Chapter

6

Common Issues

One of the weaknesses I felt the first edition of this book had was that it did not include enough Frequently Asked Questions (FAQs) of a more general nature, that is, things that might come up in the day-to-day operation of your firewall but didn't neatly fall into other chapters I've written. Since providing answers to FAQs about FireWall-1 is how I got to be well known within the FireWall-1 community in the first place, it seems fitting that I include a chapter in the book that is nothing but FAQs.

The FAQs in this chapter relate to error messages you might see in the operating system logs, on the console, and in SmartView Tracker/Log Viewer. The FAQs also cover other situations that the average firewall administrator needs to resolve that are more general in nature.

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

- Configure your firewall to deal with some common situations
- Diagnose common error messages that occur with your firewall
- Recognize common issues that appear to be firewall-related but are not

Common Configuration Questions

In the course of using or configuring FireWall-1, a number of common configuration questions come up from time to time. The following subsections document the most common ones.

6.1: How Do I Modify FireWall-1 Kernel Variables?

Over the years, Check Point has introduced some rather obscure features by exposing "kernel variables" that can be tweaked to change certain behavior. While this is not the most elegant solution, it involves the least amount of work because it requires no GUI changes. Modifying kernel variables is relatively straightforward once you know how. You perform the appropriate commands for your platform and reboot.

Let us assume that the kernel variable we want to modify is `fw_allow_udp_port0`. For the record, this particular variable allows packets to be sent from or to UDP port 0, which FireWall-1 normally drops. In order to allow these kinds of packets, we need to change the value of this parameter to `1`. The value can be specified in decimal or hexadecimal (precede with an `0x` for hexadecimal).

In general, you can substitute `fw_allow_udp_port0` and `0x1` for the variable you want to modify and the value you wish to assign it, respectively.

On Solaris machines, add the following line to the bottom of the `/etc/system` file, and reboot:

```
set fw:fw_allow_udp_port0=0x1
```

On an IPSO system (VPN-1 Appliance or Nokia IPxxx), you need to get the `modzap` utility from Resolution 1261 in Nokia's Knowledge Base. You can then use the following command line to modify the `fw_allow_udp_port0` parameter and reboot the system:

```
nokia[admin]# modzap _fw_allow_udp_port0
                 $FWDIR/boot/modules/fwmod.o 0x1
```

NOTE! On IPSO, all kernel variables begin with an underscore (_).

On a Linux platform, you simply add the following line to `$FWDIR/boot/modules/fwkern.conf` and restart FireWall-1 (no reboot required):

```
fw_allow_udp_port0=1
```

For Windows, there is no way to modify kernel variables without getting a special utility called `fwpatch` from Check Point support. In some cases, it is possible to tweak registry settings.

6.2: Can I Direct FireWall-1 Log Messages to syslog?

To log specific events to syslog, I use user-defined logging for this. My user-defined program (defined in the Global Properties section, Log and Alert frame) is `/usr/ucb/logger -p daemon.notice`. The path to the `logger` utility varies depending on the operating system.

Another alternative is to log everything to syslog. You can do this with the following command:

```
# fw log -f 2>>/var/adm/fw-log.log | /bin/logger -p \
    local5.info > /dev/null 2>&1 &
```

This command runs in the background and logs everything to syslog. Note that it might be best to put this into a boot script after FireWall-1 loads so that everything is dumped to syslog.

On Windows platforms, instead of `logger`, use the Kiwi SyslogGen program, available from http://www.kiwisyslog.com.

6.3: How Can I Disconnect Connections at a Specific Time?

Active connections stay in the connections tables until they either terminate or expire. The rulebase controls only when connections start, not how long they are allowed to stay connected.

One way to block connections at a specific time is to use the **`fw sam`** command, which is described in Chapter 5. At a specified time, run a command via **`cron`** that blocks all inappropriate traffic and disconnects any active session for a specific period of time. Once the timeout for that command expires (you can set it as low or as high as you want), everything should go through your rulebase normally. The old connections should theoretically be forgotten.

6.4: How Many Interfaces Are Supported?

FireWall-1 NG up to NG FP2 supports 256 interfaces. Versions NG FP3 and above support 1,024 interfaces. However, each IP associated with the platform might get associated with the interface slot, depending on how old a version you are running.

