Open Security (OPSEC) and Content Filtering

Solutions in this Chapter:

- OPSEC Applications
- Content Vectoring Protocol
- URI Filtering Protocol
- Application Monitoring
- Client Side OPSEC Applications
- Other Resource Options

☑ Summary
☑ Solutions Fast Track
☑ Frequently Asked Questions
Introduction

Check Point's Open Platform for Security (OPSEC) model enables you to implement third-party vendor applications into your firewall environment. Based on open protocols, the OPSEC model enables vendors to easily design their applications to conform to this standard, and therefore interoperate with the VPN-1/FireWall-1 product.

You may be asking how this can benefit you? The most notable examples are your content filtering options. You can use other vendors’ virus scanners that support the Content Vectoring Protocol (CVP) (for example, Aladdin’s eSafe Protect Gateway) to easily implement virus scanning of Simple Mail Transfer Protocol (SMTP) mail, Hypertext Transfer Protocol (HTTP), and/or File Transfer Protocol (FTP) traffic, just by adding some objects and rules to your Security Policy.

Other content-filtering applications use Website databases, which are broken into categories, so that you can easily block your users from going to specific sites, such as adult entertainment, shopping and chat sites, while on the job. Several schools that provide Internet access for their young students utilize this technology to prevent them from accessing certain categories that are considered inappropriate for children.

We will talk about other OPSEC applications, and show you how to configure CVP and UFP (Universal Resource Identifier (URI) Filtering Protocol) applications in this chapter, and also how you can use the resources available in Check Point VPN-1/FireWall-1 (CP VPN-1/FW-1) to implement limited content filtering without needing a third-party application.

OPSEC Applications

Realizing that no single product or vendor could address network security completely and do it well, Check Point designed the OPSEC standard to enable security managers to easily extend the functionality of VPN-1/FW-1 with best-of-breed third-party applications designed for specific security requirements. By using a standard set of Application Programming Interfaces (APIs) and open protocols, OPSEC applications are able to easily move data in and out of the VPN-1/FW-1 infrastructure.

An OPSEC session is a dialog between two OPSEC entities using one of the OPSEC APIs, and usually is between VPN-1/FW-1 and a third-party application that performs a specific task on the data received from the firewall. For a list of
available applications, check the OPSEC Alliance Solutions Center at www.opsec.com.

The properties of the OPSEC session are defined in the OPSEC application’s object properties in the Security Policy Editor database. As you can see in Figure 7.1, there are three major types of OPSEC servers using the CVP, UFP, and AMON (Application MONitoring) protocols, as well as six client options using the following APIs:

- Event logging API (ELA)
- Log exporting API (LEA)
- Suspicious activities monitor (SAM)
- Check Point management interface (CPMI)
- Object management interface (OMI)
- UserAuthority API (UAA)

Each one of these protocols is a specific interface used to extend the capabilities of the firewall to another application. This tight integration provides functionality exceeding what would be available with each piece operating individually.

**Figure 7.1 OPSEC Application Properties–General Tab**
Besides the required naming information, the General tab of the OPSEC Application Properties window requires you to specify the host that this server is running on. You must create the host object before creating a new OPSEC application object, as you will not be able to create a new workstation object while application properties window is open. You must then define the application properties, located in the section of that same name. To set the application properties you can select User defined from the Vendor drop-down menu, and then manually select both the server and client entities, or you can select a specific vendor, product, and version here. Vendors and products available from the Vendor menu include the following: Computer Associates’ SafeGate product, Finjan Software’s SurfinGate, as well as a variety of solutions from Trend Micro, F-Secure, Aliroo, and Aladdin Knowledge Systems. Over 70 vendors are predefined and listed in Next Generation Application Interface (NG AI), some with multiple products listed. A complete list of OPSEC certified CVP vendors and products can be found at www.opsec.com/solutions/sec_content_security.html. After selecting a predefined vendor and product from the list, the appropriate Server and Client Entities sections will be filled in automatically.

If you selected User Defined from the Vendor menu, the next step in defining a new OPSEC application object for use in your security policy is to select the Client or Server entry that matches how the application functions. As shown in Figure 7.1 with CVP checked, once you select the appropriate application type, the second tab of the OPSEC Application Properties window, which contains application-specific communication configuration information, will change to match your selection. Your final step on this tab is to configure SIC, or Secure Internal Communication, by clicking the Communication button. Setting up SIC for OPSEC applications is identical to setting up SIC for firewall modules.

The next few pages will discuss each of these communication methods in detail and give you a sense of the flexibility and ease of integration that the OPSEC standard offers.
through a database of known viruses for each HTTP connection would likely slow down the firewall). For example, CVP could be used to move all inbound SMTP e-mail messages to a content-scanning server that will check for malicious Active-X code. Most commonly, CVP is used to virus-scan file data from e-mail messages or files downloaded from the Internet as they pass through the firewall. However, it has also been used to monitor and filter incoming traffic to a SQL database from the Internet by Log-On Software’s SQL-Guard application.

**Defining Objects**

There are three steps involved in creating a new CVP object to use in your Security Policy.

1. Create a standard *workstation* object for the server. The workstation object enables you to assign an Internet Protocol (IP) address and name to the server that hosts the application you will be sending data to.

2. Create a new *OPSEC application* object to define the properties of the service you’re enabling. This can be done by selecting *Servers and OPSEC Applications* from the *Manage* menu, and then clicking *New*, or by right-clicking in the *OPSEC Applications* tab of the *Object Tree* and selecting *New*, and then *OPSEC Application*. When you complete the General tab of the OPSEC Application Properties window, you will be using the workstation object you created for the resources’ host. Figure 7.1 shows the completed General tab.

3. Configure the CVP properties. This is done on the CVP tab that appeared when you checked the CVP option under the *Server Entities*. The CVP tab is used to define how this application communicates with the firewall. As shown in Figure 7.2, CVP applications only require a few options, consisting only of a *Service* drop-down list and an optional directive to use backward compatibility.
The Service selected on the CVP Options tab defines the port on which this application will be listening for connections from the firewall, and is almost always set to FW1_cvp (Transfer Control Protocol port 18181). The Use backwards compatibility mode section replaces the function of the fwopsec.conf file that was used in the version 4.x of FireWall-1. If your OPSEC vendor has supplied instructions relating to that file, then this is the area where you implement them. Generally, applications based on the OPSEC Software Development Kit (SDK) version 4.1 or lower will require that you use backward compatibility. Typically when applications use backward compatibility they also require the legacy fw putkey command to be used on both sides to establish trust instead of SIC.

Creating a CVP Resource

Now that you’ve defined your OPSEC application server, you’ll want to start sending it data from your security policy through a resource definition. There are five resource types that can be used in your security policy to send data to a CVP server:

- **URI** URI resources are mostly used to manipulate HTTP requests.
- **SMTP** SMTP resources enable you to filter and modify e-mail message data as it passes through your firewall.
- **FTP** FTP resources provide the tools needed to control you users’ FTP sessions.
- **TCP** The Transfer Control Protocol resource enables you to work with other TCP services that are not covered by the other resources.
The Common Internet File System resource enables you to granularly filter CIFS file and printer sharing connections.

The previously listed resources are implemented by the VPN-1/FW-1 security servers. Each security server is a specialized module that provides detailed control for specific services. Located just above the Inspection Module in the firewall daemon, the security servers have the ability to monitor and manipulate SMTP, Telnet, FTP, and HTTP traffic, providing highly tunable access control and filtering capabilities.

Since each security server has full application awareness of the protocols it supports, it is capable of making control decisions based on the data and state of the session similar to how proxy firewalls function. In addition to performing specific content filtering, the security servers provide a conduit to send and retrieve data to and from third-party servers, allowing VPN-1/FW-1 to use other security applications in the traffic control process.

