Global Information Assurance Certification Paper

SANS GSEC Practical Assignment

Deploying Netfilter Firewalls with Firewall Builder & Devil Linux

Prepared By:	Nicholas Kosovich
Assignment Version:	GSEC Practical Assignment (v.1.3)
Assignment Title	Deploying Netfilter with FWBuilder
Assignment Date:	January 16, 2005

1.0 Abstract

This paper is focused towards providing an understanding of the installation and the deployment of the Devil Linux operating system, running Netfilter / IPTables (a Linux stateful inspection firewall engine) within a distributed small business enterprise. Additionally, we will discuss the the Firewall Builder (FWBuilder) application which is a graphical object-oriented policy management tool for deploying firewall. The release of both the Devil Linux operating system and the FWBuilder application signifies a major development in free open source software in today's security arena. The look and feel of the FWBuilder application rivals many of today's commercial firewall products and the Devil Linux operating is an ideal choice for a firewall platform. Because it is a pre-hardened operating system that natively supports the Netfilter firewall engine. With the acceptance of Linux in the corporate world, the present development of free open source software applications are developing at such a rapid rate that they already have the potential to quickly compete with many commercial firewall vendors, in both the ease of deployment and price.

The detailed description of stateful inspection firewall technology and other firewall technologies is beyond the scope of this document however, more information can be found at the links listed below:

http://www.cyberangels.org/net-ed/classes/firewall.html

http://netfilter.samba.org/

For the sake of brevity: the term Netfilter / IPTables are used interchangeably throughout this document.

2.0 Introduction

2.1 Overview of the Devil Linux Operating System

The Devil Linux operating system offers a security feature-rich environment for the choice of a firewall or other security applications. It is an embedded Linux-based, standard 2.4x kernel, router / firewall distribution that runs directly from a bootable live CD-ROM. The operating system is already hardened and optimized for packet forwarding. Dynamic routing protocols such as RIP v.1 and v.2, OSPF, and BGP v. 4 are supported by the Zebra GNU GPL routing software distribution.

The operating system is loaded into RAM disk at run time. The read only CD-ROM file system offers a crucial security feature: if the firewall is ever compromised, a potential

intruder will not be able to modify any files or upload any Trojan horse binaries. One would simply need to recycle the firewall and reload the operating system. With the absence of a hard drive, hard disk corruption, crashes, and file system integrity checks are eliminated. Configuration data is stored on a write-protected floppy; the mount point on the floppy disk is the only writable partition, thereby eliminating any need for hard drive partitioning. A hard disk for normal operation is not required, since the operating system is run out of RAM disk. If desired, log files can be stored on an optional hard disk or to a remote Syslog server.

The Devil Linux software distribution is Spartan at best; only the binary packages required for the essential operation of the firewall are loaded. Only a minimal set of services is enabled by default. For example, no known insecure utilities such as rsh, rlogin, rexec, finger, who, and talk can be turned on. Moreover, there is no X Window system, NIS, news server, print server, mail and POP/IMAP server, and exportable file systems such as NFS. Devil Linux also lacks a development environment; there is no way for a potential intruder to build binaries on the firewall It should also be noted that the Devil Linux binaries are proprietary to their distribution, the maintainers of the project port every binary to the Devil Linux platform. These factors greatly reduce, if not eliminate, the risk of having potentially vulnerable binary packages on the firewall.

Packages such as Secure Shell (for secure remote management) and the FreeS/WAN IPSec VPN package (supporting both x.509 certificates and pre-shared secret keys) are part of the core distribution. With only a minimal set of binary packages, the distribution also allows a much quicker installation: about fifteen to thirty minutes for the entire distribution. Updating the binary distribution is seamless: boot with the newer CD and update the settings. If more functionality is needed on the firewall, additional packages may be selected from the configuration file on the floppy disk. For example, if PPOE authentication is needed, then simply edit the /etc/sysconfig/software file and turn on the service. The official ISO images of Devil Linux can be used freely for commercial distribution and are directly supported by FWBuilder.

For a list of additional packages that have been certified to run on Devil Linux please visit the following link:

http://www.Devil Linux.org/introduction.htm

Below are the hardware requirements for Devil Linux:

- A Bootable CDROM
- A Floppy Disk for holding the configuration files
- Intel 486 or higher CPU
- 64MB RAM Minimum (Please Note: That applications such as the FreeS/WAN IPSec package are processor and RAM intensive and will therefore require more memory and CPU requirements.

2.2 Overview of Firewall Builder

FWBuilder is an X Window GTK+ GUI-based front end for managing various firewall platforms. The Linux Netfilter, FreeBSD IPfilter and OpenBSD's PF are currently supported. The prevalent feature of the FWBuilder utility is its intuitive graphical interface, which contains a high semblance to the Checkpoint firewall policy editor GUI. The FWBuilder application supports drag-and-drop operations of network objects and has a firewall "Druid" to assist in quickly setting up a firewall rule base. The firewall policy and objects are stored in an XML database format. Policy compilers and the firewall policy delivery mechanism are run over encrypted Secure Shell tunnels. They convert the XML-based policy into a text-file, and upload it to the firewall modules.

Currently, FWBuilder's supported platforms are Redhat 7.x, Mandrake & Debian (part of the regular distribution set), SuSE, Caldera OpenLinux, floppyfw, and Solaris 8. However, Redhat 7.2 is the primary development platform. The GNOME and KDE window managers are fully compliant with FWBuilder. Below are the software requirements for installing FWBuilder on Redhat 7.2:

- gtkmm-1.2.5-1
- gdk-pixbuf-0.16.0
- libsigc++-1.0.1-1
- libxml2-2.4.10
- libxslt-1.0.7-1
- ucd-snmp-4.2.3
- ucd-snmp-utils-4.2.3
- openssl-0.9.6

If installing on platforms other than Redhat Linux 7.2, please refer to the Documents References section of this document for the necessary packages and their library dependencies required to run FWBuilder.