On the Nokia platform, things are a little more complicated, depending on which version of IPSO and FireWall-1 you happen to be using. The following list shows what your interface limit is based on the versions used:

- FireWall-1 NG FP2 or earlier without VLAN hotfix: 64 interfaces
- FireWall-1 NG FP2 with VLAN hotfix: 256 interfaces
- FireWall-1 NG FP3 on IPSO 3.6: 256 interfaces
- IPSO 3.7 or above (with supported FireWall-1 version): 1,024 interfaces

What constitutes an interface varies by platform. GRE tunnels, VLANs, frame relay DLCIs, point-to-point links, permanent virtual circuits, and other similar constructs may be considered interfaces by FireWall-1.

6.5: How Do I Create a Large Number of Objects via the Command Line?

Bulk creation of objects is accomplished through the use of the command-line program **`dbedit`**, which provides a protected interface to the Check Point object database, along with object validation.

The **dbedit** commands used to create a simple network object are listed below. (x.y.z.w is the IP address, a.b.c.d is the netmask, and sample-network is the name of the object.)

```
dbedit> create network sample-network
dbedit> modify network_objects sample-network ipaddr x.y.z.w
dbedit> modify network_objects sample-network netmask a.b.c.d
dbedit> update network_objects sample-network
```

The **create** command is used to bring the object into existence, the **modify** command is used to change elements of that object, and the **update** command is used to push that change to the object database.

To create a simple host object (e.f.g.h is the host object IP), use these commands:

```
dbedit> create host_plain sample-host
dbedit> modify network_objects sample-host ipaddr e.f.g.h
dbedit> update network_objects sample-host
```

To group the objects together, use these commands:

```
dbedit> create network_object_group sample-group
dbedit> addelement network_objects sample-group ''
                   network_objects:sample-network
dbedit> addelement network_objects sample-group ''
                   network_objects:sample-host
dbedit> update network_objects sample-group
```

In the preceding example, the **addelement** command is responsible for adding the objects into the group. Since a group can potentially contain non-network objects, we have to be explicit when we add them to a group, which is why we refer to sample-network and sample-host as `network_objects:sample-network` and `network_objects:sample-host`, respectively, within the code.

You can also create a network object with automatic NAT by using the following commands:

```
dbedit> create host_plain london

dbedit> modify network_objects london ipaddr 192.168.1.1
dbedit> modify network_objects london color red
dbedit> modify network_objects london comments "This is london calling"
dbedit> modify network_objects london add_adtr_rule true
dbedit> modify network_objects london NAT NAT
dbedit> modify network_objects london NAT:valid_ipaddr 195.195.195.3

dbedit> modify network_objects london NAT:netobj_adtr_method adtr_static
dbedit> update network_objects london
```

In the preceding example, if you wanted to do hide mode NAT, replace `adtr_static` with `adtr_hide`.

By putting the appropriate **`dbedit`** commands in a file and invoking **`dbedit`** correctly, you could script the creation of network objects. To automate the process, execute something similar to the following on your management station (`dbeditcmdfile.txt` contains the **`dbedit`** commands).

```
# dbedit -s localhost -u admin -p adminpw -f dbeditcmdfile.txt
```

Common Error Messages in the System Log

One thing there is no shortage of in FireWall-1 is error messages. The following subsections highlight several common errors and what you can do to prevent them.

Several of these FAQs reference HFA-xxx versions. These are called Hotfix Accumulators, something Check Point Support started generating since FireWall-1 NG FP3. They are simply "jumbo hotfixes" that include fixes for a number of issues combined. These fixes can be obtained from Check Point Support, which users with a direct support agreement can do. Companies that provide support for Check Point products can also provide these hotfixes. The same applies for almost any other hotfix mentioned.

6.6: Local Interface Anti-Spoofing

Local interface anti-spoofing is a different sort of anti-spoofing than the one configured in the gateway object for the firewall. FireWall-1 drops any packet it receives with a source IP address of one of the firewall's local interfaces that the firewall did not originate. You might see this if you plug two or more physical interfaces on different logical interfaces into the same hub.

You can disable local interface anti-spoofing by changing the FireWall-1 kernel variable `fw_local_interface_anti_spoofing` to `0`. For more details on how to change FireWall-1 kernel variables, see FAQ 6.1.

