

Ubuntu Tutorials

Enhancing your Ubuntu experience!

HOME

CHAT

CONTACT ME

INSTANT UBUNTU

LICENSE

Getting Started with Firewall Builder

By Christer Edwards | 2009/06/03

7 Comments

If you've enjoyed this blog, please consider picking up a copy of my Ubuntu book, [Instant Ubuntu](#). Thanks for visiting!

*This article is part of a series regarding firewalling and network security using the **Firewall Builder** tool on Ubuntu. This is user-contributed content. If you would like to contribute an article, please see the About page for contact information.*

Getting Started with Firewall Builder



Author: vadim@fwbuilder.org

<http://www.fwbuilder.org>

This guide starts a series of articles about Firewall Builder. **Firewall Builder** (also known as **fwbuilder**) is a GUI firewall configuration and management tool that supports **iptables (netfilter)**, **ipfilter**, **pf**, **ipfw**, **Cisco PIX (FWSM, ASA)** and **Cisco routers extended access lists**. Both

Recommended Hosting

Professional Web Hosting
..... for \$6.95/month

- On 1 Account you get:
- » UNLIMITED Domains
- » UNLIMITED Transfer
- » UNLIMITED Space
- » FTP, SSL, PHP
- » CGI, MySQL, Ruby
- » Stats, Free Sitebuilder

 bluehost



Recent Posts

DenyHosts Synchronization Mode

Configure Automatic Updates : Ubuntu Server

Tunnel SSH over SSL

Win a Free Copy of Instant Ubuntu

Find your Ubuntu kernel version

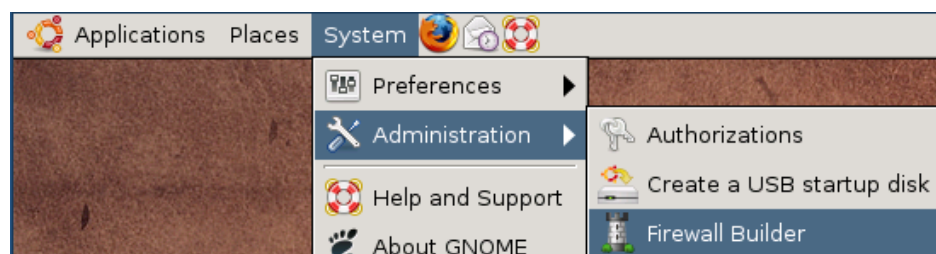
Blogroll

HubbleStack

Instant Ubuntu

professional network administrators and hobbyists managing firewalls with policies more complex than is allowed by simple web based UI can simplify management tasks with the application. The program runs on Linux, FreeBSD, OpenBSD, Windows and Mac OS X and can manage both local and remote firewalls. The first article is an introduction to the program. We will follow up with series of articles focusing on more advanced aspects of it in the coming weeks.

Firewall Builder is packaged with most Linux distributions and is available under “System/Administration” menu.



If it is not there, then it probably needs to be installed on your system. You need to install package that has supporting API library **libfwbuilder** and package **fwbuilder** that contains Firewall Builder GUI and policy compilers. Use *apt-get* or *aptitude* to find and install them:

```
# aptitude install libfwbuilder fwbuilder
```

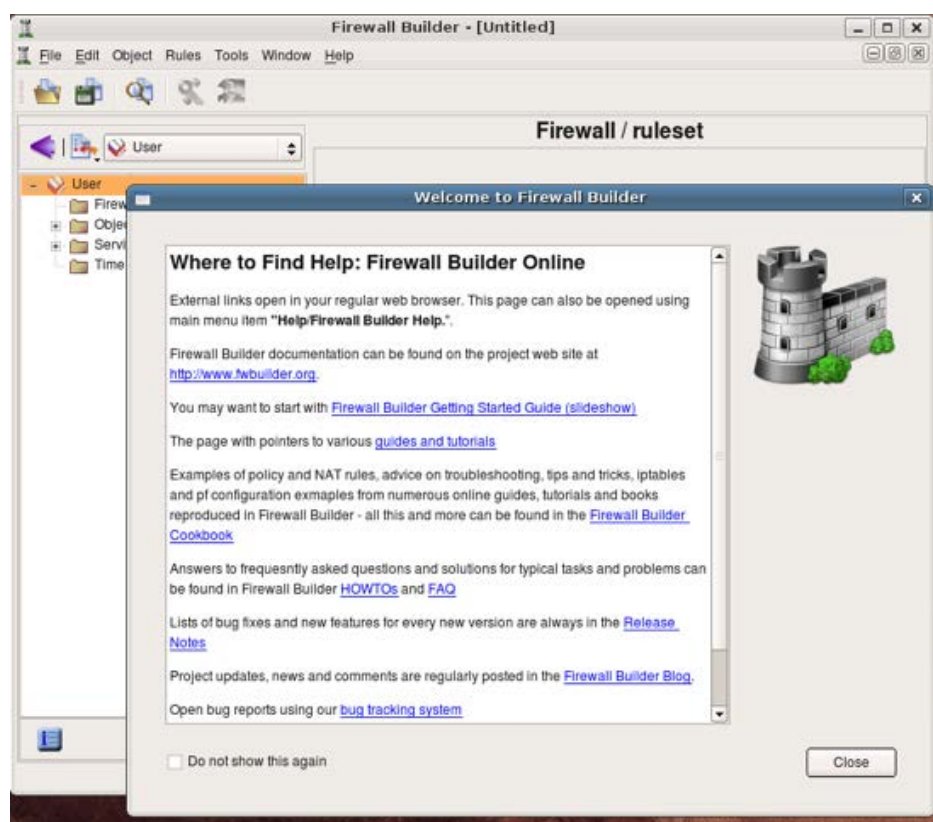
On FreeBSD and OpenBSD Firewall Builder is part of ports, you can find it in **/usr/ports/security/fwbuilder**.

Packages shipping with Ubuntu are always one or two minor revisions behind. If you want to try the latest version, you can use pre-built binary *.deb* packages offered on the project's web site or build from source using our online [installation instructions](#). Pre-built binary packages can be installed using our repositories of rpm and deb packages, see instructions on [this page](#).

