Chapter 11

Testing the Firewall

Solutions in this chapter:

- **OS Configuration**
- **Firewall Configuration**
- **Working with Firewall Builder**
- **System Administration**
- **Testing the Firewall Rulebase**
- **Identifying Misconfigurations**
- **Identifying Vulnerabilities**
- **Packet Flow from All Networks**
- **Change Control**
- **Validated Firewalls**

☑ **Summary**

Introduction

In this chapter we will introduce the concepts of auditing or testing firewalls.

First we need to define a firewall. A firewall is an application, device, system, or a group of systems that controls the flow of traffic between two networks based on a set of rules, protects systems from external (internet) as well as internal threats, separates a sensitive areas of a private network from less sensitive areas, encrypts internal and external networks that transmit sensitive data (when used as a VPN endpoint), or hides internal network addresses from external networks (network address translator). A firewall picks up where the border router leaves off and makes a much more thorough pass at filtering traffic. Firewalls come in different types, including static packet filters (for example Nortel Accelar router), statefull firewalls (for example Cisco PIX), and proxy firewalls (for example Secure Computing Sidewinder).

Similar to routers, a firewall uses various filtering technologies or methods to ensure security. These methods include packet filtering, statefull inspection, proxy or application gateway, and deep packet inspection. A firewall can use just one of these methods, or it can combine different methods to produce the most appropriate and robust configuration.

A good way to start to test a firewall is to gather information from individuals that have some responsibility for it. These people may be members of the audit team, system administrators, network administrators, members of the policy team, and information security personnel. The idea is to gather and collate each person's perceptions of what the firewall's functionally should be and what it is configured to provide for the network and systems. Obtain any existing firewall documentation and network diagrams to verify the information gathered from the interview. Ideally, the firewall is a control designed to reflect policy. This means that policy must be in place before the firewall is configured. Sadly, this is seldom the case.

After the information detailed above has been collected, the auditor can develop an understanding of the firewall architecture, and determine whether the firewall is configured to correctly segment networks and defend information. The next step is to evaluate the operating system (OS) configuration. This is the configuration of the firewall platform itself. All firewalls have an OS. Do not be fooled by vendor assertions that firewalls have an appliance. A firewall appliance typically will just have an OS that has been hardened. The appliance could in fact be running a scaled down version of Unix or, in some cases, be running a customized OS written by the firewall company, as in the case of the Cisco Adaptive Security Appliance (ASA). Firewalls and routers are all software driven; all they do is make it more difficult to see the code.

Next it is important to ensure that system administration follows best practice: user management, patch updates, change control, and configuration backups. If the firewall is not patched it will eventually be compromised. Just because it is a security device, it is not automatically secure.

Finally, it is necessary to validate that the firewall rulebase matches the organizational policy.

Testing the firewall should be coordinated with testing the other components of the organization's defense-in-depth methodology. The organization should not rely only on a single line of defense; if it does, raise a red flag. Firewalls are not the panacea for all security ills. They mainly slow attackers and log activity.

The overall result of the testing or audit of the firewall would be the identification of any security vulnerabilities, as well as an assessment of whether the firewall is fulfilling its function in relation to the security policy of the company. Assess whether the setup, configuration, and operation of the

firewall are secured sufficiently to protect the information or services that the firewall is intended to guard, considering the risks that were identified and the likelihood of occurrence.

The Center for Internet Security provides benchmarks for several specific brands of firewalls devices. The benchmarks (available at www.cisecurity.org) greatly aid in developing an audit program for firewalls. These benchmarks are the source of our checklist frameworks.

OS Configuration

When auditing the firewall, the auditor must look at the platform or the OS on which the firewall is running.

An auditor needs to check on whether the OS on which the firewall is installed is stripped to contain only the minimum functionalities or services that are required to provide the functions it runs. The firewall should be an isolated system dedicated to one purpose only, which is filtering traffic based on defined rules. The less complex the installation, the simpler its administration will be. Fewer features equates to less patching and fewer vulnerabilities.

To verify this, commands can be used for determining what services and ports are available to the OS.

Many operating systems have a number of built-in tools that may be used to determine which ports are listening. Some examples are listed here with more in the chapters associated with specific operating systems:

- **UNIX**: lsof –I, netstat –a, and ps –aef
- **Windows** the Service Microsoft Management Console (MMC), *netstat –a* and *fport*

When first determining the open ports and services, the firewall should be turned off (disabled or running with a policy that allows all traffic). This is done to test only the operating-system-specific ports and services. It is important to do this on a secure network and not connect the firewall to the Internet at this point. Remember, the firewall is a router in this mode.

In addition, the security settings and vulnerabilities of the OS that is installed should be analyzed. Every OS includes a set of security features and vulnerabilities, which varies from vendor to vendor and even between versions. For instance, the default security settings of the OS may not be modified during the installation and such settings may not meet the desired level of security that is consistent with the security policy. Some of the most common security settings that can be evaluated are the access rules, password rules, and logging rules. Other OS/version-specific settings and parameters should also be verified.

Centre for Internet Security also provides benchmarks for several OS. Those benchmarks (available at www.cisecurity.org) can greatly aid in determining whether the OS is configured based on the general industry best practices.

Firewall Configuration

After looking at the firewall platform's OS, the next stage involves the validation of the firewall configuration. All firewalls have both a configuration and policy. These should not be confused. The configuration is the set of base settings associated with the firewall software and installation.

Changes to the configuration of the firewall will change its behavior, and, hence, how it processes in accordance with the policy.

