FireWall-1 and SmartDefense

NG with Application Intelligence

CHECK POINT SECUREKNOWLEDGE DATABASE

For additional technical information about Check Point products, consult Check Point's SecureKnowledge at

http://support.checkpoint.com/kb/

Part No.: 700722
July 2003

Check Point SOFTWARE TECHNOLOGIES LTD.

We Secure the Internet.

Check Point Software Technologies Ltd.

U.S. Headquarters: 800 Bridge Parkway, Redwood City, CA 94065, Tel: (650) 628-2000 Fax: (650) 654-4233, info@CheckPoint.com
International Headquarters: 3A Jabotinsky Street, Ramat Gan, 52520, Israel, Tel: 972-3-753 4555 Fax: 972-3-575 9256, http://www.checkpoint.com

Table Of Contents

Chapter 3 Network Address Translation (NAT)

Chapter 4 Content Security

CHAPTER 1

FireWall-1 Access Control

In This Chapter

The Need for Access Control

As a network administrator you need the means to securely control access to resources such as networks, hosts, network services and protocols. Determining what resources can be accessed, and how, is the job of authorization, or Access Control. Determining "who" can access these resources is the job of User authentication, described in Chapter 4, "Authentication".

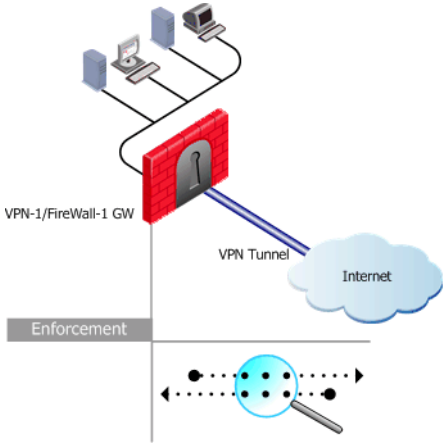
FireWall-1 Solution for Secure Access Control

In This Section

Access Control at the Network Boundary

A FireWall-1 Gateway (a “firewall”) at a network boundary acts as an enforcement point that inspects and provides access control for all traffic passing through the gateway (FIGURE 1-1). Traffic that does not pass though the enforcement point is not controlled.

FIGURE 1-1 A FireWall-1 enforcement point inspects all traffic that crosses it



The FireWall-1 administrator is responsible for implementing the company Security Policy. FireWall-1 allows the company Security Policy to be consistently enforced across multiple firewalls. To achieve this, an enterprise-wide Security Policy Rule Base is defined at the SmartCenter Server central management console. The SmartDashboard management client is used to install the Policy, and distribute it to the FireWall-1 gateways. Granular control of the Policy is possible by having specific rules apply only on specific enforcement points.

FireWall-1 provides secure access control through its granular understanding of all underlying services and applications traveling on the network. Stateful Inspection technology provides full application-layer awareness, and comprehensive access control for more than 150 pre-defined applications, services and protocols as well as the ability to specify and define custom services.

Stateful Inspection extracts state-related information required for security decisions from all application layers and maintains this information in dynamic state tables for evaluating subsequent connection attempts. For complete technical information about Stateful Inspection, see the Check Point Tech. Note at http://www.checkpoint.com/products/downloads/firewall-1_statefulinspection.pdf

The Security Rule Base

The Security Policy is implemented by defining an ordered set of rules in the Security Rule Base. A well-defined Security Policy is essential in order for FireWall-1 to be an effective security solution.

The fundamental concepts of the Security Rule Base is “That which is not explicitly permitted is prohibited”.

The Rule Base specifies what communication will be allowed to pass and what will be blocked. It specifies the source and destination of the communication, what services can be used, at what times, whether to log the connection and the logging level. Reviewing SmartView Tracker traffic logs is a very important aspect of security management, and should get careful attention.

FireWall-1 works by inspecting packets in a sequential manner. When FireWall-1 receives a packet belonging to a connection, it compares it against the first rule in the Security Rule Base, then the second, then the third, and so on. When it finds a rule that matches, it stops checking and applies that rule. If the packet goes through all the rules without finding a match, then that packet is denied. It is important to understand that the first rule that matches is applied to the packet, not the rule that best matches.

Example Access Control Rule

FIGURE 1-2 shows a typical Access Control rule. It says that HTTP connections that originate in one of Alaska_LAN group of hosts, to any destination, will be accepted, and logged.

FIGURE 1-2 Example Access Control Rule

SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK	INSTALL ON	TIME
Alaska.LAN	* Any	Any Traffic	TCP http	accept	Log	* Policy Targets	* Any

Rule Base Elements

A rule is made up of a number of Rule Base elements. Not all fields are always relevant in a given rule.

TABLE 1-1 Rule Base elements

Source and Destination	The source and destination is with respect to the originator of the connection. For applications that work in the client server model, the source is the client. Once the connection is accepted, packets in the connection are allowed in both directions. Source and destination can also be negated. You may for example find it convenient to specify that the source is NOT in a given network.
VPN	Configure whether the rule applies to any connection, either encrypted or clear, or only to VPN connections. To limit this rule to VPN connections, right-click and select **Replace...** .
Service	The service column allows predefined applications to be specified. It is also possible to define new services.
Action	A packet can either be Accepted, Rejected, or Dropped. The other possible Actions relate to authentication (see chapter 5“Authentication” on page 109). If a connection is Rejected, the firewall sends a RST packet to the originating end of the connection and the connection is closed. If a packet is Dropped then no response is sent and the connection will eventually time out.
Track	Various logging options are available. See the *SmartCenter* guide.
Install-On	Specifies the FireWall-1 Gateways on which the rule is to be installed. There may be no need to enforce a particular rule at every FireWall-1 Gateway. For example, a rule may allow certain network services to cross one particular gateway. If these services are not to be allowed to networks behind other FireWall-1 Gateways, the rule need not be installed on other gateways. For further information, see the *SmartCenter* guide.
Time	Specify the days and time of day at which this rule should be enforced.

Implied Rules

The Security Policy is made up of rules. Apart from the rules defined by the administrator, FireWall-1 also creates Implied Rules, which are derived from the Policy Global Properties. Implied rules are defined by FireWall-1 to allow certain connections to and from the firewall with a variety of different services. Examples of two important implied rules are ones that enable

- FireWall-1 Control Connections
- Outgoing Packets originating from the FireWall-1 gateway

There are also implied rules for other possible connection scenarios.

FireWall-1 creates a group of implied rules from the Policy Global Properties, that it places *first*, *last*, or *before last* in the Security Rule Base defined by the administrator. Implied rules can be logged. The rules are therefore processed in the following order:

1) Implied Rules defined as *first*. If an implied rule is *First*, the implied rule cannot be modified or overwritten in the Security Rule Base because no rules can be placed before it.

2) Rules 1 through n-1 in the Rule Base (assuming n rules).

3) Implied Rules listed as *Before Last*. Setting a property to *Before Last* makes it possible to define more detailed rules that will be enforced before this property.

4) Last rule (Rule n).

5) Implied Rules listed as *Last*. If a property is *Last*, it is enforced after the last rule in the Security Rule Base, which usually rejects all packets, and it will typically have no effect.

6) Implicit Drop Rule (no logging occurs).

Preventing IP Spoofing

Spoofing is a technique where an intruder attempts to gain unauthorized access by altering a packet's IP address to make it appear as though the packet originated in a part of the network with higher access privileges. It is important make sure that the communication does in fact originate from the apparent source.

Anti-spoofing verifies that packets are coming from, and going to, the correct interfaces on the gateway. It confirms that packets claiming to be from an internal network are actually coming from the internal network interface. It also verifies that, once a packet is routed, it is going through the proper interface.

A packet coming from an external interface, even if it has a spoofed internal IP address, will be blocked because the FireWall-1 anti-spoofing feature detects that the packet arrived from the wrong interface.

FIGURE 1-3 illustrates what anti-spoofing does.

On Alaska_GW, FireWall-1 checks that

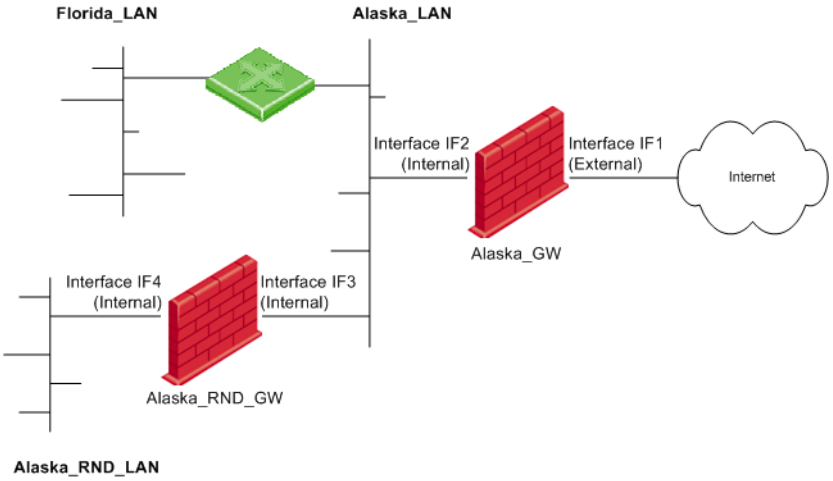
- All incoming packets to interface IF1 come from the Internet.
- All incoming packets to interface IF2 come from Alaska_LAN or Alaska_RND_LAN or Florida_LAN.

On Alaska_RND_GW, FireWall-1 checks that

- All incoming packets to interface IF3 come from Alaska_LAN or Florida_LAN or the Internet.
- All incoming packets to interface IF4 come from Alaska_RND_LAN.

When configuring anti-spoofing, you also need to specify whether the interfaces lead to the Internet or to an internal network. FIGURE 1-3 illustrates the appropriate settings.

FIGURE 1-3 Illustrating Anti-Spoofing



What are the Legal Addresses

Legal addresses are those that are allowed to enter a FireWall-1 interface. Legal addresses are determined by the topology of the network. When configuring anti-spoofing protection, the administrator must tell FireWall-1 what are the legal IP addresses behind the interface. This can be done automatically using the **Get Interfaces with Topology** option which automatically defines the interface with its topology, and creates network objects. FireWall-1 obtains this information by reading routing table entries.

New Services

In This Section

SSLv3 Service

It is possible to verify that SSL client connections are using version 3 or higher of the SSL protocol in order to prevent security problems known with earlier versions of SSL. SSLv3 enforcement is enabled using the **ssl_v3** service.

If the **ssl_v3** service is used in a rule, and an SSLv2 connection is attempted, the connection is rejected.

Many internet browsers use SSLv2. To allow their connections to pass through FireWall-1, use the **HTTPS** service in the Rule Base.

SSHv2 Service

It is possible to verify that SSH connections are using version 2 or higher of the protocol in order to prevent security problems known with earlier versions of SSH. SSHv2 enforcement is enabled using the **ssh_version_2** service.

If the SSHv2 service is used in a rule, SSHv1 connections are dropped.

FTP_BASIC Protocol Type

FTP_BASIC is a new protocol type. This protocol type enforces a reduced set of the FTP security checks done by the regular FTP protocol type. Using FTP_BASIC eliminates known connectivity problems with FTP implementations that are not fully RFC compliant. The following checks are NOT enforced by FTP_BASIC, and are enforced by the FTP protocol type:

- That every packet is terminated with a newline character, so that the PORT command is not split across packets. This protects against the FTP Bounce attack.
- Data connections to or from well-known ports are not allowed, to prevent the FTP data connection being used to access some other service.
- Bidirectional traffic on the data connection is not allowed, as it can be used improperly.

Domain_UDP Service

The **Domain_UDP** service provides access control for DNS.

- DNS performance when using this service has been improved. Many DNS connections are for queries which comprise one request and one reply packet. FireWall-1 normally maintains virtual DNS connections for the period of the UDP timeout. DNS verification speed can be improved by telling FireWall-1 to delete the connection as soon as it receives the reply packet. To do this, change the property `delete_on_reply (false)` to `true` using the Database Tool.
- DNS logs are more informative. For example, the domain of the device making a DNS query is now shown in the **Information** column.

- DNS verification of EDNS queries is supported. This allows use of BIND. EDNS headers are allowed if they contain all zeros, other than the field that controls the packet length (maximum payload size).

Blocking Visitor Mode

Introduction to TCPT

Visitor Mode and the TCP tunneling protocol (TCPT) were developed by Check Point to allow SecureClient connections from behind any gateway device with a restrictive outgoing Security Policy. An example of such as Security Policy is one that allows only HTTP and HTTPS (SSL) outgoing traffic, and prevents the various protocols (such as IKE) required for the secure connections.

Why Block Visitor Mode and Outgoing TCPT?

The VPN-1/FireWall-1 administrator can decide to block Visitor Mode by implementing a very restrictive outgoing Security Policy that allows ordinary HTTPS connections and disallows TCPT connections passing on the same port.

Visitor Mode and Incoming TCPT are allowed via the Gateway object. See the *Advanced Configuration* chapter of the *VPN-1* guide for details.

How VPN-1/FireWall-1 identifies TCPT

VPN-1/FireWall-1 performs content inspection in order to identify TCPT packets and reject them if necessary. It does not merely check the port.

The default port used by TCPT is 443, which is the same port used by SSL. This can be changed. (See “How to change the port used to block outgoing TCPT” on page 20.)

When to Block Outgoing TCPT

Only block TCPT if there is a rule that allows the port used by TCPT, for example, port 443. If there is no rule that allows the port used by TCPT, then it will be implicitly blocked, and there is no need explicitly block it.

There are a number of services that perform content inspection, rather than merely checking port numbers. If you block outgoing TCPT, and there is a rule that allows a service that uses the same port as TCPT, and that service performs content inspection, then both TCPT and that service will be blocked. The exception is the SSLv3 service. A rule that allows SSLv3, permits only SSL version 3 connections, and rejects TCPT.

Services that do content inspection are those that have a defined Protocol Type in the **TCP Service Properties>Advanced** window.

For configuration instructions, see “How to Block Visitor Mode (Blocking Outgoing TCPT)” on page 20.

Considerations for Access Control

In This Section

Spoof Protection

If you don’t protect your network against address spoofing, all your carefully crafted access control rules will be ineffective. It is easy enough for a malicious user to attempt to gain access by changing the source address of the packet. Make sure you configure anti-spoofing protection on every interface of the FireWall-1 gateway, including internal interfaces. For configuration details, see “Configuring Anti-spoofing” on page 19.

Simplicity

The key to a secure firewall is a simple Rule Base. The biggest danger to the security of your organization can be simple misconfiguration. Why should a malicious user try to sneak spoofed, fragmented packets past your firewall when you have accidentally allowed unrestricted messaging protocols? To keep your Rule Base simple, keep it short. The more rules you have, the more likely you will make a mistake. The fewer rules your Rule Base has, the easier it is to understand and maintain.

Basic Rules

Be careful to allow only the traffic that you want. Consider both traffic crossing the firewall that is initiated on the unprotected side of the firewall, and traffic initiated on the protected side of the firewall.

The following basic Access Control rules are recommended in every Security Rule Base:

- A Stealth Rule to prevent any direct access to the FireWall-1 Gateway.

- A Cleanup Rule to drop all traffic that is not permitted by the previous rules. There is an implied rule that does this, but the Cleanup Rule allows you to log any access attempts.

Remember the fundamental concept of a Rule Base: “That which is not explicitly permitted is prohibited”.

Rule Order

Rule order is critical. Having the same rules, but placing them in a different order, can radically alter how your firewall works. It is therefore best to place the more specific rules first, the more general rules last. This prevents a general rule being matched before a more specific rule, and protects your firewall from misconfigurations.

Topology Considerations: DMZ

If you have servers that are externally accessible from the internet, you should create a demilitarized zone (DMZ). Servers in the DMZ are accessible from any network, and all externally accessible servers should be in the DMZ. The purpose of the DMZ is to isolate all servers that are accessible from untrusted sources, like the Internet, so that if someone compromises one of those servers, the intruder will have only limited access to externally accessible servers. Servers in the DMZ should be as secure as possible. Do not allow the DMZ to initiate connections into the internal network, other than for specific applications such as UserAuthority.

The X11 Service

The X11 (X Window System Version 11) graphics display system is the de-facto graphics system in the Unix world. To allow X11, you must create a specific rule using the X11 service. When selecting **Any** as the **Source** or **Destination**, the X11 service is not included. This is because of the unusual nature of X11, by which the GUI application actually acts as the server, rather than the client.

When to Edit Implied Rules

Implied rules are controlled from the **Global Properties** window **FireWall-1 Implied Rules** page. In general, there is no need to change them. Some are best left unselected so that the property can be controlled with greater granularity via the Rule Base. For example, you may wish to allow ICMP pings across certain gateways only. The following are the recommended settings:

TABLE 1-2 FireWall-1 Implied Rules recommended settings

Implied Rule	Recommended Setting
Accept VPN-1 & FireWall-1 Control Connections	*First*
Accept outgoing packets originating from gateway	Unselected
Accept RIP	Unselected
Accept Domain Name Over UDP (Queries)	Unselected
Accept Domain Name over TCP (Zone transfer)	Unselected
Accept ICMP requests	Unselected
Accept CPRID connections (SmartUpdate)	*First*
Accept dynamic address Modules' DHCP traffic	*First*

Configuring Access Control

In This Section

Defining access Control Rules

An example Access control Rule is shown in Figure 1-2 on page 9. To define a rule:

1. Define the network objects for each network and host (for details, see *SmartCenter* guide).
2. From the menu, select **Rules > Add Rule** and choose one of **Bottom**, **Top**, **Below**, **Above**.
3. In the **Source** and **Destination** columns, right click and select **Add...**, choose a network object and click **OK**.

4 In the **Service** column, right click, select **Add...**, choose a service or a service group, and click **OK**.

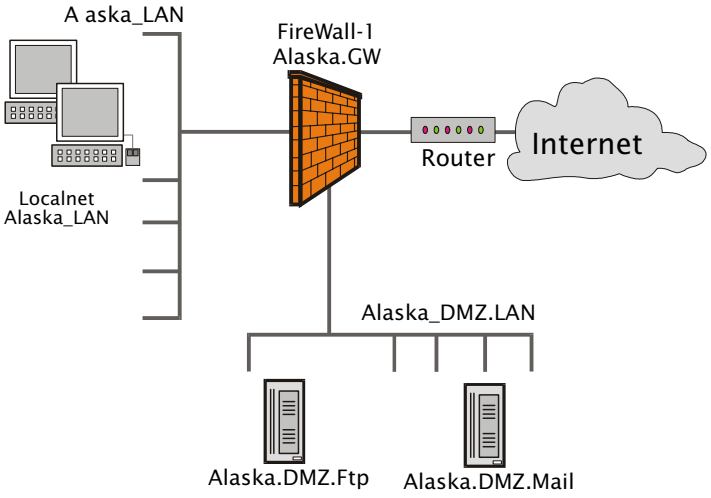
5 In the **Action** column, right click and select **Accept**, **Drop**, or **Reject**.

6 In the **Track** column, right click, select **Add...** and choose one of the tracking options.

Defining a Basic Policy

FIGURE 1-4 shows a network requiring an Access Control policy.

FIGURE 1-4 Sample network requiring an Access Control Policy



The Access Control Policy is required to

1) Allow internal users access to the World Wide Web.

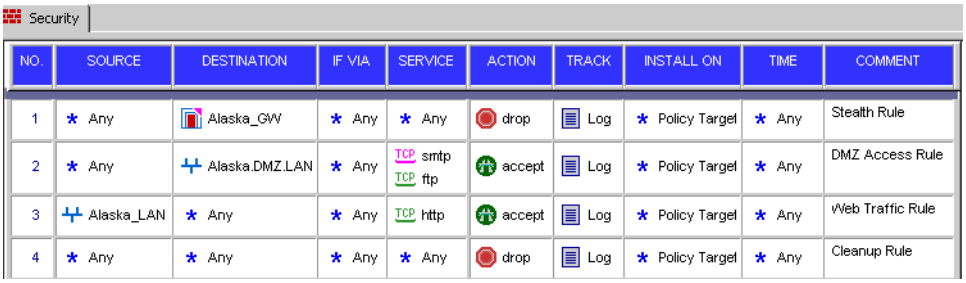
2) Allow all users access to the servers on the DMZ network.

3) Protect the network from outsiders.

The Policy also requires two basic rules: a Stealth Rule and a Cleanup Rule

To create the Policy, add rules in the SmartDashboard using the **Rules > Add Rules...** menu items, as detailed in “Defining a Basic Policy” on page 18. FIGURE 1-5 shows the resulting Access Control Security Rule Base.

FIGURE 1-5 Typical Access Control Security Rule Base



Configuring Anti-spoofing

Make sure you configure anti-spoofing protection on every interface of every FireWall-1 gateway, including internal interfaces. This basic configuration example shows how to set up anti-spoofing parameters on an external interface and the internal interface.

Define a Valid Address for the External Interface

1. In SmartDashboard, select **Manage > Network Objects**.
2. Select the Check Point Gateway and right click **Edit**.
3. In the Properties list, click **Topology**.
4. Click **Get> Interfaces** to read the interface information on the gateway machine.
5. Select the interface that faces the Internet and click **Edit**.
6. In the **Interface Properties** window, click **Topology**, and select **External (leads out to the internet)**.
7. Check **Perform Anti-Spoofing based on interface topology**, under **Spoof Tracking** select **Log**, and click **OK**.

Define a Valid Address for Internal Interfaces

8. Under the name column, select the internal interface, click **Edit**.
9. In the **Interface Properties** window, click **Topology**, and click **Internal (leads to the local network).**
10. Under **IP Addresses behind this interface**:
    - If there is only one network behind the interface, choose **Network defined by the interface IP and Net Mask**.

- If there is more than one network behind the interface, define a Group Network object that comprises all the networks behind the interface, choose **Specific** and select the group.

11 Check **Perform Anti-Spoofing based on interface topology**, under **Spoof Tracking** select **Log**, and click **OK**.

12 Repeat step 8 to step 11 for all internal interfaces.

13 Install the Security Policy.

Configuration of Visitor Mode Blocking

For background information about TCPT blocking, see “Blocking Visitor Mode” on page 14.

How to Block Visitor Mode (Blocking Outgoing TCPT)

To block outgoing TCPT, use the Database Tool on the SmartCenter Server, and locate, and change the following VPN-1/FireWall-1 property for every FireWall-1 enforcement module for which you wish to block outgoing TCPT:
`disable_outgoing_tcpt (false)`

Change the value of the property to `true`.

How to change the port used to block outgoing TCPT

To change the port used to block TCPT, use the Database Tool and locate the following VPN-1/FireWall-1 global property on the SmartCenter Server:
`tcpt_outgoing_port (443)`

Change the value of the property to the required port number.

CHAPTER 2

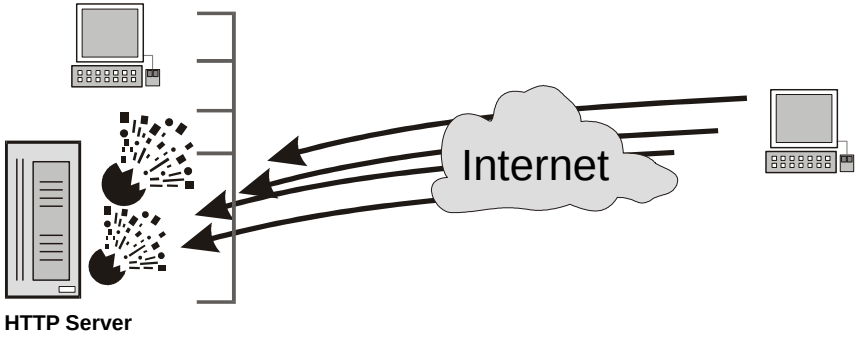
SmartDefense

In This Chapter

Need for Active Defense

There are many and increasing threats to network security, and the threats are increasing in sophistication as well as variety. The security administrator requires a way of detecting and preventing attacks, and of getting timely updates about new attacks and the means to prevent them.

FIGURE 2-1 Network Attacks



FireWall-1 Access Control prevents problematic connections from passing through a gateway, but attacks are characterized by the misuse of allowed traffic and services. For example, ICMP pings may be allowed, but a flood of them becomes a Denial Of Service attack if they succeed in preventing other connections. Another example is a SYN attack, which prevents a TCP/IP server from servicing other users. It is accomplished by not sending the final acknowledgment to the server's SYN-ACK response (SYNchronize-ACKnowledge) in the handshaking sequence, which causes the server to keep signaling until it eventually times out.

Content security, as provided by the virus checkers and the like is also not enough. It can inspect the content of individual packets, but is available only for specific services, and cannot detect patterns of malicious activity.

The SmartDefense Solution for Active Defense

In This Section

Introduction to SmartDefense

Check Point SmartDefense provides a unified security framework for various components that identify and prevent attacks. SmartDefense provides active defense for your network, even when the protection is not explicitly defined in the Security Rule Base. It unobtrusively analyzes activity across your network, tracking potentially threatening events and optionally sends notifications. It protects organizations from all known, and most unknown network attacks using intelligent security technology.

Keeping up-to-date with the latest defenses does not require up-to-the-minute technical knowledge. A single click updates SmartDefense with all the latest defenses from the SmartDefense website.

SmartDefense provides a console that can be used to

- Choose the attacks that you wish to defend against, and read detailed information about the attack.
- Easily configure parameters for each attack, including logging options.

- Receive real-time information on attacks, and update SmartDefense with new capabilities.

Note - SmartDefense is active only on Check Point Gateways of version NG FP2 and higher. Previous versions do not receive any of the SmartDefense configurations. New defenses added in the latest version will only be applied to Check Point Gateways of the latest version.

SmartDefense Subscription Service

SmartDefense functionality is freely included with FireWall-1. However, customers who purchase a SmartDefense subscription service can also obtain the following updates as soon as they are released, from the Check Point SmartDefense site http://www.checkpoint.com/techsupport/documentation/smartdefense/index.html

1) HTTP and CIFS worm patterns.

2) INSPECT file updates.

3) Dynamic Attack protection.

Subscribing customers can automatically update SmartDefense with a single click. Customers without a valid subscription license do receive on-line attack protection updates, but must manually define HTTP and CIFS worm patterns.

Categorizing SmartDefense Capabilities

Check Point SmartDefense protects organizations against attacks and other non legitimate or undesired network activity. Its capabilities can be categorized as follows:

- *Defense against attacks* page 23.
- *Implicit Defense* page 24.
- *Abnormal Behavior Analysis* page 24.

Some SmartDefense features provide more than one category of capability. The Initial Sequence Number Defender (ISN Defender) for example, provides both defense against a specific attack, and Implicit Defense.

Defense against attacks

Check Point SmartDefense protects organizations from known and unknown network attacks. Attacks are stopped at the gateway, and are prevented from affecting the target server.

SmartDefense is easy to configure, and defends against attacks while freeing the administrator from the need to understand technical attack details.

SmartDefense features protection against the following types of attack:

- Denial of Service Attacks
- TCP/IP Attacks
- Web and Application Vulnerabilities
- Network Probing
- HTTP worms

An Example: ISN Defender

A TCP connection is initiated using a three-way handshake. The client sends a SYN packet, the server replies with a SYN/ACK, and the client sends an ACK packet to acknowledge the connection. With each SYN/ACK, the server also generates a sequence number (SN) that identifies the connection.

The SNs are generated using a key of some sort, and for some operating systems it is possible to guess the next SN from the previous SN. An external client that can guess at a valid SN can open a connection by sending a SYN/ACK packet with a valid SN. This connection could be from a non-existent IP address, and may carry damaging data.

SmartDefense ISN Defender defends against this attack by replacing the SN generated by the server with a SN that it generates itself (using a strongly encrypted key).

Implicit Defense

Implicit Defense prevents information about network entities reaching the internet, where this information could be misused.

For example, when an internal server establishes a TCP connection, it sends successive SNs. For some operating systems these SNs can be used to identify the operating system. SmartDefense ISN Defender uses “fingerprint spoofing” to make it impossible for external clients to find out the operating system used by the internal servers, by replacing this fingerprint with another.

Abnormal Behavior Analysis

SmartDefense provides reporting and analysis of patterns of network behavior. It detects these patterns by analyzing logs sent to the management by the FireWall-1 enforcement modules. If a suspicious pattern is detected, the administrator can track the activity via a log or other kind of alert, depending on the configuration setting.

An example of this is Successive Events detection. A number of types of successive events can be configured. For each type of successive event, the activity is considered suspicious if the action is repeated more than the configured number, within a given time period.

The SmartDefense Tree Structure

The SmartDefense console is divided into a tree structure that classifies the defenses provided by SmartDefense. The following summarizes the major categories in the tree.

Note - The SmartDefense Update capability can add categories to the tree structure, as well as attack defenses.

Anti-Spoofing Configuration Status

This page indicates how anti-spoofing is configured on the gateways. It details the Check Point gateways on which anti-spoofing is not enabled, i.e., their interface's **IP address behind this interface** attribute is configured as *Not Defined*. You can change the settings by reconfiguring the individual gateways

Network Security

Denial of Service

Denial of Service (DoS) attacks are aimed at overwhelming the target with spurious data to the point where it is no longer able to respond to legitimate service requests. The attacks in this section exploit bugs in operating systems to remotely crash the machines.

IP and ICMP

This page allows you to enable a comprehensive sequence of layer 3 tests (IP and ICMP protocols).

For example, the fragmentation timeout logs feature generates logs when detecting packets, purposefully fragmented for a FireWall bypassing or Denial of Service attack.

TCP

VPN-1/FireWall-1 is able to identify the basic IP based protocols and analyze a packet in order to verify that it contains allowed options only.

In order to verify that TCP packets are legitimate, the following tests are conducted:

- protocol type verification
- protocol header analysis
- protocol flags analysis and verification

SYN Attack Protection prevent attacks in which TCP connection initiation packets are sent to the server in an attempt to cause Denial of Service.

The sequence verifier is a mechanism matching the current TCP packet's sequence number against a TCP connection state. Packets that match the connection in terms of the TCP session but have incorrect sequence numbers are either dropped or stripped of data.

Fingerprint Scrambling

It is sometimes possible to identify the operating system used by a machine or to impersonate an existing connection, by means of a fingerprint that characterizes the operating system or the connection. SmartDefense can prevent this by distorting the fingerprint to make such identification impossible.

Successive Events

Successive Events detection (formerly known as Malicious Activity Detection) provides a mechanism for detecting malicious or suspicious events and notifying the security administrator.

Successive Events detection runs on the SmartCenter Server and analyses logs from VPN/FireWall enforcement modules, by matching log entries to attack profiles.

The security administrator can modify attack detection parameters, turn detection on or off for specific attacks, or disable the Successive Events feature entirely.

Logs which do not reach the SmartCenter Server (for example, local logs and logs sent to Log Server) are not analyzed.

Dynamic Ports

A number of applications (such as FTP under heavy load, and SIP protocols) can set up connections by opening ports dynamically. These ports can turn out to be the same as those used by one of the pre-defined services in the SmartDashboard. Use this page to define whether or not a connection with a dynamically opened port that is the same as a pre-defined service port, will be dropped. Also use this page to choose whether or not to drop dynamic port connections that use low ports (below 1024).

Application Intelligence

Web

HTTP (HyperText Transport Protocol) is the communications protocol used to connect to servers on the World Wide Web. Its primary function is to establish a connection with a Web server and transmit HTML pages to the client browser.

This page allows you to configure various protections related to the HTTP protocol.

For example, the general HTTP Worm Catcher eliminates most attempts to exploit the security vulnerabilities in HTTP servers or clients, by matching to HTTP header patterns specified using regular expressions.

It is also possible to implicitly activate the security servers on all HTTP traffic, regardless of the Rule Base.

Mail

The SMTP security server allows strict enforcement of the SMTP protocol. It protects against malicious mail messages, provides SMTP protocol centered security, prevents attempts to bypass the Rule Base using mail relays, and prevents Denial of Service and spam mail attacks. Usually the security server is activated by specifying resources or authentication rules in the Security Rule Base.

These pages allow you to select what types of enforcement will be applied to SMTP connections passing through the security server. Clicking **Configuration applies to all connections** will forward all SMTP connections to the SMTP security server and enforce the defined settings on all connections, without having to define a resource in the Rule Base. Clicking **Configurations apply only to connections related to rule base defined objects** mean that these configurations will apply only to SMTP connections for which a resource is defined in the Rule Base.

FTP

These pages allow you to configure various protections related to the FTP protocol.

For example, preventing FTP port overflow checks foils any attempt to use an FTP server as an agent for a malicious operation.