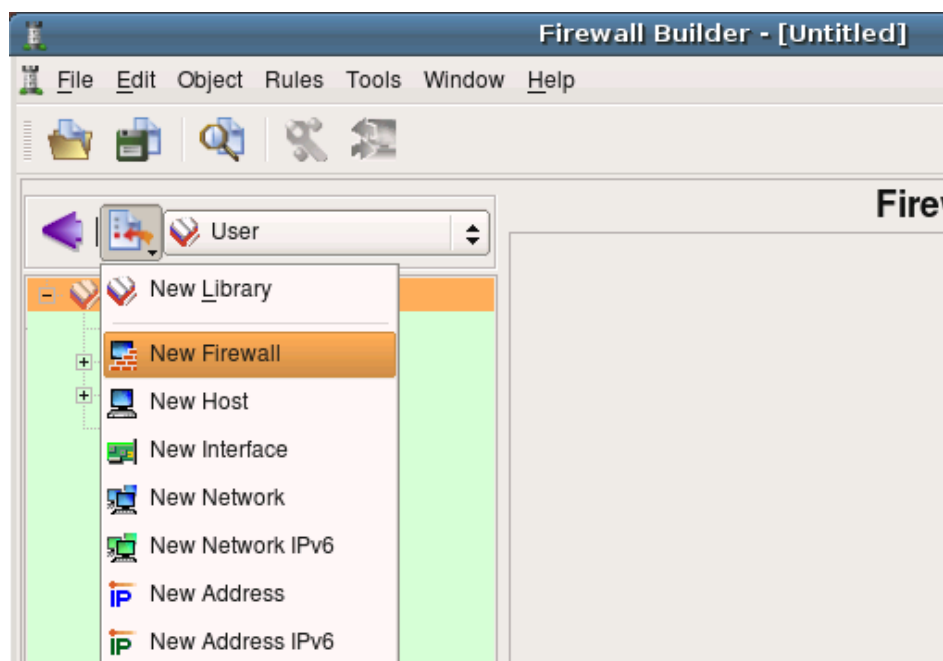
If the system menu item is not there or you have built the program from source, you can always launch it from the command line by just typing “fwbuilder” on the shell prompt:

```
$ fwbuilder
```

The program starts and opens main window and greeting dialog. The dialog provides links to the project web site where you can find more tutorials, FAQ, Firewall Builder Cookbook and other documentation, as well as bug tracking system and links to user forums and mailing list. Clicking on the link in the dialog opens corresponding web page in your web browser. This works the same on all supported OS: Linux, Windows and Mac OS X. You can always open this dialog later using an item in the main menu "Help".



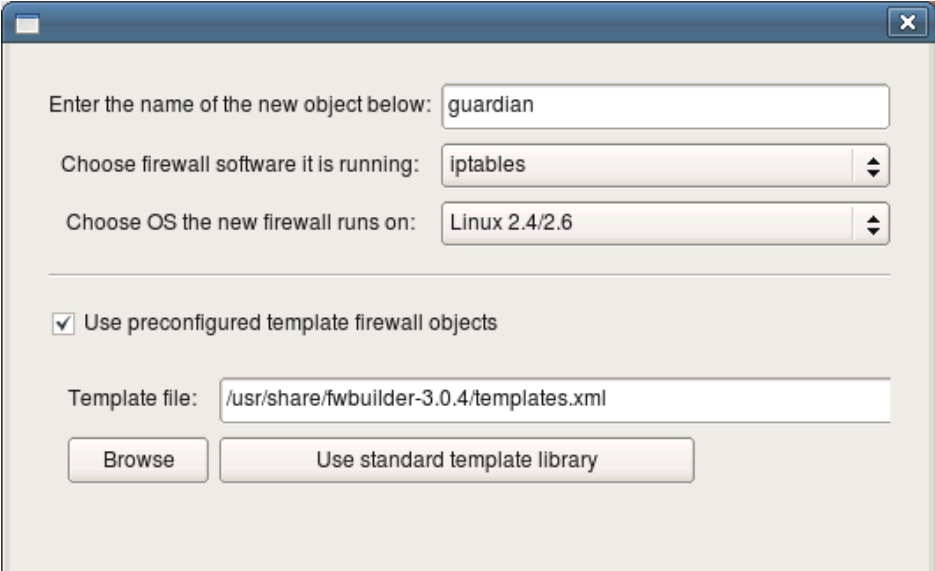
Lets create our first firewall object. To do this, we'll use object creation menu that appears when you click on the icon in the small toolbar right above the object tree. Choose menu item "New Firewall" from the menu that appears.



The program presents wizard-like dialog that will guide you through the process of creation of the new firewall object. In the first page of the wizard you can enter the name for the new firewall object (here it is “guardian”), its platform (“iptables”) and host OS (“Linux”).

There are two ways new firewall can be created: you can use one of the preconfigured template firewall objects or create it from scratch. This tutorial demonstrates the first method (using template object). To do this, check checkbox “Use pre configured template firewall objects”. Template can be taken from the library of template objects that comes with Firewall Builder package or from a file provided by the user. The latter is useful when administrator wants to distribute a library of predefined templates to other users in the enterprise. We are using one of the standard templates in this guide and therefore leave standard template library path and name in the “Template file:” input field. Click “Next” to move on to the next page of the wizard.