3.0 Implementation

3.1 Scenario

In the Figure 1 below, a management workstation is located at a Network Operations Center (NOC). The management station, running FWBuilder, creates all the firewall configurations for the branch offices, and writes them to their configuration floppies. Please note the Devil Linux distribution does not presently support remote policy installation through SSH

All of the firewalls can be pre-configured at the NOC, and shipped to their respective locations. The management station will be the only source address that will be enabled to establish SSH connections to the firewall devices for maintenance purposes.
The Syslog-ng and Stunnel packages, which are part of the Devil Linux distribution, will be used to redirect the firewall's Syslog traffic over an encrypted Secure Socket Layer tunnel to the Management station.

The firewall hardware configurations consist of:

- Pentium III 300 MHZ processor
- 128 MB of RAM
- 2 Intel e-100B fast Ethernet adapters (an external and internal interface)
- No hard drives.

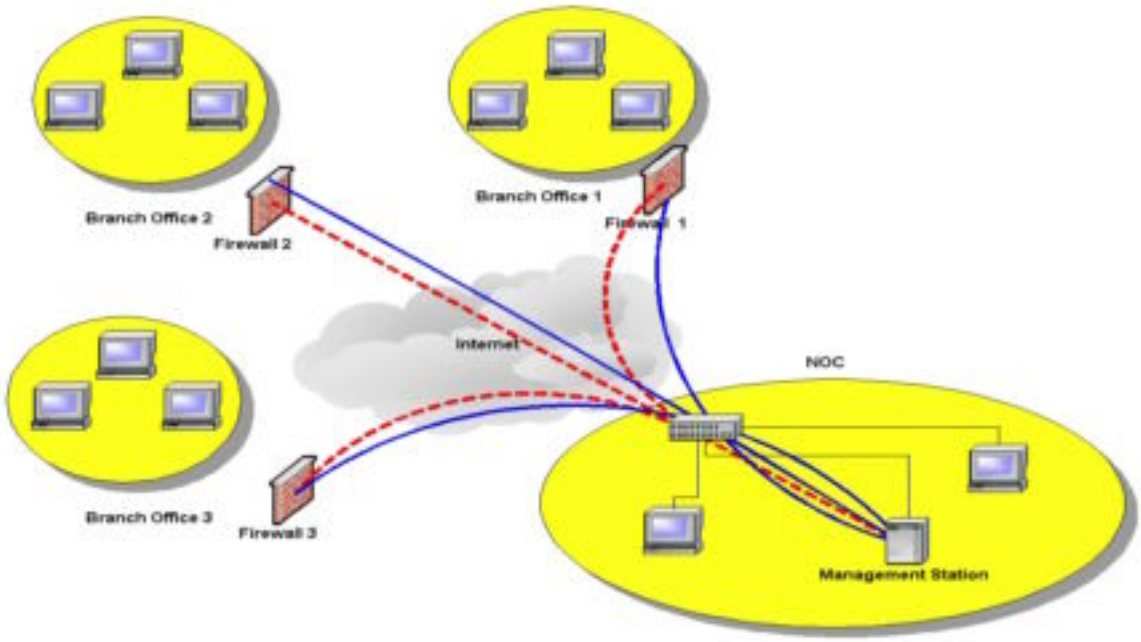


Figure 1 Implementation Scenario

The branch office firewall policy will incorporate the following features:

- Provide Network Address Translation (NAT) for the branch office LANs.
- Deny and log all IP traffic coming inbound to the external interface from the Internet. With the exception of the management station initiating SSH connections and receiving SSL traffic from the firewalls.
- Provide anti-spoofing, packet fragmentation, and source routed packet protection.
- Provide a specific set of Internet services for branch office clients initiating outbound connections to the Internet.

4.0 Devil Linux Installation

The initial stage of the installation is downloading the Devil Linux distribution from the following link:

http://www.Devil Linux.org/download.htm

In order to burn the ISO image to a CD-ROM, one must decompress the file with the bzip utility. If your CD burner is on a Windows workstation, and copying the Devil Linux ISO image from a UNIX workstation is too cumbersome, a win32 version of bzip is available at the following link:

ftp://sources.redhat.com/pub/bzip2/v102/bzip2-102-x86-win32.exe

- Decompress the image and untar the image file:

```
D:\ISO>bzip2-102-x86-win32.exe -d Devil Linux-0.44.tar.bz2
D:\ISO>pkzip Devil Linux-0.44.tar
```

- Burn the CD-ROM
- Copy the etc.tar.gz to the configuration floppy disk (FAT formatted)
- Boot the CD-ROM with the Configuration floppy without write protection
- Login as root, the default password is blank Change the password immediately with the "passwd" utility.
- In order to avoid a runtime error resulting in the firewall not being able to boot and resulting in an **rcS script "Permission denied"** error. Use the vi editor, included in the Devil Linux distribution, to modify the following configuration files (Remember to remove write protection from the configuration floppy):
    - **/etc/sysconfig/network-scripts/ifcfg-eth0**

      The external interface configuration file: Please note that you must provide the module to load the driver for the Ethernet controller. Options for the interface's IP address (dynamic or static), network mask, broadcast address, etc is provided. In our example scenario, we will be using the eepro100 module to load the Ethernet drivers.

- **/etc/sysconfig/network-scripts/ifcfg-eth1**

  The internal interface configuration file: Please note that you must also provide the module to load the driver for the Ethernet controller. Options for the interface's IP address (dynamic or static), network mask, broadcast address, etc is provided. The firewall also has capabilities such as running a DHCP server for internal clients however, for the sake of brevity the configuration of a DHCP server will not be discussed in this document.

- **/etc/sysconfig/config**

  Keyboard locale configuration file: Please note that by default the Latin keyboard layout is chosen. Remove the default Latin keyboard option and replace it with the US locale.

- **/etc/sysconfig/software**

  This file specifies what system services should be run for example SSH, IPSec, etc. By default SSH and Stunnel support are disabled and IPSec support is enabled. Since this scenario will not be requiring IPSec, turn off the option by answering "no" in the appropriate field in the file. Accordingly, enable SSH and Stunnel support by answering, "yes" in their relevant fields.

- **/etc/sysconfig/network**

  Edit the firewall's hostname, domain name, default gateway, default gateway interface, routing, firewalling, IPSec, and SSH support, and extra SSH options. Do not answer "No" the routing and firewall options, these must be enabled at a bare minimum for the firewall process to occur.

