Cipafilter documentation

Last updated: 2015-11-19 (version 8.8.5)

Introduction

Cipafilter is a powerful routing platform capable of delivering an evolving tool set to protect your enterprise. Cipafilter's philosophy is to provide a cutting edge, well rounded, and aggressive network control solution to meet your current and future needs, to commit to keeping your platform up to date, and to provide support far above and beyond simple problem solving. We believe our philosophy is what sets our product firmly apart from our competitors. Our technicians will talk you through installation every step of the way. We will advise you as to which features are a good fit for your organization and help you implement them. We will protect your return on investment by assuring that you can always easily upgrade to our latest software, often at no cost. And of course, we will help you solve any problems you encounter with our product, even when it's caused by a third-party agent, software package, or network problem.

Cipafilter is a powerful solution for many of your networking problems:

- Our anti-virus systems will protect your network by scanning mail (SMTP, POP3, and IMAP) and Web (HTTP, HTTPS, and FTP) protocols.
- The content filter can block pornographic e-mail and Web browsing to provide a more professional environment for your students or employees.
- SSL decryption provides enhanced filtering capabilities for HTTPS-secured browsing.
- IPsec VPN tunneling can secure your data as it is transmitted between branch offices.
- Download redirection can save your workstations from spyware.
- Web-usage reporting, authentication, bandwidth reporting, and bandwidth control help you keep your employees on task and stop problems before they become serious.
- Anti-spam systems can save you hours a day.

And, of course, Cipafilter provides all of the back-end tools you need to keep your network running smoothly, such as DHCP, dynamic and static routing, firewall, SMTP, and whitelisting/blacklisting.

This reference will provide the information necessary for an experienced network engineer to configure and operate these services. However, this manual is not intended to replace our excellent phone support. We will help you use our product every step of the way. Many customers never even log on to their router. Please feel free to call us at **1-800-24DERBY** if you have any questions or need advice during the lifetime of your product. There is no need to try to install the device without our help if you don't feel comfortable with these instructions.

Interface conventions

Most page/section headers, as well as certain option labels, within the Web interface can be clicked to produce the relevant section of this manual.

Clicking a SAVE CHANGES button will cause the configuration options on any particular page to be saved but not activated. The SAVE AND APPLY button will cause the configuration options to be saved and activated. If there is no SAVE AND APPLY option on any given page, the router must be restarted to activate the associated changes.

Installation

In most cases you will want to consult with Cipafilter support to decide what way the router can best be installed to meet your needs. A full over-the-phone consultation during installation is included in the standard one-year maintenance and service agreement that comes with your router.

Cipafilter is usually installed one of four ways: as a replacement for your existing router, as a bridge spliced into the link between your existing router and your switch bank, as a second router connected to your existing router with a cross-over cable, or as a server on your network.

Of the four ways that Cipafilter is usually installed, installation as a bridge is the easiest. Installation as a replacement for your existing router is most recommended, however.

The Cipafilter ships with `10.1.2.3/8` assigned to its first Ethernet interface and `192.168.0.1/24` assigned to its second interface. If your network does not use `10.0.0.0/8` IP addresses, you can simply assign an IP like `10.1.2.1/8` (netmask `255.0.0.0`) to your workstation and plug the Cipafilter into your hub or switch.

If your network *does* use these IP addresses, plugging your Cipafilter in without taking the proper precautions could cause loss of Internet access or other problems. If you're not sure, please call us for assistance.

To begin the initial configuration, go to **https://10.1.2.3** in your Web browser. The default user name to log in is `root` and the default password is `derby`. The password can (and should) be changed on the **User Manager** page.

As a replacement for your router

Often you will find that the best way to install your Cipafilter is as a replacement for your current router. Cipafilter is a top-of-the-line router and implements virtually all of the functionality of other routing systems. In this case you would configure Cipafilter with the IP addresses, routes, and NAT settings currently on your router, and then replace your current router.

As a bridge

The easiest way to install your Cipafilter is as a bridge. Simply unplug your existing router from your switch bank, connect the Cipafilter to that port on your router via a cross-over cable, and then connect the cable that was plugged into your router to the other port on the Cipafilter. Lastly, enable the **Interface Bridging** option on the **IP Settings** page of the Web-based interface. Your network should now operate normally, and you can activate the Cipafilter's features one-by-one as you're ready to implement them.

Between your router and your network

If you do not wish to replace your current router, you can connect the Cipafilter and your existing router with a cross-over cable, and then connect the Cipafilter to the rest of your network. This essentially "splices" the Cipafilter into the cable running from your router to your network. To do this you must first pick a small subnet from a private range you aren't using. If you're using `10.0.0.0/8` you might pick `192.168.0.0/24`, for example. Assuming the default gateway you use for your clients is `10.0.0.1`, you would then:

1. Replace the `10.0.0.1/8` IP address on your main router with `192.168.0.1/24`.
2. Configure your Cipafilter with `192.168.0.2/24` on the outside interface and `10.0.0.1/8` on the inside interface.
3. Disconnect your main router from the hub, and run a cross-over cable from that Ethernet port on your main router to the outside Ethernet port on the Cipafilter.
4. Connect the inside Ethernet port on your Cipafilter to the port on your hub or switch where your router was connected.
5. Create a route for `10.0.0.1/8` with a gateway of `192.168.0.2` in your main router; alternatively, you can choose to activate NAT on the `10.0.0.0/8` subnet on the Cipafilter and not create a route. If you choose the NAT option be certain that the `192.168.0.2/24` interface is selected as the **Primary Internet Connection**.

As a server

Many features of the Cipafilter can be used simply by assigning the Cipafilter an IP address on your network and connecting it like any other server.

Disadvantages to this method include:

- Transparent proxy will be unavailable.
- POP3 virus scanning will be unavailable.
- The firewall function will be ineffective.
- You will often need to add firewall rules to your existing router to force users to use the proxy server in order to make Cipafilter effective at protecting your network.
- Bandwidth control will only be effective for Internet traffic working through the proxy server.
- The bandwidth-reporting feature may not be able to monitor all traffic on your network.

Running on a virtual machine

The Cipafilter firmware is also available as an image suitable for use with a virtual-machine solution. VMware is the most common and best-supported vendor, although many others have been used successfully. Any of the four previous methods may be combined with the virtual-machine option.

Note: Although bridging mode will work with a VM-based filter, this combination is somewhat more complex than others. Enabling bridging on an improperly configured virtual machine can cause serious network issues and even (in rare cases) hardware damage. It is recommended that only the most advanced administrators make use of this option.

IP Settings

Each interface has a primary IP address as well as one or more optional secondary IP addresses. An IP address of `0.0.0.0/0` indicates that the interface is to be left unconfigured. Secondary IP addresses can be assigned to an interface by clicking **ADD IP** near the bottom of each interface section.

IP addresses are specified using CIDR notation, which uses an address/subnet bits format. The subnet bits refer to the number of bits that are set in binary in the subnet mask from the left hand side. For example, a subnet mask of `255.0.0.0` in binary has 8 bits set from the left hand side. So, `10.0.0.25` with a subnet mask of `255.0.0.0` translates to `10.0.0.25/8` in CIDR notation. See the table in **Appendix I** for a list of common dot-notation subnet masks and their CIDR-notation equivalents.

Settings like the host name and time zone can also be changed on this page. If you are using the anti-spam features of Cipafilter, make sure that the host name and domain are correct. Ensure that the `hostname.domain` combination resolves to an IP address on the Cipafilter and that the IP address reverses properly back to `hostname.domain`. This is important because many mail servers will refuse to trade e-mail with your Cipafilter if this is not done.

Field descriptions

Logging Database Connect String

Cipafilter can use a remote filter as a logging database server for the purposes of scaling performance. This value is a PostgreSQL connect string that instructs the filter to log to a remote database server; it should be configured by tech support.

Enterprise Database Connect String

This PostgreSQL connect string describes how the filter connects to the Cipafilter Enterprise monitoring product (if it has been purchased by your organization); it should also be configured by tech support.

Primary and Secondary DNS

Enter external DNS servers here. We recommend leaving these blank to use the Cipafilter's internal DNS server.

Interface Bridging

Clicking this check box will cause the Cipafilter to become a bridge, bridging all of its Ethernet interfaces together into one logical interface. In this configuration the Cipafilter acts like a switch. Installing the Cipafilter this way may require the restarting of any Cisco routers on the subnet to clear their ARP caches. Any IP addresses you wish to configure on Cipafilter for management can be applied to any interface if this box is checked.

Note: The filter should always be rebooted after disabling bridging to allow for the proper re-configuration of network interfaces.

Primary Internet Connection

The interface closest to the Internet should be chosen as your **Primary Internet connection**. This is used primarily if you enable NAT on any of your other subnets. If a subnet is being NATed, all packets coming from it through the router are translated to appear as if they are coming from the IP you designate with this setting.

NAT

NAT is used to allow a group of machines with private IP addresses like `10.0.0.0/8` or `192.168.0.0/24` to access the Internet. Packets flowing from any subnet with **NAT This Subnet** checked will have their source address modified to reflect the IP address of your **Primary Internet Connection**. Packets coming back to the **Primary Internet Connection** will then be sorted and passed passed to their real destinations. You only have to enable **NAT This Subnet** on a single IP from each subnet. For example, if you have five IPs on the `10.0.0.0/8` subnet, you only need to turn NAT on on any one of them to NAT all traffic from the `10.0.0.0/8` subnet.

Routing

The Cipafilter is a fully featured router capable of replacing the functionality of your existing routers.

Most customers will not need to reconfigure any features on this page, since the filter will automatically route between any subnets specified on the **IP Settings** page. If it is necessary to route to any additional subnets, they can be added here by specifying the destination subnet itself and the gateway to route through.

Multi-Gateway Routing

Multi-gateway routing provides advanced routing functionality which is useful for organizations with multiple Internet connections or complex network set-ups. These features provide several benefits, including load balancing and fail-over, as well as the ability to specify static routes based on the source rather than the destination (as with the **Routing** tab).

The **Multi-Gateway Routing Mode** indicates the type of routing to perform: **Destination Based** or **Source Based**. In **Destination Based** mode, the filter will distribute requests amongst the specified gateways according to their destination (external) IPs; in **Source Based** mode, the requests will be distributed according to their source (internal) IPs. The distribution is weighted according to the values specified, but is otherwise arbitrary.

Beneath the mode setting is the table of gateways and interfaces to use for routing. The **Gateway** field indicates the default gateway for the associated **Out-Interface**. The **Watchdog Target** specifies an IP address to watch for which indicates the availability of the gateway; if this target fails to respond, the filter will stop routing to the associated **Gateway** and fail over to one of the others. The **Weight** field specifies the distribution weighting; in general, the higher the **Weight** value, the more traffic will be routed through that **Gateway**.

As an example: Two gateways (Gateway A and Gateway B) are specified with equal weights using the **Source Based** method. Behind the filter are Clients A, B, C, and D. As each client passes a request through the filter, the client is "assigned" a route to one of these gateways. All subsequent requests from the clients are then passed through their associated gateways.

Ideally, given equal weights, the clients will be equally distributed — e.g., Clients A and C to Gateway A and Clients B and D to Gateway B — although this is not guaranteed. This functionality is referred to as load balancing.

If the **Watchdog Target** for Gateway B were to go down, the filter would then re-"assign" Clients B and D back to Gateway A to ensure connectivity. The exact settings the filter uses to detect this scenario can be changed under **Ping Watchdog Settings**. This functionality is called fail-over.

Note: Multi-gateway routing currently does not integrate well with the filter's DHCP-client functionality, and using the two features together is not supported. (DHCP-server functionality is unaffected.)

Ping Watchdog Settings

To ensure connectivity through a gateway, the filter must periodically ping its **Watchdog Target**. The number and frequency of pings can be set here.

Specific Connection Routing

This section is the source-based equivalent to the destination-based routes specified on the **Routing** tab.

Port Forwarding

Port forwarding is a system by which connections to ports on the router can be forwarded on to devices inside your network that only have private IPs. Oftentimes this feature will be used, for example, to forward RDP connections from your router to a server inside to enable remote management. When a client connects to a particular port on an IP address belonging to your router, the router instead causes it to connect to the **Destination IP** address on the second port number entered.

In the example table below, if your router has the IP `203.0.113.10`, and you would like to enable RDP to a server with the address `10.0.0.5` and another server with the address `10.0.0.6`, you would create a set of port forwards like the first two. This would allow you to connect to `203.0.113.10` with the default port to access the first server, and the custom port `3390` when you want to access the second server.

The third example demonstrates a dynamic port-forward, one in which the **Source IP** is in fact an interface with a dynamic IP address assigned via DHCP. When an interface is configured to use DHCP to obtain its IP address, the **`DHCP — eth#`** option will appear in the list; selecting this option creates a forwarding rule which is bound to the interface rather than to the interface's current IP address. This eliminates the need to update the rule when the interface's IP changes.

The fourth example demonstrates the use of port ranges, with the colon (`:`) character, to forward the 1000 ports starting at `64000` to `10.0.0.4`.

A comma (`,`) can be used to specify multiple ports, port ranges, or a combination of the two. Up to 15 distinct port references may be specified in a single rule. (A port range counts as two references.)

The **ALL** keyword in the port field (as shown in the second-to-last example) is used to forward all ports of a particular protocol to the same ports on another machine.

Note: Using the **ALL** keyword in the protocol field activates a 1-to-1 NAT between the **Source IP** and the **Destination IP**. In addition to standard port forwarding (all traffic to the **Source IP** will be directed to the **Destination IP**), all outbound traffic from the **Destination IP** will also appear to come from the **Source IP**.

Warning: Forwarding port `443` or port `22` on your router's only outside IP will make the Cipafilter's Web interface or remote management system (respectively) unreachable. If forwarding of these ports is required, attempt to obtain a second IP for your router. If this is not feasible, be sure that there is at least a second private IP address on the router, so you can manage the Cipafilter yourself.

Example port-forwarding configurations

Protocol	Source IP	Source Port	Destination IP	Destination Port	Comment
TCP	`203.0.113.10`	`3389`	`10.0.0.5`	`3389`	RDP to `10.0.0.5`
TCP	`203.0.113.10`	`3390`	`10.0.0.6`	`3389`	RDP to `10.0.0.6`
TCP	`DHCP — eth0`	`3390`	`10.0.0.6`	`3389`	RDP from a DHCP-enabled interface to `10.0.0.6`
UDP	`203.0.113.10`	`64000:65000`	`10.0.0.4`	`64000:65000`	1000 UDP ports (those between and including `64000` and `65000`) to `10.0.0.4`
UDP	`203.0.113.10`	`123,161`	`10.0.0.4`	`123,161`	Two UDP ports (`123` and `161`) to `10.0.0.4`
TCP	`203.0.113.10`	`20:25,70,79`	`10.0.0.4`	`20:25,70,79`	Eight TCP ports (`20` through `25`, `70`, and `79`) to `10.0.0.4`
TCP	`203.0.113.11`	**ALL**	`10.0.0.7`	**ALL**	All TCP ports to `10.0.0.7`
ALL	`203.0.113.12`	**ALL**	`10.0.0.8`	**ALL**	1-to-1 NAT of `10.0.0.8` to `203.0.113.12`

Stateful Firewall

A stateful firewall is a secure, easy-to-use firewall that tracks all open connections and the state of those connections. A regular (stateless) firewall only inspects packets.

With a regular firewall, if you wanted your client workstations to be able to access Web servers on the Internet you would have to allow any machine on the Internet to access all the high ports on any client. This is a vulnerability many worms and trojans have been written to exploit.

With a stateful firewall, you specify how connections are to be made. For example, you can allow any client inside your network to make a connection to port `80` on any server outside. When that happens, only that server will be able to send packets back to the high port on your client that originated the connection, and it will be allowed to do so only from port `80`.

The default policy of a firewall determines whether it drops or accepts connections by default. Cipafilter ships with the **Default Policy** set to **`ACCEPT`**. This mode is acceptable if the Cipafilter is in bridging mode behind another firewall. However, if the Cipafilter is the firewall in your network, we recommend setting the **Default Policy** to **`DROP`**. The **`DROP`** policy, while giving you specific control of the traffic passing through the firewall, will require the creation of rules for any traffic that you wish to allow through the firewall. Tech support will be happy to assist you with the creation of appropriate rules for your network.