When invoked by a resource, the security servers will proxy the affected connections. Aside from the possibility of adding latency to the session (normally only measurable on very busy firewalls or with servers that are improperly equipped to run the OPSEC application) and additional load to the firewall, Network Address Translation (NAT) cannot be used with data allowed (or dropped) using resources. Since the firewall must proxy the connection, all data will appear from the address of the firewall that is closest to the server. This means that any applicable NAT rules will not be used because the firewall itself will function as the server the client is connecting to. Then, once the content is approved, the firewall will create a new connection to the actual server that will service the request. This is probably not a big deal when using hide-mode NAT, but it can be a bit confusing when debugging a problem between networks where NAT is not used. In this case, you would expect the traffic to be coming from the server's IP address, but it would actually be coming from an IP address on the firewall.

To help understand how CVP servers can be used as part of the security policy, let's look at how to integrate virus scanning into the security policy. Later on, we’ll examine in detail how FTP and other resources match data streams that we can send to our CVP server, but for now let's just look at how to set up a simple FTP resource that enables users to retrieve files from the Internet and scans those files for viruses before sending them to the user. There are three steps involved in setting up this simple resource:
1. Create the resource object by selecting **Resources** from the **Manage** menu. Click **New**, then **FTP**. Set up the object name, comment, and color on the resulting FTP Resource Properties window. The other two tabs of this window will allow you to specify the details for the resource’s filter and allow you to send data to the CVP server.

2. On the **Match** tab, set **Method** to **GET**. This instructs the VPN-1/FW-1 FTP security server to only allow users to download files via FTP, since uploading would require the use of the **put** command.

3. Use the **CVP** tab, shown in Figure 7.3, to select the antivirus server object and define how it will function for this resource.

---

**Figure 7.3 FTP Resource Properties—CVP Tab**

Aside from the **Use CVP** checkbox, which enables the **CVP server** drop-down list where you select the server to use, the CVP tab has two other important options that control how the CVP server functions in your resource. The **CVP server is allowed to modify content** checkbox controls whether or not VPN-1/FW-1 will pass on data that has not come back from the CVP server in its original form. This option is particularly useful for virus scanning where an infected file may be sent to the antivirus server and cleaned before being returned. This option would allow the VPN-1/FW-1 security server (which enforces the FTP Resource definition) to accept the cleaned file and send it on
to its destination. If the **CVP server is allowed to modify content** option was not enabled, the antivirus software would only be allowed to report that the file was infected, causing the security server to discard the file completely.

The **Reply Order** options control when and how the CVP server will scan data being passed to the user. The options for controlling how data is scanned are:

- **Return data after content is approved** This option sends the entire file or data stream to the CVP server to be checked after the security server has validated the content. In our example, the GET request would be validated before the file was checked for viruses.

- **Return data before content is approved** Some packets are returned to the security server before the CVP server has approved them. This option is especially useful for resources that may deal with large files. Continuing to send the data stream before it has been approved may help stop problems with FTP or HTTP sessions timing out while the CVP server downloads and then checks the requested file. With this option the CVP server will allow all packets to be sent back to the security server and on to its destination, but the final packet will be held pending approval from the CVP server. This means the file will be incomplete and unusable at the end of the transfer if it is disallowed.

The method you select will depend greatly on what function your CVP server performs on the data, and on how the application is designed. In the antivirus server example, the CVP server controls the reply order. This allows the antivirus software maximum flexibility for scanning files and raw data differently if desired, since the application could decide to assemble a complete binary file before scanning, but scan HTML packets individually. Note that your CVP application must support this option, so check the documentation that came with your application before creating the resource to ensure compatibility.

**Using the Resource in a Rule**

The final step in using a CVP server, after creating the OPSEC application object and using it in a resource definition, is to build it into a rule in your security policy. Creating a security policy rule to use a resource is almost identical to creating a normal rule. The only exception is in the service column where, instead of selecting **Add** after right-clicking, you will select **Add With Resource**. Figure 7.4 shows the Service with Resource window that enables you to configure the resource to be used in the security policy.
The Service with Resource tab allows you to select from the supported services and define which resource to use with that service. In the case of our virus-scanning example, we’ll be using the FTP service with the ftp_get resource. Figure 7.5 shows the completed rule that allows local network traffic to FTP data from the Internet using the resource that limits access to FTP GETs only, and will use the CVP server we defined to scan all files for viruses before passing them to the user. Notice that the Service_Net is negated in the destination. This enables the user to control access to known networks separate from access to the Internet as well as to strictly adhere to the security principle of least access. If the destination field had been set to Any, it would have inadvertently opened FTP access to the network represented by the Service_Net object even though the intention was just to allow FTP GETs from the Internet. You will also notice that the icon used in the Service column indicates that we’re allowing the FTP service with the ftp_get resource.
The important thing to remember when using resources is that data is matched or denied on a per-packet basis. You could, for example, select to scan only files of type "*.exe" downloaded via HTTP, with an accept rule that uses a CVP resource. However, this will only accept the downloaded files, not the pages you must browse to find the file you want. To make this work, you must specify a rule to match all other HTTP traffic, otherwise the HTTP-browsing traffic will fall through to the cleanup rule and be discarded.

**CVP Group**

As with most other objects in the Security Policy, CVP objects can be grouped. When you combine two or more OPSEC applications into a group, additional options for load balancing and chaining become available. Figure 7.6 shows a CVP group configuration tab being used to enable load balancing across two antivirus servers.
Creating a new CVP group can be done easily by right-clicking in the **Servers and OPSEC Applications** tab of the object list. Next, select **New** and **CVP Group**. After defining the group’s name, adding a descriptive comment, and assigning the color you want for this object, you’ll need to select the servers that will be members of this group. Note that groups don’t have to be of identical object types. You can have a group consisting of a UFP server (which we’ll look at next) and a CVP server to enable application chaining.

Once the components of the group have been defined, you’ll have to select the function of this group by making the appropriate selection in the **Work distribution method** section. You have two choices:

- **Load sharing** When selected, the workload is distributed among the servers in the group. There are two distribution methods allowed: round robin or random.

- **Chaining** Chaining allows a data stream to be inspected by several servers that perform different functions. For example, a chaining group consisting of an antivirus scanner and a Web content scanner could be employed to check your incoming e-mail traffic for viruses and appropriate language. If you select chaining, you’ll have an option to abort the chain when any individual server detects a violation, or to allow all the servers to inspect the data before making a control decision.

Once you have the CVP group created, it can be used in the security policy to create a resource rule, just like any other group object would be used to create a standard rule.
URI Filtering Protocol

A Uniform Resource Identifier most commonly defines how to access resources on the Internet. URI Filtering Protocol is used to enable passing data between VPN-1/FW-1 and a third-party server for URI classification.

The most common example of UFP is to pass HTTP Uniform Resource Locators (URLs) to a server running Websense, SurfControl, or a similar product, to check that the requested URL is allowed by your organization’s acceptable Internet usage policy. Since the term URI (described in RFC 1630) and URL (RFC 1738) essentially deal with the same thing (especially when discussing HTTP), it is common to see the terms interchanged. Which term you use (URL or URI) is more a matter of preference than being technically correct, as there seems to even be disagreement between the industry standards organizations as to which is correct in which circumstances.

Load Balancing Chained Servers

CVP chaining enables you to tie servers with different functions together to apply multiple levels of control to a single data stream. For example, you may chain an antivirus and content filtering server together to inspect and clean files downloaded by your users. Load sharing enables you to spread the work to be done across multiple servers for efficiency and redundancy, but what happens if you want to do both?

You cannot apply load balancing to chained servers since load balancing must be done between two or more servers with similar functions, and a chain contains multiple servers all doing different functions. You can, however, chain multiple load balanced servers, enabling you to achieve a similar effect.

Consider that you have two antivirus servers and two content filtering servers that you want to load balance and chain. To do this, you first must create two URI groups that use load sharing, one for the antivirus servers, and one for the content filters. Then all you need to do is create a third URI group that chains the first two groups together. This provides load sharing between similar servers and enables you to chain the servers together.
Defining Objects

Creating a UFP server object is almost identical to creating a CVP object. Both objects require that you define a workstation object with at least a name and IP address for the server and that you use that workstation in the OPSEC application object. Figure 7.7 shows the General tab of the UFP server object, which enables you to define the application you are using. You can choose from the predefined list, which includes vendors such as WebSense, Symantec, SurfControl, Secure Computing, and 8e6_Technologies, or you can use the **User Defined** option to customize your UFP server object. A complete list of UFP applications from OPSEC-certified vendors is available at www.opsec.com/solutions/sec_content_security.html.