6.7: Tried to Open Known Service Port, Port xxxx

The error message "Host tried to open known service port" shows up with services that use multiple ports for their communication. This error is most common with FTP but can also occur with other services. By default, FireWall-1 does not allow services that negotiate data ports to choose a service that is defined in FireWall-1. This check can be disabled by editing `$FWDIR/lib/base.def` on the management console and reinstalling the security policy.

In theory, this check prevents anyone from using the control connection of an allowed service such as FTP to open a service that may not otherwise be

allowed between the client and server. However, this check applies only to predefined services. Someone interested in subverting the firewall in this manner could just as easily choose a service port undefined in FireWall-1 and, instead of using an FTP data connection, do something else through it. Because of this, I do not see this check providing real value, and any value it does have is overshadowed by the fact that it frequently breaks legitimate FTP usage.

In FireWall-1 NG FP1 and above, you can resolve this problem by editing `$FWDIR/lib/base.def` on the management station. Add the following line in the following location (the line to add is set in bold):

```
#ifndef __base_def__
#define __base_def__

#define NO_SERVER_PORT_CHECK

#include "services.def"

//
// (c) Copyright 1993-2001 Check Point Software Technologies Ltd.
// All rights reserved.
```

This line effectively disables the macros that check for defined services. The change will take effect once the security policy is pushed to the enforcement points.

6.8: Virtual Defragmentation Errors

In order to determine whether or not a fragmented packet should be allowed, FireWall-1 holds all fragments it receives until it can assemble the entire packet in memory. If the assembled packet would normally pass, FireWall-1 passes the packet but sends it out as it was received—fragmented—thus the term *virtual defragmentation*. If FireWall-1 doesn't receive all the fragments for the packet or the fragment table fills up, which may occur during a fragmentation-based denial-of-service (DoS) attack, FireWall-1 drops the fragments and does not forward them, generating log messages along the way.

6.9: Too Many Internal Hosts

This error shows up when you have a node-limited firewall license and FireWall-1 believes you have violated the license because it has "seen" too many hosts on the internal interfaces. Note that the configuration in the Topology section of the gateway object determines which interfaces are internal and external. (See Fun with Check Point Licensing in Chapter 2 for discussion of node-limited licenses and their enforcement.)

If you see this error, it means the number of discrete IP addresses protected by the firewall has exceeded the license limitation. Anything behind your firewall with an IP address will eventually be discovered, regardless of whether or not the host traverses the firewall. Machines with multiple IP addresses and machines that change their IP addresses will be counted more than once.

When the license is exceeded by a large number of hosts on a busy network, FireWall-1 will consume itself with logging and messages about exceeding your license. In extreme cases, this will cause the firewall to process traffic very slowly, if at all. Note, however, that FireWall-1 will still continue to pass traffic, even from those hosts that exceed the license count. However, performance may be severely degraded because FireWall-1 spends time notifying you that your license count has been exceeded.

You can get a count of the number of hosts by entering the command **`fw tab -t host_table -s`**. The entry under the `#VALS` heading corresponds to the number of hosts it has counted. You can see which IP addresses are currently being counted against your license by issuing the command **`fw lichosts`**.

You will have to reset FireWall-1 in regards to the IP addresses it has erroneously logged as internal. Remove the `$FWDIR/database/fwd.h` and `$FWDIR/database/fwd.hosts` files and restart FireWall-1. You can also reset the table with **`fw tab -t host_table -x`**.

6.10: **Pth** SCHEDULER INTERNAL ERROR: No More Thread(s) Available to Schedule

This error comes up during policy installations from SmartDashboard/Policy Editor. You can safely ignore this message.

6.11: Target localhost Is Not Defined as an NG Module, Please Use the `-l` Flag

This message also shows up during policy installations from SmartDashboard/Policy Editor. Unfortunately, this error indicates that one or more objects in the `$FWDIR/conf/objects_5_0.C` file have been corrupted. There are a few ways to proceed.

1. If the management station was upgraded recently, try downgrading to the prior release and use the Upgrade Verifier to ensure consistency. You can download this utility from http://www.checkpoint.com/techsupport/downloadsng/utilities.html.
2. With the management station stopped (**`cpstop`**), replace `$FWDIR/conf/objects_5_0.C` with `$FWDIR/conf/objects_5_0.C.backup`. Restart the management station (**`cpstart`**) and see if the problem still occurs.

3. Check for duplicate IP addresses in the firewall and management gateway objects.
4. Upgrade to NG FP3, HFA-306, later HFA hot fixes, or NG AI. These versions resolve this issue.