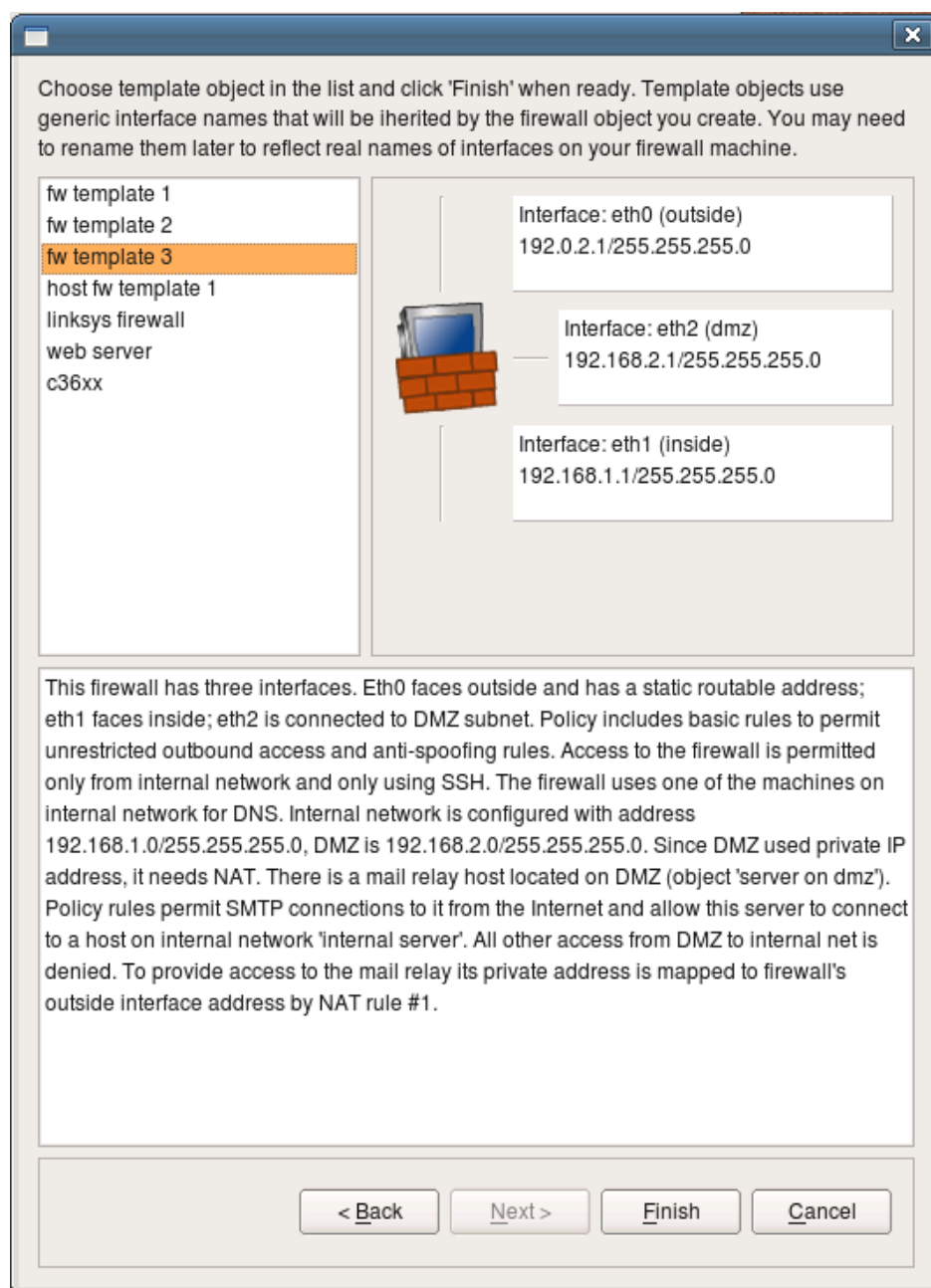
Note that template firewall object comes completely configured, including addresses and netmasks of its interfaces and some basic policy and NAT rules. This configuration is intended as a starting point only. You should reconfigure addresses of interfaces to match those used on your network and most likely will have to adjust rules to match your security policy.



The screenshot shows a window titled "Firewall Builder" with a close button in the top right corner. The window contains the following configuration options:

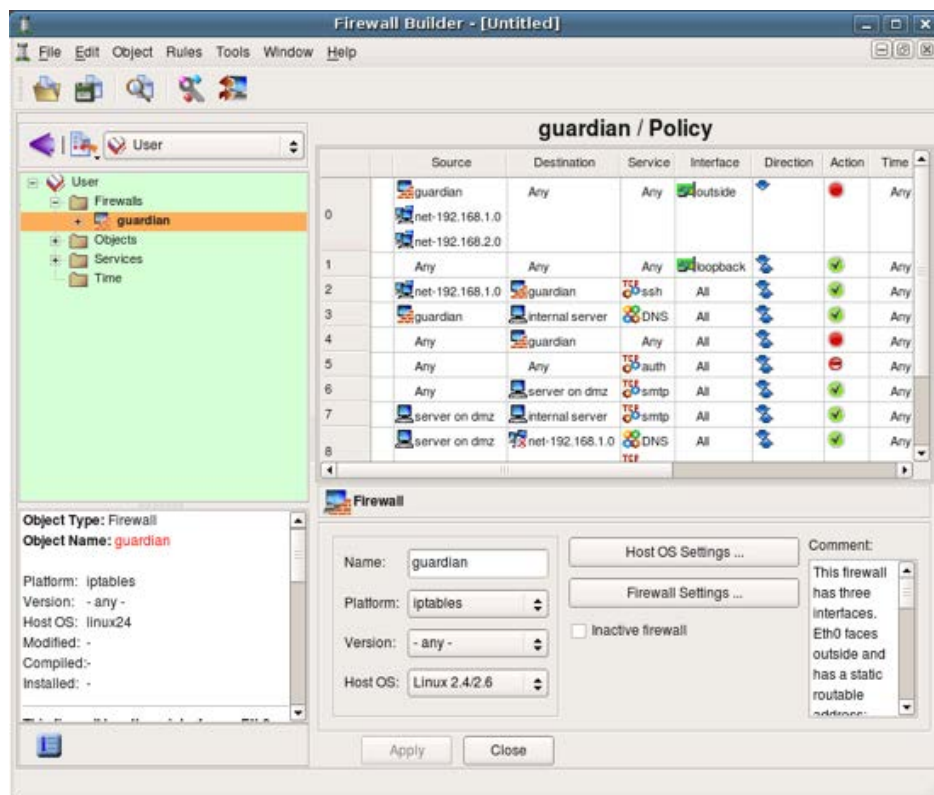
- Enter the name of the new object below:
- Choose firewall software it is running:
- Choose OS the new firewall runs on:
- Use preconfigured template firewall objects
- Template file:
-
-

This page of the wizard shows template objects and their configuration. Standard template objects represent firewalls with two or three interfaces, a host with one interface, web server or Cisco router. Choose firewall with three interfaces for this guide. Note that template comes with completely configured firewall object, including set of interfaces and their ip addresses and some basic firewall policy. You will see how addresses can be changed later on in this guide. Click "Finish" to create a new firewall object using chosen template.