**Figure 7.7 UFP Server Object—General Tab**

The difference in setting up a CVP server compared to a UFP server starts when you select **UFP** (as seen in Figure 7.7) in the **Server Entities** section of the OPSEC Application Properties window, which makes the UFP Options tab (Figure 7.8) available.
The *Service* drop-down menu defines which port the UFP service will be listening on; for most UFP applications, this is set to FW1_ufp (TCP port 18182). The backward compatibility options for UFP servers are the same as for the CVP server you looked at earlier, enabling you to configure options that, in previous versions of VPN-1/FireWall-1, were set in the now nonexistent fwopsec.conf file.

The *Dictionary* section of the UFP tab will show the category list from the UFP server. In order for the UFP server to function with VPN-1/FW-1, the servers’ *Dictionary ID* and category list are required. The dictionary is basically a list of categories and the dictionary ID is the version of the list. This is useful if you are using a dictionary that is updated often. Once you’ve set up the server object on the General tab and set the service to match your UFP server, you can click the *Get Dictionary* button to retrieve the category list and ID number from the UFP server. The category list is displayed to help you verify that the connection to the UFP server is established and to show you which categories are available on that server. Note, however, that the categories in this window cannot be manipulated here. To select which categories you would like to filter incoming URLs against, you must create a URI resource that uses UFP.

**Figure 7.8 UFP Server Object—UFP Options Tab**

![UFP Server Object—UFP Options Tab](image-url)
Creating a URI Resource to Use UFP

Unlike a CVP server, which can be used with SMTP, TCP, FTP, and URI, a UFP server can only be used with URI resources. A URI is made up of two basic parts: a scheme or protocol, and a path. The scheme is the first portion of the URI, located to the left of the colon. Common schemes are HTTP, FTP, Trivial File Transfer Protocol (TFTP), Lightweight Data Access Protocol (LDAP), and so on, and can be thought of as a protocol identifier. The remainder of the URI specifies the path to the resource, and often has scheme-dependent syntax. Part of the path may contain a method, such as GET, POST, or PUT, which the UFP server may use to make filtering decisions.

Although the UFP server actually scans the URL and makes a control decision, it's the URI resource that tells VPN-1/FW-1 where and how to send the URI to be scanned. Figure 7.9 shows the URI Resource Properties window that is used to create the resource that will enable you to validate URLs through the UFP server created above.

Figure 7.9 URI Resource Properties–General Tab

Aside from the generic object identifiers, there are some interesting URI resource options to select from. The first is the **Use this resource to** radio button set, which affects how the URI resource functions. If you select the first option, **Optimize URL logging**, all of the remaining options will gray out, and the object will only be used to log HTTP URLs into the VPN-1/FW-1 log. This option will not require the use of a security server to proxy the connection.
In order to use this resource as a conduit to an UFP server, you must select the **Enforce URI capabilities** or **Enhance UFP performance** option. The former utilizes the security server and provides extended options for filtering traffic, while the latter allows the firewall to retrieve the URL deep in the INSPECT engine (without the use of a security server), and to query the UFP server with the URL. Unfortunately, if you select the **Enhance UFP performance** option, UFP caching, CVP, certain HTTP header verifications, and authentication will not be available. For the rest of this section, we will use the **Enforce URI capabilities** option.

The **Connection Methods** section defines which modes VPN-1/FW-1 will use to examine traffic. If **Tunneling** mode is selected, you will not have access to the CVP tab and will not be able to use any URI filtering or UFP servers, since tunneling only allows the security server to inspect the port and IP address information, not the URI that you’re interested in. **Transparent** mode is used when users’ browser configurations do not contain proxy server information. In this configuration, the firewall must be the network gateway that handles Internet traffic. As your users request resources from the Internet, the firewall will send the URIs to the UFP server to be checked as part of the security policy. In **Proxy** mode, the firewall must be specified in each user’s browser as a proxy server. This configuration is very useful if you want to direct Internet service requests (such as FTP and HTTP) to a firewall that is not the default gateway for your network, as the security server will provide proxy services to Internet requests. Using the **Proxy** option also enables you to manually load balance your Internet traffic by directing users’ traffic to different firewalls, or to separate traffic based on type (for example FTP to one firewall, HTTP to another) if required.

The **URI Match Specification Type** section specifies how you want to inspect the URIs matched by this object. We’ll be examining the **File** and **Wildcards** options later in the chapter, but for now we’re only interested in the **UFP** option. Once you select the **UFP** option, then the **Match** tab, as seen in Figure 7.10, will provide you with additional UFP options needed to enable the UFP server.
The Match tab enables you to select which UFP server to use, as well as to set operating parameters to control the interaction between the firewall security server and the filtering application. The UFP caching control field allows you to increase the performance of the URI resource by reducing the number of URLs sent to the UFP server. There are four caching options.

- **No Caching** With caching disabled, the UFP server is used to check each URI. Typically, turning off the cache has a negative impact on performance, as every request must be checked by the UFP Server. However, this option is useful if your UFP server configuration changes frequently and you want to ensure that each request is filtered using the newest options. However, when using the **Enhance UFP performance** option, the overhead of a security server is removed, providing better performance than even a security server, which caches UFP requests.

- **UFP Server** This option allows the UFP server to control the caching. The UFP server may choose to check each URL or it may maintain its own cache to speed up the checks.

- **VPN-1 & FireWall-1 (one request)** The VPN-1/FW-1 security server controls UFP caching. Unique URIs will be sent to the UFP server only once before being added to the cache. This option provides the greatest performance by significantly reducing the number of URIs sent to the UFP server.
VPN-1 & FireWall-1 (two requests)  URIs previously checked by the UFP server will be sent a second time before being added to the cache. Reduced performance is traded for the added security of checking each URL twice.

The Ignore UFP server after connection failure option controls how the security server will react if the UFP server is not available to service requests. Leaving this option unchecked can have a severe impact on performance if your UFP server fails, since the security server will attempt to send each URI to the failed server and will not allow traffic to pass until the server responds with an accept message. If this option is not enabled and your UFP server fails, then you most likely will experience a Denial of Service (DoS) condition, since even acceptable sites cannot be checked. The telltale sign of this condition will be messages in your logs that read, “Unknown error while trying to connect to UFP,” and users calling your help desk complaining of a lack of access. Enabling the Ignore UFP server after connection failure option enables you to specify the Number of failures before ignoring the UFP server option, which controls how many attempts are made before considering a UFP server offline. The Timeout before reconnect to UFP server value instructs VPN-1/FW-1 on how long to wait before considering the connection to the UFP server lost.

**Warning**

The Ignore UFP server after connection failure option is not to be used lightly. By checking this box, if the UFP server fails, all access would still function without the added security the UFP server provides. This could be a circumvention of your overall security policy. Make sure to check what value the company (specifically the Human Resources and Legal departments) places on Web access and the inspection capabilities UFP provides. Because Internet access impacts the ability of users to do work, it must be balanced against any relevant legal ramifications, which means this decision typically needs to be made at an executive level by someone with the authority to decide if lost productivity takes a higher priority than content security.

Finally, the CVP tab enables you to hand data off to a third-party server for validation. In addition to the antivirus example we looked at earlier, CVP servers like Symantec’s Igear Web content scanner can provide you with fine-tuned con-
tent control for Web applications. Note that the CVP tab is not available if the Tunneling or Enhance UFP performance options are selected. The Action tab in the URI Resource Properties window is discussed later in this chapter.

Using the Resource in a Rule

Using a UFP server to validate URIs as part of your security policy is similar to using a CVP server in a resource rule. To follow the example used earlier, the UFP server can be used to scan URL requests to Internet sites. In doing so, the final step is to add the URI resource, which uses the UFP server object, as the resource in a new (or existing) rule. As with the CVP rule we created earlier, the only difference between a rule that uses a resource and a normal security policy rule is what is defined in the service column. Instead of selecting the Add option for the service, use the Add with Resource option to select the URI resource that contains the UFP server configuration you need. Figure 7.11 shows the final rule in the security policy being used to reject unacceptable data requests. Notice that the Service column shows both the scheme being used (HTTP) and the name of the URI resource (URL_Filtering).

