

Chapter

1

Introduction to Firewalls

This chapter provides a brief overview of firewalls—what they can and cannot do. It is not meant to comprehensively cover the topic of firewalls or network security in general. These topics are better covered by more general texts. In this chapter, you will explore some of the technologies used in firewalls, investigate which technologies are used by FireWall-1, and establish why FireWall-1 is the right firewall for you. Examples of how a given technology handles a specific service are also provided.

By the end of this chapter, you should be able to:

- Understand what a firewall is and is not capable of
- Understand what technologies firewalls typically employ
- Discuss the pros and cons of different firewall technologies
- Understand why FireWall-1 is the right firewall for the job

What Is a Firewall?

A firewall is a device that allows multiple networks to communicate with one another according to a defined security policy. They are used when there is a need for networks of varying levels of trust to communicate with one another. For example, a firewall typically exists between a corporate network and a public network like the Internet. It can also be used inside a private network to limit access to different parts of the network. Wherever there are different levels of trust among the different parts of a network, a firewall can and should be used.

Firewalls are similar to routers in that they connect networks together. Firewall software runs on a host, which is connected to both trusted and untrusted networks. The host operating system is responsible for performing routing functions, which many operating systems are capable of doing. The host operating system should be as secure as possible prior to installing the firewall software.

2 CHAPTER 1 • INTRODUCTION TO FIREWALLS

This not only means knowing how the operating system was installed but also making sure that all of the security patches are applied and that unnecessary services and features are disabled or removed. More details about these security issues are provided in Chapter 3.

Firewalls are different from routers in that they are able to provide security mechanisms for permitting and denying traffic, such as authentication, encryption, content security, and address translation. Although many routers provide similar capabilities (such as high-end devices from Cisco), their primary function is to route packets between networks. Security was not part of their initial design but rather an afterthought. A firewall's primary function is to enforce a security policy, and it is designed with this in mind.

What a Firewall Cannot Do

It is important to realize that a firewall is a tool for enforcing a security policy. If all access between trusted and untrusted networks is not mediated by the firewall, or the firewall is enforcing an ineffective policy, the firewall is not going to provide any protection for your network. However, even a properly designed network with a properly configured firewall cannot protect you from the following dangers.

- *Malicious use of authorized services:* A firewall cannot, for instance, prevent someone from using an authenticated Telnet session to compromise your internal machines or from tunneling an unauthorized protocol through another, authorized protocol.
- *Users not going through the firewall:* A firewall can only restrict connections that go through it. It cannot protect you from people who can go around the firewall, for example, through a dial-up server behind the firewall. It also cannot prevent an internal intruder from hacking an internal system. To detect and thwart these kinds of threats, you may need a properly configured intrusion detection/prevention system.
- *Social engineering:* If intruders can somehow obtain passwords they are not authorized to have or otherwise compromise authentication mechanisms through social engineering mechanisms, the firewall won't stop them. For example, a hacker could call your users pretending to be a system administrator and ask them for their passwords to "fix some problem."
- *Flaws in the host operating system:* A firewall is only as secure as the operating system on which it is installed. There are many flaws present in operating systems that a firewall cannot protect against. This is why it is important to properly secure the operating system and apply the necessary security patches before you install the firewall and on a periodic basis

thereafter. It also explains why “appliance” firewalls such as those provided by Nokia and NetScreen, which contain a purpose-built, hardened operating system, are becoming more popular.

- *All threats that may occur:* Firewall designers often react to problems discovered by hackers, who are usually at least one step ahead of the firewall manufacturers.

An Overview of Firewall Security Technologies

Many companies engage in marketing hype to try to prove that their technology is better. Despite the hype, all firewall security technology can be broken down into three basic types: packet filtering (stateful or otherwise), application layer gateways, and Stateful Inspection.

Packet Filters

Packet filters screen all network traffic at the network and transport layer of the TCP/IP packet. This means they look at source and destination IP addresses, protocol number, and, in the case of TCP and UDP, source and destination port numbers. Packet filtering is built into routers as well as some UNIX kernels. Usually, when site administrators start thinking about network security, they start with packet filtering because it is inexpensive. Most routers on the market today, even consumer-grade models, support some form of packet filtering. Because routers are needed to connect different networks together (especially when connecting to the Internet), the additional cost for using this technology is minimal. Packet filtering requires very little extra memory and processing power, so even a low-end router can handle a fairly moderate load. Packet filtering is also fairly transparent to legitimate users.

Traditional packet filtering is static, that is, the only criteria for allowing packets are whether or not the IP addresses or port numbers match those specified in the packet filter configuration. Many packet filters today implement some concept of the “state” of a connection, using a table and additional information in the TCP headers to track previously allowed packets within a connection. This makes it much easier to allow only, for instance, outbound connections from a trusted network to an untrusted network without inadvertently allowing unrelated packets from the untrusted network to the trusted network.

The biggest downside to packet filters is that they are difficult to maintain. Although this point is certainly arguable, even an expert can have trouble configuring a moderately complex set of access lists or Linux `ipchains` rules. Many consumer-grade routers that have packet filtering do not have an adequate interface or are very limited in what they can filter.

4 CHAPTER 1 • INTRODUCTION TO FIREWALLS

Packet filters also do not screen above the network and transport layers. This means they cannot do things like:

- Provide content security (e.g., virus scanning or filtering based on specific sites and Web pages accessed)
- Authenticate services (i.e., make sure only authorized users use a service)
- Dynamically open and close ports for applications as they require them (necessary for applications like RealAudio, FTP, and H.323 applications)
- Validate a particular port that is used only for a specific service (e.g., making sure that only valid HTTP traffic traverses port 80)

Application Layer Gateways

Application layer gateways, also known as *proxies* or *application proxies*, take requests from clients and make them connect to servers on the client's behalf. In some cases, the client explicitly connects to the proxy server. In other cases, the proxy intercepts the connection with help from the underlying operating system or network architecture. Because an application proxy is usually specific to the network service, it can be fully aware of the session. This means the proxy can do content screening, provide authentication, and ensure that only the particular service is used (e.g., an HTTP proxy can make sure that only HTTP traffic is allowed through), or it can provide other application-specific services such as caching. It also provides a well-formed connection to servers on the other side of the firewall because it opens up connections on behalf of the clients.

However, this extra capability comes at a price. Application proxies require memory and CPU cycles just like any other application. Generally speaking, application proxies use more memory and CPU cycles than packet filtering, although how much they use depends on the specific circumstances. If you want to use application proxies to provide services to the Internet, each application you want to run through your firewall must have a proxy written for it, or the application must be compatible with a “generic” proxy that will work with simple TCP or UDP connections. Because many applications are not being developed to work with an application proxy, some applications simply cannot be proxied. The client/server model is somewhat broken by application proxies because the application proxy will always originate the connection from the server's point of view.¹ In large environments, the poor throughput of application proxies is another drawback.

1. Some would argue that this is actually not a problem. Whether or not this is a problem depends on the specific application and what you are trying to track down.

Another important drawback of a proxy, particularly for internal use, is that it becomes very difficult to track who is going where for how long because the proxy often masks the original source or destination of the traffic. You might be able to track this on the firewall, but from any other vantage point on the network, how do you know?

Stateful Inspection

Stateful Inspection combines the best features of stateful packet filtering and application layer gateways. Check Point's Stateful Inspection engine rests between the data link and network layers (e.g., between the network interface card and the TCP/IP driver). TCP/IP packets from the network layer and higher are scanned according to your security policy and will be either allowed through or stopped. The TCP/IP stack will not see dropped or rejected packets, which can provide an extra layer of protection. Stateful Inspection can look at the entire packet and make security policy decisions based on the contents and the *context* of the packet, using a state table to store connection state and using knowledge of how specific protocols are supposed to operate. In the case of FTP, FireWall-1 can dynamically open ports between two hosts so that the communication will succeed and then close the ports when the connection is done. Stateful Inspection is what gives FireWall-1 "Application Intelligence" (e.g., NG with Application Intelligence, or NG AI).

Stateful Inspection requires slightly more memory and CPU cycles than packet filtering because it has to do more, but it takes substantially less memory and CPU usage than does an application proxy. Stateful Inspection is best when the engine is made aware of how a protocol functions, although Check Point does not make use of Stateful Inspection for every protocol. Because Stateful Inspection does track connection state regardless of the service, it is better than a packet filter, but you are limited to opening specific ports and allowing the traffic through without further checking.

Technology Comparison: Passive FTP

It is useful to compare how the different technologies handle complex connection types. One such connection type is Passive FTP, which is used by Web browsers when they initiate an FTP connection. Passive FTP requires:

1. A TCP connection from a client to port 21 on the FTP server.
2. A TCP connection from a client to some random high port on the FTP server for data communication. The ports used for this communication are communicated to the client when it requests passive mode via the PASV command.

6 CHAPTER 1 • INTRODUCTION TO FIREWALLS

For this comparison, assume that the FTP server is behind your firewall and that you need to allow people on the Internet to FTP to this machine.

Packet Filters

Packet filtering can handle standard FTP quite nicely because it uses fixed TCP ports (20 and 21). However, in order to allow Passive FTP, the packet filter has to open all TCP ports above 1024 to allow Passive FTP to work with the FTP server. This is a gaping hole that can be used by programs other than FTP to compromise your systems.

Application Proxies

An application proxy is aware of the FTP connection and opens all the necessary ports and connections to complete the FTP connection. However, each TCP or UDP connection through an application proxy requires twice the normal number of connections on the proxy server (one for each side of the connection). A normal Passive FTP connection requires two open connections on a client machine. On the application layer gateway, this translates to four open TCP connections.

Most operating systems have a limit to the number of simultaneous connections they can handle. If enough connections are going through the machine at the same time, this limit will be reached, and no further connections will be allowed through. In high-performance, high-capacity networks, using a proxy for FTP connections is simply asking for trouble.

Stateful Inspection

Stateful Inspection understands connection context. When the PASV command is sent from the client to the server, Stateful Inspection reads the server's response and opens the ports necessary to complete the connection. It also restricts the IP addresses that can use these ports to the client and server. The connection then goes through the firewall normally. Because Stateful Inspection allows the native operating system to route, no connections are established on the firewall itself. Once the connection is terminated, the ports opened by the PASV command are closed.

Technology Comparison: Traceroute

Traceroute is used to show the particular path a connection will take through the various routers and gateways within the network and gives you a basic idea of the latency between any two hosts on a network. It is a common troubleshooting tool used by network administrators. There are two varieties of traceroute:

UDP and ICMP. UDP traceroute is used by almost every UNIX implementation. ICMP traceroute is typically used by Microsoft operating systems, though some UNIX implementations also allow you to perform an ICMP traceroute. How traceroute functions can be used to show the strengths of Stateful Inspection and the weaknesses of packet filters and application proxies.

UDP traceroute involves sending out packets to high-numbered ports above 31000—the actual ports used will vary based on the implementation. ICMP traceroute uses ICMP Echo Requests instead. In both cases, the client generates a number of packets (usually three) over a period of time (usually one second) to the server using a time to live (TTL) value of 1. Each subsequent set of packets will have an increasingly higher TTL value, which allows the packets to get closer and closer to the server.

During a traceroute session, any of the following can occur.

- The server responds with an ICMP Echo Reply message or an ICMP Port Unreachable packet (i.e., the traceroute has finally reached the server).
- An intermediate router or gateway gets a packet with a TTL value of 1; it decrements the TTL to 0. Because a router or gateway cannot route a packet with a TTL of 0, it sends back an ICMP Time Exceeded message.
- An intermediate router or gateway determines it has no route to the server and sends back an ICMP Destination Unreachable message.
- An intermediate router or gateway fails to respond either because it is configured to not respond to or pass traceroute traffic or because it is down.
- The client decides it has sent too many sets of traceroute packets (the default is 30) and stops.

For any firewall solution to securely allow traceroute through,² it must take all of these situations into account. Let's explore how each of the firewall technologies can address passing traceroute.

Packet Filters

With packet filtering, you would have to allow the following types of traffic to pass through your packet filter:

- All UDP ports above 31000
- ICMP Echo Request

2. Most firewall administrators do not want to allow traceroute into their network from the outside but do wish to allow internal hosts to initiate it outbound and then allow only appropriate reply traffic back in.

Conversely, you would also have to allow the following types of packets to enter your network from any host:

- ICMP Echo Reply
- ICMP Time Exceeded
- ICMP Destination Unreachable
- ICMP Port Unreachable

Although these rules would allow legitimate traceroute traffic, they can also allow network access by packets that were not in response to a valid traceroute request. In the past, these kinds of unsolicited packets were used in denial-of-service (DoS) attacks. It is important that you allow in only those packets that are in response to a traceroute or ping query. Packet filtering alone is not an adequate tool to allow traceroute to function yet protect you from possible DoS attacks. It is important to note that the UDP ports allowed could also be used for something other than traceroute.

Application Proxies

UDP can be proxied to some degree, but due to its nature, ICMP cannot be proxied, though some versions of SOCKS can proxy ICMP using special SOCKS-aware ICMP programs. In a relatively small, controlled, homogeneous environment, this may be feasible. In a large, heterogeneous environment protected by application proxies, it may not be possible to allow all clients to traceroute through the firewall.

Stateful Inspection

With Stateful Inspection, you can watch for either a UDP packet with a low TTL value or an ICMP Echo Request packet coming from a particular client. Once this happens, you can temporarily permit the necessary ICMP packets to return to the client initiating the outgoing traceroute request. After you have received the appropriate response (i.e., an ICMP Echo Reply, Port Unreachable, or Destination Unreachable message) and/or after a specific period of time (e.g., 60 seconds), you can stop allowing the necessary ICMP packets to the client.

FireWall-1 statefully inspects ICMP.

What Kind of Firewall Is FireWall-1?

Check Point advertises FireWall-1 as primarily a Stateful Inspection firewall. Although this is certainly FireWall-1's biggest strength, FireWall-1 uses both Stateful Inspection and application proxies. Application proxies are used when

content security or user authentication is necessary for HTTP, Telnet, rlogin, FTP, and SMTP. Stateful Inspection is used for all other security functions. To be fair, most commercial and even homegrown firewalls employ some combination of these two technologies because none of the technologies can provide all the necessary functionality.

FireWall-1 also offers some other interesting capabilities, many of which are covered in future chapters:

- Site-to-site VPNs
- Client-to-site VPNs
- Content filtering (with the help of third-party products)
- Address translation
- Authentication (integrated with third-party authentication servers)
- Enterprise-wide policy management
- High availability (with the help of third-party products)
- INSPECT, a language with which you can modify Check Point's Stateful Inspection engine

Do You Really Need FireWall-1?

Whether or not you really need FireWall-1 might seem like a strange question to ask in a book about FireWall-1. One of the important points I make in this book is that FireWall-1 is simply a tool used to enforce a security policy. In some cases, using this tool may be overkill. In other cases, this tool is just one of many that are used.

Let's look at a one- or two-person site. In this case, whether or not to use a firewall depends on what the network connection is and what needs to be protected. If the connection is an analog dial-up connection to the Internet that does not stay up a majority of the time, a firewall may not be entirely necessary. If the connection is something more permanent, like a leased line, Digital Subscriber Line (DSL), or cable modem, or if what goes on at this site is highly sensitive or valuable, a firewall may be necessary. If the people who occupy this site are technically savvy, perhaps they will set up their external router with an access list, set up a multihomed host using a BSD or Linux-based operating system, use one of the many consumer-grade firewall devices on the market, or install personal firewall software on the computers. Depending on what the site's needs are, these solutions may be sufficient.

Now let's look at a slightly larger site, say, one that employs 25 to 50 people. This type of site is likely to have some sort of permanent Internet connection.

10 CHAPTER 1 • INTRODUCTION TO FIREWALLS

It may even have an externally accessible server or two like a mail server and a Web server. Again, as mentioned previously, this type of site could probably get away with setting up a multihomed host using a BSD or Linux-based operating system running their built-in filtering mechanisms, or an access list on a router. Perhaps the site also needs to allow one or two people access to the internal network from home. At this point, a few “holes” would be added to the firewall. At a later time, a few other people might want to use some sort of specialized application through the firewall and a few more holes would get added. Pretty soon, the firewall starts to look like Swiss cheese.

Now let’s talk about a large corporate site with thousands of people. A site like this could use a firewall or two. One obvious place to put a firewall would be at the external connection to the world, but firewalls could also be used internally to protect certain sensitive departments like human resources, research and development, or accounting. And perhaps this corporate site is also responsible for some smaller remote offices. These remote offices likely need secure access into the internal network at the corporate site. Also, the corporate site might like to be able to manage the security policy for the remote sites. And, of course, there are those who want to work from home or who need secure access to the corporate network from the Internet.

People tend to think of security needs in terms of the size of the network involved. The preceding examples are typical of what I have experienced in the real world. What type of firewall you require, if any at all, really comes down to your specific needs or the needs of an organization. A one- or two-person site might be developing source code that could potentially be worth millions of dollars; thus network security becomes important. Another example might be a university network with thousands of students, where an open environment is far more important than a secure environment—although you can bet that certain parts of the network, like admissions and finance, require very tight security. The main question you have to ask when considering a firewall is, “What is at stake if an unauthorized person gains access to my network?”

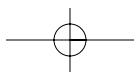
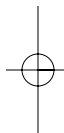
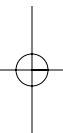
FireWall-1 is an appropriate solution for networks of all shapes and sizes. This is because FireWall-1 is one of the few firewalls that can grow with your needs. In a network with few needs, FireWall-1 can start out as a simple Internet firewall. As your needs change, you can easily add firewalls and still be able to easily keep track of and manage your corporate-wide security policy. As your network grows, you can readily upgrade or change the platform on which FireWall-1 is installed and add functionality, such as a VPN, quite easily. Because FireWall-1 works the same on all supported platforms, you will not have to spend a significant amount of time reconfiguring or relearning the product. Adding new functionality is usually as simple as adding a new license string and

modifying your configuration to support the new feature. With the added functionality of INSPECT, you can program FireWall-1 to securely support just about any service.

With the help of this book, you will be able to effectively use FireWall-1 in just about any network environment in which you work.

More Information

Many other network security topics could have been covered in this chapter, and even the topics covered could have been covered in greater depth. However, the focus of this book is not on general security topics but rather on FireWall-1. Many of the general topics are covered in depth in other books by other authors. Appendix G includes a list of Web sites with more information on interesting software. Appendix H includes a list of recommended books.



Chapter 9 Content Security

In the last chapter, I talked about restricting access based on the user; in this chapter, I talk about restricting access to certain kinds of content. Such restrictions include not allowing people to access certain kinds of sites (e.g., pornography, news), preventing people from accessing specific types of content (e.g., RealAudio, MP3), and scanning content for viruses. I also discuss the various Security Servers for HTTP, FTP, SMTP, and TCP in some detail.

By the end of this chapter, you should be able to:

- Know what CVP and UFP are used for
- Restrict content for HTTP, FTP, SMTP, and generic TCP services
- Understand the performance issues inherent in Content Security
- Understand how to tune your FireWall-1 installation to perform well
- Troubleshoot problems with Content Security

The Security Servers

FireWall-1 normally relies on Stateful Inspection. However, like authentication, the business of virus and content filtering requires more capabilities than can be provided by Stateful Inspection alone. In these cases, FireWall-1 uses the various Security Servers to perform the necessary tasks. In the last chapter, I discussed how they were used for authentication. In this chapter, I look at each individual Security Server a bit more closely and explain how to configure them.