DNS

The DNS protocol is used to identify servers by their IP addresses and aliases. DNS protocol messages can be transported over TCP or UDP.

This option checks that all the connections on the DNS port over UDP are DNS-related. In addition, certain restrictions are imposed on the type of data allowed in queries and answers.

VoIP

SIP (Session Initiation Protocol) is a Voice over IP protocol, transported over UDP. If this option is selected and there are explicit SIP rules in the Rule Base, SmartDefense will validate the SIP headers and look for invalid characters inside them.

How SmartDefense Works

A FireWall-1 gateway at a network boundary acts as an enforcement point, and controls all traffic passing through the network boundary. Some of the SmartDefense capabilities are enforced at the gateway, and are distributed as part of the Security Policy to each enforcement point from the SmartCenter Server. SmartDefense blocks attacks at the FireWall-1 gateway using Check Point's Stateful Inspection technology.

Others capabilities such as Abnormal Behavior Analysis are provided at the SmartCenter Server.

Planning Considerations for SmartDefense

SmartDefense is an easy and intuitive system to use. The planning considerations when deciding which SmartDefense features to use depend on the performance cost of selecting features versus the security advantages to your organization.

For the HTTP Security Server, it is possible to optimize performance by using the FireWall-1 kernel to perform most security checks, though the kernel does not perform all the checks that the Security server is capable of.

For the FTP and Mail Security Server you can choose whether to perform security checks using the Security Server on all connections, or only on connection for which a resource is specified in the Rule Base. Performing Security Server checks on all connections requires more system resources to maintain performance.

Configuring SmartDefense

Configuring SmartDefense is very simple. Proceed as follows:

1. In the SmartDashboard toolbar, click the SmartDefense icon
2. In the **SmartDefense Settings** window, select the SmartDefense category to view information about the category. To view details of a specific attack, click **[+]** to expand the branch, and select the attack.
3. Check the attacks you wish to defend against, and configure **Settings** for the categories and the specific attacks.
4. Install the Security Policy. You need to reinstall the Security Policy in order to implement changes to the SmartDefense configuration.

Updating SmartDefense with the Latest Defenses

To obtain updates of all the latest defenses from the SmartDefense website, open the **SmartDefense Settings>General** page, and click **Update smartDefense**.

Configuration Example: Protecting Against SYN Attacks

SYN Attack Description

SYN attack prevents a TCP/IP server from servicing other users. It is accomplished by not sending the final acknowledgment to the server's SYN-ACK response (SYNchronize-ACKnowledge) in the handshaking sequence, which causes the server to keep signaling until it eventually times out. The source address from the client is, of course, counterfeit. SYN flood attacks can either overload the server or cause it to crash.

SYN Attack Configuration

Defense against SYN Attacks can be configured in SmartDefense for all enforcement modules. It is also possible to configure specific SYN attack settings per enforcement module.

1. In the SmartDashboard toolbar, click the SmartDefense icon
2. In the **SmartDefense Settings** window, select the **TCP** category. Click **[+]** to expand the branch, and select **SYN Attack**.
3. Configure **Settings**.

   In SmartDefense check **Override module's SYNDefender configuration** to activate protection for all Modules. Check **Activate SYN Attack Protection**, and click **Configure**. Now select the desired parameters.

   If you wish to have specific SYNDefender settings per Module, do not check **Override module's SYNDefender configuration**. In that case, the settings that apply are configured per Module in the **Check Point Gateway** object, **Advanced > SYNDefender** page. For Modules that do not have specific SYNDefender settings, you can **Configure** settings for **Early Versions SYNDefender configuration**.
4. Install the Security Policy.

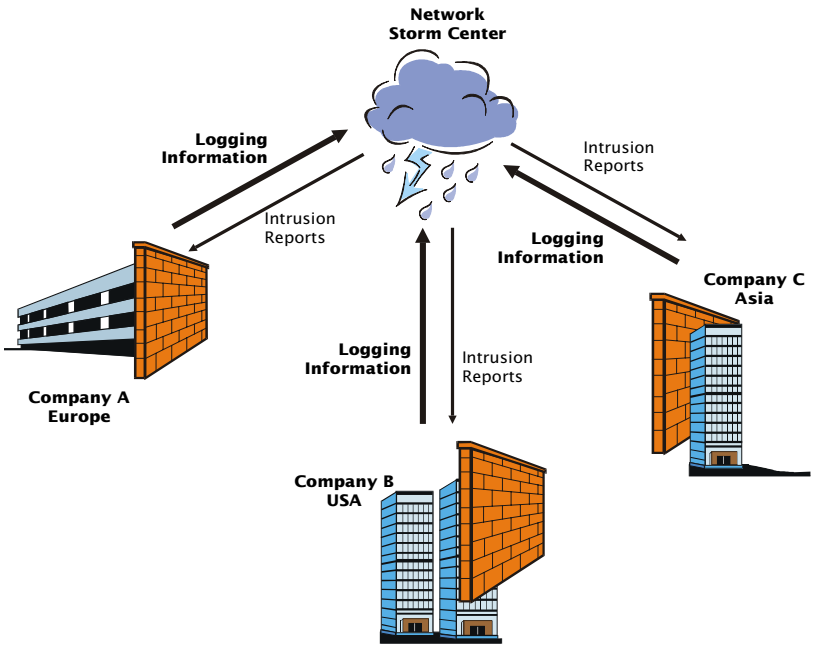
SmartDefense StormCenter Module

In This Section

The Need for Cooperation in Intrusion Detection

The range and sophistication of the techniques used by hackers and crackers to penetrate private networks is increasing all the time. Very few organizations can hope to maintain up-to-the-minute protection against the latest attacks. Network Storm Centers are collaborative initiatives that have been set up to help the beleaguered Security Administrator fight back. Storm Centers gather logging information about attacks. This information is voluntarily provided by organizations from across the world for the benefit of all. Storm Centers collate and present report on real-time threats to network security in a way that is immediately useful.

FIGURE 2-2 Cooperation between organizations and the Storm Center



Check Point Solution for Storm Center Integration

Introduction

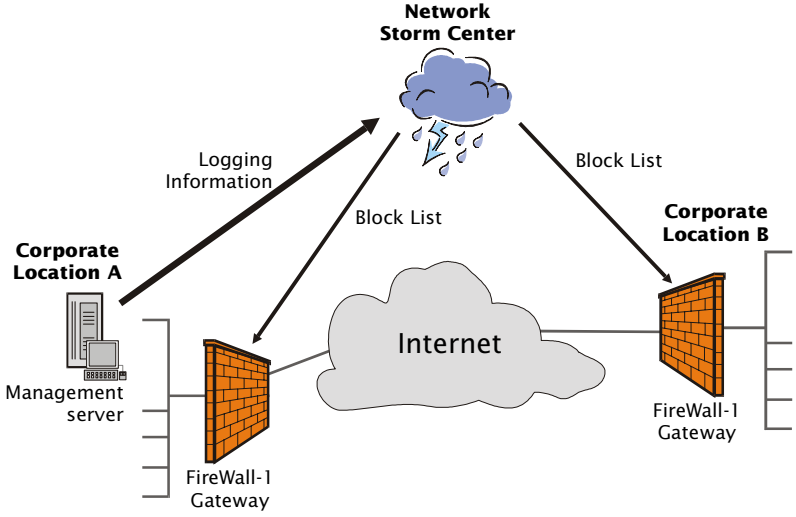
The SmartDefense Storm Center Module is included in the standard FireWall-1 product installation. It enables a two way information flow between the network Storm Centers, and the organizations requiring network security information.

One of the leading Storm Centers is SANS Dshield.org http://secure.dshield.org/. DShield.org gathers statistics and presents it as a series of reports at http://secure.dshield.org/reports.html.

Check Point SmartDefense integrates with the SANS DShield.org Storm Center in two ways, illustrated in FIGURE 2-3.

- The DShield.org Storm Center produces a Block List report, which is a list of address ranges that are worth blocking. This Block List is frequently updated. The SmartDefense Storm Center Module retrieves and adds this list to the Security Policy in a way that makes every update immediately effective.
- You can decide to send logs to the Storm Center in order to help other organizations combat the threats that were directed at your own network. You can decide which logs to send by selecting the rules for which you want to send logs.

FIGURE 2-3 How the Block List is Received and Logs are Submitted

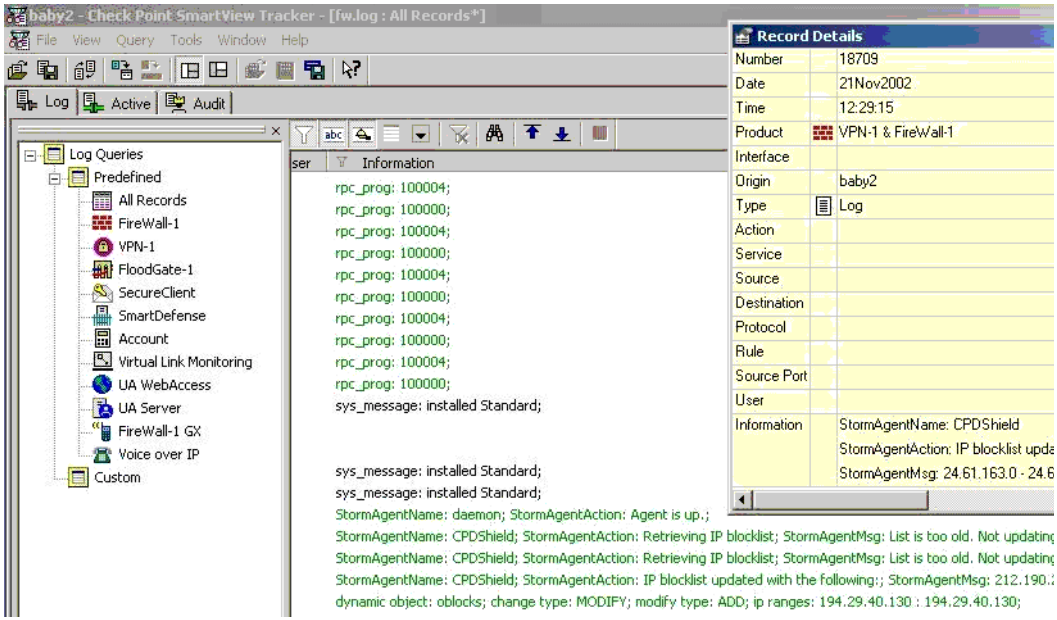


How the Block List is Received

The Security Administrator defines a Dynamic Object called CPDShield (the name is fixed) in the SmartDashboard, and places it in a Rule that defines what to do with the communication from the addresses in the Dynamic Object (typically, the traffic will be dropped), and installs the Policy on the FireWall-1 Gateways.

An agent (daemon) on each FireWall-1 Gateway on which the Storm Center Module is installed receives the Block List of malicious IP addresses from http://secure.dshield.org/block_list_info.html via HTTPS. Every refresh interval (the default is three hours), the agent takes the Block List, and "populates" the Dynamic Object with the IP address ranges in the Block List. This process is logged in the SmartView Tracker, in the FireWall-1 log, as shown in FIGURE 2-4.

FIGURE 2-4 Showing the retrieval of the Block List in the SmartView Tracker



How Logs are Submitted to the Storm Center

The Security Administrator decides which logs should be submitted. For example, it is possible to specify that logs from Rule 4, Rule 5 and Rule 12 will be submitted. Logs of detected HTTP Worm patterns can also be submitted.

A log submitting agent (daemon) on the SmartCenter Server generates two kinds of logs. As well as regular logs, a compact log digest is created. The digest includes only the number of Drops and Rejects per port.

The Storm Center tells the log submitting agent to send either regular logs, or digests, or both kinds of log.

The log submitting agent sends to the Storm Center the logs chosen by the Security Administrator, of the type requested by the Storm Center. Log submission is done via HTTPS POST. The log submitting agent is an OPSEC compliant LEA client. The logs are compressed into a database.

What a Submitted Log Contains

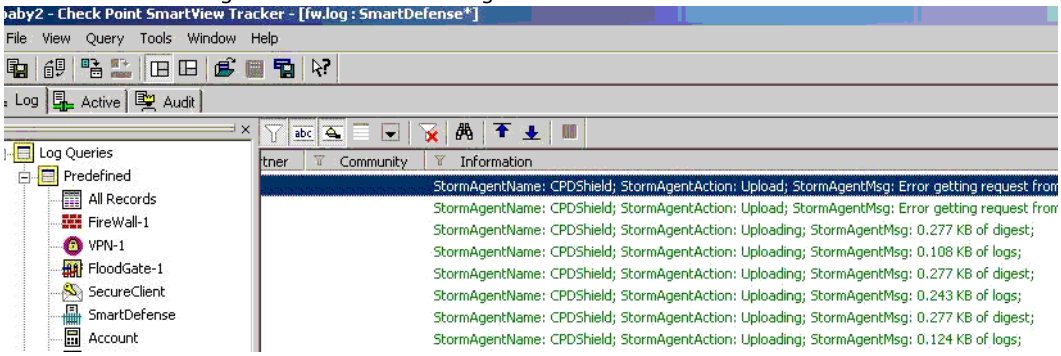
The logs that are submitted to the Storm Center contain the following information:

- Connection parameters: Source IP Address, Destination IP Address, Source Port, Destination Port (that is, the Service), IP protocol (such as UDP, TCP or ICMP).
- Rule Base Parameters: Time, action.
- A detailed description of the log.

For HTTP Worm patterns, the log contains the same connection parameters, the same Rule Base parameters, and also the name of attack and the detected URL pattern.

Submitted logs are SmartDefense logs, as shown in FIGURE 2-5.

FIGURE 2-5 Showing the submission of logs to the Storm Center in the SmartView Tracker



Removing Identifying Information from the Submitted Log

It is possible to delete identifying information from the destination IP address in the submitted log, by specifying a designated number of bits to mask. The destination IP addresses identify your organizations IP addresses because the logs are typically collected from attacks that come from outside the organization and are directed towards internal IP addresses.

The mask can be used to delete as many bits as desired from the internal IP addresses. A zero bit mask obscures the whole of the IP address. A 32 bit mask reveals the whole of the internal IP address. An 8 bit mask reveals 8 valid bits, and converts an IP address such as 192.168.46.88 to 0.0.0.88.

How Authenticity is Assured

The Block List and the Submitted logs are securely transferred and authenticated via SSL. The Certificate of the Storm Center Certificate Authority comes with the Storm Center Module, and is stored locally. The locally stored certificate is used for two purposes:

1) To check the authenticity of the origin of the received Block List, by verifying the validity of the certificate received with the Block List.

2) To establish an SSL connection with the Storm Center when submitting logs, while assuring that the logs are indeed sent to the Storm Center and to no one else.

The Certificate Authority of SANS DShield.org is Equifax. The file name of the locally stored certificate is `equifax.cer`, and it is stored in the `conf` directory of the Storm Center Module installation.

DShield.org authenticate the submitters of logs with a username and password that submitters obtain when registering with DShield.org.

Size of Logs and Effect on FireWall-1 Performance

Receiving the Block List has no effect on FireWall-1 performance because only a very small amount of data is received.

The submitted log is only a small subset of the full SmartDefense log, and is compressed. The size of the log depends on the log interval, and the maximum size of the log database. As a rough guide, 10,000 lines of logs take up 200 KB.

Planning Considerations

Where to Place the Block List Rule

Correct placement of the Block List Rule is crucial for effective operation of the Storm Center Module. Place the Block List rule as high as possible in the Security Rule Base, but below all authentication rules, and any other rules you are absolutely certain have a reputable Source. If the Rule is placed too low it will have limited effect. If it is placed too high, valid users may be blocked.

Which Logs to send to the Storm Center

Storm Centers have a special interest in receiving logging information about:

1) Unwanted port 80 traffic reaching the organization.

2) The Drop All rule (the last Rule in the Rule Base, that drops any traffic not explicitly allowed in previous rules).

3) The Rule containing the Dynamic Object, which drops all traffic from any location in the Block List.

4) HTTP Worms, caught by the SmartDefense General HTTP Worm Catcher.

Which Logs NOT to send to the Storm Center

Do not send logs from rules that log internal traffic.

Which Identifying Information to Remove from Submitted Logs

Decide on what part of your organizations IP addresses to block from the submitted logs. If all your internal addresses are private, non-routable addresses, you may not feel it is necessary to mask the addresses. On the other hand, even non-routable addresses can reveal information about your internal network topology.

Configuring the Storm Center Module

1. To send logs to DShield.org, you must register at http://secure.dshield.org/. You will receive a username and password. You can receive the Block List without registering.

2. Next, on machines on which you installed the Storm Center Module, configure
   - the Block List receiving agent on the FireWall-1 Gateway machine(s), and
   - the log submitting agent on the SmartCenter Server machine(s) (only if you wish to submit logs to the Storm Center).

   Where the SmartCenter Server and the FireWall-1 Gateway are on the same machine, configure parameters for both submitting logs and for receiving the Block List.

Configuring the Block List Receiving Agent

3. If you wish to change the default values of the parameters for receiving the Block List, then on the FireWall-1 Gateways on which you installed the Storm Center Module, edit the configuration file `Stormcenters.conf` in the `%FWDIR/conf` directory. These parameters are:

```
:DataURL ("https://secure.dshield.org/feeds/block.txt")
:CertificateFileName (equifax.cer)
:SignatureURL (dummy)
:SigPubKeyFileName (dummy)
:RefreshInterval (180)
:ExpireDataAfter (3)
```

4 Enable the Block List Receiving (downstream) Agent in the `%FWDIR/bin` directory:

```
stormc config down
```

Note - To turn off the Log Submitting (upstream) Agent and leave the Block List Receiving (downstream) Agent running, run `stormc config down` followed by `cprestart`.

5 Stop and restart FireWall-1 (perform `cpstop` and `cpstart`) to activate the configuration change. Whenever the machine is rebooted, the agent will be automatically activated together with FireWall-1.

Configuring the Log Submitting Agent

If you wish to submit logs to the Storm Center, perform the following steps on the SmartCenter Server machine(s) on which you installed the Storm Center Module. You must be logged in as root:

6 Edit the configuration file `Stormcenters.conf` in the `%FWDIR/conf` directory, and configure the parameters for submitting logs. These parameters are

```
:UploadURL ("https://secure.dshield.org/cgi-bin/stormcenter.pl")
:UploadInterval (360)
:UploadDBName (upload.db)
:MaxDBSize (4000)
:LogRules ("3 4 12")
:LogWormPatterns (1)
:UserName (john)
:Password (doe)
:DestIPMask (0)
```

Some parameters for submitting logs are pre-configured with default values that can be changed. You must specify the appropriate rule numbers in:

`:LogRules ("3 4 12")`

Note - If the numbering of the Log submitting rules changes, for example, when rules are added or deleted, you will need to reconfigure the Storm Center configuration file `Stormcenters.conf`.

You must also specify appropriate values for the following parameters:

`:LogWormPatterns (1)`

Specify '1' to collect and send Worm Catcher logs, or '0' to ignore.

```
:UserName (john)
:Password (doe)
```

```
:DestIPMask (0)
```

7 In order to see the log submission logs in SmartView Tracker with the proper IP address in the origin field, edit the `conf/stormc_opsec.conf` file, and replace the IP addresses for `lea_server` and `ela_server` with the IP address of the SmartCenter Server machine.

The `stormc_opsec.conf` file looks like:

```
lea_server    ip        127.0.0.1
lea_server    auth_port    18184
lea_server    auth_type    local

ela_server    ip        127.0.0.1
ela_server    auth_port    18187
ela_server    auth_type    local
```

8 Enable the Log Submitting (upstream) Agent in the `%FWDIR/bin` directory:

```
stormc config up
```

Note - To turn off the Log Submitting (upstream) Agent and leave the Block List Receiving (downstream) Agent running, run `stormc config down` followed by `cprestart`.

9 Stop and restart FireWall-1 (perform `cpstop` and `cpstart`) to activate the configuration change. Whenever the machine is rebooted, the agent will be automatically activated together with FireWall-1.

10 Stop and restart FireWall-1 (perform `cpstop` and `cpstart`).

Configuring the Security Policy

After installing and configuring the Storm Center Module, configure the Security Policy:

11 FireWall-1 Gateways and SmartCenter Server(s) on which the Storm Center Module is installed must be able to connect to the Storm Center using HTTPS. Define an appropriate Rule if necessary.

12 To Receive the Block List, define a Dynamic Object in the Security Rule Base called **CPDShield** (case sensitive).

13 Add the Block List Rule, as shown in FIGURE 2-6. Place the Block List rule as high as possible in the Security Rule Base, but below all authentication rules, and any other rules you are absolutely certain have a reputable Source. (See “Where to Place the Block List Rule” on page 34.)

FIGURE 2-6 The Block List Rule.

Source	Destination	Service	Action	Install On	Track	Comment
CPDShield	Any	Any	Drop	Storm Center Modules	Log	Block List Rule

Warning - Be sure to install the Block List Rule with the Dynamic Object ONLY on FireWall-1 Gateways on which the Storm Center Module is installed, as in FIGURE 2-6.

14 Decide which rule logs you want to send to the Storm Center, and set the **Track** option in those rules to anything other than NONE. If necessary, create an appropriate rule.

15 Install the Security Policy.

Storm Center Parameters

Block List Receiving Parameters

Parameter	Meaning
DataURL	The URL from which the IP Block List is fetched.
CertificateFileName	The filename of the Certificate Authority certificate that issued the Storm Center's certificate. This certificate is used to validate the DataURL using SSL. The certificate should be placed in the conf directory of the product (a full path name can also be specified.
SignatureURL and SigPubKeyFileName	For future use.
RefreshInterval	Specifies the interval in minutes for fetching a new Block List and updating the dynamic object. Allowed range: 1 to 10080.
ExpireDataAfter	Specifies the time in days after which the retrieved addresses will be erased (in other words, when the addresses in the Bloc List will be allowed), or a published Block List will be out of date (and therefore not used). Allowed range: 1 to 365.

Log Submitting Parameters

Parameter	Meaning
UploadURL	Specifies the URL to which collected data is uploaded.
CertificateFileName	Same as the Block List receiving parameter.
UploadInterval	Specifies the interval in minutes for uploading collected data. Allowed range: 1 to 10080.
UploadDBName	Specifies the name of the file which will be used to store collected logs.
MaxDBSize	Specifies the maximum size in KB of the file where logs are collected. When reached, collection is paused until space is available (for example, after data has been uploaded).
LogRules	Specifies the rules from which logs will be collected (specified rules must have a track setting other than NONE).

Parameter	Meaning
`LogWormPatterns`	Specify '1' to collect and send Worm Catcher logs, or '0' to ignore.
`UserName, Password`	Should contain the username and password respectively, for uploading data. (They should be provided upon registration to the Storm Center.)
`DestIPMask`	Specifies the number of bits to mask for the destination IP addresses stored in the collected logs. A zero bit mask obscures the whole of the IP address. A 32 bit mask reveals the whole of the internal IP address. When a value of 24 is specified, 10.11.12.13 will be sent as 0.11.12.13.

CHAPTER 3

Network Address Translation (NAT)

In This Chapter

The Need to Conceal IP Addresses

In an IP network, each computer is assigned a unique IP address that defines both the host and the network. Many computers in an organization have private, non-routable IP addresses, but nevertheless require access to the Internet. In most cases it is impossible to simply give them Internet-routable IP addresses, due to the lack of available public IP addresses, and administrative constraints.

IPv4 (the current version of IP) provides only 32 bits of address space, so available IP addresses are becoming scarce, most having already been assigned. Internet Service Providers will usually allocate only one or a few addresses at a time. Larger companies may purchase several addresses for use, but purchasing addresses for every computer on the network is usually impossible.

Even if public IP addresses become available, changing the addresses of every machine in a large network can be an administrative nightmare, being both labor intensive and time consuming.

Whether computers have a routable or a non-routable addresses, the administrator may wish to conceal their real addresses for security reasons. The administrator may wish to ensure that addresses cannot be seen from outside the organization, or even from other

parts of the same organization. Making a network's internal addresses public knowledge can reveal the topology of the network. Hiding this information can only enhance security.

Check Point Solution for Network Address Translation

In This Section

Public and Private IP addresses

Public IP addresses are those that are routable on the Internet. RFC 1918 documents private address spaces can be used on internal networks that will not have hosts directly connected to the Internet. The Internet assigned Numbers authority (IANA) has set aside the following three blocks of IP addresses for internal (private) network use:

- Class A network numbers: 10.0.0.0–10.255.255.255
- Class B network numbers: 172.16.0.0–172.31.255.255
- Class C network numbers: 192.168.0.0–192.168.255.255

In an intranet that uses private addresses, a FireWall-1 NAT gateway is put in place to connect the intranet to the Internet. The FireWall-1 **Global Properties>NAT** page specifies the address ranges that FireWall-1 considers private (non-unique).

NAT in FireWall-1

Network Address Translation (NAT) involves replacing one IP address with another. NAT can change both the source and destination address inside the packet. This means that a packet that is sent from the internal (protected) to the external (unprotected) side of the firewall appears to the destination as if it came from a different address, and packet that is sent from the external to the internal side of the firewall will arrive at the correct address.

FireWall-1 supports two kinds of NAT:

- *Static NAT*, where each private address is translated to a corresponding public address. In a typical Static NAT scenario with a number of machines in an internal network, the address of each machine is translated to a different public IP address. It is a many-to-many translation. Static NAT allows machines on both sides of the FireWall-1 Gateway to initiate connections, so that, for example, internal servers can be made available externally.
- *Hide NAT*, where a single public address is used to represent multiple computers on the internal network with private addresses. Hide NAT is a many-to-one translation. Hide NAT allows connections to be initiated only from the protected side of the FireWall-1 Gateway.

NAT can be performed on Check Points, Nodes, Networks, Address Ranges, and Dynamic objects.

NAT can be defined either *automatically*, via the network object, which automatically adds rules to the *Address Translation Rule Base*, or *manually*, by defining rules in the Address Translation Rule Base.

Manually creating NAT Rules adds extra flexibility. For example, as well as translating IP addresses, it is possible to translate the Service, in other words the destination port numbers. *Port number translation* is a type of Static NAT, in which one port number is translated to another port number.

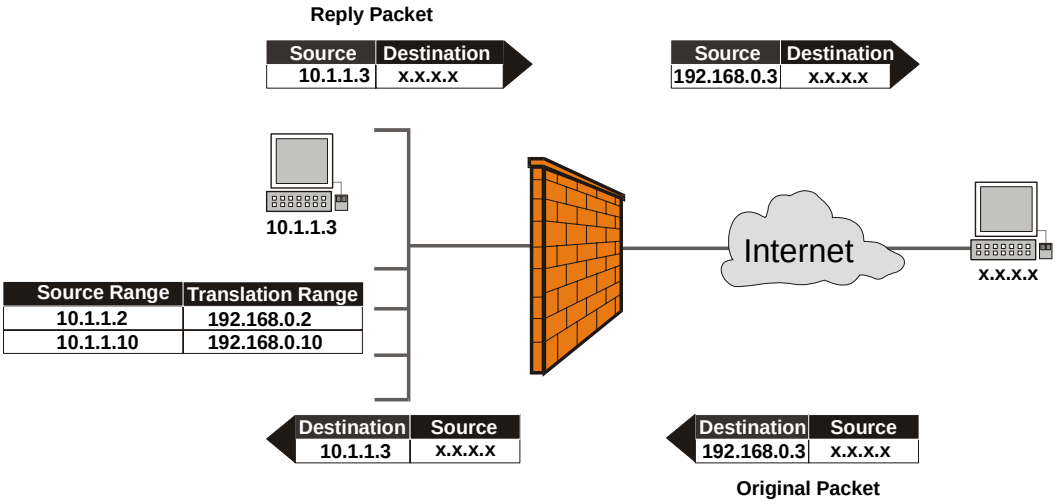
Static NAT

Static NAT translates each private address to a corresponding *public* address.

- Static NAT on a node translates the private address of the node to a public address.
- Static NAT on a network or address range translates each IP address in the network or range to a corresponding public IP address, starting from the defined Static IP address.

In FIGURE 3-1, an address range (10.1.1.2 to 10.1.1.10) is hidden behind IP address a NAT range (192.168.0.2-192.168.0.11). A connection is shown originating at 10.1.1.3, and the source and destination translation for the original and reply packet.

FIGURE 3-1 Static NAT on an Address Range



Hide NAT

With a NAT gateway, it is possible to share a single *public* address with multiple computers on your intranet that have private addresses. The Internet is unaware of the division you have created between the Internet and your intranet, and sees your multiple computer connection as simply a single connection.

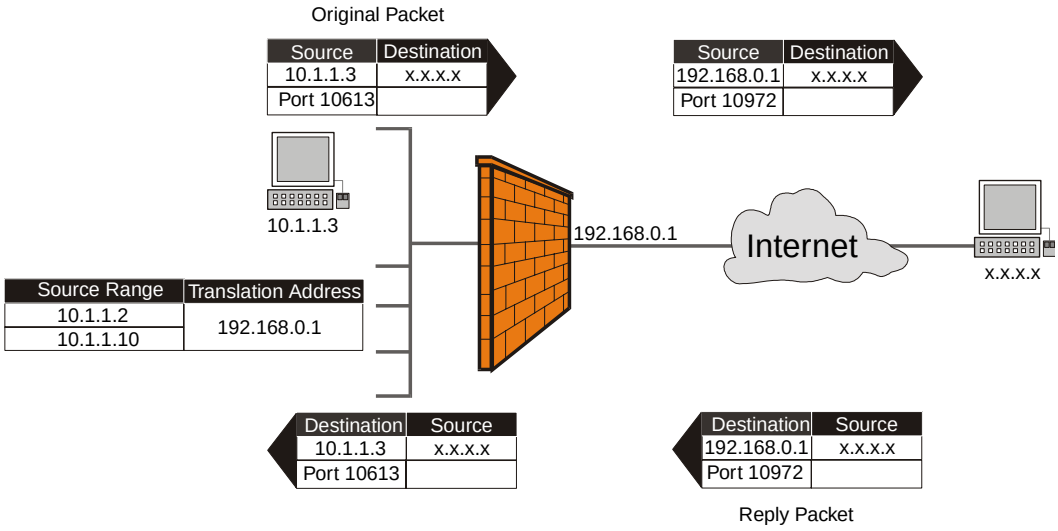
Hide NAT allows only connections that originate on the internal network. This lets an internal host initiate a connection to both inside and outside the intranet, but a host outside the network cannot initiate a connection to an internal host.

The Hide Address is the address behind which the internal network, address range or node is hidden. You can choose to hide the internal address(es)

- behind a *virtual* IP address, which is a public (routable) IP address that does not belong to any real machine, or
- behind the IP address of the FireWall-1 interface through which the packet is routed out (what used to be known as “Hiding behind IP address 0.0.0.0”).

In FIGURE 3-2, an address range (10.1.1.2 to 10.1.1.10) is hidden behind the address of the external FireWall-1 interface (192.168.0.1). A connection is shown originating at 10.1.1.3, and the source and destination translation for the original and reply packet.

FIGURE 3-2 Hide NAT on An Address Range



How Hide NAT Works

In Hide Mode, the source port numbers of the packets are modified. When return packets enter a firewall, FireWall-1 determines by port number to which internal machines the packets are destined. Port numbers are dynamically assigned from two pools of numbers:

- from 600 to 1023
- from 10,000 to 60,000

Port numbers are almost always assigned from the second pool. The first pool is used for only three services: rlogin (destination port 512), rshell (destination port 513) and rexec (destination port 514). IF the service of the connection is one of these three, AND the original source port is less than 1024, THEN a port number is assigned from the first pool. This behavior is configurable.

FireWall-1 keeps track of the port numbers assigned, so that the original port number is correctly restored for return packets. A port number currently in use is not assigned again to a new connection.

Hide NAT has a capacity of 50,000 connections per *server*. This means that the Hide NAT capacity limit is only reached if more than 50,000 connections from Hide NATed internal clients are simultaneously directed at a *single* server on the unprotected side of the FireWall-1 Gateway—a webcast of a wildly popular basketball game, perhaps?

Automatic and Manual NAT Rules

NAT can be defined *automatically* via the network object (Node, Network or Address Range). When you define NAT via the network object, rules are automatically added to the Address Translation Rule Base

You can *manually* specify NAT rules, by adding or editing NAT rules to the Address Translation Rule Base. FireWall-1 validates manual NAT rules, helping to avoid mistakes in the setup process. Creating manual NAT Rules gives maximum control over the way NAT will function. You can specify the source, destination and service separately for the original and the translated packet.