- **/etc/sysconfig/clock**

  By default the system clock is set to UTC Coordinated Universal Time. If GMT is necessary, one only needs to remove the "UTC=1" statement and replace it with "GMT=1."

- **/etc/syslog-ng/syslog.conf**

  Edit Syslog Next Generation Configuration file. Set the daemon's initialization parameter to log to the local host's loopback address by adding the appropriate entries under the following fields in the syslog.conf:

```
# set up logging to a loghost forwarded from localhost via stunnel
destination loghost {tcp("127.0.0.1" port(514));};

# send everything to loghost, too
log { source(src); destination(loghost); };
```

Place this command at the end **/etc/init.d/rcS** file to redirect Syslog traffic over SSL (where xxx.xxx.xxx.xxx is the IP Address of the Management Station):

```
/usr/sbin/stunnel -c -d 127.0.0.1:syslog-ng -r xxx.xxx.xxx.xxx:syslog-ngs -
N syslogng
```

Configure the Firewall Builder workstation's syslog.conf file to have the following parameters:

```
# network logs come from the local network and from stunnel on127.0.0.1
source src { unix-stream("/dev/log"); internal(); };
source remote { tcp(ip("127.0.0.1") port(514) keep-alive(yes)); };
```

Add entries into **/etc/services** files on both machines, both port numbers are arbitrary, just make sure they match between the syslog-ng.conf file and the stunnel command. syslog-ng 515/tcp syslog-ngs 1724/tcp. Please note the management station must have a self-signed x.509 certificate to be issued previously to work in congruence with Stunnel, this link is provided below:

http://venus.ece.ndsu.nodak.edu/~jezerr/linux/secure-remote-logging.html

- Type "save-config" to initialize configuration changes, and write protect the floppy.

5.0 Firewall Builder Installation

In this scenario, the firewall management workstation will be installed with Redhat version 7.2 running on an i386 platform and configured with the GNOME Window manager. Please note all the previously mentioned Firewall Builder dependencies in Section 2.2 have been installed.

Only three rpm packages need to be installed (in descending order):

- libfwbuilder-0.10.5-1rh72.i386.rpm (Shared Application Libraries)

- fwbuilder-1.0.1-1rh72.i386.rpm (Application Binary)
- fwbuilder-dl-0.1-1.noarch.rpm (Devil Linux Firewall Installer Application)

The FWBuilder Installation is straightforward:

```
[root@fwbuilder]# rpm –ivh libfwbuilder-0.10.5-1rh72.i386.rpm
[root@fwbuilder]# rpm –ivh fwbuilder-1.0.1-1rh72.i386.rpm
[root@fwbuilder]# rpm -i --nodeps fwbuilder-dl-0.1-1.noarch.rpm
```

Please note that the no dependencies option (--nodeps) must be used to install the FWBuilder Devil Linux installer. If it is not used the following error will be received:

```
[root@fwbuilder]# rpm -i --nodeps fwbuilder-dl-0.1-1.noarch.rpm
error: failed dependencies:
        fwbuilder-iptables >=0.99 is needed by fwbuilder-dl-0.1-1
```

There is a bug in the spec file of the package in which a failed dependency on fwb_iptables v0.9.9 points to a nonexistent package, "fwbuilder-iptables." As of this writing, a bug report has been submitted to the Firewall Builder project page at SourceForge.

The dl_install.sh shell script will be installed into the /usr/bin directory, this script is used for transferring the firewall policy to the Devil Linux configuration floppy.

Be sure to add the firewall's name and IP address to the management station's host file. This will be necessary for the next step.

5.1 Firewall Objects Creation

This process will entail creating a firewall object, brnchoff01, and a management station, fwbuilder, with the "Druid" quick start firewall wizard.

- FWBuilder is invoked from a terminal prompt:

```
[root@fwbuilder]# fwbuilder
```

