The Use of Firestarter for Securing Linux Computer Systems

Randall E Rausch

for
Partial Fulfillment
of the requirements
for the
SANS GSEC
Certification
Version 1.4b

September 14, 2003

Abstract—

In this paper the use of the Linux security utility Firestarter[1] is described. Port scans are performed on a Red Hat Linux system to establish a baseline. Firestarter is then installed and configured. The hardened system is then scanned again to determine the change. The "out of the box" system has a number of open and unfiltered ports – presenting vulnerabilities which could be maliciously exploited. All ports on the hardened system are filtered -- placing the system in a more secure state, since no information about the state of the system is being "leaked" to the outside. Additional resources – the Firestarter homepage and online manual -- and features -- interactive rule construction -- are also described.

[1] http://firestarter.sourceforge.net/

Introduction

Linux is rapidly becoming a household word as sophisticated users turn to this UNIX platform for personal computers. Because of the long history of Unix and its long use in a variety of technical, engineering and other settings, more approaches towards invasion have been developed in this application than any other. These are now being used on other systems. Linux is also the system used by most hackers. This makes the availability and functionality of security systems in this venue of premier importance. This study analyzes and quantifies the effectiveness of one of these – Firestarter.[2]

The use of a home network in this study is deliberate – as well as convenient. It is becoming more common that home computers including Linux systems are being used in distributed denial of service attacks. In addition, most domestic computer users would prefer a higher level of security. This study exposes and addresses the weaknesses of any small network.

Methodology

The study follows this course of action:

0. Redhat 9 was installed on an Intel-compatible computer (a clone with ASUS motherboard and an AMD Athlon 1800 MHz processor.)
1. This "out of the box" installation was scanned using nmap to determine a baseline.
2. Firestarter was then installed and configured using (as much as possible) the Firestarter defaults settings.
3. The system was then scanned again to determine how the how the system changed by having Firestarter installed.

In addition to this before and after analysis, other features for analyzing hits and configuring rules are also described.

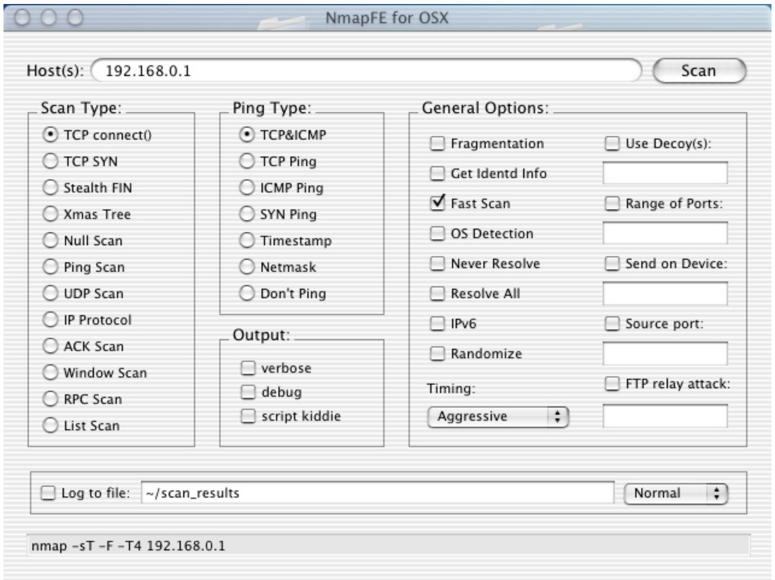
Before: the "out of the box" base-line

After performing the default installation of Redhat 9 a port scan of the system was performed from another system on the local network using nmap. Nmap is a powerful port scanning utility available from http://www.insecure.org/nmap/. Nmap is available for most platforms (the system used for these port scans was an Apple Macintosh G4 running OS10.2.6.) On UNIX systems nmap is a command line accessible binary with an associated GUI, nmapFE. There is also a windows version, nmapwin, according to the website.[3]

Nmap is powerful and as the following screen grab shows quite flexible. It is also possible to use nmap to determine -- with some degree of accuracy-- the type of

[2] http://firestarter.sourceforge.net/
[3] Fyodor, Nmap Website, http://www.insecure.org/nmap/

OS that a specified host uses. A typical "script kiddy" attack is to scan a wide range of ip addresses and determine for those addresses scanned the OS in use on as many as possible. When a vulnerability is announced it is then possible to search this list for the vulnerable OS and to then mount an attack.


NmapFE for OSX

Host(s): 192.168.0.1 Scan

Scan Type:
● TCP connect()
○ TCP SYN
○ Stealth FIN
○ Xmas Tree
○ Null Scan
○ Ping Scan
○ UDP Scan
○ IP Protocol
○ ACK Scan
○ Window Scan
○ RPC Scan
○ List Scan

Ping Type:
● TCP&ICMP
○ TCP Ping
○ ICMP Ping
○ SYN Ping
○ Timestamp
○ Netmask
○ Don't Ping

Output:
□ verbose
□ debug
□ script kiddie

General Options:
□ Fragmentation
□ Get Identd Info
☑ Fast Scan
□ OS Detection
□ Never Resolve
□ Resolve All
□ IPv6
□ Randomize
□ Use Decoy(s):
□ Range of Ports:
□ Send on Device:
□ Source port:
□ FTP relay attack:
Timing: Aggressive

□ Log to file: ~/scan_results Normal

```
nmap -sT -F -T4 192.168.0.1
```

Figure 1: Screen Grab of NmapFE for OS 10.2

I will give here a brief description of each type of scan and the results of scanning the "out of the box" system. Look at the website http://weadmin.com/satish/talk/port_scanning.html [4] and http://weadmin.com/satish/talk/scan_responses.html[5] for a nice description of the scans and the possible outcomes for each. When discussing port scanning it is good to have these definitions in mind

Open - The port will accept connections.
Filtered – The port scanner cannot determine if the port is open, typically because an intervening firewall or packet filter is preventing the packets from having access to the port.
Unfiltered – The port scanner can determine that the port is closed and no firewall or packet filter is interfering with the packets en route.
Closed – the port is not accepting connections.

> **TCP connect** - This scan connects to the targeted port and attempts to complete the full SYN, SYN/ACK, and ACK three-way handshake by sending a SYN. Possible outcomes are open a SYN/ACK is received as response, closed if a RST is received and filtered if no response or ICMP unreachable is the reply. The result of the nmap TCP connect scan was:

[4] Port Scanning, http://weadmin.com/satish/talk/port_scanning.html
[5] Ports, scan responses and ICMP, http://weadmin.com/satish/talk/scan_responses.html

```
     # nmap 3.26 scan initiated Sun Jul 13 12:33:41 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sT -oN /Users/bart/Scan Results/scan.RH.T -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
(The 65534 ports scanned but not shown below are in state: closed)
Port       State       Service
32768/tcp  open        unknown

# Nmap run completed at Sun Jul 13 12:51:01 2003 -- 1 IP address (1
host up) scanned in 1040.202 seconds
```

A bit of investigation on the Internet reveals that this port, 32768, is used by Redhat Linux for rpc calls[6]. This presents a serious vulnerability which should be addressed by disabling the service nfslock in the initialization script, rcN.d (where N is 3, 4, or most likely 5 depending on whether or not your system is running X windows) or by filtering the port to accept connections only from hosts whose ip addresses are specified (known hosts on your own local network, for example.) If you do not need to share file systems (hard disks) across your network then this service should be disabled.