Here is our new firewall object. Its name is **guardian**, it appears in the object tree in the left hand side of the main window in the folder **Firewalls**. When an object is selected in the tree, a brief summary of its properties appears in the panel under the tree. Double-clicking on the object in the tree opens it in the editor panel at the bottom of the right hand side panel of the main window. The editor for the firewall object allows the user to change its name, platform and host OS and also provides buttons that open dialogs for “advanced” settings for the firewall platform and host OS. We will inspect these little later in this tutorial.

You can always resize the main window to make all columns of the policy view be visible.



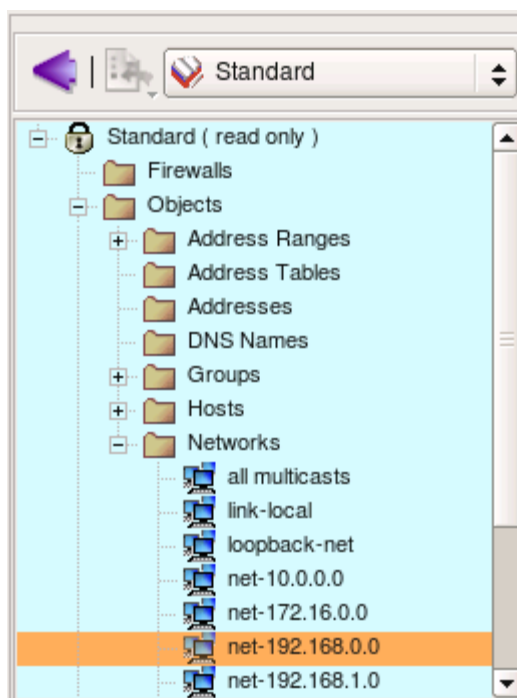
Now would be a good time to save the data to a disk file. This is done in a usual way using main menu **File/Save As**.

Lets take a little tour of the network and service objects that come standard with the program. You can use these preconfigured objects to build policy and NAT rules for your firewall.

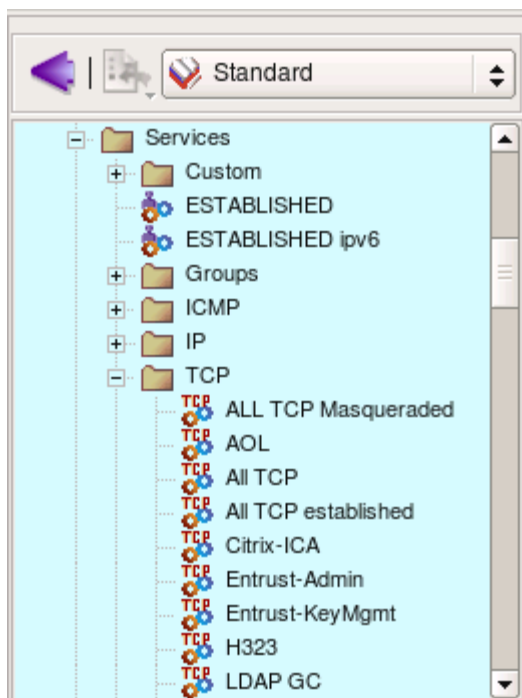
Objects in the tree are organized in libraries, you can switch between libraries using drop-down menu above the tree. Firewall Builder comes with a collection of address, network, service and time interval objects in the library called "Standard". Lets take a look at them. Notice that the background color of the panel that shows objects tree depends on the chosen object library. This makes it easier to keep track of the library currently opened in the program.



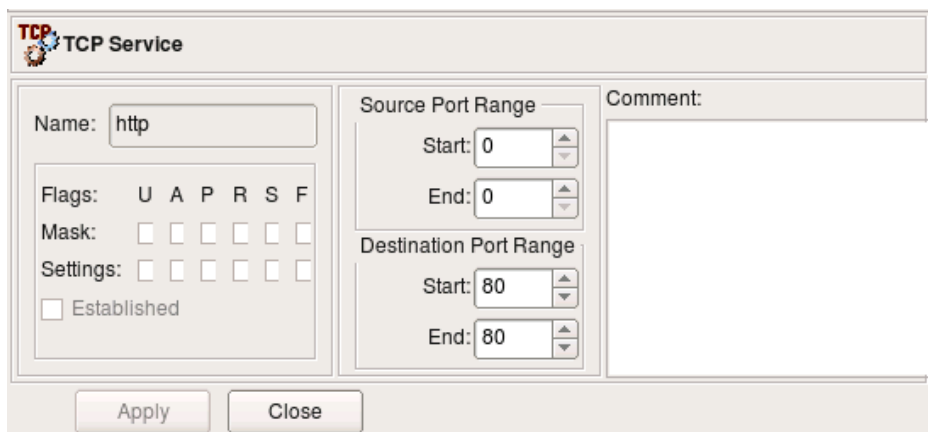
Folder **Objects/Hosts** contains few host objects used in standard firewall templates. Folder **Objects/Network** contains network objects that represent various standard address ranges and blocks, such as multicast, net 127/8, networks defined in RFC1918 and so on.