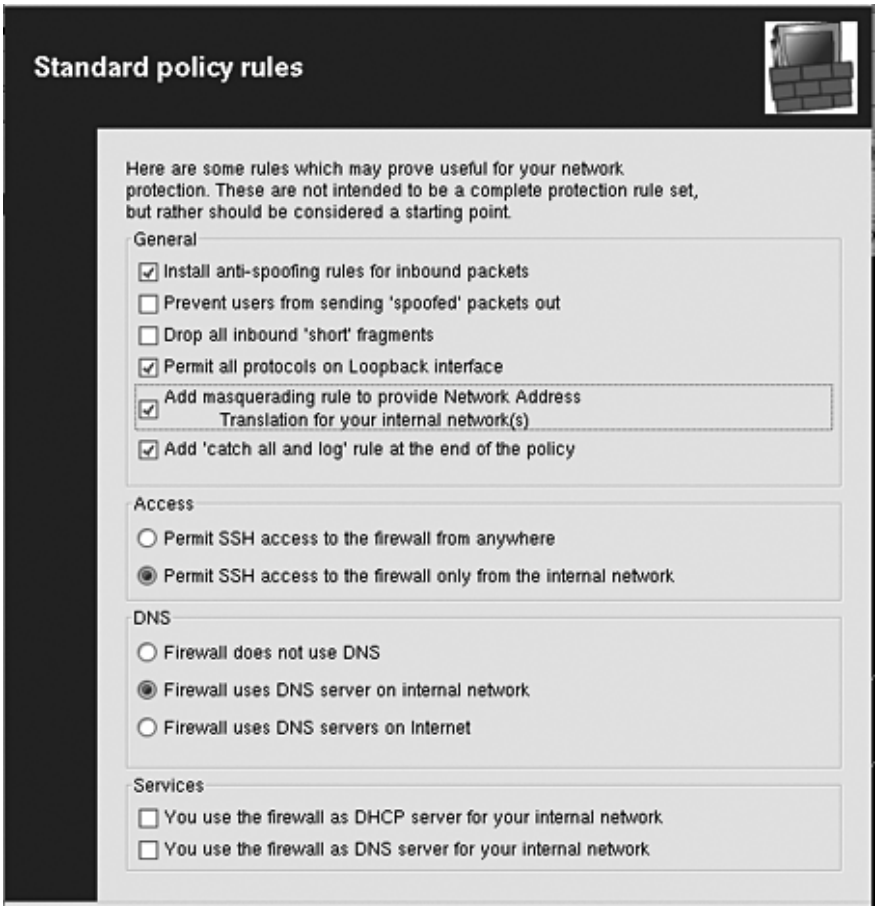
Again the auditor must check on whether the firewall sits on an isolated system dedicated to one purpose only, which is filtering packets (and logging, of course). For instance, DNS, e-mail, or server load-balancing functions should not be installed on the same host or be processed by the firewall platform. The sole exception here is that load-balancing the firewall itself is a function of a high-availability firewall and is allowed.

Since the fundamental purpose of the firewall is to manage the flow of information between two networks, the auditor must look at how it serves such a function by looking at the firewall's configuration. We need to verify whether the traffic that the firewall allows to pass through is consistent with the security policy. Testing the rulebase is discussed in the latter part of the chapter, but critical things to look at are that:

- The access rules (authentication, authorization, and accounting) for the firewall are in line with the security policy and best practices
- Access to the firewall system for management and maintenance is provided using an encrypted channel
- Physical access to the device is restricted
- The firewall is configured to hide internal restricted DNS information from external networks
- The external firewall restricts incoming SNMP queries
- The firewall is configured as fail closed
- The firewall hides internal information from external sources
- The firewall is configured to deny all services, unless explicitly allowed
- Al security-related patches are applied to the firewall system
- Configuration settings are properly backed up and accessible to authorized personnel only

Figure 11.1 illustrates an example of a firewall's standard policy rules. In this example, the standard policy rules detail the default settings that will be merged with the policy before being installed. Thus, the configuration and the policy when applied together make the rules that are enforced at the firewall.

Figure 11.1 Standard Firewall Rules Configuration



Working with Firewall Builder

Firewall Builder (www.fwbuilder.org) is a general public license (GPL) software package designed to aid administrators in configuring firewalls. The current version, Firewall Builder v 2.1.18, supports the following firewall platforms:

- **FireWall Services Module** (FWSM)
- ipfilter
- ipfw
- iptables
- PF

- Cisco **Private Internet Exchange** (PIX)
- and a number of other platforms such as:
  - FreeBSD
  - Cisco FWSM
  - Linksys/Sveasoft
  - GNU/Linux (kernel 2.4 and 2.6)
  - Mac OS X
  - OpenBSD
  - Solaris

Following the setup of standard policy, the next decision to be made by the administrator is to define the interfaces of the firewall and, consequently, the configurations for each of the interfaces. Examples of interfaces that a firewall could usually have are the external interface (untrusted) and the internal interface (trusted). Testing the firewall would therefore involve the testing of the configurations of each of the firewall's interfaces to validate their compliance with the firewall policy of the organization.

Building or Only Testing

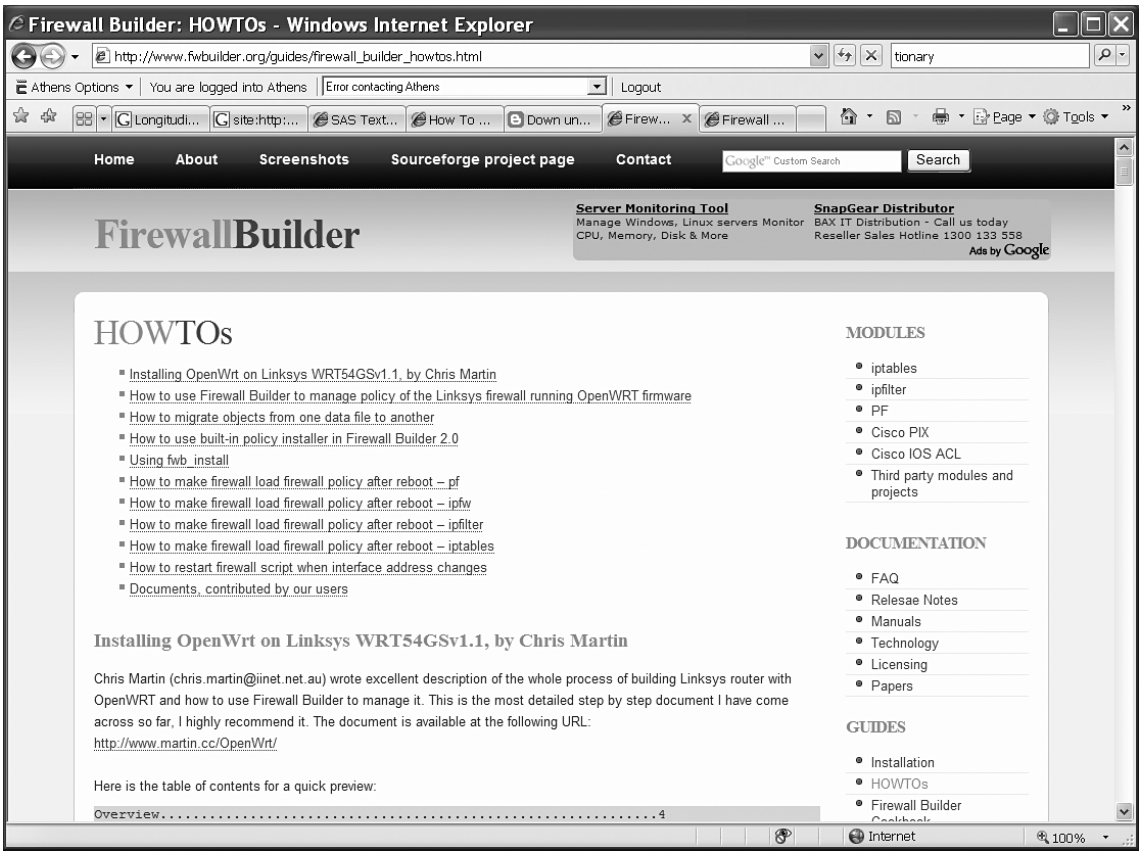
Firewall Builder has a number of configuration guides available on its Web site as shown in Figure 11.2:

www.fwbuilder.org/guides/firewall_builder_howtos.html

www.fwbuilder.org/guides/firewall_builder_cookbook.html

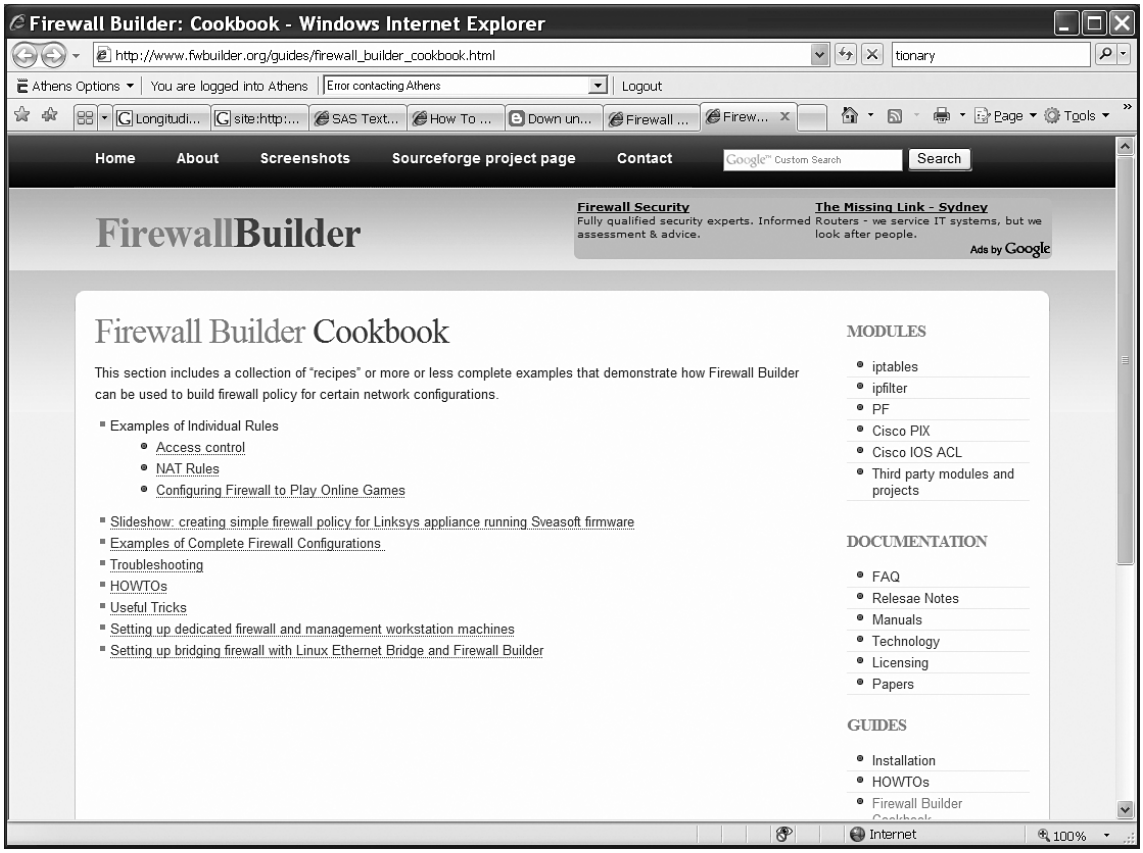
Most vendors also have their own guidelines and install guides as well. On top of this, there are a large number of good configuration books for both generalized firewall knowledge and excellent system-specific ones (such as *Check Point NGX R65 Security Administration* released by Elsevier).

Figure 11.2 Firewall Builder How-To Guides



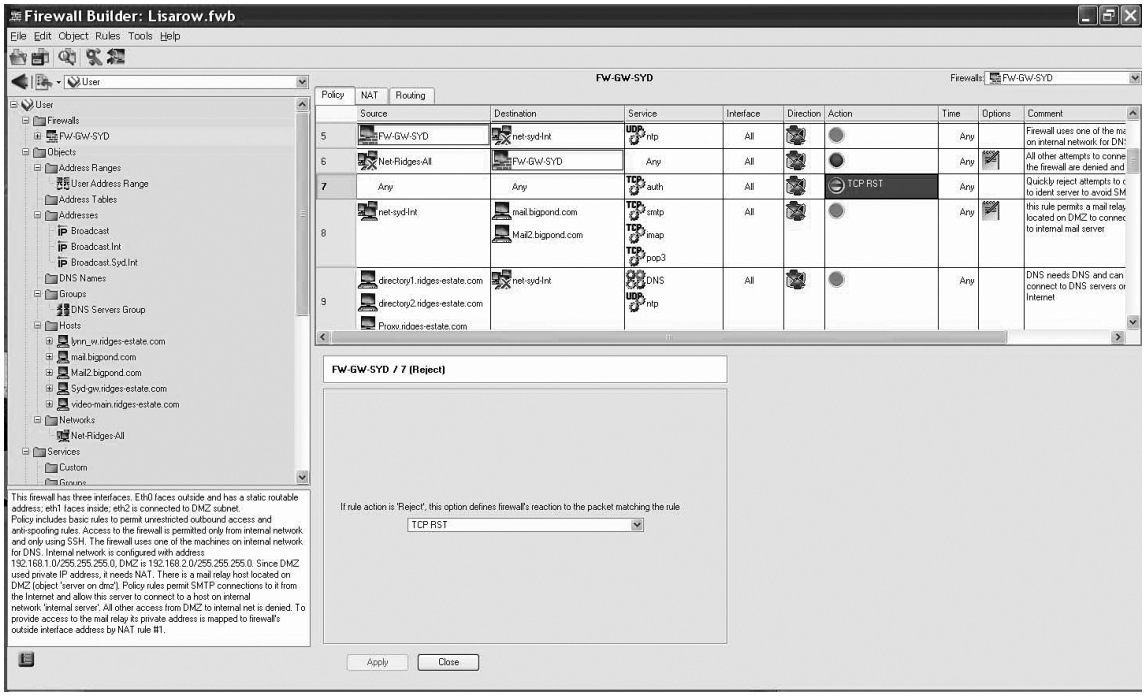
The main advantage (other than low cost, even commercially) of a tool such as Firewall Builder is that it is able to manage several systems (see Figure 11.3).

Figure 11.3 Firewall Builder Cookbook



Firewall Builder also uses an interface that is both simple and very familiar to anyone who has worked with the commercial products. Figure 11.4 is an example of the Firewall Builder user interface.