It is interesting at this point to compare the results of this port scan to those obtained by running netstat on the Linux system. Let's show the first block of output of netstat –a:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:32768                 *:*                     LISTEN
tcp        0      0 localhost:32769         *:*                     LISTEN
tcp        0      0 *:sunrpc                *:*                     LISTEN
tcp        0      0 *:x11                   *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
tcp        0      0 localhost:ipp           *:*                     LISTEN
tcp        0      0 localhost:smtp          *:*                     LISTEN
tcp       10      0 linux:35993             xmlrpc.rhn.redhat:https CLOSE_WAIT
```

And of netstat –an:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:32768           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:32769         0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN
```

[6] http://www.securityfocus.com/archive/91/210923/2001-08-27/2001-09-02/0

```
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
tcp       10      0 192.168.0.1:35993       66.187.232.101:443      CLOSE_WAIT
```

Now let's analyze this output. The first entry listening on port 32768 is the Redhat nfs service we've seen in our port scan. The next entry is for a local service presumably related to this nfs service because of the consecutive port numbers. The next three sunrpc at port 111, X11 at port 6000 and ssh at port 22 are puzzling until we do some digging in the Redhat online support for the built-in firewall and find that for the default level, medium, access to the following ports is not allowed: ports lower than 1023, the NFS server port (2049), and the local X Window System display for remote X clients[7]. Next are a couple of local services, ipp at port 631 for printing and smtp at port 25 for mail. Next is a secure http connection (port 443) to the address 66.187.232.101, this connection is part of Redhat's online update system. The security certificate supplied with the distribution has recently gone out of date accounting for the CLOSE_WAIT status – close wait indicates the connection has been closed on the remote end. This remote connection could represent some sort of security vulnerability, but it is part or our "out of the box" configuration, so we'll let it pass for now – planning to investigate more fully in the future what service this connection provides and what vulnerability it presents.

The important issue is that one should use more than one tool and that the result of those tools should mesh together in a coherent fashion. Going through the exercise of reconciling the results of nmap and netstat forced me to do some research and find that the default security level was blocking connections to some of the ports that seem like they should be open according to netstat.

TCP sync - Also called half-open scanning, a SYN is sent to the targeted port. If a SYN/ACK is received in reply this indicates the port is listening while a RST/ACK indicates it is not, no response or IMCP unreachable means the port is filtered. A RST/ACK is sent back by the scanner so that a connection is not created. This is a stealthier and faster scan than the TCP connect and less likely to be logged. Results of the TCP sync can were:

```
# nmap 3.26 scan initiated Sun Jul 13 12:57:08 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for
 OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sS -oN /Users/bart/Scan Results/scan.RH.S -p 1-65535 -T4
192.168.0.1
```

[7] http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/s1-firewallconfig.html

```
Interesting ports on 192.168.0.1:
(The 64499 ports scanned but not shown below are in state: closed)
Port       State       Service
1/tcp      filtered    tcpmux
2/tcp      filtered    compressnet
3/tcp      filtered    compressnet
4/tcp      filtered    unknown
5/tcp      filtered    rje
(all ports between 5 and 1021 were also filtered)
1021/tcp   filtered    unknown
1022/tcp   filtered    unknown
1023/tcp   filtered    netvenuechat
2049/tcp   filtered    nfs
6000/tcp   filtered    X11
6001/tcp   filtered    X11:1
6002/tcp   filtered    X11:2
6003/tcp   filtered    X11:3
6004/tcp   filtered    X11:4
6005/tcp   filtered    X11:5
6006/tcp   filtered    X11:6
6007/tcp   filtered    X11:7
6008/tcp   filtered    X11:8
6009/tcp   filtered    X11:9
7100/tcp   filtered    font-service
32768/tcp  open        unknown
```

Filtered indicates that nmap is unable to determine if the port is opened or closed. Basically, these results fall inline with those of the TCP connect scan – we can't determine much about any port except 32768 which is open.

Stealth FIN - This scan sends a FIN packet to the target. The response should be a RST for all closed ports. No response indicates that a port is open or stealthed. An IMCP port unreachable indicates the port is filtered. Results of this scan were:

```
# nmap 3.26 scan initiated Sun Jul 13 13:01:33 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for
 OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sF -oN /Users/bart/Scan Results/scan.RH.F -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
```

```
6000/tcp   open        X11
32768/tcp  open        unknown

# Nmap run completed at Sun Jul 13 13:02:16 2003 -- 1 IP address (1
host up) scanned in 43.578 seconds
```

These results are more interesting because now we are seeing some of the ports, 22, 111, and 6000, that the default firewall is blocking in the TCP connect scan – a hacker might be able to exploit this information. The next two scans – the xmas tree and null – give the same results providing corroboration of the stealth fin.

Xmas Tree - This scan sends a FIN, URG and PUSH packet to the target. Like the FIN scan, the reply should be a RST for all closed ports. No response indicates that a port is open or stealthed. An IMCP port unreachable indicates the port is filtered. The result for this system was:

```
# nmap 3.26 scan initiated Sun Jul 13 13:12:02 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for
 OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sX -oN /Users/bart/Scan Results/scan.RH.X -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
6000/tcp   open        X11
32768/tcp  open        unknown

# Nmap run completed at Sun Jul 13 13:12:46 2003 -- 1 IP address (1
host up) scanned in 43.660 seconds
```

Null - This scan sends a packet with all flags set to off. Responses are similar to those of the Xmas tree and FIN. Results were:

```
# nmap 3.26 scan initiated Sun Jul 13 13:14:37 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for
 OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sN -oN /Users/bart/Scan Results/scan.RH.N -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
(The 65531 ports scanned but not shown below are in state: closed)
```

```
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
6000/tcp   open        X11
32768/tcp  open        unknown

# Nmap run completed at Sun Jul 13 13:15:20 2003 -- 1 IP address (1
host up) scanned in 42.347 seconds
```

Ping - This scan send an ICMP packet (ICMP_ECHO, type 8) to the host. If the host is up and open to ICMP requests then an ICPM_ECHO_REPLY, type 0 will be sent as a response. This type of scan is useful for determining which hosts on a network up. This scan was not performed on this host.

UDP - This scan sends a UDP packet to the targeted port. If the reply is "ICMP port unreachable" then the port is closed. No response indicates the port is open or that the packet didn’t make it back – UDP scanning has a lot of false positives. Results were:

```
# nmap 3.26 scan initiated Sun Jul 13 13:15:36 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for
 OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sU -oN /Users/bart/Scan Results/scan.RH.U -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
Port       State       Service
1/udp      open        tcpmux
2/udp      open        compressnet
3/udp      open        compressnet
4/udp      open        unknown
5/udp      open        rje
6/udp      open        unknown
7/udp      open        echo
8/udp      open        unknown
9/udp      open        discard
10/udp     open        unknown
11/udp     open        systat
12/udp     open        unknown
13/udp     open        daytime
14/udp     open        unknown
15/udp     open        unknown
16/udp     open        unknown
17/udp     open        qotd
```

```
18/udp    open       msp
19/udp    open       chargen
20/udp    open       ftp-data
21/udp    open       ftp
22/udp    open       ssh
23/udp    open       telnet
24/udp    open       priv-mail
25/udp    open       smtp
26/udp    open       unknown
27/udp    open       nsw-fe
28/udp    filtered   unknown
29/udp    open       msg-icp
30/udp    open       unknown
(all ports from 30 to 34056)
34056/udp  open       unknown
34057/udp  open       unknown
34058/udp  open       unknown
34059/udp  open       unknown
```

Only port 28 was filtered all others were open .