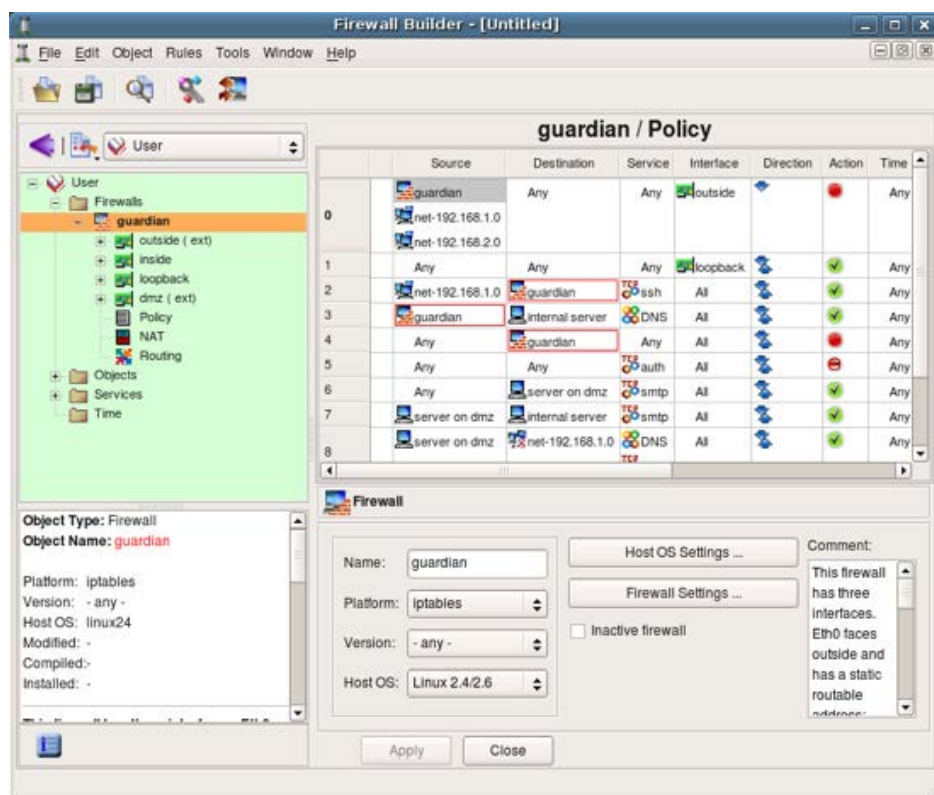
Firewall Builder also comes with extensive collection of TCP, UDP and ICMP service objects that describe commonly used protocols. This slide shows some TCP objects (all of them do not fit in the screenshot).



Here is an example of a simple TCP service. It defines source and destination port ranges (in this case source port range is not defined and there is only one destination port 80). TCP service object can also define any combination of tcp flags the firewall should inspect and also which ones of them should be set in order for a packet to match this object. In the case of the service “http” we do not need to define any flags.



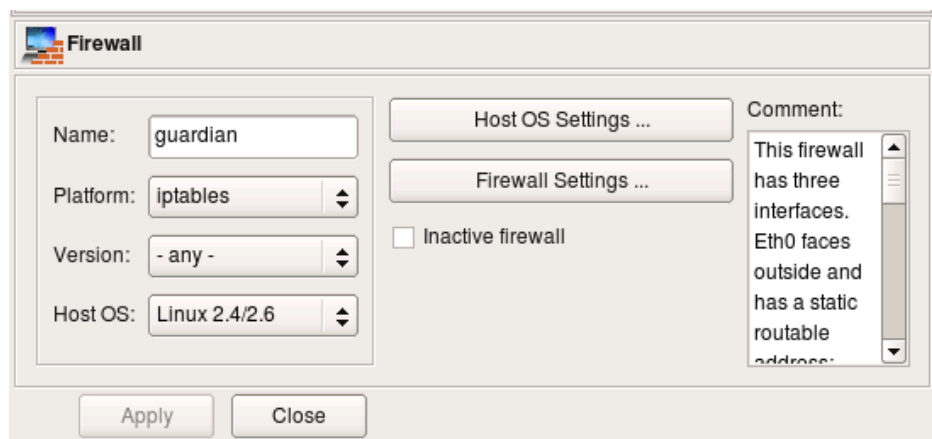
Now lets take a look at the objects created as part of the new firewall object **guardian**. In order to do this, switch to the library **User** where this object was created. To open an object in the editor panel to inspect or change it, double click on it in the tree. Also, if you click on an object in the policy rule to select it, it will automatically open in the tree on the left.



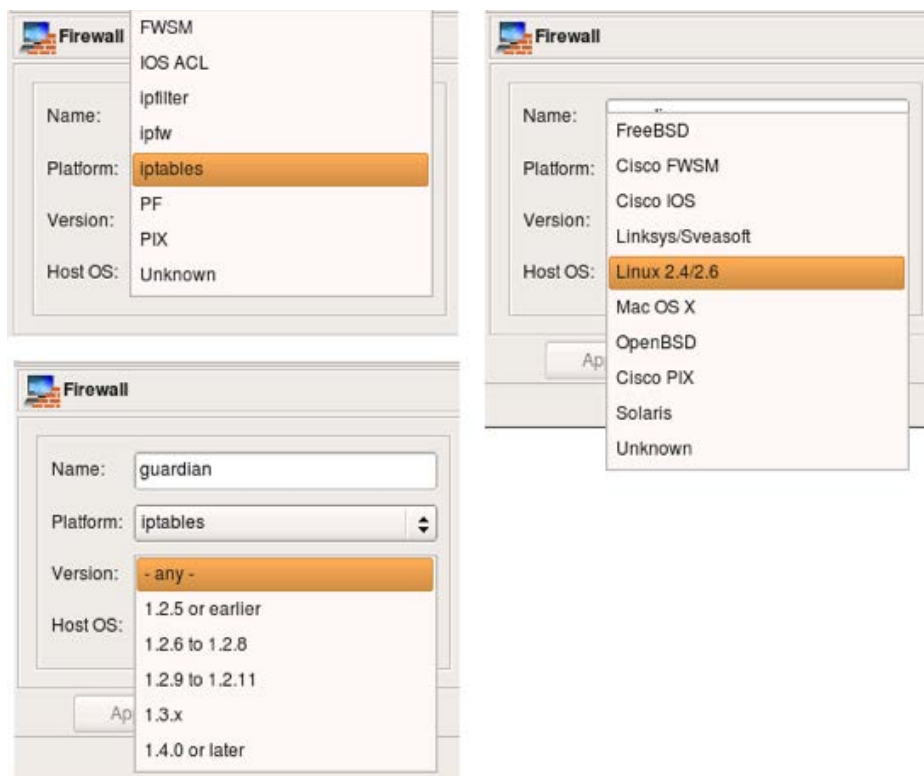
First, the firewall object itself.