Figure 7.11 Security Policy Rule Using UFP Server in URI Resource

As with CVP resources, it is necessary to remember that a match is made on the packet, not the session. For example, with UFP, you will typically create a drop or reject rule to match on the categories you want to disallow. As you can see in Figure 7.11, you must have another rule that will accept the traffic that you want to allow, or else it will be dropped on the cleanup or Drop All rule. This second rule is necessary because the resource rule only deals with dropping traffic, not with allowing it. You could, of course, use a UFP resource in the rule.
base to allow traffic based on category rather than drop it to get around this second rule requirement. The only problem with this approach is that the allowed list is often longer than the drop list, and is therefore harder to maintain. The difference between drop and reject in these two cases is that drop will silently drop the packets, whereas reject will quickly tell the user that his connection is not allowed by returning an error or redirecting the user to another Website if defined in the Action tab. Reject is typically a more useful configuration because it will allow you (and your helpdesk) to distinguish between network connectivity problems and disallowed Websites.

UFP Group

A UFP group is similar to a CVP group except that it does not support chaining. The configuration of a UFP group is similar to the other generic group configuration screens, in that you enter a name, comment, and select the appropriate color and then simply move UFP servers from the Not in group section to the In group section.

Your choices for load balancing between servers in a UFP group are either Random or Round Robin. Using Up and Down buttons will enable you to change the order in which servers are used in the round robin configuration, but since the server being used will change with each incoming session, changing the order will only slightly affect how the object performs. The final option, Load sharing suspend timeout, enables you to configure the time to ignore a failed server before attempting to reestablish communication with it. You can set this time to anywhere from 0 (ignore the failure, attempt to use server normally) to 10,000 minutes.

Application Monitoring

Using OPSEC applications as CVP and UFP resources in your security policy makes those servers an integrated part of your security environment. To allow for easy monitoring of OPSEC products that function alongside VPN-1/FW-1, Check Point developed the AMON API.

AMON is the third tab in the OPSEC Application Properties window (as shown in Figure 7.12). It allows supported applications to report status information to VPN-1/FW-1. This status information is then available in the Check Point System Status Viewer alongside the real-time status of your Check Point applications. This is very useful for monitoring all devices interoperating within
the security infrastructure, but another solution would probably be more useful for monitoring your entire network.

**Figure 7.12 AMON Application Properties—General Tab**

Enabling AMON is as simple as selecting the **AMON** option under **Server Entities**, and then setting the **Service** and **AMON Identifier** information on the AMON tab. As seen in Figure 7.13, the **Service** option is usually set to FW1_amon (TCP port 18193), but you should check the documentation that came with your application to ensure that this is the port the application is listening on. The **AMON identifier** field contains the Management Information Base (MIB) identifier, which also must be provided by your application’s vendor.

**Figure 7.13 OPSEC Application Properties—AMON Options Tab**
Client Side OPSEC Applications

In addition to the UFP and CVP application servers and the AMON monitoring service, there are six client application APIs that extend the functionality and management of VPN-1/FW-1 to third-party applications. Although complete configuration and implementation details for each of the six APIs will be dependent on which third-party application you’re using, this section will give a quick look at each to discuss the capabilities of the API and to show the integration options possible for OPSEC-certified products.

Event Logging API

The Event Logging API allows third-party applications to send log data to the VPN-1/FW-1 log database. Sending log data to the central log has two main advantages: log consolidation and alert triggering.

In many networks, the firewall gateways are the security focal point, making the VPN-1/FW-1 logs the primary data source for security auditing. By extending the log to third-party products with the ELA, Check Point has enabled you to collect your security logs into a single location, making it easier to analyze and trend your security infrastructure’s performance. An added benefit of consolidating logs from other products into the central log is that products using ELA will be able to trigger the VPN-1/FW-1 alert mechanism. This allows products like Stonesofts’ StoneBeat high-availability solution to send logs and alerts to the Check Point Management Console when a FireWall-1 product has failed over to a standby machine.

Log Export API

To securely and efficiently access the Check Point log database, third-party products can use the Log Export API. The LEA allows access to the log in both real-time and historical access modes. In order to use LEA, the product vendor must write an LEA client that will access data from the Management Console that is running the LEA server. Using the LEA client/server model, OPSEC applications reduce the need to try to access the locked, proprietary formatted logs directly or having to export the Check Point logs out to plain text before being able to work with the log data.

For example, products like the WebTrends Firewall Suite can set up a secure connection to the VPN-1/FW-1 log database to pull in historical information for report generation. Since LEA supports encryption, you can be assured that
the information used to generate the reports was not copied or corrupted during the transfer from one application to another.

Real-time data retrieval using LEA is most useful for generating alerts, based on firewall events, with a non-Check Point application. For example, LEA could be used to funnel firewall events into an Enterprise security manager (ESM) product that could correlate data with other security products, to generate trends and alerts based on a bigger view of the security infrastructure.

Suspicious Activities Monitoring

The Suspicious Activities Monitor was designed to provide a method for intrusion detection system (IDS) software to communicate with VPN-1/FW-1. This provides a method for an IDS application to create dynamic firewall rules to block traffic that the application believes is malicious.

Using a SAM-enabled application allows you to add some level of reflexive access to block previously allowed traffic. The key is in remembering that the access can only be granted with the static security policy rules, not the SAM application’s dynamic rules. For example, if an IDS system detected something suspicious like a connection attempt to a closed port, it would be able to close all access to all resources from the IP address in question for a configurable period of time. This would block traffic, such as browsing your Internet Website, which may be explicitly allowed in your security policy. The action taken by the firewall is configurable and can include anything from making an entry in the logs, disconnecting a session in progress, or blocking all further access from the offending host. You need to be especially careful when allowing SAM applications to create firewall rules. If not configured properly, you can inadvertently create a denial of service situation on your own servers. For example, if you block all data from any host that has tried to connect to a closed port for one hour, an attacker may send connection requests to your servers with spoofed IP addresses in order to cause your own firewall to block traffic from your customers.

SmartDefense can be used to block attacks it recognizes them (as discussed in Chapter 13), but other solutions may notice traffic that is also unauthorized. The SAM API allows other devices to tell the firewall to block connections as appropriate. The SAM protocol is discussed in more detail in Chapter 9.

Object Management Interface

The Object Management Interface allows OPSEC applications to interact with the management server. The OMI has been replaced by the Check Point
Management Interface, and has only been kept in NG for backward compatibility. New applications being developed with the NG OPSEC Software Development Kit (SDK) will use CPMI.

Check Point Management Interface
Replacing OMI in the NG OPSEC SDK, the Check Point Management Interface allows OPSEC applications access to the management server’s security policy and objects database. This can enable you to use objects already defined with the Policy Editor in other applications. Additionally, this secure interface can provide other applications access to create objects in the VPN-1/FW-1 database. The CPMI has three main benefits that OPSEC applications can take advantage of:

- CPMI can allow access to authentication information, enabling vendors to design single sign-on security solutions that take advantage of the authentication information already known to the firewall.
- Access to the Check Point object database can allow for report generation and alerting based on changes to monitored objects.
- Some management tasks can be automated, allowing software products to modify VPN-1/FW-1 in response to a security event.

UserAuthority API
The UserAuthority API is designed to extend the firewall’s knowledge of users’ VPN and local area network (LAN) authentication to other applications. In addition to providing the information that applications need in order to enable a single sign-on model, the UAA can also be used to provide information needed to develop billing and auditing applications that track individual users instead of just sessions.