Note: Connections are only matched against the firewall when they are first opened. If you change the firewall, any established connections will remain open even if the new firewall rules prohibit them. Also note that the firewall does not apply to Cipafilter itself; the router automatically adjusts its own firewall based on the configuration of the system to assure proper operation.

The firewall rule syntax is similar to what is used on the **Port Forwarding** page:

- `:` can be used to denote port ranges; e.g., `80:110` to match ports `80` through `110`.
- `,` can be used to string multiple ports or port ranges together; e.g., `21,22,80:110` for ports `21`, `22`, and `80` through `110`. Up to 15 port references can be used in a single rule.
- **`Accept`** allows the connection.
- **`Reject`** rejects the connection and sends an error message back to the client.
- **`Drop`** loses all the packets and sends no error message back to the client.
- Do not use any spaces in the rules.
- The firewall is fully integrated with the port-forwarding system. Make firewall rules using the actual private IP addresses of your internal servers, not the public IP address from which traffic is being port-forwarded.

Example firewall configurations

Action	Protocol	Connections From	To	On Port	Result
Reject	**TCP**	10.0.0.14/32	0.0.0.0/0	80	Workstation with IP address 10.0.0.14 cannot access the Web.
Reject	**TCP**	0.0.0.0/0	10.0.0.15/32	80,81,8080	All clients are prevented from accessing ports 80, 81, and 8080 on the internal Web server 10.0.0.15.
Reject	**ALL**	10.0.0.25/32	0.0.0.0/0	ALL	10.0.0.25 is prevented from making connections to the outside world.

The firewall rules are interpreted top down, so the first rule that matches a connection will determine its fate. For example:

Action	Protocol	Connections From	To	On Port	Result
Accept	**TCP**	203.0.113.0/24	10.0.0.15/32	80	Allow the 203.0.113.0/24 subnet to access the internal Web server.
Accept	**TCP**	198.51.100.0/24	10.0.0.15/32	80	Allow the 198.51.100.0/24 subnet to access the internal Web server.
Drop	**ALL**	0.0.0.0/0	10.0.0.15/32	**ALL**	Drop all connections to the Web server that have not already been accepted.

The first two rules will allow machines on the 203.0.113.0/24 and 198.51.100.0/24 subnets to access the internal Web server at 10.0.0.15. However, they will only be able to access the Web service on that server. All connections going to all other services or from other subnets will be dropped on the third rule.

Note: Firewall rules will only apply to traffic that is required to pass through the Cipafilter; in practice, this means rules will usually apply only to traffic between two different subnets. This is because traffic between two devices within the same subnet will generally be moved directly between them rather than going through their gateway.

ICMP Firewall

If you are experiencing a problem with ICMP traffic, simply enable the **ICMP Firewall** and select the types of ICMP packets you wish to let through.

Note: Always allow ICMP **fragmentation-needed** packets. These packets are required by Path MTU Discovery. If **fragmentation-needed** packets are blocked, you may experience problems where you can transmit small amounts of data over a connection but large amounts cause the connection to hang.

DHCP Server

The filter may act as a DHCP server for clients on the network, providing leases dynamically or statically via MAC address; this functionality is configured here.

Basic Configuration

The DHCP server functionality can be enabled or disabled (on a per-interface basis) via the **Basic Configuration** tab. Checking the **Activate** box in the table at the top of the page will enable DHCP on the corresponding interface. The **Start IP** field indicates the beginning of the dynamic lease range, while the **End IP** field indicates the end of the range. To aid in identification, the primary IP address of each interface is also listed here.

DHCP Relay

If your network already has a DHCP server you'd like to use — particularly one on a different network segment from your clients — the Cipafilter can act as a relay server. When in relay mode, the Cipafilter will capture DHCP traffic on its interfaces and forward it to/from the specified **Relay Target IP**.

Note: The **DHCP Relay** feature applies to all interfaces configured on the filter, and is incompatible with the normal **DHCP Server** function.

Static Reservations

Static IP reservations can be provided here. Each IP address entered here will be excluded from the dynamic lease range specified above, and will instead be reserved for the client with the specified MAC address. Which interface assigns the IP address is determined by the subnet it is a part of. The **Machine Name** and **Comment** fields are mostly for reference.

Advanced Configuration

By default, the filter provides to its clients a lease time of 864,000 seconds (10 days), and other sane configuration details derived primarily from those set on the **IP Settings** page. If these values are not suitable, they can be overridden here.

Active Leases

The **Active Leases** tab displays information about clients with active leases, as well as recently freed leases. Please note that **static reservations** are not treated like dynamic leases by the underlying server, and are therefore not tracked in the lease table.

SNMP Configuration

Simple Network Management Protocol (**SNMP**) is a method used to monitor network devices for connection problems and other issues.

To enable SNMP, click **Enable SNMP** and then fill in the fields below — enter each subnet (in CIDR notation) that is allowed to poll the Cipafilter in the **Polling Device Subnet** fields, and enter your community string (a sort of password which must be matched between communicating devices) in the **Community String** fields.

When SNMP is enabled, the Cipafilter will respond to MIB queries for LLDP (Link Layer Discovery Protocol) information.

VLAN Configuration

The Cipafilter can participate in VLANs on your network. Press ADD VLAN, then select the interface on the Cipafilter that participates in the VLAN, and finally enter the **VLAN Number**. Any interface references throughout the Cipafilter interface will reflect the VLAN tags created on this page.

VPN

PPTP — Client–Server VPNs

Cipafilter supports the PPTP protocol for client VPN access. PPTP is a very common standard and implementations of it are built into the Windows, Mac OS X, and Linux operating systems, as well as common hand-held appliances such as smart phones.

Simply configure a range of IPs (for example, `192.168.2.1-253`) in the **PPTP Range** box and add the appropriate user credentials to the **User Manager** page. These credentials will be used by clients to connect to the VPN. (LDAP integration is not currently available for this feature.)

Warning: PPTP VPNs are only as secure as the passwords you use. We recommend that you use very strong passwords if you are using the link to transmit confidential information.

IPsec — Router-to-Router VPNs

Cipafilter uses the more secure IPsec technology for router-to-router VPNs. Cipafilter's implementation is industry-standard and based on open-source technology, and will probably work with any IPsec-enabled device, but it is designed primarily to interconnect two Cipafilter units. If you want to interconnect a Cipafilter with another device we will try to assist you, but IPsec is a very complicated technology and we cannot guarantee the results.

To create an IPsec tunnel simply enter the public IP address of the remote end point and the remote network's subnet. Click SAVE AND APPLY and a new key will be generated for you. Copy this key to the remote end and configure the remote end with the local public IP and subnet. Click SAVE AND APPLY on the remote router and send a ping across the link to bring it up and test that it is functional. If you have any trouble establishing one of these links, please don't hesitate to call tech support so that we can troubleshoot any problems you may be having.

User Manager

The **User Manager** page allows you to add, remove, and configure user accounts used for proxy authentication and VPN services. It additionally provides configuration of a special account used for SSH console access.

Special users

Three special, non-removable accounts exist on all filters: `root`, `admin`, and and `guest`.

- The `root` user has access to everything on the filters's Web management interface, as well as all reporting and e-mail pages. This user also has permanent access to the Whitelist Management system (even if the **Cipafilter User Manager** authentication method is disabled).
- The `guest` user can access only the **Status** page, reporting systems, e-mail statistics, and this manual.
- The `admin` user can access everything `guest` can, plus **E-mail Archive**, **Group Permissions**, **Network Diagnostics**, and **Troubleshooting**.

User Accounts

User Manager accounts are added, removed, and configured from this tab. Each account can have a password and default group membership set here; in addition, regular users can also be configured for PPTP VPN access (the three special users described above are never allowed VPN access). To configure PPTP, please refer to the **VPN** page.

PPTP and (as described above) Web-management access settings configured via this page are always in effect. Authentication to the proxy server (including the captive portal) and Whitelist Management system are not permitted unless **Cipafilter User Manager** authentication method is enabled on the **Content Filtering** page (with the exceptions described above).

Note: The Cipafilter **User Manager** is designed to manage a small number of users, primarily those meant for VPN and Web-interface access — it is not intended to replace a directory service. A maximum of 2000 total user accounts is supported, and interface performance degradation may occur as this limit is approached. Customers with more than a handful of accounts are strongly encouraged to implement a full directory service (such as Microsoft Active Directory).

Advanced Configuration

This tab allows for the configuration of a special account called `cipacustomer`, which is provided for end users to access the filter console via SSH. To enable this feature, an SSH key pair must be generated using a tool like OpenSSH or PuTTY, and the public half of that pair must be entered into the **Public Key** field. After doing so and agreeing to the provided terms, the `cipacustomer` user account will be activated, and anyone with the corresponding private key installed on their local machine will have SSH access to the filter.

The Cipafilter firmware exposes the typical command-line features associated with Linux-based operating systems, and SSH access is provided as a troubleshooting aid for users who are already familar with such systems. Unlike the Web management interface, a filter's SSH console is not designed for ease of use; additionally, although most dangerous operations have been restricted, it is possible for an end user to interrupt or even irreparably damage a Cipafilter from the command line.

Users who do not already have a strong working knowledge of SSH, the UNIX file system and command shell, and the Linux environment are strongly discouraged from experimenting with the filter's console. There may be a fee of up to $1000 (in addition to any pre-agreed service costs) if support assistance is required to repair a filter that has been damaged through the use of this feature.

Hot Spare

Cipafilter's hot spare feature provides high availability by enabling one filter to monitor and take over for another in the event of failure. In a hot-spare setup, there are two filters: one is the *main* unit (sometimes referred to as the master or primary), and the other is the *monitor* unit (sometimes referred to as the spare). Under normal operating conditions, the main unit has the *active* status — that is, the status of the unit currently making use of the filter configuration and the one processing traffic. In a failure scenario, however, the monitor unit may assume the active status.

Configuration

To configure hot spare, a **Unit Default Mode** (described below) must be selected, and one of the partner unit's management IPs (as defined in the other filter's hot-spare settings) must be entered. Optionally, a wait interval (in seconds) can be set for the monitoring unit.

The management IPs of the filter being configured can be defined in the **Management Interfaces** table. All management addresses persist independently of the filter's other IP settings. Since these addresses are always available, the settings defined here must be unique for each hot-spare partner.

Unit modes

The following unit modes are selectable from the **Unit Default Mode** drop-down:

- **Main unit** — The filter with this default mode set will be in the one in the active status during normal operating conditions.
- **Monitor unit** — The filter with this default mode set will be the one which monitors the health of the main unit. Under normal operating conditions, this unit will generally be in the spare status, but during a failure scenario it will assume the active status.

Unit statuses

Each unit will, at any given time, exist in one of the following statuses:

- **Active** — This is the status of the filter which is currently processing traffic and allowing configuration changes. Normally, the main unit is the active one. Changes to filter settings will automatically replicate from the active unit to the one in the spare status.
- **Spare** — This is the status of the filter which is currently not active. When in this status, the unit is attempting to monitor the other unit; if it detects that the other has failed, it will take over the active status. A unit in the spare status will allow changes only to the hot-spare configuration itself; changes to the rest of the filter configuration will be prevented.

Fail-over

When a monitoring unit detects a failure, some form of fail-over must occur. The most common failure scenario is outlined below.

1. The main unit is in the active status, and the monitor unit is in the spare status.
2. The monitor unit detects a failure.
3. The monitor unit informs the main unit that it is going to assume the active status, and does so.
4. The main unit relenquishes the active status by rebooting.

Limitations

Although the **Hot Spare** feature does provide a solution to reduce down time, it has some limitations.

- **Communication loss with partner but not rest of network** — If the monitor unit is in the spare status, and it loses connection with the main unit in the active status, then the monitor unit will assume that the other filter has crashed and will assume the active status. However, if the main unit has not crashed but has actually lost connection to the monitor unit (without losing connection to the rest of the network), then the main unit will remain in the active status as well. In this circumstance, both filters being active will cause an IP-address conflict.

  The **Hot Spare** feature will not function correctly until communication between the two filters is restored. In the event that this problem occurs, one of the units should be manually powered off (this should be possible by connecting to a management IP). Once the connectivity problem has been resolved, it is safe to physically power the unit back on.
- **Database** — The **Hot Spare** function does not provide database redundancy. If the filters are configured to use the local database instead of a remote one, during a failure scenario the monitor unit will not have access to the database. Data from the period when the monitor unit was active will not be retained, and data on the main unit will not be accessible until normal function is recovered.
- **DHCP leases** — While static DHCP leases will be retained, dynamic leases may change sooner than otherwise expected. While worthwhile to be aware of, this should not impact network operation.
- **Bridging** — The **Hot Spare** feature is fundamentally incompatible with **Interface Bridging**.

First set-up guide

The following guide outlines the set-up procedure for a new **Hot Spare** configuration. Before beginning, decide which filter will be the monitor unit and which will be the main unit, and decide on the IP addresses you will use for their management. Then, read and understand this guide before proceeding with first-time set-up.

1. Configure both filters to have the same cable set-up. All physical connections to the main unit should be identical to those of the monitor unit, including VLAN set-up, etc.
2. Give the `root` user the same password on both filters.
3. Connect to the Web interface of the monitor unit and navigate to the **Hot Spare** page.
    a. Configure one or more **Management Interfaces** for this unit.
    b. Configure the **Unit Default Mode** for this unit (**`Monitor unit`**).
    c. Place one of the management IPs configured for the other filter (the main unit) in the next step into the **Partner Unit's Management IP** box.
    d. Press **SAVE CHANGES**.
    e. Press **APPLY MANAGEMENT INTERFACES**.
4. Connect to the Web interface of the main unit and navigate to the **Hot Spare** page.
    a. Configure one or more **Management Interfaces** for this unit.
    b. Configure the **Unit Default Mode** for this unit (**`Main unit`**).
    c. Place one of the management IPs configured for the other filter (the monitor unit) in the previous step into the **Partner Unit's Management IP** box.
    d. Press **SAVE CHANGES**.
    e. Press **APPLY MANAGEMENT INTERFACES**.
5. Return to the **Hot Spare** page of the monitor unit and press **TEST CONFIGURATION**. If everything has been configured correctly, a success message should appear. Check the **Enable Hot Spare** box and press **SAVE CHANGES**. A red error may appear, informing you that the configuration was not copied; this is normal in this situation and may be ignored.
6. Return to the **Hot Spare** page of the main unit and press **TEST CONFIGURATION**. If everything has been configured correctly, a success message should appear. Check the **Enable Hot Spare** box and press **SAVE CHANGES**. Two success messages should appear — one will inform you that the changes were saved, and another should indicate that the changes have been successfully deployed to the partner unit.
7. Reboot both filters.

Customization

The **Customization** page provides the ability to customize the appearance and content of the portal and reject pages.

Note: Although it is not required, all of the text fields on this page accept raw HTML. Advanced users may wish to take advantage of this to add JavaScript or style sheets to the page, but even novice users can benefit from the text formatting options that HTML can provide. Please also note that entering any HTML at all into this field will put the resulting text into a raw "HTML display mode"; so, for example, if a bold (`<b>`) tag is added, you must also add paragraph (`<p>`) or line-break (`<br />`) tags to display paragraph breaks correctly.

Banner Image

The **Banner Image** is an image displayed in place of the Cipafilter logo on public-facing pages. The uploaded image should be in PNG format and should be designed to display cleanly at 96 pixels tall. If older browsers (Internet Explorer 8 or below) are used in your environment, it is probably best to ensure that the image is exactly 96 pixels in height. However, if these older browsers are not a concern, the image can be of any dimensions — it will be resized automatically. One benefit to using an image taller than 96 pixels is that it will look crisp on high-resolution mobile devices.

Authentication Notice

An **Authentication Notice** will be displayed on the portal whenever a user is required to log in. This is a brief disclaimer which informs users that they need to enter a user name and password, and instructs them to contact their network administrator if they require assistance. If your organization has a different process (for example, students must contact a particular teacher), you may wish to detail it here. If no text is defined, the Cipafilter default notice will appear on the portal; this should be sufficient for most organizations.

Acceptable Use Policy

If your organization has an acceptable use policy (AUP) or terms of service, the text can be entered into this field. When present, a box containing this text will appear on the portal. If no text is defined, nothing will be displayed in its place (there is no default).