Every object in fwbuilder has basic attributes such as its name and comment. Other attributes depend on the object type.

Attributes of the firewall object include platform (can be iptables, pf, ipfilter, etc.), version (platform-dependent) and host OS. Buttons **Host OS Settings** and **Firewall Settings** open dialogs with many additional attributes that depend on the firewall platform and host OS. More on these later.

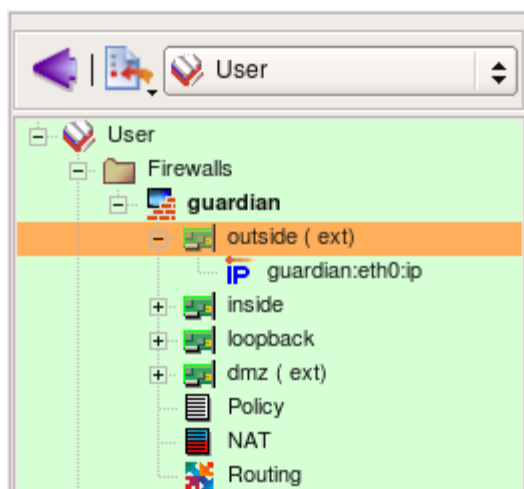


Here are the choices for the firewall platform, version (for iptables) and host OS.



Interfaces of the firewall are represented by objects located below the Firewall object in the tree. We refer to them as “children” of the firewall object. This slide demonstrates properties of the interface eth0. To open it in the editor double click on it in the tree. If editor panel is already open and shows some object, it is sufficient to select new object in the tree to reveal it in the editor panel (no need to double click).

IP and MAC addresses of interfaces are represented by child objects in the tree located below corresponding interface.



Interface object has several attributes that define its function, such as “Management interface”, “external” etc.

- **Name:** the name of the interface object in Firewall Builder must match exactly the name of the interface of the firewall machine it represents. This will be something like “eth0”, “eth1”, “en0”, “br0” and so on.
- **Label:** On most OS this field is not used and serves the purpose of a descriptive label. Firewall Builder GUI uses a label, if it is not blank, to show interfaces in the tree. One of the suggested uses for this field is to mark interfaces to reflect the network topology (‘outside’, ‘inside’) or the purpose (‘web frontend’ or ‘backup subnet’). The label is mandatory for Cisco PIX though, where it must reflect the network topology.
- **“Management interface”:** Sometimes the host has several network interfaces in which case one of them can be marked as the ‘management interface’. The management interface is used for all communication between Firewall Builder and the host.
- **“External interface (insecure)”:** marks an interface that connects to the Internet.
- **“Unprotected interface”:** marks interface to which fwbuilder should not assign any access lists (used only with Cisco IOS platform)
- **“Regular Interface”:** Use this option if the interface has an IP address assigned to it manually.
- **“Address is assigned dynamically”:** Use this option if the interface has a dynamic address (obtained by means of DHCP or PPP or another protocol); in this case an address is unknown at the moment when Firewall Builder generates the firewall policy.
- **“Unnumbered interface”:** Use this option if the interface can never have an IP address, such as the ethernet interface used to run PPPoE communication on some ADSL connections, tunnel endpoint interface, or an interface on a bridging firewall. See below Section 5.3.1 for more detailed discussion of these different types of interfaces.
- **“Bridge port”:** this option is used for port of bridged firewall.
- **“Security level”:** security level of this interface, used only with Cisco PIX (ASA)
- **“Network zone”:** network zone of this interface, used only with Cisco PIX (ASA). Network zone drop-down list shows all network objects and groups of addresses and networks present in the tree. Choose one of them to tell the compiler which networks and blocks of

addresses can be reached through this interface. Compiler uses this information to decide which interface each ACL rule should be associated with based on the addresses used in the destination of the rule.

The screenshot shows the 'Interface' configuration window. The 'Name' field contains 'eth0' and the 'Label' field contains 'outside'. The 'Security level' is set to '0'. The 'This interface is external (i)' checkbox is checked. The 'Regular interface' radio button is selected. The 'Comment' field is empty. The 'Apply' and 'Close' buttons are visible at the bottom.

Here is IP address of interface eth0, external interface of the firewall. The address and netmask are attributes of the child object of the type "IPv4 address". Here the address is "192.0.2.1" and netmask "255.255.255.0". Button "DNS Lookup" can be used to determine ip address using DNS. The program runs DNS query for the "A" record for the name of the parent firewall object.

The screenshot shows the 'IP Address' configuration window. The 'Name' field contains 'guardian.eth0:ip', the 'Address' field contains '192.0.2.1', and the 'Netmask' field contains '255.255.255.0'. The 'DNS Lookup...' button is visible. The 'Comment' field contains the text 'This is a test address, change it to your real one'. The 'Apply' and 'Close' buttons are visible at the bottom.

Lets look at the IP address of the internal interface of the firewall. The address used in the template is 192.168.1.1" with netmask "255.255.255.0". This is rather typical address used for small and home networks. Some commercial firewall appliances come preconfigured with this address.

The screenshot shows the Firewall Builder GUI. On the left, a tree view shows the configuration structure under 'User' > 'Firewalls' > 'guardian' > 'inside' > 'guardian:eth1.ip'. The selected object is highlighted in orange. Below the tree, the object details are shown: Object Type: IPv4 address, Object Name: guardian:eth1.ip, and the address 192.168.1.1/255.255.255.0.