The UAA also allows third-party applications to take advantage of the secure virtual network’s (SVN) openPKI infrastructure for authentication. This reduces the vendor’s need to develop their own authentication methods, which not only speeds development time for new applications, but also ensures compatibility with and leverages the investment in your existing infrastructure.
Other Resource Options

When we examined CVP and UFP resources, we touched on the basics of URI and FTP resources to show how to use the third-party servers in the security policy. URI resources can be used to filter based on wildcard matches and can be configured using specially formatted files, which you could create or purchase. After covering the remaining URI filtering methods and functions, we’ll have a closer look at the FTP resource that we used in the virus-scanning example earlier, and we will examine SMTP and TCP resources.

The URI, SMTP, FTP, TCP and CIFS resources can be used in the rulebase in the same fashion as a normal service (such as HTTPS). The difference is in how the firewall handles the resource. When a packet matches a rule that uses a resource, the connection is handed off to the appropriate security server (if necessary) to make a control decision after inspecting the connection’s content. This means that the packet must be approved by the resource before the rule’s action will take effect. This is important to keep in mind when creating your rules, as you don’t want to waste time virus-scanning files with a resource that will be dropped by the rule that caused the scan to be performed.

URI Resources

In addition to the resource we examined earlier (Figure 7.9) to use a UFP server in the security policy, there are two other types of URI resources. URI file resources allow you to use a specially formatted file to load complete URL strings, while wildcard resources allow you to create completely custom-match strings that may be as simple as looking for all executable files.

When you select a type of URI resource on the General tab, the Match tab will change to offer specific options for that type of object (Wildcard, File, or UFP). We’ve already looked at the UFP Match tab (Figure 7.10), and will examine the File and Wildcard tabs next, but it’s worth noting that regardless of which **URI Match Specification Type** you choose, the Action and CVP tabs remain unchanged.

As we saw when we looked at CVP servers, the CVP tab (Figure 7.3) enables you to configure the resources’ interaction with the CVP server. The Action tab, shown in Figure 7.14, enables you to specify some interesting things to further control and filter URI requests. Here you can enter a **Replacement URI**, which redirects the user’s session to a site of your choice if the rule that matches this object sets the action to reject. Many companies use this option to redirect
users to the corporate acceptable Internet-use policy when certain blocked URLs are requested.

**Figure 7.14 URI Resource Properties—Action Tab**

![URI Resource Properties—Action Tab](image)

Limited content filtering is available through the use of **HTML Weeding** on the **Action** tab. You have five options for removing Active X, JAVA, and JAVA Script code from the HTML data.

- **Strip Script Tags**  Remove JavaScript information from the selected Web page.
- **Strip Applet Tags**  Remove Java Applet information from the selected Web page.
- **Strip ActiveX Tags** Remove ActiveX information from the selected Web page.
- **Strip FTP Links**   Remove links destined for an FTP site from the selected Web page.
- **Strip Port Strings** Remove port strings from the selected page.

Although removing this data from the HTML code before the user sees it does reduce the risk of malicious code being sent to your users, the data stripping is non-selective, so all tags are removed. In addition, you have the option, under **Response Scanning**, to block all Java execution. You need to consider how these settings may reduce the functionality of some pages and have a negative impact on your users before enabling this type of filtering. To achieve more
granular control over these data types, you need to look into the services provided by a good CVP or UFP application.

**URI File**

After selecting **File** on the URI Resource Properties General tab (Figure 7.15), the Match tab will display the import and export options, as seen in Figure 7.16. These options enable you to load the match string definitions from disk rather than having to create complicated match strings manually.

**Figure 7.15 URI Resource Properties—General Tab**

![URI Resource Properties—General Tab](image)

Clicking **Import** will enable you to specify the directory and filename of the file that contains the URIs you want to apply the filter to. The **Export** option will create a file containing the currently filtered URIs.

**Figure 7.16 URI File Configuration**

![URI File Configuration](image)
A URI specification file can be bought from companies that specialize in URL classification, or you can create your own. When creating a URI specification file, be sure to use an ASCII editor that uses a newline as the new line character, as this is the character the security server expects at the end of each line. There are three parts to each line in the URI specification:

- The IP address of the blocked server.
- An optional path to filter.
- A category number. Typically, each line is set to 0 (zero), but you can pick any number you like. Be careful when applying service or feature packs to your firewall, as it is possible that Check Point may start using this field in the future, so you may need to adjust it to an acceptable value.

The completed line will look similar to this: `192.168.0.1/home 0`, which will deny any data request for information under the /home directory on the 192.168.0.1 server. Your firewall will require access to a domain name service (DNS) server if you use the name of the blocked resource rather than the IP address. Also, note that you could be generating a considerable amount of DNS traffic if you have a busy firewall and are using names rather than IP addresses, since each URI must be resolved before being checked.

### URI Wildcards

When you select the **Wildcards** option from the General tab on the URI Resource Properties window (Figure 7.17), you are offered several options on the Match tab that will help you build a customized string to search for. You’ll also notice that a new tab, SOAP, is created.

**Figure 7.17 URI Wildcard Resource General Tab**
Figure 7.18 shows the predefined checkbox options available on the Match tab. As well as the commonly used schemes and methods provided, the Other option can be used to provide even greater flexibility.

**Figure 7.18 URI Wildcards Match Specification**

Under the Schemes section, you can select from the predefined common schemes of HTTP, FTP, Gopher, mailto, NEWS, and WAIS. If what you’re looking for isn’t among the six schemes provided, you can specify exactly what you need in the Other field. Most commonly, you’ll be entering complete schemes to catch such as HTTP’s, but this field also supports wildcards, so you can, if needed, specify something similar to *tp in this field. This would enable you to catch any scheme that ended in the string ‘tp’ such as FTP, NNTP, SMTP, and HTTP, among others. You need to choose your wildcards carefully to ensure that you’re not blocking or allowing something that you hadn’t intended with a poorly written search string.

The Methods section provides the most common HTTP methods in a predefined set of options:

- **Get** The GET method is used to retrieve all the information specified by a URI. It is commonly used to download a complete HTML file as part of a Web browser session.

- **POST** Used to ask the server to accept a block of data, and is usually found in forms to send input from the user back to the server for processing.
- **HEAD** This method functions almost exactly like GET, except that the entire requested resource is not returned. HEAD is commonly used to validate URL links and to check time and date stamps for modification (normally to see if a cached copy is still current).

- **PUT** This method is used to place data (normally files) into the location specified by the URI, and is unlike the POST method, which sends data to an application as input.

The **Other** field in the **Methods** section supports the following less-common methods as well as wildcards that can be used to specify a custom pattern to match:

- **OPTIONS** This method can be used to determine the parameters available and supported at a specified URL. The OPTIONS method is commonly used to retrieve information about the server or specific resources without using a method like GET or HEAD, which would attempt to retrieve the actual object.

- **PATCH** Functions like PUT except that only a list of changes or differences between the file specified in the URL and the client’s copy is sent. This method is most likely to be used when dealing with large files that only receive small updates, so sending only the changes is more efficient than sending the entire file again.

- **COPY** The COPY method specifies a second URL in the request headers and instructs the server to place a copy of the specified resource at the location defined in the headers. This would enable the user to copy data from one server to another without having to download a copy of the data first, and is commonly used if the network between the servers is faster than between the client and the servers.

- **DELETE** Instructs the server to delete the resource (normally a file) specified in the URL.

- **MOVE** The MOVE method will first copy the data to another specified URL then delete the original.

- **LINK** Allows you to create relationships between resources and is similar to the `ln` command on UNIX systems.

- **UNLINK** Deletes the relationships created by LINK.

- **TRACE** The TRACE method is normally used for testing and will cause the server to echo back the information it receives from the
client. This allows the client to analyze the information that was received by the server and compare it to what was sent.

The final section of the Match tab allows you to specify the host, path, and query options to match. The **Host** option can be specified by name (such as www.syngress.com) or by IP address. If you specify the host by name, you will need to ensure that the firewall has access to a DNS server to resolve the name to an IP address. You can use wildcards to help build the pattern to match if needed.

The **Path** option must include the directory separation character (normally /) in order for a match to be made. When you define the path to match, you must specify the complete path, down to the individual file, or use wildcards to match all files or directories. Table 7.1 shows common strings used in the path field and how they will match to incoming data.