IP protocol - This scan type is used to determine which IP protocols are supported on a given host. Since this method is similar to UDP port scanning it was not performed

ACK - With the ACK flag set the attacker is hoping to trick the packet-filtering device into thinking this is a return packet of an existing connection. If the target machine returns a RST as our target did, it means the packet got through the packet-filtering device or no packet-filter device exists and the target port is classified as unfiltered. If no response or an ICMP host unreachable is returned to the attacker than the target port will be classified as filtered[8].

```
# nmap 3.26 scan initiated Sun Jul 13 15:29:52 2003 as:
/Volumes/data/bart/Desktop
/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sA -oN /Users/bart/Scan Results/scan.RH.ack -p 1-65535 -T4
192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: Unfiltered

# Nmap run completed at Sun Jul 13 15:30:36 2003 -- 1 IP address (1
host up) scanned
```

[8] Neil Warner, Scan 23, South Florida Honeynet Project, http://honeynet.hackers.nl/scans/scan23/sol/Neil.html

Window - This scan is similar to the ACK scan, but uses the TCP window size parameter to gain information about the target. It is primarily useful against only those systems that have anomalies in the way this parameter is handled, examples are AIX, Amiga, BeOS, BSDI, Cray, Tru64 UNIX, DG/UX, OpenVMS, Digital UNIX, FreeBSD, HP-UX, OS/2, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, QNX, Rhapsody, SunOS 4.X, Ultrix, VAX, and VxWorks.[9][10] The results of this scan were:

```
# nmap 3.26 scan initiated Sun Jul 13 15:37:17 2003 as:
/Volumes/data/bart/Desktop
/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sW -oN /Users/bart/Scan Results/scan.RH.window -p 1-65535 -
T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: closed

# Nmap run completed at Sun Jul 13 15:38:01 2003 -- 1 IP address (1
host up) scanned in 43.930 seconds
```

RPC - This scan is used to identify (on UNIX systems) remote procedure call ports, e.g. those handling nfs packets. NFS (network file system) can present a serious vulnerability. Results were:

```
# nmap 3.26 scan initiated Sun Jul 13 15:38:23 2003 as:
/Volumes/data/bart/Desktop
/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sR -oN /Users/bart/Scan Results/scan.RH.rpc -p 1-65535 -T4
192.168.0.1
Interesting ports on 192.168.0.1:
(The 64499 ports scanned but not shown below are in state: closed)
Port       State       Service (RPC)
1/tcp      filtered    tcpmux
2/tcp      filtered    compressnet
3/tcp      filtered    compressnet
4/tcp      filtered    unknown
5/tcp      filtered    rje
6/tcp      filtered    unknown
7/tcp      filtered    echo
8/tcp      filtered    unknown
9/tcp      filtered    discard
10/tcp     filtered    unknown
```

[9] http://lists.insecure.org/lists/nmap-hackers/2000/Jan-Mar/0095.html
[10] http://security.rbaumann.net/scans.php?sel=1

```
11/tcp     filtered    systat
1019/tcp   filtered    unknown
1020/tcp   filtered    unknown
1021/tcp   filtered    unknown
1022/tcp   filtered    unknown
1023/tcp   filtered    netvenuechat
2049/tcp   filtered    nfs
6000/tcp   filtered    X11
6001/tcp   filtered    X11:1
6002/tcp   filtered    X11:2
6003/tcp   filtered    X11:3
6004/tcp   filtered    X11:4
6005/tcp   filtered    X11:5
6006/tcp   filtered    X11:6
6007/tcp   filtered    X11:7
6008/tcp   filtered    X11:8
6009/tcp   filtered    X11:9
7100/tcp   filtered    font-service
32768/tcp  open          (status V1)
```

List - This method simply generates and prints a list of IPs/Names without actually pinging or port scanning them.[11] DNS name resolution will be performed unless you use -n. This scan was not performed.

OS Detection – This scan gathers various bits of sequence number information, open ports and the like and attempt to identify the OS running on the system. For our system the results were:

```
Starting nmap 3.26 ( www.insecure.org/nmap/ ) at 2003-09-17 18:53
CDT
Insufficient responses for TCP sequencing (0), OS detection may be
less accurate
Insufficient responses for TCP sequencing (0), OS detection may be
less accurate
Insufficient responses for TCP sequencing (0), OS detection may be
less accurate
Interesting ports on 192.168.0.1:
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
6000/tcp   open        X11
```

[11] Neil Warner, Scan 23, South Florida Honeynet Project, http://honeynet.hackers.nl/scans/scan23/sol/Neil.html

```
32768/tcp  open        unknown
No exact OS matches for host (If you know what OS is running on it,
see http://www.insecure.org/cgi-bin/nmap-submit.cgi).
```

As an aside it is interesting that this inability to fingerprint the OS is due to the default Redhat firewall. If we run with no firewall then our OS detection gives

```
Starting nmap 3.26 ( www.insecure.org/nmap/ ) at 2003-09-17 18:59
CDT
Insufficient responses for TCP sequencing (3), OS detection may be
less accurate
Interesting ports on 192.168.0.1:
(The 65531 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
111/tcp    open        sunrpc
6000/tcp   open        X11
32768/tcp  open        unknown
Remote operating system guess: Linux 2.4.7 (X86)

Nmap run completed -- 1 IP address (1 host up) scanned in 85.179
seconds
```

Analysis

There are many ports on this system that are not closed and that need not be open, the most noteworthy being port 32768. If there is reason for a port to be open because some service needs to be provided then the vulnerability incurred as the result of that open port is part of "the price of doing business," but as this system represents the typical home system providing no services to the world there is no reason for this vulnerability – in particular if no systems on the local network are going to be sending rpc calls then port 32768 should be closed. The next section will show how the firewall utility firestarter is installed and how it makes the system more secure.

Installing Firestarter

What is Firestarter? I will let the developers speak for themselves:

"Firestarter in a Nutshell:
Firestarter is a free firewall tool for Linux machines. Whether you simply want to protect your personal workstation or you have a network of computers to secure,

Firestarter is here to make your life easier. While a firewall can not guarantee security, it is the first line of defense against network based attacks."[12]

Firestarter is available for free download from Sourceforge and is easily installed from the root prompt by the command

```
#rpm -Uvh firestarter*rpm
```

(Installation was easy on the already described Redhat 9. Attempting to install under SuSE 8.1was not successful, some initscripts were not present.)

Once installed firestarter can be started by typing

```
%firestarter
```

The user is then prompted for the root password and configuration begins. Configuration is started by clicking on the "wizard" button (upper left in figure 2). This opens a set of windows that guide the user through a set of questions. The first has basic instructions.

The second window asks the user to select the network interface card that will be configured. At this point the user can also specify whether the firewall should be started on dial-up for those with dial-up based Internet access and to specify IP addresses are assigned by DHCP. See figure 4.

The next window is the internet connection sharing setup, see figure 5, where network address translation is enabled or disabled. To configure a host as a firewall between a private network and the internet, two network interface cards are required, one would be connected to the internet the other to the private network. The interface connected to the private network would be selected in this step by enabling NAT. In this exercise we are configuring a standalone system connected to private network, so NAT is disabled.

