Auditing Your Firewall Setup

Lance Spitzner
Last Modified: 9 September, 2000

You've just finished implementing your new, shiny firewall. Or perhaps you've just inherited several new firewalls with the company merger. Either way, you are probably curious as to whether or not they are implemented properly. Will your firewalls keep the barbarians out there at bay? Does it meet your expectations? This paper will help you find out. Here you will find a guide on how to audit your firewall and your firewall rulebase. Examples provided here are based on Check Point FireWall-1, but should apply to most firewalls.

Where to Start

This paper can help you in one of two situations. First, you have certain expectations of what your firewall can or cannot do and you want to validate those expectations. Second, you do not know what to expect, so you need to audit your firewall to learn more. Either way, this paper can hopefully help you out. We are not going to cover how to audit or "hack" a network, that is a different subject. Also, we are not going to discuss which firewall is better then others, each firewall has its own advantages and disadvantages. What is going to make or break you is not choosing the "best" firewall, but implementing it correctly. That is the purpose of this paper, making sure our firewall is correctly implemented and behaves as we expected it.

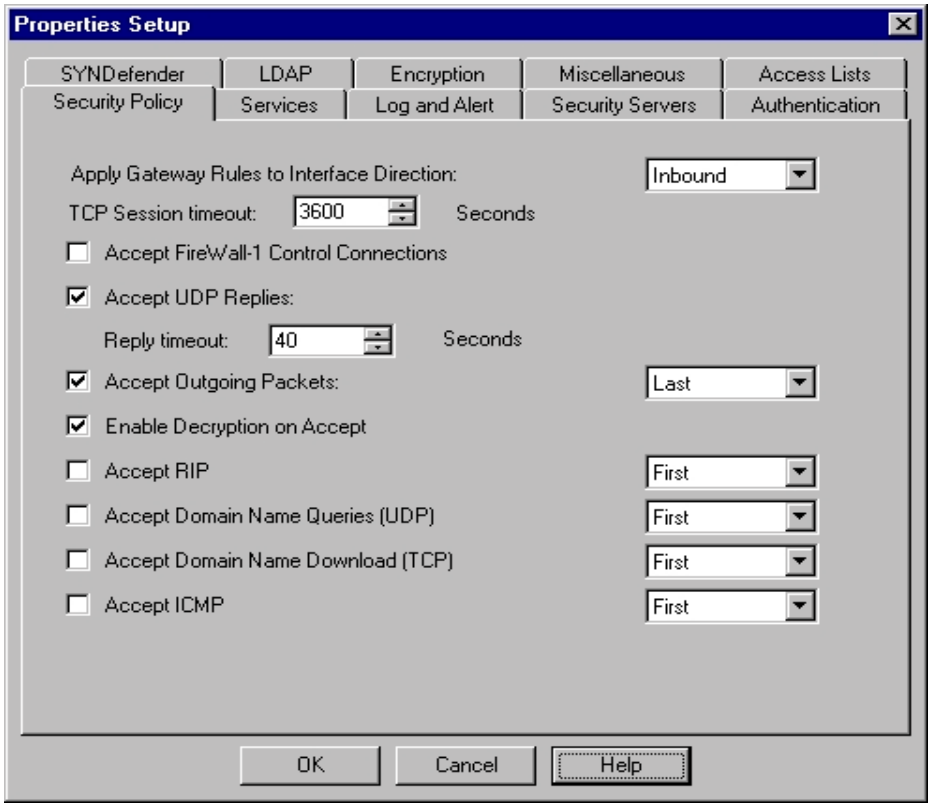
Setting Expectations

Our first step in auditing our firewall is defining what we expect. What do we want our firewall to do? Most of you should have this already defined in the form of a security policy. Make sure you have an understanding of these expectations before you verify your firewall setup. That way, when you are done with the process, you can compare the results to your expectations. Some of you may be in the situation where you don't know what to expect. Maybe you are new to the company and need to assess the situation. Or perhaps your company has just merged and you have assumed responsibility of several new networks. Regardless, try to define some goals before you start, what would you like to see happen.