SSL Decryption Notice

An **SSL Decryption Notice** will be displayed on the portal whenever a root certificate has been generated via the **Content Filtering** page. This is a brief disclaimer which alerts users that their secure (HTTPS) traffic may be subject to decryption. If this field is left blank, the Cipafilter default notice will appear on the portal; this should be sufficient for most organizations.

Reject Notice

The **Reject Notice** is a general-purpose text field for displaying on the filter reject page (displayed when a Web site has been blocked). This notice can be used for any purpose; some organizations may wish (for example) to include a link to request unblocking.

Additional Reject Options

By default, the filter reject page displays text and a button offering users the option to temporarily whitelist the blocked site via the Whitelist Management feature and/or to log in to the captive portal if they are not already authenticated. (Users must be authorized to access each feature and must always authenticate after clicking the button.) The checkboxes under this section can be used to enable/disable this functionality where applicable.

Content Filtering

The first thing to decide with regard to content filtering is whether to run individual subnets in transparent or non-transparent (proxy server) mode.

In transparent mode, no client configuration is required — the Cipafilter simply intercepts all traffic on ports `80` and `443` as it moves through the router.

In non-transparent mode, each client must be configured to make use of a proxy service provided by the Cipafilter (on port `6226`, by default). For example, in Internet Explorer, the Cipafilter proxy must be added on the Connections tab under Internet Options.

Basic Configuration

Subnets Authorized to Use Proxy Services

Only machines with IPs in the subnets listed under **Subnets Authorized to Use Proxy Services** will be allowed to use the proxy server. Subnets are provided in **CIDR notation**. If two subnets overlap, the smallest or most specific subnet's configuration applies (just like in the **Routing** configuration), with the exception of the **Transparent Proxy** option as described below.

Note: The Cipafilter's non-transparent proxy functionality is "always on" for any subnets listed here (although you need not use it).

Transparent Proxy

- **Yes** — HTTP and HTTPS connections from this subnet will be transparently intercepted by the Cipafilter. When SSL decryption is configured for a group on this subnet, the HTTPS traffic will be inspected/altered in essentially the same manner as HTTP traffic; if decryption is not available, HTTPS connections will be subject to domain-based black- and white-listing only (URL blacklisting, content-aware filtering, anti-virus, and other features which rely on the unit being able to "see inside" the connection will not function).
- **No** — Connections from this subnet will not be transparently intercepted by the Cipafilter unless they fall under another subnet entry in the list. (For example, a rule setting `10.1.2.0/24` to **`Yes`** will still apply to `10.1.2.3/32` if the latter is set to **`No`**.)
- **Disable** — A specific rule will be created to prevent transparent interception of this subnet (overriding any other **Transparent Proxy** rules that may apply).

Authentication

- **Optional** — Clients on this subnet are allowed, but not required, to authenticate as a specific user. When not otherwise authenticated, users on the subnet will be placed into the **Subnet Group**, and will be logged under their IP address. If the user chooses to authenticate, they will be logged under the specified user name and placed into the appropriate user group (if applicable).
- **Required** — Clients on this subnet are required to authenticate before the filter will allow them to access the Web. Users may authenticate via user name and password or (if available) a Cipafilter authentication client. Clients on both transparent and non-transparent subnets have the option to use the portal for authentication; by default, however, non-transparent subnet users receive authentication prompts via their operating system or Web browser.

Proxy and portal credentials are authenticated against the authentication services configured on the **Authentication** tab. Each user's Web traffic is logged by user name, if Web monitoring is active, and can have individual "filtered" or "not filtered" settings.

Portal Log-In

- **Enable** — Users on this subnet may use the captive portal system for authentication, if they visit it manually or if they would be subject to redirection due to the **`Required`** authentication type.
- **Persistent** — This option is the same as **`Enable`**, except that users on this subnet will be authenticated persistently (their browser's authentication state will be remembered by the filter).

  Normally, when a user authenticates to the portal, their session is remembered for the amount of time specified on the **Content Filtering** or **Group Permissions** page. By default, this duration is 12 hours.

  If this option is checked, each user's portal authentication state will be stored in their browser's cookies for an effectively permanent term. When the portal detects such a cookie, it will be verified for authenticity and, if successful, the user will be re-authenticated automatically — they will not actually see the portal again unless they navigate to it themselves.

  Since this option requires cookies, the effect is per-device and per-browser — a user who accesses the network via multiple machines will need to authenticate at least once from each of them. Using a different browser or deleting the original browser's cookies will clear the effect.
- **Disable** — Clients on this subnet will never be automatically redirected to the portal for authentication; users with browser proxy settings will be made to authenticate using their operating system or browser's standard credential prompt, and users subject to transparent proxy will not be prompted to authenticate at all.

  In addition to not being redirected when authentication is required, users on a disabled subnet will be unable to authenticate to the portal if they navigate to it manually (this may be significant if administrators commonly use the portal to temporarily increase privileges for a client, for instance). However, users will always have the option to *view* the portal by navigating to it manually, if for example they need to reference the AUP or install an SSL certificate.

Note: In all cases, the use of a Cipafilter authentication client will bypass redirection to the portal.

Force Subnet Group

Selecting this option forces clients on the specified subnet to always use the group selected as the **Subnet Group** (instead of using any groups that may be defined in LDAP, for example).

Subnet Group

This drop-down lists all of the groups configured on the **Group Permissions** page. The group selected here will be used for unauthenticated users on an **`Optional`** authentication subnet, and for all users on a subnet with **Force Subnet Group** enabled.

Note: If **`Required`** authentication is not selected, workstations with a Cipafilter authentication client installed will be automatically authenticated and filtered based upon the workstation users' group memberships. Workstations without the Cipafilter authentication client will be filtered based upon the **Subnet Group** setting for the subnet, with the option to authenticate as another user via the portal system or proxy settings.

Insert Remote-Filtering (1-to-1) Rule

This button, shown at the bottom of the list of authorized subnets, inserts a rule into the table to be used for remote filtering / 1-to-1 initiatives. This is simply a shortcut for creating a rule for the `0.0.0.0/0` subnet, which acts as a "catch-all" for any clients not specified on another subnet. Because the `0.0.0.0/0` rule authorizes anyone on the Internet to use the filter as a proxy, it is strongly recommended that authentication be **`Required`** for this subnet.

Advanced Configuration

When **Internet Reporting** is enabled, user activity and filter trips are collected for the **Internet Reports** system to analyze. If this feature is not enabled, the Cipafilter will not record user activity other than for the purpose of e-mailing the administrator when the content filter is tripped.

The **Proxy Port** is the port that the Cipafilter listens for proxy connections on when being used as a non-transparent proxy. The default value is `6226` (`8080` in earlier versions).

The `X-Forwarded-For` header is an HTTP feature which allows a proxy service to pass the IP address of the originating client. (For instance, if a client with IP address `1.2.3.4` attempts to access a Web server via the proxy server `4.3.2.1`, the Web server will only see the latter address, unless the `X-Forwarded-For` header is provided by the proxy server.) When this option is enabled, the Cipafilter will use the `X-Forwarded-For` address (where available) in its logs and reports.

YouTube for Schools

YouTube for Schools is a service provided by YouTube which allows school administrators to define a set of educationally appropriate videos which are accessible from the organization's network. This feature is enabled on a per-group basis, so the corresponding option on the **Group Permissions** page must also be selected. For more information on setting up YouTube for Schools, please see **YouTube for Schools - Frequently Asked Questions**.

Google Apps Domain Restriction

Many Google Web properties, including Gmail, support a custom header which restricts log-in access to accounts which are members of a specified domain. For instance, if an organization at `myschool.edu` used Google Apps for e-mail and document sharing, they could restrict users to accessing those sites only through their `myschool.edu` accounts; trying to log in with any other accounts (including "consumer" Gmail accounts) would result in an error.

To make use of this feature, select **Enable Google Apps domain restriction** and enter your organization's Google Apps-enabled domain(s) — multiple domains can be separated by commas, e.g. `myschool.edu, myschool.com` — into the **Allowed Google Apps Domains** field. Then, enable the corresponding option for each desired group on the **Group Permissions** page.

Note: The `google.com` domain must be subject to SSL decryption for this feature to work. For more information, please see **Block access to consumer accounts - Google Apps Help**.

Anti-Virus Settings

By default, Web downloads and e-mail attachments are run through the filter's anti-virus engine. Unchecking the **Enable anti-virus** option will globally disable this functionality. It is **strongly** recommended to leave this and all other anti-virus features enabled, unless there is another equivalent facility on the network or running locally on client machines.

PDF files are a common vector for infection, as the PDF format is extremely complex and featureful, potentially allowing arbitrary code execution. When a PDF is passed to the anti-virus system, it will be flagged as potentially malicious if it can not be assessed for safety due to the file being encrypted or similar. This is considered good security practice; however, some organizations which deal frequently with documents from multiple sources may be frustrated by this behavior. In this case, PDF scanning may be disabled entirely by unchecking the **Perform PDF anti-virus scanning** option.

Similarly, encrypted files (including ZIPs, PDFs, and others) may contain an infection source, and the anti-virus system by default blocks these files as well, since they can not be scanned. Unchecking **Block encrypted archives and PDFs** will disable this behavior.

Note: The options for disabling PDF scanning and disabling encrypted archive scanning both affect PDFs — the former disables all scanning of PDF files, while the latter disables all scanning of encrypted files (including encrypted PDFs). While both options have their advantages and disadvantages, disabling encrypted archiving scanning is recommended in situations where the trouble is specifically with encrypted documents, since non-encrypted PDFs will still be scanned.

Transparent Proxy Exceptions

When in transparent mode, connections to subnets listed under **Transparent Proxy Exceptions** will not be intercepted by the proxy server. This is useful for applications and services which are not compatible with the transparent proxy method.

Authentication

Authentication settings for proxy services and the captive portal are configured here. Cipafilter supports multiple authentication methods via a fall-back scheme — any number or combination of methods may be supplied, and the filter's authentication engine will try each in order until one succeeds or the list is exhausted.

Note: Because the available methods are tried one at a time, each additional entry added to the list may incur a certain performance cost at authentication time. For this reason, it is recommended that the most frequently used methods be listed first. For example, if teachers use one directory service and students use another, the students' service should usually be placed ahead of the teachers', since it will be authenticated against far more often.

Supported LDAP directory services include **Windows Active Directory**, **Apple Open Directory**, and **Novell eDirectory**. To utilize LDAP authentication, a search base and the IP of your directory server must be provided. Active Directory authentication also requires access to a user account that has read permissions for all of the directory's user/group objects.

The filter can often detect the domain and/or search base required for a specified LDAP server; clicking the **Auto-detect LDAP settings** (magic wand) button can auto-fill these values into the table, where available.

In addition to LDAP services, Cipafilter can also use **Google OAuth** as an authentication back-end. This feature enables the use of Google Apps as a directory service; users will be authenticated according to their Google credentials, and group memberships will be derived from their Google groups (distribution lists). In order to select this option, the **Google OAuth Settings** section must be filled in correctly. For more information, see **Google OAuth**.

Authentication is also supported via the local **User Manager** service — please see **User Manager** for details.

Filter groups are defined on the **Group Permissions** page. Each user who successfully authenticates will be checked for membership in a group of the same name in the directory service. If a membership is found, the user will be configured according to the permissions of that group. If no membership is found, the permissions for the `default` group (or the effective **Subnet Group**) will be used instead.

Cipafilter checks these group memberships upon each user's first access and caches the information for up to one hour, depending on the protocol. Click SAVE AND APPLY on the **User Manager** page of the Web interface to clear this cache.

Under this section you will also find a link to the **Authentication Tools** page; this page is intended for troubleshooting, particularly with the assistance of tech support, and can be used to view/modify the current state of the filter's authentication system.

Portal Settings

Users with browser proxy settings prefer portal for authentication

By default, clients using browser proxy settings to interact with the filter (that is, clients subject to the non-transparent proxy mode) authenticate via their operating system or browser's credential prompts. Enabling this option will instead redirect these users to the portal, giving them effectively the same authentication experience as transparently proxied clients. If **Use Portal** is disabled for the user's subnet, they will continue to use the normal proxy authentication methods.

Present portal even when authentication is not required (guest access)

Selecting this option will enable a "guest access" mode for all subnets which do not explicitly require authentication. In this mode, all affected clients will be directed to the captive portal, but, in addition to the normal authentication options, a "Log in as Guest" button will be present, allowing users to proceed even if they don't have credentials. This option is particularly useful for displaying the network's usage policy without requiring a log-in.

Block all non-Web traffic when authentication is required

By default, when **`Required`** authentication is enabled for a subnet, Web access (ports `80` and `443`) is blocked to affected clients until they authenticate. For some organizations, this is sufficient — however, it does leave other ports open, potentially allowing clients to access non-Web-based network resources (for example, chat clients or file-sharing software). Enabling this option will prevent this, by extending the block to (almost) all other traffic until the client has been authenticated.

Use Google OAuth for portal authentication

This option enables a special Google "front-end" log-in method, which displays a `SIGN IN WITH GOOGLE` button on the portal. Clicking this button will direct the user to a page where they can authenticate with their Google credentials, which are subsequently passed back to the filter.

In order to enable this option, the **Google OAuth Settings** section must be filled in correctly. For more information, see **Google OAuth**.

Note: This option is ignored when using the **Google OAuth** back-end; since the Google back-end and the portal can not function together without the Google front-end, the front-end is always enabled when the back-end is in use.

Allow non-Google (LDAP) log-ins from portal

By default, when using the Google OAuth portal log-in, the normal "back-end" log-in form (which accepts credentials for the authentication services specified above) is disabled. Checking this box will re-enable this log-in form. This is useful if some users do not have Google accounts, but do have credentials on the back-end.

Note: Since the **Google OAuth** back-end does not accept bare credential log-ins, this option is forcibly disabled when Google is the only configured the only configured authentication service.

Expire portal authentication after ___

This field specifies the default amount of time a portal log-in session should last for. The default value is 12 hours, but it can be increased or decreased according to an organization's needs. For instance, a school may wish to have the session last only until the end of a class period.

This value may be overridden on a per-group basis via the **Group Permissions** page (if the option is set there, that value will be used; otherwise, this one will). Please note that this value also applies to the **guest access** feature.

Google OAuth Settings

In order to make use of the **Google OAuth** options on this page, the settings in this section must be filled in, and a Google domain administrator account authorized. For detailed instructions on obtaining or configuring these values, please see **Google OAuth**.

- **Client ID for Web Application** — This field should match the *Client ID* value from the Google Developers Console.
- **Client Secret for Web Application** — This field should match the *Client Secret* value from the Google Developers Console.
- **Google Apps Domain** — This field should contain your organization's Google Apps domain (the part after the `@` in your Google Apps e-mail addresses).
- **Redirect URIs** — These values are generated by the filter; they are provided to facilitate the initial set-up of Google OAuth for your domain.
- **Google Access Token** — This field will indicate whether Google OAuth is properly configured. If it does not, you may need to click **Authorize a Google Domain Administrator Account**.

RADIUS Accounting

Cipafilter is able to track user log-ins via RADIUS accounting. In this scenario, a supported network access server (NAS), such as a wireless access point, negotiates authentication with a dedicated RADIUS server, which subsequently forwards accounting data to the filter. As the filter receives this data, it updates its authentication database accordingly, authing or de-authing users as required.

This feature requires one or more independent RADIUS servers, each configured to forward accounting information to the filter, as well as one or more NAS devices which communicate with the RADIUS server(s). In order to integrate with Cipafilter, each NAS must support the `Framed-IP-Address` accounting attribute. (This attribute is supported by many enterprise-class wireless solutions, including Cisco/Meraki and Aruba.)

To enable this feature, configure your RADIUS server(s) to forward accounting information to the filter (using a shared secret key for security). Then add each server's address and its shared key to the **RADIUS Accounting** table. (The filter's RADIUS service is automatically enabled when a server is defined here.)

When a user attempts to access the network via a NAS, the NAS device will ask the RADIUS server to authenticate the user. If successful, the server will forward the authenticated user name and IP address to the filter, which will be (re-)injected into the local authentication database. Note that the RADIUS server does not pass group or password information; as such, the filter will always use the group information from the first authentication back-end on which the user name exists.

Note: At this time, a Cipafilter can not act as a RADIUS server itself — in other words, it can not handle RADIUS authentication/authorization requests on its own. A dedicated server must be configured to interface with the NAS.