### Table 7.1 Path Field Search Examples

<table>
<thead>
<tr>
<th>String</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>/home</td>
<td>Will match a file called home in any directory. For example: /home and /mysite/mydir/home would both be matched. In either case, if home was a directory, no match would be found.</td>
</tr>
<tr>
<td>/home/*</td>
<td>This pattern will match all files and directories under the home directory. For example, /home/index.htm and /home/files/index.htm would be matched.</td>
</tr>
<tr>
<td><em>/home/</em></td>
<td>This will match any URI that contains the directory home, so files in /home would be matched as well as files in /mydir/home/mysite.</td>
</tr>
<tr>
<td>*/index.htm</td>
<td>This will match the file index.htm in any directory.</td>
</tr>
<tr>
<td><em>/</em>.mp+</td>
<td>This pattern will match three character file extensions that start with “mp,” such as mp3 and mpg.</td>
</tr>
<tr>
<td><em>/</em>.{exe,zip,gz}</td>
<td>Will match all files that end in .exe, .zip, and .gz in any directory.</td>
</tr>
</tbody>
</table>

The **Query** field can be used to match on any string of characters found after a question mark (?) in a URL. Since wildcards are supported here as well, it is not necessary to know the exact placement of the key words you are looking for in the query. For example, this will allow you to block or redirect searches for keywords that are in violation of your Internet acceptable-use policy.
When working with URI resources, it is common to use a single asterisk in the three match fields so that all possible requests can be matched. However, when using CVP servers, it is often useful to do specific file matching with wildcards in the patch field to ensure that only supported data types are sent to the server to be scanned.

The final tab is the SOAP tab. SOAP stands for Simple Objects Access Protocol. It is a lightweight protocol used in the exchange of information in a decentralized, distributed environment. SOAP messages are encoded in XML (extensible markup language). A full discussion of SOAP and XML is well outside the scope of this book. More information can be found in other books or at http://www.w3.org/TR/SOAP/.

The SOAP option can only be used with HTTP connections that are accepted. It is not usable if the action is drop or reject. The additional checking that VPN-1/FW-1 does when **Allow all SOAP requests** is selected is to confirm that the SOAP requests conform to RFC standards (see Figure 7.19). When selecting **Allow SOAP requests as specified in the following file**, a file named scheme1 through scheme10 in the management station’s $FWDIR/conf/XML directory will specify the namespaces and methods used for the exchange. The namespace and XML methods being passed can be viewed in SmartView Tracker by setting the Track SOAP connections option. An example can be seen in $FWDIR/conf/XML/SchemeSample.dat. The syntax for the file is as follows:

```
namespace method
```

Example:

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

www.syngress.com
SMTP Resources

The SMTP resource defines the methods used by VPN-1/FW-1 to control and manipulate incoming and outgoing e-mail. There are many options, including the ability to remove active scripting components, rewriting fields in the envelope (such as to: or from:), or filtering based on content. The configuration of an SMTP resource is similar to that of URI resources, including the ability to use a CVP server to provide third-party content filtering. Figure 7.20 shows the General tab of the SMTP Resource Properties window that is used to set basic operational parameters for the resource.

Figure 7.20 SMTP Resource Properties—General Tab
This tab includes the standard initial object setup of name, comment, and color. If you want to forward all messages to another server, specify its name or IP address in the **Server** text field. Enable the **Deliver messages using DNS/MX records** option to have these messages delivered directly to the specified server rather than to a group of servers used for redundancy purposes. The **Check Rule Base with new destination** option can be used to instruct the security server to recheck the SMTP message’s destination server against the security policy after being modified by the SMTP resource. Identical settings are available for the handling of error mail messages if the **Notify sender on error** option is selected.

The Match tab, shown in Figure 7.21, has only two option fields that control how to match messages being examined by the security server. The **Sender** and **Recipient** fields are used to define the addresses you want to work with. Wildcards are supported in these fields to provide the ability to specify all addresses (using *) or all users in a specific domain (with *@domain.com) if needed. The example shown in Figure 7.21 shows how an administrator would allow incoming mail to mycompany.com, but not allow relays or outgoing mail. In most cases an administrator would configure two resources, one for inbound mail and another for outbound mails.

**Figure 7.21 SMTP Resource Properties—Match Tab**

When you create a new SMTP resource, the **Sender** and **Recipient** fields are blank and must be filled in before the resource will function. You need to be careful with these options, though: it’s common to just set the **Recipient** field to an asterisk to save time. You need to keep in mind that the resource defines how the security server will function, and by placing an asterisk in both of the available fields, you could be allowing external hosts to bounce mail off your firewall. This makes your firewall an open relay for SMTP traffic, and aside from the possibility of your server being used to send unsolicited bulk e-mail (spam), many domains and even some ISPs may refuse to accept SMTP traffic from your server. 
domain if it’s found that you have an open relay. For information on blocking open relays from your domain, or checking to see if you’ve become blacklisted, check an open relay database site such as www.ordb.org and check your Postmaster@yourdomain.com mailbox.

The Action1 tab has a few simple options that allow you to re-address messages and change limited content. The **Sender** and **Recipient** fields allow you to re-address messages on a single-user basis, or by using wildcards, to translate addresses for an entire domain. The **Field** option allows you to modify data in any of the other standard SMTP fields such as the carbon copy (cc), blind carbon copy (bcc), or subject. Once you’ve specified the field to change, you need only specify the string to look for, and what to replace it with. Shown in Figure 7.22, this tab is very useful if you have recently changed your SMTP domain name but still have a few messages coming to the old domain. Using the simple rewrite options shown, you could easily translate an address joe@olddomain to joe@newdomain.com. The **Help** button for this section has some useful information in the section entitled *Using wildcards and Regular Expressions in Resources*. It also defines how you can specify multiple rewriting rules even though you see only one text box.

**Figure 7.22** SMTP Resource Action Tab Showing Address Rewrite

![SMTP Resource Action Tab Showing Address Rewrite](image)

The Action2 tab allows the removal of information found within the body of the message. The **Attachment handling** section provides two simple methods
of discarding attachments from messages. In Figure 7.23, the resource is configured to strip attachments of the message/partial type. There are seven supported options, as defined in RFC 2046, for removing specific file.

- Text
- Multipart
- Image
- Message
- Audio
- Video
- Application

You can use the **Strip file by name** field to remove files based on a pattern, using wildcards if needed, rather than by Multipurpose Internet Mail Extension (MIME) type. This field is often used to stop "zero day" or new viruses and worms that spread via e-mail. It's often faster to start filtering out viruses by their specific attachment names (once known), than it is to update the virus signatures throughout your entire enterprise. In Figure 7.23, files ending with the extension .exe, .vbs, or .scr will be stripped. If nothing else, this function will buy you enough time to update your signatures properly while you block new infections from entering (or leaving) your network.

Use the **Do not send mail larger than** field to specify the maximum allowable message size. Use the **Allowed Characters** options to specify whether the security server will accept messages in either 7- or 8-bit ASCII. The **Weeding** section allows you to remove JAVA, JAVA Script, Active X, FTP URI links, and Port strings from the message’s headers and body.
One common mistake made when creating SMTP resources is not checking the **Do not send mail larger than** field. By default, the messages larger than 10,000 KB will be dropped. Note that in NG AI the default maximum message size has been raised to 10,000 KB, compared with 1,000 KB in its predecessor. This is because many attachments are larger than the previous limit of just under one megabyte. Aside from irritating users, failing to check this option often resulted in e-mail administrators spending hours troubleshooting lost SMTP messages, since the security server would discard the entire message.

The CVP tab of the SMTP Resource Properties window provides the standard options we discussed when examining CVP servers. The only exception, as shown in Figure 7.24, is the addition of a single SMTP-only option to **Send SMTP headers to CVP server**. This option instructs the CVP server to scan messages’ full headers in addition to the message body.
FTP Resources

We looked at FTP resources briefly when we first examined CVP servers. In addition to enabling you to send FTP data streams to another server for content filtering, FTP resources can be used without a CVP server to just control FTP sessions.

The General tab in the FTP Resource Properties window (Figure 7.25) allows you to specify the normal VPN-1/FW-1 object information, but the interesting options (aside from the CVP tab) are on the Match tab.