6.12: Invalid Value in the Access Attribute: Undefined: File Exists

This error occurs when the topology settings have not been defined in the FireWall-1/VPN-1 version 4.1 object interfaces. This error message is harmless, and the policy does get installed on the version 4.1 module. To correct this situation, edit the FireWall-1/VPN-1 version 4.1 object interfaces properties and configure the topology settings with the appropriate options for your network configuration.

6.13: mbuf_alloc(1500): Cluster Alloc

If the firewall policy is installed when there is heavy traffic, the "mbuf_alloc" debug message may be displayed on the console. The message can be safely ignored.

6.14: Log Buffer Is Full, Error: Lost xxx Log/Trap Messages

The kernel module maintains a buffer of waiting log messages that it gives to `fwd` to send to the management module. The buffer is circular, so high levels of logging may cause buffer entries to be overwritten before they can be sent to `fwd`. When this happens, the system log will display messages indicating that log entries are being lost.

One solution to this issue is to reduce the amount of logging done. Disable any accounting rules that you can. Eliminate as much logging as possible.

Another solution is to increase the size of this buffer. In FireWall-1 NG, you will need to change the `fw_log_bufsize` kernel variable. This should be set to a value of `0x40000` or higher. FAQ 6.1 explains how to set these kernel variables.

Service-Related Questions

By design, firewalls restrict the use of certain services. Some services are more problematic than others. The following FAQs relate to the use of certain services through FireWall-1.

6.15: Why Doesn't Windows Traceroute Work?

This problem originally existed in pre-4.0 versions of FireWall-1. It does not exist in 4.0 or 4.1 versions of FireWall-1. Though the reason has changed, the problem has returned in FireWall-1 NG FP1 and FP2.

With an NG FP1/FP2 firewall using hide NAT, a packet sniffer shows that the client is being sent ICMP "time exceeded" messages as it should. However, the client appears to ignore these ICMP messages and displays "Request Timed Out" messages for hops past the firewall. Analysis of these ignored packets shows both an invalid checksum and less data than was sent by the ICMP echo-request packet (56 bytes of data received versus the 64 bytes sent). These are the likely reasons the packets are being ignored.

With an NG FP1/FP2 firewall using static NAT, the ICMP "time exceeded" packets at each hop after the firewall are dropped by the firewall with the message "ICMP packet out of state" in the logs.

Check Point issued hotfix SHF_FW1_FP2_0068 to resolve this issue. Upgrading to NG FP3 or later also solves the problem.

6.16: How Does FireWall-1 Support UNIX RPC?

Each service based on Remote Procedure Call (RPC) uses its own unique program number (within each service, a version number). When an RPC-based program starts, it uses a random TCP and/or UDP port number. The portmapper is used to map each program number to a particular port used by the RPC-based program at that moment. The connection to the portmapper process must be UDP for FireWall-1 to support it—TCP connections to the portmapper are currently not supported.

FireWall-1 supports RPC by monitoring the client RPC request to the portmapper. The portmapper replies with the port number. FireWall-1 temporarily opens that port number for the connection from the client to the server. Once the connection is over, FireWall-1 closes the port.

In terms of custom applications, 99% of the time, you can simply define your custom application as a new service using the following parameters:

- Type of connection (e.g., TCP, UDP, RPC)
- Port number (for TCP and UDP)
- Program number for RPC

Once done, you can use the newly defined service like any other network services.

6.17: How Do I Block AOL Instant Messenger?

To block AOL Instant Messenger, block access to the IP addresses listed in Table 6.1.

Table 6.1. IP addresses known to be used for AOL Instant Messenger

64.12.161.153	152.163.214.108	152.163.241.121	152.163.242.28
64.12.161.185	152.163.214.109	152.163.241.128	205.188.1.56
152.163.214.75	152.163.241.96	152.163.241.129	205.188.4.106
152.163.214.76	152.163.241.120	152.163.242.24	205.188.147.114

Table 6.2. IP addresses known to be used for Yahoo Messenger

204.71.177.35	204.71.201.48	216.115.107.64	216.115.107.103
204.71.200.54	204.71.202.58	216.115.107.65	216.115.107.104
204.71.200.55	204.71.202.59	216.115.107.66	216.115.107.105
204.71.200.56	216.115.105.214	216.115.107.67	216.136.173.179
204.71.200.57	216.115.105.215	216.115.107.101	216.136.172.221
204.71.200.68	216.115.107.63	216.115.107.102	204.71.202.73
204.71.201.47			

6.18: How Do I Enable or Block Yahoo Messenger?