When creating Manual NAT Rules, you must explicitly define the translated network objects in addition to the original objects. With Automatic rules this is not necessary.

Automatic NAT rules cannot be edited in the Address Translation Rule Base.

Address Translation Rule Base

The Address Translation Rule Base is shown in FIGURE 3-3.

FIGURE 3-3 Address Translation Rule Base

Security - Standard | Address Translation - Standard | VPN Manager | Desktop Security - Standard

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON	COMMENT
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE		
1	Alaska_Web	* Any	* Any	Alaska_Web (Hiding Address)	= Original	= Original	* All	Automatic rule (see the network object data).

Each rule specifies what happens to the first packet of a connection. Reply packets travel in the opposite direction to the original packet, but are matched to the same rule.

The Address Translation Rule Base is divided into two sections, the Original Packet section, and the Translated Packet section. The Original Packet section specifies the conditions when the rule is applied. The Translated Packet section specifies the action taken when the rule is applied.

Each section in the Address Translation Rule Base Editor is divided into Source, Destination, and Service. The action is always the same:

- Translate Source under Original Packet, to Source under Translated Packet
- Translate Destination under Original Packet, to Destination under Translated Packet
- Translate Service under Original Packet, to Service under Translated Packet

Rule Match Order

Rule matching in the Address Translation Rule Base follows the same principle as in the Security Rule Base (see "The Security Rule Base" on page 9). When FireWall-1 receives a packet belonging to a connection, it compares it against the first rule in the Rule Base, then the second, then the third, and so on. When it finds a rule that matches, it stops checking and applies that rule.

The exception to this is when two automatic rules can match a connection, and Bidirectional NAT is turned on.

Bidirectional NAT

Bidirectional NAT applies to automatic NAT rules in the Address Translation Rule Base, and allows two automatic NAT rules to match a connection. Without Bidirectional NAT, only one automatic NAT rule can match a connection.

When NAT is defined for a network object, an automatic NAT rule is generated which performs the required translation. If there are two such objects and one is the source of a connection and the other the destination, then without Bidirectional NAT, only one of these objects will be translated, because only one of the automatically generated NAT rules will be applied, and so a connection between the two objects will only be allowed in one direction. With Bidirectional NAT, both automatic NAT rules are applied, and both objects will be translated, so connections between the two objects will be allowed in both directions.

The detailed logic of Bidirectional NAT is as follows:

- If the first match on a connection is on a Manual NAT rule, no further checking of NAT Rule Base is done.
- If the first match on a connection is on an Automatic NAT rule, then the rest of the NAT Rule Base is checked, one rule at a time, to see if another Automatic NAT Rule matches the connection. If it does, both rules are matched, and no further checking is performed.

The operation of Bidirectional NAT can be tracked using the SmartView Tracker, using the fields `NAT Rule Number` and `NAT Additional Rule Number`. The "additional rule" is the rule that matches the automatic translation performed on the second object in Bidirectional NAT.

Understanding Automatically Generated Rules

NAT can be defined *automatically* via the network object (Node, Network or Address range). When you define NAT via the network object, rules are automatically added to the Address Translation Rule Base.

Hide NAT on a Node adds one rule to the Address Translation Rule Base. It specifies that the source address of the packet is translated for connections that originate in the Node in the internal network. This is called a *Source Hide Rule.*

Static NAT on a Node adds two rules to the Address Translation Rule Base. In addition to the Source Hide rule, another rule specifies that for connections that originate in the external network, the Destination address of the packet is translated. This is called a *Destination Static Rule.*

If NAT (Hide or Static) is performed on a Network or an address range, an extra rule is added. The extra rule specifies that communication within the network or address range is not translated, that is, a packet sent from one machine to another in the same network is not changed.

Example of Automatically Generated Rule — Hide NAT

For the scenario in Figure 3-2, “Hide NAT on An Address Range,” on page 45, automatically defined Hide NAT on the address range Node adds two rules to the NAT Rule Base, as shown in FIGURE 3-5.

FIGURE 3-4 Automatically Generated NAT Rules for Hide NAT on an Address Range

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON	COMMENT
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE		
1	Range_Hide	Range_Hide	Any	Original	Original	Original	All	Automatic rule (see the network object data).
2	Range_Hide	Any	Any	Range_Hide (Hiding Address)	Original	Original	All	Automatic rule (see the network object data).

Rule 1 says that for connections within the internal (unprotected) side of the firewall, no NAT takes place.

Rule 2 says that for connections initiated on the internal (protected) side of the firewall, the source address of the packets is translated to the public Hide NAT address.

In automatic Hide NAT rules, the translated address is known as the *Hiding Address*, and this is the address that is known and used on the unprotected side of the FireWall-1 Gateway. The “real” addresses are the private addresses that are used on the protected side of the FireWall-1 Gateway.

Example of Automatically Generated Rules — Static NAT

For the scenario in Figure 3-1, “Static NAT on an Address Range,” on page 44, automatically defined Static NAT on the Node adds two rules to the NAT Rule Base, as shown in FIGURE 3-5.

FIGURE 3-5 Automatically Generated NAT Rules for Static NAT on an Address Range

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON	COMMENT
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE		
1	Range_static	Range_static	Any	Original	Original	Original	All	Automatic rule (see the network object data).
2	Range_static	Any	Any	Range_static (Valid Addresses)	Original	Original	All	Automatic rule (see the network object data).
3	Any	Range_static (Valid Addresses)	Any	Original	Range_static	Original	All	Automatic rule (see the network object data).

Rule 1 says that for connections within the internal (unprotected) side of the firewall, no NAT takes place. A packet sent from one machine to another in the same network is not changed.

Rule 2 says that for packets originating on the internal (protected) side of the firewall, source addresses are translated to valid (public) static NAT addresses.

Rule 3 says that for packets originating on the external (unprotected) side of the firewall, valid (public) destination addresses are translated to static NAT addresses.

In automatic Static NAT rules, statically translated public addresses are called *Valid Addresses*, and these are the addresses that are known and used on the unprotected side of the FireWall-1 Gateway. The “real” addresses are the private addresses that are used on the protected side of the FireWall-1 Gateway.

Precedence In Automatic Rules

Automatic Rules are placed in the Address Translation Rule Base as follows:

1) Static NAT rules before Hide NAT rules.

2) NAT on a node before NAT on a network or an address range.

Port Translation

Port Translation allows multiple application servers in a hidden network to be accessed using the a single IP address, based on the requested service (destination port). This has the benefit of saving on scarce public IP addresses. An typical implementation could allow an FTP server (accessible via port 21), an SMTP server (port 25) and an HTTP server (port 80) to be accessed using a singe IP public address.

To use Port Translation you need to craft manual NAT rules. Port Translation rules are supported on FireWall-1 enforcement points of version NG FP3 and higher.

NAT and Anti-Spoofing

NAT is always performed after the anti-spoofing checks, and anti-spoofing checks are performed only on the source IP address of the packet. This means that irrespective of NAT, spoofing protection is configured on the interfaces of the FireWall-1 Gateway in the same way. Unlike in previous versions of FireWall-1, there no special issues regarding anti-spoofing configuration and NAT.

Routing Issues

Static Routes on the FireWall-1 Gateway

This section is intended only for administrators who have upgraded the SmartCenter Server, where in the pre-upgrade

- pre-NG version, automatic NAT for the server was performed on the server side, or in the
- pre-NG FP3 version, manual NAT for the server was performed on the server side.

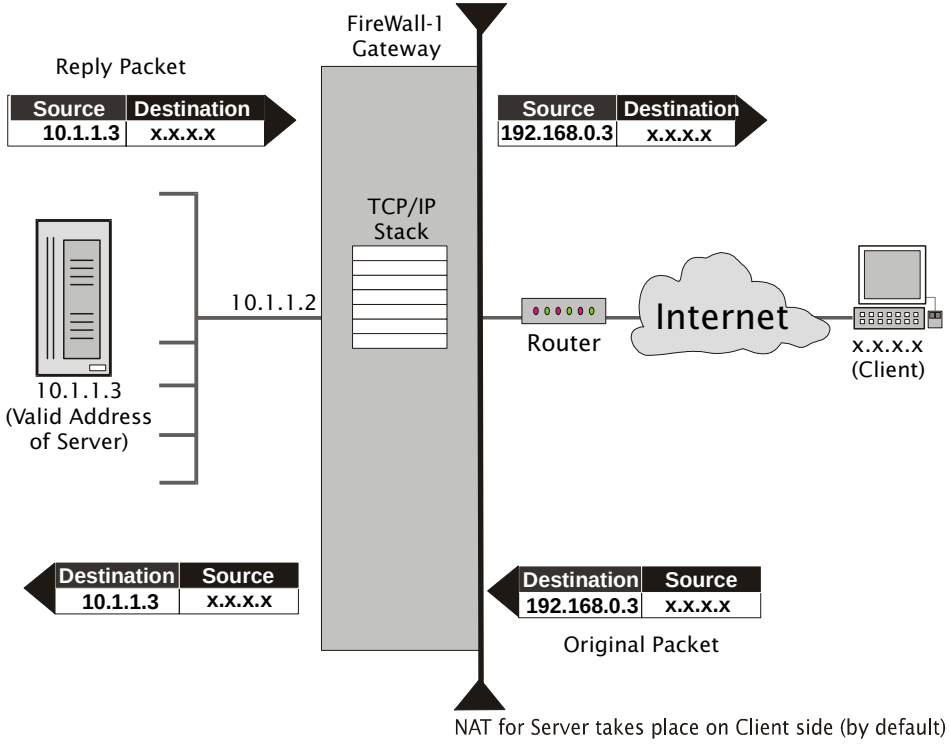
In a client-server connection across the FireWall-1 gateway, connections originate at the client, and the server sends reply packets back to the client.

In NG versions of FireWall-1, for both manual and automatic rules, NAT for the *server* is performed by default on the *client* side of the FireWall-1 gateway (FIGURE 3-6). This ensures that the Operating System routes the packets correctly.

In FIGURE 3-6, for the original packet, the firewall translates the destination address to the valid address of the server, and then the packet is routed to destination.

For reply packets, no NAT is performed on the destination, and the OS correctly routes the packet back to the client.

FIGURE 3-6 Illustrating NAT on Client side, which ensures that static routes are not needed



The NG default setting ensures reliable anti-spoofing and routing. It is recommended to stick to the default setting unless you have upgraded your SmartCenter Server from a pre-NG version, and you have FireWall-1 enforcement modules whose configuration requires other settings.

If NAT for the server destination is configured to be performed on the *server* side, the operating system receives the packet for routing before NAT is performed. The operating system therefore sees a valid address as the destination. will therefore route the packet back out to the Internet router rather than to the server.

To resolve this, configure Static Host Routes on the FireWall-1 gateway, so that it forwards packets to the correct interface. For example:

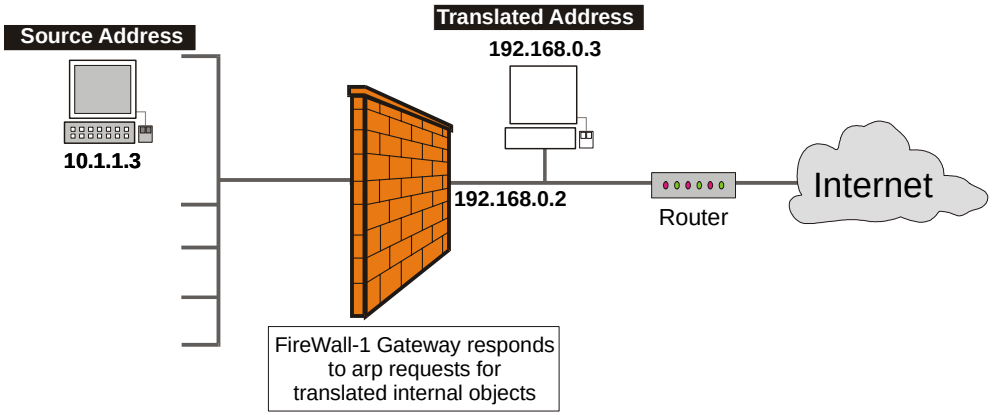
```
route add 192.168.0.3 10.1.1.2
```

Automatic and Proxy arp

Giving a machine in the internal network an external IP address using NAT makes that machine appear to the Internet to be on the external network, on the Internet side of the firewall.

When NAT is configured automatically, the firewall machine will reply on behalf of translated network objects to arp requests from the internet router for the address of the internal machine (FIGURE 3-7).

FIGURE 3-7 Illustrating Automatic Arp configuration



If using manual rules, you must configure proxy-arps. In other words, you must associate the translated IP address with the MAC address of the FireWall-1 gateway interface that is on the same network as the translated addresses.

IP Pool NAT

An IP Pool is a range of IP addresses (an Address Range, a network or a group of one of these objects) routable to the gateway.

IP Pool NAT ensures proper routing for two connection scenarios:

- SecuRemote/SecureClient to MEP (Multiple Entry Point) Gateways
- Gateway to MEP Gateways

When a connection is opened from a SecuRemote/SecureClient or a client behind a Gateway, to a server behind the MEP Gateways, the first packet is routed through one of the MEP Gateways. Return packets in the connection must be routed back through the same Gateway in order to maintain the connection. To ensure that this happens, each of the MEP Gateways maintains a pool of IP addresses that are routable to the Gateway itself. When a connection is opened to a server, the gateway substitutes an IP address from the IP Pool for the source IP address. Reply packets from the server return to the gateway, which restores the original source IP address and forwards the packets to the source.

The pool of IP addresses is defined as an IP range or collection of ranges in the NAT page of the Gateway object.

Disabling NAT in a VPN Tunnel

When communicating within a VPN, it is usually not necessary to perform NAT. It is possible to disable NAT in a VPN tunnel with a single click in the VPN community object. Disabling NAT in a VPN tunnel by defining a NAT rule will slow down the performance of the VPN.

Planning Considerations for NAT

In This Section

Hide Versus Static

For protocols where the port number cannot be changed, Hide NAT cannot be used.

When the external server must distinguish between clients based on their IP addresses, Hide NAT cannot be used, because all clients share the same IP address under Hide NAT.

To allow connections from the external network to the internal network, only Static NAT can be used.

Automatic Versus Manual Rules

Automatic NAT rules are easy to configure and so are less prone to configuration errors. Automatic ARP configuration is only effective for automatic rules.

Manually defining NAT Rules is complicated, but gives complete control over NAT. The following can only be done using Manual NAT Rules:

- Restricting rules to specified destination IP addresses, as well as to specified source IP addresses.
- Translating both source and destination IP addresses in the same packet.
- Performing Static NAT only in one direction
- Translating services (destination ports).
- Restricting rules to specified services (ports).
- Performing NAT on Dynamic objects.

Choosing the Hide Address in Hide NAT

The Hide Address is the address behind which the network, address range or node is hidden.

It is possible to either hide behind the interface of the Install on Gateway, or to hide behind a specified IP address.

Choosing a fixed public IP address is a good option if you wish to hide the address of the FireWall-1 Gateway. However, it means using an extra publicly routable IP address.

Choosing to hide behind the address of the Install On Gateway is a good option for administrative purposes. If the external IP address of the firewall changes, there is no need to change the NAT settings.

Configuring NAT

In This Section

General Steps for Configuring NAT

The steps for configuring NAT are always the same:

1. Determine the IP addresses to be used for translation.
2. Define Network Objects.
3. Define the Access Rules in the Security Rule Base. When defining Manual NAT rules, you must define network objects with translated addresses, whereas if using Automatic NAT Rules, you need define only one network object per real object. For example, if Static NAT is defined on an object called Alaska_Web, then the Security Rule Base need only refer to Alaska_Web (as in FIGURE 3-8), and there is no need to define a rule for Alaska_Web (Valid Address).

FIGURE 3-8 Example Security Rule Base Rule for an object with Automatic NAT

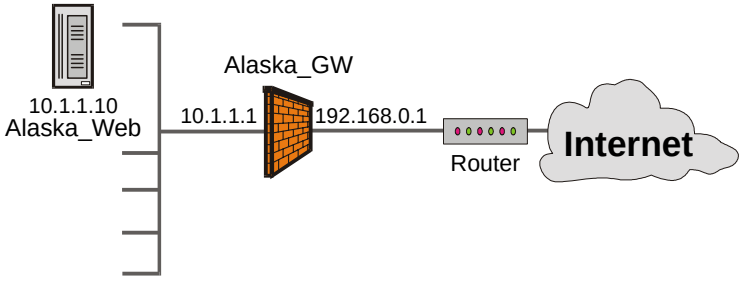
Source	Destination	Action
Any	Alaska_Web	Accept

4. Define NAT Rules (Automatic and/or Manual).
5. Install the Security Policy.

Basic Configuration - Network Node with Hide NAT

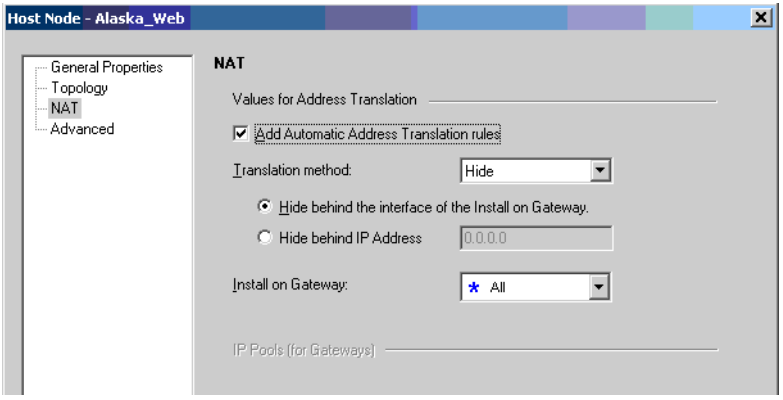
The following example shows how to set up basic Hide NAT for the configuration in FIGURE 3-9. The aim is to hide the IP address of the Alaska_Web web server (10.1.1.10) from connections that originate on the Internet. Alaska_GW has three interfaces, one of which faces the network on which Alaska_Web resides.

FIGURE 3-9 Example Network Showing Network Node with Hide NAT



1 Edit the Node object for Alaska_Web, and in the NAT page, select **Add Automatic Address Translation rules** (FIGURE 3-10).

FIGURE 3-10 Hide NAT configuration for a Node- NAT page



2 Select **Translation Method Hide**, and the option **Hide behind the interface of the Install on Gateway.**

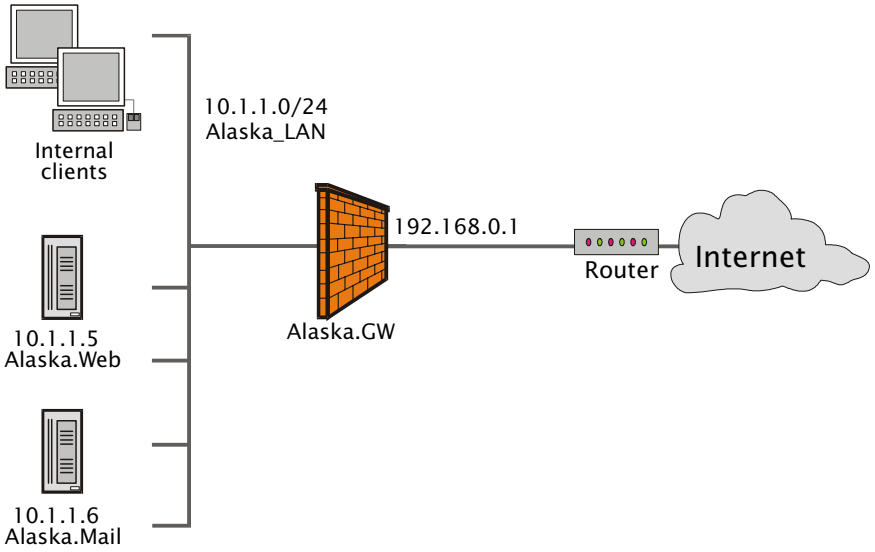
3 Select the **Install on Gateway.** The NAT Gateway in this example is Alaska_GW, so you can select either **Alaska_GW** or **All**.

Packets originating in Alaska_Web with the Internet as their destination will have their source address translated from 10.1.1.10 to 192.168.0.1. Packets originating in Alaska_Web with the DMZ as their destination will have their source address translated from 1.1.1.10 to 172.16.0.1.

Sample Configuration - Static and Hide NAT

The goal is make the SMTP server and the HTTP server on the internal network available to the Internet using public addresses, and provide Internet access for all users on the internal network.

FIGURE 3-11 Sample Configuration - illustrating Static and Hide NAT



The web and mail servers require static translation because incoming connections will be made to them from the Internet. Two routable addresses are available. 192.168.0.5 will be used for the Alaska.Web HTTP server, and 192.168.0.6 for the Alaska.Mail SMTP server.

The internal clients require hide translation because they will initiate connections. No incoming connections are allowed to them from the Internet. They will hide behind the external interface of the FireWall-1 Gateway.

1. Define network objects for Alsaka.Web (10.1.1.5), Alaska.Mail (10.1.1.6), Alaska_LAN (10.1.1.0 with Net Mask 255.255.255.0), and the FireWall-1 gateway (Alaska.GW).

2. Edit the Alaska.Web object, and in the **NAT** page check **Add Automatic Address Translation Rules**, select **Translation Method** *Static*, and define the **Translate to IP Address** as 192.168.0.5.

3. Similarly for Alaska.Mail, select **Translation Method** *Static*, and define **Translate to IP Address** as 192.168.0.6.

4. Edit the Alaska_LAN object, and in the **NAT** page select **Translation Method** *Hide*, and select **Hide behind the interface of the Install On Gateway**. The effective Hide address for the internal clients on Alaska_LAN is therefore 192.168.0.1.

The resulting Address Translation Rule Base is shown in FIGURE 3-12.

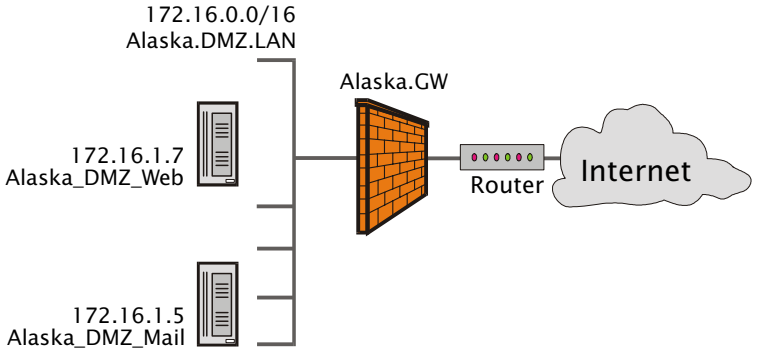
FIGURE 3-12 Automatic Address Translation Rule Base for Static and Hide NAT

NO.	ORIGINAL PACKET			TRANSLATED PACKET			INSTALL ON
	SOURCE	DESTINATION	SERVICE	SOURCE	DESTINATION	SERVICE	
1	Alaska.Mail	* Any	* Any	Alaska.Mail (Valid Addres	= Original	= Origina	* All
2	* Any	Alaska.Mail (Valid Address	* Any	= Original	Alaska.Mail	= Origina	* All
3	Alaska.Web	* Any	* Any	Alaska.Web (Valid Addre	= Original	= Origina	* All
4	* Any	Alaska.Web (Valid Address	* Any	= Original	Alaska.Web	= Origina	* All
5	Alaska_LAN	Alaska_LAN	* Any	= Original	= Original	= Origina	* All
6	Alaska_LAN	* Any	* Any	Alaska_LAN (Hiding Addr	= Original	= Origina	* All

Sample Configuration - Using Manual Rules for Port Translation

The goal is to make an both a web server and a mail server in a DMZ network available from the Internet using a single IP address. All addresses in the DMZ are to be hidden

FIGURE 3-13 Sample Configuration - illustrating Port Translation using Manual NAT

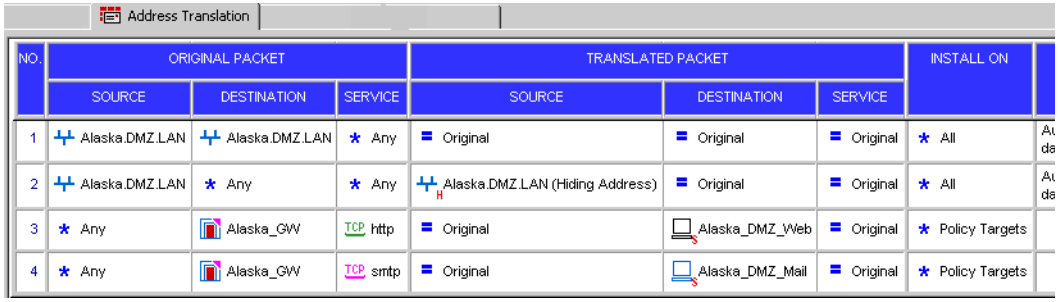


1. Define network objects for the network Alaska.DMZ.LAN (172.16.0.0 with Net Mask 255.255.0.0), the web server Alaska_DMZ_Web (172.16.1.7), and the Mail server Alaska_DMZ_Mail (172.16.1.5), and the FireWall-1 gateway (Alaska.GW).

2. On the Alaska.DMZ.LAN network object, in the **NAT** tab, select **Add Automatic Address Translation Rules**, and **Translation Method** *Hide*, and select **Hide behind the interface of the Install on Gateway**. This adds two automatic rules to the Address Translation Rule Base (Rules 1 and 2 in FIGURE 3-14).

3. In the Address Translation Rule Base, define a Manual NAT Rule that translates requests for the HTTP service to the Web server (Rule 3 in FIGURE 3-14), and a Manual NAT Rule to translate SMTP requests to the SMTP server (Rule 4 in FIGURE 3-14).

FIGURE 3-14 Address Translation Rule Base for Port Mapping



CHAPTER 4

Content Security

In This Chapter

The Need for Content Security

Blocking undesirable content is an important part of a corporate security policy. For example:

- Malicious code that can bring down entire networks is carried by computer viruses, Trojans and ActiveX components.
- Viewing undesirable Web content wastes time and resources.
- Peer-to-peer applications such as instant messaging and file sharing applications are designed to bypass firewalls and virus checking and avoid logging and tracking. They allow uncontrolled file transfers, and give Internet users unrestricted use of corporate storage, bandwidth and CPU resources.

Securing a network based solely on a connection's source, destination and port provides only very limited security. Undesirable content may be carried inside the packet by the application layer protocols. It is important to inspect HTTP, SMTP and FTP and other application layer protocols in order to block the undesirable content.

FireWall-1 Solution for Content Security

In This Section

Introduction to FireWall-1 Content Security

The most basic function performed by FireWall-1 is to decide whether or not to allow a connection. These Access Control checks are performed by the FireWall-1 Inspection Module (also known as the FireWall-1 *kernel*), usually only on the first packet in the connection. These checks are based on inspection of the IP headers (Network layer) and TCP Headers (Transport layer) in the packets. However, content security, which requires a deep examination of the data in the connection, also requires inspection of *Application layer* traffic.

FireWall-1 enforces Application layer security both in the FireWall-1 kernel, and using *Security Servers*. Many of the application layer security capabilities are enforced in the kernel, and all of them can be enforced using Security Servers.

FireWall-1 can also provide content security by integrating with third-party OPSEC certified content security applications.

Kernel inspection

Enforcement decisions are made in the kernel in two ways. For HTTP, the application layer data in a TCP connection stream is inspected, which allows correct inspection even where TCP data is re-ordered, segmented and re-transmitted. For other protocols, including SMTP and FTP, the FireWall-1 kernel makes application layer checks on individual packets.

Security Servers

Security Servers are Check Point processes that are integrated into FireWall-1. They are user mode processes that perform Content Security for HTTP, FTP, and SMTP. There is also a generic TCP Security Server. Security Servers employ many ways of enforcing content security including, for example, checking whether the connections for these protocols are well formed, stripping script tags for HTTP, email address translation for SMTP and file name matching for FTP.

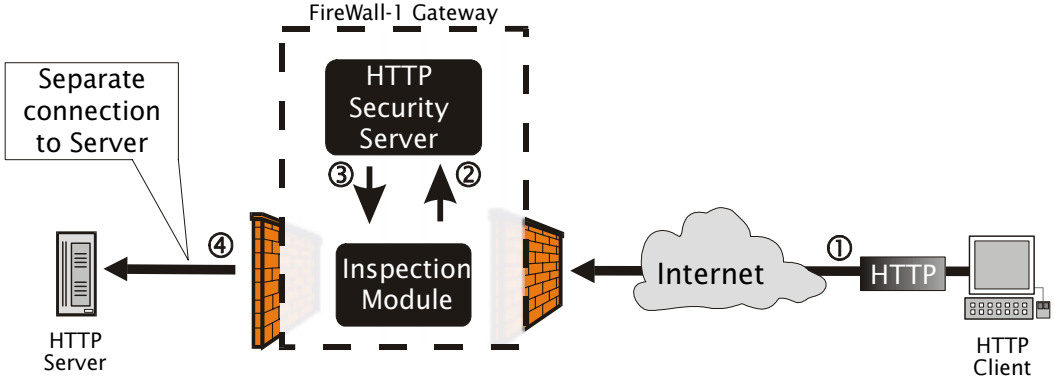
As well as Content Security, Security Servers also perform Authentication. The Authentication functions of the Security Servers are covered in “Authentication” on page 44.

How a Security Server Mediates a Connection

FIGURE 4-1 shows how the Security Servers mediate a connection. The HTTP Security Server is used as an example, but the method is the same for all Security Servers.

When a packet is matched to a Rule which contains a Resource, the Inspection Module diverts (also called “folds”) a connection to a Security Server. The Security Server performs the content security checks and return the connection to the Inspection Module, which opens a second connection that is sent on the destination HTTP Server.

FIGURE 4-1 How the Security Server mediates a connection



The source IP address that appears to the destination server is the IP address of the client that originally opened the connection. The connection leaves the security server with the source IP address of the FireWall-1 enforcement module, and the outbound kernel performs NAT so that the source IP address is that of the original client.

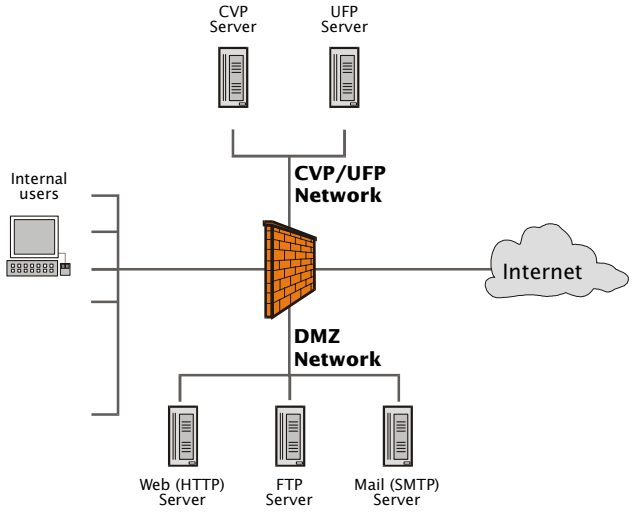
OPSEC Certified Content Security Products

FireWall-1 integrates with a wide range of third-party OPSEC certified content security applications that provide many additional security capabilities. This enables organizations to choose the content screening applications that best meet their needs, while managing content security centrally, as part of the security policy. Three widely used capabilities are virus protection (using CVP servers), URL filtering (using UFP servers), and protection against malicious code.

Deploying OPSEC Servers

OPSEC solutions, such as CVP and UFP servers can be deployed on the FireWall-1 Gateway, or on dedicated servers (FIGURE 4-2). These servers are typically either placed in the DMZ, or on a private network segment. This allows fast, secure connections between the CVP servers and the FireWall-1 Gateway.

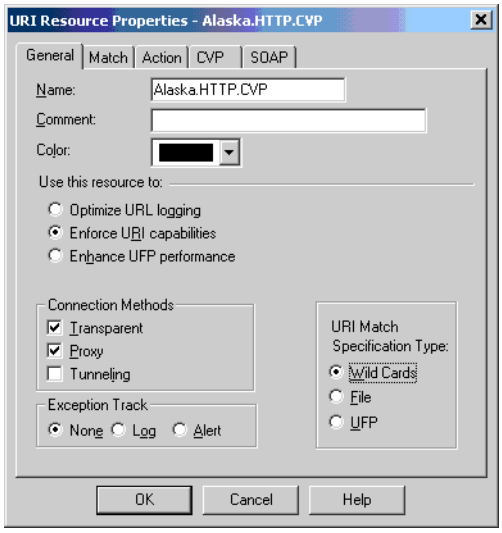
FIGURE 4-2 OPSEC Server Integration with FireWall-1



Resources: What They Are and How to Use Them

To perform content security, a Rule Base object called a *Resource* is defined in SmartDashboard (FIGURE 4-3). Resources are used to match a specific kind of application layer content, in other words, to specify what content you are looking for, and to perform some action on the content.