SSL Configuration

By default, the Cipafilter is unable to see inside of secure (HTTPS) connections. **SSL decryption** is a process whereby the filter inserts itself between the client (browser) and the server (Web site) and builds its own HTTPS connection between the two. This allows the filter to make use of all of its standard capabilities, such as URL blacklisting, pornography detection, and anti-virus, just as it could with an insecure connection. The initial set-up for this process is handled on the **SSL Configuration** tab.

To perform SSL decryption, the Cipafilter must be able to generate and serve local SSL certificates to clients connecting to secured sites. A root certificate authority (CA) is required to sign these certificates; creating this root CA is accomplished by entering the desired information into the **SSL Certificate Information** section and then clicking the GENERATE CERTIFICATE button below. The specified information will appear in the certificate details for any filtered HTTPS connection.

After the certificate authority has been generated, SSL decryption will be available to all filter groups; the exact decryption behavior, including which sites should be decrypted, can be configured on the **Group Permissions** page.

Please note that, although generating the certificate authority is technically the only requirement to enable SSL decryption functionality, users will repeatedly be prompted to ignore security alerts in their browsers (or even be prevented from accessing certain sites entirely) unless they have trusted the generated CA. Users can download the CA certificate at any time by visiting the captive portal (**portal.cipafilter.com/ssl**); it will also be installed automatically (for browsers which respect the operating system's certificate store) by the Cipafilter authentication clients for Mac and Windows.

A note about the HTTPS protocol:

HTTPS is not a fully separate protocol from HTTP; rather, it is HTTP *over* a secure connection. In other words, all of the traffic *within* the secure layer works in the same way as non-secure HTTP traffic. This is an important distinction for a filtering appliance, as it affects what the filter can and can't do with secure connections.

When a client (such as a Web browser) accesses a secure Web site, it creates the secure connection *before* any actual HTTP traffic is passed back and forth. This is why it is not possible to blacklist entire URLs when SSL decryption is disabled — the URL request occurs inside of the secure connection, so the filter can't see it without intercepting that connection.

Another consequence of this behavior is that the domain name the client establishes the connection with may not be the same as the domain name of the specific Web site it is trying to reach. This is because multiple Web sites may use the same IP address, and the traditional method of distinguishing between different sites on the same address (the HTTP `Host` header) is, once again, only visible from inside of a secure connection.

To address this problem, a technology called Server Name Indication (SNI) was created, and Cipafilter uses this to make the black- and white-listing of HTTPS sites more accurate. However, this functionality only works when the client supports it. Notably, older mobile browsers and versions of Internet Explorer running on Windows XP have no SNI support, nor do many non-browser applications (such as Google synchronization tools). Accordingly, blacklist entries may have unexpected results when these clients are involved.

Portal Certificate

The captive portal is SSL-encrypted (that is, it uses the HTTPS protocol) for security. Without this encryption, it would be extremely easy for anyone on your organization's network to view the traffic moving between other clients and the portal, and by inspecting that traffic, they would be able to capture the user names and passwords of anyone logging in.

To achieve protection from this sort of snooping, the portal must use a certificate. This certificate is in turn signed by a certificate authority (CA). By default, the CA used to sign the portal's certificate is the same one which is created in the **SSL Configuration** section (mentioned above).

However, this default configuration can be problematic:

- The portal site will be accessible only from the `portal.cipafilter.com` address, which some organizations are uncomfortable with for security or branding reasons.
- The portal will display a security warning message to any user who has not yet trusted the filter's CA. This message is unattractive and confusing.
- Some clients, particularly the browsers on older mobile devices, won't display a warning at all — the portal will simply fail to load entirely.

The **Portal Certificate** feature provides a solution to all of these concerns, by allowing an organization to specify its own custom certificate, particularly one which has been signed by a trusted public CA.

To use the feature, enter the desired information into the **Custom Certificate Information** section. Much of this will be the same as what is entered under **SSL Certificate Information** (described above). The **Common Name** field is of particular importance — it represents the site address that will be used for the portal when this feature is active. As an example, an organization with the domain `myschool.edu` may wish to use `portal.myschool.edu` as their portal's **Common Name**. The **Common Name** must **not** match the host name of the filter itself.

Once the certificate information has been entered, clicking GENERATE REQUEST will generate a certificate signing request (CSR) for download. This CSR file is used by a certificate authority to generate the needed certificate file. Simply submit this file to the certificate provider of your choice, such as Namecheap or GoDaddy (Note: Cipafilter does not endorse or recommend any particular certificate provider). Typically, one can expect a certificate suitable for this purpose to cost between 5 and 30 USD per year.

The certificate provider should respond with a public certificate and a CA bundle; once these are in hand, they need simply to be uploaded to the filter via the **Upload Custom Certificate Data** section.

It is also possible for an organization to use a wildcard certificate or one signed by an internally trusted CA, but this is not currently exposed to the Web interface. If you would like assistance configuring the portal this way, or if you have any other questions at all, tech support will be happy to assist you through the process.

Finally, whatever **Common Name** is chosen for the portal must have a record on or accessible to whatever DNS server (internal or external) the portal's clients are using. The recommended configuration is to point the custom domain to the special "bogon" IP address that the filter uses for the portal internally — this address is `192.0.2.1`.

PAC files / secure proxy

Many browsers and operating systems support the use of a PAC (proxy auto-configuration) file. This file provides a centralized method of indicating to the client how to choose a proxy server when making a Web request. Cipafilter supports several PAC files with various functions.

The following PAC files are available for standard HTTP proxy settings at the filter's Web-management address:

- **`filter.domain.com/proxy.pac`** (where *filter.domain.com* is one of the filter's Web-management addresses) — This file directs clients to an HTTP-only proxy at the filter's canonical Web-management address.

Secure proxy

The standard proxy functionality used by most client software is implemented using special plain-text HTTP headers. This makes proxy requests somewhat insecure, in that the user's credentials are passed to the server in a non-encrypted form which is readable by other devices on the same network. Recently, some browsers have implemented *secure* proxy functionality, which wraps the proxy connection in an HTTPS/SSL tunnel. This is beneficial not only because it hides the user's proxy credentials, but also because it prevents traffic between the user and the proxy from being intercepted by third parties, even when the user is requesting an insecure Web site.

The following PAC files are available for secure HTTPS proxy settings at the filter's captive-portal address:

- **`portal.domain.com/proxy.pac`** (where *portal.domain.com* is the filter's custom portal address) — When a custom portal certificate is installed, this file directs clients to both an HTTPS proxy and an HTTP proxy at the custom portal address. The HTTPS proxy will be used by browsers that support secure proxy, while other browsers will use the HTTP variant. If no custom portal certificate is available, the file directs clients to an HTTP-only proxy at the default portal address (portal.cipafilter.com). This file is the easiest to use, since it works with most browsers and provides a secure connection where available.
- **`portal.domain.com/proxy-https.pac`** (where *portal.domain.com* is the filter's custom portal address) — When a custom portal certificate is installed, this file directs clients to an HTTPS-only proxy at the custom portal address. If no custom portal certificate is available, the file has no function.

There are three requirements for using secure proxy with Cipafilter:

1. The Cipafilter must be configured to use a custom portal certificate.
2. The client must be using a browser which supports secure proxy. This includes recent versions of Google Chrome and Mozilla Firefox.
3. The browser must be configured to use one of the PAC files available at the filter's custom portal address (described above).

Please note that the proxy configuration screens of many applications have one text field for an HTTP/insecure proxy address and one for an HTTPS/secure proxy address. This usually **does not** refer to the secure proxy functionality described above — it simply specifies the *insecure* proxy to be used for HTTPS requests from the browser. Most software which supports secure proxy currently requires the use of a PAC file to enable it.

Portal

The Cipafilter portal is a Web site that acts as a central point for Web-based authentication and SSL certificate installation. It can be accessed manually from any client which is proxying through the Cipafilter via **portal.cipafilter.com**; some users may also be redirected to the portal automatically (for authentication or other purposes) depending on the settings specified on the **Content Filtering** page.

The latter feature is particularly useful for mobile devices accessing the network through an organization's "BYOD" (bring your own device) policy — by enabling the portal for these devices, you can ensure that mobile users are aware of your network usage policy and the need to trust your Cipafilter's root CA for an optimal experience when SSL decryption is enabled.

By default, a user logging into the portal will authenticate that user's IP address for 12 hours. This time-out period can be adjusted from the **Authentication** section of the **Content Filtering** page, or within the group configuration sections of the **Group Permissions** page.

Authentication is a key aspect of the portal system, but it is not required — by enabling the **guest access** feature, administrators can force users to the portal without actually requiring them to authenticate. This is useful if, for example, it is desired to show the network usage policy to all users.

The portal also provides an easy way for authorized personnel to re-authenticate with another account; this can be helpful if, for example, a teacher needs to use a computer which would otherwise be filtered according to the student policy.

Because clients are authorized by IP address, one user's authentication state can, in some configurations, carry over for the next user who accesses the same machine. One way to prevent this is to use the following **log-out URL** in a log-on script or as the client's default browser home page:

https://portal.cipafilter.com/?logout=true&request=http://google.com

This URL immediately deauthorizes the client and then redirects the browser to the Web site specified by the optional **`request`** variable (`http://google.com`, in this example).

Lastly, the portal's SSL guides (available at **portal.cipafilter.com/ssl**) provide detailed instructions for end users to trust the Cipafilter root CA, when one has been generated. The portal automatically detects the user's browser/platform and displays the appropriate guide.

Google OAuth

Google APIs support the use of the **OAuth 2.0** protocol; among other things, this allows Cipafilter to securely interface with Google Apps domains for the purpose of authentication.

Two distinct but related Google authentication methods are provided by the Cipafilter software: a back-end service and a portal front-end.

Google OAuth back-end

Adding **Google OAuth** to the authentication services on the **Content Filtering** page enables the Google OAuth back-end feature. This provides for Google Apps itself to act as a directory service, bypassing the need for an LDAP server at all. When this feature is selected, the Cipafilter will check for a user's existence in Google Apps and, if applicable, derive group memberships from that user's Google Apps groups (aka distribution lists).

There are some caveats to this authentication method, however, which do not apply to the LDAP methods. Most notably, Google does not support actual credential authentication through their APIs; because of this, the Cipafilter can not directly perform user-name and password validation against Google Apps user accounts. Instead, authentication may be provided by the Google OAuth front-end feature (described below) or by a compatible Cipafilter authentication client. The authentication clients for Windows and OS X can "simulate" a successful Google log-in when the user's domain or work-station user name matches that of their Google Apps account; the authentication is assumed to be successful based on the fact that the user was able to log in to the work station.

The Google OAuth back-end does **not** work with browser proxy prompts or with the Cipafilter Chrome Authenticator extension.

Google OAuth front-end

Checking **Use Google OAuth for portal authentication** on the **Content Filtering** page activates the Google OAuth portal front-end. This feature enables the captive portal system to hand off authentication to Google itself, which subsequently reports the success or failure of the operation back to the Cipafilter. After confirming the success of the user's log-in, the Cipafilter can use any configured authentication service to match the user to a group.

Example: Suppose Central High School has an internal Active Directory server as well as Google Apps accounts for all students. A student named Joe Bloggs has an account `jbloggs` in Active Directory, and an account `jbloggs@centralhigh.edu` in Google Apps. With the Google OAuth front-end feature, Joe can sign in to the portal with his Google e-mail address, and the Cipafilter will match the user name from that address to the user name in Active Directory. Upon finding a match, the filter will use the Active Directory groups to place the user into the appropriate group on the filter.

This feature can also be used in conjunction with the Google OAuth back-end (described above). By combining the two, organizations can use Google Apps as a complete substitute for more traditional directory services.

Because it requires Web-based interaction from the users themselves, the front-end feature is only beneficial to portal users; it does not affect browser proxy prompts or Cipafilter authentication clients.

Combining Google OAuth with LDAP

For organizations looking to combine traditional LDAP services with Google authentication, Google provides a free product for Windows and Linux called **Google Apps Directory Sync**, which automatically synchronizes directory information between Google Apps and a local LDAP server.

Using this synchronization solution is generally more robust than the Google OAuth back-end, due to the limitations described above.

Google OAuth initial setup

Both the front-end and back-end Google OAuth features require a one-time setup on the Google Apps side before they can be used with the Cipafilter:

1. Log in to the Google Admin Console for your organization's domain via **admin.google.com**. (If this link doesn't work, try browsing to `https://google.com/a/<yourdomain.com>`, where `<yourdomain.com>` represents your organization's Google Apps domain.)
2. From the front page of the Admin Console, click *Security*, then *API reference*; then, under the *API access* section, check *Enable API access*. A *Save changes* button will appear at the bottom of the page; click it. **Note:** This settings change can take up to 24 hours to take effect.
3. Log in to the Google Developers Console via **console.developers.google.com**.
4. Click the drop-down menu at the top of the Developers Console page, and select *Create a project...*. You will be prompted for a project name and ID; enter `Cipafilter OAuth` or similar for the name (the ID can be left as the default), then click *Create*.
5. After a moment, you should be directed to the dashboard for the new project (if not, go back to the projects drop-down and click on the new project name). On the left-hand side of the screen, click *APIs* under *APIs & auth*.
6. Under the *API Library* tab, locate the *Admin SDK* and *Google+ API* options, and enable them by clicking their names and then clicking *Enable API*. **Note:** This settings change can take up to 24 hours to take effect.
7. On the left-hand side of the screen, under *APIs & auth*, click *Credentials*.
8. Click the *OAuth consent screen* tab and enter (at least) an *Email address* and *Product name*, then click *Save*. The information on this tab will be used to display the consent screen that users see when they try to authenticate via Google.
9. Click back to the *Credentials* tab, then click the *Add credentials* drop-down and select *OAuth 2.0 client ID*.
10. On the next page, select *Web application*. A series of configuration fields should appear.
11. Under *Name*, enter a memorable name for the client ID you're creating, such as `Cipafilter OAuth`.
12. Leave the *Authorized JavaScript origins* section blank.
13. Under *Authorized redirect URIs*, paste each of the **Redirect URIs** listed on the **Authentication** tab of the filter's **Content Filtering** page. (You may have to copy and paste each line individually.)
14. Click *Create*. After a moment, you will be shown a dialogue containing the OAuth client ID information. (You can view this information later by clicking the client ID name you entered previously from the *Credentials* tab.) This information will be used to configure the filter.

Having completed the initial configuration on the Google side, you should be able to configure the Cipafilter itself:

1. In another tab or window, access the Cipafilter Web interface, then navigate to the **Content Filtering** page, then to the **Authentication** tab, and scroll down to **Google OAuth Settings**.
2. Paste the client ID from the Google Developers Console into the **Client ID for Web Application field on the filter interface.**
3. Paste the client secret from the Google Developers Console into the **Client Secret for Web Application field on the filter interface.**
4. Enter your Google Apps domain name (e.g., `yourdomain.com`) into the **Google Apps Domain** field.
5. Click **SAVE AND APPLY**.
6. On the same page, click **Authorize a Google Domain Administrator Account**. Note: You may have to be filtered behind the Cipafilter for this link to work correctly.
7. You will be redirected to Google. If prompted to log in, do so. Note that you must use an account that has administrator access to the Google Apps domain you're configuring. Google will prompt you to allow Cipafilter's OAuth feature to view user and group information the domain; click *Accept*.

After following all of the steps above, you should be redirected back to the Cipafilter's **Content Filtering** page, where the **Google Access Token** field should show *OK*. At this point, you should be able to select any of the Google OAuth-related features you'd like to use — see those options' respective descriptions for more information.

Group Permissions

Permissions for groups of users are managed here. Each group has individual settings for the different filtering technologies available, as well as a separate whitelist and blacklist. On this page, you can also edit the global whitelist and blacklist which can apply to any or all groups at once.

Groups

Filtering and blacklist configuration is performed according to groups. The default group is called `default`. Each Cipafilter additionally comes configured with `nointernet` and `unfiltered` groups, which block all and no sites, respectively. Groups can be managed via the **Group Management** tab.

When a user is authenticated, the LDAP tree is queried to determine if the user is part of any group matching the names of the groups configured here. If so, the first matching group applies to the user. If not, the `default` group applies.

There are also a global blacklist and a global whitelist which may apply to all groups.

Group Configuration

Permissions for individual groups are managed via the **Group Configuration** tab. The group to work with is set using the **Manage group** drop-down at the top of the page.