Figure 7.24 SMTP Resource Properties—CVP Tab

Figure 7.25 FTP Resource Properties—General Tab
The Match tab, shown in Figure 7.26, contains three options that allow you to control the actual FTP session. The Path field allows you to specify specific file paths, using wildcards if desired, to perform actions on. The most interesting and useful part of the FTP resource is the use of GET and PUT, since they enable you to control FTP functions. Using these options will allow you to control the commands that your users can issue to remote servers. Allowing your users to GET but not PUT will prohibit them from pushing data out of your network, while still allowing them to download files as needed. Allowing PUT but not GET would be a good solution for a publicly accessible FTP server used to receive files from your business partners, since they could upload files to you, but could not download anything.

**Figure 7.26 FTP Resource Properties—Match Tab**

![FTP Resource Properties](image)

The FTP Resource CVP tab enables you to specify a CVP server to send matched data to, and defines the interaction between the FTP security server and the CVP server. Similar to the example you looked at when examining CVP server objects, Figure 7.27 shows how to scan incoming files for viruses. By enabling the CVP Server is allowed to modify content option, you can specify that infected files are to be cleaned. If this option was unchecked, all infected files would be discarded.
TCP

The TCP resource allows you to work with services not handled by built-in security servers, and has only two methods of operation. You can use the TCP resource as a generic daemon, providing an alternative to the HTTP security server, for interaction with a CVP server.

Additionally, you can use the TCP resource to screen URLs via a UFP server without the intervention of the security server. Note that the UFP server must support this sort of interaction, as the format of its incoming data stream will not be in full URI format, since only the IP-based URL is available without the security server. The TCP resource has three possible tabs, only two of which are displayed at any time. The **Type** option on the General tab (Figure 7.28) enables you to select either **UFP** or **CVP**, and this dictates which other tab (UFP or CVP) is offered for configuration.

![Figure 7.27 FTP Resource Properties—CVP Tab](image-url)
After checking UFP on the General tab, you can then access the UFP tab (shown in Figure 7.29) and configure the associated tab. The UFP configuration on this tab is similar to other resources that use UFP servers. You need only to select the UFP server that this resource will be using, configure the caching method, and select the categories against which this data stream will be checked from the supplied list.

If you select CVP on the General tab, you will be presented with the CVP tab (Figure 7.30), which will allow you to configure the resource’s interaction with the CVP server. You will need to specify which CVP server to use from the drop-down list on the CVP tab. The other options here are identical to the CVP objects you’ve looked at before, and will enable you to configure options such as whether the CVP server is allowed to modify the content passed to it, and to specify the method in which data is returned to the security server.
CIFS

With a CIFS resource, an administrator can grant granular access to shares on a server to different user groups or to everyone. CIFS resources are most common when controlling access to internal servers from the LAN or controlling access to a file server across a site-to-site VPN.

CIFS is the protocol used for file and print services between clients and servers on the network. Legacy CIFS connections (implemented over NetBIOS) run over port 139. In Windows 2000 and later, the Microsoft-DS protocol (running over port 445) is used. A single CIFS resource can be used with both ports to ensure consistent enforcement across both file-sharing protocols. In Figure 7.31, the resource could be used in a rule to grant access to the shared for only certain source address, to certain users, or to deny access to the shares to the entire LAN. It all depends on how the resource rule is created.

Figure 7.31 CIFS Resource Properties—General Tab
Summary

Check Point’s OPSEC standards program certifies that third-party applications meet minimum integration and compatibility requirements with the VPN-1/FW-1 products. This, in essence, extends the reach of your VPN-1/FW-1 security infrastructure to encompass areas where highly specialized or customized solutions are required to meet the needs of your network.

Through the use of CVP and UFP application servers, you are able to extend the information used by VPN-1/FW-1 to make data control decisions to include input from third-party solutions. In addition to providing you with greater flexibility, this enables you to build best-of-breed solutions into your firewall from vendors that specialize in the task you need to perform.

CVP is used to send an entire data stream, such as a downloaded file, to another server to be validated either as a whole or in parts. This validation can be as simple as checking the file for viruses or using image recognition software to discard images that may not be acceptable in your environment. In many cases, such as when using a virus scanner, the CVP server may modify the data before returning it to the security server to be passed along to its final destination. CVP objects can be grouped together to share load among servers performing a similar function, or servers can be chained together to perform multiple actions and validation checks on the data before returning it to the firewall.

UFP is used to check the scheme and path of data resource requests. UFP is most commonly used for HTTP traffic to control access to sites that may not be appropriate in a corporate setting, but can also be used with other protocols. UFP servers enable you to choose from predefined categories to specify which sites are to be filtered or denied from the data requests passing through the firewall. UFP applications often come with a subscription service that will provide updates to the database of sites and categories known to the product, as well as enabling you to specify your own so that your protection is kept up to date. As with CVP resources, you can group UFP servers together to provide high availability and load sharing among servers providing the same service. You cannot, however, chain UFP servers together.

AMON is new to the NG version of VPN-1/FW-1 and provides a method for third-party servers to report status information to the firewall products. This allows you to monitor the status of other security devices using the tools from Check Point, or other vendor tools that you’re already using to keep an eye on your firewalls.
OPSEC applications can also access VPN-1/FW-1 information and resources by using LEA, ELA, SAM, OMI, CPMI or UAA. These client applications are not normally used in the data control process as OPSEC servers are, but often make use of the status, log, and object databases to report on and manipulate VPN-1/FW-1 devices and applications.

There are five major types of resources in VPN-1/FW-1: URI, SMTP, FTP, CIFS, and TCP. URI is the most common and offers the greatest flexibility, since URI resources can be created using wildcards or from specially formatted files that define the pattern to match on. Most commonly, URI resources are used with CVP or UFP servers as a method to move data between the security policy and third-party servers.

SMTP resources allow you to manipulate e-mail messages and provide a method to replace or substitute information in certain fields as messages pass through the firewall. FTP resources allow you to control FTP sessions down to the level of being able to specify whether users can issue GET or PUT commands, as well as the ability to stop users from accessing specific paths on the server. Both SMTP and FTP resources support using CVP servers to validate data coming into or leaving your protected networks. The TCP resource enables you to use either a UFP or a CVP server with TCP data that is not handled by one of the built-in security servers. A CIFS resource is used to granularly control access to file and print servers based on user, server, or share name.

Solutions Fast Track

OPSEC Applications

- Using third-party OPSEC-certified applications enables you to build onto your existing Check Point security infrastructure to address specific security needs, while ensuring compatibility and interoperability.

- There are three types of OPSEC server applications: CVP, UFP, and AMON. UFP and CVP servers interoperate with VPN-1/FW-1 by passing data back and forth and participating in the control process, whereas AMON is used by other applications to report status information back to the firewall management server.

- OPSEC client applications, as a general rule, either send data to or pull data from VPN-1/FW-1, and generally do not affect the control process.
directly as servers do. There are six methods for OPSEC clients to send or receive data from VPN-1/FW-1: LEA, ELA, SAM, OMI, CPMI, and UAA.

☑ ELA allows third-party applications to send log data to the VPN-1/FW-1 log database for consolidation and alerting functions.

☑ LEA provides a method for applications to extract log data from the central log database, either historically or in real time.

☑ SAM provides a conduit for IDS devices to signal and make changes to the current security policy, such as blocking traffic from a specific host.

☑ The OMI provides support for legacy applications that need to access the VPN-1/FW-1 object database.

☑ CPMI replaces OMI in the NG version of VPN-1/FW-1. CPMI allows applications to access the object database as well as authentication information known to the firewall. CPMI also provides the needed APIs to allow third-party applications to make limited changes to the security policy.

☑ The UAA can be used to access VPN and LAN authentication information from VPN-1/FW-1. This allows applications to be designed to use existing logon information to provide single sign-on capabilities.

**Content Vectoring Protocol**

☑ CVP is normally used for sending data, such as binary files or e-mail messages from VPN-1/FW-1, to a third-party server to be scanned. The results of the scan have a direct impact on the control decision for that data, which can include blocking the data entirely or just modifying it to an acceptable format (in the case of removing a virus).