FIGURE 4-3 A URI Resource, showing the General tab



Using a Resource turns on either kernel inspection or the security servers, depending on what the resource is used for.

For instance, a rule can be created that will drop the connection and generate an alert if there are GETs or PUTs in an FTP transfer or if a specifically named file is part of the transfer. Another rule can drop email addresses or attachments while allowing the rest of the content through.

To specify the content you are looking for, regular expressions and wildcards can be used in the Resource.

The Resource is triggered when a Rule includes the Resource, and a packet matching that rule is encountered. A Resource is applied per Service. If a connection matches the source and destination of the rule and the match parameters of the Resource, then both the Action in the Rule and the Action in the Resource are applied.

Web Security

In This Section

Introduction to Web Security

Web-based traffic is a large and important portion of the traffic to be secured as it passes through the FireWall-1 enforcement point.

FireWall-1 has a number of Web server and Web surfer protection features. This section includes the following:

- "Understanding HTTP Sessions" on page 66, and "What is a URI Resource?" on page 67 give important background information for understanding Web Security in FireWall-1.

When considering how to approach Web security, it is convenient to divide the task into protecting the organization's *Web servers*, on the one hand, and protecting the *Web surfers*, or Web clients on the other. The sections that follow describe the categories of Web Security:

- "Granular URL Security" on page 68 describes URL-base protection that applies equally to both Web servers and Web surfers.
- "Protecting Web Servers Applications" on page 69 describes protection of Web servers that involves inspection of the application protocols, such as XML, HTML and SOAP, in addition to the HTTP protocol level inspection. These defenses protect Web server applications even from attacks which use valid HTTP.
- "Protecting Web Servers at the HTTP Layer" on page 73. describes HTTP protocol level inspection of Web servers.
- "Protecting Web Surfers" on page 75 describes how to protect Web surfers from external threats.
- "Limiting Web Surfers" on page 77 describe how to prevent the misuse of internal resources, by means of URL Filtering and by blocking peer-to-peer applications.

The final section: "Balancing Connectivity Versus Security" on page 82 describes properties that can be changed to provide either greater security, or greater connectivity.

Understanding HTTP Sessions

The following is an example of an HTTP session. It is useful for understanding Web security.

An HTTP session is made up of an HTTP request and an HTTP response. In other words:

HTTP Session = HTTP Request + HTTP Response

Both the HTTP request and the HTTP response have a header section and a body section

HTTP Request Example

Header section

The URL is marked in **bold** for clarity.

```
GET http://www.site.com/path/file.html?param1=val1&param2=value2 HTTP/1.1
Host: www.site.com
Range: 1000-2000
Cookie: cookiename=A172653987651987361BDEF
```

Body section

```
<Some content (usually a filled form which will be submitted)>
```

HTTP Response Example

Header section

```
HTTP 200 OK
Content-Encoding: gzip
Content-Type: text/html
Transfer-encoding: chunked
Content-Disposition: http://alternative.url.com
```

Body section

```
<Some content (usually an HTML page or a binary file)>
```

What is a URI Resource?

Many Web security features are implemented using a SmartDashboard object called a URI Resource. Resource objects are explained in “Resources: What They Are and How to Use Them” on page 64.

URI stands for Uniform Resource Identifier. A URI is more-or less identical to the familiar URL (Uniform Resource Locator).

Granular URL Security

In This Section

The FireWall-1 kernel and the HTTP Security Server provide the following URL-based protection that applies equally to both Web servers and Web surfers.

URLs and methods

Blocking specific attacks and performing access control can be based on URLs and methods. The URI Resource provides a powerful mechanism for blocking URLs. It supports regular expressions and a very fine granularity that utilizes the power of the FireWall-1 Rule Base.

All URL based attacks such as Code Red and Nimda can be blocked using a URI resource in SmartDashboard. The URI can be broken into filterable components using the **Host**, **Path** and **Query** parameters that are specified in the **Match** tab.

For configuration details, see “Blocking URL-based Attacks using a URI Resource” on page 101.

Basic URL Filtering

Basic URL Filtering capability is integrated into FireWall-1. Use this capability to restrict user access to many URLs, without having to define a separate resource for each one. Use it for restricting less than 50 URLs. This is because the list of banned sites must be defined in a file, and then manually edited and maintained, which makes it difficult to maintain a large list of banned sites.

For configuration details, see “Basic URL Filtering” on page 98.

More comprehensive URL Filtering is available using third party OPSEC-certified applications (see “URL Filtering and UFP” on page 77).

URL Logging

Normally, a logged connection shows the source or destination Web server and domain (for example http://foo.bar.com).

It is possible to generate extra URL logging information by performing kernel inspection on the HTTP connection, rather than using a URI Resource, which gives a less detailed log. This shows in the log the full path and query in the requested URL,

not just the name of the Web server (e.g. http://foo.bar.com/products/servlet/Satellite?pagename=1234). Do this by defining a URI resource and selecting **Optimize URL Logging**.

For configuration details about logging URLs, either by performing kernel inspection on the HTTP connection, or using a URI Resource, see "URL Logging" on page 97.

Protecting Web Servers Applications

In This Section

This section describes how FireWall-1 protects Web servers by inspecting application protocols such as HTML, in addition to the HTTP protocol level inspection. These defenses protect Web server applications even from attacks which use valid HTTP.

Securing Web Servers via the Network Object

FireWall-1 NG with Application Intelligence introduced a new way of defining security for Web servers. It is possible to define attack defense settings per Web server object. This approach is an easier and more intuitive way of defining content security, than using a resource and placing the resource in a rule in the Security Rule Base. An additional advantage of this approach is that when configuring content security for the Web server, the Web server is protected, irrespective of any rules in the Security Rule Base. Access control, which restricts the allowed source and destination of the HTTP connection, is independently defined in the Rule Base.

Currently, Web servers can be protected from cross-site scripting attacks via the Web server object.

For configuration details, see "Configuring Web Server Security via the Network Object" on page 104

Cross-Site Scripting Attack

Understanding the Cross-Site Scripting Attacks

Cross-site scripting attacks place malicious code in locations where other users see it.

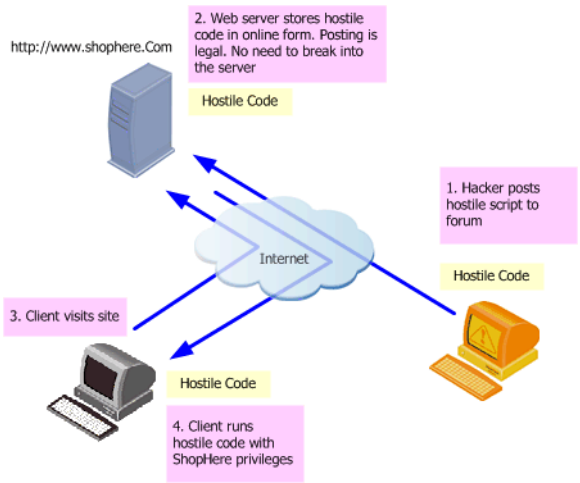
One kind of cross site scripting attack is intended to steal cookies that contain user identities and credentials. A hacker may want to steal cookies in order to impersonate another user. The client sends cookies to the Web site with every HTTP Request.

Many Web sites contain forms, which are used to post information such as names and addresses, or comments on bulletin boards. The hacker can inject scripting code into the attacked Web server by adding scripting code to these forms.

The Web server will send cookies only to the client that owns them. Since the hostile code runs in the client machine, the code can instruct the web browser on the client machine to send its cookies to another location, such as another Web site (hence the name: Cross Site Scripting), where the hacker can see the cookies.

This attack is illustrated in FIGURE 4-4.

FIGURE 4-4 Stealing cookies using the Cross site scripting attack



This attack is especially dangerous because neither the user nor the Web site administrator knows that the attack is taking place, unless they analyze the posted source code.

A second variety of Cross-site scripting attack does not steal cookies, but rather dupes the victim into supplying his or her credentials to the attacker. For example, an online auction site may allow users to post advertisement that will say something like:

> **For sale: Rolls Royce, 1995, Only $20,000, As new.**

A hacker can post an advert that contains scripting code, for example:

> **For sale: Rolls Royce, 1995,** `<script>alert(“Connection broken. Please re-enter username and password”)</script>` **Only $20,000, As new.**

When a user views the advert, the script causes a popup form to appear that asks the user to supply his or her details. The user viewing the second advert assumes the pop-up is from the auction site, and fills in his username and password. The script sends those details to the attacker.

Defending Against Cross-Site Scripting Attacks

Vulnerable Web sites can be configured to reject scripting code, but this can be time consuming and expensive.

To protect against Cross-Site Scripting attacks, FireWall-1 rejects HTTP requests that contain scripting code in the URL or the body. The scripting code is not stripped from the request, but rather the whole request is rejected.

FireWall-1 gives the administrator three ways of rejecting of scripting code.

An least strict approach is to reject any occurrence of a request that contains one of the following list of banned tags (the search is case insensitive). The list cannot be edited:

.cookie	GetFolder	onDragDrop	onMove
ActiveXobject	GetParentFolder	ONfocus	onResize
applet	GetSpecialFolder	onkeyDown	ONselect
CopyFile	javascript	onKeyPress	onsUbmit
copyparentfolder	livescript	onKeyUp	ONunload
CreateObject	mocha	ONload	OpenAsTextStream
CreateTextRange	object	onMouseDown	Opentextfile
DeleteFile	ONabort	onMouseMove	RegWrite
DriveType	ONBlur	ONmouseout	Replace
FileExist	ONchange	onmouseOver	script
GetFile	onClick	onMouseUp	vbscript

The more cautious approach is to reject all HTTP requests that contain the < or > characters. The disadvantage of this approach is that it can block access to pages that contain innocent tags, such as <Title>.

The strictest and safest approach rejects everything that is rejected by the less strict approaches. In addition, ampersand encoding of the characters "<" (that is, &lt, and &#60) and ">" (&gt and &#62) is also blocked. Ampersand (&) encoding is used to encode special ASCII characters in HTML.

For configuration details, see "Protection Against Cross-Site Scripting Attacks" on page 105.

Malicious activity Prevention - General HTTP Worm Catcher

A worm is self-replicating malware (malicious software) that propagates by actively sending itself to new machines. Many worms propagate by using security vulnerabilities in HTTP servers or clients.

SmartDefense allows administrators to configure worm signatures that will be detected and blocked by VPN-1/FireWall-1. This detection takes place at very high speed in the kernel and does not require a security server. By default, a number of patterns are defined, which protect against worms such as Nimda, CodeRed, and 'htr overflow'.

Important Capabilities include:

- **Encoding support:** Malicious content that is encoded in various formats can traverse some firewalls. In contrast, VPN-1/FireWall-1 fully decodes content for inspection and security enforcement. SmartDefense can identify and block URLs and worms that contain Hex, UTF-16 and UTF-8 encoding as well as white spaces and mixtures of upper- and lowercase characters.
- **Fragmentation Reassembly:** Malicious content that is split across fragments can traverse some firewalls. In contrast, VPN-1/FireWall-1 collects and reassembles all the fragments of a given IP packet so that security checks can be run against the complete contents.
- **User-defined worm patterns:** The SmartDefense Subscription Service regularly updates signature patterns for common worms. In addition to the SmartDefense service, an administrator can define custom worm patterns in a simple way, using regular expressions.

Directory Traversal Attacks

Directory traversal attacks allow a hacker to access files and directories that should be out of his reach. In many attacks, this leads to running executable code on the Web server with one simple URL. Most of the attacks are based on the “..” notation within a file system.

FireWall-1 blocks requests in which the URL contains an illegal directory request. For example, http://www.server.com/first/second/../../.. is illegal because it goes deeper than the root directory. http://www.server.com/first/second/../ is legal because it is equivalent to http://www.server.com/first/. FireWall-1 supports the same capability for URLs that are encoded with Unicode and % encoding. This provides protection against attacks such as “IIS Unicode directory traversal (MS00-086)” and “IIS double Decode (MS01-026)”.

The protection is enabled by default when using the HTTP Security Server.

To check for Unicode variations, set the property “http_web_encoding” to true using the Database Tool, available at http://www.checkpoint.com/techsupport/downloadsng/utilities.htm.

Securing XML Web Services (SOAP)

Email and Instant Messaging (using protocols such as SMTP and MIME) are for person-to-person communication, and Web pages (using HTML and DHTML) are for person-to-program communication. XML Web services (using XML Schema and SOAP) allow program-to-program communication, and represent an important new way of communicating using Internet protocols and standards.

The Simple Object Access Protocol (SOAP) provides a way for applications to communicate with each other over the Internet, independent of platform. SOAP relies on XML to define the format of the information and then adds the necessary HTTP headers to send it.

XML passes information using commands called *Methods* that are intended to run on the destination computer.

FireWall-1 uses a Security Server to prevent potential attacks by verifying that the HTTP, XML, SOAP methods in SOAP requests conform to the RFC. FireWall-1 also checks that only a predefined list of acceptable methods is being passed in the SOAP packet.

The way that FireWall-1 treats SOAP packets is defined in a URI resource that uses HTTP. The URI specifies whether FireWall-1 SOAP packet will always be Accepted, or only the Methods specified in a predefined file will be Accepted.

The SOAP processing defined in the URI resource is performed only if the HTTP connection carrying the SOAP message was already Accepted by the rule in which the URI resource is used. In other words, the connection must match the rule, and the rule Action cannot be Reject or Drop.

Protecting Web Servers at the HTTP Layer

In This Section

FireWall-1 protects Web servers using the following HTTP protocol level inspections. These capabilities ensure that the HTTP is used in a valid way.

Limiting the URL length

FireWall-1 limits the length of URL that is requested from the server. While in the HTTP RFC there is no restriction on the maximum length of the URL, it is good security practice to limit it. This reduces the chance for buffer overruns and limits the size of code that can be inserted using the overflow.

The "Netscape Enterprise Buffer Overflows" attack requires 4080 characters to create a buffer overrun. The default maximum for Security Servers is 2048. The "Apache Long Slash Directory Listing" attack requires 8000 characters to reveal directory structure.

The maximum allowed length is adjustable using SmartDefense.

ASCII Only Request Headers

FireWall-1 checks that the HTTP Method in the Request from the server is RFC compliant. It also blocks non-ASCII characters (32-127) in the HTTP request headers. Other than the fact that the HTTP RFC does not allow binary characters anywhere in the HTTP headers, blocking them is good security practice because executables and buffer overrun exploits usually need binary characters.

This protection provides protection against attacks such as the ISAPI PRINTER buffer overflow, which is injected in the Host header (and not in the URL itself), and uses binary characters.

The defense can be turned on or off using SmartDefense.

Blocking non-standard HTTP methods

The HTTP RFC allows a restricted set of HTTP methods (GET, PUT, HEAD, POST). By default, FireWall-1 blocks all other methods. Many of the non-standard methods have a very bad security record. It is possible to allow the WebDAV methods, but they are blocked by default. The administrator can further limit the allowed set of HTTP methods using a URI Resource. For example, the administrator can allow only the GET and PUT methods.

This protection provides protection against attacks such as the "Internal IP Addresses and IIS" attack (described at http://www.nextgenss.com/advisories/iisip.txt) that uses the non-standard PROPFIND method. The "Netscape Enterprise Buffer Overflows" attack uses the non-standard GETPROPERTIES method.

This defense is turned on by default. It can be turned off changing the property "enable_propfind_method" to true using SmartDashboard. In the **Global Properties**, in **SmartDashboard Customization > Advanced Configuration** page, under **Web Security**.

HTTP Headers Centered Security

FireWall-1 enforces the maximum total HTTP Header Length, the maximum number of HTTP headers (both of these are configurable via SmartDefense), that only allowed schemes appear in the URL (configurable via a URI Resource), and the session time-out.

Protecting Web Surfers

In This Section

This section describes how FireWall-1 protects Web surfers.

CVP and Anti-Virus Protection of Web Traffic

Computer virus detection is vital to enterprise security. FireWall-1 makes it possible to perform virus scanning at the network access point, which is both safer and more efficient than performing Virus scanning on the desktop or on the application servers.

A Content Vectoring Protocol (CVP) server examines and reports on the contents of both incoming and outgoing files. For example, whether a file contains a virus, or malicious JAVA or ActiveX applets. It can also change the contents of the file.

To perform virus scanning, the HTTP security server transfers packets from the FireWall-1 Gateway to another server running an OPSEC Certified virus scanner. This method uses the Content Vectoring Protocol (CVP) to transfer packets to and from an OPSEC virus scanning server.

The virus scanning CVP server determines if there is a virus. If it finds a virus it can either

- return the file to the FireWall-1 Gateway with the offending content removed (if the CVP server is allowed to modify content), or
- drop the file (if the CVP server is not allowed to modify content).

CVP uses TCP port 18181, by default.

The following discussion describes how a connection is handled by the HTTP Security Server in order to perform CVP checking. The FireWall-1 Gateway that runs the HTTP Security Server acts as a proxy, and so is not an active participant in the connection.

The connection request/response process without a CVP server is:

1. HTTP client to HTTP server (request)
2. HTTP server to HTTP client (response)

The data which needs to be checked is carried in the response which comes from the Web server, so when a CVP server is used the response is always checked. In that case, the connection request/response process is:

1. HTTP client to HTTP server (request)
2. HTTP server to CVP server (response)
3. CVP server to HTTP client (response)

Normally, only the HTTP *responses*, which come from the Web server, are sent to the CVP Server for checking. However, you also may wish to protect against undesirable content in the HTTP *request*, for example, when inspecting peer-to-peer connections. In this case, the connection request/response process is

1. HTTP client to CVP server (request)
2. CVP server to HTTP server (request)
3. HTTP server to CVP server (response)
4. CVP server to HTTP client (response)

The HTTP Security Server can be configured to send HTTP *headers* to the CVP server, as well as the HTTP message data.

Improving CVP Performance for Web Traffic

HTTP Security Server performance can be significantly improved by ensuring that safe traffic is not sent to the CVP server. The aim is to reduce the traffic sent to the CVP server, and reduce the number of connections opened with the CVP server.

FireWall-1 considers safe, files such as pictures and video that are not executable, because they do not normally contain viruses.

The FireWall-1 Security Server identifies safe content by actually examining the contents of a file. It does not rely on examining the URL (for file extensions such as *.GIF) nor does it rely on checking the MIME type (such as image/gif) in the server response.

For configuration details, see “Improving the Performance of the CVP Server” on page 106.

ASCII Only Response Headers and HTTP RFC Compliance

FireWall-1 checks that the HTTP Response from the server is RFC compliant. It also blocks non-ASCII characters (32-127) in the HTTP response headers. Other than the fact that the HTTP RFC does not allow binary characters anywhere in the HTTP headers, blocking them is good security practice because executables and buffer overrun exploits usually need binary characters.

This protection provides protection against attacks such as the ISAPI PRINTER buffer overflow, which is injected in the Host header (and not in the URL itself), and uses binary characters.

The defense can be turned on or off using SmartDefense.

Java and ActiveX Security

FireWall-1 is able to control incoming JAVA and ActiveX code according to specific conditions, such as host, URL, or authenticated user name.

Capabilities of JAVA and ActiveX screening include the following:

- Stripping ActiveX tags from HTML pages
- Stripping JAVA applet tags from HTML pages
- Blocking JAVA attacks by blocking suspicious back connections

More comprehensive scanning of Java, ActiveX and other executables can be accomplished with content security applications from OPSEC certified vendors.

Limiting Web Surfers

In This Section

This section describes how the FireWall-1 protects Web surfers.

URL Filtering and UFP

The Security administrator can choose to prevent access to specific destinations on the Internet in order to allow access only to appropriate Web page information, and make it impossible to access particular Web sites, or download certain file types.

This is done with the help of third party, OPSEC certified URL management applications. The security administrator can define a corporate Security Policy that includes URL screening, in order to block undesirable Web pages and record the types of URLs accessed for internal analysis and reporting needs.

Integrated URL management applications are available from many OPSEC partners. For a listing of OPSEC Content Security solutions, see http://www.opsec.com/solutions/sec_content_security.html.

A URI Filtering Protocol (UFP) server is used to maintain a list of URLs and their categories. As a user requests a URL, FireWall-1 checks that URL against a UFP Server, which returns the category under which the URL falls. In SmartDashboard, permitted categories can be selected. Access to the Web page is allowed if the URL is in a permitted category.

UFP uses TCP port 18182 by default.

Note - A basic URL filtering capability that can be used to block a specific list of URLs without a UFP server is built into FireWall-1. For details, see "Basic URL Filtering" on page 68.

FireWall-1 can integrate with OPSEC certified solutions in two ways:

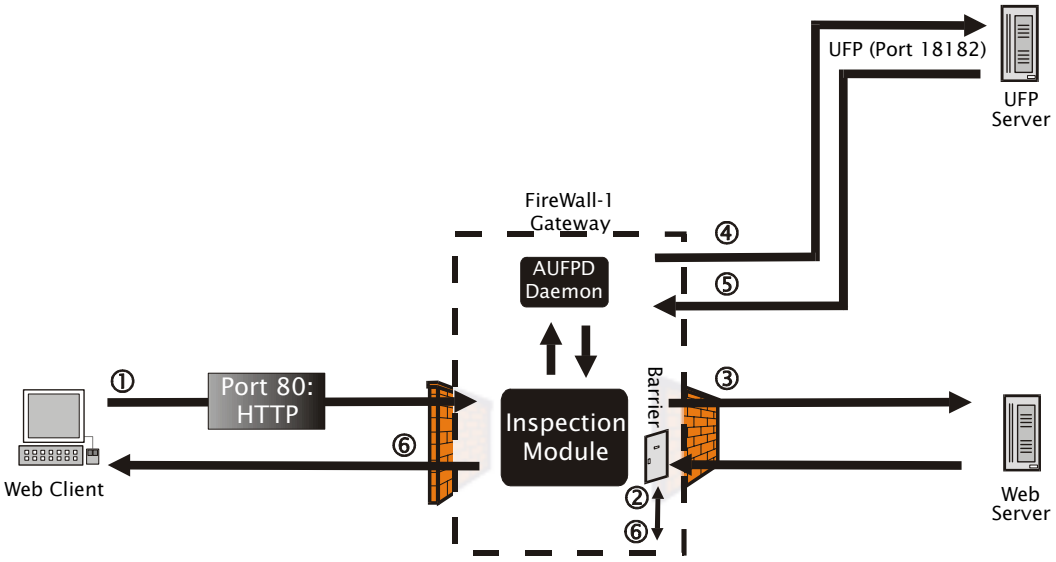
- Enhanced UFP Performance mode (called **Enhanced UFP Performance** in the URI Resource) uses FireWall-1 kernel inspection together with a dedicated UFP daemon (`aufpd`). UFP caching, CVP checking and authentication cannot be used with this mode, and certain HTTP Header verifications, such as methods and length are not performed.
- The regular UFP checking mode uses the FireWall-1 HTTP Security Server to mediate UFP connections. This can add significantly to the response time seen by clients that browse Web sites, in comparison to the Enhanced UFP Performance mode.

For configuration details, see "URL Filtering with a UFP Server" on page 98. An explanation follows that describes how these two modes work.

Enhanced UFP Performance Mode

FIGURE 4-5 illustrates how enhanced URL Filtering (UFP) performance of an HTTP connection works.

FIGURE 4-5 Enhanced URL Filtering (UFP) process, using kernel inspection

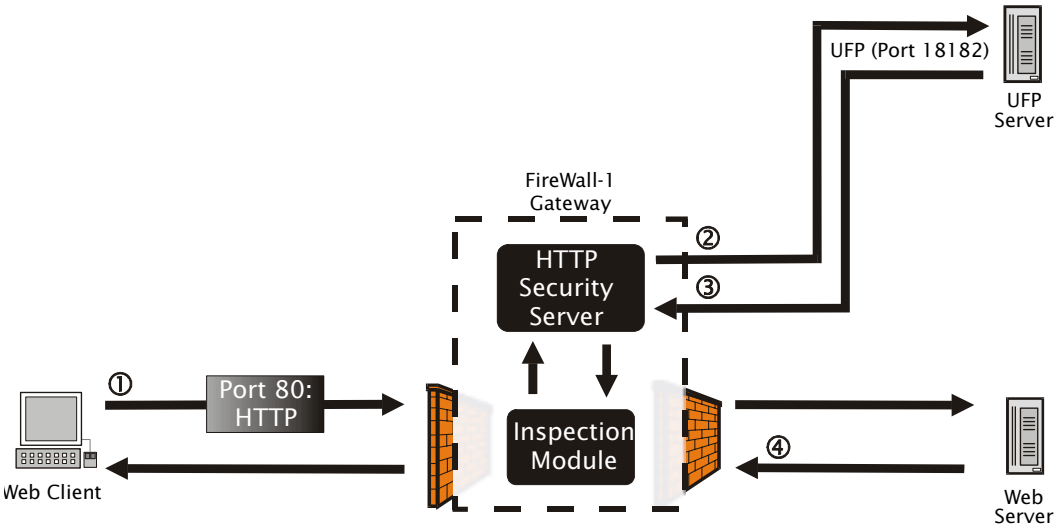


1. Web client invokes a connection through the FireWall-1 Inspection Module.
2. The kernel Inspection Module puts up a barrier that prevents returning TCP packets from sending their data on to the Web client, but does return ACK packets to the client. This prevents the client retransmitting the HTTP request, which it would otherwise do, thinking that there is network congestion.
3. HTTP requests destined for the Web Server go through FireWall-1 uninterrupted.
4. At the same time as step 3, the Inspection Module extracts the URL, and the AUFPD daemon establishes a UFP session with the UFP server to categorize the URL.
5. Based on the Validation Result message, AUFPD tells the Inspection Module whether or not to block the URL.
6. If the URL is permitted, the barrier is removed, and the HTTP response from the Web Server is allowed through FireWall-1.
7. If the URL is blocked, the HTTP response is blocked, and the kernel Inspection Module sends TCP reset packets to the Web client and server to stop them from retransmitting data.

UFP Checking using the Security Server

FIGURE 4-6 illustrates how FireWall-1 performs URL Filtering of an HTTP connection using the HTTP Security and a UFP server.

FIGURE 4-6 URL Filtering (UFP) Process for an HTTP Connection



1. Client invokes a connection through the FireWall-1 Inspection Module.
2. The HTTP Security Server uses UFP to send the URL to be categorized to the third-party UFP Server.
3. The UFP Server inspects the file and returns a Validation Result message, notifying the security server of the result of the inspection.
4. Based on the Validation Result message, the Inspection Module either allows or disallows the viewing of that particular Web page.

Securing Peer-to-Peer applications

Peer-to-peer applications have gained immense popularly on the internet in the last few years. Peer-to-peer applications can be roughly divided into 2 major categories:

- Instant messengers — whose main goal is to enable direct online communication between people.
- File sharing networks — whose main goal is to share resources such as storage.

Instant messengers applications include ICQ, Yahoo messenger, MSN messenger and AOL messenger. As of 2003 the leading file sharing applications are Kazaa, Morpheus, Gnutella, Limewire, MusicCity and eDonkey.

Security problems with Peer-to-Peer applications

Peer-to-Peer applications...

- break the client-server model; therefore, everyone is both a client and a server and desktops become non-secure servers.

- are designed to bypass firewalls and virus checking and avoid logging and tracking, while allowing file transfers, chat, games, voice and mail. While they might start by using known protocols, they all have the ability to masquerade as “standard” HTTP traffic. Most have unique fingerprints that can be detected by FireWall-1.
- often have multiple security vulnerabilities. Some Peer-to-Peer application vendors have gone so far as to state: “This is not a secure application” during the installation.
- open direct tunnels. Anonymity, expected by NAT and proxies, is lost. You get a direct connection on a user basis. This is even more dangerous than email, because the connection is live.
- consume corporate resources — storage, bandwidth and CPU are made available for all Internet users.
- lead to employees spending time on non-productive activities.
- raise legal issues — copyrights issues and offensive use can become major problems.
- are commonly used to download illegal, possibly trojaned software.
- installation packages often include add-ons and extra software (Spyware) that is known to leak information to the net. Spyware installed has severe security implications and tends to be very hard to remove from a computer.

Blocking Peer-to-Peer applications

FireWall-1 blocks Peer-to-Peer traffic by checking the HTTP Header request and response patterns using the HTTP Security Server, or by using the kernel, which checks the HTTP Header request.

For configuration details, see “Blocking Peer-to-Peer applications and HTTP Header Based Attacks” on page 102).

Blocking Peer-to-Peer Applications By Port

Kazaa uses port 1214 by default and is defined in SmartDashboard list of services. Gnutella has default port 6346 and can be blocked like any other service. MSN Messenger uses port 1863. These are merely default ports which can be changed, so it is better to block all unknown ports and open the ones you need.

Blocking HTTP Headers using Regular Expressions

A Web server or application parses not only the URL, but also the rest of the HTTP header data. Wrong parsing can lead to buffer overrun attacks and other vulnerabilities. Such attacks, while RFC compliant, can be blocked using signatures that are defined using regular expressions. The signatures are not limited to URL fields, but can be applied to any field and value.

This protection provides protection against attacks such as the “O'Reilly Website GET Buffer Overflow Vulnerability (CAN-2000-0623)” that can use the Referrer field in order to create a buffer overrun.

For configuration details, see “Blocking Peer-to-Peer applications and HTTP Header Based Attacks” on page 102).

Balancing Connectivity Versus Security

It is possible to tune FireWall-1 to provide greater Web security, at the expense of Web connectivity, or vice versa.

Security versus connectivity tuning can be done using a number of FireWall-1 database properties, as described in this section.

Allowing or Restricting Content

enable_propfind_method

To allow users access to popular applications such as Microsoft Hotmail, Outlook Web Access, and FrontPage, the following non-RFC compliant WebDAV HTTP methods must be allowed

BCOPY, BDELETE, BMOVE, BPROPFIND, BPROPPATCH, COPY, DELETE, LOCK, MKCOL, MOVE, NOTIFY, POLL, PROPFIND, PROPPATCH, SEARCH, SUBSCRIBE, UNLOCK, UNSUBSCRIBE.

WebDAV has certain security issues, as discussed in the SmartDefense Advisories page:

http://www.checkpoint.com/securitycenter/advisories/index.html

The `enable_propfind_method` property controls whether or not these non-RFC compliant WebDAV HTTP methods are allowed. By default, they are not allowed, and the property is false. When the flag is false, only the following methods are allowed:

OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT.

To change the value of a property, in SmartDashboard, edit the **Global Properties** in the **SmartDashboard Customization > Advanced Configuration** page, under Web Security.

http_allow_content_disposition

The Content-Disposition header in the HTTP Response header suggests to the client a location where the client should save content (such as a file) carried in the HTTP response. This location can potentially point to a crucial OS file on the client. Some clients may take up this suggestion without question and save the content to that location.

The `http_allow_content_disposition` property controls whether or not the HTTP response is allowed to include the Content-Disposition header. By default, this header is not allowed, and the property is false.

To change the value of a property, in SmartDashboard, edit the **Global Properties** in the **SmartDashboard Customization > Advanced Configuration** page, under Web Security.

http_allow_ranges

Partial range requests allow the content in an HTTP response to be split over more than one response. However, content security checks are only completely effective if the responses are not split in this way. FireWall-1 does not allow range requests by default. Set `http_allow_ranges` to true to allow range requests in HTTP requests, and range responses in HTTP responses.

Adobe Acrobat® uses HTTP ranges to allow pages of Acrobat PDF files to be viewed as soon as they are downloaded. Not allowing ranges means that the whole file must be downloaded before it can be viewed. Some download managers also use HTTP ranges.

To change the value of this property, in SmartDashboard, edit the **Global Properties** in the **SmartDashboard Customization > Advanced Configuration** page, under Web Security.

Content Compression

http_disable_content_enc and http_disable_content_type

Compressing content in HTTP responses is a way of increasing the speed of the connection. However, content security checks such as HTML weeding and CVP checking cannot be performed on compressed content.

The Content-Encoding and Content-Type headers in the HTTP response indicate whether or not the content is compressed, for example: Content-Encoding: gzip, Content-Type: application/gzip.

The `http_disable_content_enc` and `http_disable_content_type` properties control whether or not to allow data in the HTTP response to be compressed. If these properties are false (the default value), compression of content in an HTTP response is not allowed. Both these properties can be either true or false. One may be true when the other is false. Each one affects it own header.