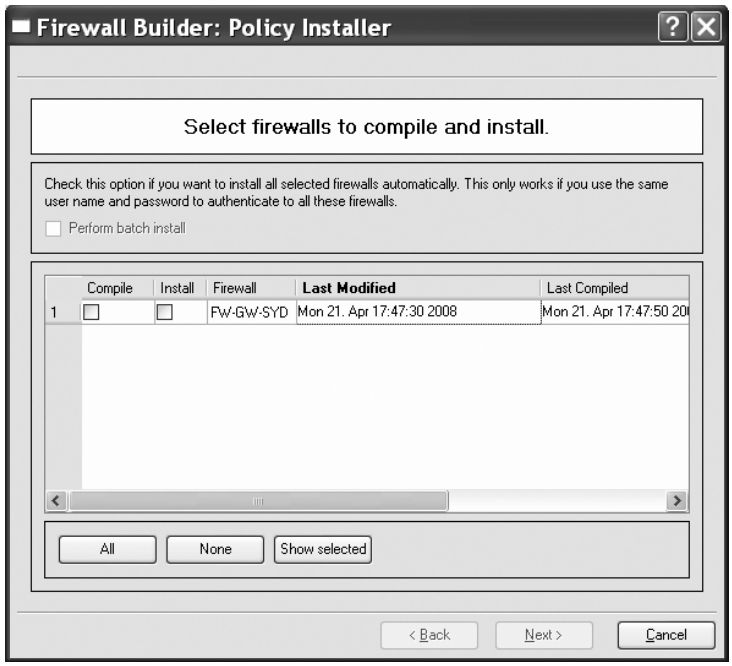
Figure 11.4 Standard Firewall Rules Configuration



This interface allows the auditor to quickly validate configuration against the policy. Also, this tool provides the capability to save rulesets. This feature enhances change management. By being able to go back and view previous rulesets, the auditor can see the patterns of change as they occur over time and also seek reasons for rules that have been added.

The *policy installer* (see Figure 11.5) adds the capability to quickly view the date when the policy was last compiled and last installed (and if these are the same).

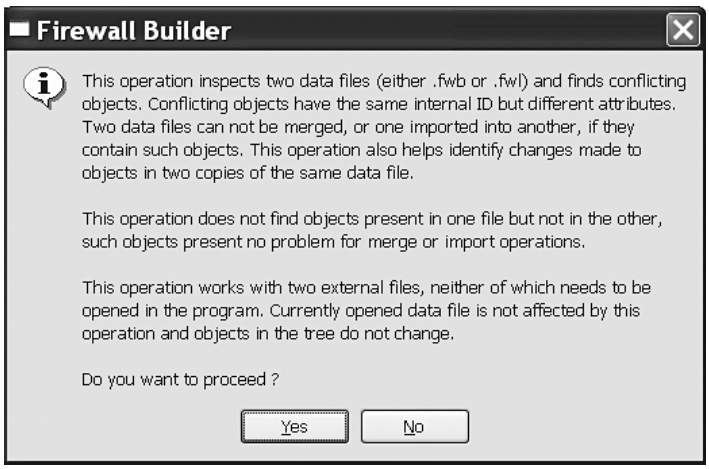
Figure 11.5 Firewall Builder Policy Installer Rules Compilation and Installation



Conflicting Rules

From time to time it is necessary to merge rulebases. For this reason the Firewall Builder tool has a validation function (see Figure 11.6).

Figure 11.6 Firewall Builder Rules Conflict Checker



System Administration

When you are auditing a firewall, the next thing to look at is how the operation of the firewall is being managed, continually tuned in, and monitored. Processes to be looked at are:

- The process of user administration (that is, who can access the firewall device and make changes to its configuration)
- The process of making changes to the configuration and firewall rulebase.
- The process of updating and applying security patches to both the OS and the firewall
- The process for monitoring new bugs or weaknesses of the firewall software
- The process of determining whether all necessary firewall activities are being logged
- The process of determining whether rule activity, logs, and rules violations are monitored
- The process of determining whether continuity plans for firewalls are in place

Testing the Firewall Rulebase

The Firewall rulebase is the set of rules that dictate which packets are allowed or rejected (dropped) as they are encountered by the firewall. Packets can come from inside and outside sources, and the firewall's rulebase determines whether a packet is allowed to pass through, based on several criteria or rules.

Most firewalls come with default settings. However, it is not surprising to know that these settings do not provide even the most basic level of security that most organizations would like to have. For example, some of Checkpoint's firewall appliances allow, by default, unrestricted and unlogged Domain name system (DNS), Internet Control Message Protocol (ICMP) and Routing Information Protocol (RIP) access both in and out of the firewall. These default settings leave the firewall open to Trojan horses, ping attacks (Ping of Death, smurfing, etc.), man-in-the-middle attacks, and others that exploit the open ports.

Testing the rulebase can bring to light certain misconfigurations and vulnerabilities that can affect the firewall's performance and the security of the network that it was installed to protect.

Rulebase management is certainly a problem area for many firewall administrators. It's easy for firewall rulebases to become riddled with incorrect, overlapping, and unused rules, even in the presence of a change management system. There has been a bit of academic research into this topic during the past few years, and researchers have identified a number of anomalies worthy of an administrator's attention.

- Overlapping/shadowed rules often occur when administrators create one high-priority rule that generalizes lower priority rules. For example, the administrator might create a rule that appears high in the rulebase, allowing all SMTP traffic. An older rule, lower in the base, might specifically allow SMTP traffic to a mail server. Because of its similarity and lower priority, however, this more specific rule will never be triggered. The situation could be made worse when the lower rule is intended to block traffic to a particular server. Since the generalized rule appears first, the block would never take effect.

- Orphaned rules occur when services or systems disappear from the network or other changes render a rule obsolete. All too often, these rules are never removed from the firewall, creating a potential security hole and adding to a firewall administrator's burden.
- Unused rules are similar to orphaned rules, except these rules were never used in the first place. Unused rules could be the result of change requests from projects that never materialized, or the unused rules could result from administrator errors when creating rules.

A number of commercial tools attempt to tackle these problems. Examples of these tools are FireMon from Secure Passage LLC and Firewall Analyzer from Algorithmic Security (AlgoSec) Inc. The true solution, however, is to keep your rulebase simple, limit it to a manageable size; and conduct regular audits.

Identifying Misconfigurations

Some areas to consider when assessing the firewall policy include:

- Has the design taken planned growth into account?
- Is the system patched and tested? (Do not assume that all patches work.)
- Does the policy provide defense in depth; does the architecture consider all layers of the Transmission Control Protocol/Internet Protocol (TCP/IP) stack?
- What is allowed into the network? What is allowed out? All traffic entering or leaving the network, should have a justification. Some regulations and standards, such as the PCI Data Security Standard (DSS) require this justification for all traffic, but it is good practice, even when not specified by an adopted standard.

The SANS GIAC Certified Firewall Analyst (GCFW) GCFW gold paper repository and the reading room subsite are great places to find papers on firewall design and architecture (www.sans.org/reading_room/whitepapers/firewalls/).

Identifying Vulnerabilities

A firewall vulnerability can be an error or a weakness in the firewall's design, implementation, or configuration that anyone with malicious intent can exploit to attack that which the firewall is believed to protect.