A Word about Licensing and Third-Party Products

All firewall modules can use Content Security. Certain kinds of embedded firewalls cannot use Content Security. However, in order to use more than the rudimentary functions of Content Security in FireWall-1, third-party software is required. Check Point maintains a list of compatible applications and software vendors at <http://www.opsec.com>.

CVP and UFP

Inevitably, Check Point recognized that it could not do everything in terms of providing security. Consequently, Check Point created a program called Open Platform for Security (OPSEC), which allows third-party products to “hook in” to Check Point FireWall-1 and provide services. Two of these protocols, Content Vectoring Protocol (CVP) and URL Filtering Protocol (UFP), are discussed in this section. Some of the other OPSEC functions include the following:

- *Suspicious Activity Monitoring Protocol (SAMP)*: Provides for intrusion detection
- *Log Export API (LEA)*: Offers the ability to analyze firewall logs
- *Event Log API (ELA)*: Helps other applications tie into Check Point’s alerting mechanisms
- *OPSEC Management Interface (OMI)*: Allows third-party products to access the security policy
- *Public Key Infrastructure (PKI)*: Enables Check Point to use third-party certificate authorities for authentication and encryption
- *Secure Authentication API (SAA)*: Allows Check Point to use a variety of authentication mechanisms such as hardware-based tokens and biometric authentication
- *High Availability (HA)*: Provides for highly available firewall servers
- *User-to-Address Mapping (UAM)*: Helps authenticate and track users more effectively
- *User Authority API (UAA)*: Offers the ability to pass authentication information to other servers in order to reduce the number of authentication prompts

CVP is used to scan content. It is typically used to scan for viruses, but it can also be used to scan for malicious Java applets or ActiveX controls, depending on which CVP server you decide to use. CVP works this way: A content stream is intercepted by one of the Security Servers. FireWall-1 determines that the content needs to be scanned by the CVP server before allowing the content to be given to the end user. As the content is downloaded through the firewall, it is sent to the CVP server, which typically runs on a separate server from the firewall. The CVP server then takes one of three actions toward the content (this action is configured in the security policy).

1. Send the content as is, without any modifications.
2. Send corrected content, with the virus or other offending content removed.
3. Do not send the content at all.

Table 9.1 Wildcards usable in all resources

| Character | Description |
|-----------|---|
| * | Matches any string of any length. For example, <code>*@phoneboy.com</code> matches all e-mail addresses at phoneboy.com. |
| + | Matches any single character. For example, <code>pink+@acmelabs.com</code> would match <code>pinky@acmelabs.com</code> but not <code>pinkie@acmelabs.com</code> . |
| {} | Matches any of the listed strings. For example, <code>brain@{acmelabs,animaniacs}.com</code> would match both <code>brain@acmelabs.com</code> and <code>brain@animaniacs.com</code> . |

UFP is used to filter HTTP traffic destined for the Internet based on URLs and the categories under which they fall. As a user requests a URL in his or her Web browser, FireWall-1 uses the HTTP Security Server to check that URL against a UFP server, which returns the category for the URL. Based on that category and the defined security policy, FireWall-1 either permits the connection to the URL or rejects it. This allows, say, Christian organizations to filter non-Christian content or for workplaces to filter pornography.

Resources and Wildcards

Resources are simply a way to match a specific kind of content and then perform some action on it. Some resources are matched based on a query to a UFP server. Others are matched based on specific resources that you define.

The wildcards listed in Table 9.1 can be used in all resources. SMTP provides an additional wildcard, which is discussed in the SMTP Security Server section later in this chapter.

The HTTP Security Server

Of all the Security Servers in FireWall-1, the HTTP Security Server is used most often. Because the HTTP Security Server can be used with both CVP and UFP, I will cover how to set up both types of Content Security.

The HTTP Security Server is enabled when the following situations are true.

- There is a line that permits `in.ahhttpd` to start up in `$FWDIR/conf/fwauthd.conf`. This is normally present by default.
- A resource is used in your security policy or in a rule that involves User Authentication for HTTP.

272 CHAPTER 9 • CONTENT SECURITY


Refer to Chapter 8 for information on authentication. Defining resources is first discussed in the Filtering HTTP without a UFP or CVP Server subsection. The proper line for the HTTP Security Server in `$FWDIR/conf/fwauthd.conf` should have no comment symbol (#) at the beginning of the line and should look like this:

```
80      fwssd      in.ahttpd      wait      0
```

The first argument is the port on which the HTTP Security Server runs (port 80). The second argument states that it uses the binary `fwssd` to run the Security Server. The third argument specifies which server it will be. In this case, it is the HTTP Security Server (i.e., `in.ahttpd`). The fourth argument, which is usually `wait`, is used to indicate one of two things: which port it listens on (if greater than or equal to zero) or how many instances of the server to run (if negative). I discuss the latter point in the Performance Tuning section later in this chapter. The only time you want the Security Server listening on a particular port is when users will use the firewall as a nontransparent proxy for HTTP. If this line is not present or is commented out, the HTTP Security Server will not run, and any process that relies on it will fail.

Filtering HTTP without a UFP or CVP Server

FireWall-1 has some rudimentary filtering features that can be used without a UFP or CVP server. These features should be used only for the most basic of filtering needs. Anything too complex should be done with a UFP or CVP server.

You can use these filtering features by creating URI resources. From Smart Dashboard/Policy Editor, select Manage, then choose Resources or click on the  icon in the objects tree, right-click on URI, and select New URI. Next, select New, and then choose URI. You are presented with the window shown in Figure 9.1.

You can set the following properties on the General tab.

Name: In this field, enter the name of the resource (must be unique).

Comment: Enter any information you like in this field.

Color: Select a color to represent the resource.

Use this resource to: This choice allows you to determine the resource's primary goal: logging URLs that users access (Optimize URL logging) or providing content security (Enforce URI capabilities, which also logs URLs that people access). The latter relies on the HTTP Security Server; the former operates directly in the kernel and thus is faster. In NG with Application Intelligence, this tab also presents the option Enhance UFP performance. This allows you to move UFP functions into the kernel module to increase

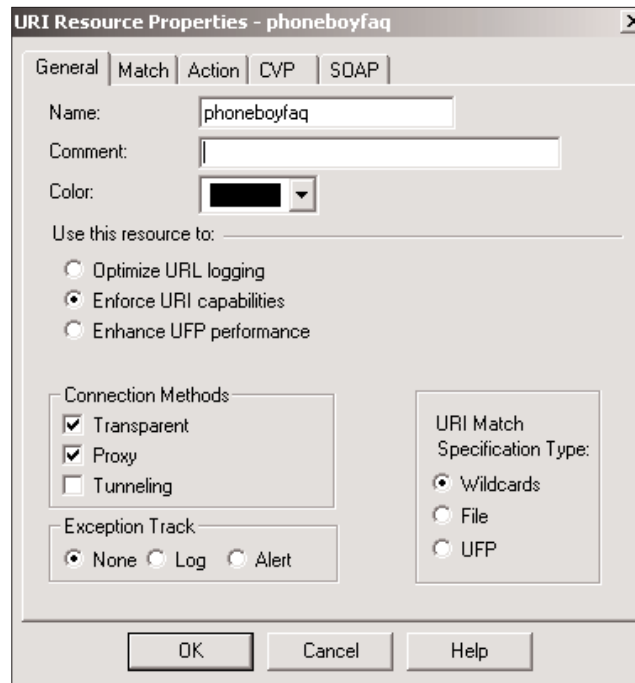


Figure 9.1 URI Resource Properties, General tab

performance but prevents you from using UFP caching (explained later in this chapter), CVP, or authentication. It also removes the ability to perform certain HTTP protocol checks, such as validating HTTP methods and content length.

Connection Methods: In this section of the tab you can specify when this resource is applied. Transparent means that the user will use the service normally, and FireWall-1 will transparently intercept the communication. Proxy means that this resource is applied when people specify the firewall as the proxy in their browser. Tunneling is used when FireWall-1 (defined as the proxy to the client's Web browser) cannot examine the content of the request, only the hostname and port number. An example of this is HTTPS. Only the hostname and port number are sent in cleartext; the rest of the content is encrypted. The hostname and port number are the only specifications that can be filtered on using the Tunneling connection method. If Tunneling is specified, all Content Security options in the URI specification are disabled.

URI Match Specification Type: This option specifies how you define this resource: as type Wildcards, File (which requires that you create a URI file),

274 CHAPTER 9 • CONTENT SECURITY

or UFP. The first two methods are discussed later in this subsection. The last method is discussed in the UFP with the HTTP Security Server subsection.

Exception Track: Here you can specify how to log anything this resource acts upon.

After setting these properties, you must then specify which URLs to filter by clicking on the Match tab. Figure 9.2 shows how it looks when the Wildcards option is selected on the General tab.

You can configure the following parameters.

Schemes: This parameter matches the different protocols you can use through the HTTP Security Server. It is relevant only if the firewall is specified as the proxy for these protocols. Normally, it is safe to just select the http checkbox.

Methods: This parameter specifies methods for HTTP. GET is used when you request a particular page (or element on a page); POST is used when sending data to a Web site (filling out forms and so on); HEAD is usually used by caching servers and Web browsers to determine whether or not an element has changed (and thus to decide whether or not to download it); PUT is a less commonly used method for uploading files via HTTP. If another method is required, you can specify it in the Other field. To allow any method, use * in the Other field.

Host, Path, Query: These fields break down the various parts of the URL into filterable components. For example, in the URL `http://www.phoneboy.com/search/wwwwais/wwwwais.cgi?keywords=content+security`, the host part of the URL is `www.phoneboy.com`, the path is `/search/wwwwais/wwwwais.cgi`, and the query is basically everything else (usually for CGI scripts such as search engines). You can filter on any part of the URL.

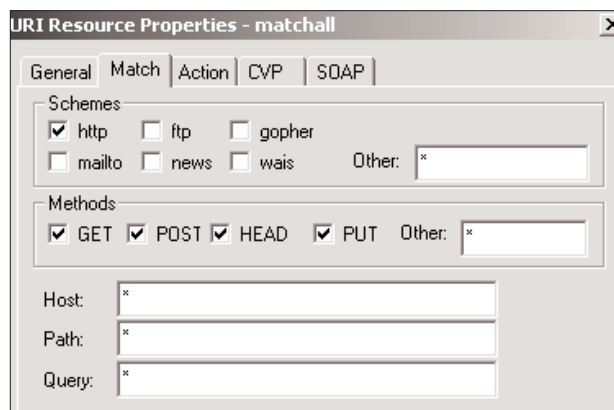


Figure 9.2 URI Resource Properties, Match tab

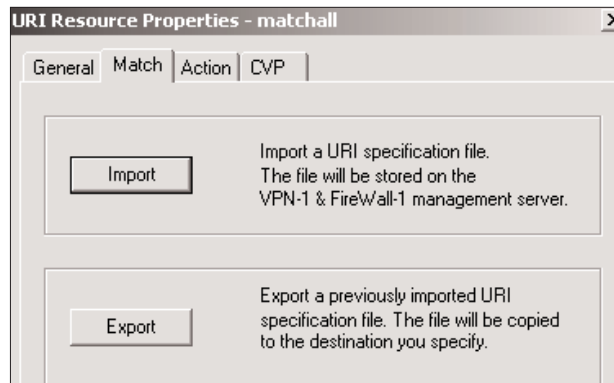


Figure 9.3 URI Resource Properties, Match tab for File resources

If you selected the File option under URI Match Specification Type on the General tab to create a resource of type File (i.e., to filter URIs based on a file) rather than Wildcards, the Match tab shown in Figure 9.3 appears.

A URI specification file is a series of lines in the following format:

```
ip-addr    /path    0
```

`ip-addr` is the IP address of the Web server you want to match against. For sites that resolve to multiple IP addresses, you need to list each one specifically. You can also use fully qualified domain names in this file, though it requires that DNS be enabled and configured on the firewall. `/path` is optional. If you want to restrict a certain subdirectory of a site (or a certain URL), enter it here. 0 (or any hexadecimal number) is required at the end of each line.

Here is a sample file:

```
10.0.146.201 0
10.251.29.12 0
10.91.182.100 /support.d 0
10.184.151.198 /support 0
```

There must also be a blank line at the end of the file. Once you have created this file, click the Import button and specify the path to this file on your Smart Console system. It will then be uploaded to your management console.

You then need to specify the action to take if this resource matches, so click the Action tab (see Figure 9.4).

On this tab, you can configure the following parameters.

Replacement URI: If the rulebase action this resource is used in is dropped or rejected, the user should be redirected to this URL. This could, for

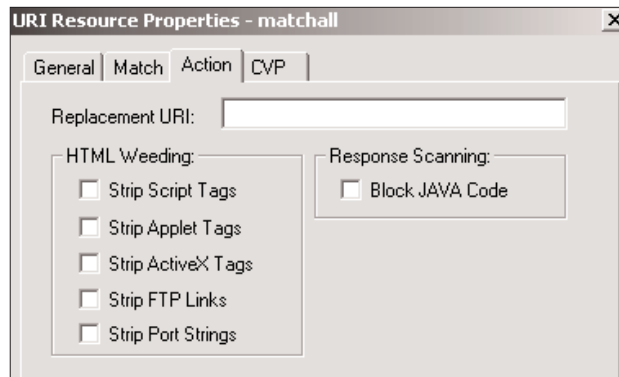


Figure 9.4 URI Resource Properties, Action tab

instance, be a policy document telling people the rules and regulations of Web usage.

HTML Weeding: In this section of the tab you can select which tags to strip out if the action is accepted. The HTTP Security Server does not really strip them but rather comments out the offending HTML so that the tags are not active when downloaded. A user could theoretically save the HTML and reload a modified, local copy.

Block JAVA Code: FireWall-1 can block the download of any Java code if you select this checkbox. It does not match JavaScript, which is done with the HTML Weeding option Strip Script Tags.

Figure 9.5 shows the CVP tab, where you can specify whether or not this resource will enforce virus scanning and the parameters that control how it is done.

This tab includes the following options.

Use CVP: Enable the use of CVP in this resource. If this property is unchecked, all other fields on this screen will be greyed out.

CVP server: Select an OPSEC Application server that has CVP in it. I will show how these are defined in the CVP with the HTTP Security Server subsection.

CVP server is allowed to modify content: This allows the CVP server to attempt to disinfect a file that has a virus. If this option is not checked and the content is determined to have a virus, the communication will be rejected.

Send HTTP Headers/requests to CVP server: These options allow the CVP server to make security or filtering decisions based on data contained in the HTTP request headers.

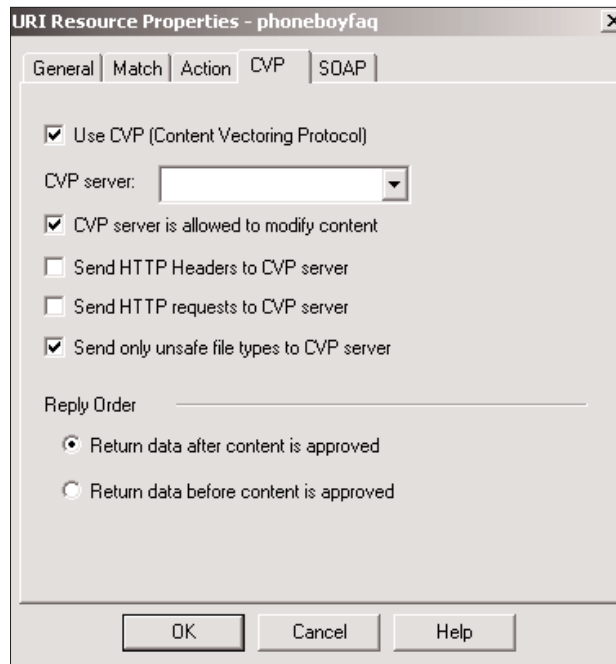


Figure 9.5 URI Resource Properties, CVP tab

Send only unsafe file types to CVP server: This option was introduced in NG with Application Intelligence. Normally, FireWall-1 sends all traffic through the CVP server. If this option is enabled, FireWall-1 inspects the content of the traffic to determine whether it is a kind of file that may contain a virus—FireWall-1 does not trust file extensions or MIME types for these checks. Graphic or movie files are considered “safe” and thus are not sent to the CVP server. Executable and Microsoft Office documents are sent to the CVP server for virus scanning.

Return data after content is approved: Data is sent to the CVP server for approval. Only after all the data has been received and scanned is it sent back from the CVP server. The problem with this option is that with large files on slow links, this can cause the client connection with the server to take a very long time before any data is returned. The client may time out in this case.

Return data before content is approved: This allows the CVP server to scan and correct content “on the fly.” This option solves the problem of transferring large files over slow links, but it may mean the client receives part of a file that the CVP server will ultimately reject because, for instance, it finds a virus it cannot disinfect.

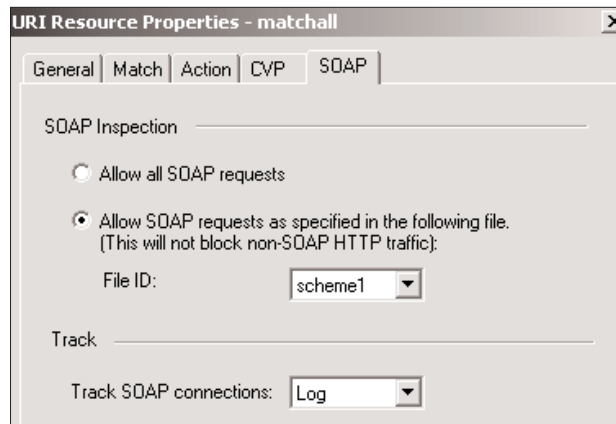


Figure 9.6 URI Resource Properties, SOAP tab

Figure 9.6 shows the SOAP tab, which is relevant only when using Wildcard URI types and NG FP3 and later. This allows you to filter and/or log Simple Object Access Protocol (SOAP) requests over HTTP. You may either allow all SOAP requests or filter for specific ones.

The schemes you can select are defined in files in `$FWDIR/conf/XML` on the management console. There are several files in this directory (`scheme1` through `scheme10`) where you can define specific sets of allowed SOAP requests. The files must contain entries of the following format:

```
namespace      method
```

For example:

```
http://tempuri.org/message/ EchoString
http://tempuri.org/message/ SubtractNumbers
```

You can then use this resource in a rule, as shown in Figure 9.7.

UFP with the HTTP Security Server

The UFP server is a third-party application that should be run on a different platform from the firewall. A variety of UFP servers available for FireWall-1 run on Windows or Solaris. I will not cover their setup in this book. It is sufficient to say that once they are set up correctly, FireWall-1 can then communicate with them on TCP port 18182.

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|------------------|-------------|-------------------|--------|-------|
| internal_network | * Any | http->phoneboyfaq | reject | Log |

Figure 9.7 Sample HTTP resource rule

To configure UFP to work with FireWall-1 and the HTTP Security Server, perform the following steps.

1. Define the workstation object on which the UFP server is running (if necessary).
2. Define the OPSEC Application object that represents the UFP server.
3. Define a URI resource of type UFP.
4. Add a rule using the resource, and install the policy.