Technologies

Multiple filtering technologies are available which can be applied to the users in any particular group. These technologies will not be applied to Web sites on the whitelist.

- **Safe Search Enforcement** — This technology detects when a user accesses a popular search engine (as well as certain other sites with search features, such as Flickr) and enables the site's built-in "safe search" feature. This technology will reduce the amount of pornography and other objectionable content returned by these sites. All groups with this feature turned on will be subject to **Enhanced Safe Search**, if enabled.

  Note: This option activates the basic safety filter for YouTube and also applies **Enhanced Safe Search** to YouTube searches. It does not activate the full Restricted Mode — to enforce this, please see the **YouTube Restricted Mode Enforcement** option below.

  Note: For legacy reasons, whitelisting a search engine will not bypass **Safe Search Enforcement** (adding it to the **Super Whitelist** will, however).

- **YouTube Restricted Mode Enforcement** — This technology enforces Restricted Mode (aka Safety Mode) on YouTube. This is a YouTube feature which provides much stricter filtering than the basic safety mode enforced by the **Safe Search Enforcement** technology. For more information on Restricted Mode, see **YouTube Help**.

  This feature is also compatible and highly recommended for use with Google Apps for Education (GAFE) YouTube settings. When a user whose account is on a GAFE domain logs into YouTube, they are subject to the content settings defined by the GAFE domain administrator; however, if the user were to log out, these settings would no longer apply. **YouTube Restricted Mode Enforcement** prevents users from bypassing restrictions using this method by forcing YouTube to apply Restricted Mode to all visitors, even ones who are not loggged in. YouTube will advise visitors to log in with their GAFE account when a video is blocked. For more information on GAFE YouTube settings, see **Google Apps Administrator Help**.

- **Send *Prefer: safe*** — `Prefer: safe` is an HTTP header which communicates to Web sites that the client is requesting only non-objectionable ("safe") content. Sites that support this header should, upon detecting it, enable any safe-search or parental-control features they support. This functionality is very similar in concept to **Safe Search Enforcement**, but, since it is implemented by the sites themselves, it does not require Cipafilter firmware updates to stay current.

  The header is enabled by default for existing groups with **Safe Search Enforcement** or **Pornography Detection** turned on, but it is an independent feature and can be used in any combination with the other group technologies.

  Note: As of 2015, the `Prefer: safe` header is a very recent standard, and most Web sites do not support it. The most notable site that does support the feature is Bing.

- **Proxy/Anonymizer Detection** — This technology is designed to detect and block Web-based proxy services. This system uses fingerprints of popular Web sites along with a built-in pre-compiled blacklist of known proxy servers to offer a high success rate in eliminating this problem. Proxies detected by this system are shared by the rest of the Cipafilters in the cloud, which enables quick discovery of new proxy servers that are made available on the Internet.

  Please note that some Web sites which offer proxy-like functionality may be specifically whitelisted from proxy detection in order to make the automatic blacklists under **Filter Circumvention** more useful. To block these additional sites, please refer to the respective automatic blacklist options.

-

Pornography Detection — Cipafilter's content-aware **Pornography Detection** feature analyzes the contents of Web pages and dynamically ascertains whether they are pornographic, primarily by measuring the presence or absence of certain words on the page. Each time a Web site is blocked, the administrator may be e-mailed, and the block will be recorded in the Web monitoring system (if active).

To filter sites which may not be detected as pornographic through their content alone (including very lightly moderated sites which are filled with user-generated content), this technology additionally employs a pre-compiled blacklist of Web sites which are known to contain pornography.

- **Lingerie/Undergarment Detection** — The **Lingerie/Undergarment Detection** technology is an extension of **Pornography Detection** which blocks pages containing references to women's undergarments and lingerie. Please note that there is currently no pre-compiled list of lingerie sites, so pages pertaining to lingerie which are not detectable with this technology (usually photo galleries) must be blacklisted manually.
- **Extreme Language Detection** — The **Extreme Language Detection** technology is another extension of **Pornography Detection** which blocks pages containing offensive language. This feature is designed for organizations with a zero-tolerance language policy; as such, it is quite strict in its filtering — a single instance of profanity will trigger a block. At the same time, however, the set of filtered words is quite small; specifically, it is limited to the most severe profanities (those which would not be permitted on, for example, broadcast television).
- **Games Detection** — This is another content-detection technology which blocks pages related to the playing or discussion of games. The primary block target of this feature is browser-based games, but potentially many related subjects (including PC and console games, card games, board games, and occasionally sports) may also be affected. Unlike **Pornography Detection**, this technology does not currently incorporate a pre-compiled blacklist — it is designed to be used instead of or in addition to the gaming-related automatic blacklists described in the section below.
- **Download Blocking** — This feature prevents users from downloading restricted files. Files that are installable or executable, or may otherwise contain viruses, spyware, or trojans, and are not required by the average user on a daily basis, are considered restricted. The following extensions are currently filtered by this feature:

  `.7z`, `.bat`, `.bin`, `.cab`, `.com`, `.cpl`, `.crx.data`, `.dll`, `.dmg`, `.exe`, `.iso`, `.mar`, `.mpkg`, `.msi`, `.ocx`, `.pkg`, `.rar`, `.scr`, `.sit`, `.tar`, `.tgz`, `.vbs`, `.wsf`, `.xpi`, and `.zip`

  Note: It is not possible to selectively enable or disable individual extensions.
- **YouTube for Schools** — This feature allows users in the selected group to access YouTube according to the organization's **YouTube for Schools** policy. **YouTube for Schools** can be configured via the **Advanced Configuration** tab of the **Content Filtering** page.
- **Google Apps Domain Restriction** — **Google Apps Domain Restriction** allows users in the selected group to access Google properties using only accounts under those domains specified under **Google Apps Domain Restriction** on the **Advanced Configuration** tab of the **Content Filtering** page. This feature requires `google.com` to be subject to decryption.
- **Device Authorization** — This feature allows users in the selected group to authenticate arbitrary

devices for network access via the captive portal. Members of groups which do not have this option enabled will receive a "not authorized" message when they try to authenticate.

- **Override Portal Time-out** — This option allows for per-group portal authentication time-out values. For example, a group for students may have a 1-hour time-out (forcing them to log back in after the hour has elapsed), while a group for teachers might have a 12-hour time-out. If not otherwise set, the default value (as set on the **Content Filtering** page) will be used.
- **Temporary Whitelist Management** — When this feature is enabled, users in the selected group will be able to access the **Whitelist Management** feature by clicking the Add to Whitelist link on the filter block page and logging in with their credentials.

Automatic Blacklists

Automatic Blacklists are lists of Web sites compiled by Cipafilter and are updated twice daily from our corporate office. For ease of use, these blacklists are organized into categories; each category may have any number of constituent lists enabled. To toggle the display of lists, use the disclosure triangles next to the category names, or the EXPAND ALL and COLLAPSE ALL all buttons at the top.

- **Gaming and Gambling** — Sites devoted to playing, downloading, or discussing non-educational games.
    - **Gambling** — Sites which encourage gambling, such as betting sites and online casinos.
    - **Games** — Sites related to computer and electronic games, particularly sites where users can play games. This blacklist also includes game download sites, game review sites, and Web sites devoted to the discussion of games. This category does not include games sites devoted to educational use only.
    - **Games (DMOZ Enhanced)** — Additional game sites listed in the Open Directory Project's Games category.
- **Crime, Violence, and Ethical Issues** — Sites related to violence or questionable activity.
    - **Hate** — Sites which promote hatred or bigotry via the use of vulgarity, slurs, calls for violence, and other clearly hate-filled language. This does not include sites whose content is mostly political in nature, but does include those with a great deal of bigoted user content. The line between these types of sites is somewhat controversial, but this selection is designed to be as objective and legally defensible as possible.
    - **Violence** — Sites related to gore, street fighting, and other violent imagery or activity (death videos, shock sites, backyard wrestling, etc.).
    - **Combat Sports** — Sites related to professional and authoritatively sanctioned martial arts and fighting sports (including boxing, wrestling, and MMA).
    - **Weapons** — Sites dedicated to the discussion or sale of guns, explosives, and other weapons (potentially including sport- and hunting-related sites).
    - **Drugs and Alcohol** — Sites promoting the use, manufacture, or sale of illegal substances, tobacco, and alcohol, as well as sites related to the intentional abuse of legal drugs.
    - **Academic Cheating** — Sites devoted to plagiarism, essay writing, and similar academic dishonesty.
    - **Piracy** — Sites devoted to copyright infringement or piracy of software, music, videos, and other intellectual property.
- **Media** — Sites related to online multimedia, including images, videos, and music.
    - **Images** — Sites dedicated to sharing, viewing, or uploading images (including photo sharing sites, "meme" images, etc.). This list does not include image sites handled by the **Safe Search Enforcement** technology.
    - **Video** — Sites dedicated to sharing, viewing, or uploading videos (including Web cam galleries, online television services, etc.). This list does not include sites that are primarily news-oriented.
    - **Music/Radio** — Sites that offer online music/radio streaming, such as Pandora, Last.fm, SHOUTcast, and Live365, as well as terrestrial radio broadcasting sites. Also included are services such as iTunes, Napster, Rhapsody, and SoundCloud.
- **Commerce** — Sites related to online purchases.

    - **Shopping** — Sites dedicated to online shopping and auctions, such as Amazon and eBay.
    - **Travel Purchases** — Sites dedicated to travel-related purchases and exchanges, including plane/bus/train tickets, hotel reservations, car rentals, and "couch-surfing".
- **Social Networking** — Sites dedicated to personal profiles, messaging, blogs, and other discussion.
    - **Blogs** — Sites dedicated to blogging, including both popular individual blogs and general blog-hosting services (such as Tumblr and WordPress).
    - **Dating and Personals** — Sites related to dating, marriage, and sexual encounters.
    - **Chat** — Sites that provide chat, instant messaging, and texting services.
    - **Social Media/Networking** — Sites providing any other social components, including online communities, networking/sharing sites, and social news sites. Also included are sites which fall under another list but contain secondary social components, such as Last.fm and Flickr.
- **Filter Circumvention** — Sites which may allow users to circumvent the Web filter. The lists in this category are particularly complementary to the **Proxy/Anonymizer Detection** technology.
    - **Remote Access** — Sites related to remote-control and screen-sharing tools, such as LogMeIn and TeamViewer, as well as downloadable software such as VNC clients.
    - **VPN** — Sites related to VPN and tunnelling services, including Hamachi and Tor, as well as downloadable VPN clients.
    - **Web Translation** — Language translation services such as Google Translate. These sites are not designed for circumvention, but a side effect of the way they work is that they can be used to bypass content filters.
    - **Alternative Search Engines** — Search-engine sites that may bypass the Safe Search Enforcement feature. This includes sites with no safe search to enforce at all and sites which simply aren't supported yet, as well as alternative domains for major search engines (such as google.de and yahoo.co.jp). The list explicitly excludes those sites which *are* affected by Safe Search Enforcement, including the US English versions of Google, Bing, Yahoo!, and DuckDuckGo.
- **Web Mail** — Sites which offer Web-based e-mail services, such as Gmail and Yahoo! Mail.
- **Advertising** — Sites providing advertising content and tracking services. Please note that URL blacklisting is not a completely effective means of blocking ad content (browser-based ad-blocking extensions use more advanced detection methods); however, this list can be useful for blocking many of the more common advertisement sources. With this list enabled, users may often find filter block pages embedded in unblocked sites.

Manual Lists

The whitelist/blacklist system allows you to control Web access by using a basic domain-oriented syntax or a sophisticated regular-expression URL-matching technology to either allow or reject Web sites based on their URL.

Whitelist entries always override blacklist entries. Therefore, allowing a single sub-domain while blocking the rest of the site can be done by whitelisting `subdomain.domain.com` and blacklisting `domain.com`.

To apply **Global Lists** to the selected group, check **Apply Global Lists to this group**.

For information on the syntax of list entries, see the **Entry syntax** section below.

SSL Decryption Settings

After generating a root certificate via the **Content Filtering** page, SSL decryption functionality can be configured on a per-group basis via this section.

Each group can be configured for one of four decryption behaviors:

- **Never** — HTTPS connections made by members of this group will never be decrypted. This option prevents the filter from applying advanced technologies, such as pornography detection, to secure Web sites accessed by this group.
- **For only the following domains** — HTTPS connections made by members of this group will not be decrypted, except for connections to domains listed in the text field below. This option can be thought of as a decryption whitelist, enabling administrators to limit decryption and its related features to a small number of specific domains.
- **For all but the following domains** — This option is the reverse of the above. HTTPS connections made by members of this group will be decrypted, except for connections to domains listed in the text field below. This option can be thought of as a decryption blacklist, enabling administrators to bypass decryption of problematic domains (such as those that use certificate pinning).
- **Always** — All HTTPS connections made by members of this group will be decrypted, and any appropriate lists or technologies will be applied according to the data within the connection.

Decryption exemptions may be provided in the form of a URL or `REGEX:` entry — however, all entries **must** match against a request to the root of the domain (for example, the URL entry `http://www.domain.com/` will behave as expected, but `http://www.domain.com/page.html` will have no effect, because the content filter is not able to see the full URL at the time of decision-making).

Please note that the **Super Whitelist** bypasses SSL decryption in all cases, regardless of the settings on this page.

Group Management

The **Group Management** tab allows for the addition, removal, and renaming of groups. Groups can be re-ordered by dragging and dropping the left side of the entry. The copy button on the far right of the entry area will "clone" the specified group, preserving all of its permissions and list entries.

Global Configuration

The **Global Configuration** tab contains configuration options which are not group-specific.

Super Whitelist

The **Super Whitelist** is a special whitelist that applies to all groups and bypasses all filtering operations. By default this list is automatically updated from Cipafilter's corporate office, but this option can be disabled. Administrators can also specify their own **Super Whitelist** entries, either in place of or in addition to the automatic ones.

This feature is designed primarily to allow software updates from trusted sources, such as Microsoft and Apple, to pass through without causing heavy load to the Cipafilter or being slowed down by the anti-virus system. This feature can also be used to prevent issues with Web sites which are incompatible with the **SSL decryption** function.

The **SNI Super Whitelist** works just like the normal one, except that it only has an effect when the client supports SNI (or when SNI isn't relevant, as with unsecured HTTP requests and requests using browser proxy settings). This is useful to work around issues that might occur when clients that don't support SNI (such as older mobile devices, Internet Explorer on Windows XP, and others) attempt to access a site with many different domain names. Without the **SNI Super Whitelist**, these clients might be able to access domains they shouldn't, because the content filter is unable to know for certain what site they are trying to access.

An additional component to the **SNI Super Whitelist** is the **Google Chromebook compatibility** list, which activates **SNI Super Whitelist** entries designed to enhance the filter's compatibility with Google Chromebooks. Please note that, due to limitations imposed by Google, this option is fundamentally incompatible with Google Apps domain restriction.

A **Google OAuth compatibility** list is also provided; however, most customers will not need to enable this feature manually, as the OAuth list additions are automatically enabled when the Google portal front-end is in use.

Domains and URLs placed on either **Super Whitelist** will bypass nearly **all** functionality of the filter, including virus scanning, SSL decryption, authentication, and portal interception. It is important therefore to be careful when modifying these entries, as even a small misconfiguration can open a large hole in the filter. Please consult with tech support if you have any questions about this list.

Enhanced Safe Search

When the **Safe Search Enforcement** technology is enabled, the Cipafilter will enforce the safe search option on major search sites such as Google and Bing. However, because different search providers have different concepts of "safe search", there may exist gaps in their filtering capabilities. In particular, most search engines only catch explicit pornography terms.

The **Enhanced Safe Search** (ESS) feature (and its image-only sub-component, **Enhanced Safe Image Search** (ESIS) is designed to address this problem — by entering key words here, an administrator can designate additional terms which will trigger a block when used in a search. A number of common terms are provided by default in the automatic lists.

The entries on the **All Searches** (ESS) tab apply to all types of searches (with very few exceptions, such as map directions) — text, news, videos, images, etc. The **Image Searches** (ESIS) entries apply *only* to image searches. This distinction is maintained because some terms which may be inappropriate for image searches are valid for other types; for example, schools may wish to limit image searches for the word *breast* without affecting textual searches for terms like *breast cancer*.