To do this, you need to allow or block access via port 5050 to the IP addresses listed in Table 6.2.

6.19: How Do I Block ICMP Packets of a Particular Length?

You can block ICMP packets by specifying a maximum acceptable length. For example, to block packets that are longer than 100 bytes, first define a service of type Other. Then set the protocol number to 1 and put the following in the Match field:

```
ip_len > 100
```

This will match any ICMP packets greater than 100 bytes in length (including headers). Create a rule with this new service to drop the packet.

Problems with Stateful Inspection of TCP Connections

The problem with using a stateful firewall is that if the applications that go through it have a slightly different concept of what proper TCP state should be, or if the firewall makes invalid assumptions, some services will cease to function. The following subsections explain what some of those errors are and how to fix them.

6.20: TCP Packet Out of State

The "TCP Packet out of state" error message means that FireWall-1 sees a TCP ACK packet for which it does not have a matching state table entry. This may occur because the connection was inactive for a period of time or the connections tables were flushed (e.g., because of a policy installation or restart).

A little history is in order here. In FireWall-1 4.0 and earlier, if FireWall-1 received a TCP ACK packet that didn't match an entry in the connections tables, it would strip off the data portion of the packet (thus making it harmless), change the TCP sequence number, and forward the packet to its final destination. Because the destination host would see an unexpected sequence number, it would send a SYN packet to resynchronize the connection. The SYN packet would then go through the rulebase. If the rulebase permitted the packet, the entries in the connections tables would be recreated and the connection would continue.

In FireWall-1 4.1 and FireWall-1 4.1 SP1, FireWall-1 allows the unsolicited TCP ACK packet only if it comes from the server. If the TCP ACK packet comes from the client (i.e., the machine that originated the connection), the TCP ACK packet is dropped.

Then someone figured out that this handling of ACK packets could be used to cause a DoS attack against both the firewall and the host behind it. Since FireWall-1 4.1 SP2, by default FireWall-1 drops ACK packets for which there are no entries in the state tables. However, in NG FP3 and above, you can revert back to the pre-4.1 SP2 behavior by going into the Global Properties frame, Stateful Inspection tab, and unchecking the "Drop out of state TCP Packets" box. In NG FP2 and before, use **dbedit** as described in FAQ 4.2 and enter the following commands:

```
dbedit> modify properties firewall_properties
            fw_allow_out_of_state_tcp 1
dbedit> update properties firewall_properties
```

NOTE! FireWall-1 NG FP2 does have the option in the GUI to make this change. However, the option doesn't entirely work due to a coding error that still uses the NG FP1 method.

6.21: Configuring FireWall-1 to Allow Out-of-State Packets for Specific TCP Services

Some application vendors use TCP connections in ways that do not follow the standards documented in RFC793. Since FireWall-1 attempts to enforce strict adherence to the standards, applications that do not comply will have difficulties

communicating through FireWall-1 or any other stateful packet filter. NG FP2 and above provide a functionality that allows TCP packets for a specific port number even if they do not conform to Check Point's idea of state. This allows out-of-state TCP packets for specific services provided the packets would normally be passed by the rulebase. To do this, edit `$FWDIR/lib/user.def` on the management station and add a line of code (set in bold) within the following context:

```
#ifndef __user_def__
#define __user_def__

//
// User-defined INSPECT code
//

deffunc user_accept_non_syn() { dport = 22 };

#endif /* __user_def__ */
```

The INSPECT code between the curly braces defines the service(s) you wish to allow. The preceding example is SSH (TCP port 22). To define multiple services—for example, SSH (port 22), https (port 443), and ldap (port 389)—replace the bold line in the preceding example with this one:

```
deffunc user_accept_non_syn() { dport=22 or dport=443 or
        dport=389 };
```

To permit non-SYN packets between hosts a.b.c.d and x.y.z.w in addition to non-SYN packets on port 22, use the following:

```
deffunc user_accept_non_syn() { (src=x.y.z.w, dst=a.b.c.d) or
                                (src=a.b.c.d, dst=x.y.z.w) or
                                dport=22 };
```

(See Chapter 14 for more information on INSPECT.) If the rulebase is constructed carefully enough, the firewall should be relatively safe from an ACK-type DoS attack because all packets allowed by this change must still pass the rulebase.