We can classify firewall vulnerabilities as:

- **Validation error** A validation error occurs when the program interacts with the environment without ensuring the correctness of environmental data. Three types of environmental data need validation: input, origin, and target. Input validation ensures that the input is as expected. This includes the number, type, and format of each input field. Origin validation ensures that the origin of data is actually what it is claimed to be, for example, checking the identity of the IP source. Target validation ensures that the information goes to the place it is supposed to. This includes ensuring that protected information does not go to an untrusted target.

- **Authorization error** An authorization error (authentication error) permits a protected operation to be invoked without sufficient checking of the authority of the invoking agent.
- **Serialization/aliasing error** A serialization error permits the asynchronous behavior of different system operations to be exploited to cause a security violation. Many time-of-check-to-time-of-use flaws fall into this category. An aliasing flaw occurs when two names for the same object can cause its contents to change unexpectedly, and, consequently, invalidate checks already applied to it.
- **Boundary checking error** A boundary checking error is caused by failure to check boundaries and ensure constraints. Not checking against excessive values associated with table size, file allocation, or other resource consumption leads to boundary checking errors. Buffer overflow is a result of a boundary checking error.
- **Domain error** A domain error occurs when the intended boundaries between protection environments have holes. This causes information to implicitly leak out.
- **Design error** Design errors can be traced to the system design phase. For example, a weak encryption algorithm falls into this category.

Following are the most common effects of the vulnerabilities described above:

- **Execution of code** Execution of unwanted code occurs when a vulnerability can lead to code being illegitimately executed. This includes, but is not limited to, code written by an attacker.
- **Change of target resource** Change of target resource occurs when a vulnerability allows the state of a resource to be illegitimately changed by an attacker. A resource could be a host, a firewall rule table, or any entity that should be protected by the firewall.
- **Access to target resource** Access to a target resource occurs when a vulnerability allows an attacker illegitimate access to some resource. Again, a resource may be any entity that is protected by the firewall. Examples of this vulnerability effect include allowing an attacker to read the firewall rule tables or to find out which services are available on a protected host.
- **Denial of service (DoS)** DoS occurs when a vulnerability is exploited to disrupt a service provided to legitimate users. Services in this context may range from packet forwarding or network address translation to administration.

Firewall vulnerabilities are best identified by using automated tools called *vulnerability scanners*. These scanners determine the firewall's vulnerabilities by comparing its configuration against known weaknesses and vulnerabilities. The following are the most common tools used in the industry:

- Active vulnerability scanners such as Internet Security Systems (ISS) Internet Scanner, Symantec NetRecon, and Nessus (see the chapter on scanning with Nessus)
- Host-based scanners such as Microsoft Baseline Security Analyzer (MBSA), ISS System Scanner, and Symantec Enterprise Security Manager (see the chapter on scanning with MBSA)

Packet Flow from All Networks

Vulnerability scanners should be complemented with other specialized tools designed to analyze the packets going through the network.

Scanning the Network

Apart from assessing misconfigurations and vulnerabilities of the rulebase directly, the network itself should be scanned from every possible interface, both from the inside and outside, in all directions. For these scans, several tools that perform network mapping and port reconnaissance are available for download from the Internet, such as nmap, NmapWin, hping, Superscan and nemesis. Passive vulnerability assessment tools (packet sniffers) are also available; these capture and display network traffic for analysis. Examples of these tools are Wireshark, tcpdump, and windump, to name a few. Lastly, there are active vulnerability scanners, wherein especially crafted probes via plugins are sent through the network to see how the target will respond. Examples of active vulnerability scanners are Nessus, Saint, SARA, and others.

Using the aforementioned tools, you can perform some basic tests such as:

- Using Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) to scan the firewall for all possible 65535 ports.
- Performing a ping sweep to see if echo-requests can pass through
- Performing a SYN scan subnet to look for open ports (use a full TCP Connect scan for proxies)
- Performing a slow SYN scan to see if port scans are detected
- Performing a scan with FIN packets to see if they are handled differently
- Performing a scan with ACK packets to see if they are handled differently
- Fragmenting ACK packets to see if they are handled differently
- Performing a UDP scan subnet to look for open ports

It is recommended that security administrators use more than a couple of tools to scan and monitor the network. This use of multiple tools will minimize false positives and false negatives, and will give a more complete picture of the network.

When scanning, ensure that sniffers are configured to monitor traffic passing through the firewall. Do not trust the firewall logs alone.

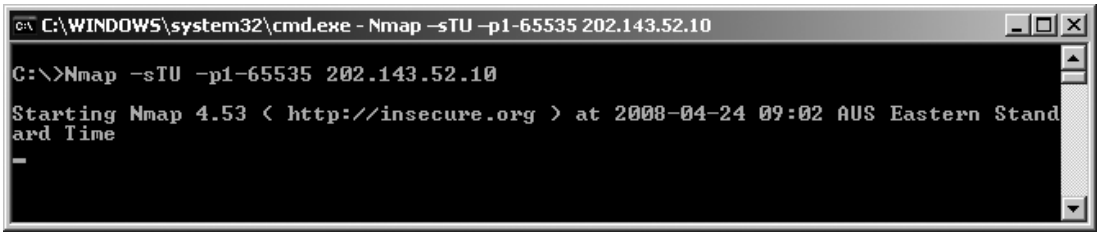
Using nmap

The following are screenshots captured while performing some of the basic tests listed above using nmap. Note that several types of information, such as open ports and running services, are displayed as output.

TCP and UDP scan the firewall for all possible 65535 ports; see Figure 11.7.

```
Nmap -sTU -p1-65535 <target>
```

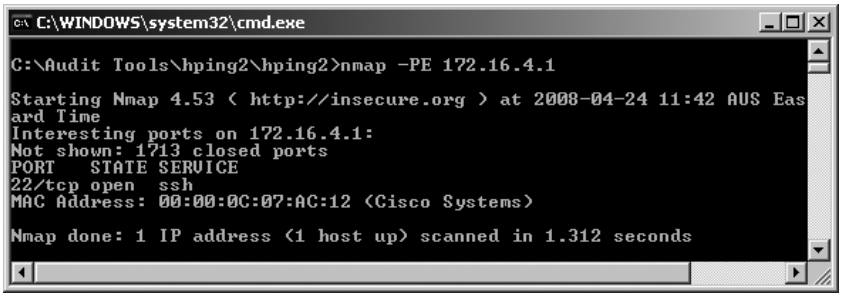
Figure 11.7 nmap Scanning for 65535 Ports



Perform a ping sweep to see if echo-requests can pass through; see Figure 11.8.

```
Nmap -PE <target>
```