ESS key words must be literal strings (no regular expressions or wildcards are supported) of one or more words to be blocked. Each key word will be looked for in the URL when a search is detected, and, if it appears to exist as a search term (not part of the actual domain or path), the request will be blocked. Note that key words are matched only against the URL and only as whole words, so the feature may not catch searches for auto-corrected misspellings, plurals, etc. For example, the key word `breast` will catch searches for `breast` and `breast cancer`, but not `breasts` or `braest`. It is recommended that plurals and common misspellings be entered as separate key words, if it is desired to catch them.

All ESS (and ESIS) entries apply to all groups which have **Safe Search Enforcement** turned on.

Note: For legacy reasons, whitelisting a search engine will not bypass **Enhanced Safe Search** (adding it to the **Super Whitelist** will, however).

Global Lists

The **Global Whitelist** and **Global Blacklist** work the same as their corresponding manual lists (described above), except that they apply to all groups which have the **Global Lists** option enabled. This feature allows an administrator to define one entry that applies to all groups, without having to edit each group manually.

As previously mentioned, whitelist entries always override any overlapping blacklist entries. Therefore, one can add a site to the **Global Blacklist**, and then allow it for a single group by also adding it to that group's **Manual Whitelist**.

For information on the syntax of list entries, see the **Entry syntax** section below.

Automatic Blacklist Exemptions

Administrators occasionally wish to exclude particular sites from the **Automatic Blacklists**, but continue to apply content detection to those sites. For instance, an organization may want to allow students access to non-pornographic Tumblr blogs. In this case, the domain can be added as an **Automatic Blacklist Exemption** here.

Unlike the other list fields, neither full URLs nor `REGEX:` expressions are supported here — only simple domains and sub-domains. Any entries added here will be removed from all **Automatic Blacklists** for all groups; manual lists will be unaffected.

Temporary Whitelist

Options for the temporary whitelist system (Whitelist Management) are set via the **Temporary Whitelist** tab.

When a user logs into Whitelist Management, they are provided a list of time durations (15 minutes, 1 hour, etc.) for which the site should be whitelisted. The table on this tab allows the administrator to configure the available durations to their precise requirements; durations may be added/removed, disabled, and set as the default. A duration is defined as a time period specified in `hh:mm` format; for example, `01:00` is 1 hour. The minimum duration is 1 minute (`00:01`); the maximum is quite high to accomodate special needs, but it is not recommended to use Whitelist Management as a semi-permanent custom whitelist by specifying very high values.

Note: A list of common durations is provided by default. If at any point the durations table is cleared of all entries, it will re-populate with these default durations.

Current Entries

This section displays any temporary whitelist entries that are currently active. The Manage Temporary Whitelist Entries link allows an administrative user to view the history of temporary whitelist entries and remove active items.

List entry syntax

This section describes the syntax used for adding entries to the manual and global lists.

Any line beginning with a `#` (hash) is a comment and will not be treated as a list entry. Comments appear in the *Comment* field on the filter-reject page and are applied to all following entries (until the next comment). Empty lines and lines containing only white-space are also ignored as non-entries.

To block an entire Web site, simply enter its domain. For example, `google.com` will block everything at `google.com`, `www.google.com`, `images.google.com`, etc. It is also possible to block individual sub-domains; for example, `groups.google.com`.

To block a single page or directory on a Web site, enter the URL up to the point at which the filter should stop matching. For example, to block all pages under `http://www.domain.com/directory`, simply enter that into the list. The list parsing is intelligent enough to handle both complete URLs as well as partial URLs (such as `www.domain.com/directory`).

For ease of use the sub-domain `www` will be stripped from "simple" list entries, leaving the top-level domain (e.g., the entry `www.google.com` will be interpreted the same as `google.com`). If it is necessary to block the `www` sub-domain specifically, the advanced syntax may be used.

Specific parts of a Web site or even ranges of sites can be blocked by using a regular-expression entry. These entries are somewhat more complicated, but also much more powerful. Two different styles of regular-expression entry are supported: the simplified `REGEX:` style and the more advanced `PCRE:` style; both make use of the **Perl Compatible Regular Expressions** (PCRE) engine and its pattern syntax. Equivalent examples of each entry style are provided below:

Example	Description
`REGEX:domain.com:foo.*bar`	The simplified `REGEX:` style takes the form of three colon-separated fields: the `REGEX` entry prefix, the host or domain name, and the PCRE pattern. All three fields are case-insensitive.
`PCRE:domain.com:/foo.*bar/i`	The advanced `PCRE:` style takes the form of three colon-separated fields: the `PCRE` entry prefix, the host or domain name, and the PCRE pattern in a Perl-style delimited format. The pattern may optionally be followed by any combination of modifiers representing flags supported by the PCRE engine. Unlike the simplified syntax, the pattern in this type of entry is **not** case-insensitive unless the `i` modifier is supplied, as in this example.

`PCRE:`-style entries support delimited patterns similar to those used by Perl and PHP. Any non-alphanumeric, non-white-space, non-backslash character may be used as a delimiter. The Perl match operator `m` is supported but not required. Examples of valid patterns include: `/foo/`, `%foo%`, `m<foo>`. Valid modifiers are `imsuxADJUX`, of which only `imsU` are currently supported. The `g` modifier has no effect on match patterns and is silently dropped. An error will appear in the content-filter log if an invalid or unsupported modifier is used, but the entry (minus the bad modifier) will be accepted anyway.

Clearly, most users will prefer the `REGEX:` style, since its syntax is far simpler and more forgiving. In either case, the specified pattern (the third field) is matched against the full URL of each request to the specified host (the second field). For example, the entry `REGEX:youtube.com:watch` will match any URL containing the text `watch` on any `youtube.com` Web site.

For performance reasons, pattern matching is performed against "normalized" domains. As an example, the normalized domain for the entry `REGEX:m.youtube.com:` is `youtube.com`; therefore, the entry will be matched against any YouTube sub-domain, not just `m.youtube.com`. Alternatively, a a wildcard (`*`, or asterisk) can be used to apply a match to all domains (e.g., `REGEX:*:porn` will match any URL containing the text `porn` on any Web site). Please note, however, that matching an entry against all domains does incur a performance penalty. The extent of this penalty depends on several factors, but on filters with many clients or many global wildcard entries, the effect can be quite significant. For this reason, entries of this type should be considered a last resort.

The syntax of PCRE patterns is described fully in the **PCRE documentation**; however, the following can be used as a quick reference:

- `^` and `$` match the beginning and end of a URL, respectively
- `(` and `)` treat a series of characters as a single group
- `*` matches 0 or more of the preceding group/character
- `+` matches 1 or more of the preceding group/character
- `?` matches exactly 0 or 1 of the preceding group/character
- `.` matches any single character
- `[^/]` matches any single character except for `/`
- `\d` matches any single digit (`0–9`)
- `\w` matches any single word character (`a–z`, `0–9`, and _)
- `\b` matches the start or end of a word (the boundary between a word character and a non-word character)
- `\` can be used in front of any special character to treat it literally

Blacklist examples

Example	Description
`youtube.com`	This is a basic domain entry; it tells the content filter to blacklist or whitelist all pages on all Web sites which are part of the `youtube.com` domain. This would not only match video pages, but also, for example, `accounts.youtube.com` and `m.youtube.com`.
`mail.google.com`	This is a basic sub-domain entry; it tells the content filter to blacklist or whitelist all pages on all Web sites which are part of the `mail.google.com` domain. This would also match sub-domains further down; for example, it would affect `chatenabled.mail.google.com`. It would **not** match any other Google sub-domain — for example, `images.google.com` would be unaffected.
`# Social networking` `reddit.com`	This is another basic domain entry; this time it is preceded by a comment. The `# Social networking` line will not be interpreted as a list entry; however, it will appear in the *Comment* field on the filter-reject page. This is useful for explaining why a page has been blocked; it can also be used to (for example) give the name of the person who added the entry and/or the date they added it.
`REGEX:*:porn`	The `*` after the first colon indicates that this rule is a "wildcard" entry — the content filter will try to match the pattern against every URL request that passes through it. As mentioned above, this does incur a certain performance hit, so it is important to use this type of rule only when absolutely necessary. The `porn` at the end indicates that the entry should match if the text `porn` is found anywhere in the URL. (Note that this entry will also match the word `anti-pornography`, for example, since it still contains `porn`.)
`REGEX:*:^https?://[^/]+\.edu[:/]`	The `*` after the first colon indicates that this rule is a "wildcard" entry — the content filter will try to match the expression on every Web site that passes through it. Once again, this does incur a certain performance hit.

Example	Description
	The `^` after the second colon is an anchor that means the expression should only be matched at the very beginning of the URL (without this, the expression would match anywhere). `https?://` matches `http://` or `https://` (the `?` means "zero or one of the preceding character" — in this case, the preceding character is an `s`). `[^/]+` matches one or more (`+`) of any character that is not a slash (`[^/]`). Matching only non-slash characters ensures that we only look at the first part of the URL (the domain). `\.edu` matches the text `.edu`. The backslash is necessary because, in regular expressions, a dot by itself means "any character." Finally, `[:/]` matches either a `:` or a `/`. This is useful to ensure that the pattern matches only at the very end of the domain name (otherwise, it might match a domain like `example.education.com`. This rule would blacklist or whitelist all `.edu` Web sites (`harvard.edu`, `mit.edu`, and so on).
`REGEX:reddit.com:\b(cat|dog)s?\b;`	The `reddit.com` after the first colon indicates that this rule should be matched only against Web sites under the `reddit.com` domain. Therefore, this rule would affect `www.reddit.com`, `ssl.reddit.com`, and so on. The `\b(cat|dog)s?\b` at the end indicates that the entry should match if any of the following whole words appear anywhere in the URL: `cat`, `cats`, `dog`, `dogs`. (`\b` matches the start or end of a whole word; the `(x|y)` structure means "either `x` or `y`"; and the `s?` means that the letter `s` may or may not appear.) Because of the "whole word" requirement, this rule would not match, for example, the words `vacation` or `bulldog`. However, it would still match `dog-catcher`, since the hyphen makes two separate words.

Permissions Scheduling

This page allows you to configure certain groups or subnets to be mapped to a different group at specific times of day.

If a subnet is mapped to a group, all users on that subnet will be treated as members of the group during the times of day specified in the rule.

If a group is mapped to another group, all users in the first group will be treated as members of the second group during the times of day specified by the rule.

Selecting the box under **Invert** allows you to invert (or reverse) the times of day the rule is active. For example, you can select the hours school is in session with a rule and then select **Invert** in order to make a rule that will be in effect when school is out of session.

Bandwidth Control

Bandwidth Control allows you to keep heavy bandwidth users from consuming your bandwidth capacity to the detriment of other activities on your network. Bandwidth can be controlled by IP or subnet and is allocated in kilobits per second (kbps). Please be advised that if a user is using their entire bandwidth allotment, new packets to them will be delayed or dropped in order to limit them to their prescribed cap.

Within each IP or subnet's bandwidth allocation, ICMP, TOS `0x10` (ECN), and TCP ACK packets are prioritized to facilitate performance of real-time protocols and pings and to eliminate wasteful packet re-transmissions due to delayed acknowledgments.

IP addresses and subnets not listed on this page are all lumped together in a single queue which is treated the same way as above, but is prioritized as a whole behind everything else. Therefore, IP addresses and subnets listed on this page will have their traffic served first, before any traffic from IP addresses or subnets not listed here.

When bandwidth is limited, all machines are first allocated their committed information rate (**CIR**). After all CIRs have been met, the remaining bandwidth is divided between IP addresses in ratio with their CIRs — but no machine is allowed to exceed its maximum allocation.

Note: It is currently not possible to limit the upload on a subnet or IP which is subject to NAT. This limitation will be addressed in a future version of the product.

Internet Reports

The Web-usage reporting system is activated on the **Content Filtering** page. After it is activated, it collects information on every URL visited by your Internet users. It will track each user by user name and password if Web authentication is activated; otherwise it can only track users by IP address.

Status

The status page provides a nearly real-time view of the Web traffic moving through the Cipafilter. Filtered traffic statistics are displayed at the top of the screen, while notable events appear below.

To the top right, a number of textual statistics are displayed, including **Web Clients** (a daily total of distinct IP addresses which have been filtered), **Web Users** (a daily total of authenticated users who have been filtered), and **Client Installs** (a cumulative total of distinct IP addresses which have had the Cipafilter client software installed).

Internet Reports (beta)

A new Internet Reports system is currently in beta. This system features an improved interface, better filtering, new data, and several other additions. However, not all functionality from the legacy reporting system has yet been integrated.

Bandwidth Reports

Clicking on **Bandwidth Reports** launches the BandwidthD application. This software tracks the Internet usage of all clients on your network. Select an interface under **Select A Sensor** and click GO to request a report. Custom reports and graphs can be requested by calling special URLs on the Cipafilter. Please consult with tech support if you wish to embed these reports into other Web consoles.

Live Bandwidth Reports

Live Bandwidth Reports is an enhanced version of the earlier bandwidth-reporting system which employs a friendlier interface with a number of new options. In addition to displaying data using several different chart formats, it is also possible to view a near-real-time indication of the Cipafilter's total bandwidth utilization.

The first time this feature is accessed, you will be prompted to enter a maximum upload and download speed in kbps. These values should be set in a manner that corresponds to the bandwidth capacity available on the unit's primary Internet connection. Setting these values calibrates the meters and charts displayed in each reporting view so that the numbers reflect the total bandwidth capacity. For example, the ratio of bandwidth that is available versus bandwidth that is in use is calculated based on these values. (Providing incorrect numbers here may therefore result in inaccurate calculation of utilization ratios, but it should not affect the "raw" bandwidth data.)

Notifications

Informational notices from the Cipafilter, including content-filter trips and mail errors, are sent to the address(es) specified under **E-mail Addresses**. The Cipafilter can send immediate, daily, weekly, and custom reports to each of the specified address(es). Each address has its own separate report settings.

Custom reports take the form of complex SQL queries — please contact tech support for assistance with creating these.

E-mail Configuration

Cipafilter provides several e-mail security and content-filtering features, including a robust anti-spam system. The filter acts as a proxy and firewall to protect your mail server from the Internet; this allows you to take full advantage of your mail server's advanced features without the risk of having it accessible to hackers and worms. We recommend placing your existing server behind the Cipafilter and configuring it with only a private IP address; then, change your `MX` records to point to the Cipafilter, and configure the filter to route e-mail for your domains to the private IP of your actual mail server. This will cause all mail to be delievered to the Cipafilter, where it will be virus-scanned, spam-filtered, content-filtered, and then forwarded on to your actual mail server for processing.

Because of the problems with spam in recent years, many companies have become more cautious about the mail servers with which they communicate. Many unofficial rules have been adopted in a piecemeal fashion across the Internet. If your mail server is configured incorrectly, you may find that most people will receive your messages but for some they arrive marked as spam — and for some they don't arrive at all. Please allow us to help you through setting up your mail server to be compliant with all official and unofficial guidelines for message processing.

However, if you are an expert, and wish to configure the server yourself, please be certain to comply with the following guidelines. If you don't understand the reasoning behind any of the following, our tech support representatives will be happy to go over it with you.

- Always make sure the IP address pointed to by your `MX` record reverses to the same name that is contained in the `MX` record.
- If you are using Cipafilter for anti-spam, do not use secondary `MX`s.
- Make sure that the Cipafilter host and domain name match the name from your `MX` record.

General Configuration

E-Mail Content Filtering

The **E-Mail Content Filtering** option applies the Web content filtering technology to incoming and outgoing mail messages. If a message is inappropriate, an e-mail is sent to the person in your organization who either was sending it or was the intended recipient, and the message is blocked. This is useful for filtering out pornographic spam.

E-mail Archive

The **E-mail Archive** option can be set to one of three settings:

- **Rcpt: cipafilter_email_archive** — This setting is designed to work with e-mail journaling support in products like Microsoft Exchange. Many mail-server products support journaling, but some, like GroupWise, require third-party software. If your mail server supports journaling, set the journaling address to `cipafilter_email_archive@x.x.x.x`, where `x.x.x.x` represents the Cipafilter's internal IP address.
- **All Messages Passing Through** — If your mail server does not support journaling, you can still archive all messages that pass through the Cipafilter using this setting.
- **None** — This option disables the **E-mail Archive** functionality.

Archived messages can be browsed via the **E-mail Archive** page. If you are archiving mail for retention purposes, be sure to speak with tech support about creating a suitable backup and recovery plan.