6.22: SmartView Tracker Log Error: Rule 0: Reason: Violated Unidirectional Connection

FireWall-1 can mark a connection in the connections table to allow traffic to pass in one direction only. This can either be a connection that started from the inside, in which case FireWall-1 would mark the table to read that only outbound packets are allowed, or it can be a connection that originated from the

outside, in which case FireWall-1 would mark the table to read that only inbound packets are allowed. This means that data can pass in only one direction (ACK packets as part of normal TCP are acceptable). When a packet violates a unidirectional connection, Check Point logs an entry into SmartView Tracker/Log Viewer.

UDP services have an option to set a service to accept replies. In a sense, that is unidirectional. Unidirectional TCP connections occur with FTP. Some programs that use FTP do so in a nonstandard way that requires all the connections used by the FTP connection to be bidirectional.

To allow for bidirectional FTP connections in FireWall-1 NG, perform the following steps.

1. Stop the FireWall-1 management station with **`cpstop`**.
2. Edit `$FWDIR/lib/base.def` on the management station. Add the following bolded lines within the context shown:

```
deffunc ftp_port_code() {
ftp_intercept_port(CONN_ONEWAY_EITHER) or (IS_PASV_MSG,reject or 1)
};

deffunc ftp_pasv_code() {
ftp_intercept_pasv(CONN_ONEWAY_EITHER) or (IS_PORT_CMD,reject or 1)
};

deffunc ftp_bidir_code() {
ftp_intercept_port(NO_CONN_ONEWAY)
or
ftp_intercept_pasv(NO_CONN_ONEWAY)
};

deffunc ftp_code() {
ftp_intercept_port(CONN_ONEWAY_EITHER)
or
ftp_intercept_pasv(CONN_ONEWAY_EITHER)
};
```

3. Edit `$FWDIR/conf/tables.C` on the management station as follows (changes are set in bold):

```
: (protocols
        :table-type (confobj-dynamic)
        :location (protocols)
        :read_permission (0x00000000)
        :write_permission (0x00040000)
```

```
        :queries (
                :all (*)
        )
)
```

Note that `table-type` will be changed from `confobj-static` to `confobj-dynamic`.

4. Start the FireWall-1 management station with **`cpstart`**.
5. Use **`dbedit`** to enter the following commands:

```
dbedit> create tcp_protocol FTP_BI
dbedit> update protocols FTP_BI
dbedit> modify protocols FTP_BI handler ftp_bidir_code
dbedit> modify protocols FTP_BI match_by_seqack true
dbedit> modify protocols FTP_BI res_type ftp
dbedit> update protocols FTP_BI
dbedit> quit
```

This allows you to create the bidirectional FTP service.

6. Open up SmartDashboard/Policy Editor and create a new service of type TCP. It will be on port 21. Give it a name other than FTP_BI (e.g., ftp_bidir). Click the Advanced button and select FTP_BI as the protocol type.
7. Use the new service in a rule. Install the security policy.

6.23: th_flags X message_info SYN Packet for Established Connection

This error can be seen in SmartView Tracker/Log Viewer when FireWall-1 receives a new connection from a source to a destination over the same port/service as a connection that was recently closed with a FIN or RST. FireWall-1 hangs onto these connections until the TCP end timeout is reached, which defaults to 60 seconds. This behavior is normal and expected.

The first step in alleviating this issue is to lower the TCP end timeout to see if that helps remove the connection from the connections table in time for the new connection to be received without a conflict. In FireWall-1 NG FP2 and later, the TCP end timeout can be modified via the GUI in the Stateful Inspection frame of the Global Properties section.

If the problem still occurs, the solution is to use TCP Sequence Verifier in NG FP3 to enable FireWall-1 to see the connection as a new connection, not an established one. For this to work properly, you need to run NG FP3 or above. On Nokia platforms, ensure that you have disabled flows. Contact Nokia Support for assistance.

Another option exists in hotfix SHF_FW1_FP3_0114, which is included in NG FP3 HFA-311 and above. You can change the behavior by modifying the value of the kernel variable `fw_reuse_established_conn` in three ways: change it to the TCP port number on which you need this behavior, change it to `-1` for all ports, or change it to `-2` to disable the behavior. See FAQ 6.1 for instructions on how to edit FireWall-1 kernel variables.