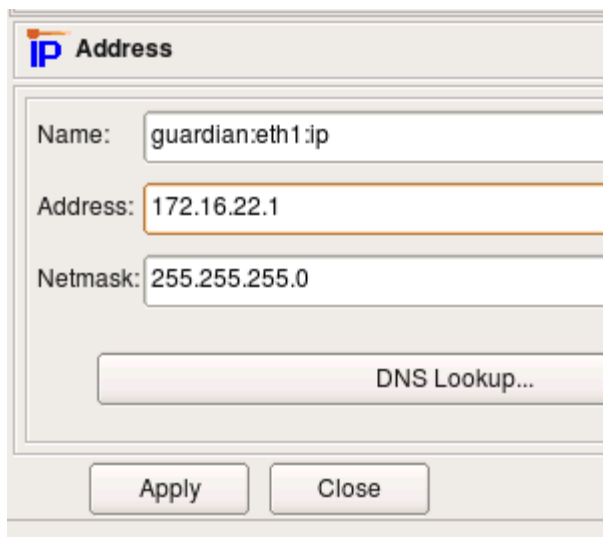
On the right, a table lists firewall rules. The table has columns for rule number, Source, and Destination. Rules 2, 3, and 4 are highlighted with red boxes, indicating they are related to the configuration being shown.

	Source	Dest
0	guardian net-192.168.1.0 net-192.168.2.0	Any
1	Any	Any
2	net-192.168.1.0	guardian
3	guardian	intern
4	Any	guardian
5	Any	Any
6	Any	serve
7	server on dmz	intern
8	server on dmz	net-192.168.1.0

Below the table, the 'IP Address' configuration dialog is open. It shows the following fields:

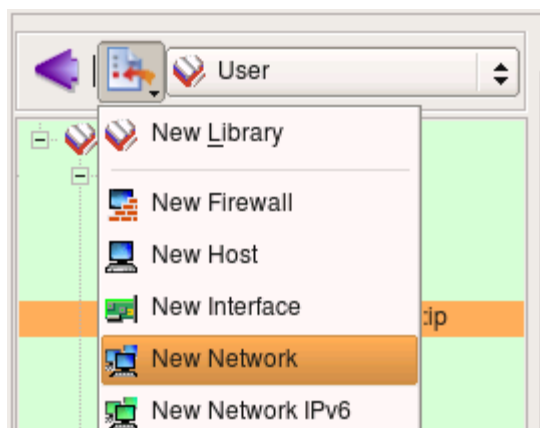
- Name: guardian:eth1.ip
- Address: 192.168.1.1
- Netmask: 255.255.255.0
- DNS Lookup... button
- Apply and Close buttons

If address **192.168.1.0/24** matches address of your local network, you can skip this part of the guide and move to the page 4. Otherwise, you need to reconfigure the address of the internal interface of the firewall object that you just created in fwbuilder and also change address object used in the policy rules. Start with changing address attribute (and possibly netmask, if necessary) of the object **guardian:eth1.ip** as shown in the screenshot:

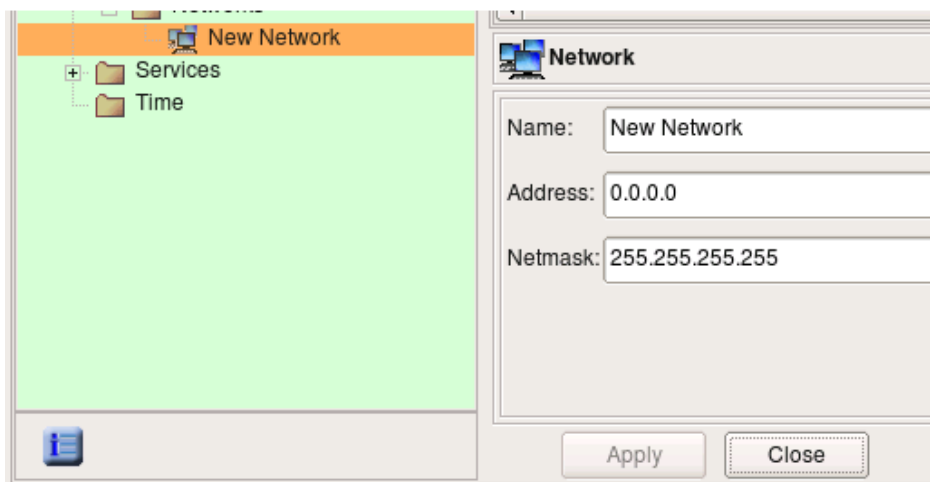


Now we need to change IP address used in the rules. To do this, we create new Network object with correct address and replace object **net-192.168.1.0** in all rules with this new network object.

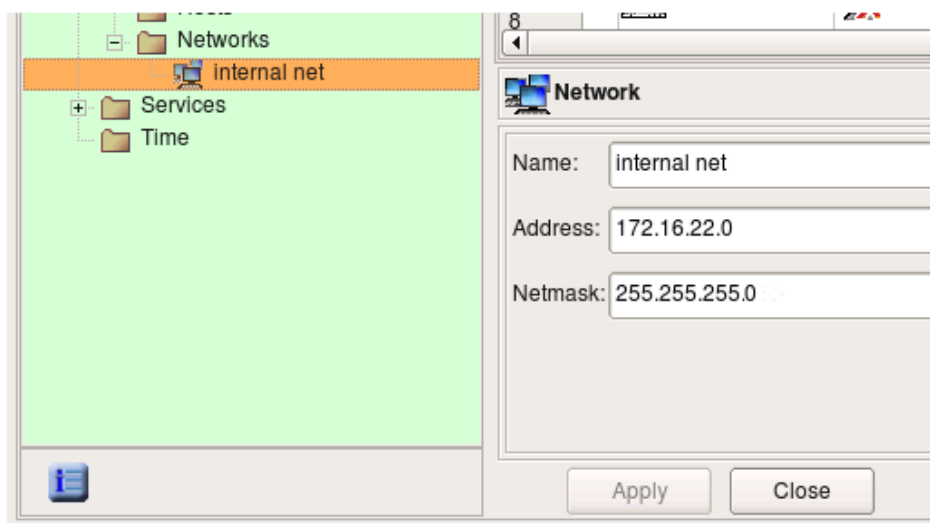
Use new object menu to create Network object.