- Figure 2 below shows the initial startup screen of Firewall Builder

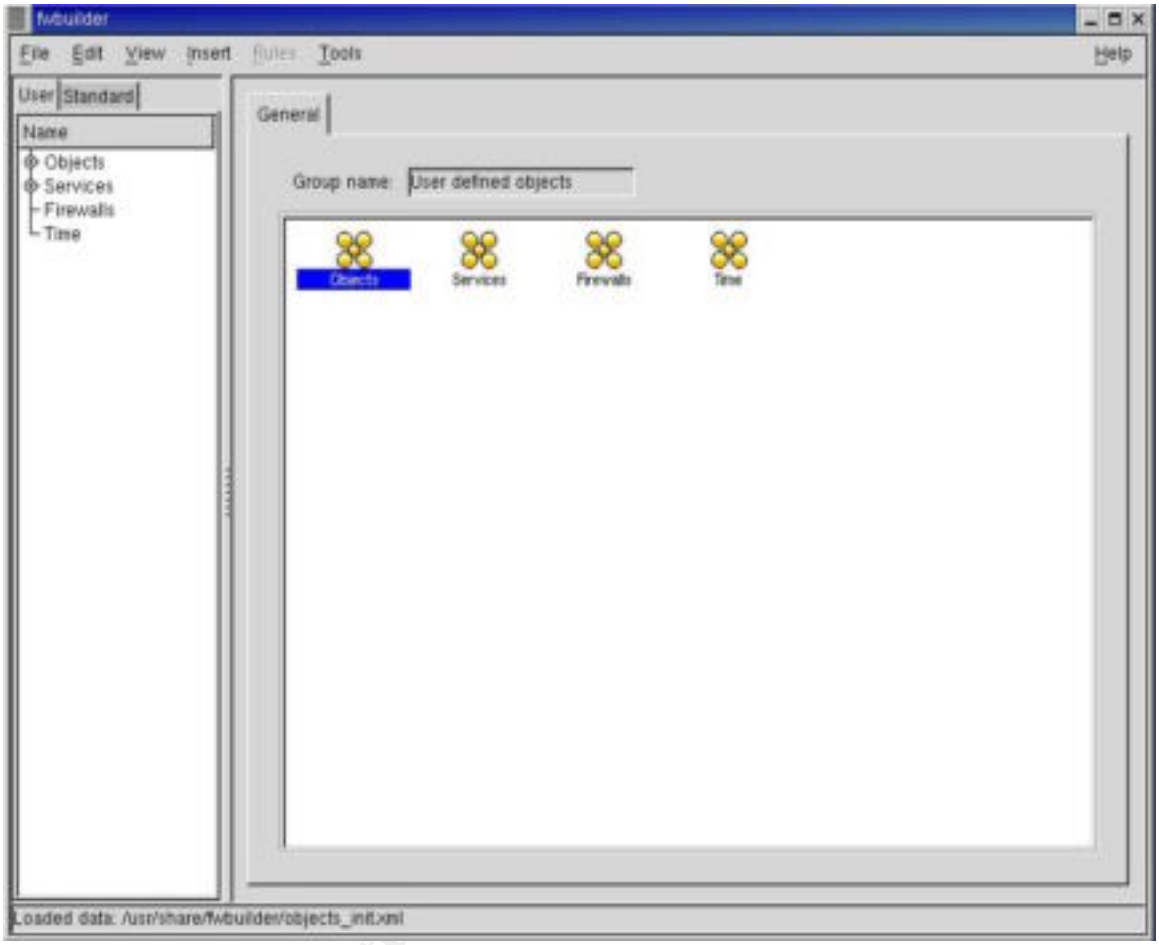


Figure 2 Initialization Screen

- From the **Tools** menu select **Discover Objects**. Then choose the **Read File from Hosts(5)** option for the **Discovery Method** and click the **NEXT** radio button.

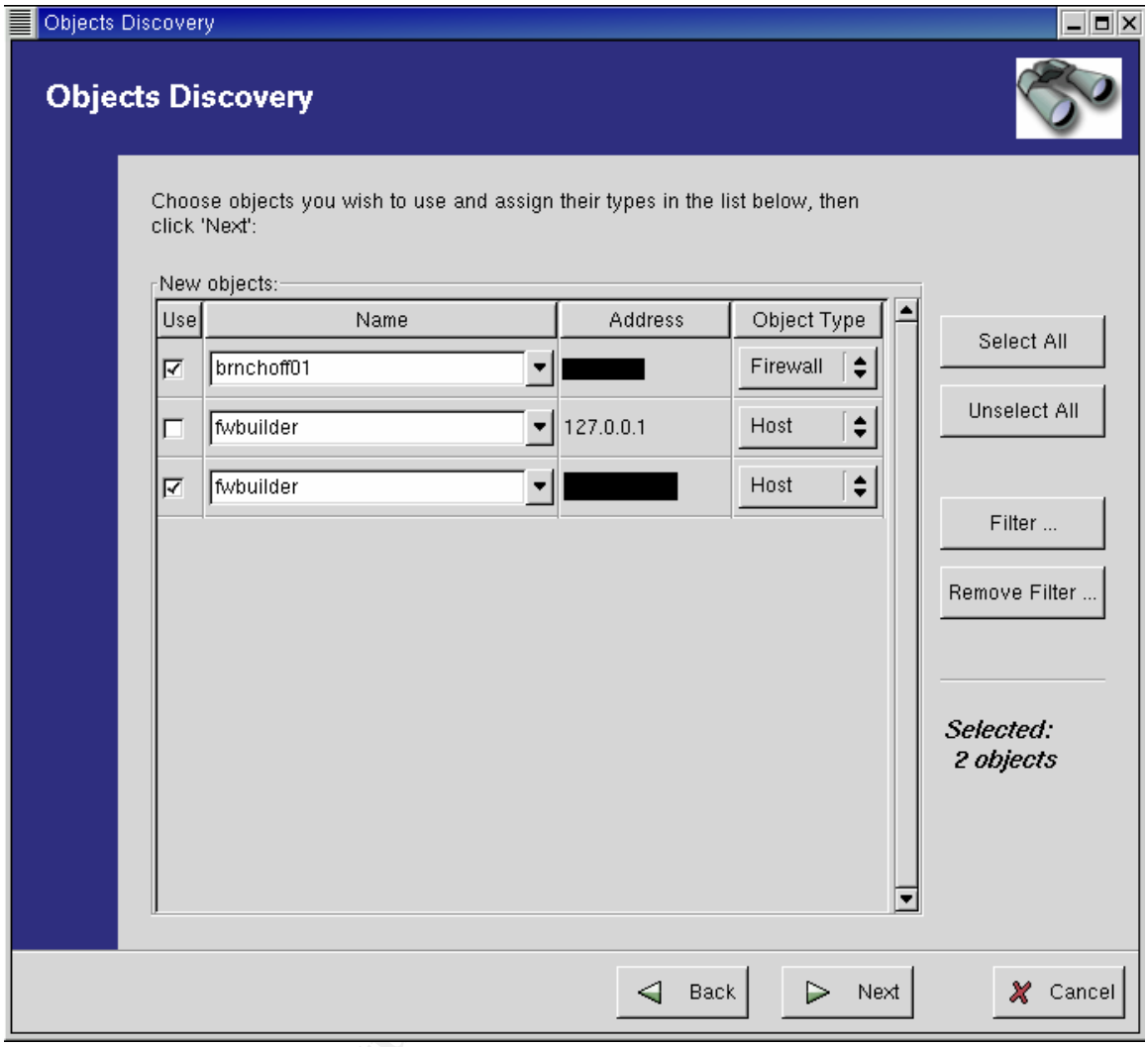


Figure 3 Network Discovery Screen

- You should now see your firewall and management station objects. Deselect the **loopback** selection for the management station; this definition is not needed for the firewall policy.

- Click the **Next** and **Finish** radio buttons to write your objects to the FWBuilder database.

- The firewall interfaces need to defined. Select the firewall object from the **user** tab and click the **Interfaces** tab.

- Enter the IP address and network mask, and a label for the interface. Press the **OK** radio button.

Figure 4 Interface Screen

- Select the **Security Level** radio button and check the **This interface is External (insecure)** box and the **OK** radio button. If you select this radio button for both the internal and external interfaces, you will apply the firewall rules on all interfaces, and block all permitted outbound internal traffic.