These properties only affect content on which one or more of the following content security checks are performed: HTML weeding, blocking Java code, CVP, SOAP.

To change the value of this property, in SmartDashboard, edit the **Global Properties** in the **SmartDashboard Customization > Advanced Configuration** page, under Web Security.

HTTP Format sizes

The following parameters allow you to configure upper bounds to HTTP headers. If these bounds are exceeded, it becomes easier to include some malicious code or cause a buffer overflow using the headers. Connections that try to exceed this limit are rejected. Default values are indicated in **bold**. These parameters are controlled via SmartDefense, in **Application Intelligence > Web > HTTP Protocol Inspection**.

The parameters are:

- Maximum URL length (**2048**) bytes — HTTP Request headers contain a URL, such as the one in the request line: GET <URL> HTTP/1.1. The URL can include parameters, and can be very long, as shown in the example (see "Understanding HTTP Sessions" on page 66). This parameter is equivalent to the property `http_max_request_url_length`.
- Maximum HTTP Header length (**2100**) bytes — An HTTP Request header may contain a cookie, and may be very long. The HTTP standard does not limit header length. However, excessive header length falls outside of normal or expected usage. Limiting headers of excessive length reduces the chance of buffer overflows and limits the size of code that can be inserted using the overflow. This parameter is equivalent to the property `http_max_header_length`.
- Maximum Number of HTTP Headers (**500**) — Having too many HTTP headers may constitute a security risk. The HTTP standard does not limit the number of headers. However, an excessive number of headers falls outside of normal or expected usage. This parameter is equivalent to the property `http_max_header_num`.Why?

The value 0 can be used to disable any of these protections. Note that on enforcement modules of version NG FP3 or earlier, a value of 0 for `http_max_request_url_length` means that all connections are blocked.

Mail Content Security using the SMTP Security Server

In This Section

Introduction to Mail (SMTP) Content Security

The SMTP protocol, used for sending email, and files attached to email messages, poses a challenge to the Security Administrator who wants to maintain connectivity, while protecting the internal network from damaging content.

The SMTP Security Server protects against malicious mail messages, provides SMTP protocol centred security, prevents attempts to bypass the Rule Base using mail relays, and prevents Denial of Service and spam mail attacks.

CVP checking can be performed on mail traffic. This is done by redirecting the mail traffic to a CVP server.

The SMTP Security Server is used by creating an SMTP Resource in SmartDashboard, and placing the Resource in a Rule. CVP checking is configured in the SMTP Resource object. In addition, many SMTP Security Server parameters can be configured per object.

Protection against Malicious Mail Messages

MIME attachments of the specified type are stripped from the message. For example, the “message/partial” MIME type is stripped to prevent fragmented and reassembled messages (RFC2046, section 5.2.2.1). This mime type can be used to bypass most of the security restrictions imposed on email messages, because the messages are cut into smaller segments, so that the malicious message cannot be detected by virus scanners or other content testing mechanisms

File attachments of the specified names are stripped from the message.

JAVA, JAVA Script, ActiveX code, FTP links, and port strings can be stripped from a mail message to prevent Denial of service and virus attacks

SMTP Protocol Centered Security

Strict compliance is enforced of various SMTP fields with RFC821 and RFC822 to prevent attacks based on buffer overruns.

Some unsafe SMTP commands, such as EXPN or VRFY are disallowed or disabled.

Trying to Bypass the Rule Base

Mail relay messages, that use the “%- hack” or “!” characters in the recipient fields are not accepted.

Preventing Denial of Service Attacks

The number of mail messages that are accepted in the spool can be limited. If the number of mail messages exceeds the designated limited, new mail messages will not be accepted.

The number of SMTP error commands is limited.

Mail messages larger than a configured limit are not allowed to pass.

The number of recipients of a mail message can be limited.

Spam Mails

The SMTP Security Server adds the "add received" header to the mail to prevent address spoofing.

Mail messages are not accepted if the sender/recipient does not contain a domain name.

Other SMTP Security Server Capabilities

CVP can be used to perform anti-virus checking on the mail, by configuring an SMTP Resource and using it in a Rule. In addition, the Security Administrator can configure an SMTP Resource to:

- hide outgoing mail's "From" address behind a standard generic address that conceals internal network structure and real internal users
- resolve the DNS address for mail recipients and their domain on outgoing connections — MX Resolving
- perform a mail-user based policy to:
  - perform different mail actions per recipient of a given mail
  - enable the generation of different mail contents on a per user basis
  - apply content security features at the user level

Mail scanning settings may need to be adjusted to meet demand. For example, mail may be coming in faster than it can be scanned, accumulating, and not being sent to the mail server. It is possible to adjust the performance of the SMTP Security Server per network object by adjusting the load generated by the mail dequeuer in two different ways:

- control the number of connections per site
- control the overall connections generated by the mail dequeuer

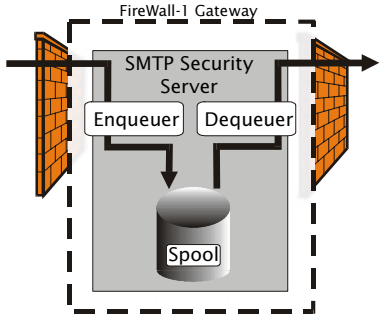
Secure Sendmail Application

The SMTP Security Server provides additional security over standard `sendmail` applications. Functionality is split between two separate modules, ensuring that there is no direct path connecting mail servers. In addition, the `sendmail` application is protected by the firewall, thus preventing direct online connections to it.

Also, a spool dequeuer mechanism provides efficient FIFO (First-In-First-Out) spool scanning, which gives priority to new mail over undeliverable old mail.

FIGURE 4-7 illustrates how one process, the enqueuer, writes incoming messages to a disk cache, and another process, the dequeuer, empties the cache.

FIGURE 4-7 SMTP Security Server



FTP Content Security

In This Section

Introduction to FTP Content Security

Content Security for FTP connections is provided both by the FireWall-1 kernel and the FTP Security Server.

CVP checking can be performed on FTP traffic. This is done by redirecting the FTP traffic to a CVP server. This is configured in the FTP Resource object.

FTP Enforcement by the FireWall-1 kernel

The FireWall-1 kernel enforces RFC compliant use of the PORT commands issued by the client, to ensure that no arbitrary syntax is sent. FireWall-1 enforces additional security limitations, including

- Proper use of the IP field in the PORT command, to protect against the Bounce Attack.
- Proper use of the Port in the PORT command. Data connections to well-known ports are not allowed.
- Unidirectional data flow on the data connections. This is a second line of defense to avoid using the data connection for non-FTP data that require bi-directional data flow.

FTP Enforcement by the FTP Security Server

Control the Allowed Protocol Commands

- Only a predefined list of FTP commands is allowed, which gives full control over the character of the FTP traffic.
- Some seldom-used FTP commands may compromise FTP application security and integrity, and so are blocked. These include the commands SITE, REST, MACB, and mail commands such as MAIL and MSND.
- The SITE command is enabled once, upon login, to allow common FTP applications to work properly.
- FireWall-1 enables control over the desired mode of FTP traffic, both for passive FTP, where the client initiates the data connection, and for active FTP, where the server initiates the data connection. Typically, the firewall should block connections initiated from outside the protected domain.

Maintaining Integrity of Other Protected Services

The FireWall-1 security Server validates the random ports used in PORT command by the FTP client or by the FTP security server. This prevents a port being randomly chosen that is in use by a defined service. This protects against the risk of data connection initiation to another active/working service in the protected domain

Avoiding Vulnerabilities in FTP Applications

An exploit could be placed in the value of the PORT command. PORT commands are usually interpreted using string manipulation functions that cause security risks. The FTP Security Server performs a sanity validation for the PORT command parameter.

Protecting against the FTP Bounce Attack

The FireWall-1 security Server verifies that an IP address presented on a PORT command is identical to the source address of the client. This protects against the FTP bounce attack

Content Security via the FTP Resource

It is possible to allow only file downloads (by specifying GET as an allowed method) or only uploads (by specifying PUT as an allowed method), or both in an FTP resource.

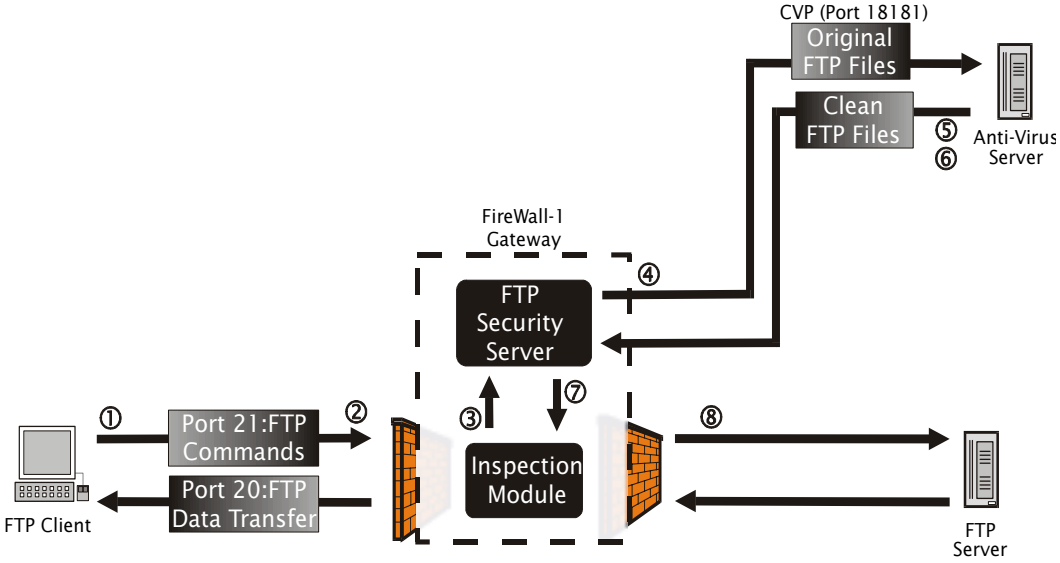
It is also possible to restrict connections to a particular path and/or filename. This protects against exposure of the FTP server's file system.

Virus checking an FTP connection using CVP

Virus scanning on FTP connections can be performed by transferring the file to a third party anti-virus application using the CVP protocol. This is done by defining an FTP resource in SmartDashboard.

FIGURE 4-8 illustrates how FireWall-1 implements CVP for virus checking in an FTP connection.

FIGURE 4-8 CVP Inspection process during an FTP Connection



The relevant rule for the connection specifies a resource that includes Content Vectoring Protocol (CVP) for anti-virus checking.

1 The FTP client establishes a connection via port 21 to the FTP server.

2. The Inspection Module monitors port 21 for GET and PUT commands, and determines that the CVP Server must be invoked,
3. When the client initiates a data transfer over port 20, the Inspection Module folds the connection into the FTP Security Server.
4. The FTP Security Server sends the file to be inspected to the CVP Server.
5. The CVP Server scans the FTP files and returns a Validation Result message, notifying the FTP Security Server of the result of the scan.
6. The CVP Server returns a clean version of the file to the FTP Security Server.
7. The FTP Security Server determines whether to transfer the file, based on the Validation Result message, and takes the action defined for the resource, either allowing or disallowing the file transfer.
8. If allowed, the FTP Security Server relays the FTP file on to the FTP server.

TCP Security Server

The TCP Security Server is used to perform CVP or UFP Content Security by a third party OPSEC compliant application, on any TCP Service.

For configuration details, see “Performing CVP or UFP Inspection on any TCP Service” on page 108.

Securing Microsoft Networking Services (CIFS)

CIFS (Common Internet File System) is a protocol used to request file and print services from server systems over a network.

The protocol is an extension of the Server Message Block (SMB) protocol. CIFS is used as the underlying transport layer for the NETBIOS session (nbsession) service over TCP using port 139. In Windows networking, CIFS is used over the Microsoft-DS protocol (port 445) for networking and file sharing. More information on CIFS can be found at http://samba.org/cifs/.

By default, a Windows server has default shares open for administrative purposes (C$, ADMIN$, PRINT$) and so is an easy target for internal attacks such as brute-force password attacks on file servers.

FireWall-1 secures Microsoft Networking Services in the Inspection Module, without requiring a Security Server. This meets the high performance requirements of LAN security (Fast Ethernet and Gigabit Ethernet).

The CIFS resource can be used to enforce the following security checks on CIFS connections:

- Correctness of the protocol
- Preventing CIFS and NETBIOS messages issued by the client from pointing to beyond message boundaries.
- Restricting access to a list of CIFS servers and disk shares.
- Logging disk share access.

For configuration details, see "Restricting Access to Servers and Shares (CIFS Resource)" on page 108.

Considerations for Web Security

In This Section

Choosing the Web Security Feature Set

FireWall-1 provides the widest possible range of Web Security protection features.

These capabilities can be implemented using either the HTTP Security Server or the Inspection Module in the FireWall-1 kernel.

Security features implemented in the FireWall-1 kernel provide a higher performance to users than those implemented with the Security Server. On the other hand, the Security Server provides a wider range of Web Security capabilities.

TABLE 4-1 to TABLE 4-5 list the full range of FireWall-1 Web Security capabilities. The tables show whether each capability can be implemented using the kernel, or Security Server, or both. The tables show that all capabilities can be implemented in the Security Server (other than the General HTTP Worm Catcher), and some of them can also be implemented in the kernel. These capabilities can all be configured via SmartDashboard using a URI Resource, unless noted otherwise, and some can also be configured in SmartDefense.

It is recommended to use the kernel feature set, if sufficient. This gives the best possible performance. Otherwise, use Security Servers.

It is possible to select whether to use the Security Servers or the kernel.

Security Rule Base rules that use URI Resources with the **Enforce URI capability** option selected, always invoke the Security Server.

For those rules that do not use a Resource, three options are available in SmartDefense, in the branch **Application Intelligence > Web > HTTP Protocol Inspection**:

1) **Configurations apply only to connections related to resources**

   No other Web Security capabilities are enforced.

2) **Configurations apply to all connections: Perform optimized protocol enforcement**

   Three of the four SmartDefense-selectable HTTP Protocol Enforcement capabilities will be enforced in the kernel. These being **HTTP Format Sizes**, **ASCII Only Request Headers**, and **Peer to Peer**.

3) **Configurations apply to all connections: Perform strict protocol enforcement**

   - Three of the four SmartDefense-selectable HTTP Protocol Enforcement capabilities will be enforced in the Kernel. These being **HTTP Format Sizes**, **ASCII Only Request Headers**, and **Peer to Peer**.
   - The following additional set of capabilities are enforced by the Security Server (even though no Resource is used in the Rule Base):
     - *Directory Traversal Defense*
     - *Allowing only RFC Compliant HTTP Methods*
     - *Response is HTTP*

Summary of FireWall-1 Web Security Capabilities

TABLE 4-1 to TABLE 4-5 list the full range of FireWall-1 Web Security capabilities. The tables show whether each capability can be implemented using the kernel, or Security Server, or both. The tables show that all capabilities can be implemented in the Security Server (other than the General HTTP Worm Catcher), and some of them can

also be implemented in the kernel. These capabilities can all be configured via SmartDashboard using a URI Resource, unless noted otherwise, and some can also be configured in SmartDefense.

TABLE 4-1 Granular URL Security

Security Capability	Kernel
▪ Custom URL Blocking See "Understanding HTTP Sessions" on page 66.	Yes
▪ HTTP Worm Catcher See "Malicious activity Prevention - General HTTP Worm Catcher" on page 71. SmartDefense: **Web Security > General HTTP Worm Catcher** **Note** - Cannot be implemented in the Security Server.	Yes

TABLE 4-2 Protecting Web Servers at the Application Layer

Security Capability	Kernel
▪ Cross-Site Scripting Defense See "Cross-Site Scripting Attack" on page 69.	No
▪ Directory Traversal Defense See "Directory Traversal Attacks" on page 72. (Not available via SmartDashboard)	No
▪ Securing XML Web Services (SOAP) See "Securing XML Web Services (SOAP)" on page 73.	No

TABLE 4-3 Protecting Web Servers at the HTTP Layer

Security Capability	Kernel
▪ Maximum URL Length See "Limiting the URL length" on page 74. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection > HTTP Format size**	Yes
▪ Maximum Header Length See "HTTP Headers Centered Security" on page 75. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection > HTTP Format size**	Yes[1]
▪ Maximum Number of Headers See "HTTP Headers Centered Security" on page 75. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection>Peer to Peer**	Yes[1]

TABLE 4-3 Protecting Web Servers at the HTTP Layer

▪ ASCII Only Request Headers See “ASCII Only Request Headers” on page 74. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection>ASCII Only Request Headers**	Yes
▪ Allowing only RFC Compliant HTTP Methods See “Blocking non-standard HTTP methods” on page 74. (Not available via SmartDashboard)	No
▪ Allowed Schemes See “HTTP Headers Centered Security” on page 75.	No

1. Only on HTTP Request, not on Response.

TABLE 4-4 Protecting Web Surfers

Security Capability	**Kernel**
▪ CVP and Anti-Virus Protection See “CVP and Anti-Virus Protection of Web Traffic” on page 75.	No
▪ Response is HTTP See “ASCII Only Response Headers and HTTP RFC Compliance” on page 77. (Not available via SmartDashboard)	No
▪ ASCII Only Response Headers See “Improving CVP Performance for Web Traffic” on page 76. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection > ASCII only response headers**	No
▪ Strip Script Tags See “Java and ActiveX Security” on page 77.	No
▪ Strip Applet tags See “Java and ActiveX Security” on page 77.	No
▪ Strip ActiveX tags See “Java and ActiveX Security” on page 77.	No
▪ Strip FTP links See “Java and ActiveX Security” on page 77.	No
▪ Strip Port Strings See “Java and ActiveX Security” on page 77.	No
▪ Block Java Code See “Java and ActiveX Security” on page 77.	No

TABLE 4-5 Limiting Web Surfers

Security Capability	Kernel
▪ URL Filtering using a UFP Server See "URL Filtering and UFP" on page 77.	Yes
▪ Securing Peer-to-Peer applications See "Securing Peer-to-Peer applications" on page 80. SmartDefense: **Application Intelligence > Web > HTTP Protocol Inspection>Peer to Peer**	Yes[1]

1. Only on HTTP Request, not on Response.

Enhancing CVP and UFP Performance

Enhancing UFP Performance

Enhanced UFP Performance mode and UFP checking using the FireWall-1 HTTP Security Server are different ways of doing UFP Filtering, and are explained in "URL Filtering and UFP" on page 77. When deciding which way to use, you must balance performance against security.

Users browsing Web sites when Enhanced UFP Performance mode is used, can be expected to experience significantly improved response times as compared to UFP checking using the FireWall-1 HTTP Security Server. However, in this mode (called **Enhanced UFP Performance** in the URI Resource), UFP caching, CVP checking and authentication cannot be used, and certain HTTP Header verifications, such as methods and length are not performed.

Enhancing CVP Performance

HTTP Security Server performance can be significantly improved by ensuring that safe traffic is not sent to the CVP server, which reduces the amount of traffic sent to the CVP server, and reduce the number of connections opened with the CVP server.

However, sending all content for CVP checking gives more certain protection.

See "Improving CVP Performance for Web Traffic" on page 76 for details.

Factors that Affect Security Server Performance

On multiple CPU machines, running more than one instance of the Security Servers increases the performance seen by users. This is because each Security server uses a different CPU. Run at least one Security Server instance for each CPU.

It may well be worthwhile to run more than one Security Server even in a single CPU machine, in order to allow more concurrent connections. However, this will increase the memory usage.

The number of Simultaneous Security Server Connections

Each Security Server allows up to 1024 file descriptors, which limits the number of simultaneous connections. In an ordinary connection, packets pass in both directions through the FireWall-1 enforcement module, as follows:

1. Web client to FireWall-1 to Web server (request)
2. Web server to FireWall-1 to Web client (response)

512 descriptors are available for use in each direction, so that a total of 512 simultaneous connections are possible.

Where a CVP or UFP server is used, packets in each connection pass through FireWall-1 three times:

1. Web client to FireWall-1 to Web server (request)
2. Web server to FireWall-1 to CVP/UFP server (response)
3. CVP/UFP server to FireWall-1 to Web client (response)

Therefore the available file descriptors are split three ways, so that a total of 341 simultaneous connections are possible.

How To Run Multiple Instances of the HTTP Security Server

To run multiple instances of the HTTP Security Server,

1. Edit `$FWDIR/conf/fwauthd.conf`, and include the line

   ```
   80     in.ahttpd     wait     -2
   ```

   The last digit on the line is the number of instances of the Security Server. In this example, there are two instances of the HTTP Security Server.
2. Restart the FireWall-1 enforcement module (`cpstart`).

Configuring Content Security

In This Section

Creating a Resource and Using it in the Rule Base

1. To create a resource, select the Resources branch of the objects tree. Select the Resource Type, right click, select a resource type, such as **New URI...** or **New SMTP...**.
2. Define the resource parameters in the **General** tab, and in the other tabs as required.
3. To use service with a resource in a rule, right click in the **Service** column of the rule, right click, and select **Add with Resource...**. In the **Service with Resource** window, select the service, and then select the Resource that will operate on the service. Click **OK**.

If a connection matches the source and destination of the rule and the match parameters of the Resource, then both the Action in the Rule and the Action in the Resource are applied.

URL Logging

1. Create a URI Resource.
2. Log URL in the connection in one of the following ways:
   - To log the URLs including the URL paths and query, by performing kernel inspection: In the **General** tab of the **URI Resource Properties** window, select **Optimize URL Logging**.
   - For basic URL logging using the Security Server: In the **General** tab of the **URI Resource Properties** window, select **Enforce URI Capabilities**.

     The **Exception Track** option specifies how to track connections that match this rule but fail the content security checks. An example of an exception is a connection with an unsupported scheme or method.

3 Place the URI Resource in a rule with the **Action** *Accept*.

4 Select *Log* in the **Track** column. This logs the URL of all connections that match this rule.

For background information, see "URL Logging" on page 68.

Basic URL Filtering

To prevent access to selected forbidden Web sites, proceed as follows:

1 Specify a list of forbidden sites in a file that lists the site URIs. The URI specification file is an ASCII file consisting of a list of lines. Each line has the format

```
ip-address /path category
```

- `ip-address` is the IP address of the Web server to be matched. Host names can be used, but DNS must be enabled and configured on the FireWall-1 gateway.
- `/path` is optional. Use it to restrict a particular directory in a site.
- `category` is an optional parameter that can be any Hex number. It is not currently used.

Make sure there is no white space after the category. The last line in the file must be blank. For example

```
192.168.56.78     /games
192.168.25.58
```

The file should contain no more than a thousand records.

2 Define a Resource that uses this file.

3 Use this resource in a Rule, on HTTP Traffic.

4 Define the Action as *Reject*.

For background information, see "Basic URL Filtering" on page 68.

URL Filtering with a UFP Server

Rule Match in UFP Modes "Enforce URI Capabilities" and "Enhance UFP Performance"

There are differences in rule match behavior between UFP rules in **Enforce URI Capabilities** mode (that use the FireWall-1 kernel) and rules in **Enhance UFP Performance** mode (that use the HTTP Security server).

- In **Enforce URI Capabilities** mode, the connection is matched to the Source, Destination, Service, *and* UFP category of the Resource in the rule.

  If the connection does not match to all these, FireWall-1 compares the connection to each rule in the Rule Base until it finds a rule that matches.
- In **Enhance UFP Performance** mode, the connection is matched only to the Source, Destination, and Service in the rule. The connection is *not* matched to the UFP category. If the connection matches the Source, Destination, and Service in the rule, it is *not* matched to any other rule further down the Rule Base.

  In this mode, if connection matches the UFP category, FireWall-1 performs the Action in the rule. If the connection does not match the UFP category, FireWall-1 performs the opposite of the Action specified in the Rule.

What this means is that you may only have one rule with an **Enhance UFP Performance** resource, for a given Source/Destination/Service. In the **Match** tab of the resource, you must include all UFP categories. The Action in the rule takes place if *any* of the selected categories match the connection.

If using **Enforce URI Capabilities** mode in a UFP resource, you may have more than one rule with a resource using this mode, for a give Source/Destination/Service. However, in order to have a simpler and less error prone Rule Base, it is recommended to use only one resource, as for the **Enhance UFP Performance**.

For example, consider the following rules:

No.	Source	Destination	Service	Action
1	Any	Any	Resource with UFP Category “Drugs”	Drop
2	Any	Any	Resource with UFP Category “Alcohol”	Drop

If a connection fits the UFP category of “Alcohol”:

- In **Enhance UFP Performance** mode, the connection matches on Rule 1 and the connection is Accepted — which is not the desired behavior.
- In **Enforce URI Capabilities** mode, the connection matches on Rule 2 and the connection is Dropped.

The correct way to build this rule in a way that will work in all mode, and for greater simplicity, is as follows:

No.	Source	Destination	Service	Action
1	Any	Any	Resource with UFP Categories “Drugs” and “Alcohol”	Drop

Configuring "Enforce URI Capabilities" and "Enhance UFP Performance" UFP modes

This procedure describes how to configure a URL Filtering using the FireWall-1 kernel or using the Security Server. For background information, see "URL Filtering and UFP" on page 77.

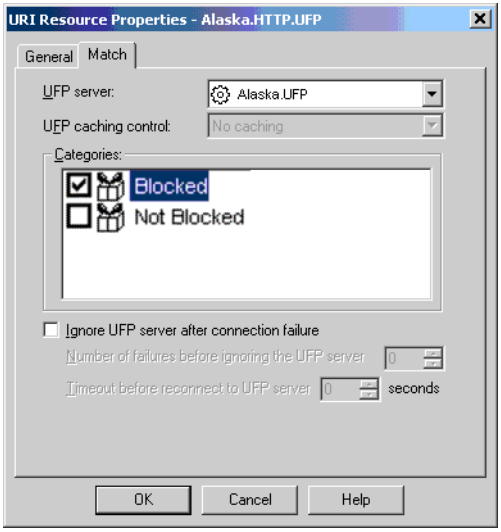
1. Create a Node object for the machine on which the third-party OPSEC Server application is installed.
2. Create an OPSEC Application object (Alaska_HTTP_UFP) to represent the OPSEC application server, and associate it with the Node object created in step 1.
3. Create a new URI resource that uses the OPSEC Application object, and associate it with the OPSEC Application object created in step 2.
4. To perform URL Filtering using the FireWall-1 kernel, select **Enhance UFP Performance**.

   To perform URL Filtering using the Security Server, select **Enforce URI capabilities**, and select **URI Match Specification Type: UFP**.

   In the **Match** tab, select the **UFP server** object that was created in step 2. Check the appropriate **Categories**. Some UFP Servers show just two categories: **Blocked** and **Not Blocked**. Others show many categories.

   FIGURE 4-9 shows a restrictive resource that matches on the **Blocked** category.

FIGURE 4-9 Match tab for a URI Resource for UFP



5 Associate the Resource with the HTTP Service, and place it in a Rule in the Security Rule Base. See the sample rules shown in TABLE 4-6.

The Action in Rule 1 is *Drop* because the resource matches on Blocked categories. If the resource were to match on Not Blocked categories, the Actions in Rules 1 and 2 would be reversed: *Allow* in Rule 1, and *Drop* in Rule 2.

Rule 2 is required for the **Enforce URI Capabilities** mode. For the **Enhance UFP Performance** mode it is recommended to avoid problems in cases where more than one URI resource is used in the Rule Base.

TABLE 4-6 Sample UFP Rule Base Policy

No.	Source	Destination	Service	Action
1	Any	Any	http->Alaska_HTTP_UFP	Drop
2	Any	Any	http	Accept

Blocking URL-based Attacks using a URI Resource

All URL-based attacks such as Code Red and Nimda can be blocked using a URI resource in SmartDashboard. Each resource can block one attack. For background information, see "URLs and methods" on page 68.

Proceed as follows:

1 Create a new URI Resource, and give it a name (such as Alaska.Web.Protection)

2 In the **General** tab, select
- **Use this resource to: Enforce URI capabilities**
- **Connection Methods:** Normally **Transparent** and **Proxy** are selected
- **URI Match Specification Type: Wild Cards**

3 Specify the URL pattern, using regular expressions in the Match tab. For example, to block Code Red, use the following values:
- **Host:** `*`
- **Path:** `\.ida\?`
- **Query:** `*`

4 If you wish to specify a replacement URL to redirect the connection if the pattern is found, specify a **Replacement URI** in the **Action** tab.

5 Associate the Resource with the HTTP Service, and place it in a Rule in the Security Rule Base. See the sample rules shown in TABLE 4-7.

TABLE 4-7 Sample URI Resource in a Rule Base

No.	Source	Destination	Service	Action
1	Any	Any	http->Alaska.Web.Protection	Drop
2	Any	Any	http	Accept

The Action in Rule 2 is the opposite of the Action in Rule 1. Rule 2 is required for the **Enforce URI Capabilities** mode. For the **Enhance UFP Performance** mode it is recommended to avoid problems in cases where more than one URI resource is used in the Rule Base.

Blocking Peer-to-Peer applications and HTTP Header Based Attacks

To block Peer-to-Peer applications and HTTP header based attacks, use SmartDefense to specify forbidden HTTP headers and header patterns using regular expressions. SmartDefense is pre-configured with settings to block a number of applications, including ICQ, Kazaa, MSN Messenger, Yahoo Messenger and Gnutella.

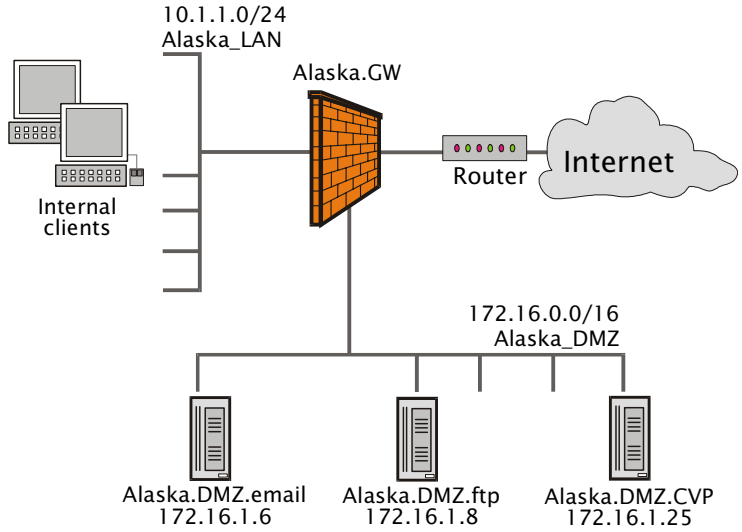
In SmartDefense, select **Application Intelligence > Web > HTTP Protocol Inspection > Peer to Peer**.

For background information, see “Securing Peer-to-Peer applications” on page 80.

Anti-Virus Checking for Incoming Email

The goal is to check for viruses in incoming mail, as illustrated in FIGURE 4-10. The mail server in the DMZ is to be accessible via SMTP from anywhere, and can send mail to anywhere.

FIGURE 4-10 Sample Configuration - illustrating Anti-Virus Checking for Incoming Email



General Procedure

1. Create a Node object for the machine on which the third-party OPSEC Server application is installed.
2. Create an OPSEC Application object to represent the OPSEC application server, and associate it with the Node object created in step 1.
3. Define an SMTP resource that uses the OPSEC Application object, and associate it with the OPSEC Application object created in step 2. Specify the matching, and the content checking to be done.
4. Define rules that use the resource.

Implementation

1. Create a Node object for the machine on which the third-party OPSEC Server application is installed.
2. Create an OPSEC Application object to represent the OPSEC application server, and associate it with the Node object created in step 1. Initialize **Secure Internal Communication** between the OPSEC Application and the SmartCenter Server. In the **CVP Options** tab, verify that FW1_cvp is selected, and click **OK**.

3 Define an SMTP resource that uses the OPSEC object, and associate it with the OPSEC Application object created in step 2. Specify the matching, and the content checking to be done.

In the **General** Tab, give the Resource a **Name** (such as Virus_Check). Choose the **Mail Delivery** and the **Error Mail Delivery** options, and the **Exception Tracking**.

In the **Match** tab, for the **Sender** put *, and for the **Recipient** put **@your_domain*, (for example *@company.com)

In the **Action1** tab, define the **Rewriting Rules**, if any.

In the **Action2** tab, define the **Attachment handling**, if any. Define the largest allowed email attachment.