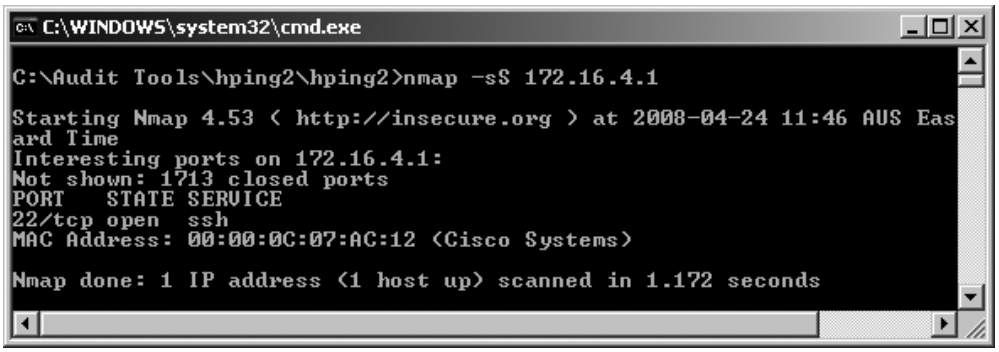
Figure 11.8 nmap Scanning Ping Sweep



SYN scan subnet to look for open ports (use a full TCP Connect scan for proxies); see Figure 11.9.

```
Nmap -sS <target>
```

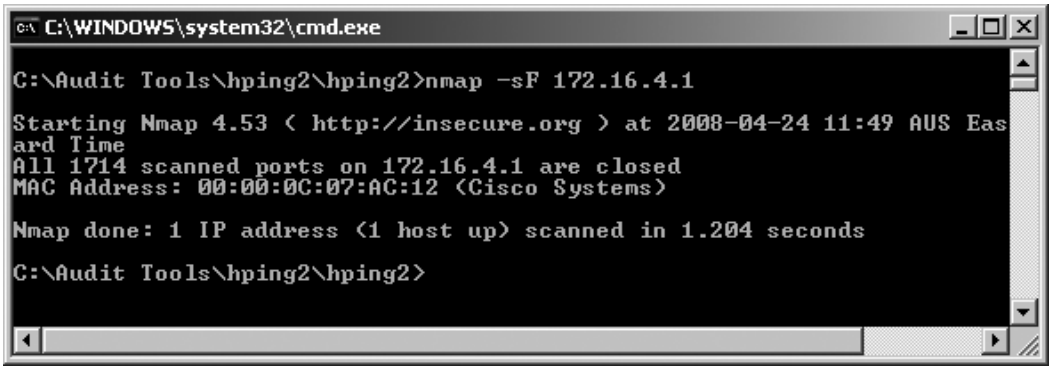
Figure 11.9 Nmap SYN Scanning for Open Ports



Scan with FIN packets to see if they are handled differently; see Figure 11.10.

```
Nmap -sF <target>
```

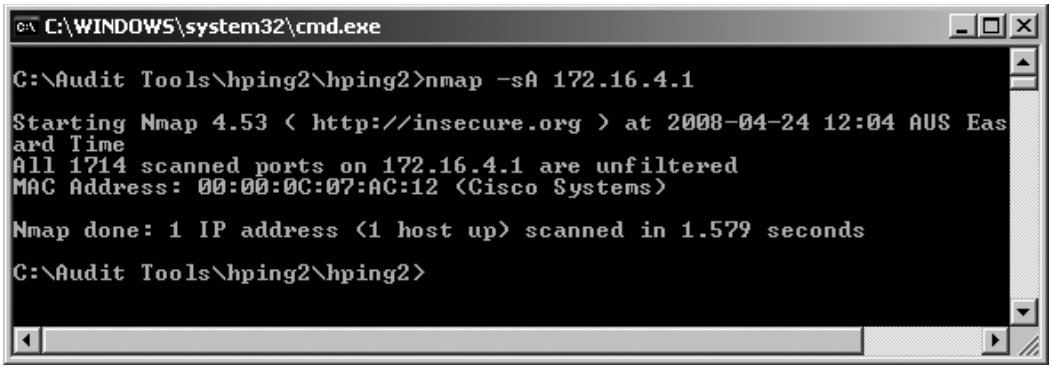
Figure 11.10 nmap Scanning with FIN Packets



Scan with ACK packets to see if they are handled differently; see Figure 11.11.

```
Nmap -sA <target>
```

Figure 11.11 nmap Scanning with ACK Packets



UDP scan subnet to look for open ports; see Figure 11.12.

```
Nmap -sU <target>/24
```

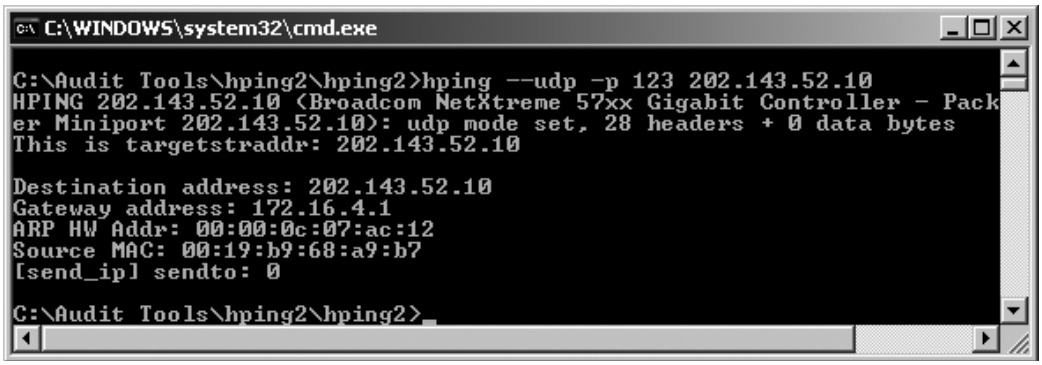
Figure 11.12 nmap UDP Scanning for Open Ports

Using hping2

Also available is hping2, a command-line oriented TCP/IP packet assembler/analyzer. Patterned after the *ping(8)* Unix command, hping supports TCP, UDP, ICMP and Raw IP protocols, has a *traceroute* mode, the ability to send files through a covert channel, and many other features. All header fields can be modified and controlled using the command line. Some of the uses of hping are firewall testing, advanced port scanning, network testing using different protocols, type of service (ToS), fragmentation, manual path maximum transmission unit (MTU) discovery, advanced traceroute under all the supported protocols, remote OS fingerprinting, remote uptime guessing, and TCP/IP stacks auditing.

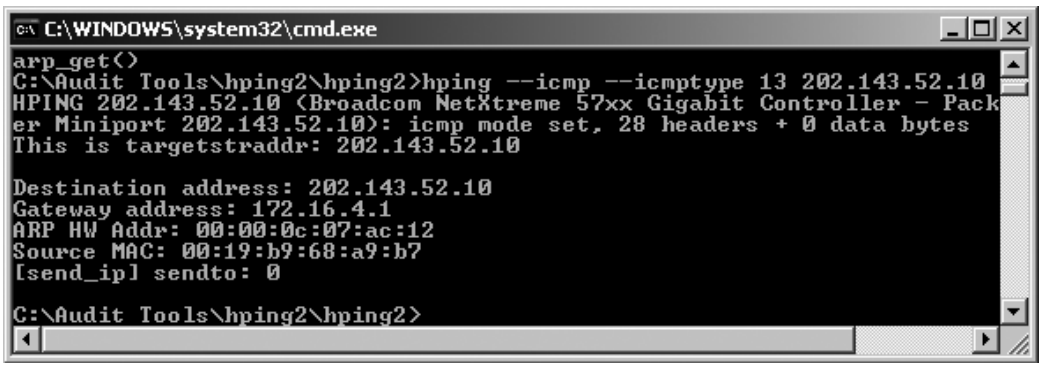
Execute an hping for UDP scan of port 123; see Figure 11.13.