[12] http://firestarter.sourceforge.net/

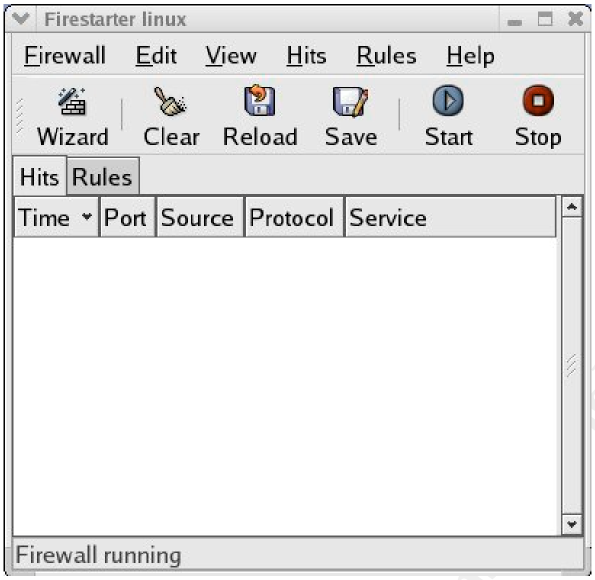
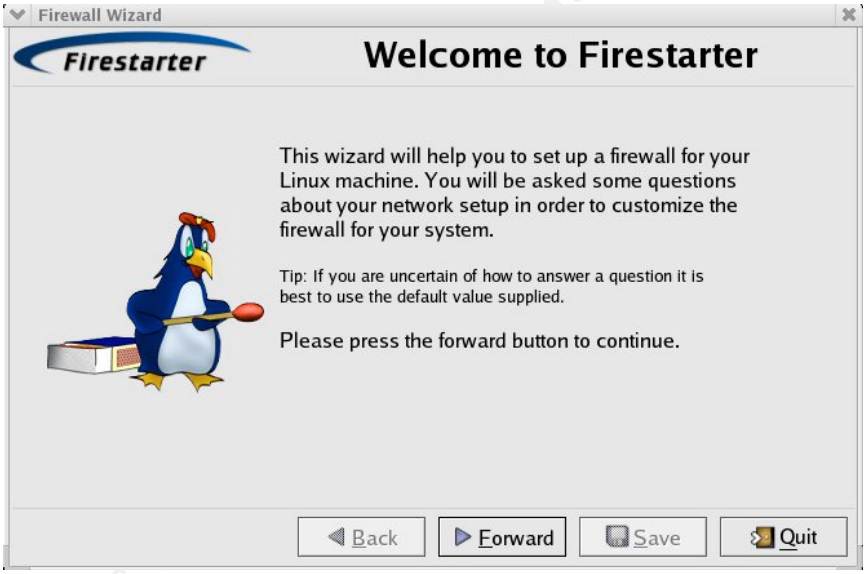


Figure 2: The Firestarter Main Window



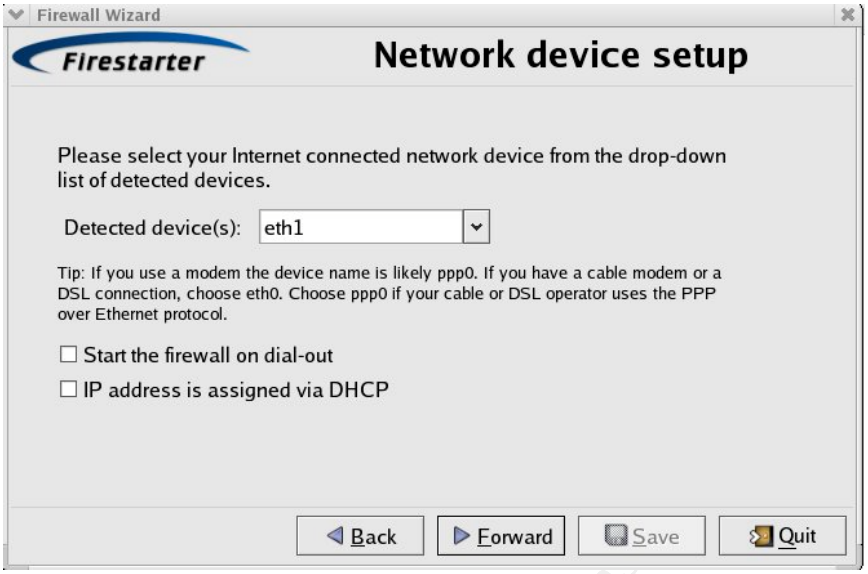
Welcome to Firestarter

This wizard will help you to set up a firewall for your Linux machine. You will be asked some questions about your network setup in order to customize the firewall for your system.

Tip: If you are uncertain of how to answer a question it is best to use the default value supplied.

Please press the forward button to continue.

Figure 3: The Firestarter Wizard



Network device setup

Please select your Internet connected network device from the drop-down list of detected devices.

Detected device(s): eth1

Tip: If you use a modem the device name is likely ppp0. If you have a cable modem or a DSL connection, choose eth0. Choose ppp0 if your cable or DSL operator uses the PPP over Ethernet protocol.

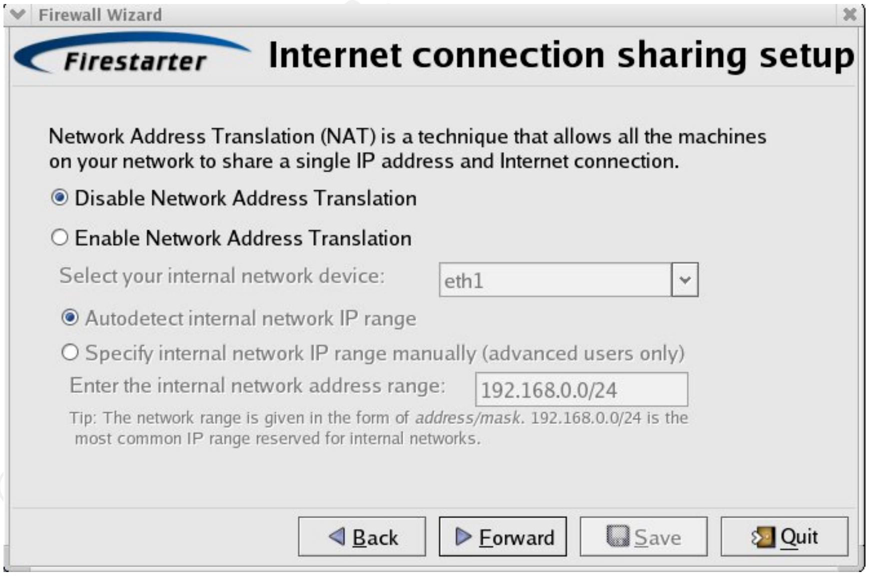
□ Start the firewall on dial-out

□ IP address is assigned via DHCP

Figure 4: Network Device Selection

The next configuration window is type of service filtering setup. This filtering prioritizes network services based on packet type, something that would be useful for a server. We will keep this filtering disable.

The last wizard window is ICMP filtering. To keep our configuration as simple as possible we shall choose to disable.



Internet connection sharing setup

Network Address Translation (NAT) is a technique that allows all the machines on your network to share a single IP address and Internet connection.