☑ CVP resources are created using an OPSEC Application object as the server to send data to, and contain configuration settings for what actions the CVP server is to perform on the data.

☑ CVP groups allow you to load share between servers or chain multiple CVP servers together to perform different tasks one after another.

☑ Load sharing splits the incoming work to be done evenly among the defined servers, using the method that you specify.
URI Filtering Protocol

- A URI describes how to access a resource and is made up of two parts. The scheme defines which protocol (such as HTTP) to use and is separated by a colon from the path to the desired resource.

- UFP can be implemented through the use of URI resources in the security policy, and allows you to examine and filter URIs passed from the VPN-1/FW-1 security servers as part of the control decision.

- UFP is commonly used to verify that requested or returned URLs conform to an acceptable standard, by classifying URLs into categories and enabling you to choose which categories are permissible in your environment.

- UFP groups enable you to share load between multiple UFP servers to increase efficiency and provide availability, if a UFP server should fail.

Other Resource Options

- URI file resources allow you to use a specially formatted file to define the URIs that you want to filter on. This option is commonly used when you have many URIs to filter but do not want to use a UFP server.

- URI wildcards allow you to build a completely customized URI string to match to incoming data. The flexibility of wild cards enables you filter on a specific file extension or even specify entire IP address blocks.

- SMTP resources enable you to inspect and modify e-mail traffic passing through your firewall. You can, for example, modify sender or recipient information in addition to the data within the body of the message. It is also possible to perform limited screening for potentially malicious content by removing Active X and/or JAVA code from the messages. For more granular screening capabilities, the SMTP Resource enables you to send e-mail messages, with complete headers, to a CVP server to be analyzed.

- FTP resources allow you to control FTP data streams. In addition to looking for certain paths or file names being requested, you can control when and where your users can use the FTP GET and PUT commands to control data moving into or out of your network.
The TCP resource allows you to send data from TCP protocols not covered by the normal security servers to a CVP or UFP server for inspection.

The CIFS resource enables an administrator to very granularly define access to file and print sharing servers over NetBIOS and Microsoft-DS protocols.

**Frequently Asked Questions**

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to [www.syngress.com/solutions](http://www.syngress.com/solutions) and click on the “Ask the Author” form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

**Q:** My URI specification file looks okay, but it doesn’t work properly. What should I look for?

**A:** There are three major parts to each line in the URI specification file. After you’ve entered the IP address, path, and category, you must end each line with a new line character (`\n`). If you use a Windows-based computer to build your file, ensure that you use an editor that uses only `\n` when you end a line. The WordPad application or Edit (run from a cmd.exe window) will create the file properly, whereas the Notepad application may not. When in doubt, add an extra new line character at the end of the file.

**Q:** What are the valid wildcard characters?

**A:** There are only four characters that can be used as wildcards in resource definitions, such as a URI wildcard object:

- The asterisk (*) can be used to match any number of characters.
- The plus sign (+) can be used to match a single character only. For example, ‘+tp’ will match ‘ftp’ but not ‘http.’
- The ampersand (&) can only be used with SMTP addresses and allows you to manipulate information on either side of the @ symbol for address replacement objects. For example, changing from
“jim@yoursite.com” in an object to “&@yournewsite.com” results in “jim@yournewsite.com.”

- A list of strings may be separated with commas (,) to match any one of the specified strings. The case of “hr,sales,” “@yoursite.com” will match “hr@yoursite.com” and “sales@yoursite.com.”

Q: What OPSEC applications are available?
A: The list of OPSEC-certified applications grows everyday. At the time of this writing, there are over 300 certified OPSEC vendors, each with one or more certified applications. This means that when you’re looking for a third-party product to fill a specific security need in your organization, odds are that there is an OPSEC-certified product available. The current list of OPSEC-certified products and vendors can be found at www.opsec.com.

Q: How do I block the latest virus that is spreading today?
A: In addition to the capabilities of SmartDefense discussed later in this book, if the virus is spread through http/ftp downloads and/or through e-mail attachments, then you can use VPN-1/FW-1 resources to block these connections. Using the Nimda virus as an example, you could use the SMTP file and/or MIME stripping to match MIME attachments of type audio/x-wav and the filename of readme.exe. Then use a URI wildcard resource to match HTTP, GETs to any host and any query match. Fill in the Path field with the following string: {*,cmd.exe,root.exe,admin.dll,readme.exe, readme.eml,default.ida}. Then just use these resources in rules that drop or reject the connections. For more information on blocking Nimda, see Check Point’s public knowledge base (support.checkpoint.com/public) article sk7473.

Q: Why do my users receive the error, “FW-1 Unknown WWW Server,” intermittently?
A: If your firewall cannot resolve the Website name to an IP (DNS), then it will present this error when a Web browser has the firewall defined as a proxy. Sometimes other problems with the HTTP security server may result in this error as well. You may want to try some of the objects_5_0.C changes or contact support for assistance.
Chapter 7 • Open Security (OPSEC) and Content Filtering

Q: My users are complaining that they cannot connect to certain sites and they are receiving the following message: “Web site found. Waiting for reply...” All of these sites seem to include a double slash in them. Is there a problem with the firewall?

A: If the site your users are trying to access contains a double slash within the URL GET command, then the GET command does not conform to RFC 2616 standards (according to Check Point), and the security server will not allow a connection. Your only option (if you must pass the site) is to bypass the security server by creating an HTTP accept rule specifically for this destination above any HTTP resource rules defined in your VPN-1/FW-1 security policy. See Check Point’s public knowledge base article skI3834 for more information.

Q: In FireWall-1 4.1, there were several objects.C file modifications for the HTTP security server that resolved several problems. Are the same changes available in NG?

A: Yes, most of the changes that you implemented in 4.1 can be used in NG as well. To edit the objects_5_0.C file, you need to use the dbedit utility in NG. Some changes are as follows.

```c
:http_disable_content_type (false)
:http_disable_content_enc (true)
:http_enable_uri_queries (false)
:http_max_header_length (8192)
:http_max_url_length (8192)
:http_avoid_keep_alive (true)
```

These are the default settings that are in the objects.C file in NG HF1:

```c
:http_allow_content_disposition (false)
:http_allow_double_slash (false)
:http_allow_ranges (false)
:http_avoid_keep_alive (false)
:http_block_java_allow_chunked (false)
:http_buffers_size (4096)
:http_check_request_validity (true)
:http_check_response_validity (true)
```
:http_cvp_allow_chunked (false)
:http_disable_ahttpdhtml (false)
:http_disable_automatic_client_auth_redirect (false)
:http_disable_cab_check (false)
:http_disable_content_enc (false)
:http_disable_content_type (false)
:http_dont_dns_when_star_port (false)
:http_dont_handle_next_proxy_pw (false)
:http_failed_resolve_timeout (900)
:http_force_down_to_10 (0)
:http_handle_proxy_pw (true)
:http_log_every_connection (false)
:http_max_auth_password_num (1000)
:http_max_auth_redirect_num (1000)
:http_max_connection_num (4000)
:http_max_header_length (1000)
:http_max_header_num (500)
:http_max_held_session_num (1000)
:http_max_real_num (1000)
:http_max_server_num (10000)
:http_max_session_num (0)
:http_max_url_length (2048)
:http_next_proxy_host ()
:http_next_proxy_port ()
:http_no_content_length (false)
:http_old_auth_timeout (0)
:http_process_timeout (43200)
:http_proxied_connections_allowed (true)
:http_query_server_for_authorization (false)
:http_redirect_timeout (300)
:http_servers (  
    :ers ()
    :Uid "{6CAC812A-202F-11D6-AB57-C0A800056370}*"  
)
:http_session_timeout (300)
:http_skip_redirect_free (true)
Chapter 7 • Open Security (OPSEC) and Content Filtering

.http_use_cache_hdr (true)
.http_use_cvp_reply_safe (false)
.http_use_default_schemes (false)
.http_use_host_h_as_dst (false)
.http_use_proxy_auth_for_other (true)
.http_weeding_allow_chunked (false)