The Methodology

There are two parts to auditing your firewall setup. First, you want to test the firewall itself. As a critical system in your security plan, you want to ensure this is secure. Second, you want to test the rulebase, what traffic can pass through the firewall? The whole purpose of the firewall is to control traffic, you want to verify it is doing its job.

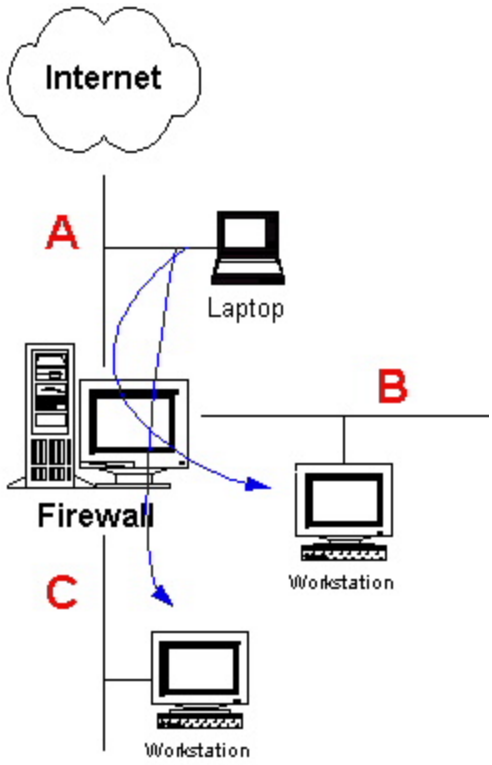
The Firewall. To audit your firewall you want to ensure that it is secure, that no one from the outside or the inside can access or modify your firewall. First, you want to ensure it is physically secured with controlled access. Once someone has physical access, game over. Next, the operating system itself should be fully armored. I recommend you review an armoring checklist specifically designed for your operating system. You need to ensure the operating system fully complies with the armoring checklist. You can find more information about armoring and checklists here for Linux, Solaris, or NT. The next step is to port scan your firewall, from both your internal network and the Internet, scanning for icmp, udp and tcp. We want to identify, what, if any ports are open on your firewall. On most properly configured firewalls, you should find no open ports, you should not even be able to ping it.



A properly armored firewall should have few services to start with. Once the firewall is running, no ports should be exposed unless they absolutely have to. Many of you CheckPoint FireWall-1 users will get a nasty surprise when you find several ports open, such as 256, 257, and 258. These ports are for administration , open by default in the control properties.

I highly recommend you disable them. ICMP is also open by default, I highly recommend you disable this also. If ICMP is open, your network can easily be mapped from the Internet. If you need these ports or services to administer your firewall, then set up a rule that limits what source IPs can connect to them. The idea in securing your firewall is to deny access whenever possible. Every rulebase should have a lockdown rule at the beginning that denies any traffic to the firewall. That way your firewall is "sealed" from the world. If you need access to the firewall, have the rule go before the lockdown rule. All other rules should go after the lockdown rule. Many people consider this a "ghosting" rule, thinking it hides the firewall, it doesn't. What it does do is protect your firewall, ensuring that whatever other rules you put in later, your firewall will still be protected. For example, in the demo rulebase, if you put rule #3 first, now everyone on your internal network has full access to your firewall. The lockdown rule, when placed first, protects you against that. That is the whole purpose of scanning your firewall, ensuring that you have not accidentally exposed your firewall to unauthorized users. To learn more about rulebase design, check out Building Your Firewall Rulebase.