● Disable Network Address Translation

○ Enable Network Address Translation

Select your internal network device: eth1

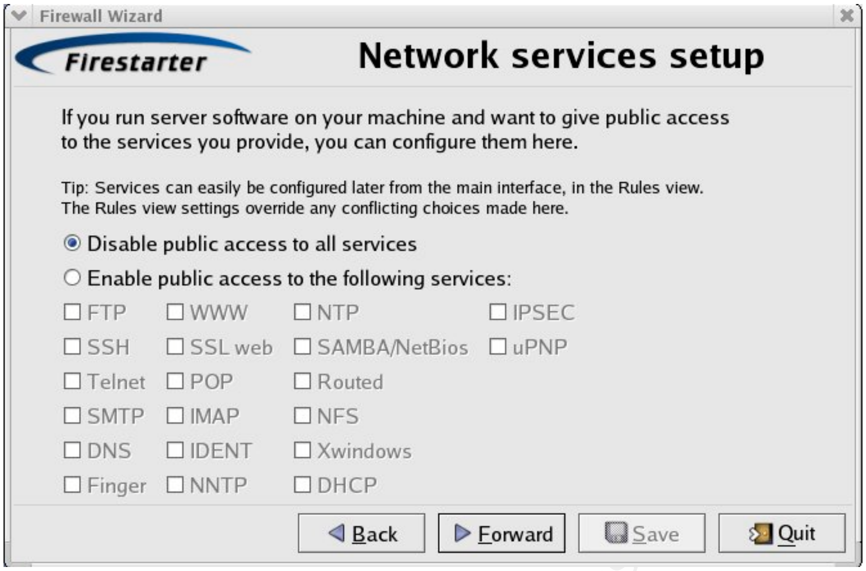
● Autodetect internal network IP range

○ Specify internal network IP range manually (advanced users only)

Enter the internal network address range: 192.168.0.0/24

Tip: The network range is given in the form of *address/mask*. 192.168.0.0/24 is the most common IP range reserved for internal networks.

Figure 5: Internet Connection Sharing Setup



Firewall Wizard

Firestarter

Network services setup

If you run server software on your machine and want to give public access to the services you provide, you can configure them here.

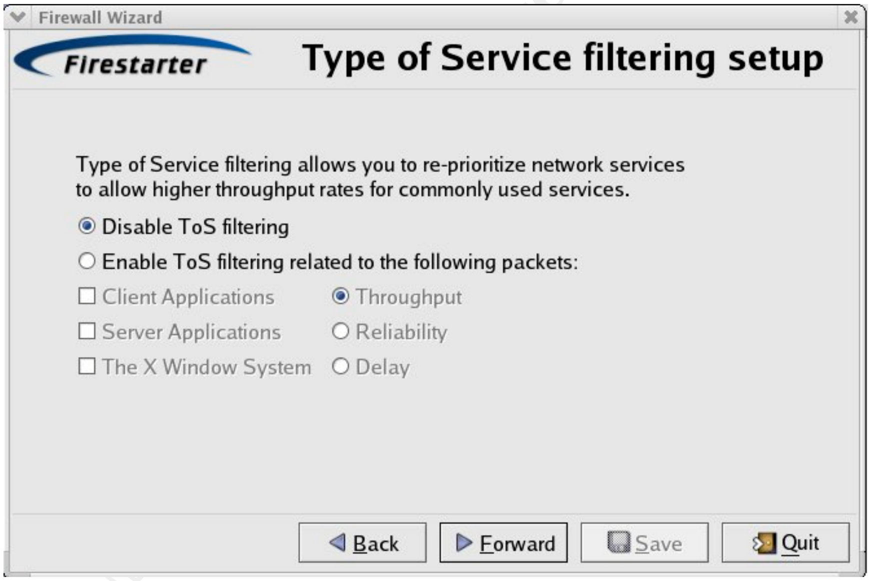
Tip: Services can easily be configured later from the main interface, in the Rules view. The Rules view settings override any conflicting choices made here.

● Disable public access to all services

○ Enable public access to the following services:

□ FTP	□ WWW	□ NTP	□ IPSEC
□ SSH	□ SSL web	□ SAMBA/NetBios	□ uPNP
□ Telnet	□ POP	□ Routed	
□ SMTP	□ IMAP	□ NFS	
□ DNS	□ IDENT	□ Xwindows	
□ Finger	□ NNTP	□ DHCP	

Figure 6: Network Services Setup



Firewall Wizard

Firestarter

Type of Service filtering setup

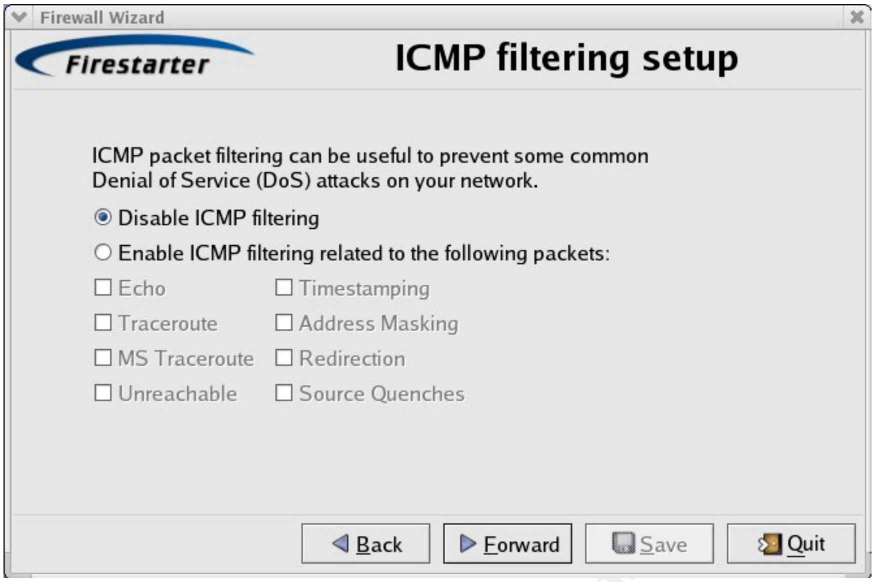
Type of Service filtering allows you to re-prioritize network services to allow higher throughput rates for commonly used services.

● Disable ToS filtering

○ Enable ToS filtering related to the following packets:

□ Client Applications	● Throughput
□ Server Applications	○ Reliability
□ The X Window System	○ Delay

Figure 7: Type of Service Filtering



ICMP filtering setup

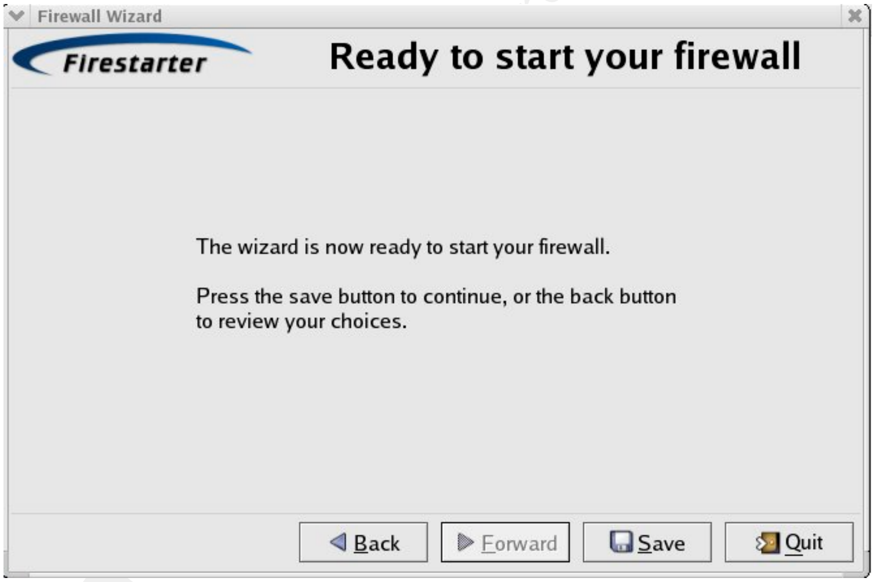
ICMP packet filtering can be useful to prevent some common Denial of Service (DoS) attacks on your network.