Let's assume you have created a workstation object named *babyike* where the UFP server is installed. In SmartDashboard/Policy Editor, do one of the following.


- Select Manage and then choose OPSEC Applications.
- Click on the following icon in the objects tree: . Then right-click on OPSEC Application, and select New OPSEC Application.

Figure 9.8 shows the resulting screen.

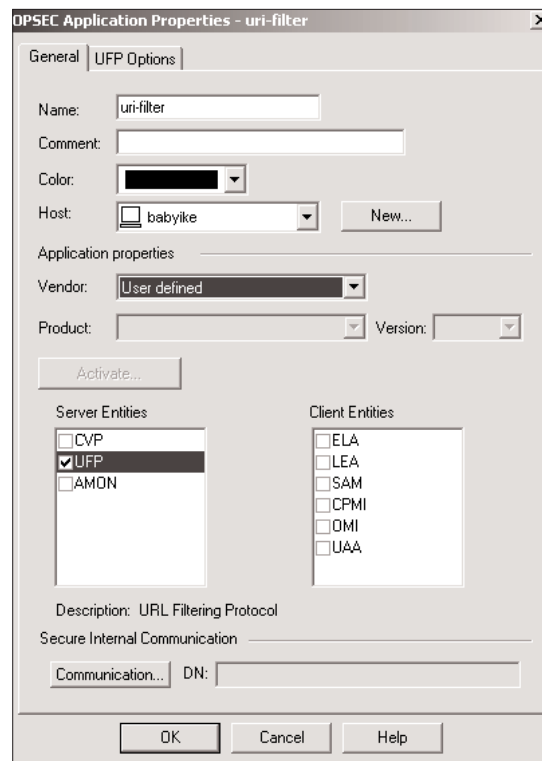


Figure 9.8 OPSEC Application Properties, General tab

The General tab contains the following options.

Name: Enter the name of the resource (must be unique).

Comment: In this field, you can add a note about this OPSEC Application server.

Color: Select whichever color you would like.

Host: This is the workstation object on which the UFP server is running.

Application properties: In this section of the tab, select the vendor of the application, the product, and the version as appropriate. If your vendor isn't listed, you may want to select User defined. In this case, make sure that UFP is checked under Server Entities.

Secure Internal Communication: In FireWall-1 NG FP1 and later, SIC is used to authenticate communications with third-party OPSEC applications. This is where you configure the one-time password used during the initial certificate exchange. Additional steps will need to be performed in your OPSEC application to perform this exchange.

Figure 9.9 shows the UFP properties described below.

Service: This field specifies the service used to communicate with this server. Normally, this should be FW1_ufp (TCP port 18182).

Dictionary: The information in this section is used to validate the connection to your UFP server. Categories are shown if a connection can be successfully established. You can choose the actual categories that are allowed or disallowed in the individual URI resource.

Use early versions compatibility mode: In FireWall-1 4.1, authentication between the UFP server and the firewall module uses something other than SIC. In these cases, check this option and select the appropriate authentication method.

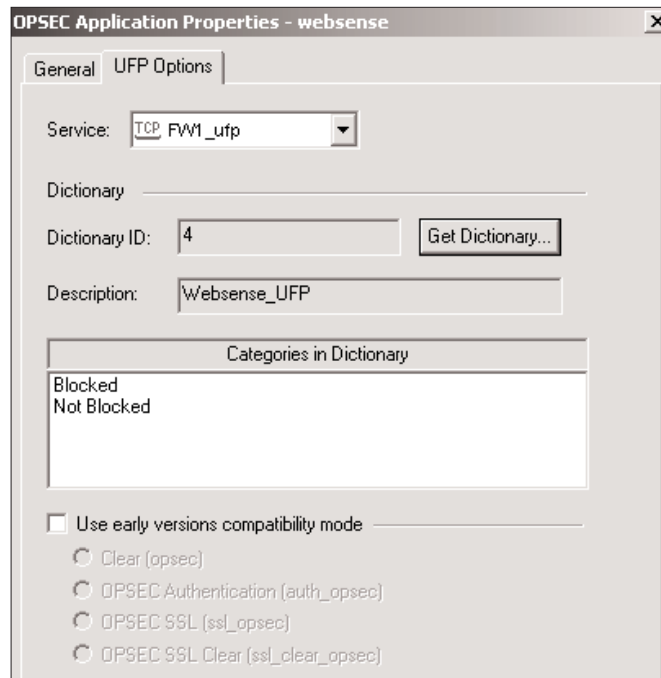
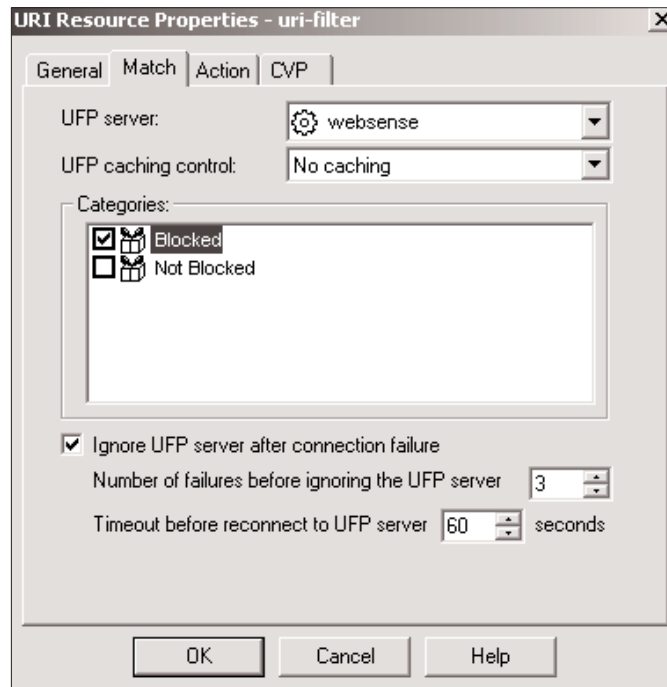
After setting these properties, you can create your URI resource. For this example, the resource is called uri-filter. The URI resource needs to be of type UFP.

Next, go to the Match tab, select the UFP Server websense, and select the Blocked category, as shown in Figure 9.10.

This tab contains the following options.

UFP server: Specify the OPSEC Application object you created that defines the UFP server.

UFP caching control: This option will be explained in the UFP Caching subsection later in this chapter.

**Figure 9.9** OPSEC Application Properties, UFP Options tab**Figure 9.10** URI Resource Properties, Match tab

Categories: Specify the categories on the UFP server to which this URI will apply. For Websense servers, this should be Blocked.

Ignore UFP server after connection failure: FireWall-1 will continually connect to the UFP server. If for some reason the UFP server fails to respond in a timely fashion, this option allows you to specify whether to “fail closed” (i.e., keep trying to connect to the UFP server until successful, meanwhile blocking all HTTP traffic) or “fail open” (i.e., after the specified amount of time, ignore the UFP server and do not categorize the traffic). You can specify the number of failures permitted and the amount of time between each communication attempt before it ignores the UFP server.

If you wanted to, you could go to the Action tab and specify other filtering options, but instead, for this example, let’s move on to create the rules to block Web sites that Websense has been configured to block (see Figure 9.11).

The first rule is created by using the Add with Resource option for the Service column, selecting http, and then selecting uri-filter. This rule catches all Websense-filtered URLs. The second rule permits URLs that are not filtered by Websense. This second rule is necessary to allow access to all URLs except those prohibited by Websense.

CVP with the HTTP Security Server

The CVP server is a third-party application that should be run on a different platform from the firewall. A variety of CVP servers available for FireWall-1 run on Windows or Solaris. I will not attempt to cover their setup in this book. It is sufficient to say that once they are set up correctly, FireWall-1 can then communicate with them on TCP port 18181.

To configure CVP to work with FireWall-1 and the HTTP Security Server, perform the following steps.

1. Define the workstation object on which the CVP server is running (if necessary).
2. Define the OPSEC Application object that represents the CVP server.
3. Define a resource that uses the CVP server (or modify an existing one).
4. Use the rule with the resource, and install the policy.

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|--|-------------|--|--|---|
|  internal_network | * Any |  http->uri-filter |  reject |  Log |
|  internal_network | * Any |  http |  accept |  Log |

Figure 9.11 Sample rules for URI filtering

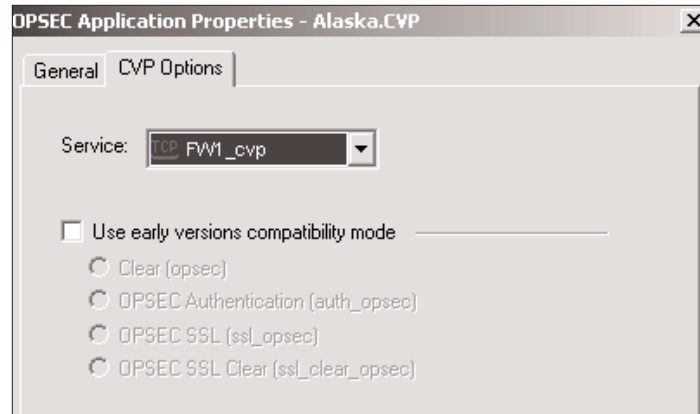


Figure 9.12 OPSEC Application Properties, CVP Options

As in the UFP example above, let's assume you have created a workstation object named babyike where the CVP server is installed. In SmartDashboard/Policy Editor, do one of the following.

- Select Manage and then choose OPSEC Applications.
- Click on the following icon in the objects tree: . Then right-click on OPSEC Application, and select New OPSEC Application.

A screen similar to Figure 9.8 (shown earlier) appears. For this example, the OPSEC Application object is named f-secure-cvp.

Since a CVP server is being defined, choose the relevant options for CVP—choose the correct CVP server information or select User defined in the Vendor field, and make sure CVP is checked under Server Entities. Also define SIC, if relevant.

Figure 9.12 shows the CVP Options tab.

The properties are listed below.

Service: Select the service used to communicate with this server. Normally, this should be FW1_cvp (TCP port 18181).

Use early versions compatibility mode: In FireWall-1 4.1, authentication between the CVP server and the firewall module uses something other than SIC. In these cases, check this option and select the appropriate authentication method.

You then need to create a resource that performs CVP. Create a new resource called virusscan (see Figure 9.13), which matches all URIs and performs virus scanning.

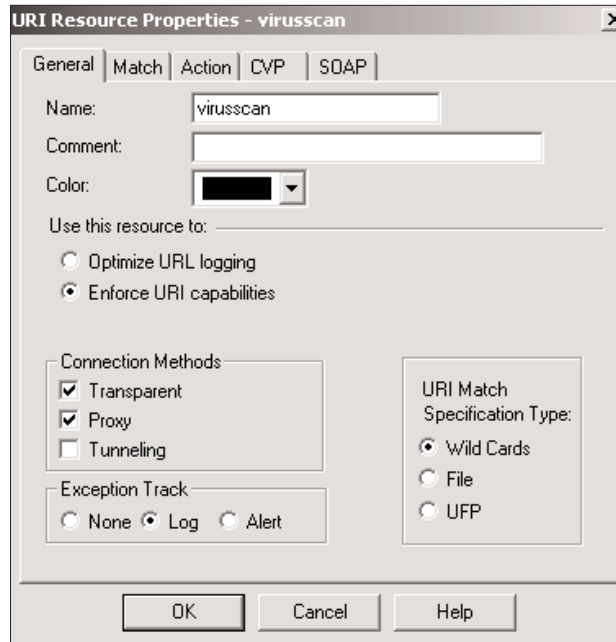


Figure 9.13 URI Resource Properties, General tab

The Match tab, shown in Figure 9.14, shows the settings used to make this resource match all URLs.

The CVP tab, shown in Figure 9.15, is where you define which CVP resource to apply. You then need to add this resource to a rule. You can combine it with the UFP example so that both URLs and content are filtered (see Figure 9.16).

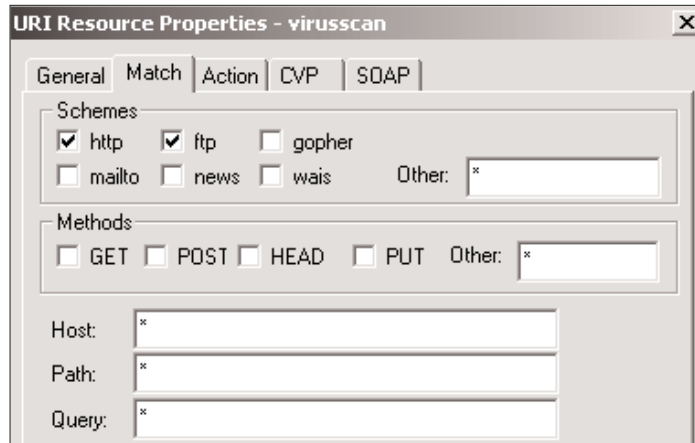


Figure 9.14 URI Resource Properties, Match tab

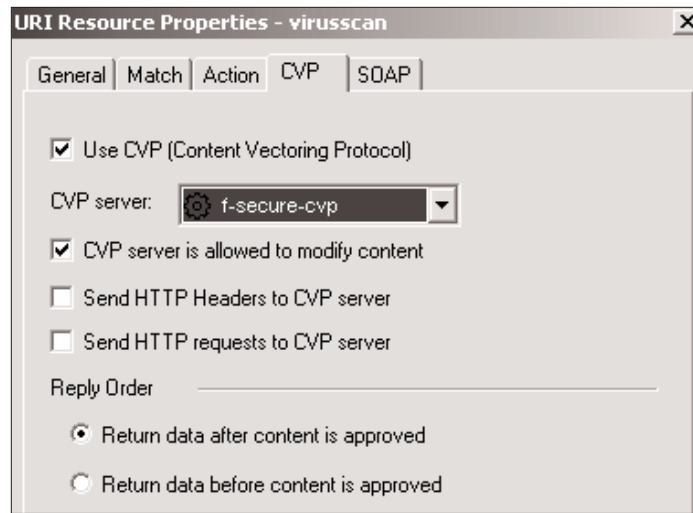


Figure 9.15 URI Resource Properties, CVP tab

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|------------------|-------------|----------------------|--------|-------|
| internal_network | * Any | HTTP http->virusscan | accept | Log |

Figure 9.16 Sample rule with CVP

Frequently Asked Questions about the HTTP Security Server

To keep all the information about a particular Security Server together, I provide a corresponding FAQs subsection at the end of each Security Server section.

9.1: Can I Filter HTTP on Other Ports (e.g., Port 81)?

There are five steps necessary to enable filtering on other ports.

1. Create a TCP service for the port in question (e.g., http81), and make it of type URI.
2. Add a rule with a resource using the new service.
3. Install the security policy.
4. Reconfigure `$FWDIR/conf/fwauthd.conf` to run the Security Server on that port.
5. Bounce the firewall (**cprestart**).

Creating the service is straightforward. Create a new service of type TCP. Set the Protocol Type to HTTP and the port as necessary (e.g., port 81). If you add a resource by right-clicking in the Service part of a rule, you can associate a

286 CHAPTER 9 • CONTENT SECURITY

resource with the new service you created (e.g., http81). If you filter with wild-card resources, you need to enter the host part of the URL as `host:port`. For example, to match all, instead of entering `*`, you need to type it as `*:*`. If you do not do this, your resource will fail. To reconfigure `$FWDIR/conf/fwauthd.conf`, you need to add a line to this file for each unusual port you want to filter on. For port 81, for instance, the line would read as follows:

```
81 fwssd in.ahttpd wait 0
```

Reinstall the security policy, and bounce the firewall after making these changes (**cprestart**).

9.2: Can the HTTP Security Server Forward Requests to a Caching Proxy Server?

Yes, but only if your clients are configured to use the firewall as their proxy server. Set the Use Next Proxy setting in your gateway object definition, Authentication frame.

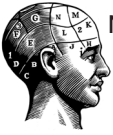
9.3: Why Do I Get the Error “Request to Proxy Other Than Next Proxy Resource http://proxy.foo.com” When Filtering Traffic to a Proxy Server?

Set the Use Next Proxy setting in your gateway object definition, Authentication frame, to point at the proxy server. This setting only allows you to filter traffic to one HTTP proxy server.

9.4: How Do I Redirect People to a Usage Policy Page?

The assumption here is that end users are redirected to a policy page only when they try to access a site that is against the usage policy. You can take one of two approaches.

- Create a resource that matches the sites you *do not want* to allow access to. Use this resource in a rule as shown earlier in the chapter, setting the replacement URL accordingly in this resource.
- Create a resource that matches the sites you *want* to allow access to. If you want to then redirect users to a policy page when they try to load a page they are not allowed to access, use the `matchall` resource and set the replacement URL accordingly. If you want to allow users access to only the sites matched by the resource `allowsites` and deny access to everything else (via a `matchall` resource), the rules would look like those shown in Figure 9.16.



NOTE! If you use the replacement URL in conjunction with User Authentication and a user is redirected to a policy page, the user will get FireWall-1's Authentication Failed page with a link to the redirected page.

9.5: How Do I Prevent People from Downloading Files or Accessing Streaming Media via HTTP?

You can use the HTTP Security Server to deal with both of these issues. If you have CVP, you may be able to use the CVP server to screen out those MIME types. If you are not using a CVP server, you can do this with a wildcard URI. In the Path section of the Match tab, you can specify all file extensions that you do not want people to download.

- To block Real Audio/Real Video, enter `*.{ra,ram,rm,rv}`.
- To block most downloads, enter `*.{exe,zip,com,bat,sit,tar,tgz,tar.gz,lha,rar,r0+}`.

You would then create a rule that uses this resource and denies access to anything matching this resource. Place this rule before your other rules that permit HTTP.



WARNING! Attempting to filter based on file extension or even MIME type is futile. There are plenty of ways to get around these filters by using different extensions or different MIME types, which are merely suggestions for how the file should be treated. In order to filter out all the files you don't want, you will likely filter some files that you *do* want (throwing the baby out with the bath water).

9.6: Can I Allow Certain Users to Download Files Provided They Authenticate?

In order for this to work correctly, all users need to authenticate, even to use normal HTTP. The rules to do this are shown in Figure 9.17.

| SOURCE | DESTINATION | SERVICE | ACTION |
|------------------------|--------------|----------------------|-----------|
| All Users@internal-net | internal-net | http->bad-file-types | User Auth |
| internal-net | internal-net | http->bad-file-types | drop |
| All Users@internal-net | internal-net | http | User Auth |

Figure 9.17 Rules to allow file downloads with authentication

288 CHAPTER 9 • CONTENT SECURITY

Once a packet potentially matches a User Authentication rule—that is, the source, destination, and service match what is specified in the rule—the least restrictive rule in the rulebase is the one that will actually apply. Therefore, it is important to place the rule that denies access to downloads before the rule that allows everyone to use HTTP.

9.7: How Can I Set Up FireWall-1 to Support Content Security for Outbound HTTPS?

Due to the nature of HTTPS, it is possible to authenticate or provide Content Security for HTTPS only when the client specifies the firewall as the proxy for HTTPS. Some other steps must be performed as well.