4 In the **CVP** tab, check **Use CVP (Content Vectoring Protocol)**, select the **CVP Server** defined in step 1, and define the **CVP Server Options** and **Reply Order**.

Click **OK**. A message may appear regarding stripping MIME of type "message/partial'". Accepting the MIME strip of type "message/partial" will result in a configuration change to the Action2 tab. The Strip MIME of Type field will contain message/partial. Stripping the Multipurpose Internet Mail Extension (MIME) type of message/partial will not allow multiple-part messages to be accepted for scanning.

5 Define a Rule that will perform virus checking on incoming mail, and a Rule to allow outbound email. The Rules should look similar to the one in FIGURE 4-11.

6 Install the Security Policy.

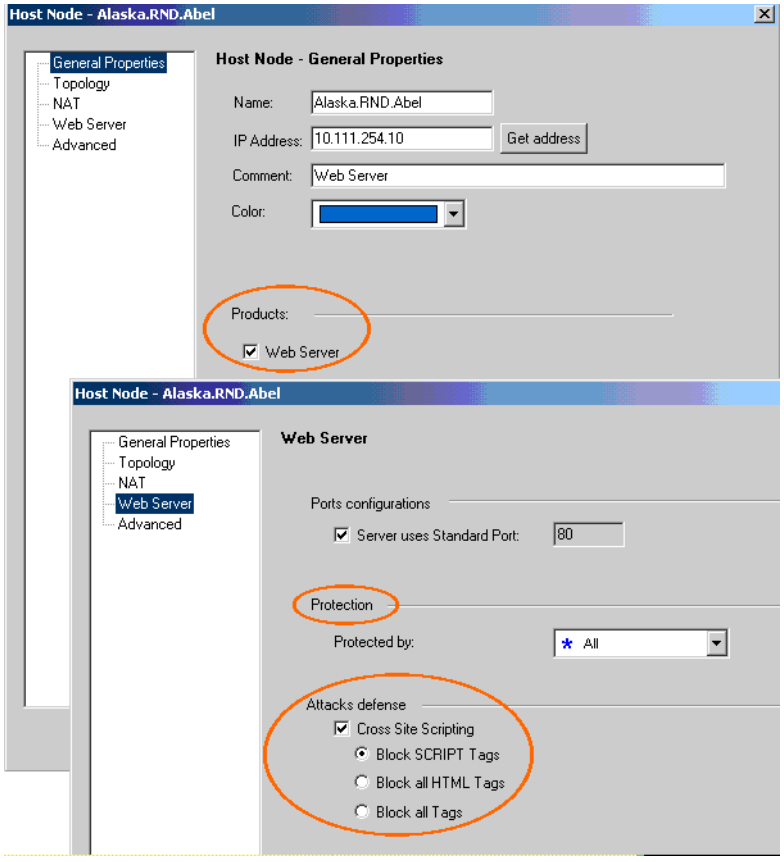
FIGURE 4-11 Example Rules for SMTP Virus Scanning

NO.	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
1	alaska.DMZ.emai	* Any	*	TCP smtp	acce	Log	* P	* Ar	Outgoing email
2	Alaska_LAN	alaska.DMZ.ema	*	HTTP http->Virus_Check	acce	Log	* P	* Ar	Incoming email virus scan

Configuring Web Server Security via the Network Object

To secure Web servers via the network object, Check **Web Server** in the **General Properties** page of the network object. A new properties page appears called **Web Server**, where attack defense settings are configured. This is shown in FIGURE 4-12.

FIGURE 4-12 Defining a Network Object as a Web Server



In this version of FireWall-1 defense against Cross Site Scripting attack can be configured from the Web server page.

Web Servers can also be configured via SmartDefense (**Application Intelligence > Web > > Cross site Scripting**). It is possible to centrally configure protection against Cross Site Scripting attacks for all defined Web servers, or per Web server.

For background information, see “Securing Web Servers via the Network Object” on page 69.

Protection Against Cross-Site Scripting Attacks

To configure Web servers via the network object, Check **Web Server** in the **General Properties** page of the network object. A new properties page appears called **Web Server**. Configure defense against Cross Site Scripting attacks in this page.

Web Servers can also be configured via SmartDefense. In SmartDefense it is possible to centrally configure protection against Cross Site Scripting attacks for all defined Web servers, or to per Web server. In the SmartDefense tab, select **Application Intelligence > Web Security > Cross site Scripting**.

- **Block** *Script Tags* is the least strict approach. It rejects any occurrence of a request for a keyword in a predefined list of banned tags. The list cannot be edited.
- **Block** *HTML tags* is a more cautious approach. It rejects all HTTP requests that contain the < or > characters. The disadvantage of this approach is that it can block access to pages that contain innocent tags, such as <Title>.
- **Block** *All Tags* is the strictest and safest approach. It includes all tags in the **Block** *HTML tags option*. In addition, ampersand encoding of the characters "<" (that is, &lt, and &#60) and ">" (&gt and &#62) is also blocked. Ampersand (&) encoding is used to encode special ASCII characters in HTML.

For background information, see "Cross-Site Scripting Attack" on page 69.

Improving the Performance of the CVP Server

The performance of the CVP server when inspecting HTTP connections can be enhanced by ensuring that only unsafe file types are sent to the CVP server for inspection. For background information, see "Improving CVP Performance for Web Traffic" on page 76.

Proceed as follows:

1. Create a Node object for the machine on which the CVP Server application is installed.
2. Create an OPSEC Application object to represent the CVP server, and associate it with the Node object created in step 1.
3. Define a URI resource that uses the OPSEC Application object, and associate it with the OPSEC Application object created in step 2. Give it a name (such as Alaska.HTTP.CVP), specify the matching, and the content checking to be done.
4. In the CVP tab, select **Send only unsafe file types to the CVP server**, and the other required CVP options.

5 Associate the Resource with the HTTP Service, and place it in a Rule in the Security Rule Base. See the sample rules shown in TABLE 4-7.

TABLE 4-8 Sample URI Resource in a Rule Base

No.	Source	Destination	Service	Action
1	Alaska_LAN	Alaska.DMZ.Web	http->Alaska.HTTP.CVP	Accept
2	Alaska_LAN	Alaska.DMZ.Web	http	Drop

Make sure the Action in Rule 2 is the opposite of the Action in Rule 1.

FTP Content Security- Restricting access to a specific Directory

The goal is to restrict access to a specific directory on the FTP Server when uploading files from the internal network, but to allow files to be downloaded from anywhere on the FTP Server to the internal network.

Referring to FIGURE 4-10, the FTP Server is Alaska.DMZ.ftp.

Two resources must be created. One for upload, and one for download.

1 Create an FTP Resource to allow file downloads (**Manage > Resources**, click **New > FTP...**).

In the **General** tab, **Name** the resource (Downloads, for example), and choose a **Tracking Option** (such as *Log*).

In the **Match** tab, type the allowed directory path using wildcards, for example, * to allow any directory and filename. Under **Methods**, select *GET*.

2 Create an FTP Resource to allow file uploads.

In the **General** tab, **Name** the resource (Uploads, for example), and choose a **Tracking Option**.

In the **Match** tab, type the allowed directory path and filename, using wildcards. For example `/uploads/*`. Under **Methods**, select *PUT*.

3 Define a Rule to allow file uploads, and another Rule to allow file downloads. The Rules should look similar to the ones in FIGURE 4-13.

FIGURE 4-13 Example Rules for FTP Upload and Download

NO.	SOURCE	DESTINATION	IF VIA	SERVICE	ACTION	TRACK	INSTALL ON	TIME	COMMENT
3	Alaska_LAN	Alaska.DMZ.ftp	*	ftp->Upload	accept	Log	* P	* Ar	ftp upload to /uploads/*
4	Alaska_LAN	Alaska.DMZ.ftp	*	ftp->Download	accept	Log	* P	* Ar	ftp download from *

4 Install the Security Policy.

Performing CVP or UFP Inspection on any TCP Service

To configure CVP or UFP inspection on any TCP service:

1 In the **TCP Service Properties**, **Advanced** tab, check **Enable for TCP Resource**.

2 Create a TCP Resource. In the **General** tab, choose **CVP** or **UFP**, and the **Exception Track** method.

3 Configure settings in the **CVP** or **UFP** tab.

4 Add a Rule to the Rule Base, and in the **Service** column, select **Add with Resource**,

5 In the **Service with Resource** window, select the TCP service configured in step 1. under **Resource**, select the resource created in step 2.

6 Install the Security Policy.

For background information, see "TCP Security Server" on page 90.

Restricting Access to Servers and Shares (CIFS Resource)

1 Define a new CIFS Resource

2 Configure the CIFS Resource. **Allowed Disk\Print Shares** is a list of allowed CIFS servers and disk shares. Note that the use of wildcards is allowed. Use **Add/Edit/Delete** to modify the list. For example to allow access to the disk share PAUL on the CIFS server BEATLES proceed as follows:

Click **Add** and type BEATLES in the **Server Name** field and IPC$ in the **Share Name** field. Click **OK**.

Click **Add** again and type BEATLES in the **Server Name** field and PAUL in the **Share Name** field. Click **OK**.

3 Add a new rule. Under **Service**, add either nbsession or Microsoft-DS together with the configured Resource.

4 Install the Security Policy.

For background information, see "Securing Microsoft Networking Services (CIFS)" on page 90.

CHAPTER 5

Authentication

In This Chapter

The Need for Authentication

People in different departments and with different levels of responsibility must be given different access permissions to different parts of the network. It is therefore necessary to allow access only to valid users. Determining who is a valid user is the job of Authentication.

FireWall-1 Solution for Authentication

In This Section

Introduction to FireWall-1 Authentication

FireWall-1 authenticates individual users via the use of a credentials. FireWall-1 can manage credentials using a number of different Authentication Schemes. All Authentication Schemes in FireWall-1 rely on some sort of username and password. Some of these schemes involve storing the passwords on the FireWall-1 enforcement module. In other schemes, passwords are stored on external servers.

There are three ways in which users can authenticate themselves to FireWall-1. The Administrator controls this by deciding on the Authentication Method for each user. The available Authentication Methods are: User Authentication, Session Authentication, and Client Authentication.

Authentication ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual. Giving individuals access to network resources based on their identity, is called authorization. Basic FireWall-1 authorization is performed by taking a decision to accept or to drop a connection.

More sophisticated authorization requires a separate Check Point product called UserAuthority. UserAuthority enables applications to make intelligent authorization decisions based on FireWall-1 authentication and security information.

Authentication Schemes for Password Management

This section discusses how FireWall-1 can manage user credentials in a number of different ways, called Authentication Schemes.

All authentication schemes in VPN-1/FireWall-1 rely on some sort of username and password. Some of these schemes involve storing the passwords on the FireWall-1 enforcement module. In other schemes, passwords are stored on external servers.

All the schemes can be used with users defined on an LDAP server.

VPN-1 & FireWall-1 Password

The VPN-1 and FireWall-1 password scheme is a simple, static password that is maintained internally on the FireWall-1 Gateway machine. No additional software is needed.

OS Password

FireWall-1 can use the user and password information that is stored in the operating system of the machine on which FireWall-1 is installed. It is also possible to use passwords that are stored in a Windows domain. No additional software is needed.

Radius

Radius is an authentication server usually used by those providing dial-up access for authentication. Radius Servers and Radius Server Group objects are defined in SmartDashboard.

SecureID

SecureID uses a hardware token with a value that changes every minute or so. The card is synchronized with an ACE/Server, which validates the authentication attempt. The FireWall-1 enforcement module acts as an ACE/Agent 5.0. There are no scheme-specific parameters for the SecureID authentication scheme. For agent configuration see ACE/Server documentation.

TACACS

TACACS provides access control for routers, and network access servers and other networked devices via one or more centralized servers. A newer version of the protocol called TACACS+ provides enhancements to the original protocol, including the use of TCP instead of UDP.

Undefined

The authentication scheme for a user can be specified as undefined. If a user with an undefined authentication scheme is matched to a Security Rule, he or she is always denied access.

Asking for Passwords Using Authentication Methods

There are three ways in which users can authenticate themselves to FireWall-1. The Administrator controls how users authenticate by deciding on the Authentication Method for each user. The available Authentication Methods are:

- *User Authentication*.
- *Session Authentication*.

- *Client Authentication*.

Both transparent and non-transparent authentication can be used. In transparent authentication the user initiates a connection directly to the target server. In non-transparent authentication, the user explicitly connects to the FireWall-1 Gateway.

This section describes how the user authenticates using each of these methods, and how they work. How to set up the different authentication methods is described in "Configuring Authentication" on page 121.

User Authentication

User Authentication provides authentication for five services: TELNET, FTP, HTTP, HTTPS, and RLOGIN. By default, User Authentication is transparent. The user does not explicitly connect to the FireWall-1 Gateway, but initiates a connection directly to the target server.

The following example demonstrates a Telnet session to 10.11.12.13, with User Authentication and the OS Password authentication scheme (Rlogin works in almost exactly the same way):

```
# telnet 10.11.12.13
Trying 10.11.12.13...
Connected to 10.11.12.13.
Escape character is '^]'.
Check Point FireWall-1 authenticated Telnet server running on
tower
User: fbloggs
FireWall-1 password: *******
User fbloggs authenticated by FireWall-1 authentication
Connected to 10.11.12.13
...
...
login:
```

User Authentication works as follows:

When a rule specifies user authentication, the corresponding Security Server process is invoked. The Security Server authenticates the session, and then passes it on to the remote server.

1. FireWall-1 intercepts the communication between the client and server. If a rule specifies user authentication, the relevant Security Server process is invoked.
2. The FireWall-1 Security Server prompts for a username and password.

3 If the user successfully authenticates, the Security Server authenticates the session, and then passes it on to the remote host. If the correct authentication information is not supplied by the user within the allowed number of connection attempts, the connection is dropped.

4 The remote hosts prompts for its own username an password.

The following example demonstrates an FTP session to 10.11.12.13, with User Authentication and the OS Password authentication scheme.

```
# ftp 10.11.12.13
Connected to 10.11.12.13.
220 london Check Point FireWall-1 Secure FTP server running on tower
Name (10.11.12.13:fbloggs):
```

Now the username must be entered in the following format:

```
FireWall-1 User@Destination Host
```

This format is demonstrated as follows:

```
fbloggs@10.11.12.13
331-aftpd: FireWall-1 password: you can use FW-1-password
```

Now enter the FireWall-1 password

```
Password: xyz987
230-aftpd: User fbloggs authenticated by FireWall-1 authentication.
230-aftpd: Connected to 10.11.12.13. Logging in...
230-aftpd: 220 bigben ftp server (UNIX(r) System V Release 4.0) ready.
ftp>
```

At this point you will be connected to the remote FTP server. Log in using the user command:

```
ftp> user anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password: fbloggs@checkpoint.com
230 Anonymous user logged in.
ftp>
```

Timeout Considerations for User Authentication of HTTP

In HTTP, the Web browser automatically supplies the password to the server for each connection. This creates special security considerations when using User Authenticating for HTTP with one-time passwords.

To avoid forcing users of one-time passwords to generate a new password for each connection, the HTTP Security Server extends the validity of the password for the time period defined in **User authentication Session timeout** in the **Authentication** page of the **Check Point Gateway** window. Users of one-time passwords do not have to reauthenticate for each request during this time period.

Each successful access resets the timer to zero. Because the authorization period is renewable, and the Web browser keeps supplying the password, the time period during which a one-time password can be used can be unlimited.

This problem can be solved by using the Reauthentication options in the **HTTP Server** definition of the **Global Properties>FireWall-1>Security Server** page. For example, you can specify that every request to a specific HTTP server requires a new password, or that requests that change a server's configuration require a new password.

Session Authentication

Session Authentication can be used for any service, but requires either a Session Authentication agent to get the user identity, or UserAuthority. Like User Authentication, it requires an authentication procedure for each connection.

UserAuthority can be used to get the user identity. It can do this in one of three ways:

1) From a SecureAgent

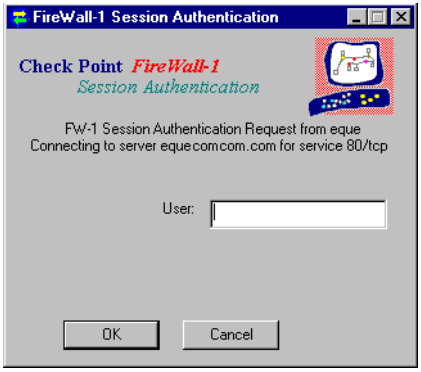
2) From SecureClient, if the user authenticated via SecureClient connected to the FireWall-1 Gateway.

3) From the FireWall-1 Gateway, if the user authenticated via an HTTP connection to port 900 or Telnet to port 256 on the Gateway.

A Session Authentication agent can also be used to get the user identity. The Session Authentication agent is normally installed on the authenticating client, in which case the person who wants to the connection to the destination host supplies the authentication credentials. However, the Session Authentication agent can also be installed on the destination machine, or on some other machine in the network. In that case, the person at the machine on which the Agent is installed is asked to supply the username and a password.

FIGURE 5-1 shows the Session Authentication login prompt. After typing his or her username, another prompt asks the user to supply a password.

FIGURE 5-1 Session Authentication login prompt



Session authentication using a Session Authentication agent works as follows:

1) The user initiates a connection directly to the server.

2) The FireWall-1 Inspection Module intercepts the connection. If a rule specifies Session Authentication, the Inspection Module connects to a Session Authentication agent.

3) The Session Authentication agent challenges the user for authentication data and returns this information to the Inspection Module.

4) If the authentication is successful, then the FireWall-1 inspection module allows the connection to pass through the gateway and continue on to the target server.

Client Authentication

Client Authentication can be used to authenticate any service. It allows access from a specific IP address for an unlimited number of connections. The user working on a client performs the authentication by successfully meeting an authentication challenge, but it is the client machine that is granted access.

Client authentication can be used with any one of five different sign on methods. These sign on methods provide a choice of Authentication Methods for authenticated and other services, as summarized in TABLE 5-1. For all sign on methods other than Manual Client Authentication, the FireWall-1 Gateway is transparent to the user. This means that the user authenticates directly to the destination host.

TABLE 5-1 Client Authentication Sign On Methods

Client Authentication Sign On Method	Authentication Method for authenticated services: TELNET, FTP, HTTP, HTTPS, RLOGIN	Authentication Method for other services
Manual	Telnet to port 259 on Gateway HTTP to port 900 on Gateway	Telnet to port 259 on Gateway HTTP to port 900 on Gateway
Partially automatic	User Authentication	Not available
Fully automatic	User Authentication	Session Authentication
Agent Automatic	Session Authentication	Session Authentication
Single Sign On	UserAuthority	UserAuthority

There are two other choices to make with Client Authentication:

- Standard Sign On
- Specific Sign On

Standard Sign On allows the user to use all the services permitted by the rule, without having to perform authentication for each service.

Specific Sign On allows the user to access only the services they specify when they authenticate, even if the rule allows more than one service. If the user wishes to use another service, they need to reauthenticate for that specific service.

At the end of the session, the user can sign off. When a user signs off, he or she is signed off from all services, and the connection is closed by the remote host.

An explanation follows of each of the of the Client Authentication sign on methods summarized in TABLE 5-1:

Manual Sign On

Manual Sign on is available for any service, as long as it is specified in the Client Authentication rule.

In Manual Sign On, the user must first connect to the Gateway in order to authenticate (in other words, the authentication is not transparent). The user must authenticate in one of the two ways:

1) A TELNET session to the Gateway on port 259

2) An HTTP connection to the Gateway on port 900, through a Web browser. The requested URL must include the gateway name and the port number, such as http://Gateway:900

The following example shows what Client Authentication with Standard, Manual Sign On looks like to a user. Before opening a connection to the destination host, user **fbloggs** first authenticates to **london**, the FireWall-1 Gateway:

```
tower 1% telnet london 259
Trying 191.23.45.67 ...
Connected to london.
Escape character is '^]'.
CheckPoint FireWall-1 Client Authentication Server running on
london
Login: fbloggs
FireWall-1 Password: ********
User authenticated by FireWall-1 auth.

Choose:
    (1) Standard Sign On
    (2) Sign Off
    (3) Specific Sign On

Enter your choice: 1

User authorized for standard services (1 rules)
Connection closed by foreign host.
```

The following example shows what Client Authenticating with Specific, Manual Sign On looks like to a user. In the example, two services are specified: `rstat` and `finger`, each one to a different host.

```
tower 3% telnet london 259
Trying 191.23.45.67 ...
Connected to london.
Escape character is '^]'.
CheckPoint FireWall-1 Client Authentication Server running on
london
Login: jim
FireWall-1 Password: ********
User authenticated by Internal auth.

Choose:
    (1) Standard Sign On
    (2) Sign Off
    (3) Specific Sign On

Enter your choice: 3
Service: rstat
Host: palace
Client Authorized for service
Another one (Y/N): Y
Service: finger
Host: thames
Client Authorized for service
Another one (Y/N): n
Connection closed by foreign host.
```

Wait Mode for Client Authentication

Wait Mode is a Client Authentication capability for the FireWall-1 Gateway that applies to Manual Sign On, when the user initiates a Client Authenticated connection with a Telnet session to port 259 on the Gateway.

Wait Mode makes it unnecessary to open a new Telnet session in order to Sign Off and withdraw Client Authentication privileges. In Wait mode, the initial Telnet session remains open, and Client Authentication privileges remain valid so long as the connection is open. The privileges are withdrawn when the Telnet session is closed.

FireWall-1 keeps the Telnet session open by pinging the authenticating client. If for some reason the client machine stops running, FireWall-1 closes the Telnet session and Client Authentication privileges from this IP address are withdrawn.

Enable Wait Mode works only with Client Authentication rules which specify Standard Sign On. If you select **Enable Wait Mode**, Client Authentication rules which require Specific Sign On are not applied.

Partially Automatic Sign On

Partially Automatic Sign On is available for the authenticated services TELNET, FTP, HTTP, HTTPS, and RLOGIN services, as long as they are specified in the Client Authentication rule. No other services can be authenticated with Partially Automatic Client authentication.

If the user attempts a connection to a remote host using one of the authenticated services, he or she are asked to authenticate by means of User Authentication.

Fully Automatic Sign On

Fully Automatic Sign On is available for any service, as long as the required service are specified in the Client Authentication rule.

If the user attempts a connection to a remote host using an authenticated service (TELNET, FTP, HTTP, HTTPS, and RLOGIN), he or she are asked to authenticate by means of User Authentication.

If the user attempts a connection to a remote host using any other service, he or she are asked to authenticate by means of the Session Authentication agent, which must be properly installed.

Agent Automatic Sign On

Agent Automatic Sign On is available for any service, as long as the required service are specified in the Client Authentication rule, and as long as the Session Authentication agent is properly installed.

If the user attempts a connection to a remote host using any service, he or she are asked to authenticate by means of the Session Authentication agent.

Single Sign On

Single Sign On is available for any service, as long as the required service are specified in the Client Authentication rule. UserAuthority must be installed.

Single Sign On is the Check Point address management feature that provides transparent network access. In this method, the FireWall-1 consults the user IP address records to determine which user is logged on at a given IP address. If a connection matches a Single Sign On enabled rule, the FireWall-1 sends a query to the UAS with the packet's source IP. The UAM returns the name of the user who is registered to the IP. If the user's name is authenticated, the packet is accepted; if not, it is dropped.

Which Authentication Method to Choose

With User Authentication, the administrator can allow the user who is away from his or her desk, to work on the local network without extending access to all users on the same host. However, User Authentication is available only for the services TELNET, FTP, HTTP, HTTPS, and RLOGIN.

Client Authentication is less secure than User Authentication because it allows multiple users and connections from the authorized IP address or host. The authorization is per machine. For example, if FINGER is authorized for a client machine, then all users on the client are authorized to use FINGER, and will not be asked to supply a password during the authorization period. For this reason, Client Authentication is best enabled for single user machines.

The advantage of Client Authentication is that it can be used for any number of connections, for any service, and the authentication can be set to be valid for a specific length of time.

Session Authentication supplies an authentication mechanism for any service, and demands that users supply their credentials per connection (session). It therefore requires either UserAuthority, or a Session Authentication agent for every authenticating client. It is therefore not suitable for authenticating HTTP, which opens multiple connections per session. As with Client Authentication, only use it on single user machines, where only one user can come from a given IP at any one time.

Configuring Authentication

In This Section

Creating Users and Groups

Authentication Rules are defined in terms of user groups, rather than in terms of individual users. You must therefore define users and add them to groups. You can define users using the FireWall-1 proprietary user database, or using an LDAP server.

This simple example shows how to create a group, create FireWall-1 users from a template, add the users to the group, and install the user information in the database.

Creating a User Group

1 Select the **User Groups** from the Users and Administrators tab of the Objects tree. Right Click, and select **New Group...**.The **Group Properties** window opens. Give the group a **Name**. You will populate the group later, when creating the users.

Creating a User Template

2 Select the **Users** from the Users and Administrators tab of the Objects tree. In the **Templates** branch right click and select **New Template...**. The **User Template Properties** window is displayed.

3 Give the template a name. In the **Groups** tab, add this user template to all the groups to which users based on this template need to belong. In the **Authentication** tab, choose the appropriate authentication scheme for the user. In the remaining tabs, enter the other properties of the user template.

Once you have created a template, any user you create based on the template will inherit all of the template's properties, including membership in groups. If you modify a template's properties, the change will affect all users created from the template in the future. Users already created from the template will not be affected.

Creating Users

4 In the **Users** branch of the objects tree, right click and choose the template on which the new user's properties will be based. The **User Properties** window is displayed.

5 Enter the data for the user. For any user, you can freely change the properties that user inherited from the template, but they will be changed for the user only. The template remains unchanged.

Install the user information in the database

6 Users and groups can be installed separately from the Rule Base. This means you can update users and groups without re-installing the Rule Base. To install the User Database, select **Policy>Install Database...** from the SmartDashboard menu.

Setting Up Supported Authentication Schemes

7 Edit the **Check Point Gateway** object representing the FireWall-1 Gateway, and in the **Authentication** page, enable the required authentication schemes. The FireWall-1 Gateway must support all the authentication schemes defined for the users. For example, if some users use VPN-1 & FireWall-1 Password, and others use Radius Authentication, check both these schemes.

Configuring User Authentication

1 Configure the Users and Groups that are needed for authentication, and install the User Database (see "Creating Users and Groups" on page 121).

2 Each authenticated service has its own Security server. Ensure that the relevant Security Server is enabled in the `$FWDIR/conf/fwauthd.conf`. Making sure that the line for the required Security Server is not commented out.

3 Define a User Authentication access rule.

a In the **Source** column, right click to select **Add User Access...**, and choose the group.

b If you would like to restrict the location of authenticating users: In the **Location** section of the same window, check **Restrict To** and choose the host, group of hosts, network or group of networks from which users can access.

c In the **Service** field choose the services you would like to authenticate.

d In the **Action** column, choose **User Auth.**

TABLE 5-2 shows an HTTP User Authentication Rule.

TABLE 5-2 User Authentication Rule for HTTP and FTP

SOURCE	DESTINATION	VPN	SERVICE	ACTION
Aaska_Users@Any	Alaska_LAN	Any Traffic	HTTP FTP	User Auth

4 Double click the **Action** column to edit the **User Authentication Action Properties**.

5 If you wish, adjust the **User authentication session timeout** in the **Authentication** page of the FireWall-1 Gateway object.

6 Install the Security Policy.

The importance of Rule Order for User Authentication

When defining one or more User Authentication rule for the services TELNET, FTP, HTTP, HTTPS, and RLOGIN, and there are other non-authentication rules that use these services, make sure the User authentication rule is placed last among these rules.

Configuring Session Authentication

1 If using the Session Authentication agent, install and configure it for all the machines desktops that are to allow Session Authentication (see “Installing and Configuring the Session Authentication Agent” on page 124).

2 Configure the Users and Groups that are needed for authentication, and install the User Database (see “Creating Users and Groups” on page 121).

3 Edit the **Check Point Gateway** object representing the FireWall-1 Gateway, and in the **Authentication** page, enable the required authentication schemes. The gateway must support all the authentication schemes defined for the users. For example, if some users use VPN-1 & FireWall-1 Password, and others use Radius Authentication, check both these schemes.

4 Define a Session Authentication access rule.

a In the **Source** column, right click to select **Add User Access...**, and choose the group. Don't close the window yet.

b If you would like to restrict the location of users: In the **Location** section of the same window, check **Restrict To** and choose the host, group of hosts, network or group of networks from which users can access.

c In the **Service** field choose the services you would like to authenticate.

d In the **Action** column, choose **Session Auth.**

TABLE 5-3 shows a typical Session Authentication Rule.

TABLE 5-3 Session User Authentication Rule for HTTP and FTP

SOURCE	DESTINATION	VPN	SERVICE	ACTION
Aaska_Users@Any	Alaska_LAN	Any Traffic	HTTP FTP	Session Auth

5 Double click the **Action** column to edit the **User Authentication Action Properties**.

6 If you wish, adjust the **Failed Authentication Attempts** settings for Session Authentication in the **Authentication** page of the **Global Properties**.

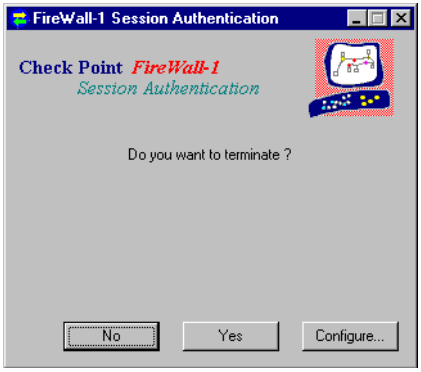
7 Install the Security Policy.

Installing and Configuring the Session Authentication Agent

1 Install the Session Authentication agent from the CD-ROM. The Session Authentication agent is normally installed on the authenticating client, in which case the person who wants to the connection to the destination host supplies the authentication credentials. However, the Session Authentication agent can also be installed on the destination machine, or on some other machine in the network. In that case, the person at the machine on which the Agent is installed is asked to supply the username and a password.

2 Open the Session Authentication agent. On Windows machines, double-click its icon in the system tray. The **FireWall-1 Session Authentication** window (FIGURE 5-2) is displayed.

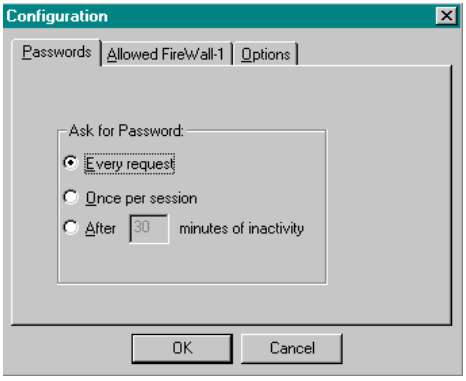
FIGURE 5-2 FireWall-1 Session Authentication window



3 Click **Configure**. The **Configuration** window (FIGURE 5-3) is displayed. The **Configuration** window has three tabs, explained below.

Passwords Tab

FIGURE 5-3 Configuration window — Passwords tab



The **Passwords** tab of the **Configuration** window enables you to specify how frequently the user is asked to supply a password (that is, to authenticate himself or herself). One-time passwords (such as SecurID) cannot be cached.

Check one of the available choices:

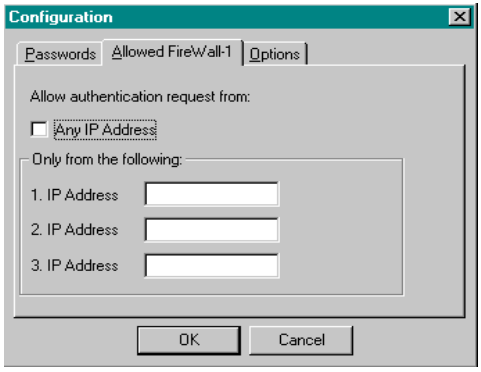
Every request — The user will be prompted for the password each time FireWall-1 requests authentication. Each time the user initiates a session to which a Session Authentication rule applies, the user will be prompted for a password. In this case, no password caching occurs.

Once per session — The user will be prompted for a password once per Session Authentication agent session. In this case, the user supplies the password once and the Session Authentication agent caches the password indefinitely. This option cannot be used with one-time passwords. If the Session Authentication agent is terminated and then re-started, the user will have to supply the password again.

After ... minutes of inactivity — This option is the same as **Once per session**, except that the user will be prompted again for a password if there has been no authentication request for the specified time interval.