E-mail Footers

The Cipafilter can automatically append a message footer to each e-mail passing through it. Simply compose your desired footer and upload it through the Web interface. Footers can be selected for plain-text messages as well as HTML-formatted mail.

MailRoutes

To configure the Cipafilter to proxy e-mail for an existing server, add a route for your domain pointing to your mail server's private IP address in the **Mail Routes** table.

Subnets Authorized for SMTP Relay

Only e-mail clients using IP addresses matching IPs or subnets listed here can use the Cipafilter as an outgoing mail server. Addresses may be entered as bare IPs, **CIDR-style subnets**, or `sendmail`-style ranges.

Anti-Spam Configuration

Anti-Spam Sensitivity

The **Anti-Spam Sensitivity** option controls how aggressively the filter's automated anti-spam system will score suspicious e-mail attributes. Different sensitivity levels may be added from time to time; higher sensitivity levels will block more spam, but they may also increase the risk of blocking legitimate mail.

Spam Forwarding

The Cipafilter can redirect all spam with a score above a certain threshold to a spam mailbox instead of delivering this mail to the intended recipient. To use this feature, select the desired **Spam Forwarding Level** and then enter the spam mailbox address as the **Spam Forwarding Address**.

Anti-Spam Whitelist

E-mail to and from servers on this list will not be intercepted by the anti-spam system. Enter the server IP, domain name, or subnet here if you have difficulty receiving e-mail from another party. Cipafilter's anti-spam system works with all standards-compliant mail servers; however, some older and custom systems may have problems.

Note: The **Anti-Spam Whitelist** does not bypass virus scanning for e-mail items.

Anti-Spam Blacklist

All e-mail from the specified domains, subnets, and e-mail addresses will be rejected with a `550` error.

Custom Anti-Spam Rules

While Cipafilter's anti-spam detection system is designed to function without manual user configuration, some organizations may find that their individual circumstances require a more active approach. For those customers, the **Custom Anti-Spam Rules** table can be used to define scoring rules for the anti-spam system.

Each custom rule will trigger a search of the message attribute selected under the **Match Against** column for the text or expression defined in the **Match Expression** column. If found, the anti-spam system will increase or decrease the overall score of the message as specified in the **Score** option.

Message attributes

The following message attributes are available from the **Match Against** drop-down:

- **Sender** — Matches the expression against the message's sender (From) names/addresses.
- **Recipient(s)** — Matches the expression against the message's recipient (To and Cc) names/address.
- **Subject** — Matches the expression against the message's subject line.
- **Message Body** — Matches the expression against the full contents of the message body.
- **URL in Body** — Matches the expression against any URLs detected in the message body. Although the normal message body option can be used for matching URLs, this option is simpler and more accurate.

Match expressions

Three **Match Expression** syntaxes are supported:

- **Literal string** — Literal string expressions are, as the name suggests, interpreted literally — that is, whatever text is written in the field is the text that is searched for in the message attribute. Literal string expressions are treated as whole words and are matched case-insensitively. For example, the literal expression `spam` will match `SPAM`, `Anti-Spam`, and `spam@spam.com`, but will not match `spammer`, `antispam`, or `spam1@spam2.com`.

- **Glob (wildcard)** — A glob (aka wildcard) expression is a simple pattern-matching expression that uses asterisks (`*`) to substitute for zero or more characters that may appear in the search subject. For example, the glob expression `foo*bar` will match both `foobar` and `foo baz bar`.

  Custom spam rules support a limited globbing syntax wherein all asterisks in an otherwise literal string are converted to the regular-expression pattern `.*?`. As with literal strings, globs are treated as whole words and matched case-insensitively (so `foo*bar` will also match `FOOBAR`).

- **Perl Compatible Regular Expression (PCRE)** — Cipafilter's underlying anti-spam system uses Perl regular expressions for rule matching; the PCRE entry syntax provides direct user access to these powerful expressions. Cipafilter's `PCRE:` entry syntax for anti-spam rules uses a combination of features from Perl and the PCRE library; the syntax is very similar to those used for **blacklist entries** (just without the host component).

  All PCRE expressions must be prefixed with the string `PCRE:`. Expressions that do not begin with this prefix will be treated as literal strings or globs. After the prefix, a Perl-style delimited match pattern must be supplied. `PCRE:/foo/` is an example that uses the most common pattern syntax (with forward slashes as delimiters). This example would match the lowercase text `foo` anywhere in the search subject. To match case-insensitively, the `i` modifier may be used, as in `PCRE:/foo/i`.

  Please note that, for usability reasons, custom anti-spam rules support only the sub-set of Perl's syntax features that are supported by the PCRE library (which, despite the name, is not fully compatible with Perl). This means that Perl-specific features such as `\L` are unsupported and may behave strangely. The exception to this limitation is that Cipafilter's implementation supports only those modifiers allowed by Perl, not any PCRE-specific ones (such as `U`). These implementation differences will affect only the very most advanced users, however.

  For more information about Perl regular-expression syntax, please see the **Perl Programming Documentation** or (more generally) the Cipafilter documentation on **list entry syntax**.

Scoring

Cipafilter's anti-spam detection system uses a scoring mechanism whereby higher scores indicate more suspicious e-mail. For example, the score 10.0 is very suspicious, while the score -10.0 is very trustworthy. Messages with a final score of 5.0 or greater are classified as spam. By default, scores are assigned according to rules developed by the greater anti-spam community (with some proprietary additions/modifications); these rules are updated frequently and all Cipafilter units receive the latest changes daily.

Custom spam rules, upon match, add or subtract from a message's final score, potentially causing or preventing a spam classification. For simplicity, Cipafilter provides score modifiers of +/- 0.250, 1.000, 3.000, and 999.0. Because a score of only 5.0 is required to classify a message as spam, it does not take very much to affect the final classification; it is preferable, therefore, to use the minimum score modifier (+/-0.250) whenever feasible. Greater score modifiers should be used only when lower ones have been tested and found not to provide the desired effect.

The +/-999.0 scores are so high/low that usually nothing but another +/-999.0 score can counter it. This means that (e.g.) giving a custom rule a +999.0 score will result in a near 100% spam classification (block) rate for any message matching the rule. This score modifier should be used *extremely* sparingly.

Notes about anti-spam

Cipafilter provides a very "hands-off" anti-spam solution which does not rely on intervention from end users or administrators. This differs from many other vendors, whose solutions involve users/administrators manually classifying mail as spam or not spam.

The manual classification approach does have merit; it often results in exceptionally high accuracy, for example, especially in environments with a very large number of users participating in classification. This is the way most Web-mail providers' spam filters work, for instance — the Web interface allows users to easily mark their messages as spam or not spam, and, after this happens to a certain type of message a certain number of times, that information is used to block those messages for all other users of the service.

The primary advantage of the Cipafilter approach — and the Cipafilter product in general — is ease of configuration and use. Administrators do not need to provide a large body of mail to train the filter during set-up, nor must they become mail experts in order to configure policies and rules, nor must they or end users continuously monitor their mail for spam. The disadvantage is that, although the Cipafilter solution does have a very high rate of success, it may never be quite as accurate as a major Web-mail provider or an enterprise product that requires manual user intervention.

Please also be aware that, by design, the Cipafilter product does not consider newsletters, social-networking notifications, legitimate advertising, and other forms of *user-solicited* bulk mail to be spam. The Cipafilter anti-spam system does not and will not ever (intentionally) block these legitimate mailings, however irritating some users may find them. It is recommended that users simply unsubscribe from legitimate bulk mailings they do not wish to receive.

Anti-Spam Statistics

This page provides statistics related to the **Anti-Spam** system. In particular, details about the level of spam activity, SPF validation, greylisting, and SpamAssassin are available.

Notes:

- Statistics currently do not persist across reboots — when the Cipafilter is shut down, the previously compiled data will be erased and new statistics will not be available until a sufficient amount of mail traffic has passed through the unit.
- Greylisting data may not be strictly accurate — the numbers are an approximation based on expected server responses to greylisting.

Internet Reports

The Web-usage reporting system is activated on the **Content Filtering** page. After it is activated, it collects information on every URL visited by your Internet users. It will track each user by user name and password if Web authentication is activated; otherwise it can only track users by IP address.

Status

The status page provides a nearly real-time view of the Web traffic moving through the Cipafilter. Filtered traffic statistics are displayed at the top of the screen, while notable events appear below.

To the top right, a number of textual statistics are displayed, including **Web Clients** (a daily total of distinct IP addresses which have been filtered), **Web Users** (a daily total of authenticated users who have been filtered), and **Client Installs** (a cumulative total of distinct IP addresses which have had the Cipafilter client software installed).

E-mail Archive

This selection opens the window used to view the e-mail archive contained on the Cipafilter. From the **E-mail Archive** interface, messages can be selected by mailbox, `To` address, and `From` address. You can search for specific text in the subject or body of the e-mail messages, and messages can be selected within a specified date range. Selected messages can then be downloaded to a file in the `EML` format.

Client Software

This page contains the latest Cipafilter authentication clients and their options, as well as links to two forms of the EICAR anti-virus test file.

Client Settings

This section contains options that apply to any supported authentication clients connected through the filter. The filter will report these settings to the clients and they will alter their behavior accordingly.

- **Hide icon from tray or menu bar** — By default, the client software for Windows and OS X shows an icon in the tray or menu bar (respectively) indicating the software's presence and providing some additional options and status information. The display of this icon does not pose any particular security risks, but some administrators may wish to disable it anyway.
- **Prompt for root certificate installation** — When a root certificate authority has been generated via the **Content Filtering** page, the Windows and OS X clients will check the local machine for the filter's certificate and prompt to install it if it is not already present. Administrators who are not using or not ready to fully deploy SSL decryption may wish to disable this behavior.

Client Downloads

The authentication client is supplied for both Microsoft Windows (32-bit and 64-bit) and Mac OS X. The Windows client is provided in the form of a silent `MSI` installer and is designed to be automatically distributed. The Mac client is similarly provided as an `MPKG`, which can be silently deployed using Apple Remote Desktop or similar. A client suitable for Chromebooks and Chrome-based browsers is also available in the form of the Cipafilter Authenticator browser extension. If you are considering employing this software in your environment, please consult with tech support for advice on deployment scenarios.

Anti-Virus Test File

The EICAR anti-virus test file is a standard way of testing the effectiveness of your anti-virus system's configuration. It is a harmless text file that the various anti-virus vendors have all agreed to detect as a virus for the purpose of testing.

Please be certain to attempt a test virus transmission through each protocol you intend to protect to ensure proper setup after installation.

Notes:

- HTTP virus scanning requires the Web proxy feature to be activated and functioning in order to operate. HTTP anti-virus will work in both transparent and proxy server modes.
- SMTP anti-virus requires that your `MX` record points to the Cipafilter. Your mail can then be forwarded on to your internal mail server with the **Mail Routes** options on the **Anti-Spam** page.

Config Save/Restore

This page allows you to download a copy of your Cipafilter's current configuration, to upload previously saved configuration data, or to revert to the factory default configuration.

Configuration files represent a snapshot of the Cipafilter's settings (essentially those options which can be changed from the Web interface). This includes all interface settings, firewall rules, portal customizations, group permissions, and mail settings. However, it does **not** include non-configuration information such as logs, reporting data, archived mail, and so on. If you would like to retrieve or restore copies of any such data, please contact tech support.

Please be aware that, although uploaded or reverted configuration data will appear on the Web interface immediately, the settings will (with a few exceptions) not actually take effect until the unit is rebooted or you press SAVE AND APPLY on each page where a change occurred. It is therefore recommended that you reboot the system following an import or factory revert.

Note: Your Cipafilter sends daily configuration back-ups to the Cipafilter Enterprise system, so it is not strictly necessary to maintain your own collection of configuration back-ups.

Warning: Configuration files downloaded from this page are not encrypted — it is therefore possible for anyone with a copy of the file to retrieve sensitive information such as the administrator password, LDAP authentication credentials, and the private key used to sign the portal and decrypted Web sites. Please secure any downloaded configuration data accordingly. (The back-ups sent to the Enterprise system are, however, encrypted prior to transmission.)

From this page it is also possible to update the unit's recovery partition. This copies the state of the unit's flash chip (where the primary copy of the operating system and its settings are maintained) to the recovery partition located on the hard drive. This provides a fall-back in case of a hardware failure or similar problem.

Database Maintenance

This page is where management and backup of the Cipafilter database is performed. From here you can configure your purge windows, examine the latest backup log, download a backup, or configure backups to be pushed to a remote server.

A Cipafilter's database is separate from its configuration data and includes the following:

- **Bandwidth reporting data** — Information about bandwidth utilization (interfaces, IP addresses, user and group IDs, packet counts, protocols, etc.) over time.
- **Client data** — Information about devices using the Cipafilter authentication client.
- **Internet reporting data** — Information about Internet usage (IP addresses, user and group IDs, URLs, searches, blacklist trips, etc.) over time.
- **Mail archive data** — Information about archived mail (addresses, headers, message bodies, etc.).

The database backup and purge process runs automatically every Saturday at 2:30 AM. It can also be run manually at any time by clicking START RUN. Progress can be monitored by refreshing the page and viewing the log. Backup and purge applies only to the database stored locally on the router. The remote database is not affected by these settings in any way. If the remote database server is not a Cipafilter, the customer is responsible for implementing backup and purge.

If restoration of the database after failure is important for your implementation, the following responsibilities are incurred by you (the customer). Please consult with tech support to determine a sensible test and backup procedure that meets your requirements.

- Check this page weekly for anomalies in the backup/purge log.
- Periodically download database backups for use in recovery in the case of a mirror failure.
- Periodically test the backup by restoring a copy to a PostgreSQL server.

Warning: Increasing purge windows will increase load, database size, and backup size.

Backup Push

Under this section, the unit can be configured to upload a copy of the database back-up it creates each week. The back-up file is transferred to a Windows (CIFS/SMB) file share immediately after the back-up process has completed on Saturday morning. Use of this feature is highly recommended, especially for **E-mail Archive** customers, in order to prevent data loss in the case of catastrophic failure of the unit.

The user name for the **Backup Push** function can optionally be specified using `domain\username` syntax. The domain may be mandatory in some configurations.

The TEST SETTINGS button can be used to verify that your back-up settings are valid; the test function will attempt to create and delete a file using the supplied configuration, and then report back the result.

Firmware Updates

Automatic Updates

The Cipafilter can be configured to automatically receive updates to the current firmware version. To activate this functionality, select **Enable Automatic Updates** under this section and click SAVE AND APPLY.

Each time an automatic update is installed (typically between 10 PM and 2 AM), an e-mail notification will be sent to the **Notification Address**.

Note: Automatic updates are not currently enabled on the Cipafilter servers, so these settings will not have an effect in the immediate term. However, it is recommended that you enable the feature anyway, if you are interested in it, as this setting will be respected when automatic updates eventually do become available.

Manual Update

The Cipafilter firmware can be manually updated by simply clicking FIRMWARE UPGRADE under this section. The firmware upgrade process typically takes 2–5 minutes including the time required to restart the Cipafilter.

UPS Configuration

Cipafilter firmware supports enhanced integration with UPSes connected via USB. The **UPS Configuration** page provides configuration and status information related to this feature.

By default, filters connected to a compatible power supply are configured to shut down automatically when the power has been lost. This allows the filter to finish writing out any configuration and database changes and to cleanly unmount any attached file systems. Without this feature, if the power is lost and the battery dies, the filter may experience data loss.

If this feature is somehow incompatible with your hardware set-up, you may disable it by unchecking **Shut down safely when power is lost and a supported UPS is present**.

The **UPS Status** section of this page displays information about any attached and supported UPSes. If no UPS is connected, or if it is incompatible with the Cipafilter firmware, a message will indicate this here.

It may be wise to visit this page after any changes are made to the system's power supply set-up, in order to ensure that the filter has detected the new configuration.

Note: At this time, only locally connected power supplies from major manufacturers are officially supported. If you have a UPS which doesn't function with the filter, please contact technical support for investigation.

Override Console

The **Override Console** is used by Cipafilter tech support to enter custom configuration changes. This page is used to work around problems or implement customizations which are specific to a single customer.