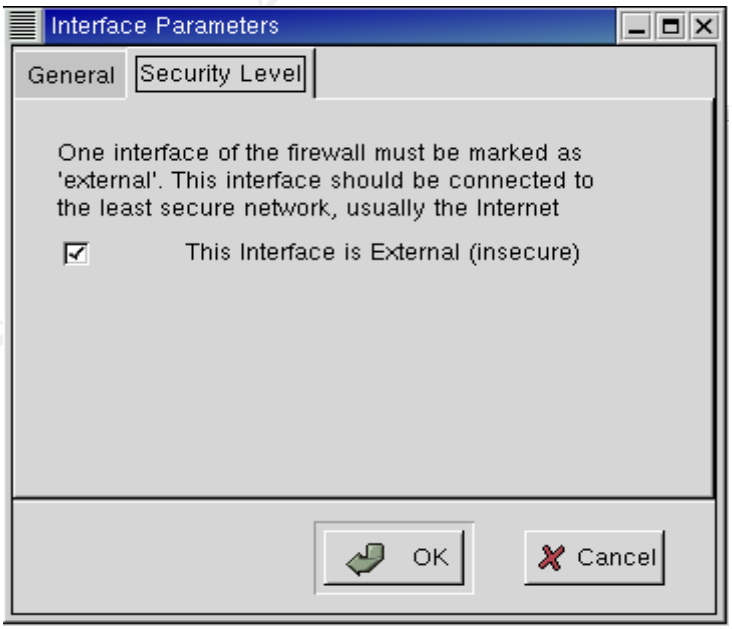


Figure 5 Security Level Screen

- Click the **File** menu and save your objects to the default file location: **/usr/share/fwbuilder.objects_init.xml.**

- Create an internal network object, int_net, to represent the branch office LAN. Click the **insert** pull down menu and select **Network.** For this scenario a private IANA network of 192.168.0.0 and a 24-bit subnet mask will be chosen.

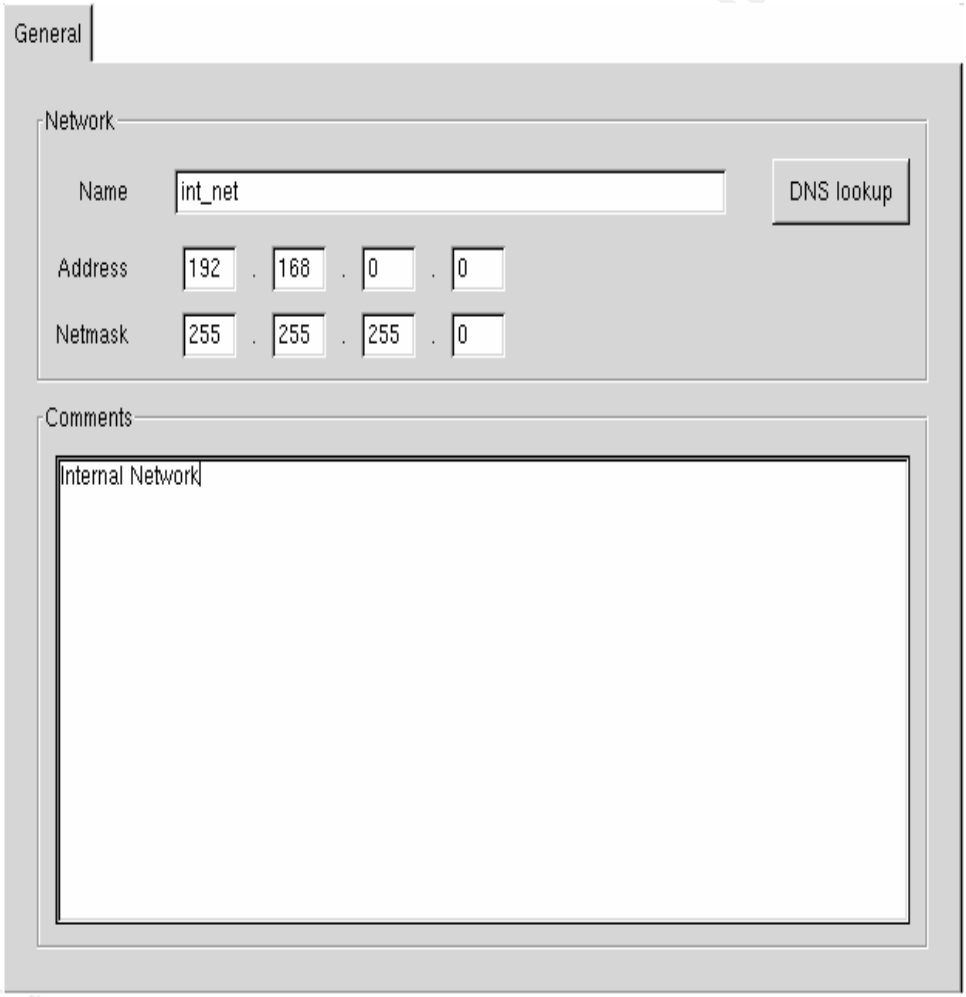


Figure 6 Network Object

5.2 Rule Base Creation

The firewall "Druid" will also be used in the creation of firewall's rule base. However, modifications will be made to the rule base to reflect the parameters of the branch office firewall policy in Section 3.1

- From the **Rules** drop down menu select the **Help Me Build Firewall Policy** option and select the middle option and press the **Next** radio button.