6.24: TCP Flags Do Not Make Sense

These errors show up in SmartView Tracker in FireWall-1 NG FP3 and above. SmartDefense is dropping packets with the SYN and RST flags set as malformed instead of as a normal RST packet.

Check Point provides a fix for this issue in hot fix SHF_FW1_FP3_0114. This fix is included in NG FP3 HFA-311 and above. After applying the fix, you can change the behavior by modifying the value of the kernel variable `fw_accept_syn_rst` to the TCP port number on which you need this behavior, to `-1` for all ports, or to `-2` to disable the behavior. See FAQ 6.1 for instructions on how to edit FireWall-1 kernel variables.

6.25: Unexpected SYN Response

These error messages show up in SmartView Tracker on FireWall-1 NG FP3 and above when the firewall receives unexpected SYN-ACK packets. To allow these packets, change the kernel variable `fw_allow_out_of_state_syn_resp` to `1`. FAQ 6.1 explains how to change kernel variables.

6.26: Enabling the TCP Sequence Verifier

Prior to FireWall-1 NG FP1, FireWall-1 did not perform any checking of TCP sequence numbers. NG FP1 introduced this functionality, which validates the TCP sequence numbers used in a connection. It provides better tracking of the state of TCP connections. Enabling this feature can eliminate certain kinds of error messages in the logs and possibly create others.

To enable TCP Sequence Verifier on NG FP3 or above, in SmartDashboard, select SmartDefense from the Policy menu. The option is listed under TCP as Sequence Verifier.

To enable TCP Sequence Verifier on NG FP2, check the "Drop out of sequence packets" option under TCP Sequence Verifier in the Stateful Inspection frame in the Global Properties section.

To enable TCP Sequence Verifier on NG FP1, use **`dbedit`** to edit the following property to `true` in the `objects_5_0.C` file:

```
dbedit> modify properties firewall_properties fw_tcp_seq_verify 1
dbedit> update properties firewall_properties
```

6.27: Adjusting TCP or UDP Timeouts on a Per-Service Basis

In FireWall-1 NG, you can set these timeouts in the GUI directly. For both TCP and UDP services, go into the Advanced section of the service in question. For TCP services, edit the session timeout. For UDP services, edit the virtual session timeout. Reinstall the security policy.

6.28: Disabling TCP Timeouts

It is usually better to use some of the other tricks discussed to permit TCP packets that are out of state, such as the method described in FAQ 6.21. I don't even want to think about the security implications of leaving idle TCP connections open forever, but my gut tells me that this is not a good idea.

If you absolutely need to disable timeouts for a service because the vendor of your application refuses to implement a mechanism for periodically checking to see whether a connection is alive, this is how you would do it with **`dbedit:`**

```
dbedit> modify services service-name timeout 2147483647
dbedit> update services service-name
```

The value specified in the preceding example is used internally by the kernel to specify connections that do not time out. However, if you set any smaller number slightly less than 2,147,483,647, you still get connections that should last many years, assuming you do not stop your firewall for that long.

Problems with FTP

While FTP has been around since before the Internet ran on TCP/IP, every client and server seems to act a little differently. Stateful firewalls like FireWall-1, which expect things to happen only in certain ways, get tripped up by clients and servers that are RFC compliant, but choose to implement the RFCs differently. The following FAQs are related to FTP problems.

6.29: Problems with Newline Characters

Some FTP implementations send a `PORT` command in one packet and the newline character in another. By default, FireWall-1 assumes the `PORT` command and the newline will appear in the same packet. To enable checking for this, uncomment out the bolded `#define` statement (i.e., remove the `//` characters at the beginning of the line) in `$FWDIR/lib/base.def` on the management console and reinstall the security policy.

```
//    Use this if you do not want the FW-1 module to insist on a
// newline at the end of the PORT command:
// #define FTPPORT(match)       (call KFUNC_FTPPORT <(match)>)
```

Some other sites do not send out a proper newline at all. To resolve this, comment out the following line in `$FWDIR/lib/base.def` on the management console (i.e., add `//` at the beginning of the line) and reinstall the policy.

```
#define FTP_ENFORCE_NL
```

6.30: FTP on Ports Other Than 21

Some FTP servers use an alternate port for their control connection. By default, FireWall-1 knows how to handle FTP control connections only on port 21. To allow FTP using an alternate port, create a new service of type TCP. In the advanced configuration for the service, set the protocol type to FTP. Use this new service in a rule.