Authentication Tools

This page provides information and tools related to authentication (particularly LDAP authentication). The table at the top of the page lists all records currently recognized by the system's authentication engine. For each record, the IP address, host name, user name, group name, and expiry time are listed, where applicable. Users who are authorized but not actually authenticated are listed with their IP address as the user name. A record may be manually expired (deleted) by clicking its trash icon. The REFRESH ALL button will force the system to reload all records.

Inject/Replace Authentication Record

This feature allows one to manually inject or replace an authentication record. The only required field is the IP address, but a particular duration or user name may be supplied to force these values. Manual specification of the group field is not currently implemented; injected records will be placed into a group according to the user name and/or any applicable subnet rules. It is not possible to inject a record for an IP address that doesn't match a subnet rule.

Query Group Memberships

This tool can be used to test the group membership that would be applied to a particular user/IP combination. Only the user name is required. When no IP address is supplied, the specified credentials will be queried against the LDAP server, and the tool will (where applicable) report the first group result which matches the group configuration on the filter. If a password is supplied, the query will test its validity; an incorrect password result in a failed query. If an IP address is supplied, it will be matched against the filter's subnet rules.

Authentication Cache and Services

During intensive troubleshooting, it may be necessary to clear the filter's LDAP membership cache and/or reload the authentication engine and content filter. Pressing the CLEAR CACHE button will clear the membership cache, forcing the filter to refresh each user's group memberships the next time they make an authentication request. The CLEAR CACHE AND RELOAD SERVICES button will additionally trigger the authentication engine and content filter to reload, which should clear any invalid entries, update any existing ones, and inject any new ones.

Please note that both of these actions may put a temporary but significant load on the filter and any authentication back-ends it is configured for.

Log Viewer

The **Log Viewer** page provides a simple interface for viewing and searching the filter's internal log files. By default, this page displays the contents of `syslog`, the primary logging facility for the system. The drop-down at the top of the page contains a list of other log files, including those for mail, DHCP, content-filtering, authentication, and the portal.

Optionally, a search expression may be provided; in this case, the output will be restricted to lines matching the provided expression. Searching is performed based on the file drop-down selection; selecting the **(all files)** option will search all log files at once, while any other option will limit the search to the specified file.

By default, the search expression is case-insensitive but is otherwise treated as a literal sub-string — that is, the selected file(s) will be searched for the precise sequence of characters entered. To disable the case-insensitivity, check the **Preserve case** option. To perform a regular-expression search instead of a literal one, select **Use regular expressions**. Regular-expression searches use the **extended GNU** regex syntax (see also **POSIX Regular Expressions**).

Because each search is logged in the Web-server access logs, search results may become cluttered with redundant entries. To prevent this, these entries are stripped from the output by default. To disable this feature, select **Don't filter Web-server access logs**.

For performance reasons, log output (both when searching and when viewing) may be limited to the last several-thousand lines returned. Additionally, when performing an all-files search, an error may be returned if there are too many results. In either case, the page will indicate this above the result.

Log entries are printed one per line by default, so horizontal scrolling may be required to view a full line. To force each line to wrap around (eliminating horizontal scrolling), the **Word wrap** option can be checked below the results. This option does not persist across page loads.

Note: Please be aware that the results returned by this tool may be of a technical nature which might require advanced knowledge of the underlying components to interpret. Some proprietary log files were not originally intended for end-user viewing at all. This feature is provided for those administrators who wish to have more advanced self-troubleshooting capabilities; other customers may prefer instead to contact tech support for assistance.

Network Diagnostics

The **Network Diagnostics** page serves as a basic front-end for common network troubleshooting utilities such as `ping` and `traceroute`. These utilities can be used to confirm the filter's Internet connectivity and network configuration:

- The **ping** diagnostic tests the reachability and latency between the filter and another host using ICMP (a layer-3 protocol). Pinging a well-known public site like google.com can confirm Internet access or problems with DNS resolution.
- The **arping** diagnostic tests the reachability and latency between the filter and another host using ARP (a layer-2 protocol). This tool is primarily useful for testing the local network; in particular, it can be used to determine if an IP address is currently in use, check for duplicate IPs, or resolve an IP to a MAC address.
- The **traceroute** diagnostic displays the routing paths taken to reach the specified host. It additionally displays the latency to each "hop" along the route. This tool is useful primarily for confirming routing problems (usually caused by an upstream ISP or firewall).
- The various **dig** diagnostics query a host for the specified DNS records. This is often used to examine records necessary for proper mail function, among other things.
- **tcpdump** is a more advanced diagnostic which provides a capture of the packets flowing across the filter's network interfaces. For example, by providing the filter expression `port 80 or port 443`, one can examine all HTTP traffic going through the filter. The capture file may be opened using an external tool like **Wireshark**.
- **lldpctl** displays LLDP neighbor data. LLDP is a protocol that allows devices to advertise information about their identity and capabilities (MAC addresses, host names, etc.).
- The **emergency support tunnel** feature establishes a reverse tunnel to Cipafilter technical support, allowing techs to access the filter's console and Web interface securely and without having to re-configure customer-side inbound firewall and port-forwarding rules. This feature is normally initiated manually by the customer, but may also be activated remotely by support personnel, if required.

  Note: This tool makes a series of outbound TCP connections to Cipafilter's tunnel servers. Most network configurations will not require any special configuration to allow these outbound connections, but very strictly secured networks may require the opening of ports. The tunnel tool uses the TCP ports `61022` and `61080` for initialization and server status, and one or more random ports in the range `10000–60000` for the subsequent tunnel connections.

Troubleshooting

The **Troubleshooting** page runs diagnostics on the Cipafilter and reports back the results. Each test returns one of the following results:

- **OK** (green): The test returned a passing/positive result — the tested function is behaving as expected.
- **CAUTION** (yellow): The test returned a non-critical failing/negative result. This may indicate a minor problem, although in some cases it is simply a way to draw attention to configuration details which tech support may need to take into account during troubleshooting.
- **WARNING** (red): The test returned a critical failing/negative result. This usually indicates a problem that should be addressed as soon as possible. Accordingly, any tests with a WARNING result will also appear under the **Trouble notices** header at the top of the Web interface. If any such notices appear, please contact tech support.
- **N/A** (gray): The test was not performed, either because an earlier test that it depended on failed or because the test is not applicable to the unit's configuration. If there are no WARNING notices elsewhere on the page, this is usually OK.

In addition to performing the tests whenever the page is accessed from the Web interface, the Cipafilter runs the same tests on an hourly basis to maintain up-to-date results. These results are reported to the Cipafilter Enterprise system, which enables tech support to monitor trending problems.

Further, each unit periodically queries the Cipafilter servers to pull the latest revision of the page. This allows tech support to identify new problems without requiring customers to perform a full update of their Cipafilter units.

Note: Many of the tests on this page are designed with the most typical Cipafilter setups in mind. Accordingly, highly customized installations may return failing results when there is not actually a problem. In this scenario, it is possible to suppress errant warnings — please contact tech support for assistance.

Status

The **Status** page is the "front page" of the Cipafilter Web interface; it provides at-a-glance information about the state of the filter and its hardware, including CPU and memory usage, drive capacities, hardware temperatures, network errors, and more.

Appendix I: Dot notation to CIDR notation translation table

The following table shows how to convert common dot-notation subnet masks to CIDR-notation subnet bits.

As an example: The subnet mask `255.255.0.0`, when applied to `172.16.0.0`, will cover the addresses `172.16.0.0` – `172.16.255.255`. To refer to this same range of addresses in CIDR notation, use `172.16.0.0/16`. To refer to the single IP address `172.16.0.1`, use `172.16.0.1/32`.

Subnet mask	Subnet bits	Number of IP addresses
`255.0.0.0`	`/8`	16,581,375
`255.255.0.0`	`/16`	65,025
`255.255.255.0`	`/24`	256
`255.255.255.128`	`/25`	128
`255.255.255.192`	`/26`	64
`255.255.255.224`	`/27`	32
`255.255.255.240`	`/28`	16
`255.255.255.248`	`/29`	8
`255.255.255.252`	`/30`	4
`255.255.255.254`	`/31`	2
`255.255.255.255`	`/32`	1

Appendix II: POSIX Regular Expressions

The following is an excerpt from the documentation for Henry Spencer's `regex` library:

> Regular expressions (“RE”s), as defined in POSIX.2, come in two forms: modern REs (roughly those of egrep; POSIX.2 calls these “extended” REs) and obsolete REs (roughly those of ed(1); POSIX.2 “basic” REs). Obsolete REs mostly exist for backward compatibility in some old programs; they will be discussed at the end. POSIX.2 leaves some aspects of RE syntax and semantics open; ‘(!)’ marks decisions on these aspects that may not be fully portable to other POSIX.2 implementations.
>
> A (modern) RE is one(!) or more non-empty(!) branches, separated by ‘|’. It matches anything that matches one of the branches.
>
> A branch is one(!) or more pieces, concatenated. It matches a match for the first, followed by a match for the second, etc.

A piece is an atom possibly followed by a single(!) '*', '+', '?', or bound. An atom followed by '*' matches a sequence of 0 or more matches of the atom. An atom followed by '+' matches a sequence of 1 or more matches of the atom. An atom followed by '?' matches a sequence of 0 or 1 matches of the atom.

A bound is '{' followed by an unsigned decimal integer, possibly followed by ',' possibly followed by another unsigned decimal integer, always followed by '}'. The integers must lie between 0 and RE_DUP_MAX (255(!)) inclusive, and if there are two of them, the first may not exceed the second. An atom followed by a bound containing one integer i and no comma matches a sequence of exactly i matches of the atom. An atom followed by a bound containing one integer i and a comma matches a sequence of i or more matches of the atom. An atom followed by a bound containing two integers i and j matches a sequence of i through j (inclusive) matches of the atom.

An atom is a regular expression enclosed in '()' (matching a match for the regular expression), an empty set of '()' (matching the null string)(!), a bracket expression (see below), '.' (matching any single character), '^' (matching the null string at the beginning of a line), '$' (matching the null string at the end of a line), a '\' followed by one of the characters '^.[$()|*+?{\' (matching that character taken as an ordinary character), a '\' followed by any other character(!) (matching that character taken as an ordinary character, as if the '\' had not been present(!)), or a single character with no other significance (matching that character). A '{' followed by a character other than a digit is an ordinary character, not the beginning of a bound(!). It is illegal to end an RE with '\'.

A bracket expression is a list of characters enclosed in '[]'. It normally matches any single character from the list (but see below). If the list begins with '^', it matches any single character (but see below) not from the rest of the list. If two characters in the list are separated by '-', this is shorthand for the full range of characters between those two (inclusive) in the collating sequence, for example, '[0-9]' in ASCII matches any decimal digit. It is illegal(!) for two ranges to share an endpoint, for example, 'a-c-e'. Ranges are very collating-sequence-dependent, and portable programs should avoid relying on them.

To include a literal ']' in the list, make it the first character (following a possible '^'). To include a literal '-', make it the first or last character, or the second endpoint of a range. To use a literal '-' as the first endpoint of a range, enclose it in '[.' and '.]' to make it a collating element (see below). With the exception of these and some combinations using '[' (see next paragraphs), all other special characters, including '\', lose their special significance within a bracket expression.

Within a bracket expression, a collating element (a character, a multi-character sequence that collates as if it were a single character, or a collating-sequence name for either) enclosed in '[.' and '.]' stands for the sequence of characters of that collating element. The sequence is a single element of the bracket expression's list. A bracket expression containing a multi-character collating element can thus match more than one character, for example, if the collating sequence includes a 'ch' collating element, then the RE '[[.ch.]]*c' matches the first five characters of 'chchcc'.

Within a bracket expression, a collating element enclosed in '[=' and '=]' is an equivalence class, standing for the sequences of characters of all collating elements equivalent to that one, including itself. (If there are no other equivalent collating elements, the treatment is as if the enclosing delimiters were '[.' and '.]'.) For example, if o and ^ are the members of an equivalence class, then '[[=o=]]', '[[=^=]]', and '[o^]' are all synonymous. An equivalence class may not(!) be an endpoint of a range.

Within a bracket expression, the name of a character class enclosed in '[:' and ':]' stands for the list of all characters belonging to that class. Standard character class names are:

```
alnum       digit       punct
alpha       graph       space
blank       lower       upper
cntrl       print       xdigit
```

These stand for the character classes defined in wctype(3). A locale may provide others. A character class may not be used as an endpoint of a range.

In the event that an RE could match more than one substring of a given string, the RE matches the one starting earliest in the string. If the RE could match more than one substring starting at that point, it matches the longest. Subexpressions also match the longest possible substrings, subject to the constraint that the whole match be as long as possible, with subexpressions starting earlier in the RE taking priority over ones starting later. Note that higher-level subexpressions thus take priority over their lower-level component subexpressions.

Match lengths are measured in characters, not collating elements. A null string is considered longer than no match at all. For example, 'bb*' matches the three middle characters of 'abbbc', '(wee|week)(knights|nights)' matches all ten characters of 'weeknights', when '(.*).*' is matched against 'abc' the parenthesized subexpression matches all three characters, and when '(a*)*' is matched against 'bc' both the whole RE and the parenthesized subexpression match the null string.

If case-independent matching is specified, the effect is much as if all case distinctions had vanished from the alphabet. When an alphabetic that exists in multiple cases appears as an ordinary character outside a bracket expression, it is effectively transformed into a bracket expression containing both cases, for example, 'x' becomes '[xX]'. When it appears inside a bracket expression, all case counterparts of it are added to the bracket expression, so that, for example, '[x]' becomes '[xX]' and '[^x]' becomes '[^xX]'.

No particular limit is imposed on the length of REs(!). Programs intended to be portable should not employ REs longer than 256 bytes, as an implementation can refuse to accept such REs and remain POSIX-compliant.

Appendix III: Privacy / remote access disclosures

Your Cipafilter contains several features which, while designed to enhance usability and provide for excellent support, may create privacy concerns for some. In the interest of maintaining openness and an informed user base, these features are documented in this section.

Remote access

DerbyTech / Cipafilter retains the administrative keys and Web interface passwords for all Cipafilter units. This is necessary for certain technical functionality (for instance, remotely deploying updates to components such as the **Troubleshooting** page), but perhaps more importantly it allows Cipafilter employees to, at any time, remotely access your Cipafilter.

This is intended to provide speedy tech support — technicians are able to quickly log in to diagnose problems or update settings without requiring customers to set up special accounts and so on. From time to time, tech support will also forward bugs or configuration issues to Cipafilter development for resolution; this may entail logging in to diagnose the problem and (less commonly) to deploy a fix.

As many customers are uncomfortable having maintenance performed or serious changes applied to their devices (particularly during business hours), all reasonable effort is taken to avoid making any such modifications to running units without receiving prior approval from the customer. However, exigent circumstances, such as the disclosure of a serious vulnerability in one of the Cipafilter firmware's constituent packages, may require changes to be applied without notification (usually after business hours or during the weekend).

Configuration data, database contents, and logs

All Cipafilter units automatically upload backups of their configuration data to the Cipafilter Enterprise system each night. These backups are encrypted prior to (and then during) transmission, and are stored in the same encrypted format. However, in order to provide support and disaster recovery, Cipafilter employees have the ability to decrypt these configurations. This means that DerbyTech / Cipafilter does have access to certain sensitive configuration data, like LDAP authentication credentials, **User Manager** credentials, and SSL private keys.

As part of their administrative remote-access capabilities, Cipafilter employees also have access to data stored locally on customers' hardware. This includes not only the previously mentioned configuration information, but also the contents of the Cipafilter database and its logs. These sources usually do not contain critically sensitive data like passwords, but they can reveal otherwise detailed information about an organization's network configuration, devices, and Internet usage, including (if so configured) those of staff and administrators.

Although copies of this local data are not maintained in Cipafilter's Enterprise system, excerpts may be forwarded to other Cipafilter employees in order to diagnose technical issues.

This information will never be shared externally or used for any purpose except backup, recovery, problem diagnosis, and similar technical support.

Appendix IV: Legal notices

In addition to both proprietary and open-source first-party components, Cipafilter firmware contains numerous open-source components derived from third-party projects. Cipafilter aims to meet the licensing requirements of all of these projects.

For legal notices, acknowledgements, and copyright/licensing information, please see the file **NOTICE.txt** available on every filter's Web interface. If you have received this documentation or a related product without this file, or if you have any other questions or requests regarding copyright and licensing, please contact Cipafilter Support or refer to the **Terms of Use** page, also present on every filter's Web interface.