First, ensure the following line exists in `$FWDIR/conf/fwauthd.conf`:

```
443 fwssd in.ahttpd wait 0
```

If this line does not exist or it is commented out, add/uncomment it, and bounce FireWall-1. Second, modify the predefined service HTTPS. Change the protocol type from None to http. You can then use HTTPS for authentication or Content Security as appropriate provided the client is configured to use the firewall as a proxy for HTTPS requests.

9.8: Can I Block the Use of KaZaA, Instant Messages, and Other Applications That Can Tunnel over HTTP?

These applications are difficult to filter because they can use legitimate HTTP requests. However, a closer inspection of these HTTP request headers reveals telltale traces of what kinds of applications they are. FireWall-1 NG FP3 and later have a way to filter this traffic by using the HTTP Security Server and some properties that you need to add manually to `$FWDIR/conf/objects_5_0.c` on the management console. (See FAQ 4.2 in Chapter 4 for caveats on editing this file.)

The following example shows patterns that block KaZaA and Gnutella. Several other patterns may also already exist in your `objects_5_0.c` file.

```
(firewall_properties:
:fields (

...

:http_header_detection (
    :http_detect_header_pattern_mode (true)
    :http_detect_header_pattern_log (alert)
```

```

: http_header_names (
  : (
    : match_string (X-kazaa)
    : regular_exp (X-Kazaa)
  )
)
: http_header_names_values (
  : (
    : match_string (Server)
    : regular_exp ([kK]a[zZ]a[aA])
  )
  : (
    : match_string (Host)
    : regular_exp ([kK]azaa)
  )
  : (
    : match_string (User-Agent)
    : regular_exp ([gG]nu)
  )
)
)

```

The properties used in these patterns are described below.

http_detect_header_pattern_mode: This property determines whether or not header detection is enabled. By default, it is *false*. To enable header detection, set this property to *true*.

http_detect_header_pattern_log: This one determines what kind of log to generate when one of the defined patterns is detected. Valid values are *none* (log nothing), *log* (generate a log entry), and *alert* (generate an alert).

http_header_names: This property specifies the header names to watch for. Each header you wish to filter has a stanza with two elements: *match_string* (simply for your reference, not actually used by FireWall-1) and *regular_exp* (to specify the regular expression that matches the desired header).

http_header_names_values: Similar to the previous one, but with this property you can look for a header that has a specific value, as specified in *match_string*. If this header is found, the specified *regular_exp* is compared against the header's value.

Once you have added these patterns, you need to use the HTTP Security Server to perform this filtering. This can be done with a simple matchall resource. For Gnutella specifically, you need to add GNUTELLA* to the match field of the resource and make sure the resource is used in a drop or reject rule.

9.9: Why Do I Have Problems Accessing Some Sites When the HTTP Security Server Is Enabled?

This happens because the HTTP Security Server requires tweaking to access many popular sites. Table 9.2 shows the various `firewall_properties` tweaks you can perform by using **dbedit** on the management server or by manually editing the `objects_5_0.c` file (see FAQ 4.2 for details).

Table 9.2 Recommended `firewall_properties` tweaks for the HTTP Security Server

| Property | Setting | Description of Property |
|--|---------|---|
| <code>http_enable_uri_queries</code> | true | If this is set to false, FireWall-1 will strip ASCII encoding of certain characters. |
| <code>http_allow_content_disposition</code> | true | Content disposition is a way to let the Web browser know what it is about to receive. This could potentially allow people to download a type of file that the security policy may not allow them to download. |
| <code>http_log_every_connection</code> | true | This enables logging of all sites an authenticated user visits. |
| <code>http_buffer_size</code> | 32768 | This allows you to specify the buffer size used by the HTTP Security Server to process connections; 32768 is the maximum, and 4096 is the minimum. |
| <code>http_sup_continue</code> | true | This enables the HTTP Security Server to support the HTTP 1.1 CONTINUE command. |
| <code>http_force_down_to_10</code> | true | This forces the HTTP connection version down to 1.0. You need to do this when working with CVP servers. |
| <code>http_avoid_keep_alive</code> | true | This forces the HTTP Security Server to ignore the “keep-alive” directive in HTTP 1.1. You need to do this when working with CVP servers. |
| <code>http_cvp_allow_chunked</code> <code>http_weeding_allow_chunked</code> <code>http_block_java_allow_chunked</code> <code>http_allow_ranges</code> | true | These properties allow the HTTP Security Server to handle requests that occur as byte ranges, often used in HTTP 1.1 requests. <i>(continued)</i> |

Table 9.2 *continued*

| Property | Setting | Description of Property |
|--|---------|---|
| http_use_host_h_as_dst | true | After authentication with partially automatic Client Authentication, the user is normally redirected to the site's IP address instead of the name. With this property set to <i>true</i> , the user will instead be redirected to the host as shown in the HTTP host header (which reflects the host that is being accessed). |
| http_disable_content_enc http_disable_content_type http_allow_ranges | true | Web browsers like Mozilla and Web servers like Apache support "compressed" encoding types. The page and the elements are sent to the client compressed in order to save bandwidth. Enabling these properties allows those kinds of pages to be sent through the HTTP Security Server. |
| http_max_url_length http_max_header_length | n | Enabling these properties prevents FireWall-1 from truncating long URLs. <i>n</i> refers to the number of characters allowed in the URL (for the first property listed) and in HTTP headers (for the second). |
| http_max_auth_redirected_num | n | This allows you to increase the number of partially automatic Client Authentication connections the firewall can process at one time. |
| http_check_request_validity http_check_response_validity | false | Disabling these checks allows Internet Explorer to browse URLs that contain characters not between ASCII 32 and ASCII 127. Normally, FireWall-1 would reject any URLs that contain these characters. |

9.10: How Can I Permit Schemes Other Than FTP and HTTP through the HTTP Security Server?

Enable the following properties by modifying the appropriate `firewall_properties` section in `objects_5_0.C`:

```
:http_allow_double_slash (true)
:http_use_default_schemes (true)
```

The first property enables the HTTP Security Server to accept double slashes (//) in a substring of a URL. In order to allow this, the Security Server defines a set of schemes that it will accept, which is what the second property covers.

292 CHAPTER 9 • CONTENT SECURITY

The default set includes prospero, gopher, telnet, finger, mailto, http, news, nntp, wais, file, and ftp. You may also define new schemes to add to this set. This requires manual editing of `objects_5_0.C`. For example, to add the schemes fish and trouble to the permitted list, add the following code to the `firewall_properties` section of `objects_5_0.C`. (Note that the colons are needed.)

```
:scheme (
  : ("fish:")
  : ("trouble:")
)
```

9.11: How Can I Customize the Error Messages Given by the HTTP Security Server?

On the firewall module, edit the file `$FWDIR/conf/cspc/cspc.en_us`. This enables you to modify just about any message that any of the Security Servers generate. Some common messages to edit include:

- “FW-1 at host”
- “Failed to connect to the WWW server”
- “Unknown WWW server”

Each line in `$FWDIR/conf/cspc/cspc.en_us` is of the following format:

```
IDENTIFIER      size      string
```

`IDENTIFIER` is a unique string that identifies the message to FireWall-1. Do not change this. `size` is the maximum number of characters the message can be. Do not change this either. `string` is the actual string that FireWall-1 will display. It may contain some special words surrounded by # signs, such as `#host#` or `#html#`.

For the three examples listed above, the lines look like this:

```
CPSC_HTTP_FW_AT_HOST      1024      "FW-1 at #host#:"
CPSC_HTTP_CONN_FAIL_ERR   1024      "\n#local_host# Failed
to connect to the #.40server#."
CPSC_HTTP_UNKNOWN_SERVER_ERR 1024      "\n#local_host# Unknown
WWW server."
```

These lines could be changed so they read:

```
CPSC_HTTP_FW_AT_HOST      1024      "Message from firewall:"
CPSC_HTTP_CONN_FAIL_ERR   1024      "\n#local_host# Failed
to connect to the #.40server#. This may be a transient
problem, in which case simply reloading the page will work."
```


If this problem persists, it may be a problem with the remote server."

```
CPSC_HTTP_UNKNOWN_SERVER_ERR 1024      "\n#local_host# Unknown
WWW server. This could mean you typed an incorrect URL or
there was a problem looking up the site in DNS. If the URL
is correct and the problem persists, contact your
administrator."
```

Performance Tuning the HTTP Security Server

One of the most common complaints about Content Security is performance. This is partially the result of the HTTP Security Server running in user space, versus the kernel space where much of FireWall-1 lives. Some of the performance issues can be overcome by tuning the platform on which the HTTP Security Server is running. However, there are some inherent limitations in the Security Servers in terms of the number of users who can go through a single system because Content Security overall requires significantly more resources than simply passing traffic. Personally, I would not use the HTTP Security Server for more than 1,000 users. Check Point has always claimed it is making strides in this area, and the company has increased performance in some circumstances by moving stuff to the kernel. However, I continually hear complaints from administrators who attempt to implement the HTTP Security Server in a large enterprise setting and end up doing something else.

In this subsection, I talk about what you need to do to improve performance of the Security Servers, which will increase the efficiency of the HTTP Security Servers. You should also apply the general performance-tuning suggestions in Appendix E.

Increasing the Number of Allowed Entries in `proxied_conns`

By default, the number of entries in `proxied_conns` (a table that stores connections via the Security Servers) is 25,000. For best performance, you should modify this number to twice the number of connections you actually expect to handle. In `$FWDIR/lib/table.def` on the management console, modify the following line:

```
proxied_conns = dynamic expires AUTH_TIMEOUT kbuf 4;
```

To modify the line to support 50,000 connections, for example, make it read:

```
proxied_conns = dynamic limit 50000 expires AUTH_TIMEOUT kbuf 4;
```

Increasing the HTTP Buffer Size

The default size is 4,096 bytes. It can be increased to a maximum of 32,768. A larger buffer size means fewer system calls; however, each connection will take

294 CHAPTER 9 • CONTENT SECURITY

up that much more memory, so there is a trade-off. See Table 9.2 in FAQ 9.9 for the property that sets the buffer size.

Increasing the Number of Security Server Instances

It's usually necessary to increase the number of Security Server instances for the HTTP Security Server, but you can do it for any Security Server. If you have multiple processors on your firewall, increasing the number of instances allows you to take advantage of these processors. You can use this trick if you are using a single processor system, too.

To increase the number of instances for any Security Server, you need to modify its line in the `$FWDIR/conf/fwauthd.conf` file, which has the following format:

```
<listen-port><binary><daemon-name>wait -<instances>
```

For example, if you want to run four instances of `in.ahhttpd`, which will all listen on port 80, the corresponding line should look like this:

```
80      fwssd      in.ahhttpd      wait      -4
```

Connections from the same HTTP client will always be directed to the same daemon within the authenticated session timeout. Connections begin to use alternate daemons only after the previous daemon fills up. All connections from a client will always be handled by the same daemon.

UFP Caching

When an HTTP request is made, the IP address of the destination is checked against the cache. If the IP address is in the cache, the category associated with that IP address is used. If it is not in the cache, the HTTP Security Server sends the request to the UFP server, which returns the appropriate category information and is then cached. Caching can be controlled by FireWall-1 or by the UFP server. Check Point recommends the latter method, which is thought to be more accurate.

If FireWall-1 controls the caching, FireWall-1 uses two methods to update the cache.

- *One-request method:* FireWall-1 takes the information returned by the UFP server and writes it to the cache.
- *Two-request method:* FireWall-1 makes a second request to the UFP server to determine whether the IP address of the site could match multiple categories. Only if the entire site uses the same category is the data written to the cache.

The one-request method is more aggressive in caching at the expense of cache integrity. The two-request method is slower, but the cache integrity is significantly improved.

Where the UFP server controls the caching, information necessary to update the cache is returned with each request looked up.

To enable UFP caching, create a URI resource, or edit an existing one. Go to the Match tab and enable the caching accordingly. Use the new URI resource in a rule.

Kernel URL Logging

Kernel URL logging allows you to log URLs without having to divert the connection to the HTTP Security Server. This improves overall performance in these situations. Kernel URL logging is enabled in a resource on the General tab by selecting Optimize URL logging (see Figure 9.1 earlier in this chapter). This resource cannot also be used for Content Security or URL filtering.

Adding More Memory, Physical and Virtual

The HTTP Security Server requires lots of memory, especially when it is busy. I have personally witnessed a busy `in.ahhttpd` process on a Nokia platform handling just 1,024 concurrent connections require as much as 87MB of memory! Memory usage for `in.ahhttpd` has proven to be similar on other platforms. The more physical memory you have, the better. Also, your swap size should be fixed (preferably on a dedicated device) and should be twice the size of the amount of physical memory you have.



NOTE! On a Nokia platform that was running a version of IPSO prior to 3.4 and then upgraded, the system will have a swap partition size of only 256MB. For systems newly installed with IPSO 3.4 and later, the swap partition was increased to the lesser of a quarter of the overall available disk space or 1GB. A fresh reinstallation of IPSO from boot manager or boot floppy is required to obtain this larger swap size.

Adjusting File Descriptors Globally and Per Process

On a UNIX platform, there is a limit to the number of file descriptors available both to a specific process and globally. When started, `in.ahhttpd` attempts to reserve the maximum number of file descriptors allowed by the operating system. On Solaris, this is 1,024. On IPSO, this is 2,048. Windows NT does not have this issue.

An HTTP connection going through the Security Server requires two sockets: one for the connection from the client and one for the client to the server.

296 CHAPTER 9 • CONTENT SECURITY

Each socket requires a file descriptor. A limit of 2,048 file descriptors means that fewer than 1,024 concurrent active connections can go through each instance of the `in.ahhttpd` daemon. Other things like logging require file descriptors as well. When the maximum number of file descriptors has been reached, a “Too many open files” error is entered in `$FWDIR/log/ahhttpd.elg`.

Allowing each `in.ahhttpd` daemon to handle more than 1,024 concurrent connections is *not* recommended. Another factor to consider is the amount of memory that each process requires. Recall that earlier I stated that an instance of `in.ahhttpd` handling 1,024 connections took 87MB of memory. Limiting the file descriptors to 1,024 (thus 512 connections per process) reduces the memory utilization to 47MB. The more concurrent connections each process can handle, the larger the process will get. In some cases, it might actually be better to decrease the number of file descriptors and increase the number of processes running.

On IPSO, the number of file descriptors allowed is limited by two kernel variables: `kern:maxfiles` (global limit) and `kern:maxfilesperproc` (per-process limit). The limits are 8,096 and 2,048, respectively. To modify these values, use the **`ipsctl`** command:

```
# ipsctl -w kern:maxfiles 4X
# ipsctl -w kern:maxfilesperproc X
```

`X` is the number you want to modify these values to; `4X` means four times the value you choose for `X`. Because these values are set to their defaults at boot time, you need to add these commands to `/var/etc/rc.local` so they are changed at each startup.

On Solaris, add the following line to `/etc/system` and reboot:

```
set rlim_fd_max = X
```

On Linux, you need to do two things. In `/etc/security/limits.conf`, add the following lines:

```
* soft nofile 1024
* hard nofile X
```

These lines allow users to set their own file descriptor limits on login. You also need to change the system-wide limits by executing the following commands. (Add these to a startup script to be done on each reboot.)

```
# echo X >/proc/sys/fs/file-max
# echo 3X >/proc/sys/fs/inode-max
```

Troubleshooting Issues with the HTTP Security Server

Many of the following issues also apply to authentication because Security Servers are used for authentication. In this subsection, I talk about how to resolve common problems with the HTTP Security Server. A separate section on gathering debug information from Security Servers (Debugging the Security Servers) appears later in this chapter.

9.12: The HTTP Security Server Won't Work

A Security Server cannot share the same port as another application. For example, if you are using the HTTP Security Server bound to a specific port (say, port 80) and you have something else bound to that port (such as Voyager on a Nokia platform), one of the services must be moved.

9.13: My Users See the Error Message "FW-1 at Kyle: Unknown WWW Server"

This message could mean a few different things.

- The URL typed was incorrect.
- The firewall is not configured to use DNS for name resolution. The HTTP Security Server requires that the firewall be configured to use DNS.
- FireWall-1 timed out when it attempted to look up the name for the site.
- Your DNS server is configured to cache negative responses to DNS requests so that the same request is not made again. The client may also be running a name service-caching daemon that does something similar. You may want to consider disabling these features or setting the timeouts sufficiently high so that proper time is given to resolve the DNS queries.

If desired, you can change the error message text as described in FAQ 9.11.

9.14: My Users See the Error Message "Failed to Connect to WWW Server"

There are two possible reasons for this message.

- Connection to the site timed out or was refused at the remote end. In this case, you can usually do a refresh and the page will load correctly.
- The remote site either has a missing or inconsistent reverse DNS entry for its IP address.

Check Point considers the latter a security risk and does not allow these sites to be contacted through the HTTP Security Server. Check Point also does not allow you to turn off this feature. You have the following workaround options.

298 CHAPTER 9 • CONTENT SECURITY

- Contact the administrators of the remote site in question to ask them to fix the site's reverse DNS entry.
- Add an entry in your firewall's local host file, and have the system resolve against the host file first.
- Exclude the site in question from going through the Security Server by adding a rule above your Security Server rule that permits normal HTTP to the site.

If desired, you can change the error message text as described in FAQ 9.11.

9.15: I Have Problems When I Try to Use Internet Explorer (or Other Browsers That Support HTTP 1.1) through FireWall-1

To solve this issue, enable the following properties as described in FAQ 9.9 (Table 9.2).

```
:http_cvp_allow_chunked (true)
:http_weeding_allow_chunked (true)
:http_block_java_allow_chunked (true)
:http_allow_ranges (true)
:http_force_down_to_10 (true)
:http_sup_continue (true)
:http_avoid_keep_alive (true)
```

9.16: I Can't Access Certain Web Sites through the HTTP Security Server

Various sites have issues when they are accessed via the HTTP Security Server. If you've enabled the properties suggested previously and are still having problems, do not use the HTTP Security Server for these sites. Place a rule that permits access to these sites above any rule that uses the HTTP Security Server.

9.17: The Memory Usage of `in.ahhttpd` Keeps Growing

In just about every version of the HTTP Security Server that I've seen, heavy use of the HTTP Security Server seems to cause the process to grow without bounds until the system crashes. This and performance issues are the reasons I hesitate to recommend that large sites use the HTTP Security Server. You may have to write a script to monitor `in.ahhttpd`'s memory usage and kill this process when it grows beyond a certain limit (25MB is the limit several of my customers have used).

The FTP Security Server


The FTP Security Server is used to restrict people from uploading or downloading files as well as to virus scan all FTP file transfers. The FTP Security Server is enabled when the following situations are true.

- There is a line that permits `in.ftpd` to start up in `$FWDIR/conf/fwauthd.conf`. This line is usually present by default.
- A valid resource is defined in your security policy or in a User Authentication rule involving FTP.

The proper line for the FTP Security Server in `$FWDIR/conf/fwauthd.conf` looks like this (with no comment character, #, at the beginning of the line):

```
21      fwssd          in.ftpd      wait      0
```

If this line is not present or is commented out, the FTP Security Server will not run, and any process that relies on it will fail.

To filter FTP, you need to create a resource of type FTP and use it in the rulebase. Let's create a resource called `ftp_downloads` to allow FTP downloads through the HTTP Security Server. From SmartDashboard/Policy Editor, select Manage and then Resources. Next select New, and choose URI. You may also click on the  icon in the objects tree, right-click on FTP, and select New FTP. Then create a new resource of type FTP, as shown in Figure 9.18.