6.31: FTP Data Connections with a Random Source Port

Unfortunately, FireWall-1 NG does not currently allow for modification of INSPECT code to allow random FTP data port return traffic. The only existing workaround is the use of PASV transfers. Some FTP clients do not support passive mode, which means you may need to use a client that does.

6.32: FTP Servers Sending FIN Packets out of Sequence

It has been reported that Solaris clients cannot FTP to an NT SP6a platform. Microsoft's TCP/IP stack sends the FIN packets out of sequence. Installing the latest version of patch 105529 on Solaris resolves this issue.

6.33: FTP Servers That Require ident

Some FTP servers require a connection back to the FTP client on port 113 (ident). You will have to create an explicit rule permitting ident back to the client, which means that if you're using hide translation, you will simply not be able to access this FTP server.

6.34: Encrypting FTP Connections with SSL

Firewalls do not normally pass FTP connections encrypted with SSL—commonly referred to as FTP over SSL. The reason for this is simple: A firewall cannot inspect the FTP control connection because it is encrypted. FireWall-1 therefore cannot predict the FTP ports used by the FTP over SSL session.

Some people have been able to get this to work by simply applying FAQ 6.29, assuming the ports used are the standard TCP port 21 for control and 20 for data. Some variants of FTP over SSL operate over different ports—using

SOURCE	DESTINATION	SERVICE	ACTION
ftp-client	ftp-server	TCP ftp-ssl-control	accept
ftp-server	ftp-client	TCP ftp-ssl-data	accept

Figure 6.1 Rulebase for an FTP over SSL connection

port 990 for control and port 989 for data. In this case, you simply need to create the following TCP services:

- *ftp-ssl-control:* port 990
- *ftp-ssl-data:* port number higher than1024, source port 989

In other words, ftp-ssl-data accepts connections with a destination port of any TCP high port provided the source port is 989. The rulebase to permit access would look similar to Figure 6.1.

NOTE! In no case will FTP over SSL be supported with hide NAT. This is because FireWall-1 is unable to see the control portion of the connection—it is, after all, encrypted. Thus the ports used by the control connection cannot be modified. FTP over SSL will work with static NAT.

Problems That Aren't the Firewall's Fault

There are several issues that some people think are related to the firewall or think that their firewall should be able to do. This section documents some of these issues that have nothing to do with FireWall-1.

6.35: Some Services Are Slow to Connect

Some services are slow to connect either because the remote server is not able to do a reverse DNS lookup on the IP address you are coming from (it is timing out while looking) or because they are expecting an answer to their query on the ident port. To fix the latter problem, see FAQ 6.36. To fix the former problem, you must ask your DNS administrator to modify the reverse lookup tables so that the IP address you are coming from is resolvable.

6.36: The ident Service

When attempting to use certain services like SMTP or IRC, the server tries to send a communication back to the client on the ident service port. The ident service is typically used to provide identification for certain services. In general, it is not necessary. It is highly recommended that you create a rule that rejects

all ident traffic (instead of dropping it) without logging so that services that rely on ident will start faster because they won't wait for the ident connection to time out.

6.37: Different DNS Definitions for Internet and Intranet

When you have different DNS definitions available for internal and external hosts, you want what is commonly referred to as *split-horizon DNS*.

Your external DNS servers (i.e., the ones responsible for serving DNS queries to the outside world) contain only the bare minimum information—mail exchanger (MX) records, externally accessible hosts, and reverse lookup for your IP space. The internal DNS is a superset of the external DNS server, containing both inside and outside names and IP numbers. Your internal hosts and the firewall use the internal DNS server, which may use the external DNS server as a forwarder to answer requests (i.e., resolve queries for domains outside your own).

Each DNS server should be set up on different systems. Your internal DNS server should be inside your firewall on the internal network. Your external DNS server should be either on the DMZ/service network or outside the firewall entirely (perhaps your ISP manages it). Some firewalls run a DNS server on the firewall itself. You can do this, but most people (myself included) do not recommend this configuration.

Summary

This chapter dealt with common situations that have occurred in many FireWall-1 installations, some for several years. While this is not the first time most of these issues have been disclosed, it is the first time solutions to them have been made available in print.