Allowed FireWall-1 Tab

FIGURE 5-4 Configuration window — Allowed FireWall-1 tab



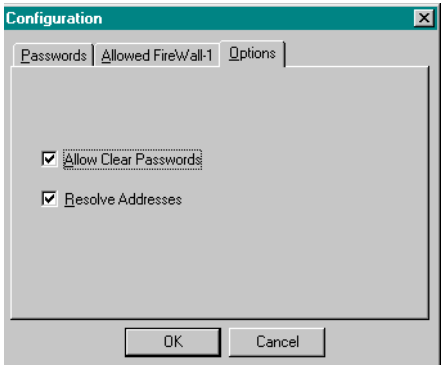
The **Allowed FireWall-1** tab of the **Configuration** window enables you to specify the FireWall-1 Gateways for which this Session Authentication agent may provide authentication services.

Any IP Address — This Session Authentication agent may provide authentication services for any FireWall-1 Gateway.

IP Address — This Session Authentication agent may provide authentication services only for a FireWall-1 Gateway running on the specified IP address. You can specify up to three IP addresses.

Options Tab

FIGURE 5-5 Configuration window — Options tab



The **Options** tab of the **Configuration** window (FIGURE 5-5) enables you to specify whether to allow clear passwords and resolve addresses.

Pre-configuring the Session Authentication agent Before Distribution

The PRODUCT.INI file in the DESKTOP PRODUCTS\SESSIONAGENT directory enables you to pre-configure the Session Authentication agent. This feature is useful if you plan to distribute the Session Authentication agent to many users and you do not want them to configure the agent themselves.

The file is in the standard .INI format. It is divided into sections, each of which consists of a list of parameters and their values:

```
[Startup]
AppName=CheckPoint Session Authentication NG
FreeDiskSpace=0
EnableLangDlg=Y
[ISUPDATE]
UpdateURL=
[Languages]
Default=0x0009
count=1
key0=0x0009
```

The Session Authentication agent for Windows included with FireWall-1 provides for password caching and for restricting authentication to specific FireWall-1 Gateways.

Starting the Session Authentication Agent

When you start the Session Authentication agent, it is minimized and its icon appears in the system tray. From this point on, one of two things can happen:

- The user can open the Session Authentication agent and configure it.
- The Session Authentication agent can receive an authentication request from a FireWall-1 Gateway.

Configuring Client Authentication

1. Configure the Users and Groups that are needed for authentication, and install the User Database (see "Creating Users and Groups" on page 121).
2. Edit the **Check Point Gateway** object representing the FireWall-1 Gateway, and in the **Authentication** page, enable the required authentication schemes. The gateway must support all the authentication schemes defined for the users. For example, if some users use VPN-1 & FireWall-1 Password, and others use Radius Authentication, check both these schemes.
3. Define a Client Authentication access rule.
   a. In the **Source** column, right click to select **Add User Access...**, and choose the group.
   b. If you would like to restrict the location of authenticating users: In the **Location** section of the same window, check **Restrict To** and choose the host, group of hosts, network or group of networks from which users can access.
   c. In the **Service** field choose the services you would like to authenticate.
   d. In the **Action** column, choose **Client Auth.**

   TABLE 5-4 shows a typical Client Authentication Rule.

TABLE 5-4 Client Authentication Rule for HTTP and FTP

SOURCE	DESTINATION	VPN	SERVICE	ACTION
Aaska_Users@Any	Alaska_LAN	Any Traffic	HTTP FTP	Client Auth

4. Double click the **Action** column to edit the **Client Authentication Action Properties**. The settings for Requires Sign On and for Sign On Method are described in "Client Authentication" on page 115.
5. Make sure all Client Authentication Rules are placed *above* the Rule that prevents direct connections to the FireWall-1 Gateway (the "Stealth Rule"), so that they have access to the FireWall-1 Gateway.
6. If you wish, adjust the **Failed Authentication Attempts** settings for Client Authentication in the **Authentication** page of the **Global Properties**.

7 Install the Security Policy.

Allowing Client Authentication Wait Mode

When using Manual Sign On, and the user authenticates with a Telnet session to port 259 on the Gateway, Wait Mode makes it unnecessary to open a new Telnet session in order to Sign Off and withdraw Client Authentication privileges.

To enable Wait Mode, edit the Check Point Gateway object, and in the **Authentication** page, check **Enable Wait Mode for Client Authentication**.

In Client Authentication Wait Mode, FireWall-1 monitors the Telnet connection to port 259 of the Gateway by pinging the user's host. You should define rules to allow the ping as follows:

1 Allow the echo-request service from the FireWall-1 Gateway to the user's host.

2 Allow the echo-reply service from the user's host to the FireWall-1 Gateway.

Configuring a FireWall-1 Gateway to use a RADIUS Server

1 In SmartDashboard, define a RADIUS Server object. In the **Servers and OPSEC applications** object tree, select **Servers**, right click and select **New Radius...**.

2 Define the FireWall-1 Gateway on the RADIUS server using the Internet Authentication Service. Enter a shared secret used by both sides.

3 In SmartDashboard, define a user, and in the Authentication tab select RADIUS.

4 Define the user on the RADIUS server using Active Directory.

CHAPTER 6

Securing Voice Over IP (VoIP)

In This Chapter

The Need to Secure Voice Over IP

Many organizations are using IP connectivity over the Internet between different branches of the company to carry not only data but voice and video. This eliminates the costs of long distance calls using traditional telephony. IP connectivity can also be used for video conferences and for other uses that can lead to significant cost savings for an organization.

Voice and video traffic, like any other information on the corporate IP network, has to be protected as it enters and leaves the organization, and encrypted as it is transmitted through a VPN connection. Possible threats to this traffic are

- Call redirections, where calls intended for the receiver are redirected to someone else.
- Stealing calls, where the caller pretends to be someone else.

Check Point Solution for Secure VoIP

In This Section

Introduction to the Check Point Solution for Secure VoIP

FireWall-1 secures VoIP traffic in H.323 and SIP-based environments.

Both H.323 and SIP calls involve a whole series of complex protocols, each of which can carry potentially threatening information through many ports.

FireWall-1 makes sure that the caller and receiver are who they are supposed to be by examining the source and destination of the packet. In addition, FireWall-1 examines the contents of the packets passing through every allowed port, to make sure they contain proper information. Full stateful inspection on H323 and SIP commands ensures that H.323 or SIP packets are structurally valid, and that they arrive in a valid sequence.

For SIP, FireWall-1 ensures packets fully conform to RFC 3261 (other than SIP over TCP/IP), and inspects SIP-based Instant Messaging protocols. It can successfully protect against Denial of Service attacks, and against penetration attempts such as connection hijacking and connection manipulation.

VoIP in the Security Rule Base

To allow VoIP conversations you need only create rules to allow the SIP or H.323 control signals through the FireWall-1 Gateway. There is no need to create rules for the media. Given a particular VoIP rule, FireWall-1 always opens RTP/RTCP ports automatically to allow the endpoint to endpoint media stream.

There is no need to define a rule for which ports to open and which endpoints will talk. FireWall-1 derives this information from the signalling.

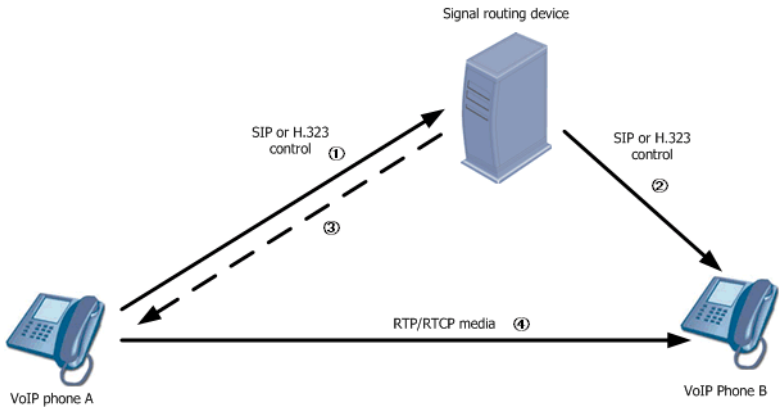
How FireWall-1 Enforces Handover

A VoIP conversation is made up of separate control and media streams. The media is the actual voice and video data, and always passes endpoint to endpoint over the RTP/RTCP protocol. In Proxy mode, the control signals are handed over to the destination endpoint by the Proxy. In Peer to Peer mode, the control and media streams are sent endpoint to endpoint.

For VoIP, FireWall-1 can enforce unidirectional handover. Enforcing handover is an important aspect of VoIP security. It prevents the opening of media stream “holes” in the firewall, and therefore prevents free phone calls being made.

To understand how FireWall-1 enforces handover, consider a conversation that VoIP terminal A initiates with VoIP Terminal B in Proxy mode (FIGURE 6-1).

FIGURE 6-1 How FireWall-1 Enforces Handover



A is on the trusted side of the firewall and the Proxy and B are on the untrusted side.

1. The control signals pass between A and the Proxy.
2. The Proxy hands over the SIP invitation to B.
3. The Proxy returns to A the IP address and destination port of B.
4. A sends the media directly to B, endpoint to endpoint.

FireWall-1 ensures that the address of B that the Proxy returns to A (step 3) is in the handover domain of the Proxy. in this way, FireWall-1 can ensure that the signal and media streams reach B alone. If the Proxy were allowed to handover the call without restriction, and if it were able to open a port back through the firewall, unwanted outside callers would be able to get in through the firewall.

VoIP Domain Objects

For SIP, FireWall-1 enforces unidirectional handover by Proxies, Redirect Servers and Registrars.

For H.323 FireWall-1 enforces unidirectional handover by Gateways and Gatekeepers.

The set of allowed handover locations is called a *VoIP Domain* in the Security Rule Base. A VoIP Domain will typically be a network or group of networks. The source will be allowed to initiate conversations to any destination that is in the VoIP Domain.

If the proxy elements are on a machine with the same IP address, only one VoIP Domain need be defined. If the elements have different IP addresses, define a VoIP for each one.

VoIP Logging

If VoIP logging is turned on in the Global Properties, SIP and H.323 events are logged in the SmartView Tracker. There are Log fields for

- Call registration event. For SIP, the Reg. IP-phones field shows the SIP URL (for example, `example@checkpoint.com`). For H.323 this field shows the phone number (`#1234`, for example)
- Call setup events (Source and Destination IP Phone).
- Media Type (audio, video, instant messaging, applications, unknown) flowing between the source and destination IP Phones.

The Information field shows messages such as `H.323 Message: H.225 Setup Messge`

There is also a predefined Voice Over IP log query.

If VoIP logging is not turned on, only standard logging will take place, showing the source, destination, protocol and so on.

Securing SIP Based VoIP

In This Section

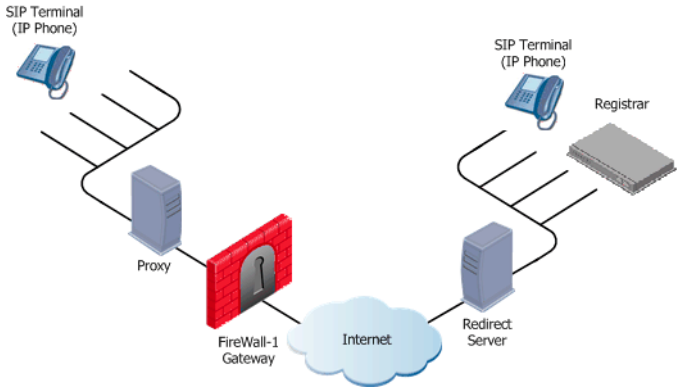
Supported SIP Architectural Elements

SIP has the following architectural elements, all of which are supported by FireWall-1:

- SIP Terminal (IP Phone)—Supports real-time, two way communication with another SIP entity. It supports both signalling (that is, the SIP commands themselves) and media. In SIP, only IP phones can be used. IP Phones can be "soft phones", or computers with appropriate software.
- Proxy—Manages a number of IP phones (or soft phones). Contacts one or more clients or next-hop servers and passes the call request further.
- Redirect Server—performs DNS-like functions preceding a VoIP connection. It accepts SIP requests, maps the SIP URL address into zero or more new addresses, and returns those addresses to the client. It does not initiate requests or accept calls.
- Registrar—Provides information about a caller's possible location.

FIGURE 6-2 shows the SIP architectural elements. It does not necessarily represent an actual topology.

FIGURE 6-2 SIP Architectural Elements



The Proxy, Redirect Server and Registrar can be on the same machine or on different machines, and they can be on the protected or the unprotected side of the FireWall-1 Gateway, or in a DMZ.

To secure a SIP VoIP network with FireWall-1, network objects are defined to represent one or more of the architectural elements. There is normally no need to define a network object for an individual IP Phone. Usually a network of IP Phones is represented by an object that represents a whole network.

SIP Topologies

FireWall-1 supports all SIP topologies. Typical SIP topologies include

- Peer to Peer—where both signalling and media pass peer to peer.

- One or more proxies—where signalling passes through one or more proxies. Once the call has been set up, the media passes peer to peer.

Note - In this and all subsequent discussions, when referring to a *Proxy*, the meaning is to a *Proxy*, *Redirect Server* or *Registrar*.

SmartDefense Application Intelligence for SIP

SmartDefense Application Intelligence can perform content security checks for SIP VoIP connections. The administrator can add these checks when ordinary access control does not provide a sufficient level of trust.

Configuration is done in SmartDefense by selecting **Application Intelligence> VoIP> Verify SIP header content**. This performs the following SIP header checks:

- Strict RFC enforcement for header fields
- Header fields length restrictions
- Removal of characters that should not be used for addresses.
- Removal of unknown media types.

NAT Support for SIP

All FireWall-1 features can be used with SIP, with the following restrictions regarding Network Address Translation (NAT):

- Static NAT can be used in both automatic and manual NAT rules.
- Hide NAT can be used in both automatic and manual rules for outgoing VoIP calls. For incoming calls, automatic rules must be used.
- For security reasons, when using Hide NAT for incoming calls, the Destination of the VoIP call in the appropriate rule in the Security Rule Base cannot be *Any*.
- Internal calls cannot be made from the same source to two endpoints, where one endpoint uses NAT (of either kind) and the other does not.
- Bidirectional NAT of VoIP calls is not supported.
- When using NAT, the proxy must resides in the external network in order to enable VPN-1/FireWall-1 to track the registration messages.
- NAT on a proxy is not supported.

ClusterXL Support for SIP

ClusterXL Gateway clusters are supported in High Availability mode. Load Sharing mode is not supported.

Securing H.323-Based VoIP

In This Section

Signaling and Media Protocols for H.323

H.323 calls are made up of signaling and media.

Signalling is handled by the following H.323 protocols:

- RAS manages registration, admission and status. Uses fixed ports.
- Q.931 manages call setup and termination. Uses fixed ports.
- H.245 negotiates channel usage and capabilities. Uses dynamically assigned ports.

Media streams are transported over RTP/RTCP. RTP carries the actual media and RTCP carries status and control information. It uses dynamic ports.

Supported H.323 Architectural Elements

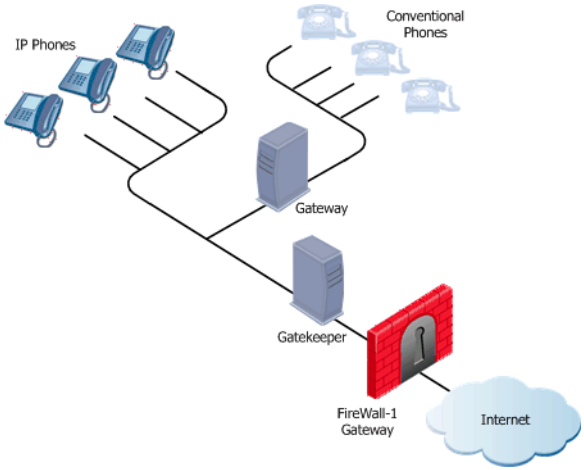
- A *Gatekeeper* manages a collection of H.323 devices such as phones. It converts phone numbers to IP addresses.
- A *Gateway* provides interoperability between different networks. It translates between the telephony protocol and IP.

Both Gatekeepers and gateways are defined in SmartDashboard as a “virtual machine” hosted on a real machine, which is defined as a host node (see FIGURE 6-3).

- IP phones - Either IP telephones or “soft phones” (computers with the appropriate software). They connect to an H.323 gatekeeper.
- conventional telephones - These are not IP devices, and are connected to an H.323 gateway.

IP Phones must be defined in SmartDashboard, but conventional phones are not.

FIGURE 6-3 Supported H.323-based architectural elements



Gatekeeper and Gateway Call Routing

As an H.323 call is processed by a Gatekeeper, RAS, Q.931 and H.245 signaling protocols are used in sequence, and then the media passes. To end a call, the signaling protocols are used in reverse order: H.245, Q.931 and then RAS.

The Gateway only uses RAS for Gateway to Gatekeeper communication. As an H.323 call is processed by a Gateway, Q.931 and H.245 signaling protocols are used in sequence, and then the media passes. To end a call, first H.245 is used, and then Q.931.

The RTP/RTCP media always passes endpoint to endpoint, but for the signaling, it is possible to define which part of the H.323 session should be between the Gatekeepers or Gateways, and which part between the endpoints. The protocol that starts the call on a Gatekeeper is RAS, so that RAS must always pass through the Gatekeeper. The protocol that starts the call on a Gateway is Q.931, so that Q.931 must always pass through the Gateway.

FireWall-1 can be configured to allow one or more of the following routing modes. At least one of the routing modes *must* be selected. If FireWall-1 is configured to allow all routing modes, the Gatekeeper/Gateway is free to decide on the routing mode is use. The Gatekeeper/Gateway is then able to decide which routing mode to use based on the network topology.

- **Direct** —For Gatekeepers only. Only the RAS signals pass through the Gatekeeper. All other signalling (Q.931 and H.245) as well as the RTP/RTCP media passes directly endpoint to endpoint.
- **Call Setup (Q.931)** — RAS (used only by Gatekeepers) and Q.931 pass through the Gatekeeper/Gateway. H.245 and the RTP/RTCP media pass endpoint to endpoint.

- **Call Setup (Q.931) and Call Control (H.245)** — RAS (for a Gatekeeper only), Q.931 and H.245 pass through the Gatekeeper/Gateway. Only the RTP/RTCP media passes endpoint to endpoint. Forcing H.245 to pass through the Gatekeeper/Gateway ensures that FireWall-1 can log the type of media being passed (video or audio).

Supported H.323 Topologies

FireWall-1 supports three H.323 Topologies:

- Gatekeeper to Gatekeeper— each Gatekeeper controls a separate endpoint domain (see FIGURE 6-4).
- Gatekeeper in the DMZ—the same Gatekeeper controls both endpoint domains. This topology makes it possible to provide Gatekeeper services to other organizations (see FIGURE 6-5).
- Endpoint to Endpoint —the IP Phones communicate directly, without a Gatekeeper or a Gateway (see FIGURE 6-6).

FIGURE 6-4 H.323 Topology - Gatekeeper to Gatekeeper

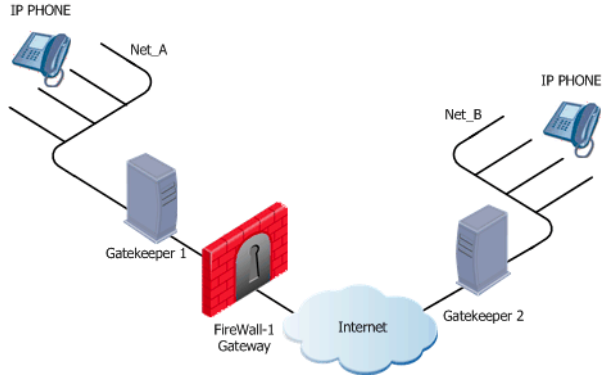


FIGURE 6-5 H.323 Topology - Gatekeeper in the DMZ

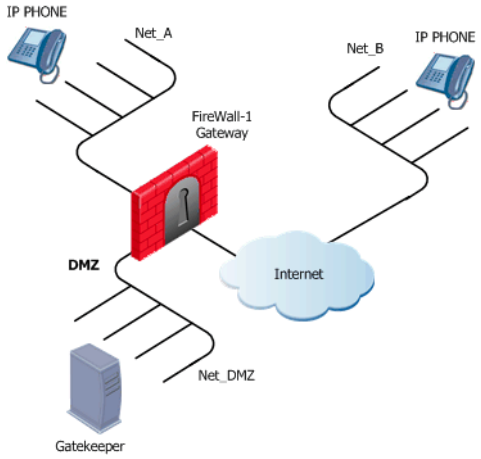
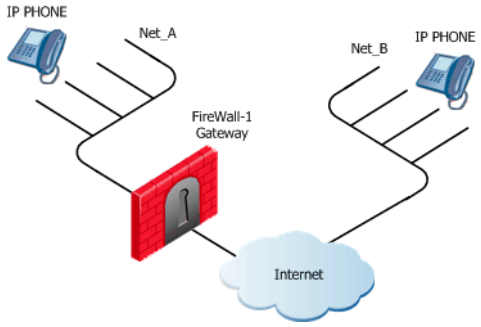


FIGURE 6-6 H.323 Topology- Direct endpoint-to-endpoint communication



H.323 Services

- `H.323_ras_only` — Only RAS. Use for call registration only. Cannot be used to `make calls`.
- `H.323_ras` — Allows a RAS port to be opened, followed by a Q.931 port. Q.931 then opens a H.245 port, which in turn opens ports for RTP/RTCP.
- `H.323` — Allows a Q.931 to be opened, followed by a H.245 port, which in turn opens ports for RTP/RTCP.
- `H.323_any` — Like the H.323 service, but also allows the Source and Destination in the rule to be ANY rather than a network object. Only use H.323_any if you do not know the network topology, for example, if supplying Gatekeeper services.

NAT and ClusterXL Support for H.323

- H.323 NAT (both hide and static mode) is only supported in direct (peer to peer) mode.

- ClusterXL gateway clusters are supported in both High Availability and Load Sharing modes.

Considerations for Secure SIP-Based VoIP

In This Section

Call Direction: Incoming and Outgoing calls

Each Rule allows calls in one direction, either incoming or outgoing. To allow both incoming and outgoing calls you need to create Rule pairs. For example, the following rule allows calls to be made from IP phones in Net_A to phones in Net_B.

Source	Destination	Service	Action	Comment
Net_A	Net_B	SIP	Accept	Outgoing SIP calls

To allow IP phones in Net_B to call Net_A, add the reverse Source and Destination to the rule, as follows:

Source	Destination	Service	Action	Comment
Net_A Net_B	Net_B Net_A	SIP	Accept	Incoming and outgoing SIP calls

Know Your Network Topology

To create the rule, you need to know the location of the VoIP machine (Gateway, Gatekeeper, Proxy) relative to the firewall. For example, for outgoing calls, if the VoIP machine is on the trusted side of the firewall, then the VOIP machine is in the Source of the Rule. If the VoIP machine is behind an untrusted interface (in the DMZ or in the Internet), then it will be the Destination.

Enforcing Handover

Enforcing unidirectional handover using a VoIP Domain adds security. However, It is sometimes not possible to define a VoIP domain for a SIP Proxy. The Proxy may be maintained by an external carrier, so that you do not know which machines the Proxy

controls. It may also be the case that the Proxy is trusted. In these cases it is either impossible or unnecessary to enforce the handover by the Proxy, and there is no need to define a VoIP Domain.

Two predefined SIP services are available: *sip* and *sip_Any.*

- To enforce handover, use a VoIP domain in the Source or Destination, and the sip service. Inspection on the source or destination to ensure handover is only performed if the sip service is used.
- If you don't enforce handover, use Any (or a network object) and the sip_Any service. If a VoIP Domain is used with the SIP_Any service, the packet will be dropped.

The following rule shows handover enforced. Domain {Net_B} is the VoIP Domain containing location B.

Source	Destination	Service	Action
Net_A	VoIP Domain {Net_B}	sip	Accept

If your organization maintains a SIP Proxy on the trusted side of the FireWall-1 Gateway, and you wish to allow outgoing calls to locations that uses some other Proxy, consider enforcing handover for inbound traffic only, as follows:

Source	Destination	Service	Action	Comment
Net_A	Any	sip_Any	Accept	Outgoing SIP calls
Any	VoIP Domain {Net_A)	sip	Accept	Incoming SIP calls with handover enforcement

Configuring SIP-Based VoIP

In This Section

Note - All references to a Proxy in this section also apply to a Redirect Server or a Registrar. If these elements are on a machine with the same IP address, define only one VoIP Domain. If the elements have different IP addresses, define a VoIP Domain for each IP address.

Basic Configuration of SIP-Based VoIP

To secure VoIP, the general configuration procedure is as follows:

1. Define the Network Objects (Nodes or Networks) for the IP Phones that will be managed by the VoIP SIP Proxy.
2. Define the Group Object for the VoIP endpoint domain. This is a group all the Network Objects defined in step 1.
3. Define the object for the machine on which the VOIP gateway is installed.
4. Define the VoIP Domain Object.

   From the SmartDashboard menu, select **Manage>Network Objects>New...>VoIP Domains>VoIP Domain SIP**. Give the Domain object a **Name**. For the **Related endpoints donain** choose the Group Object defined in step 2. For the **VoIP Gateway installed at** choose the Node object defined in step 3.
5. Define the VoIP Rule(s)) that are appropriate for the topology. See the scenarios in the following sections.
6. Make sure that calls using a proxy or a redirect server are allowed in the Global Properties, and set the other Global Properties appropriately.
7. Install the Security Policy.

SIP Rules for an Endpoint to Endpoint (no-Proxy) Topology

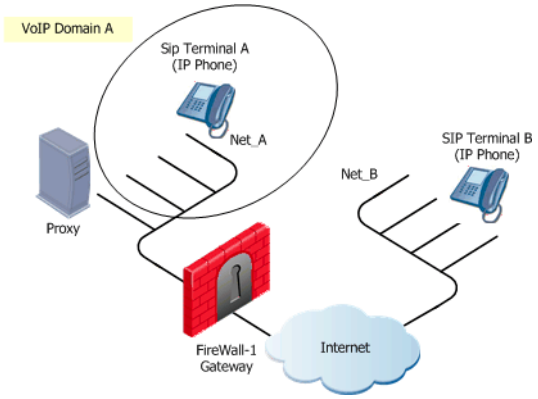
For peer to peer topology, define the following rule:

Source	Destination	Service	Action
network object	network object	sip *or* sip_any	Accept

SIP Rules for a Proxy Topology

A typical topology with a SIP proxy is shown in FIGURE 6-7:

FIGURE 6-7 Typical Proxy Topology



To allow calls from the VoIP Domain of the Proxy, define the following rule:

Source	Destination	Service	Action	Comment
VoIP Domain {Alaska_LAN}	Florida_LAN	sip	Accept	Outgoing SIP. Handover enforced

If you do not wish to enforce handover, define the following rule:

Source	Destination	Service	Action	Comment
Alaska_LAN	Florida_LAN	sip_any	Accept	Outgoing SIP. No handover enforcement

SIP Rule for Internal Network Access

Where the end user IP Phones are behind the firewall, and the SIP Registrar and Proxy are on the other side of the firewall, following rule allows conversations between all the internal phones.

Source	Destination	Service	Action
All internal	VoIP Domain	sip *or* sip_any	Accept

There is no need for an explicit rule permitting control connections between the Registrar and the IP phones. This is because in this case all registration events cross the firewall, so the firewall tracks the registration of the SIP entities and automatically opens

the appropriate ports between the Proxy and SIP participants. This allows SIP control connection between the Proxy and the end user IP phones, so no explicit rule between them is needed.

Hidden SIP Properties

The following SIP-related properties cannot be changed from SmartDashboard. Default values are indicated in **bold**.

- `fw_sip_allow_mcast (true,` **`false`**`)` — Allows multicast RTP traffic. This is a per-module property. To change the value, run the following command on the module:
  `fw ctl set int  fw_sip_allow_mcast`
- `sip_accept_unknown_messages (true,` **`false`**`)` — Accept unknown SIP messages. This is a global property. To change the value, use the dbedit command line or the graphical Database Tool.

Troubleshooting SIP

To see a list of all the online IP phones, you can view the list of phones that the FireWall-1 notes as having registered. Run the command

```
fw tab -t sip_registration -f
```

The output of this command is a list with the format

```
username; IP address
```

To obtain a lot of useful information about the current SIP calls, run the following command:

```
fw tab -t sip_state -f
```

The output of the command lists all the calls with the following information for each:

- Control connection (source, destination).
- RTP connection (endpoint IP addresses).
- Call state (established, ended, registration).
- Media type (audio, video, audio/video, application).
- Number of reinvites (number of participants in a conference call).

Configuring H.323-Based VoIP

In This Section

Basic Configuration of H.323-Based VoIP

To secure VoIP, the general configuration procedure is as follows:

1. Define the Network Objects (Nodes or Networks) for the IP Phones that will be managed by the VoIP H.323 Gatekeeper or Gateway.
2. Define the Group Object for the VoIP endpoint domain. This is a group all the Network Objects defined in step 1.
3. Define the object for the machine on which the VOIP gateway is installed.
4. Define the VoIP Domain Object, either a Gatekeeper or a Gateway.

   From the SmartDashboard Network Objects tree, right click and select **New...>VoIP Domains>VoIP Domain H.323 Gatekeeper** or **VoIP Domain H.323 Gateway**.
5. Give the Domain object a **Name**. For the **Related endpoints donain** choose the Group Object defined in step 2. For the **VoIP Gateway installed at** choose the Node object defined in step 3.
6. In the **Routing Mode** tab, define permitted routing modes for the Gateways and Gatekeepers. If in doubt, select all the options. It is important to select at least one option.
7. In the **Global Properties** window, **FireWall-1 > VoIP** page, make any required changes.
8. Define the VoIP Rule(s) that are appropriate for the topology. See the scenarios in the following sections.
9. It is recommended to make the timeout of the H.323 services identical to the Gatekeeper registration timeout. Configure the timeouts in the **Advanced Properties** window of the Service object.
10. Make sure that call redirection is allowed in the Global Properties, and set the other Global Properties appropriately.

11 Install the Security Policy.

H.323 Rules for a Gatekeeper to Gatekeeper Topology

For H.323-based VoIP where a Gatekeeper is installed in the VoIP domain, define Security Rule Base rules either with or without handover domain, as follows:

TABLE 6-1 H.323 VoIP rules with a handover domain

Source	Destination	Service	Action
VoIP Domain {Net_A} VoIP Domain Net_B}	VoIP Domain {Net_A} VoIP Domain {Net_A}	H323_ras	Accept

This rule allows any host in Net_A to initiate H.323 call to any host in Net_B, and vice versa,

TABLE 6-2 H.323 VoIP rules without a handover domain

Source	Destination	Service	Action
Host {Net_A}	Host {Net_A}	H323_ras_only	Accept
Endpoint domain {Net_A}	Endpoint domain {Net_B}	H323	Accept

Host is the machine on which the VoIP Domain Gatekeeper is installed. The rules without a handover domain are more restrictive than the rule with a handover domain, allowing only specific Gatekeeper to initiate the H.323 session. The second rule in this rule set allows the conversation since the H323_ras_only service includes only the RAS part of H.323 (without handover).

H.323 Rules for a Gatekeeper in DMZ Topology

Starting NG with Application Intelligence, FireWall-1 supports H.323-based VoIP where a Gatekeeper is installed in the DMZ. In this case, one can define rules either with or without a handover domain, as follows:

TABLE 6-3 H.323 rules with a handover domain, for a Gatekeeper in the DMZ

Source	Destination	Service	Action
Endpoint domain {Net_A, Net_B}	VoIP Domain {Net_A, Net_B}	H323_ras	Accept

This rule allows any host in Net_A to initiate H.323 call to any host in Net_B, and vice versa. To allow calls in one direction only, place the network objects that represent the originators of the call in the source, and place the network objects that represent the receivers of the call in the destination.

TABLE 6-4 H.323 rules without a handover domain, for a Gatekeeper in the DMZ

Source	Destination	Service	Action
Endpoint domain X	Gatekeeper Host	H323_ras_only	Accept
Endpoint domain X	Endpoint domain Y	H323	Accept

These rules are unidirectional. To allow bidirectional calls, include both endpoint domains in the source and destination. Where using a VoIP Domain, include both endpoint domains in the VoIP domain.

H.323 Rule for an Endpoint to Endpoint Topology

For peer to peer topology, define the following rules:

Source	Destination	Service	Action
network object	network object	H.323	Accept

If the Source or Destination is *Any*, use the H.323_any service.

Hidden H.323 Properties

The following H.323-related global properties cannot be changed from SmartDashboard. Change all these properties using the `dbedit` command line or the graphical Database Tool. Default values are indicated in **bold**.