The General tab is fairly self-explanatory, so let's move on and look at the Match tab, shown in Figure 9.19.

Path refers to a specific location on the FTP server. For instance, you could allow some people to upload to a specific directory but deny that directory to others. An example of this is shown in the Sample Configurations section later in this chapter.

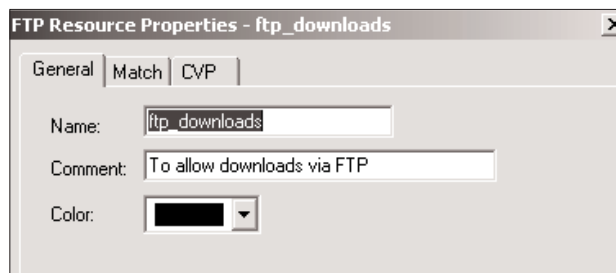


Figure 9.18 FTP Resource Properties, General tab

300 CHAPTER 9 • CONTENT SECURITY

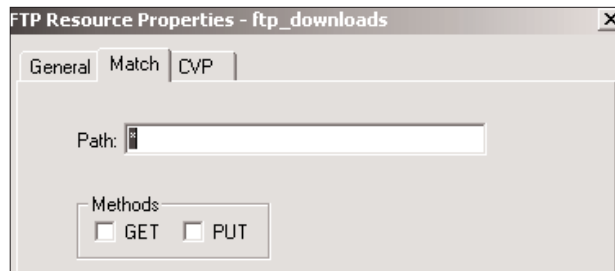


Figure 9.19 FTP Resource Properties, Match tab

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|------------------|-------------|------------------|--------|-------|
| internal_network | * Any | ftp->ftp_downloa | accept | Log |

Figure 9.20 Sample FTP rule with CVP

You can match two types of methods: GET and PUT. Aside from matching the GET command, allowing GET commands also allows RETR, RNFR, and XMD5 commands. Aside from matching the PUT command, allowing PUT commands also allows STOR, STOU, APPE, RNFR, RNTD, DELE, MKD, and RMD commands. Most other commands are passed to the FTP server for execution.¹

The CVP tab is where you specify the CVP server to use, if any. This is similar to what was shown earlier in Figure 9.5 except that the CVP tab under FTP Resource Properties excludes the HTTP-specific options.

After setting these properties, you can add a rule with this resource to the rulebase, as shown in Figure 9.20.

Frequently Asked Questions about the FTP Security Server

9.18: Why Won't the FTP Security Server Let Me Use Certain FTP Commands?

The following commands are enabled by default:

```
USER PASS ACCT REIN BYE QUIT BYTE SOCK PASV TYPE STRU MODE PORT
RETR STOR STOU APPE ALLO REST RNFR RNTD ABOR DELE LIST NLST SITE
MLFL MAIL MSND MSOM MSAM MRSQ MRCP CWD PWD RMD MKD HELP NOOP CDUP
SYST XMKD XCWD XRMD XPWD XCUP XMD5 FIND MDTM SIZE MACB FW1C
```

1. For details on these and other FTP commands, see RFC959, which you can obtain from <http://www.rfc-editor.org>, among other places.

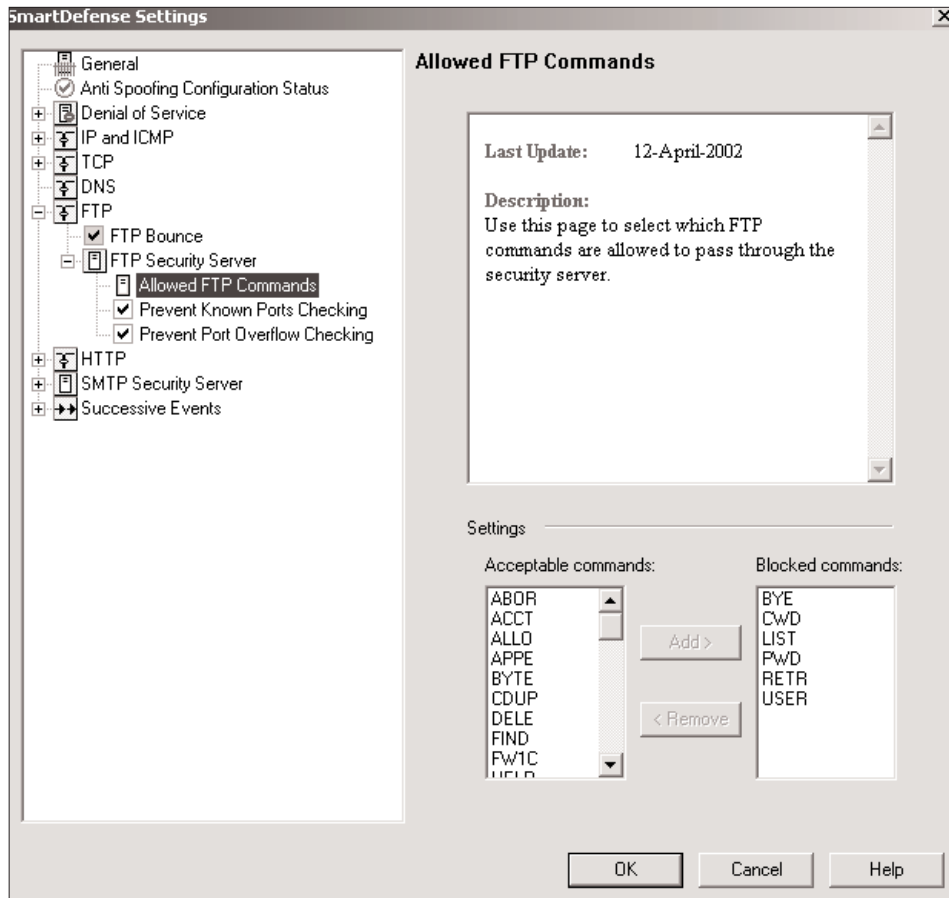


Figure 9.21 SmartDefense Settings, Allowed FTP Commands frame

The list of allowed commands is stored in the property `ftp_allowed_cmds`, which can be edited with **dbedit** or by manually editing `$FWDIR/conf/objects_5_0.c` on the management console. (See FAQ 4.2 for guidelines on how to edit `objects_5_0.c`.) You can also edit this property in SmartDefense, available in FireWall-1 NG FP2 with SmartDefense supplement or NG FP3 and later. Figure 9.21 shows how to do this.

9.19: Why Do I Always Have Problems with Certain Sites When Using the FTP Security Server?

When the user issues a **get**, **put**, **delete**, **mkdir**, or **rename**, the FTP Security Server issues a **PWD** command in order to get the full path. The FTP server must respond to the **PWD** command with a 257 message, which, according to RFC 959, must contain the absolute path in quotes. If the **PWD** command is

302 CHAPTER 9 • CONTENT SECURITY

disabled on the remote server or the PWD does not respond in the correct manner, the FTP Security Server will deny the request. See the previous question for how to allow PWD replies without quotes.

9.20: Why Do I Have a Problem FTPing to Any Site with the FTP Security Server?

If name service caching is occurring, particularly if the name server is caching negative responses to DNS requests, the FTP Security Server will have a problem. This may occur on the DNS server itself or on the firewall (e.g., `nscd`). The problem with the specific site should resolve once the cached entry expires or you disable name service caching.

The SMTP Security Server

The SMTP Security Server is used to prevent certain types of mail from passing your gateway. The SMTP Security Server is enabled when the following situations are true.

- There is a line that permits `in.asmtpd` to start up in `$FWDIR/conf/fwauthd.conf`. This line is present by default.
- A valid resource is defined and used in your security policy.

The proper line for the SMTP Security Server in `$FWDIR/conf/fwauthd.conf` looks like this (with no comment character, `#`, at the beginning of the line):

```
25          fwssd          in.asmtpd          wait          0
```

If this line is not present or is commented out, the SMTP Security Server will not run, and any process that relies on it will fail.

The SMTP Security Server acts a bit differently than the other Security Servers. There are actually two separate processes involved: `in.asmtpd` and `mdq`. `in.asmtpd` intercepts SMTP connections and spools the messages to disk. That is all they do. The `mdq` process periodically scans the spool directory and delivers the messages to their final destinations, performing the necessary filtering, header rewriting, and content rewriting. This is more secure than attempting to do everything in one process. Figure 9.22 shows a diagram of this.

SMTP Security Server Parameters

Aside from using resources, the SMTP Security Server has parameters that are configured in `$FWDIR/conf/objects_5_0.C`. Descriptions of these parame-

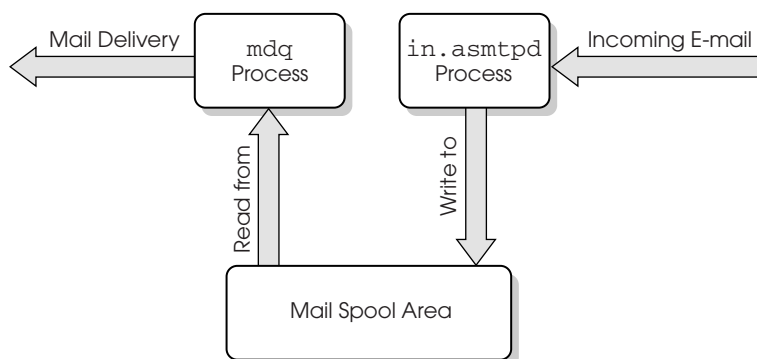


Figure 9.22 How in.asmtpd and mdq interact

ters and their default settings follow, using their names as shown in `objects_5_0.C`. A few of these parameters can be modified via the gateway object definition in the SMTP frame (under the Advanced frame; see Figure 9.23).

The items you can modify in `objects_5_0.C` include those listed below.

timeout: This is the amount of time (in seconds) FireWall-1 will spend on CVP scanning a message and delivering it to the Mail Transfer Agent (MTA). It is recommended that this value be at least 90 seconds (the default setting), if not longer. If you change this parameter, it is recommended that you reboot in order for it to take effect. (Corresponds to “Connection timeout” in Figure 9.23.)

scan_period: This is the amount of time (in seconds) that mdq will check the spool directory for e-mail to be delivered. By default, mdq checks for mail every 2 seconds. Mail found during this scan is delivered. mdq will take a number of e-mails based on the `max_conns` and `max_conns_per_site` parameters. Once `max_load` is reached, mdq stops taking mail messages from the spool directory. (Corresponds to “Dequeuer scan period” in Figure 9.23.)

resend_period: This is the amount of time (in seconds) that a message that previously failed to be delivered will be resent from the SMTP Security Server. The default is 600 seconds (10 minutes). (Corresponds to “Mail resend period” in Figure 9.23.)

abandon_time: This is the amount of time (in seconds) that a message is allowed to live in the spool directory before FireWall-1 returns the message to the sender. The default is 432,000 seconds (5 days). The error server defined in the specific resource matched by the rule the message was accepted under is used to deliver the message. (Corresponds to “Mail abandon time” in Figure 9.23.)

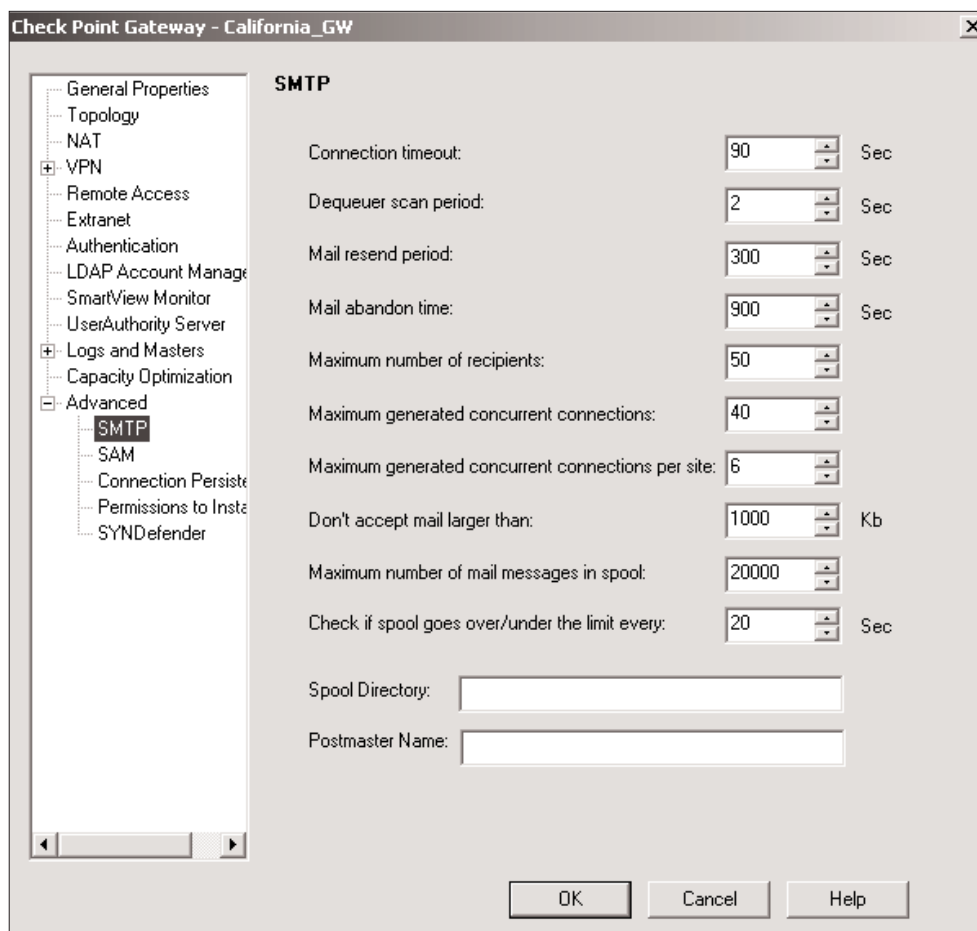


Figure 9.23 Gateway Objects, SMTP frame

rundir: This is the location where FireWall-1 spools incoming e-mail. The default is `$FWDIR/spool`. If you expect to process lots of e-mail, make sure this is on a partition with lots of disk space. Specify a path to the new spool directory (this must exist). (Corresponds to “Spool Directory” in Figure 9.23.)

spool_limit: This allows you to specify the maximum number of messages allowed in the spool directory. Incoming messages that exceed this limit will be rejected. (Corresponds to “Maximum number of mail messages in spool” in Figure 9.23.)

spool_limit_scan_period: This specifies how frequently to check to see that the `spool_limit` has been reached. (Corresponds to “Check if spool goes over/under the limit every” in Figure 9.23.)

detailed_smtp_err_mail: If this parameter is set to a value other than 0 (the default), an error mail will be generated when the SMTP Security Server cannot deliver the e-mail to some of the recipients. The error mail includes information regarding which users could not receive the original mail and a reason for each recipient. This mail message is sent only when the “Notify sender on error” flag (see the SMTP Resources subsection) is enabled in the resource that matches the incoming message. (As of NG FP3, can be tweaked only via **dbedit**.)

detailed_av_err_mail: When this parameter is set to a value other than 0 (the default) and a mail fails a Content Security check, the generated error mail will include a notification of the failure as well as the explanation message received from the Content Security Server. Note that in case of a malicious attempt to insert a virus into an organization, it may be preferable not to use this flag because it allows the generator of the mail containing the virus to receive feedback on the malicious attempt. This error mail is sent only when the “Notify sender on error” flag is enabled in the resource that matches the incoming message. (As of NG FP3, can be tweaked only via **dbedit**.)

detailed_rb_err_mail: When this parameter is set to a value other than 0 (the default) and mail message would not be allowed by the rulebase, the sender is notified. (As of NG FP3, can be tweaked only via **dbedit**.)

max_conns: This is the maximum number of connections that the SMTP Security Server will generate to SMTP servers to deliver e-mail. (Corresponds to “Maximum generated concurrent connections” in Figure 9.23.)

max_conns_per_site: This is the maximum number of connections that the SMTP Security Server will generate to a single SMTP server to deliver e-mail. (Corresponds to “Maximum generated concurrent connections per site” in Figure 9.23.)

max_mail_size: This is the maximum number of kilobytes that the SMTP Security Server will allow a message to be. (Corresponds to “Don’t accept mail larger than” in Figure 9.23.)

max_mails_per_conn: This is the maximum number of e-mails the SMTP Security Server will attempt to send on a single connection to a remote SMTP server. (As of NG FP3, can be tweaked only via **dbedit**.)

max_mx_node_per_mail: A domain can have multiple MX records associated with it. This property tells the SMTP Security Server how many MX records to attempt to use to deliver a message. (As of NG FP3, can be tweaked only via **dbedit**.)

306 CHAPTER 9 • CONTENT SECURITY

max_ips_per_mx_node: An MX record for a domain may have more than one IP associated with it. For each MX record processed by the SMTP Security Server, this is the maximum number of IP addresses it will attempt to use per MX record. (As of NG FP3, can be tweaked only via **dbedit**.)

maxrecipients: This is the maximum number of recipients an e-mail message can contain. The SMTP Security Server will reject messages that contain more recipients than specified in this parameter. (Corresponds to “Maximum number of recipients” in Figure 9.23.)

postmaster: This specifies the e-mail address of the postmaster. This e-mail address is sent copies of any messages returned for nondelivery. (Corresponds to “Postmaster Name” in Figure 9.23.)




WARNING! In FireWall-1 4.1, these parameters were changed by editing `$FWDIR/conf/smtp.conf` on the firewall module. On an NG module, this file will be overwritten on a policy installation. It is not recommended that you edit `smtp.conf` any longer.

To modify any of these parameters, you can either edit `$FWDIR/conf/objects_5_0.C` manually on the management console, use **dbedit**, or use **Database Tool**. To edit the properties in **dbedit**, use the following commands:

```
dbedit> modify network_objects craig
          firewall_settings:smtp:detailed_smtp_err_email
          true
dbedit> update network_objects craig
```

In the preceding example, `craig` is the relevant firewall object, `detailed_smtp_err_mail` is the property, and `true` is the value it was set to.

SMTP Resources

To filter SMTP, you need to create a resource of type SMTP. From SmartDashboard/Policy Editor, select Manage and choose Resources. Next select New, and then choose SMTP. You may also click on the  icon in the objects tree, right-click on SMTP, and select New SMTP. You are presented with the window shown in Figure 9.24.

The options on this tab are described below.

Name: Enter the name of the SMTP resource object (must be unique).

Comment: In this field, you can describe the resource in more detail.

Color: Select whichever color you like.

