCP/Secure protects the integrity of all data by adding a message authentication code (MAC) which is like a strong checksum. To prevent replay, insertion, or deletion attacks, VTCP/Secure incorporates a sequentially incrementing initialization vector (IV).

### **Authorization**

Authorization, also known as access control, defines which resources are available to users. Administrators can use this capability to limit access to just the servers that are required, or to block access to servers that are not. For example, everyone on the network may require access to their e-mail server, but only people in sales need to have access to the sales tracking server.

## Remote VPN Performance

This section describes some of the techniques for improving end user performance and reducing unnecessary CPU usage on the gateway.

### Remote PC Configuration

Using data compression can improve performance for users with slower connections to the Internet. The effective throughput will depend on the type of data being sent and received. In general, data compression is recommended for users with sub-56 kbps connections. Users with higher speed connections will be unlikely to benefit from data compression as much and should disable data compression to reduce server load. Client scripts with and without compression are included with the standard client software.

### Server Configuration

The factors affecting the server load is the concurrent data throughput of all users, the encryption algorithm, and data compression. InfoExpress has a white paper describing the effects of encryption and throughput on server load that is available upon request. Gateway data compression can be disabled for all users by setting `complevel` in `vsgate.cfg`, although this is generally not recommended.

```
complevel 0
```

### Internet

When using the Internet, performance may exceed or drop below the performance of dedicated dial-up connections. File transfers may often run faster due to the superior data compression offered by the gateway, but highly interactive applications may run slower because of transit delays. This is sometimes offset by a user's higher speed connection to the Internet (e.g. cable modem) than would be available through a direct dial-up line.

The pairing between the remote ISP and the corporate ISP may also cause performance anomalies. Although an ISP can connect to some ISPs all of the time and an ISP can connect to all ISPs some of the time, no ISP always has good connectivity to all other ISPs. In general, if remote users are using the same ISP that is providing Internet access to the corporation, performance is likely to be better.

## Acknowledgements

### **VSGate 4 Administrator's Guide**

4.3r1

**Copyright © InfoExpress, Inc. 2000  
All rights reserved**

VTCP/Secure Software  
Copyright © InfoExpress, Inc., All rights reserved.

DES, Triple DES, and Diffie-Hellman Software  
Copyright © Eric Young, All rights reserved.

MD5 Message Digest Algorithm Software  
Copyright © RSA Data Security, Inc., All rights reserved.

Portions of the Software  
Copyright © Jean-loup Gailly and Mark Adler.

InfoExpress, VTCP/Secure, VSGate, VSCClient are trademarks of InfoExpress, Inc.  
Microsoft, Windows is a registered trademark of Microsoft Corporation.  
All other names are trademarks or registered trademarks of their respective companies.

## Company Background

InfoExpress was founded in 1993 to provide network connectivity and security software. For more information about the company, visit the InfoExpress Web site at: <http://www.infoexpress.com>

## Contact Information

|        |                                         |
|--------|-----------------------------------------|
| E-Mail | <code>info@infoexpress.com</code>       |
| Web    | <code>http://www.infoexpress.com</code> |