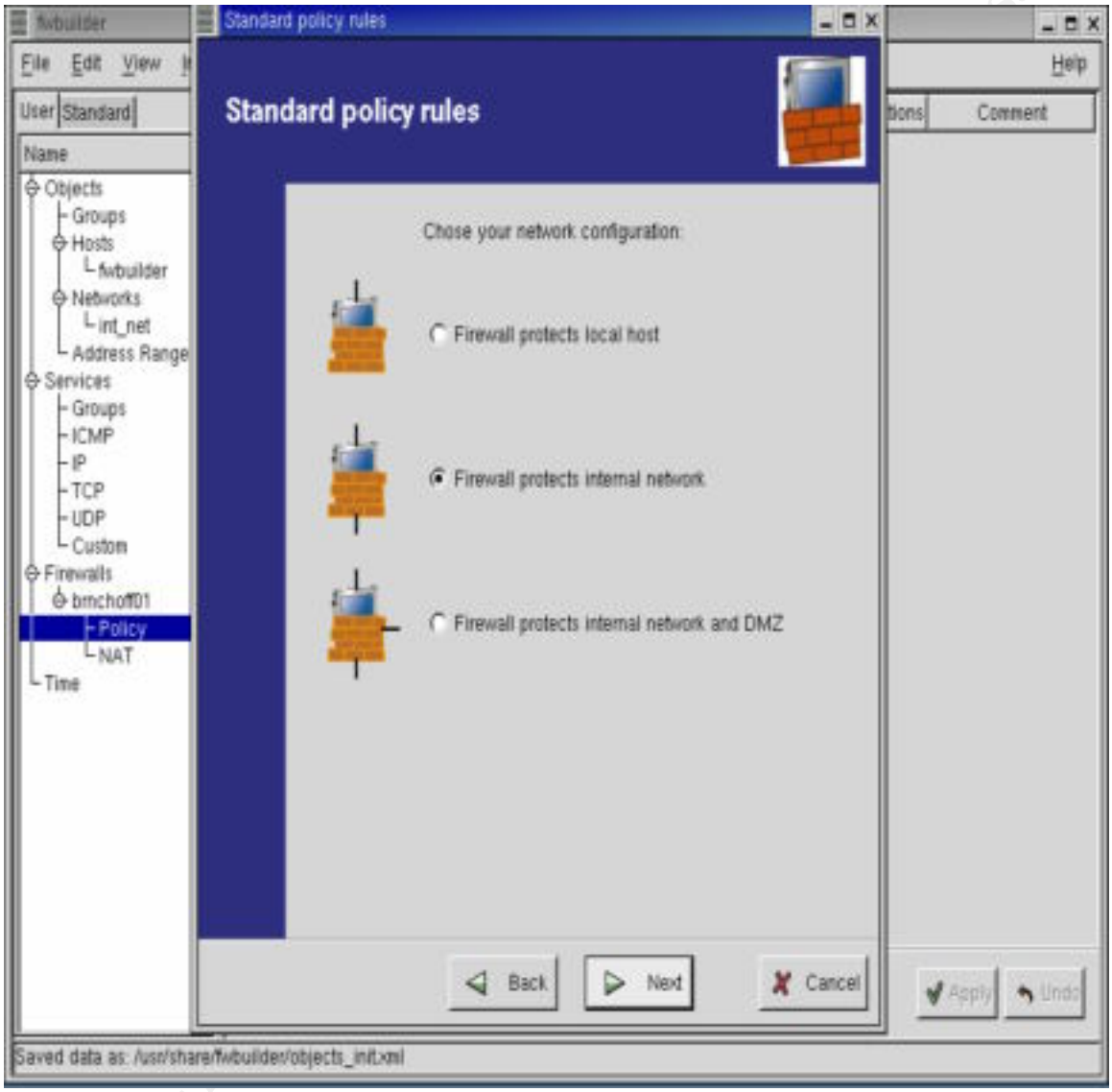


Figure 7 Policy Screen

- Next choose the firewall's external network interface, and press the **Next** radio button.