● Disable ICMP filtering

○ Enable ICMP filtering related to the following packets:

□ Echo □ Timestamping
□ Traceroute □ Address Masking
□ MS Traceroute □ Redirection
□ Unreachable □ Source Quenches

Figure 8: ICMP Filtering Setup



Ready to start your firewall

The wizard is now ready to start your firewall.

Press the save button to continue, or the back button to review your choices.

Figure 9: Firestarter Wizard Save Window

The configuration is now done. Clicking the save button will write the configuration to /etc/firestarter/firestarter.sh. In order to have the firewall enabled automatically during the boot process this script should be added to the /etc/rc.d/rc.local startup script.

Now that Firestarter has been installed let's look at the effect on the security of the system.

After: the Post Installation Analysis

We repeated the port scans performed before installing Firestarter. These results are shown below.

Stealth FIN

```
# nmap 3.26 scan initiated Sun Jul 13 16:53:47 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sF -oN /Users/bart/Scan Results/post/scan.RH.F.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 18:01:14 2003 -- 1 IP address (1
host up) scanned in 4046.680 seconds
```

Null

```
# nmap 3.26 scan initiated Sun Jul 13 19:37:54 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sN -oN /Users/bart/Scan Results/post/scan.RH.N.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 20:45:20 2003 -- 1 IP address (1
host up) scanned in 4046.022 seconds
```

RPC

```
# nmap 3.26 scan initiated Mon Jul 14 02:32:26 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sR -oN /Users/bart/Scan Results/post/scan.RH.R.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Mon Jul 14 02:52:09 2003 -- 1 IP address (1
host up) scanned in 1182.543 seconds
```

TPC Sync

```
# nmap 3.26 scan initiated Sun Jul 13 16:29:34 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sS -oN /Users/bart/Scan Results/post/scan.RH.S.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered
```

```
# Nmap run completed at Sun Jul 13 16:49:16 2003 -- 1 IP address (1
host up) scanned in 1182.258 seconds
```

TCP Connect

```
# nmap 3.26 scan initiated Sun Jul 13 16:06:55 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sT -oN /Users/bart/Scan Results/post/scan.RH.T -p 1-65535 -
T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 16:26:25 2003 -- 1 IP address (1
host up) scanned in 1169.812 seconds
```

UDP

```
# nmap 3.26 scan initiated Sun Aug 10 15:28:05 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sU -oN /Users/bart/Scan Results/post/scan.RH.U.post -v -F -T5
192.168.0.1
All 1001 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Aug 10 15:28:38 2003 -- 1 IP address (1
host up) scanned in 32.527 seconds
```

Xmas Tree

```
# nmap 3.26 scan initiated Sun Jul 13 18:21:27 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sX -oN /Users/bart/Scan Results/post/scan.RH.X.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 19:28:52 2003 -- 1 IP address (1
host up) scanned in 4045.061 seconds
```

ACK

```
# nmap 3.26 scan initiated Sun Jul 13 20:50:25 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sA -oN /Users/bart/Scan Results/post/scan.RH.ack.post -p 1-
65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 21:10:14 2003 -- 1 IP address (1
host up) scanned in 118
```

```
9.316 seconds
```

Window

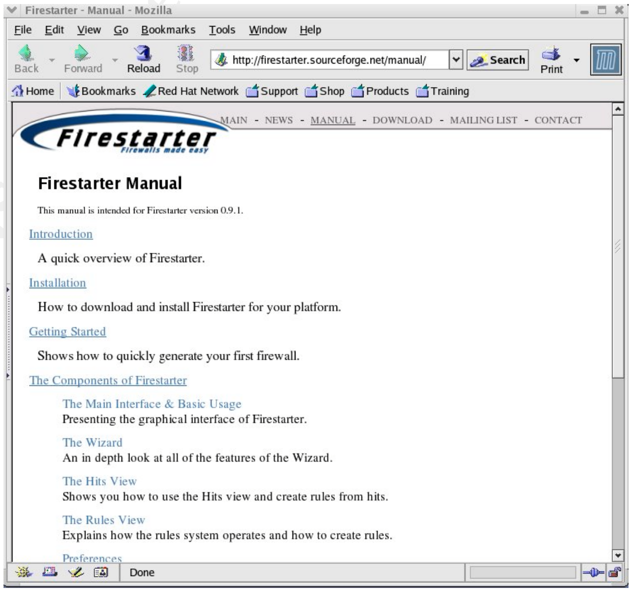
```
# nmap 3.26 scan initiated Sun Jul 13 22:17:39 2003 as:
/Volumes/data/bart/Desktop/portscanners/NmapFE for OSX/NmapFE
Launcher.app/Contents/Resources/NmapFE.app/Contents/Resources/
nmap -sW -oN /Users/bart/Scan Results/post/scan.RH.window.post -p
1-65535 -T4 192.168.0.1
All 65535 scanned ports on 192.168.0.1 are: filtered

# Nmap run completed at Sun Jul 13 22:37:21 2003 -- 1 IP address (1
host up) scanned in 1182.114 seconds
```

We can see now that all ports are filtered against all scans performed. What this means is that nmap is unable to determine if the port is open or closed. No information about the system is leaked to the outside -- our system is now more secure.

Additional Features and Resources

Let me mention briefly two valuable resources for Firestarter. One is the Firestarter Manual available online at http://firestarter.sourceforge.net/manual/, this manual walks the user through configuration, it is clearly written and easy to understand. The other resource is the Firestarter homepage http://firestarter.sourceforge.net/ which you will already have visited to download the package. The homepage has the usual information: package description, download links, and system requirements.


Firestarter - Manual - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://firestarter.sourceforge.net/manual/
Search Print
Home Bookmarks Red Hat Network Support Shop Products Training
MAIN - NEWS - MANUAL - DOWNLOAD - MAILING LIST - CONTACT
Firestarter
Firewalls made easy
Done

Firestarter Manual

This manual is intended for Firestarter version 0.9.1.

Introduction

A quick overview of Firestarter.

Installation

How to download and install Firestarter for your platform.

Getting Started

Shows how to quickly generate your first firewall.

The Components of Firestarter

The Main Interface & Basic Usage
Presenting the graphical interface of Firestarter.