The Rulebase. Once you have audited your firewall, you want to audit your rulebase. The goal is to ensure that the firewall is enforcing what you expect it to, in other words, no surprises can get through the firewall. This is done by scanning every network segment from every other network segment to see what packets can and cannot get through the firewall. You want to validate that the firewall is accepting ONLY the traffic that you allow. You do this by placing your auditing system on one side of the firewall (I use a laptop) and then scan another system on the other side of the firewall. This will determine what packets can and cannot get through. For an example in this diagram, you see the audit laptop (network A), testing the Firewall rulebase by scanning a system on the DMZ (network B) and the internal network (system C).



Many firewalls have several network segments, such as protected DMZs. Make sure you validate your rulebase by scanning from every one of these segment. I highly recommend you position your audit system on your DMZ and attempt to penetrate you internal network. This simulates if one of your DMZ systems is compromised (such as a DNS or web server) and that your internal network is still protected by the firewall. Plan on each scan taking 30-60 minutes. These scans can take a long time due to the timeout period. Many firewalls are configured to deny/drop traffic with no response (i.e. no RST packet or ICMP Error message). This means your scans will take longer, as the scanner does not get immediate feedback, but rather has to timeout on blocked ports. Some scanners, such as nmap, allow you to set the default timeout period.

Remember, your firewall rulebase should deny everything, allowing only that which is specifically allowed. The fewer services you accept and the fewer rule you have, the more secure your environment. If during your audit you are not sure if a service should be blocked, block it. If no one complains, then it was not needed.

Authentication / Encryption: There are several other features you want to test, specifically authentication and encryption. Often firewalls are expected to authenticate users to access a resource. FW-1 has several different authentication options, be sure to test them. For example, if you expect users to be authenticated before they access your website, confirm this for yourself. Try accessing the website without authenticating and see what happens. It is extremely easy to make a mistake when you implement a rulebase. What you thought was password protected may be wide open to the world. Apply the same test for encryption. If you have resources that should only be accessed while encrypted, test it out. Try accessing the resources without encryption, can you get there? Also, run a sniffer such as snoop or tcpdump during the test. Make sure your data is actually being encrypted. You need to verify your expectations. Are your resources protected in the manner you expected?

Additional Services: Firewalls today, such as FW-1, can work with third party software for additional services. For example, virus scanning in email or web content filtering. If you are using any of these 3rd party services with your firewall, you need to test them. For example, for virus scanning, send an infected email through the firewall to ensure your virus scanning is working. If it does not, you will need to review your configuration and resolve the problem. Be sure to re-test the configuration to ensure the fix works.

Digging Deeper: Once you have identified what resources are available, you can begin to dig deeper. You've determined what the firewall allows through, now what threat does that pose? This is where things become fuzzy, where auditing your firewall setup can become auditing your network. You are no longer auditing your firewall, but auditing the resources behind the firewall. However, since this information will be important to you, we will cover the basics . The goal is to determine what potential vulnerabilities exist for the accessible resources. I recommend reviewing each accessible resource and identify what vulnerabilities exist. For example, you determine that the firewall allows http access, in your case to several IIS web servers. Now you have to determine what threats that poses (hint, there are a lot!). Or, you identify a system running ftpd, in your case wu-ftpd 2.4.2 VR17 (in this case, upgrade to the latest version). If a vulnerability exists, you either have to fix the vulnerability, or decide if the risk is worth the service. One of the best resources for identifying vulnerabilities (both Unix and NT) is Bugtraq's vulnerability database at securityfocus.com. I highly recommend you review this database for every resource you have accessible. There are also a variety of tools that will help you identify what vulnerabilities exist. Find several tools you feel the most comfortable with and use this.

Logging: After you have verified your firewall and rulebase, review the firewall logs. Did the firewall detect all of your scans, did it set off the expected alerts? What traffic did it log, and how? If your firewall did not detect most of this activity, something is wrong, you need to be able to see this information. Also, by reviewing the rule base, you will have a better understanding of what to look for in the future when auditing your logs. For FW-1, I always recommend Track Long. If you are going to log the rule, log it long so you get all the information. For more information on logs and alerts with FW-1, check out Intrusion Detection for FW-1.