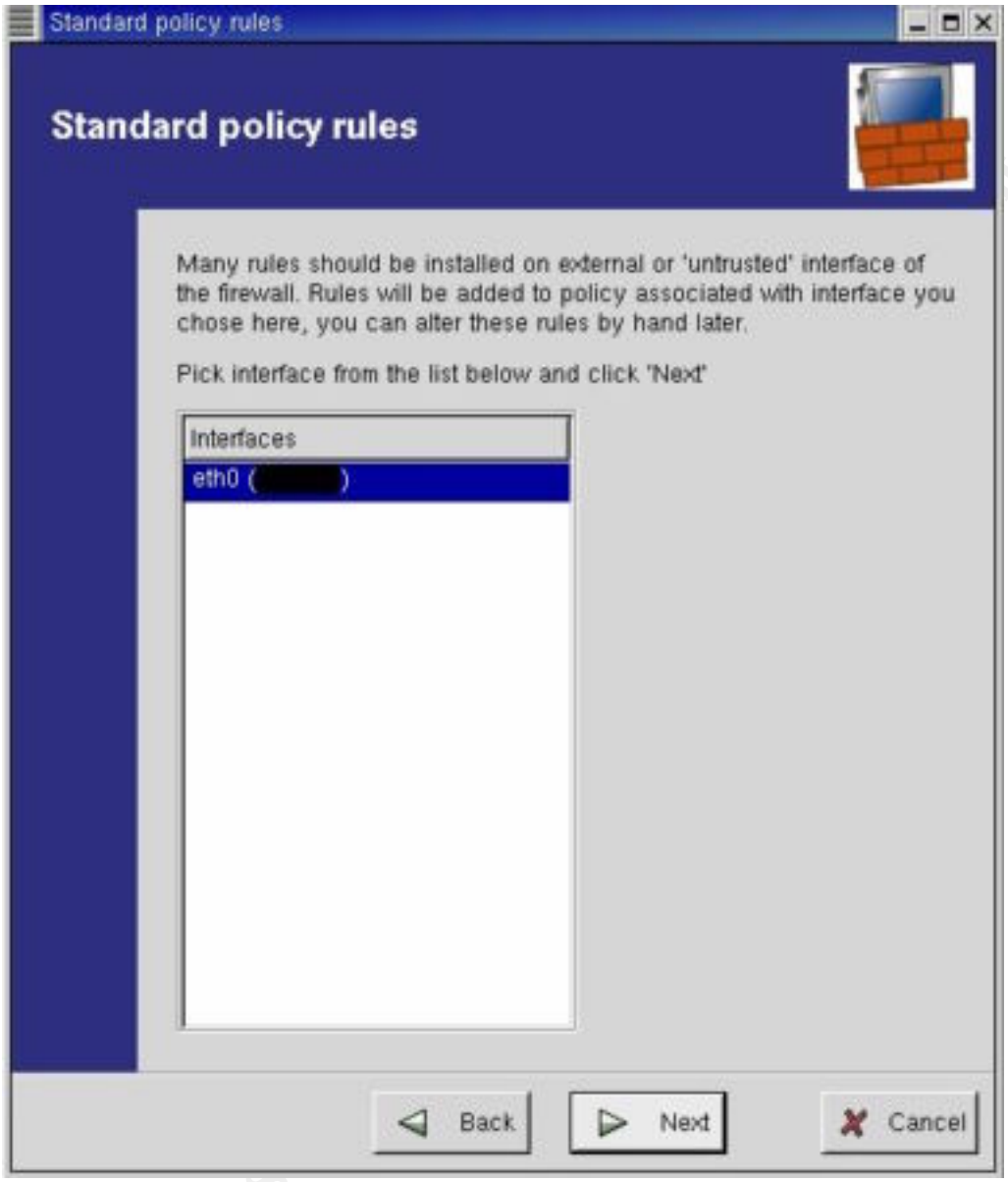


Figure 8 Policy Rule Screen

- Select all of the options for firewall policy property dialog boxes and press the **Next** radio button.