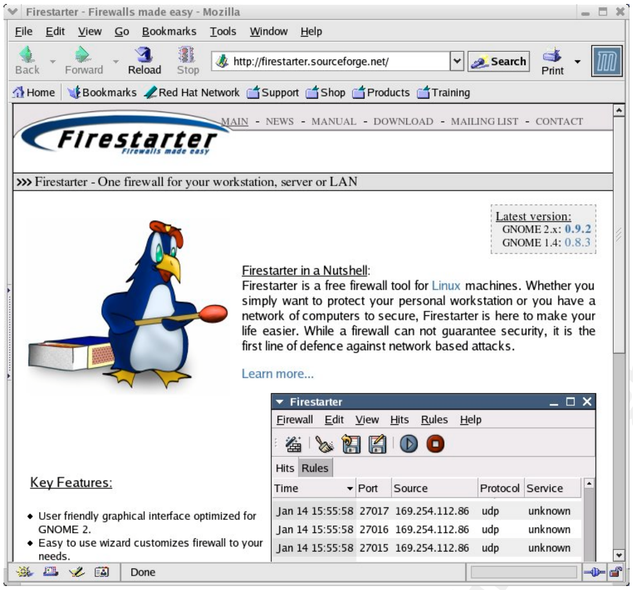
The Wizard
An in depth look at all of the features of the Wizard.

The Hits View
Shows you how to use the Hits view and create rules from hits.

The Rules View
Explains how the rules system operates and how to create rules.

Preferences

Figure 10: The Firestarter Manual


Firestarter - Firewalls made easy - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://firestarter.sourceforge.net/
Search
Print
Home Bookmarks Red Hat Network Support Shop Products Training
MAIN - NEWS - MANUAL - DOWNLOAD - MAILING LIST - CONTACT
Firestarter
Firewalls made easy

>>> Firestarter - One firewall for your workstation, server or LAN

Latest version:
GNOME 2.x: 0.9.2
GNOME 1.4: 0.8.3

Firestarter in a Nutshell:
Firestarter is a free firewall tool for Linux machines. Whether you simply want to protect your personal workstation or you have a network of computers to secure, Firestarter is here to make your life easier. While a firewall can not guarantee security, it is the first line of defence against network based attacks.

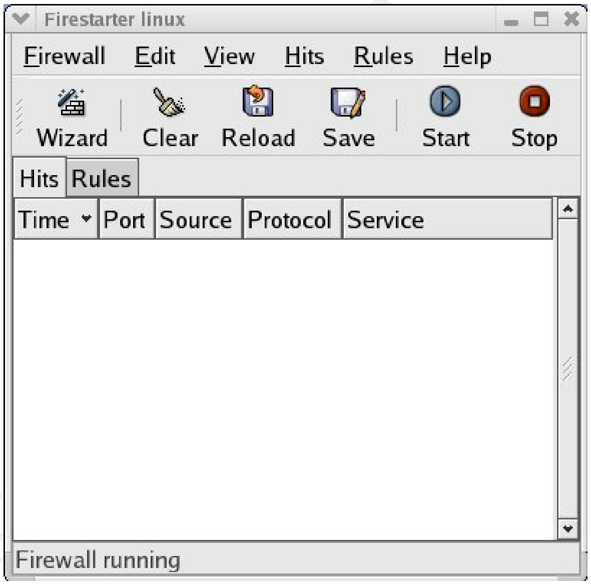
Learn more...

Key Features:

- User friendly graphical interface optimized for GNOME 2.
- Easy to use wizard customizes firewall to your needs.

Time	Port	Source	Protocol	Service
Jan 14 15:55:58	27017	169.254.112.86	udp	unknown
Jan 14 15:55:58	27016	169.254.112.86	udp	unknown
Jan 14 15:55:58	27015	169.254.112.86	udp	unknown

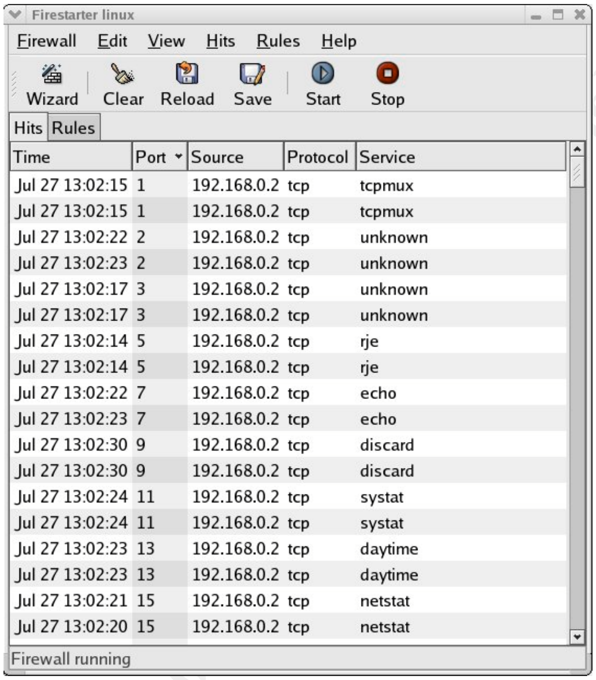
Figure 11: The Firestarter Homepage


Firestarter linux
Firewall Edit View Hits Rules Help
Wizard Clear Reload Save Start Stop
Hits Rules
Time Port Source Protocol Service

Firewall running

Figure 12: The Firestarter Main Window

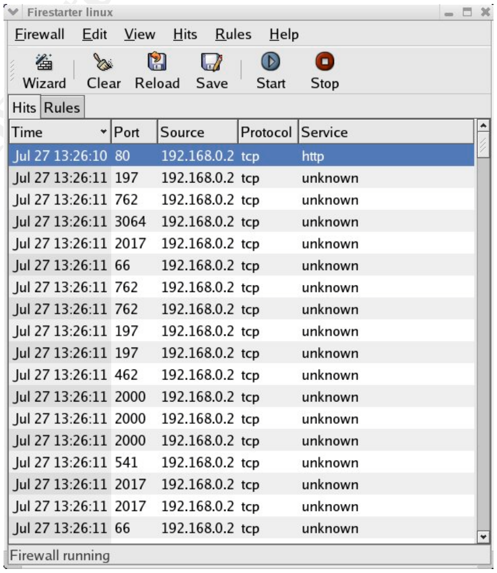
Let us describe the Firestarter main window -- see figure12 -- and how the rules and hits panes within that window function. The hits pane shows closed ports that have been hit -- a connection was attempted -- see figure 13. Next we will construct a rule to allow a service. First we select the port by clicking on that line -- see figure 14. We have selected port 80, http, and by right clicking we specify the manner in which this service is to be configured and/or how connections from this host to this port are to be handled. Figure 15 shows the options for configuration we are offered by right clicking. We can choose to block all connections from this host, trust this host, open the port, open the port to this host only, block future connection to this port by this host and stop logging, or to look up the host name. In our example we choose to open the port. If we then click on the rules tab, in the rules pane of the main window we will see under open ports Port 80 (http). Other rules can be specified in a similar fashion to block connections from particular hosts or to trust them, etc. Rules can also be entered "manually" by choosing new rule under Rules on the task bar at the top of the main window.