Figure 11.13 hping Scanning of Port 123



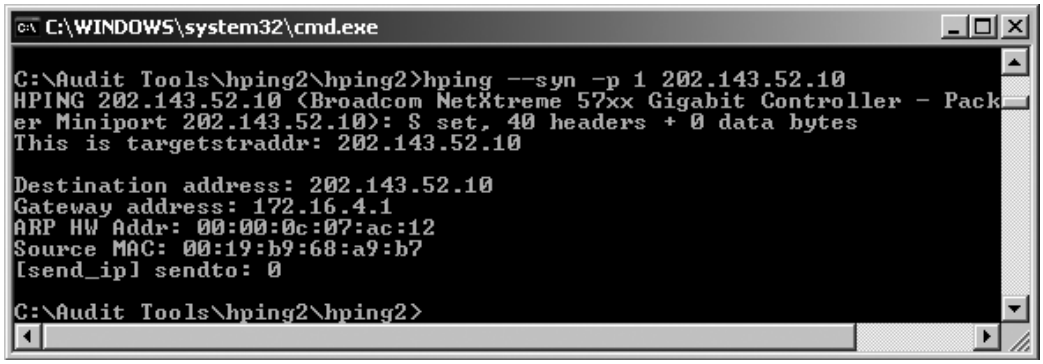
Send an ICMP timestamp request packet (icmptype 13); see Figure 11.14.

Figure 11.14 hping Sending Timestamp Request Packet



Do hping SYN scan of port 1; see Figure 11.15.

Figure 11.15 hping SYN Scanning of Port 1



Change Control

A properly configured firewall rulebase soon becomes weak if it is not given a regular checkup. It comes to no surprise that some firewall administrators configure their firewalls just once and then never worry about it again. New vulnerabilities in both operating systems and firewall software are constantly being discovered. If the firewall operating system and software, including the rulebase, are not being updated, the firewall will not be able to withstand an attack, and would have little claim to *due diligence*, and *reasonable and prudent precautions* in any legal proceedings.

However, changes to the firewall should never be done arbitrarily or on impulse. A proper change management procedure, as part of the overall security policy, is highly recommended. The following information should be included as comments whenever a rule is modified:

- name of person modifying rule
- date/time of rule change
- reason for rule change
- approval from management

The best part here is that this type of check is custom designed to by baselines and placed into an automated check. Why not let the system do the work for you and send an alert when anything changes without going through the correct change process?

Validated Firewalls

Firewall configurations should be validated before they are put into production (a live environment). Validation means checking that the configuration would enable the firewall to perform the security functions that we expect it to do and that it complies with the security policy of the organization. You cannot validate a firewall by looking at the policy alone. The policy is an indicator, but not the true state. The ***only*** way to ensure that a firewall is behaving correctly is to test it using the thing it is set to control, packets. To validate a firewall, you need to fire packets at it.

Validated firewalls need to be constantly monitored for health and stability. Proper change management procedures and policies around the firewall rulebase should be observed at all times. Every time a new rule is made, the firewall should be validated again as a whole, not just for the particular rule that was added or changed.

Abnormal traffic patterns should be investigated immediately. If servers that normally receive a low volume of traffic are suddenly responsible for a significant portion of traffic passing through the firewall (either in total connections or bytes passed), then this might be a situation worthy of further investigation. While sudden peaks and spikes are to be expected in some situations (such as a Web server during a period of unusual interest), sudden peaks and spikes are also often signs of misconfigured systems or maybe even attacks in progress.

Rule violations should be treated as incidents. Looking at traffic denied by your firewall may lead to interesting discoveries, but it is unlikely that even the smallest of organizations could watch all the logs (if they are working). This is especially true for traffic that originates from inside your network. The most common cause of this activity is a misconfigured system or a user who is not aware of traffic restrictions, but analysis of rule violations may also uncover attempts at passing malicious traffic through the device.

Detecting probes originating from inside the trusted network should be performed periodically. These are extremely interesting, as they most likely represent either a compromised internal system seeking to scan Internet hosts or an internal user running a scanning tool, which are both scenarios that merit attention.

Apart from those previously mentioned, firewall log files should be regularly monitored to check for significant events. These fall into three broad categories: critical system issues (such as hardware failures or performance bottlenecks), significant authorized administrative events (ruleset changes, administrator account changes), and network connection logs.

- **Host operating system log messages** For the purposes of this document, we will capture this data at the minimum severity (maximum verbosity) required to record system reboots, which will record other time-critical OS issues, too.
- **Changes to network interfaces** We need to test whether or not the default OS logging captures this information, or if the firewall software records it somewhere. (Is UNIX ifconfig (or the equivalent) invoked?)
- **Changes to firewall policy**
- **Adds/deletes/changes of administrative accounts**
- **System compromises**
- **Network connection logs** The information in these logs includes dropped and rejected connections, time/protocol/IP addresses /usernames for allowed connections, and amount of data transferred.

There are several tools that can automate firewall log monitoring, including such features as real-time alerts and notifications, and customized reports.

Configuration reviews may be mandatory for firewalls that process regulated data. In fact, the Payment Card Industry Data Security Standard (PCI-DSS) requires quarterly firewall reviews for systems involved in payment card processing.

Manual Validation

A manual validation of the rulebase is most effective when done as a team exercise by the security manager, firewall administrator, network architect, and everyone else who has a direct involvement in the administration and management of the organization's network security.

First and foremost, the rulebase should conform to the organization's security policy, hence the recommendation that security managers and administrators be present in the rulebase review.

Prior to the validation, the rulebase should be backed-up to ensure that, if anything goes wrong after implementing changes to the firewall, the previous rulebase can be installed and troubleshooting can be done from there.

In validating the rulebase, unneeded rules should be eliminated. Keeping the rulebase as short and simple as possible conforms to best practices. If there is a rule that everyone is unsure of, it should be removed. The same applies to redundant rules. Some rules can also be grouped together.

While on the topic of best practices, it is recommended that any changes be documented for future reference. Any exceptions to the rules should also be documented, along with an explanation for why these exceptions exist. (This could be a good place to create a baseline for future audits).