The Tools

Now comes the fun stuff, the tools. How do we accomplish what we just discussed? One of the best tools for auditing your firewall and firewall rulebase is a good port scanner. As you saw, the biggest priority is identifying what resources are accessible. Personally, I believe one of the best port scanners is Fyodor's nmap. There are two reasons why I believe this. First, no other open source port scanner has more options then nmap, including OS detection, stealth scanning, rpc detection, etc. Second, and just as importantly, this tool is a what many of the bad guys are using. If the black-hat community is running this tool against your firewall, so should you. If you run nmap, I highly recommend you try the Stealth Scan option, such as "-sS" or "-sF" (NOTE: Test stealth scanning on a test remote host first. Some systems panic/crash from a stealth scan). Be sure to verify if your firewall logs can detect the stealth scans (FW-1 ver 4.x should detect all nmap stealth scans). Another option I like is "-g", which lets you set the source port. You can test for misconfigured rules that allow packets based on source ports, such as ftp data (port 20), dns lookups (port 53) or return http traffic (port 80). A new option is "-sA" which is designed specifically to test firewall rulebases. One example of how to run nmap is as follows:

```
mozart #nmap -v -sR -sA -P0 -T Aggressive -o nmap.out <system IP>
```

However, with newer firewalls, such as Firewall-1 ver 4.1SP2, this will not work. Some firewalls will not allow you to build sessions using an ACK packet, but with SYN only. So you have to test your firewall rulebase using SYN packets. For example, with the newest firewalls, I use something like the following:

```
mozart #nmap -v -sS -sR -P0 -O -p1-65000 -o nmap.out <system IP>
```

Notice how in this scan I am scanning all 65,000 possible ports. This will take a looong time, around 60 minutes. However, it is thorough. You may want to play with the '--max_rtt_timeout milliseconds' setting for faster scanning, or use the default

number of ports for scanning (1500). Also, don't forget to scan the UDP ports also (-sU). You will be amazed what you find open there (SNMP, routed, etc). For more information on nmap options and usage, I highly recommend you read the excellent docs. For all your Linux users out there, it comes in rpm format, so you just install and run. Now a days nmap even has a Windows version, so no excuses!

For you Window users, there are several Window GUI options. My personal favorite is WS Ping ProPack. Not only does it include a port scanner, but a variety of other great Unix tools, such as whois, snmpwalk, etc. Unfortunately, the port scanner is not as flexible as you would find in most Unix port scanners.

Once you have identified what resources can be accessed with your port scanner, you can dig deeper. As discussed above, there are a variety of methods and tools to digging deeper. All of these tools are shareware/freeware, so no excuses. Here are several of my favorite.

Nessus (runs on Unix, client can run on 95/NT)	I consider one of the best, free vulnerability scanners.
Whisker (runs on anything that has PERL)	Searches websites for vulnerabilities
Hping2 (runs on Unix)	Build your own ICMP/TCP/UDP packets.
Winfingerprint (runs on 95/NT)	Enumerates NetBIOS Shares, Users, Groups, and Services
legion (runs on 95/NT)	From the guys at Rhino9, scans for smb shares
Sam Spade (runs on 95/NT)	Similar to WS Ping ProPack, but with some different goodies

If you want to learn more about auditing tools, I recommend you check out securityfocus.com tool database. Tool learn more about exploit tools, I recommend you check out technotronic.com exploit tool database.

Conclusion

A firewall is only as good as it's implementation. In today's dynamic world of Internet access, it is easy to make mistakes during the implementation process. By auditing your firewall setup, you can ensure that the firewall is enforcing what you expect it to, in a secure manner.

Author's bio

Lance Spitzner enjoys learning by blowing up his Unix systems at home. Before this, he was an *Officer in the Rapid Deployment Force,* where he blew up things of a different nature. You can reach him at *lance@spitzner.net*