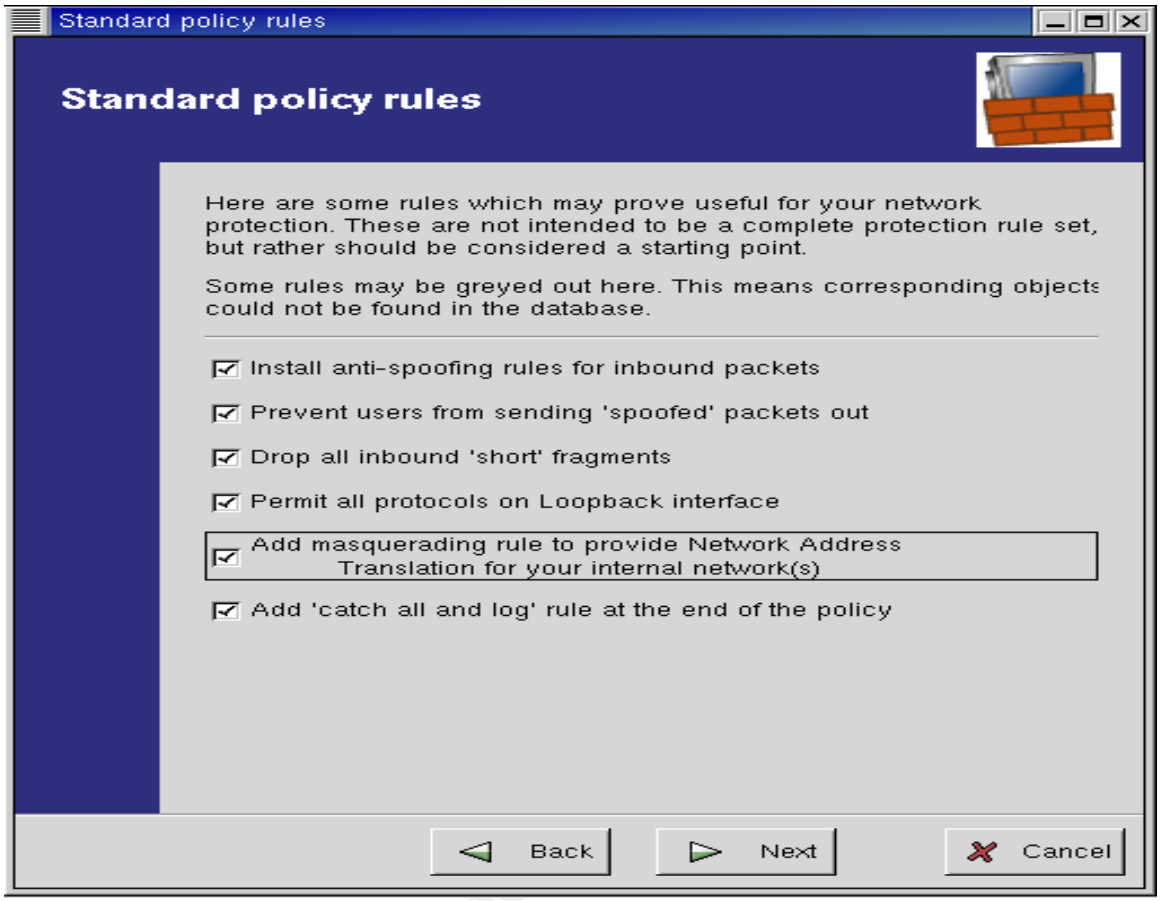


Figure 9 Standard Policy Rules

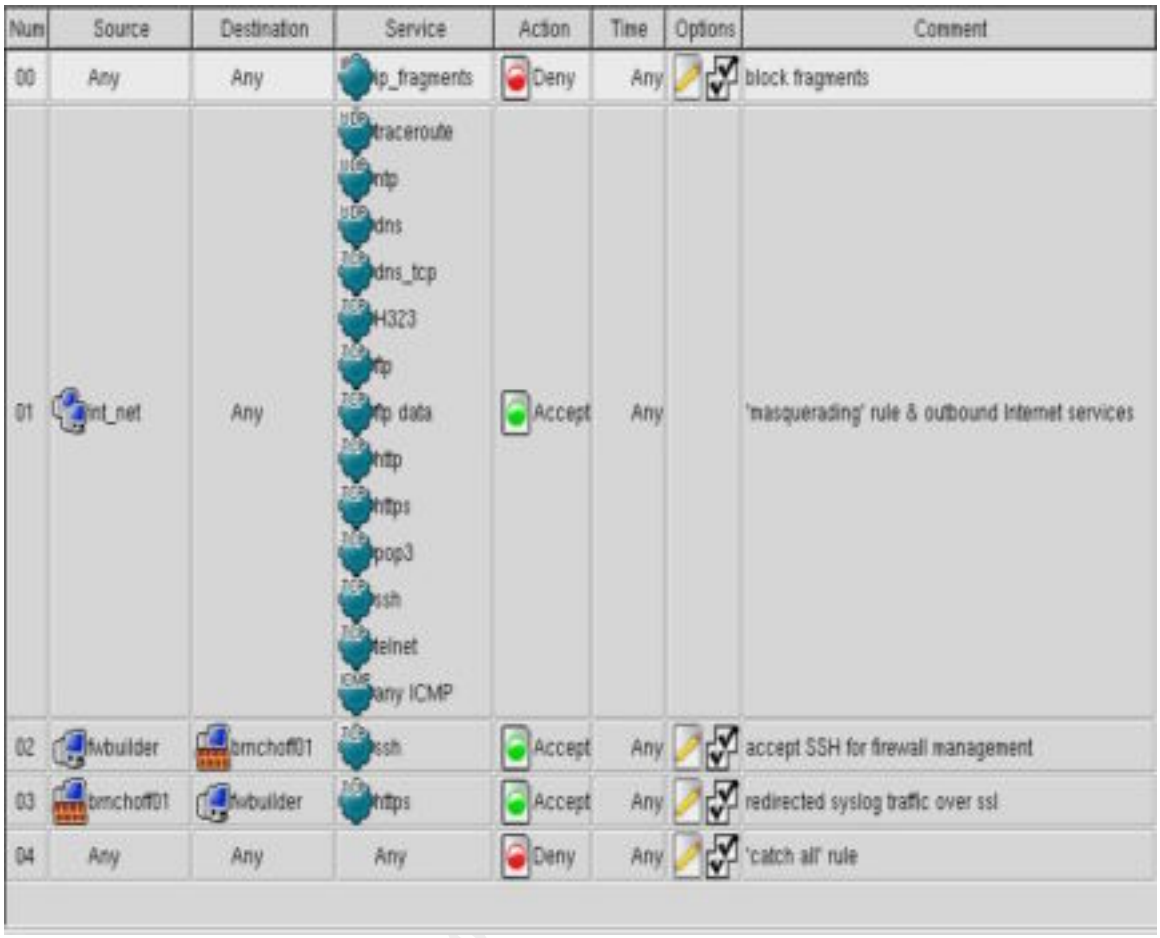
- Select the internal network object, int_net, as the firewall's protected object and select the **Next** and **Finish** Radio buttons to generate the policy.

- The figure below shows the firewall configuration generated by the "Druid" wizard.

Num	Source	Destination	Service	Action	Time	Options	Comment
00	Any	Any	ip_fragments	Deny	Any		block fragments
01	int_net	Any	Any	Accept	Any		'masquerading' rule
02	Any	Any	Any	Deny	Any		'catch all' rule

Figure 10 Automatic Rule Base

- Customize the firewall policy to limit the internal LANs outbound access to a selected number of services and apply the modified policy and add SSH and SSL access for the management station.



Num	Source	Destination	Service	Action	Time	Options	Comment
00	Any	Any	ip_fragments	Deny	Any		block fragments
01	int_net	Any	traceroute ntp dns dns_tcp H323 ftp ftp data http https pop3 ssh telnet any ICMP	Accept	Any		'masquerading' rule & outbound Internet services
02	fwbuilder	bmchoff01	ssh	Accept	Any		accept SSH for firewall management
03	bmchoff01	fwbuilder	https	Accept	Any		redirected syslog traffic over ssl
04	Any	Any	Any	Deny	Any		'catch all' rule

Figure 11 Custom Rule Base

5.3 Compiling and Installing the Policy

Before the firewall policy can be compiled and installed to the configuration floppy, the FWBuilder application needs to know the path to the Devil Linux installer script, dl_install.sh.

- In the **Compile/Install** tab of the firewall objects properties pick the compiler that you would like to use creating your firewall rule sets and its parameters. Additionally, pick the path and command line parameters for the dl_install.sh script in the **Installer** section of the page.

- From **Tools** menu choose the **compile** and than **install** options.

- Insert your configuration floppy and remove the write protection, and copy over the policy file, to the /etc/init.d/ directory. Write protect the disk

6.0 Conclusion

Firewall Builder is a prevailing application in the configuration and deployment of Linux/BSD based firewalls. With rumors of a Win32 binary being developed and support for SSL/TLS tunnels for firewall management, one is only left to wonder how long this product will remain under the GNU GPL license? The proof that the product is evolving at such a fervent pace is seen in the frequency of new version releases, and the postings for bug resolutions on the Sourceforge repository.

Linux and BSD based firewalls have had a long running presence in the corporate realm. However, with the combination of stateful inspection firewalls, embedded operating systems, and a highly configurable firewall management interface: A powerful trend is emerging in the security market; one of stability, security, and cost effectiveness.

7.0 References

Netfilter project Homepage
http://netfilter.samba.org

Firewall Builder Homepage
Vadim Kurland- vadim@vk.crocodile.org
http://www.fwbuilder.org

Firewall Builder Software Dependencies for other Operating Systems
http://www.fwbuilder.org/pages/Documents/Requirements.html

Devil Linux Installation page
Heiko- heiko@devil-linux.org
http://www.devil-linux.org/ADMIN.html

Devil Linux Troubleshooting
rcS script "Permission denied" error
http://ww.devil-linux.org/cgi-bin/ikonboard/topic.cgi?forum=1&topic=5

Secure Remote Logging with syslog-ng and stunnel HOWTO
Jeremy Zerr - jezerr@venus.ece.ndsu.nodak.edu
http://venus.ece.ndsu.nodak.edu/~jezerr/linux/secure-remote-logging.html

Stunnel Homepage
http://www.stunnel.org

Openssl Homepage

http://www.openssl.org

Man pages for OpenBSD's port of Secure Shell
http://www.openssh.org/manual.html

General Firewall Information
http://www.cyberangels.org/net-ed/classes/firewall.html

UTC Time coordinates
http://www.timeanddate.com/time/abbreviations.html

Linux Firewalls
http://www.linux-firewall-tools.com/linux/

FreeS/WAN Linux IPSec Package
http://www.freeswan.org/

Syslog Next Generation
http://www.balabit.hu/en/downloads/syslog-ng/

Zebra Free Dynamic Routing Protocol Software distributed by *GNU Public License*
http://www.zebra.org/

GNU GPL License Policy
http://www.gnu.org/copyleft/gpl.html - SEC1