Mail Delivery Server: If the server in this field is specified and this resource is matched, the e-mail will be forwarded to this server. If nothing is specified,

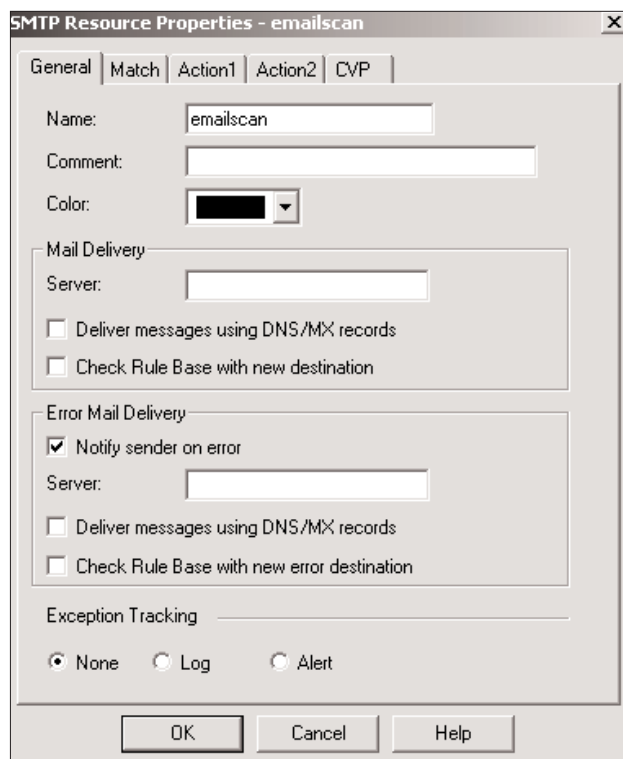


Figure 9.24 SMTP Resource Properties, General tab

the e-mail will be forwarded to the server the message was originally being sent to. To specify multiple mail servers to attempt, use the form {mailserver1, mailserver2,...}.

Error Mail Delivery Server: If the SMTP Security Server is not able to deliver the message within the abandon time and “Notify sender on error” is selected, the message will be sent through this e-mail server. If nothing is specified, the e-mail will be forwarded to the server the message was originally being sent to. To specify multiple mail servers to attempt, use the form {mailserver1,mailserver2,...}.

Notify sender on error: This setting indicates whether or not to notify the originator of the message when a delivery problem occurs. If this option is enabled, “bounce” messages will be generated if an e-mail is not successfully delivered. The level of detail of these error reports depends on `objects_5_0.C` modifications discussed in the previous subsection.

Deliver messages using DNS/MX records: This option exists for both the Mail Delivery Server and the Error Mail Delivery Server. If this option is enabled under the Mail Delivery section, Check Point will attempt to look

up the MX record for the destination address and deliver to the IP(s) specified there. If this option is enabled under the Error Mail Delivery section, error messages will be sent to the originating e-mail via the domain's MX record. The operating system of the firewall must be configured with DNS enabled. This option should be used for e-mail being delivered to the Internet. For e-mail coming in from the Internet, this option may not be necessary.

Check Rule Base with new destination: This option is also available under both Delivery sections. Normally, any e-mail initially accepted by the SMTP Security Server will be delivered by the firewall without being matched against the rulebase. You can force this delivery connection to go through the rulebase for either normal or error delivery by checking the appropriate box.

Exception Tracking: If the resource finds anomalous behavior, you can either log that information, generate an alert, or do nothing.

The Match tab is shown in Figure 9.25. The Sender and Recipient fields can be matched with any of the wildcards listed earlier in the chapter.

The Action1 tab, shown in Figure 9.26, allows you to rewrite the e-mail headers. The left part of each line is what is matched (the original); the right part is its replacement (what you are changing it to). On the original side, you can

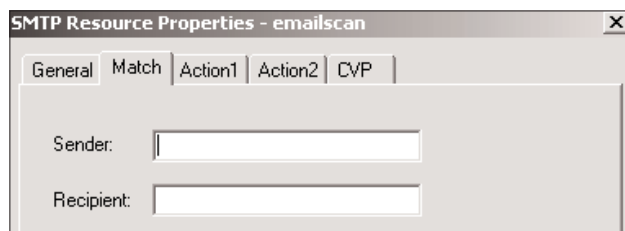


Figure 9.25 SMTP Resource Properties, Match tab

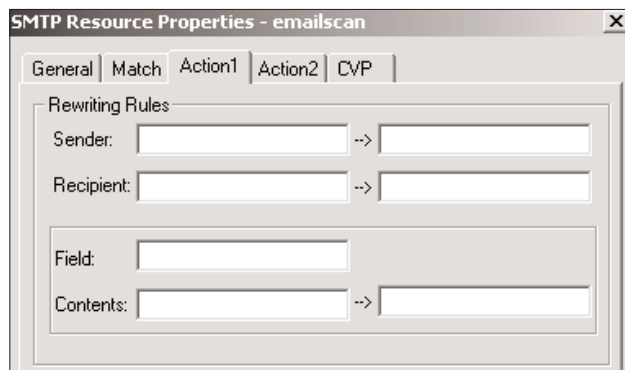


Figure 9.26 SMTP Resource Properties, Action1 tab

use normal wildcards. If you want to make what you matched part of what you translate it to, use the & wildcard to signify that. For example, if you want to rewrite all addresses of the form user@smtp.weirdal.com to user@weirdal.com, on the left side you would specify *@smtp.weirdal.com, and on the right, you would specify &@weirdal.com.

You can do this for a custom field as well. If you want to eliminate a field (say, the Received: lines), on the left enter *, and leave the right side blank.

Figure 9.27 shows the Action2 tab. If you do not want to allow certain MIME types or files with certain extensions, you can specify them on the Action2 tab. You can specify multiple MIME types or files using the normal wildcards (see Table 9.1 earlier in the chapter). Filtering on MIME types or file extensions is an inexact science at best because the e-mail client can use any MIME type it wants, and different e-mail clients use different MIME types for the same type of document. For instance, application/msword can be used for Microsoft Word documents. UNIX machines often send these kinds of documents as application/octet-stream, which can also be used for applications or

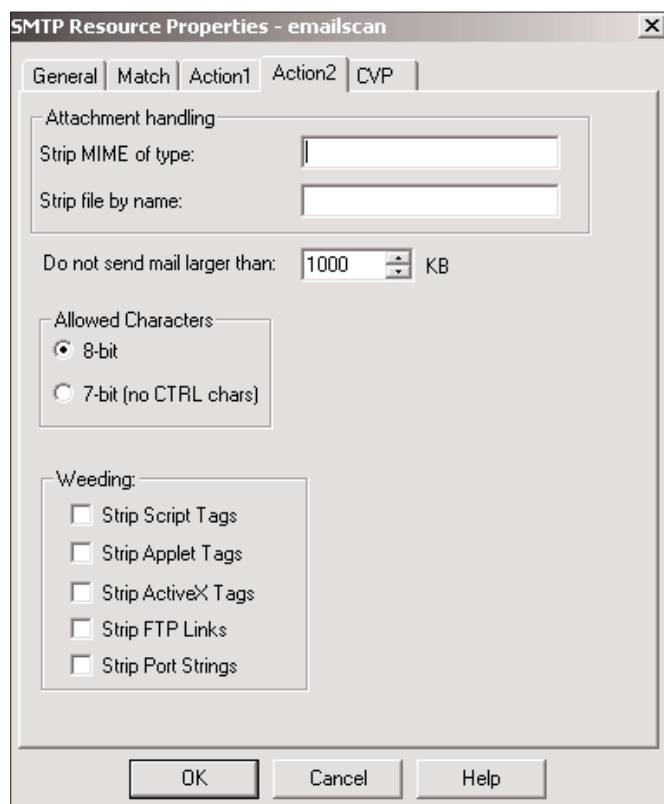


Figure 9.27 SMTP Resource Properties, Action2 tab

310 CHAPTER 9 • CONTENT SECURITY

any binary file. On the Action 2 tab you can also restrict message size and strip out undesirable HTML tags from e-mails.

Some examples of how to use the SMTP Security Server are described in the next subsection.

Frequently Asked Questions about the SMTP Security Server

The following FAQs are provided for events that may occur with the SMTP Security Server.

9.21: When I Use the SMTP Security Server, to What Should the MX for My Domain Point?

Your MX records should point to your normal SMTP server. Provided your rules are set up correctly, FireWall-1 will intercept this traffic automatically. You do not have to change anything with respect to your MX records for your domain.

9.22: Can I Have the Firewall Be the MX for My Domain?

Usually, the SMTP Security Server intercepts communications destined for an internal SMTP server. In some cases you may want to have the firewall be the mail exchanger. You can do this with the proper SMTP resource, making sure the following fields are defined.

- **Mail Delivery Server** (on the General tab): Enter the IP address of your inbound server in this field. If you have more than one SMTP server, enter each in the format {ip-address-1,ip-address-2,...}. If your DNS is correctly defined, you can use the DNS/MX records option instead.
- **Error Mail Delivery Server** (on the General tab): Select this option if you want to notify the sender that his or her message has been rejected or in case of some other problem. If your DNS is correctly defined, you can use the DNS/MX records option instead.
- **Don't accept mail larger than** (on the SMTP frame): This option should be set appropriately. The default is 1,000KB (or roughly 1MB).

Once you have defined the resource, add a rule similar to the one shown in Figure 9.28, and reinstall the security policy.

Note that your SMTP Server should be responsible for delivering outbound messages. See the next question for more details.

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|--------|--|---|--|---|
| * Any |  firewall |  smtp->store_and_forward |  accept |  Log |

Figure 9.28 SMTP Security Server rule


















| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|--|--|---|--|---|
| * Any |  smart-smtp-server |  smtp |  accept |  Log |
|  smart-smtp-server |  Internal-Mail-Server |  smtp->smtp_inbound_from_dmz |  accept |  Log |
|  smart-smtp-server | * Any |  smtp |  accept |  Log |
|  internal-smtp-server | * Any |  smtp->smtp_outbound |  accept |  Log |

Figure 9.29 Rules for a “smart” SMTP server

9.23: Why Won't the SMTP Security Server Use the MX Records?

You might think this FAQ was meant for the previous version of my book, where I covered FireWall-1 4.1 and earlier. It's true that the SMTP Security Server did not use MX records and thus was unsuitable for use in delivering outbound e-mail. Sadly, even if the MX record option is enabled, it may not work correctly for all domains. Check Point released hotfixes for this issue in NG FP2 and possibly other releases.

A workaround, and what I consider to be “best practice,” is to use a “smart” SMTP server—that is, a conventional SMTP server that knows how to properly handle MX records. This SMTP server should probably be in your DMZ. You have to make sure that any SMTP traffic from this host destined to the Internet does not get processed by any SMTP resources. To ensure that this does not happen and to have the most secure setup, use the rules shown in Figure 9.29.

The first rule matches all SMTP traffic from the Internet to the smart SMTP server (which is the MX for your domain). The second rule forces all SMTP traffic from the smart SMTP server to the internal network through the SMTP Security Server using the smtp_inbound_from_dmz resource. This resource should have the mail server configured to point to the internal SMTP server(s) and perform any necessary checking. The third rule allows the smart SMTP server to communicate to any SMTP server without going through the SMTP Security Server. The fourth rule forces your internal SMTP server to communicate through the SMTP Security Server. This last rule is optional, but you should use it if you want to perform virus scanning on any outbound e-mail.

9.24: Can I Use the SMTP Security Server to Help Fight Incoming Spam?

Spam is a notoriously difficult nuisance to filter properly. Many individuals and companies have written various programs to attempt to filter spam. Although not specifically designed to handle this task, FireWall-1 does have some features that can be used to help, namely, the SMTP Security Server.

312 CHAPTER 9 • CONTENT SECURITY

Your inbound SMTP server is likely to be a better tool to stop spam. Most SMTP servers (with the notable exception of Microsoft Exchange 5.0 and earlier) have the capability to turn off unauthorized relaying and/or implement some checks to prevent unauthorized use. You can even subscribe to the Mail Abuse Prevention System (MAPS) or a similar system that maintains a blacklist of known bad sites.

If you are going to use the SMTP Security Server to filter spam, your SMTP resources should have the recipient and the sender defined. The recipient should be `*@yourdomain.com`. If you have multiple domains, it should read `*@{yourdomain.com,yourotherdomain.com,...}`. The send should be configured with a `*` to match all incoming mail.

9.25: Can the SMTP Security Server Accept E-mails of Any Size?

You should be able to type a number into the appropriate field. Keep in mind that you need to have sufficient disk space on your firewalls to accept these e-mails.

Personally, if a user wants to e-mail a large file, I feel he or she should use FTP or something similar. All e-mailed files are encoded in a 6-bit format for transmission over SMTP, which expands the file size by at least 33%.

9.26: When Does CVP Get Performed on E-mails in the SMTP Security Server?

This action is actually done as the SMTP message is being delivered to the remote SMTP server. This means the timeout value specified in the SMTP configuration on the gateway object is long enough for both tasks to occur.

Troubleshooting the SMTP Security Server

The following solutions are provided for problems that occur with the SMTP Security Server.

9.27: I See the Message "Connection to Final MTA Failed" in the SmartView Tracker/Log Viewer

This message shows up when the SMTP Security Server is unable to connect to the remote MTA (i.e., the SMTP server) to deliver that message. This may simply be a transient problem, or the remote SMTP server may simply no longer exist. The next subsection shows you how to resolve that problem.

9.28: Mail Appears to Get Stuck in the SMTP Security Server Spool Directory

Make sure your rules are ordered in such a way as to not use the SMTP Security Server when not necessary. The following subsections contain some other events that will cause messages to back up.

Maximum Concurrent Sent Mails Limit Reached

You might see the following message in the `mdq.elg` file. It means that the number of recipients specified in the mail header has exceeded the limit.

```
[195@firewall] max concurrent sent mails limit reached -
pool scanning stops
```

The number of messages that can be sent at one time is specified with the `max_conns` and `max_conns_per_site` parameters, described in the SMTP Security Server Parameters subsection earlier in this chapter.

Messages Continue to Fail on Delivery Attempts

If sending an e-mail continues to fail (or takes a long time), the whole spool will be held up until the troubled e-mail times out. Decreasing the `abandon_time` parameter from its default of 432,000 seconds (5 days) to 3,600 seconds (1 hour) clears out the spool more quickly, returning error messages to the sender stating that the e-mail could not be delivered. (See the SMTP Security Server Parameters subsection earlier in the chapter.)

Forcing the Queue to Empty Out

You can force a queue run by using the command **fw mdq**. However, the only way to fix this (short of letting FireWall-1 attempt to deliver these messages until they expire) is to edit each individual file in `$FWDIR/spool` (or whatever `rundir` is set to), so that the destination it tries to contact is an SMTP server that will deliver the message. There are actually three directories under `$FWDIR/spool`, which contain the individual files.

- **D_resend**: This directory contains all the messages that the SMTP Security Server had problems delivering in the initial attempt. The SMTP Security Server attempts to resend messages from this directory until the message has been around longer than the `abandon_time`. Files in this directory are of the form `Rxxxxxxx` (for messages not yet expired) or `Exxxxxxx` (for messages that have expired).
- **D_state**: This directory contains all the messages currently being received by the SMTP Security Server, that is, partial e-mails. Once the messages are received fully, they are copied into the `D_sender` directory.

314 CHAPTER 9 • CONTENT SECURITY

- **D_sender**: This directory contains messages that the SMTP Security Server has fully received and is about to send for the first time. If a message is not successfully sent, it is copied over to the **D_resend** directory.

In all cases, the beginning of a spool file looks like this:

```

AV_SETTING:      none
AV_IPADDR:       0.0.0.0
AV_PORT:         0
AV_HEADERS:      0
COMPOUND:        0
SRC:             192.168.230.24
SPORT:           3446
DST:             10.158.5.2
DPORT:           25
ERR_SERVER:      10.158.5.2
RULE:            11
RULEACT:         16
ERRMAIL:         0
ACCT:            0
LOG_OK:          MDQ_LOG
LOG_BAD:         MDQ_LOG
LOG_ERR:         MDQ_ALERT

```

The headers are described as follows.

AV_SETTING (none|check|cure): A header that indicates whether or not this message will be checked for viruses. **none** means do not scan, **check** means do not send if it contains a known virus, and **cure** means clean out any known viruses in the message before sending.

AV_IPADDR (a.b.c.d): The IP address of the antivirus scanner. This should list 0.0.0.0 if no virus checking is to be done.

AV_PORT (n): The port used for CVP, normally 18181 or 0.

AV_HEADERS (0|1): A header that indicates whether or not the CVP server is to be sent the SMTP headers. If so, then this is set to 1, otherwise 0.

COMPOUND (?): Unknown, but usually set to 0.

SRC (a.b.c.d): The source IP address of the machine that sent the e-mail.

SPORT (n): The source port used for the connection to the SMTP Security Server from the source IP address listed.

DST (a.b.c.d): The destination IP address of the machine to send the e-mail to. This is the IP address of the mail server that the source IP was originally trying to send to unless the matching resource or the `default_server` parameter is set to some other value, in which case, this value

reflects that instead. This is the IP address that FireWall-1 will use to deliver this message.

DPORT (n) : The destination port that will be used to send the mail to. Normally this is 25.

ERR_SERVER (a.b.c.d) : The SMTP server that will be used to attempt to deliver the failure notice if a message has been in \$FWDIR/spool beyond the abandon_time. If no error_server is defined in the appropriate resource, this will be the same as the SRC setting.

RULE (n) : The rule number by which this message was originally accepted. When the message is actually delivered, this is the rule number that will be logged.

RULEACT (?) : Unknown. The value of this is usually 16.

ERRMAIL (0|1) : A header that determines whether an actual error e-mail is sent out if the mail errors out. This corresponds to the “Notify sender on error” checkbox in the appropriate resource. The value of this header is 1 if it will attempt to deliver an error, 0 if not.

ACCT (0|1) : A header that indicates whether to perform accounting tracking on this connection (value of 1) or not (value of 0).

LOG_OK (MDQ_LOG|MDQ_ALERT) : A header that specifies how to log the successful delivery of a message.

LOG_BAD (MDQ_LOG|MDQ_ALERT) : A header that specifies how to log an unsuccessful delivery attempt (e.g., a transient failure). This corresponds to Exception Tracking on the appropriate resource (see Figure 9.23 earlier in the chapter). This line will not exist in the file if Exception Tracking is set to None.

LOG_ERR (MDQ_LOG|MDQ_ALERT) : A header that specifies how to log that an error mail needs to be sent (because the message has passed the abandon time). This corresponds to the “Notify sender on error” checkbox in the appropriate resource.

Below these entries in the spool file, you will find two sets of e-mail headers: the original headers (as they were sent with the message) and the modified headers, which are what FireWall-1 will actually send to the remote SMTP server.

You can move the R file with the lowest numerical value of the spool directory, enabling the mdq process to work with the next file. The idea is that you can either modify or relocate the problem files, enabling the forwarding of the rest of the spool directory.

Once you have done this to all the files, you can type either

```
# fw mdq
```

316 CHAPTER 9 • CONTENT SECURITY

or

```
# fw kill fwd
```

to force the SMTP Security Server to reprocess the mail spool.




WARNING! `fw kill fwd` kills the `fwd` process, which will prevent FireWall-1 from logging, using the Security Servers, or performing encryption tasks. The `cpwatchdog` process should restart `fwd` within a minute or so of executing this command.

The TCP Security Server

The TCP Security Server allows you to perform Content Security on any TCP service by sending the raw data stream to the CVP server. The CVP server inspects the content stream and returns the results to the TCP Security Server, which then takes the specified action in the resource. The TCP Security Server can also do URL filtering, though only the destination IP address will be sent to the UFP server.

CVP with the TCP Security Server

To have the TCP Security Server perform CVP on a particular TCP service, start by creating a new TCP Resource. From SmartDashboard/Policy Editor, select Manage and then Resources. Next select New, and choose SMTP. Or you may click on the  icon in the objects tree, right-click on TCP, and select New TCP. You are presented with the window shown in Figure 9.30. Give the resource a name (`tcp-virusscan` is used in this example), and set the type to CVP.