- `allow_h323_h245_tunneling (true,` **false**`)`— Allow H.245 tunneling through H.225 connections
- `allow_h323_through_ras (`**true**`, false)` — Allow all H.225 connections thorough RAS.
- `h323_t120_timeout (`**3600**`)` — Define T.120 timeout values. Value is in seconds.

CHAPTER 7

FireWall-1 Advanced Configuration

In This Chapter

Network Address Translation Advanced Configuration

In This Section

Allowing Connections Between Translated Objects on Different FireWall-1 Gateway Interfaces

The goal is to allow connections in both directions between statically translated objects (nodes, networks or address ranges) on different FireWall-1 gateway interfaces.

If NAT is defined via the network object (as opposed to using Manual NAT Rules), then you will need to ensure that Bidirectional NAT is enabled.

Enabling Communication for Internal Networks with Overlapping (or partially overlapping) IP addresses

Overview

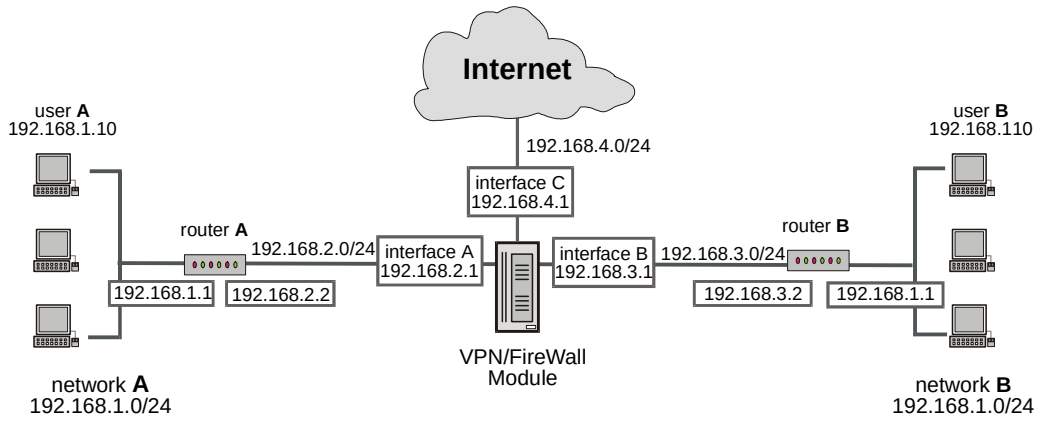
Where two internal networks have overlapping (or partially overlapping) IP addresses, VPN-1/FireWall-1, makes it possible to:

- enable communications between the overlapping internal networks
- enable communications between the overlapping internal networks and the outside world
- enforce a different Security Policy for each of the overlapping internal networks, if desired.

Network Configuration

The network shown in FIGURE 7-1 will be used as an example.

FIGURE 7-1 Example — Class C network



Both network A and network B share the same address space (192.168.1.0/24), so the standard VPN-1/FireWall-1 NAT cannot be used to enable communications between network A and network B. Instead, overlapping NAT must be performed on a per-interface basis.

Users in network A who wish to communicate with users in network B will use the 192.168.30.0/24 network as a destination. Users in network B who wish to communicate with users in network A will use the 192.168.20.0/24 network as a destination.

The VPN/FireWall Module will translate the IP addresses differently on each interface, as follows:

interface A

- inbound source IP addresses will be translated to virtual network 192.168.20.0/24
- outbound destination IP addresses will be translated to network 192.168.1.0/24

interface B

- inbound source IP addresses will be translated to network 192.168.30.0/24
- outbound destination IP addresses will be translated to network 192.168.1.0/24

interface C

Overlapping NAT will not be configured for this interface. Instead, use NAT Hide in the usual way (not on a per-interface basis) to hide source addresses behind the interfaces's IP address (192.168.4.1).

Communication Example

Suppose you wish to allow communication between internal networks and between an internal network and the Internet, as follows:

Between Internal Networks

Suppose user A at IP address 192.168.1.10 in network A wishes to connects to user B at IP address 192.168.1.10 (the same IP address) in network B. User A opens a connection to IP address 192.168.30.10.

step	source IP address	destination IP address
interface A — before NAT	192.168.1.10	192.168.30.10
interface A — after NAT	192.168.20.10	192.168.30.10
VPN/FireWall Module enforces Security Policy for packets from network 192.168.20.0/24 to network 192.168.30.0/24.		
interface B — before NAT	192.168.20.10	192.168.30.10
interface B — after NAT	192.168.20.10	192.168.1.10

Between an Internal Network and the Internet

Suppose user A at IP address 192.168.1.10 in network A connects to IP address 10.10.10.10 on the Internet.

step	source IP address	destination IP address
interface A — before NAT	192.168.1.10	10.10.10.10
interface A — after NAT	192.168.20.10	10.10.10.10

step	source IP address	destination IP address
VPN/FireWall Module enforces Security Policy for packets from network 192.168.20.0/24 to the Internet.		
interface C — before NAT	192.168.20.10	10.10.10.10
interface C — after NAT Hide	192.168.4.1	10.10.10.10

Routing Consideration

In order to allow routing from network A to network B, routing needs to be configured on the firewall machine. The following examples are for Windows and Linux. For other Operating Systems, use the equivalent commands:

On Windows

route add 192.168.30.0 mask 255.255.255.0 192.168.3.2

route add 192.168.20.0 mask 255.255.255.0 192.168.2.2

On Linux

route add -net 192.168.30.0/24 gw 192.168.3.2

route add -net 192.168.20.0/24 gw 192.168.2.2

VPN-1/FireWall-1 Object Database Configuration

To implement the overlapping NAT feature, use the `dbedit` database editor GUI (or command line utility).

In the example configuration, you would set the per interface values for interface A and interface B as follows:

parameter	value
`enable_overlapping_nat`	`true`
`overlap_nat_dst_ipaddr`	The overlapping IP addresses (before NAT). In the example configuration, this would be 192.168.1.0 for both interfaces.
`overlap_nat_src_ipaddr`	The IP addresses after NAT. In the example configuration, this would be 192.168.20.0 for interface A, and 192.168.30.0 for interface B.
`overlap_nat_netmask`	The net mask of the overlapping IP addresses. In the example, 255.255.255.0.

Management Behind NAT

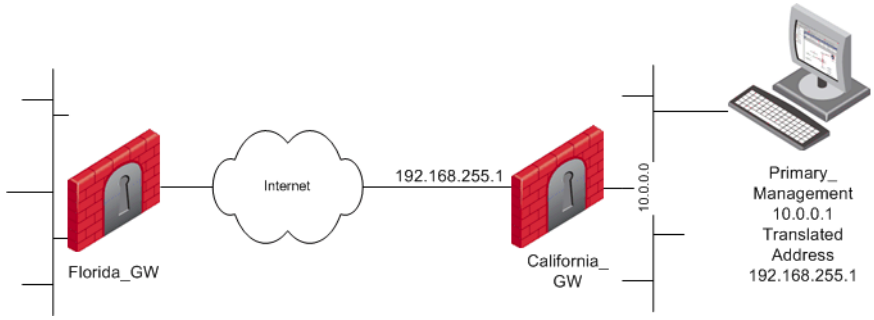
The SmartCenter server sometimes uses a private IP address (as listed in RFC 1918), or some other non routable IP address. Using private addresses for the internal networks has become common, mainly because the lack of IP addresses.

Network Address Translation for the SmartCenter Server IP address is easy to configure. Static or Hide NAT on the SmartCenter Server address can be configured in one click, while still allowing connectivity with managed enforcement modules. All enforcement modules can be controlled from the SmartCenter Server, and logs can be sent to the SmartCenter Server.

Network Address Translation can be configured for the SmartCenter Server, Management High Availability, and the Log Server.

FIGURE 7-2 shows a typical scenario. The SmartCenter Server is in a network on which Network Address Translation is performed (the "NATed network"). The SmartCenter Server is able to control Check Point enforcement modules inside the NATed network, on the border between the NATed network and the outside world, and outside the NATed network.

FIGURE 7-2 Typical configuration with NAT for the SmartCenter



In ordinary Hide NAT configurations, no connections can be established from external side the FireWall-1 NAT gateway. In contrast, when using Hide NAT on the SmartCenter Server, enforcement modules are able to send logs to the SmartCenter Server.

When an inbound connection from a managed module comes in to the FireWall-1 gateway, port mapping is used to translate from the hiding address to the real IP address of the SmartCenter Server.

NAT for the SmartCenter Server is enabled in the **NAT** page of the SmartCenter Server object by defining NAT and selecting **Apply for VPN-1 & FireWall-1 control connections**.

Note that:

- Only one object can be defined with these settings, unless the second object is defined as a Secondary SmartCenter Server or a Log Server.
- It is important to properly define the Topology settings on all enforcement modules. In FIGURE 7-2 for example, on California_GW, you must define the Primary_Management on its internal interface.
- All managed modules, and the SmartCenter Server must be of version NG with Application Intelligence and above. Otherwise you must use a workaround that involves defining a dummy object (see SecureKnowledge Solution SK15558).
- In previous versions, various workarounds were required. All previous workarounds will continue to work, with no changes in behavior.

Configuring the SmartCenter Server Object

1. On the Primary_Management object, In the **NAT** page, choose either *Static NAT* or *Hide NAT*.

   If using Hide NAT, select **Hide behind IP Address** (for example, 192.168.55.1). Do not **Hide behind Gateway** (address 0.0.0.0).

2. Install on the Gateway that protects the NATed objects or network. Do not select **All**. In FIGURE 7-2, **Install on Gateway:** California_GW.

3. Check **Apply for VPN-1 and FireWall-1 control connections**.

Configuring the Enforcement Module Object

California_GW must know that Primary_Management is behind it. In the California_GW **Topology** page, define:

- Interface Eth3

In the **General** tab of the **Interface Properties** window of this interface:

- **IP Address** 10.0.0.0
- **Netmask** 255.255.0.0

In the **Topology** tab of the **Interface Properties** window of this interface:

- **Network defined by the interface IP and Net Mask**.

Configuring Pre-NG with Application Intelligence Enforcement Module Objects

For managed modules that are not of version NG with Application Intelligence, you must define a dummy object. Referring to FIGURE 7-2, if Florida_GW and California_GW have a version lower than NG with Application Intelligence, the dummy objects ensure that

- Florida_GW knows that its SmartCenter Server has the address 192.168.255.1.

- California_GW knows that its SmartCenter Server has the address 10.0.0.1.

Proceed as follows:

Define a dummy object with the translated address of the Primary_Management:

1. Give it a **Name** (Dummy-Management)
2. In the **General Properties** page, in the **Check Point Products** section, select **Secondary Management Station** and **Log Server**.

Define a dummy object for the California_GW object:

1. Give it a **Name**.
2. Give it the **IP Address** 192.168.255.1.
3. Give it the address of the Primary Management NAT definition
4. In the **General Properties** page, in the **Check Point Products** section, select **Secondary Management Station** and **Log Server**.
5. In the **Logs and Masters** page:
   - Define Dummy-Management as a Master
   - Define Dummy-Management as a Log Server (if the log server is on a separate machine, define two virtual objects).

Content Security Advanced Configuration

CVP Chaining and Load Sharing

In This Section

Introduction to CVP Chaining and Load Sharing

Traffic that crosses the FireWall-1 enforcement point can be checked using CVP servers. CVP checking is available for Web, Mail, FTP and TCP traffic. For detailed explanations, see:

- “CVP and Anti-Virus Protection of Web Traffic” on page 75

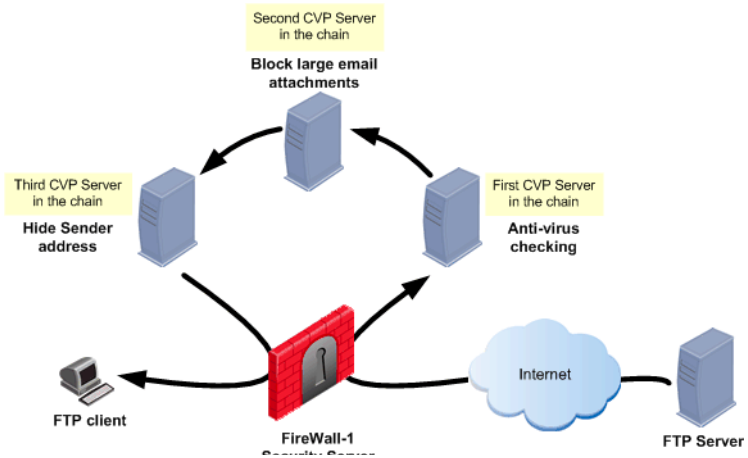
- "Virus checking an FTP connection using CVP" on page 89.

It is possible to chain CVP servers in order to combine functionality, and to perform load sharing between CVP servers, in order to speed up CVP checking.

CVP Chaining

CVP servers can be chained for the purpose of combining functionality. Chaining is useful when each of the CVP servers performs a different task, such as scanning for viruses, or blocking large email attachments. In the configuration shown in FIGURE 7-3, the FireWall-1 Security Server invokes the first, second, and third CVP servers in turn.

FIGURE 7-3 CVP server chain



Chained CVP servers are invoked in the order chosen by the administrator in the CVP Group object. When choosing a chaining order, consider whether there are any security or connectivity issues. For example, in FIGURE 7-3, you may wish the virus scanning to take place first.

The order in which the chained servers are called, is relative to the *response* of the server. This is the case whether the server is on the unprotected (external interface) side of the FireWall-1 enforcement point or on the protected (internal interface) side.

For example, in FIGURE 7-3, consider a user at an internal FTP client who is downloading a file from an external FTP server. CVP checking is done on the response from the FTP server (that is, on the downloaded file) in the order defined in the CVP group object.

There is one exception to this order. The HTTP Security server allows CVP checking to be done on the HTTP *request*. CVP checking of HTTP requests is performed by the CVP servers in the reverse of the order specified in the CVP Group object.

CVP chaining works only if all servers in the chain are available. If one or more of the servers is unavailable, the whole CVP session is dropped. This is because skipping one of the servers may contradict the security policy. For example, the security policy may specify that both virus scanning and blocking of large attachments are mandatory.

CVP Load Sharing

Identical CVP servers can be configured to share the load among themselves. Load sharing can speed up CVP checking by allowing many CVP sessions to run simultaneously on more than one CVP server.

Two load sharing methods are available:

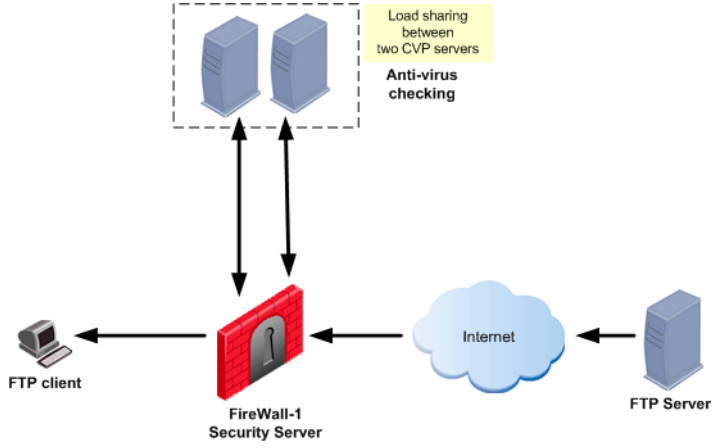
- In the *round robin* method, the FireWall-1 Security Server sends each new CVP session to a different CVP server in turn.
- In the *random* method, the FireWall-1 Security Server sends each new CVP session to a randomly chosen CVP server.

It is possible to configure a load sharing suspension period for a CVP server that does not respond. During that period of time, that CVP server does not take part in the load sharing group.

CVP load sharing is implemented by defining a Resource that invokes a group of CVP servers. The order in round robin mode is configured in the CVP Group object.

FIGURE 7-4 shows two CVP servers that share the load among themselves.

FIGURE 7-4 Load sharing between CVP servers

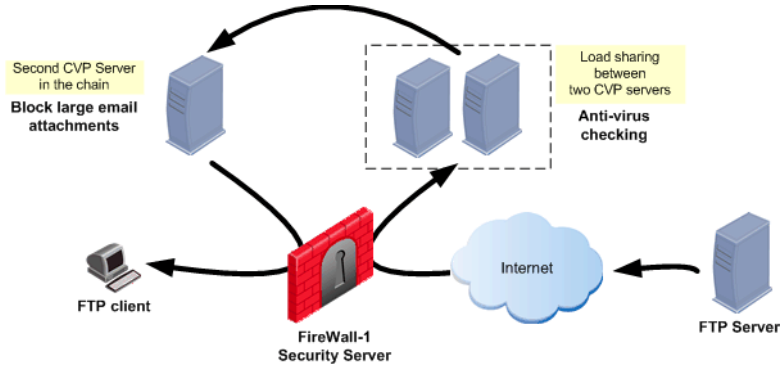


Combining CVP Chaining and Load Sharing

It is possible to combine CVP chaining and load sharing. FIGURE 7-5 shows three CVP servers. Two perform load sharing between themselves, and the load sharing group is chained with another CVP server.

It is possible to put a load sharing group into a CVP chain, but it is not possible to perform load sharing between chained CVP groups

FIGURE 7-5 A chained Load-Sharing CVP server group



Configuring CVP Chaining and Load Sharing

1. For each CVP server, define a CVP Server object.

   To define a CVP Server object, right click in the **Servers and OPSEC Application** tree, and select **New > OPSEC Application...**. In the **OPSEC Application Properties** window, **General** tab, make sure that the selected **Server Entities** include *CVP*.

2. Define a CVP Group object. A CVP Group object contains CVP server objects, and is used in the same way as an OPSEC Application object for a CVP server.

   To define a CVP Group object, right click the **Servers and OPSEC Application** tree, and select **New > CVP Group**.

3. In the **CVP Group Properties** window, add the CVP servers to the group.

4. Choose the **Work distribution method**: Either *Load sharing* or *Chaining*.

5. If you chose *Load sharing*, define the **Load sharing method**, and the **Load sharing suspend timeout**, if any.

6. Create a Resource object. In the **Resources** tree, right click and select one of the following: **New > URI...**, **New > SMTP...**, **New > FTP...**, or **New > TCP...**. Define the content security capabilities.

7. In the **CVP Server** field in the **CVP** tab of the Resource object, select the CVP Group defined in step 2.
8. In the Security Rule Base, define a rule that uses the Resource.
9. Save and install the Security Policy.

CHAPTER A

Security Before VPN-1/FireWall-1 Activation

In This Appendix

Achieving Security Before VPN-1/FireWall-1 Activation

There are several scenarios in which a computer does not yet have a VPN-1/FireWall-1 Security Policy installed, and is vulnerable. Two features provide security during these situations: Boot Security, which secures communication during the boot period, and Initial Policy, which provides security before a Security Policy is installed for the first time. As an outcome, there is no instant of time when the computer is left unprotected.

Boot Security

During the boot process, there is a short period of time (measured in seconds) between the point when the computer is capable of receiving communication (and can be attacked) and the point when the Security Policy is loaded and is enforced. During this time, VPN-1/FireWall-1 Boot Security feature protects both the internal networks behind the VPN-1/FireWall-1 Gateway, and the computer itself. Boot Security is provided by two elements working together:

- Control of IP Forwarding on boot

- The Default Filter

The Default Filter also provides protection in a scenario where VPN-1/FireWall-1 processes are stopped for maintenance.

Control of IP Forwarding on Boot

For networks protected by a VPN-1/FireWall-1 module, protection is available at boot by disabling IP forwarding in the OS kernel. This ensures that there will never be a time when IP Forwarding is active at a time when no Security Policy is enforced. This ensures that networks behind the gateway are safe.

Disabling IP Forwarding protects networks behind the VPN-1/FireWall-1 computer, but it does not protect the VPN-1/FireWall-1 computer itself. For this purpose, VPN-1/FireWall-1 implements a Default Filter during the period of vulnerability.

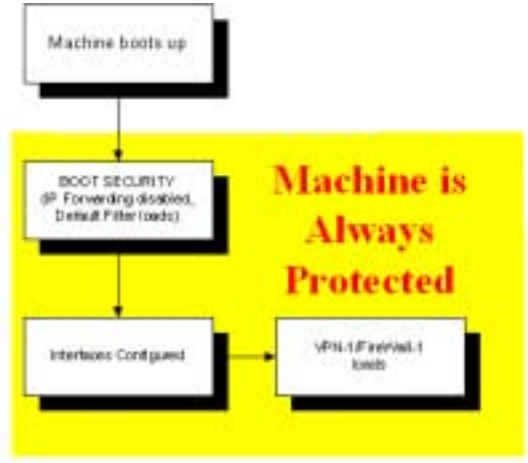
The Default Filter

The sequence of actions for a VPN-1/FireWall-1 Gateway when booting with the Default Filter, is illustrated in FIGURE A-1:

- computer boots up
- Boot security takes effect (Default Filter loads and IP Forwarding is disabled)
- Interfaces are configured
- FireWall-1 services start

The computer is protected as soon as the Default Filter loads.

FIGURE A-1 How a Default Filter Protects the VPN-1/FireWall-1 Gateway computer



There are five Default Filters:

- General - accepts no inbound communication (this is the default option).
- Drop Filter - accepts no inbound or outbound communication. This filter drops all communications into and out of the gateway during a period of vulnerability. Note, however, that if the boot process requires that the gateway communicate with other hosts, then the Drop Filter should not be used.
- Default Filter for IPSO allowing SSH incoming communication to support remote Administration.
- Default Filter for IPSO allowing HTTPS incoming communication to support remote Administration.
- Default Filter for IPSO allowing SSH and HTTPS incoming communication to support remote Administration.

The appropriate Default Filter should be selected based on platform and communication needs. The general filter is selected by default.

The Default Filter also provides anti-spoofing protection for the VPN-1/FireWall-1 Gateway. It makes sure that packets whose source are the VPN-1/FireWall-1 Gateway computer itself have not come from one of its interfaces.

Using the Default Filter for Maintenance

It is possible to stop VPN-1/FireWall-1 processes for maintenance while at the same time protecting the VPN-1/FireWall-1 Gateway and the internal network.

During maintenance, the Default Filter allows open connections to the Gateway to remain open, without dropping them.

The Initial Policy

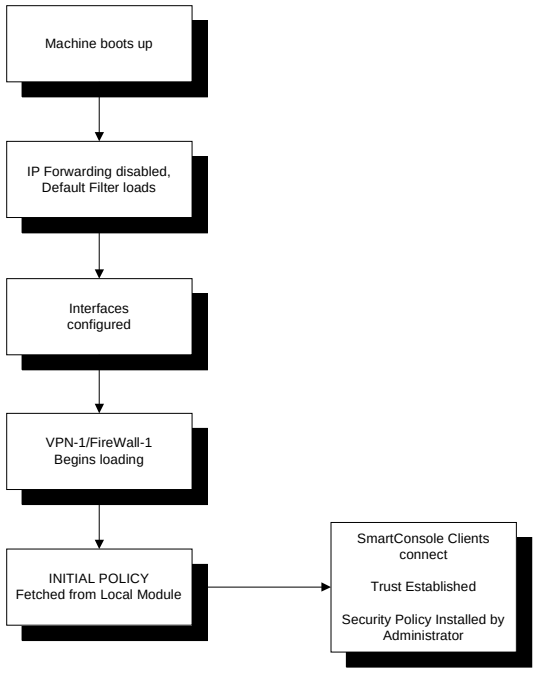
Until the VPN-1/FireWall-1 administrator installs the Security Policy on the Gateway for the first time, security is enforced by an Initial Policy. The Initial Policy operates by adding “implied rules” to the Default Filter. These rules forbid most of the communication but allows the communication needed for the installation of the Security Policy. The Initial Policy also protects a Gateway during Check Point product upgrades, or when a SIC certificate is reset on the module, or in the case of a Check Point product license expiration.

Note - During a Check Point upgrade, a SIC certificate reset, or license expiration, the Initial Policy overwrites the user-defined policy.

The sequence of actions during boot of the VPN-1/FireWall-1 Gateway computer until a Security Policy is loaded for the first time, is illustrated in FIGURE A-2.

- computer boots up
- Default Filter loads and IP Forwarding is disabled
- Interfaces are configured
- FireWall-1 services start
- Initial policy is Fetched from the Local Module
- SmartConsole clients connect or Trust is established, and the Security Policy is installed

FIGURE A-2 Initial Policy- Until the First Policy Installation



The Initial Policy is enforced until a policy is installed, and is never loaded again. In subsequent boots, the regular policy is loaded immediately after the Default Filter.

There are different Initial Policies for stand-alone and distributed setups. In a stand alone configuration, where the SmartCenter Server and VPN-1/FireWall-1 module are on the same computer, the Initial Policy allows CPMI communication only. This permits SmartConsole clients to connect to the SmartCenter Server.

In a distributed configuration, where the Primary SmartCenter Server is on one computer and the VPN-1/FireWall-1 module is on a different computer, the Initial Policy allows the following:

- Primary SmartCenter Server computer — allows CPMI communication for SmartConsole clients.

- VPN-1/FireWall-1 Gateway — allows `cpd` and `fwd` communication for SIC communication (to establish trust) and for Policy installation.

In a distributed configuration, the Initial Policy on the VPN-1/FireWall-1 module does not allow CPMI connections. The SmartConsole will not be able to connect to the SmartCenter Server, if the SmartConsole must access the SmartCenter Server through a Gateway running the Initial Policy.

There is also an Initial Policy for a Secondary SmartCenter Server (Management High Availability). This Initial Policy allows CPMI communication for SmartConsole clients and allows `cpd` and `fwd` communication for SIC communication (to establish trust) and for Policy installation.

Default Filter and Initial Policy Configuration

In This Section

Verifying the Default Filter or Initial Policy is Loaded

You can verify that the Default Filter and/or Initial Policy are indeed loaded as follows:

1. Boot the system.
2. Before installing another Security Policy, type the following command:

```
$FWDIR/bin/fw stat
```

The command's output should show that `defaultfilter` is installed for the Default Filter status. It should show that `InitialPolicy` is installed for the Initial Policy.

Change the Default Filter to a Drop Filter

For a typical setup there are two Default Filters: `defaultfilter.boot` and `defaultfilter.drop`. They are located in `$FWDIR/lib`.

1. Copy over and rename the relevant desired Default Filter Inspect file (`defaultfilter.boot` or `defaultfilter.drop`) to `$FWDIR/conf/defaultfilter.pf`
2. Compile the Default Filter by running the command:
   `fw defaultgen`
   The output will be in `$FWDIR/state/default.bin`
3. Run `fwboot bootconf get_def` to print the Default Filter file path.
4. Copy `default.bin` to the Default Filter file path.
5. If the Security Policy has not yet been installed, run `cpconfig` to regenerate the Initial Policy.

User-Defined Default Filter

For administrators with Inspect knowledge, it is possible to define your own Default Filter as follows:

1. Create an Inspect script named `defaultfilter.pf` in `$FWDIR/conf`.:

Warning - Ensure that the script does not perform any of the following functions:

- Logging
- Authentication
- Encryption
- Content security

2. Continue as from step 2 of "Change the Default Filter to a Drop Filter" on page 165."

You must ensure that your Security Policy does not interfere with the boot process.

Using the Default Filter for Maintenance

It is sometimes necessary to stop VPN-1/FireWall-1 processes for maintenance, and it is impractical to disconnect the VPN-1/FireWall-1 Gateway computer from the network (for example, the computer may be at a remote location). The `cpstop -fwflag -default` and `cpstop -fwflag -proc` commands allow VPN-1/FireWall-1 processes to be temporarily stopped for remote maintenance without exposing the computer to attack.

To Unload a Default Filter or an Initial Policy

To unload a Default Filter or an Initial Policy from the kernel, use the same command that is used for unloading a regular policy. Do this only if you are certain that you do not need the security provided by the Default Filter or an Initial Policy.

- To unload the Default Filter locally, run the `fw unloadlocal` command.
- To unload an Initial Policy from a remote management machine, run the following command on the SmartCenter Server:

  ```
  fwm unload <hostname>
  ```

  where `hostname` is the SIC_name of the Module.

If You Cannot Complete Reboot After Installation

In certain configurations the Default Filter may prevent the VPN-1/FireWall-1 Gateway computer from completing the reboot following installation.

First, examine the Default Filter and verify that the Default Filter allows traffic that the computer need to boot.

If the boot process cannot complete successfully, remove the Default filter as follows:

1. Reboot in **single user** mode (for UNIX) or **Safe Mode With No Networking** (for Windows 2000).
2. Ensure that the Default Filter does not load in future boots. Use the command `fwbootconf bootconf Set_def`
3. Reboot.

Command Line Reference for Default Filter and Initial Policy

control_bootsec

Enables or disables Boot Security. The command affects both the Default Filter and the Initial Policy.

Usage

```
$FWDIR/bin/control_bootsec [-r] [-g]
```

TABLE 2-1 options control_bootsec

Options	Meaning
`-r`	Removes boot security
`-g`	Enables boot security

fwboot bootconf

Use the `fwboot bootconf` command to configure boot security options. This command is located in `$FWDIR/boot`.

Usage

```
$FWDIR/bin/fwboot bootconf  <command> [value]
```

TABLE 2-2 options fwboot bootconf

Options	Meaning
`Get_ipf`	Reports whether VPN-1/FireWall-1 controls IP Forwarding. ▪ Returns 1 if IP Forwarding control is enabled on boot. ▪ Returns 0 if IP Forwarding is not controlled on boot.
`Set_ipf 0/1`	Turns off/on control of IP forwarding for the next boot. `0` - Turns off `1` - Turns on
`Get_def`	Returns the full path to the Default Filter that will be used on boot.
`Set_def <filename>`	Will load `<filename>` as the Default Filter in the next boot. The only safe, and recommended, place to put the `default.bin` file is `$FWDIR\boot`. (The `default.bin` filename is a default name.) **Note** - Do NOT move these files

comp_init_policy

Use the `comp_init_policy` command to generate and load, or to remove, the Initial Policy.

This command generates the Initial Policy. It ensures that it will be loaded when the computer is booted, or any other time that a Policy is fetched, which would also occur at `cpstart`, or with the `fw fetch localhost` command. After running this command, `cpconfig` will add an Initial Policy if there is no previous Policy installed.

Usage

```
$FWDIR/bin/comp_init_policy [-u | -g]
```

TABLE 2-3 options comp_init_policy

Options	Meaning
`-u`	Removes the current Initial Policy, and ensures that it will not be generated in future when `cpconfig` is run.
`-g`	Can be used if there is no Initial Policy. If there is, make sure that after removing the policy, you delete the `$FWDIR\state\local\FW1\` folder. Generates the Initial Policy and ensures that it will be loaded the next time a policy is fetched (at `cpstart`, or at next boot, or via the `fw fetch localhost` command). After running this command, `cpconfig` will add an Initial Policy when needed.

The `comp_init_policy -g` command will only work if there is no previous Policy. If you perform the following commands:

```
comp_init_policy -g + fw fetch localhost
comp_init_policy -g + cpstart
comp_init_policy -g + reboot
```

The original policy will still be loaded.

cpstop -fwflag -default and cpstop -fwflag -proc

To stop all VPN-1/FireWall-1 processes but leave the Default Filter running, use `cpstop -fwflag -default`. To stop all VPN-1/FireWall-1 processes but leave the Security Policy running, use `cpstop -fwflag -proc`.

To stop and start all Check Point processes, use `cpstop` and `cpstart`. These commands should be used with caution.

On Win32 platforms, use the **Services** applet in the **Control Panel** to stop and start Check Point Services

Usage

```
cpstop -fwflag [-default | -proc]
```

TABLE 2-4 Options for fwflag

Options	Meaning
`-default`	Kills VPN-1/FireWall-1 processes (`fwd`, `fwm`, `vpnd`, `snmpd` etc.). Logs, kernel traps, resources, and all security server connections stop working. The Security Policy in the kernel is replaced with the Default Filter.
`-proc`	Kills VPN-1/FireWall-1 processes (`fwd`, `fwm`, `vpnd` etc.). Logs, kernel traps, resources, and all security server connections stop working. The Security Policy remains loaded in the kernel. Therefore allow/reject/drop rules that do not use resources, but only service, continue working.

cprestart

Use this command to stop and immediately restart all Check Point processes.

Usage

```
cprestart
```

Appendix **B**

FireWall-1 Command Line Interface

All command line commands that relate to FireWall-1 are documented in the *Command Line Interface (CLI)* Guide.

Index

N

P

Q

R

S

U

V

W

X