Lastly, the rules should be validated for correct order. Rule order is very critical. Most firewalls (such as SunScreen EFS, Cisco IOS, and FW-1) inspect packets sequentially. When a packet is received, it is compared against the first rule, then the second, then the third, and so on. When a matching rule is found, checking is stopped; the rule is applied. If the packet goes through each rule without finding a match, then that packet is denied (or it should be. The only last rule on a firewall that should ever exist is a default drop or reject, this is not always the case).

It is critical to understand that the *first* rule that matches is applied to the packet, not the rule that *best* matches. Based on this, it is recommended that the more specific rules be first, and the more general rules be last. This arrangement of rules prevents a general rule being matched before hitting a more specific rule, helping to protect the firewall from misconfiguration.

Automated Rulebase Validation

There are readily-available tools that perform an analysis of the rulebase by matching it against a standard or benchmark, such as the Router Audit Tool (RAT) and Nipper. (See the router and network devices chapter for separate how-to manuals for RAT and Nipper.) These tools run every rule in the rulebase against known weaknesses and vulnerabilities, and then provide a report at the end, with recommendations on how best to rectify the discovered errors.

Using automated tools is much faster than manual validation and, often as an added feature, can detect whether the latest firewall patches/updates have been installed. However, automated tools do have their limitations. One limitation is that they cannot guarantee that the rulebase is in line with the security policy. In this case, manual validation has an advantage.

Creating Your Checklist

The most important tool that you can have is an up-to-date checklist for your system. This checklist will help define your scope and the processes that you intend to check and validate. The first step in this process involves identifying a good source of information that can be aligned

to your organization's needs. The integration of security check lists and organizational policies with a process of internal accreditation will lead to good security practices and, hence, to effective corporate governance.

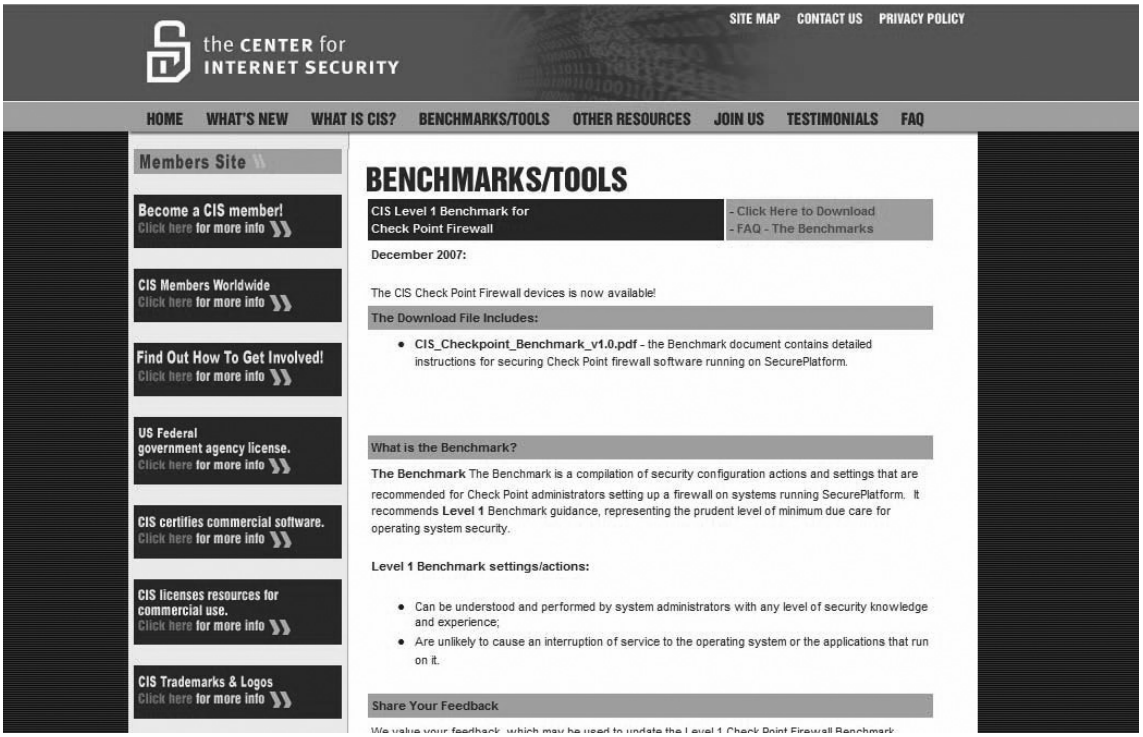
The first stage is to identify the objectives associated with the systems that you seek to audit. Once you have identified the objectives, a list of regulations and standards to which the organization needs to adhere may be collated. The secret is not to audit against each standard, but rather to create a series of controls that ensure you have a secure system. By creating a secure system you can virtually guarantee that you will comply with any regulatory framework.

The following sites offer a number of free checklists that are indispensable in the creation of your firewall audit framework.

CIS (Center for Internet Security)

CIS provides a large number of benchmarks, not only for operating systems, but also for network devices and even firewalls. (CIS is mentioned throughout this book.) CIS offers both benchmarks and tools that may be used to validate a system. The site is www.cisecurity.org. Part of the CIS checklist for checkpoint firewalls is shown in Figure 11.16.

Figure 11.16 CIS Checklist for Checkpoint Firewalls



SANS

The SANS Institute has a wealth of information available that will aid in the creation of a checklist and many documents that detail how to run the various tools.

The SANS reading room (www.sans.org/reading_room/) has a number of papers that have been made freely available:

- GCFW Audit Gold Papers (firewall-specific)
- GCUX UNIX Gold Papers and GCWN Windows Gold Papers (and maybe others)
- general tools papers (www.sans.org/reading_room/whitepapers/tools/)

SANS SCORE (Security Consensus Operational Readiness Evaluation) is directly associated with CIS.

NSA, NIST and DISA

The US government through the National Security Agency (NSA), Defense Information Systems Agency (DISA) and National Institute of Standards and Technology (NIST) has a large number of security configuration guidance papers and benchmarks.

NIST runs the US *National Vulnerability Database* (see http://nvd.nist.gov/chklst_detail.cfm?config_id=58), which is associated with a number of network and operating system Security Checklists from DISA (http://iase.disa.mil/stigs/checklist). These are covered in more detail in each of the sections for the operating systems. (See the UNIX and Windows chapters for more information.)

Summary

Many people and groups such as Gartner (www.gartner.com) have come out stating that firewalls are dead. The truth is that this is far from reality. It may be true that firewalls are changing, but they are an essential component of security. Though protocols such as RPC over HTTP and peer-to-peer networks eat away at the effectiveness of the firewall, allowing traffic inside the network, it is difficult to think about securing a site without a firewall. It is impossible to meet the compliance requirements of any system without one.

It is better and easier to defend a small subset of network traffic and access through a limited number of choke points that to think about everything at once. This is what firewalls have traditionally done, and they still add to the security of any site. An administrator without a firewall is putting out fires. This is where the validation of a firewall is so important. It is not enough to have one; it must be effective. This means auditing and testing.