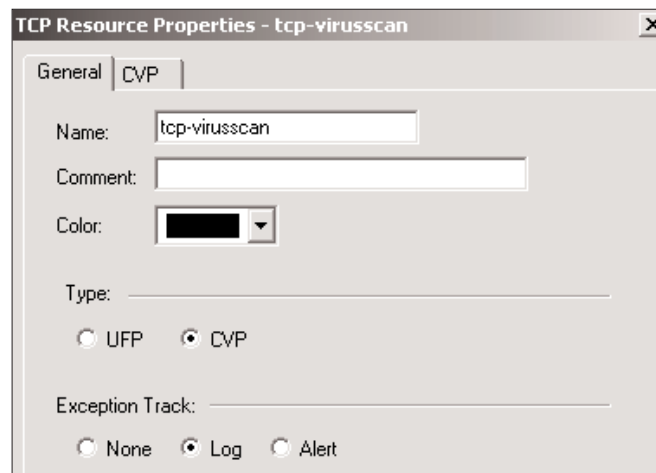


Figure 9.30 TCP Resource Properties, General tab

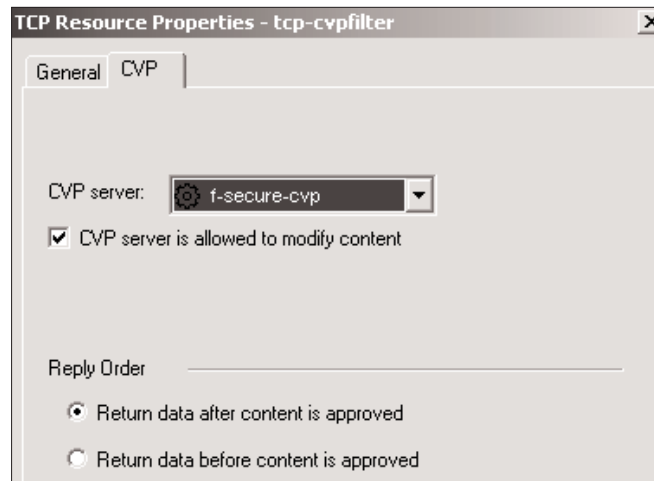


Figure 9.31 TCP Resource Properties, CVP tab

| SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|------------------|-------------|------------------------|--------|-------|
| internal_network | * Any | TCP AOL->tcp-cvpfilter | accept | Log |

Figure 9.32 Sample rule with TCP resource

Figure 9.31 shows the CVP tab. The options here are similar to those described in the HTTP Security Server section. Once you've created the resource, edit the TCP service on which you want to enable CVP scanning. In the Advanced configuration, check the "Enable for TCP resource" box.

Create a rule similar to the one shown in Figure 9.32, and push the security policy.

You must now configure the firewall module to listen on the TCP service port in question. For AOL, used in this example, the port is 5190. To do this, edit `$FWDIR/conf/fwauthd.conf` on the firewall module and add the following line to this file:

```
5190      fwssd      in.genericcd      wait      0
```

Once you have done that, bounce the `fw` process with the command **fw kill fwd**.



WARNING! `fw kill fwd` kills the `fwd` process, which will prevent FireWall-1 from logging, using the Security Servers, or performing encryption tasks. The `cpwatchdog` process should restart `fwd` within a minute or so of executing this command.

UFP with the TCP Security Server

Before proceeding with this, make sure that your UFP server can handle IP-based URLs (not all servers can).

The steps to use the TCP Security Server with UFP scanning are almost identical to the steps needed for CVP, with the following exceptions:

1. Create the TCP resource as type UFP instead of CVP.
2. You do not need to modify the firewall module to listen on the specified TCP port.

General Questions about the Security Servers

Most of the questions regarding the Security Servers are specific to each Security Server used. However, a few questions relate to the use of Security Servers in general. These FAQs cover such issues.

9.29: Why Don't the Connections I Make through the Security Servers Appear to Originate from the Firewall?

If you're familiar with FireWall-1 4.1 and earlier versions, you know that whenever the Security Servers were used, all connections through them appeared to come from the firewall. Some administrators designed their networks around this feature. In FireWall-1 NG, the firewall no longer appears to originate these connections, so designs that rely on packets originating from the firewall no longer hold true. To get the old behavior back, do one of two things:

- Create NAT rules hiding the communications behind the firewall's IP address. (NAT is discussed in Chapter 10.)
- Change a few parameters by using **dbedit** or by manually editing `objects_5_0.c`. (See FAQ 4.2 for details.) The parameters in question are in the firewall object properties: `http_transparent_server_connection`, `ftp_transparent_server_connection`, `rlogin_transparent_server_connection`, and `telnet_transparent_server_connection`. An example of how to edit these settings is shown below. Note that `craig` is the name of my firewall object.

```
dbedit> modify network_objects craig
        firewall_settings:http_transparent_server_connection
        false
dbedit> update network_objects craig
```

The notable exception to all of this is the SMTP Security Server, which still functions as it did in FireWall-1 4.1 and earlier.

9.30: Why Is the Security Server Used Even if the Rule Matched Does Not Use a Resource?

In this situation, a prior rule using the Security Servers needed evaluation by the Security Server in order to determine whether or not the rule applied. Refer to Figure 9.11 earlier in this chapter for an example.

If the connection originates from the internal network and requires HTTP, the Security Server becomes involved because it must determine whether or not the URL meets the criteria specified in the resource. If it does, the connection is rejected. If not, the HTTP is allowed by the second rule. However, because the packet required evaluation by the HTTP Security Server to determine that it did not meet the previous rule's criteria, the HTTP Security Server processes the connection even though the second rule does not explicitly use a resource.

If you want to ensure that the Security Server is not used in a particular case, you need to place a rule above any other rules that avoids using the Security Servers (i.e., no resources or User Authentication).

9.31: Can I Mix User Authentication and Content Security?

For HTTP and FTP, yes. Simply use an action of User Authentication where it is appropriate.

9.32: Can I Mix Session Authentication and Content Security?

No, you cannot. The Security Servers are required for Content Security anyway, so it makes no sense to use the Session Authentication in conjunction with Content Security. You can certainly use Session Authentication for other services, just not the ones where Content Security is required.

Debugging the Security Servers

In FireWall-1 4.1 and earlier, in order to debug the Security Servers, you were required to set environment variables and restart the `fwd` process. In FireWall-1 NG, you can now perform debugging without restarting any processes. When these variables are set, FireWall-1 logs the information generated into the various files in `$FWDIR/log`. Each Security Server has its own file with a `.elg` extension (e.g., the HTTP Security Server has `ahhttpd.elg`, the FTP Security Server has `ftpd.elg`, and so on).

To enable debugging for the HTTP Security Server, issue the following command from your firewall module:

```
# fw debug on in.ahhttpd FWAHTTPD_LEVEL=3
```

320 CHAPTER 9 • CONTENT SECURITY

To disable debugging, issue the following command from your firewall module:

```
# fw debug off in.ahhttpd FWAHTTPD_LEVEL=3
```

To enable debugging for the other Security Servers, use similar syntax. Table 9.3 shows the variables to set for the Security Servers. You can assign the variables values of 1 through 3. The larger the number, the more verbose the debugging information.

This method permits setting only one environment variable at a time, which means multiple Security Servers cannot be debugged. If you need to debug multiple Security Servers, you need to manually set the environment variables on the command line. The following example on a UNIX-based firewall using a Bourne-type shell shows you how to enable debugging for the HTTP Security Server and the SMTP `mdq` process.

```
# fw kill fwd
# FWAHTTPD_LEVEL=3; export FWAHTTPD_LEVEL
# FWMDQ_LEVEL=3; export FWMDQ_LEVEL
# fwd
```

Table 9.3 Debug variables for the Security Servers

| Variable | Description |
|-------------------|--|
| FWAHTTPD_LEVEL | Debug information from the HTTP Security Server (<code>in.ahhttpd</code>) |
| FWAFTP_LEVEL | Debug information from the FTP Security Server (<code>in.aftpd</code>) |
| FWACLIENTD_LEVEL | Debug information from the Client Authentication daemon over Telnet (<code>in.aclientd</code>) |
| FWAHCLIENTD_LEVEL | Debug information from the Client Authentication daemon over HTTP (<code>in.ahclientd</code>) |
| FWASMTPD_LEVEL | Debug information from the SMTP Security Server receiving process (<code>in.asmtpd</code>) |
| FWMDQ_LEVEL | Debug information from the SMTP Security Server mail dequeuer process (<code>mdq</code>) |
| FWARLOGIND_LEVEL | Debug information from the rlogin Security Server (<code>in.arlogind</code>) |
| FWATELNETD_LEVEL | Debug information from the Telnet Security Server (<code>in.atelnetd</code>) |
| FWGENERICD_LEVEL | Debug information from the TCP Security Server (<code>in.genericd</code>) |

To do this on a Windows-based firewall, use the following commands.

```
> fw kill fwd  
> SET FWAHTTPD_LEVEL=3  
> SET FWMDQ_LEVEL=3  
> fwd
```

Summary

Content Security provides additional control over what can be accessed through your firewalls. The HTTP, FTP, SMTP, and TCP Security Servers provide the means to filter content in FireWall-1. Resources are what tell the Security Servers how to filter content.

CVP, a feature present in all of the Security Servers, provides a mechanism for third-party products to scan content for viruses and Trojan horses. UFP, present only in the HTTP Security Server, provides a way for third-party applications to filter users' attempts to access various Web sites.

Sample Configurations

The following subsections present three situations that build on each other as the network and the needs of the enterprise change.

SMTP Content Security

The Situation

Your company wants to gradually roll in Content Security throughout its enterprise. One of the most prevalent sources of viruses has been incoming e-mail, so the managers decide to start with e-mail. Both incoming and outgoing e-mail will be scanned. HTTP and FTP traffic will eventually be scanned. (This functionality will be added in the next two situations.) The CVP and UFP servers are sitting in the DMZ. Figure 9.33 shows the network diagram for this company.

The Goals

- The Web server in the DMZ will be accessible via HTTP from anywhere.
- The e-mail server in the DMZ will be accessible via SMTP from anywhere and can send e-mail to anywhere on the Internet.

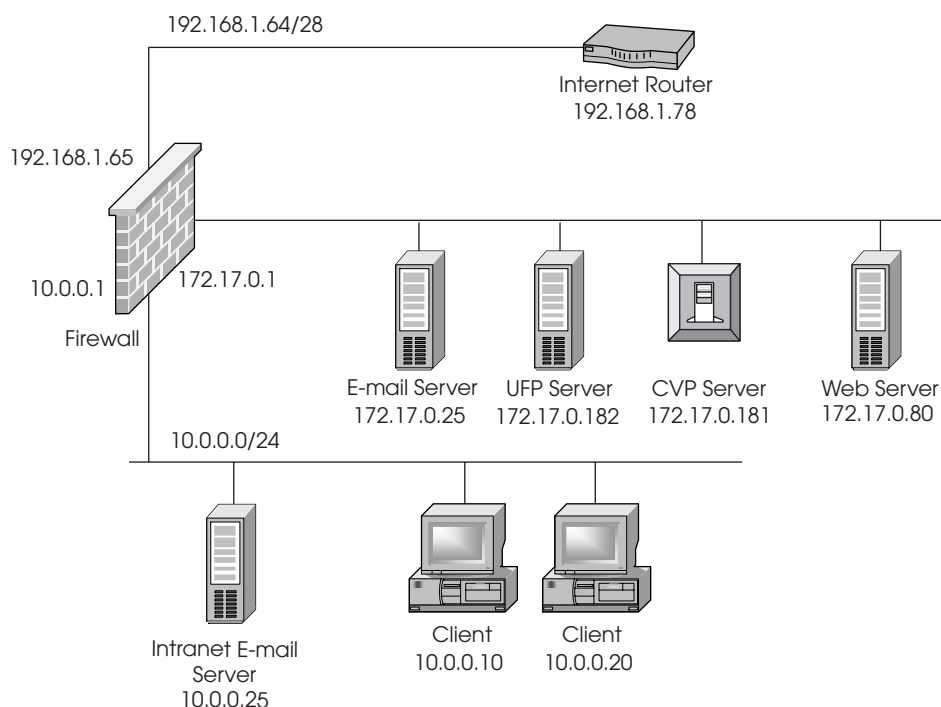


Figure 9.33 Network diagram for sample configuration

- The e-mail server in the DMZ can talk to the internal e-mail server via SMTP and vice versa. In either case, all traffic will be scanned for viruses.
- Clients on the internal network can talk to any host via HTTP and FTP.
- The UFP and CVP servers need to access their respective vendors' Web sites via HTTP to download updates (www.cvp-vendor.com and www.ufp-vendor.com).
- All other traffic should be denied.

The Checklist

- Create the necessary network objects.
- Ensure that the SMTP Security Server is enabled in `$FWDIR/conf/fwauthd.conf`.
- Create two SMTP resources to handle inbound and outbound e-mail between the internal and the DMZ e-mail servers and scan viruses.
- Create the appropriate rule(s) in the rulebase.
- Verify and install the policy.

The Implementation

The proper line for the SMTP Security Server in `$FWDIR/conf/fwauthd.conf` looks like this (with no comment character, #, at the beginning of the line):

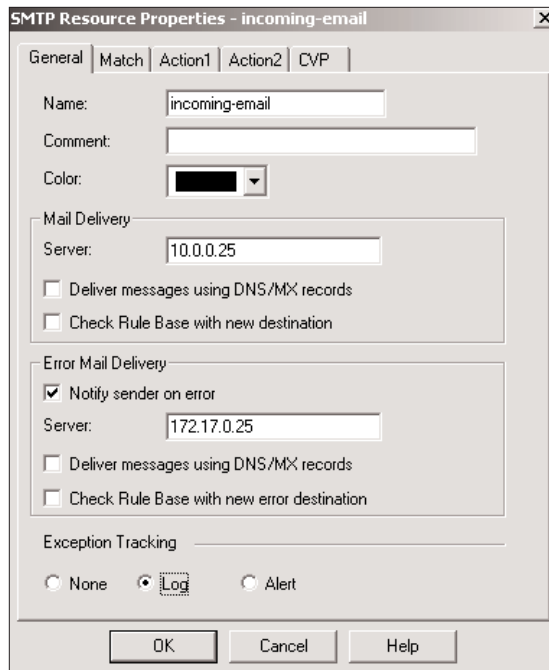
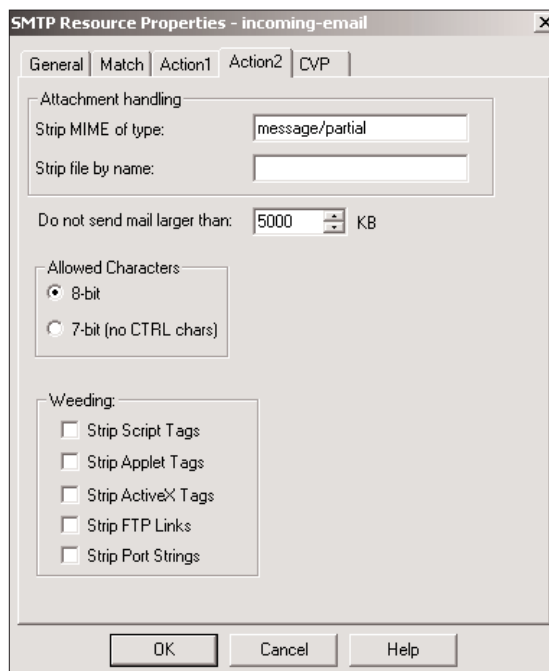
```
25          fwssd          in.asmtpd          wait          0
```

On most systems, this line is enabled by default, but it never hurts to make sure that this line exists and is not commented out.

The two resources then need to be created. The first resource is for e-mail coming from the DMZ-based e-mail server that forwards it to the internal e-mail server. All incoming e-mail should be forwarded to the internal SMTP server. Any errors generated should go to the DMZ-based e-mail server so they can be sent to the sender. Figure 9.34 shows how this resource looks.

You *can* leave the Match tab blank if you want because the internal and DMZ-based e-mail servers can do address checking on their own. For this example, let's leave the Action1 tab alone as well because there is no need to do any address rewriting.

In the Action2 tab, shown in Figure 9.35, strip out all messages of MIME type message/partial because it is used for sending a file across more than one message. This makes it difficult, if not impossible, to scan the file for viruses. The default message size should also be increased from 1,000KB to 5,000KB as a courtesy to users. However, I generally do not recommend sending large files

**Figure 9.34** Incoming e-mail SMTP resource, General tab**Figure 9.35** Incoming e-mail SMTP resource, Action2 tab

by e-mail because e-mailed files are actually *bigger* than they are if downloaded from an FTP or HTTP server. In order to e-mail binary files, they must be turned into a nonbinary format, which increases the overall file size.

Configure the CVP tab as shown in Figure 9.36.

Then create the outbound resource as shown in Figure 9.37.

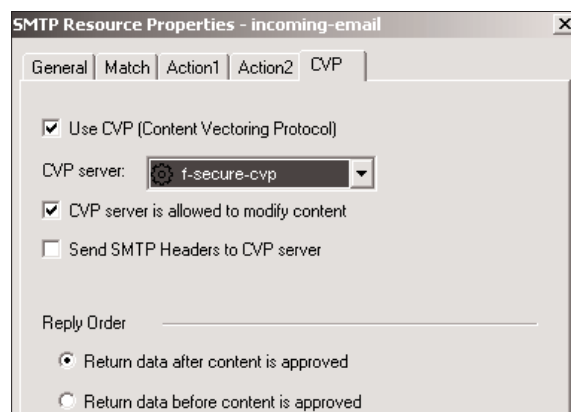


Figure 9.36 Incoming e-mail SMTP resource, CVP tab

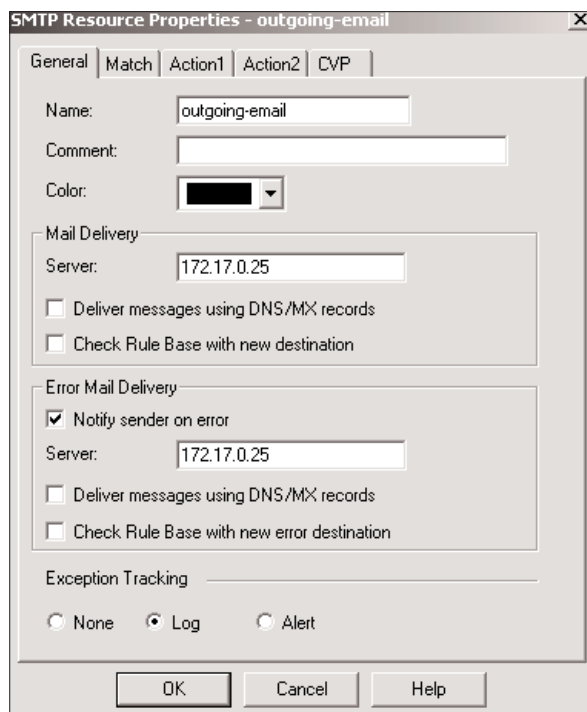


Figure 9.37 Outgoing e-mail SMTP resource, General tab

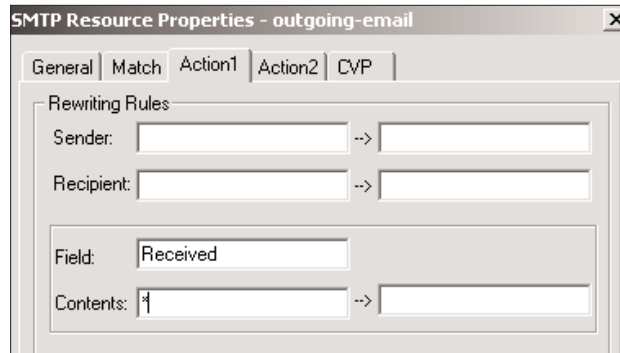


Figure 9.38 Outgoing e-mail SMTP resource, Action1 tab

| NO. | SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|-----|-----------------------|--------------------|---------------------------|--------|-------|
| 1 | * Any | Web_Server | TCP http | accept | None |
| 2 | Email_Server_Internal | Email_Server_DMZ | SMTP smtp->outgoing-email | accept | Log |
| 3 | net-10.0.0.0-24 | Email_Server_DMZ | TCP smtp | accept | Log |
| 4 | Email_Server_DMZ | net-10.0.0.0-24 | SMTP smtp->incoming-email | accept | Log |
| 5 | Email_Server_DMZ | net-10.0.0.0-24 | TCP smtp | accept | Log |
| 6 | net-10.0.0.0-24 | * Any | TCP http TCP ftp | accept | Log |
| 7 | CVP_Server | www.cvp-vendor.com | TCP http | accept | None |
| 8 | UFP_Server | www.ufp-vendor.com | TCP http | accept | None |
| 9 | * Any | * Any | * Any | drop | Log |

Figure 9.39 Rulebase for SMTP Security Server sample configuration

It might also be nice to strip out the “received” lines of messages being sent from the inside to the outside. This is done on the Action1 tab (see Figure 9.38).