Time	Port	Source	Protocol	Service
Jul 27 13:02:15	1	192.168.0.2	tcp	tcpmux
Jul 27 13:02:15	1	192.168.0.2	tcp	tcpmux
Jul 27 13:02:22	2	192.168.0.2	tcp	unknown
Jul 27 13:02:23	2	192.168.0.2	tcp	unknown
Jul 27 13:02:17	3	192.168.0.2	tcp	unknown
Jul 27 13:02:17	3	192.168.0.2	tcp	unknown
Jul 27 13:02:14	5	192.168.0.2	tcp	rje
Jul 27 13:02:14	5	192.168.0.2	tcp	rje
Jul 27 13:02:22	7	192.168.0.2	tcp	echo
Jul 27 13:02:23	7	192.168.0.2	tcp	echo
Jul 27 13:02:30	9	192.168.0.2	tcp	discard
Jul 27 13:02:30	9	192.168.0.2	tcp	discard
Jul 27 13:02:24	11	192.168.0.2	tcp	systat
Jul 27 13:02:24	11	192.168.0.2	tcp	systat
Jul 27 13:02:23	13	192.168.0.2	tcp	daytime
Jul 27 13:02:23	13	192.168.0.2	tcp	daytime
Jul 27 13:02:21	15	192.168.0.2	tcp	netstat
Jul 27 13:02:20	15	192.168.0.2	tcp	netstat

Firewall running

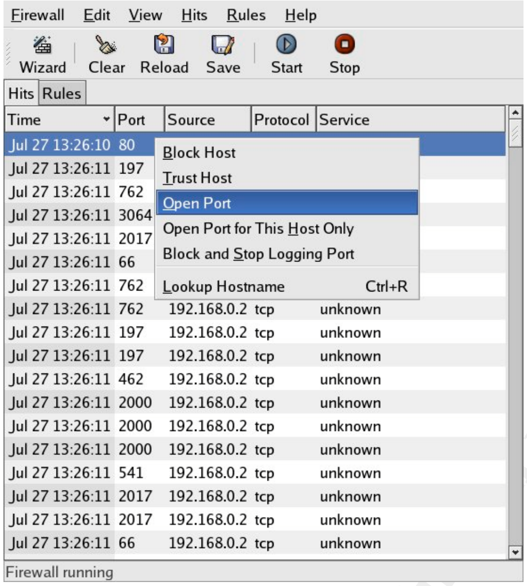
Figure 13: The Hits Pane



Time	Port	Source	Protocol	Service
Jul 27 13:26:10	80	192.168.0.2	tcp	http
Jul 27 13:26:11	197	192.168.0.2	tcp	unknown
Jul 27 13:26:11	762	192.168.0.2	tcp	unknown
Jul 27 13:26:11	3064	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2017	192.168.0.2	tcp	unknown
Jul 27 13:26:11	66	192.168.0.2	tcp	unknown
Jul 27 13:26:11	762	192.168.0.2	tcp	unknown
Jul 27 13:26:11	762	192.168.0.2	tcp	unknown
Jul 27 13:26:11	197	192.168.0.2	tcp	unknown
Jul 27 13:26:11	197	192.168.0.2	tcp	unknown
Jul 27 13:26:11	462	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	541	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2017	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2017	192.168.0.2	tcp	unknown
Jul 27 13:26:11	66	192.168.0.2	tcp	unknown

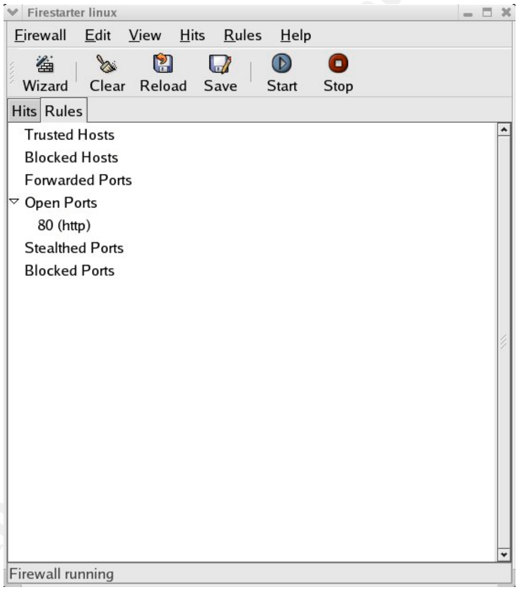
Firewall running

Figure 14: The Hits Pane with Port 80 Selected



Time	Port	Source	Protocol	Service
Jul 27 13:26:10	80			
Jul 27 13:26:11	197			
Jul 27 13:26:11	762			
Jul 27 13:26:11	3064			
Jul 27 13:26:11	2017			
Jul 27 13:26:11	66			
Jul 27 13:26:11	762			
Jul 27 13:26:11	762	192.168.0.2	tcp	unknown
Jul 27 13:26:11	197	192.168.0.2	tcp	unknown
Jul 27 13:26:11	197	192.168.0.2	tcp	unknown
Jul 27 13:26:11	462	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2000	192.168.0.2	tcp	unknown
Jul 27 13:26:11	541	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2017	192.168.0.2	tcp	unknown
Jul 27 13:26:11	2017	192.168.0.2	tcp	unknown
Jul 27 13:26:11	66	192.168.0.2	tcp	unknown

Figure 15: Port Configuration in the Hits Pane



Trusted Hosts
Blocked Hosts
Forwarded Ports
Open Ports
80 (http)
Stealthed Ports
Blocked Ports

Figure 16: The Rules Pane

Conclusions

Firestarter is a powerful and simple to use firewall utility. While it is true that everything Firestarter does can be done by writing a shell script to configure iptables[13], it is still easier, even for the experienced administrator, to allow a utility with a graphical front end to write the script for them. If there is anything that Firestarter is not capable of doing the script firestarter.sh can always be

[13] Benjamin D. Thomas, 10 minutes to an iptables based Linux firewall, *LinuxWorld,* 9/21/2001,http://www.linuxsecurity.com/articles/firewalls_article-3707.html

tweaked. For most users, ease of use would be a great plus over doing it "by hand." I plan on using Firestarter to configure a standalone firewall host as part of my upcoming SANS firewall course.

Additional Reading

Stuart McClure, Joel Scambray and George Kurtz, Hacking Exposed: Network Security Secrets and Solutions, 3rd Edition, Osborne/McGraw Hill, 2001

Joseph D. Sloan, Network Troubleshooting Tools, O'Reilly and Associates, 2001

Red Hat Linux 9: Red Hat Linux x86 Installation Guide, Red Hat Inc., 2003

Evi Nemeth, Garth Snyder and Trent R. Hein, Linux Administration Handbook, Prentice Hall PTR, 2002

Citations

http://firestarter.sourceforge.net/

Michael Russell Grimaila, The Role of Bastille Linux in Information Security, GSEC Practical Assignment, February 18, 2002

Richard Bajusz, Security Applications of Bootable Linux CD-ROMs, GSEC Practical Assignment, November 30, 2001

Fyodor, Nmap Website, http://www.insecure.org/nmap/

Port Scanning, http://weadmin.com/satish/talk/port_scanning.html

Ports, scan responses and ICMP, http://weadmin.com/satish/talk/scan_responses.html

http://www.securityfocus.com/archive/91/210923/2001-08-27/2001-09-02/0

http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/s1-firewallconfig.html

Neil Warner, Scan 23, South Florida Honeynet Project, http://honeynet.hackers.nl/scans/scan23/sol/Neil.html

http://lists.insecure.org/lists/nmap-hackers/2000/Jan-Mar/0095.html

http://security.rbaumann.net/scans.php?sel=1

Neil Warner, Scan 23, South Florida Honeynet Project, http://honeynet.hackers.nl/scans/scan23/sol/Neil.html

Benjamin D. Thomas, 10 minutes to an iptables based Linux firewall, *LinuxWorld,* 9/21/2001,http://www.linuxsecurity.com/articles/firewalls_article-3707.html