The Action2 and CVP tabs should look identical to what was used for the inbound resource. In the end, the rulebase should look similar to the one shown in Figure 9.39.

Once this configuration is set up, verify and install the security policy.

FTP Content Security

The Situation

The next step in the company’s Content Security plan is to turn on Content Security for FTP. Additionally, the company wants to use the FTP Security Server to allow people to upload files only to a specific directory on the Web server.

The Goal

- Create and implement resources to perform the necessary Content Security for FTP.

The Checklist

- Ensure that the FTP Security Server is enabled in `$FWDIR/conf/fwauthd.conf`.
- Create an FTP resource to scan for viruses.
- Create an FTP resource to scan for viruses and restrict access to a specific directory on the Web server.
- Create an FTP resource to allow people to download files from the Web server.
- Modify the rulebase to use these FTP resources.
- Verify and install the policy.

The Implementation

The proper line for the FTP Security Server in `$FWDIR/conf/fwauthd.conf` looks like this (with no comment character, #, at the beginning of the line):

```
25          fwssd          in.aftpd          wait      0
```

On most systems, this line is enabled by default, but it never hurts to check.

The first resource you need to create is the more general one to match everything and allow users to upload and download as they please, with the exception that all file transfers will be scanned for viruses. This configuration is shown in Figure 9.40.

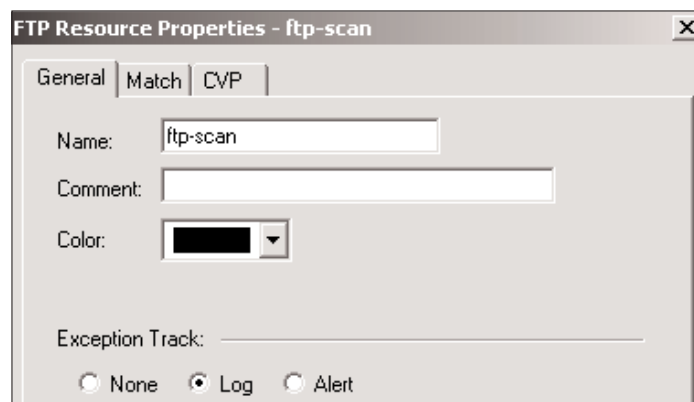


Figure 9.40 FTP scan resource, General tab

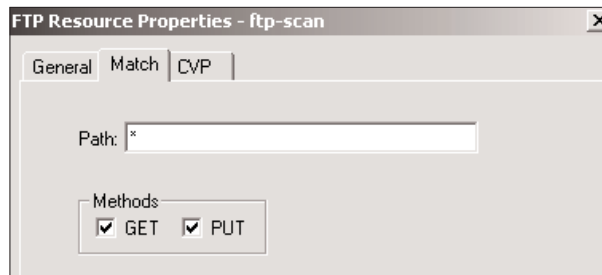


Figure 9.41 FTP scan resource, Match tab

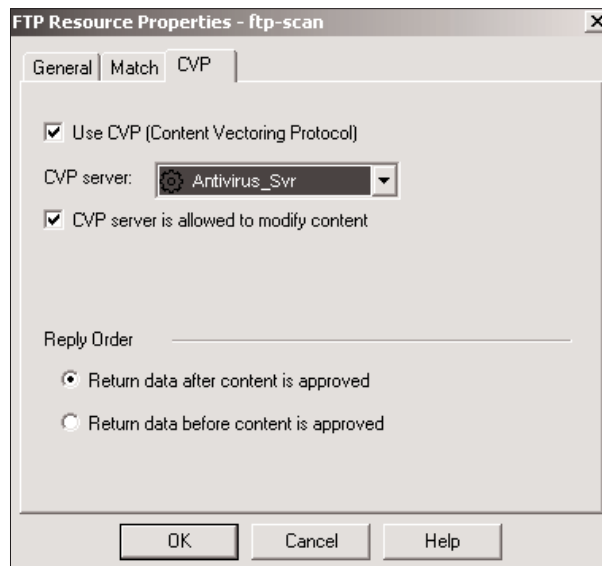


Figure 9.42 FTP scan resource, CVP tab

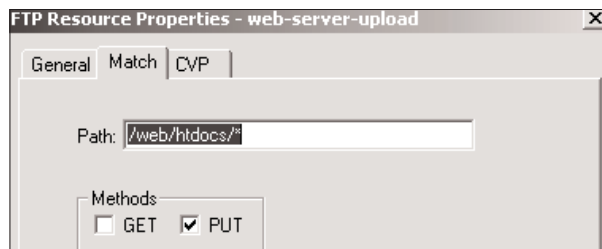


Figure 9.43 FTP Web server upload resource, Match tab

The Match tab should match all actions and paths as shown in Figure 9.41.

The CVP tab, of course, should scan for viruses (see Figure 9.42).

The Web server upload resource needs to match a specific path. Use the Match tab as shown in Figure 9.43.

You are going to virus scan anything uploaded to make sure you do not pass any viruses to the outside world. The CVP tab for this configuration is shown in Figure 9.42.

Finally, you need to create a resource to allow the internal users to download from the Web server. Because you are not terribly concerned about where on that server users download from, the Match tab should match any path for GET commands, as shown in Figure 9.44.

Once the resources are created, you can modify the existing policy to use them. The rulebase should look similar to the one shown in Figure 9.45. Rules 6, 7, and 8 were added to the previous rulebase (pushing the original Rules 6 through 9 shown in Figure 9.39 into their new positions as Rules 9 through 12 in Figure 9.45).

After updating the rulebase, verify and install the security policy.

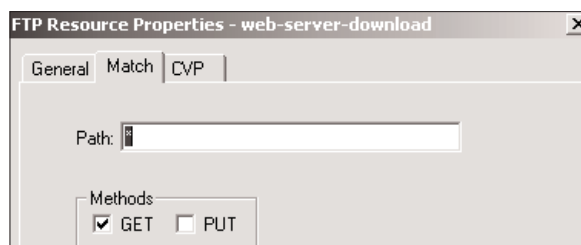


Figure 9.44 FTP Web server download resource, Match tab

| NO. | SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|-----|-----------------------|--------------------|------------------------------|--------|-------|
| 1 | ★ Any | Web_Server | TCP http | accept | None |
| 2 | Email_Server_Internal | Email_Server_DMZ | SMTP smtp->outgoing-email | accept | Log |
| 3 | ✖ net-10.0.0.0-24 | Email_Server_DMZ | TCP smtp | accept | Log |
| 4 | Email_Server_DMZ | ✚ net-10.0.0.0-24 | SMTP smtp->incoming-email | accept | Log |
| 5 | Email_Server_DMZ | ✖ net-10.0.0.0-24 | TCP smtp | accept | Log |
| 6 | ✚ net-10.0.0.0-24 | Web_Server | FTP ftp->web-server-upload | accept | Log |
| 7 | ✚ net-10.0.0.0-24 | Web_Server | FTP ftp->web-server-download | accept | Log |
| 8 | ✚ net-10.0.0.0-24 | Web_Server | FTP ftp->ftp-scan | accept | Log |
| 9 | ✚ net-10.0.0.0-24 | ★ Any | TCP http | accept | Log |
| 10 | CVP_Server | www.cvp-vendor.com | TCP http | accept | None |
| 11 | UFP_Server | www.ufp-vendor.com | TCP http | accept | None |
| 12 | ★ Any | ★ Any | ★ Any | drop | Log |

Figure 9.45 Rulebase with SMTP and FTP Security Server sample configuration

330 CHAPTER 9 • CONTENT SECURITY**HTTP Content Security*****The Situation***

The final step in implementing the company's Content Security plan is to implement both virus scanning and URL filtering for HTTP traffic.

The Goals

- Create and implement a resource for URL filtering and Content Security for HTTP.
- Make sure Content Security is *not* performed for internal users accessing the DMZ.

The Checklist

- Ensure that the HTTP Security Server is enabled in `$FWDIR/conf/fwauthd.conf`.
- Create a resource of type URI for URL filtering for HTTP.
- Create a resource of type URI that matches all URLs and does virus scanning.
- Modify the rulebase to use the HTTP resources.
- Verify and install the policy.

The Implementation

The proper line for the HTTP Security Server in `$FWDIR/conf/fwauthd.conf` looks like this (with no comment character, #, at the beginning of the line):

```
25          fwssd          in.ahttpd          wait      0
```

On most systems, this line is enabled by default, but it never hurts to check.

You first need to create a resource to filter URLs, as shown in Figure 9.46.

The Match tab should match the categories of Web sites you do not want the employees to view (see Figure 9.47).

Configure the settings on the Action tab (see Figure 9.48) to redirect rejected URLs to a policy page.

This completes the setup for the URL filtering resource. You then need to do virus scanning on any URL that is accepted. A second resource must be created, as shown in Figure 9.49.

The Match tab settings should match all URLs, and the CVP tab settings should filter viruses (see Figures 9.50 and 9.51, respectively).

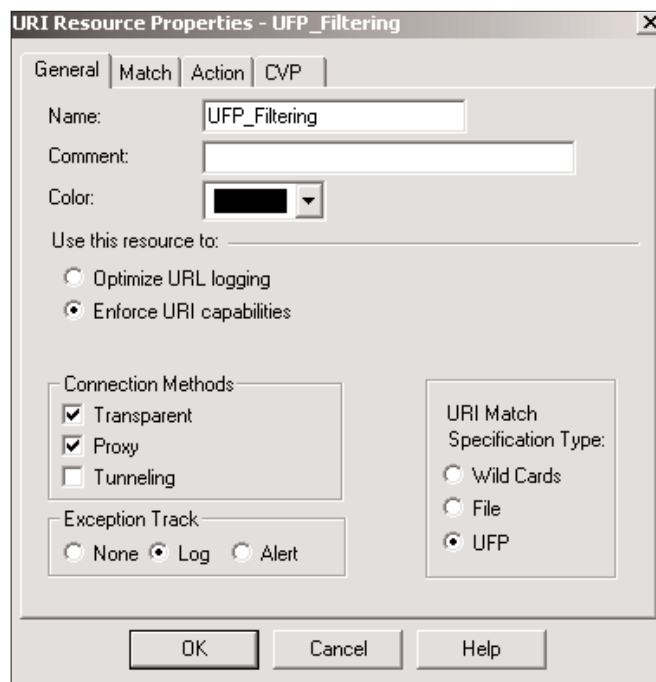


Figure 9.46 URI-filtering resource, General tab

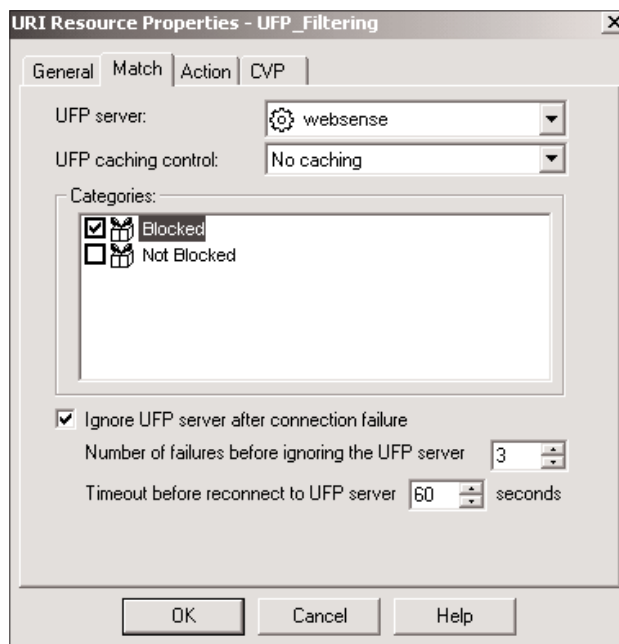


Figure 9.47 URI-filtering resource, Match tab

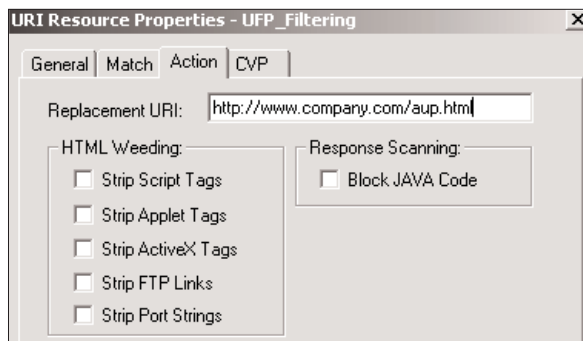


Figure 9.48 URI-filtering resource, Action tab

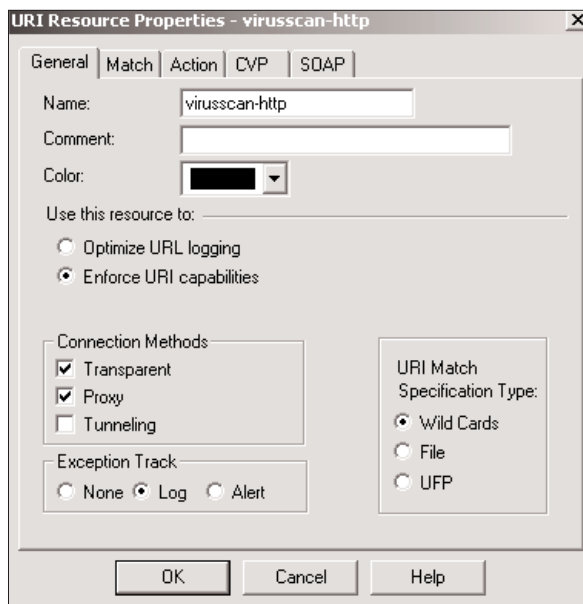


Figure 9.49 HTTP virus-scanning resource, General tab

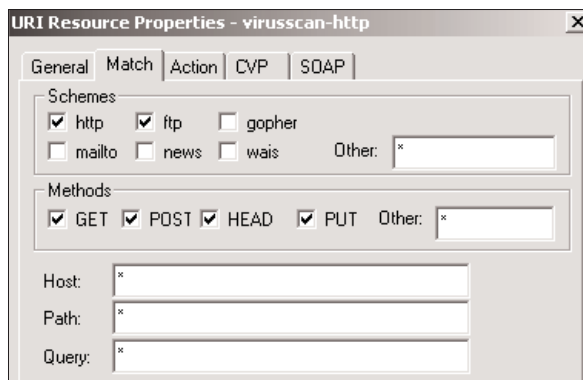


Figure 9.50 HTTP virus-scanning resource, Match tab

You then implement these resources in the rulebase, as shown in Figure 9.52. Rule 9 in the previous rulebase was replaced by a different Rule 9, and a new rule was added in the Rule 10 position, shifting down the remaining rules.

After making these changes, verify and install the policy.

Notes

Rule 1 also matches internal users' attempts to access the Web server on the DMZ.

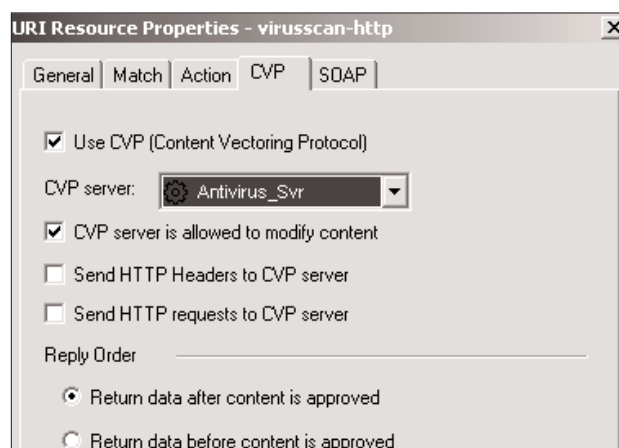


Figure 9.51 HTTP virus-scanning resource, CVP tab

| NO. | SOURCE | DESTINATION | SERVICE | ACTION | TRACK |
|-----|-----------------------|--------------------|------------------------------|--------|-------|
| 1 | ★ Any | Web_Server | TCP http | accept | None |
| 2 | Email_Server_Internal | Email_Server_DMZ | SMTP smtp->outgoing-email | accept | Log |
| 3 | ✖ net-10.0.0.0-24 | Email_Server_DMZ | TCP smtp | accept | Log |
| 4 | Email_Server_DMZ | net-10.0.0.0-24 | SMTP smtp->incoming-email | accept | Log |
| 5 | Email_Server_DMZ | ✖ net-10.0.0.0-24 | TCP smtp | accept | Log |
| 6 | net-10.0.0.0-24 | Web_Server | FTP ftp->web-server-upload | accept | Log |
| 7 | net-10.0.0.0-24 | Web_Server | FTP ftp->web-server-download | accept | Log |
| 8 | net-10.0.0.0-24 | Web_Server | FTP ftp->ftp-scan | accept | Log |
| 9 | net-10.0.0.0-24 | ★ Any | HTTP http->UFP_Filtering | reject | Log |
| 10 | net-10.0.0.0-24 | ★ Any | HTTP http->virusscan-http | accept | Log |
| 11 | CVP_Server | www.cvp-vendor.com | TCP http | accept | None |
| 12 | UFP_Server | www.ufp-vendor.com | TCP http | accept | None |
| 13 | ★ Any | ★ Any | ★ Any | drop | Log |

Figure 9.52 Rulebase with SMTP,FTP, and HTTP Security Servers sample configuration