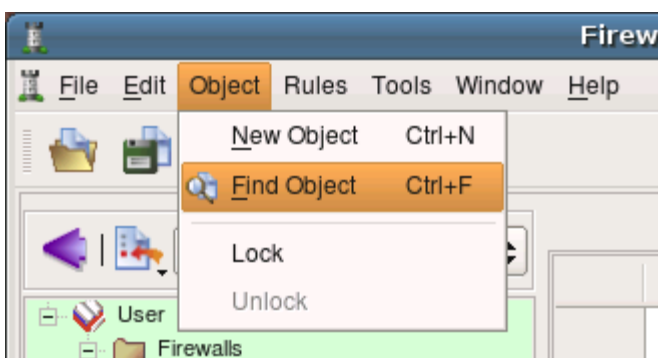
New Network object is created with default name 'New Network' and IP address 0.0.0.0.



Edit object name and address, then hit “Apply”.

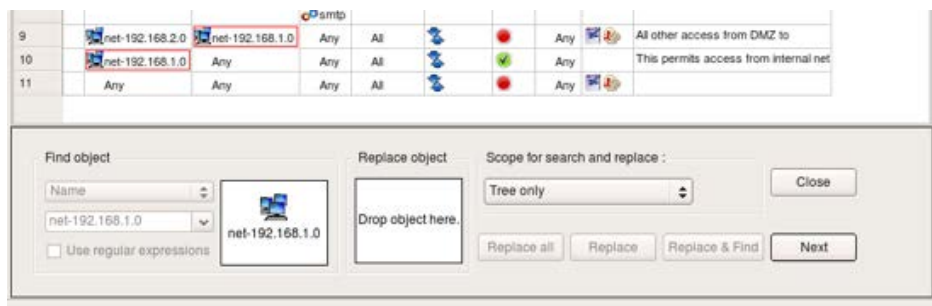


Use menu Object / Find to activate search and replace dialog. The Find and Replace dialog opens at the bottom of the right hand side panel in the main window, below the policy rules view.



Locate object object **net-192.168.1.0** in any policy rule where it is used or

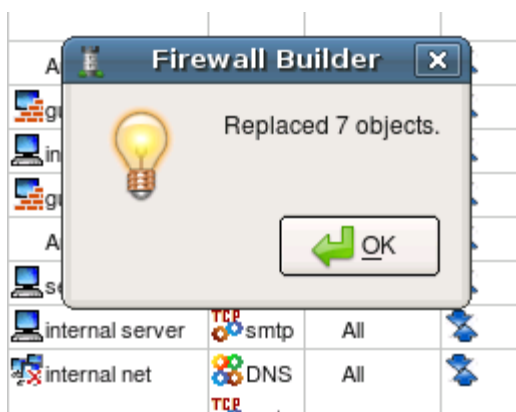
in its location in the tree in library **Standard** and drag and drop it to the left object well in the search and replace dialog as shown on the screenshot:



Change the scope setting to “Policy of all firewalls”. If you have many firewalls in the tree, use scope “policy of the opened firewall” instead. Locate new Network object you just created in the tree and drag and drop it to the right object well in the search and replace dialog as shown on the screenshot:

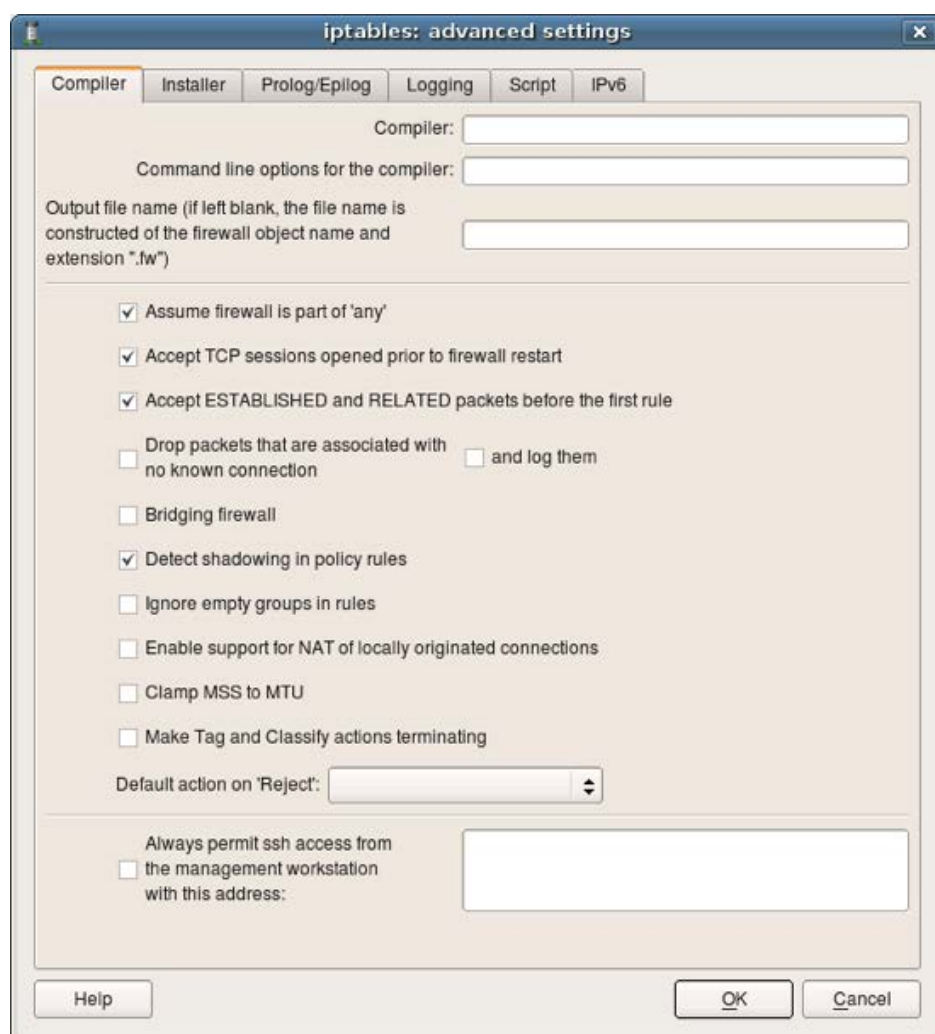


Now hit “Replace all” button. Pop-up dialog should appear and report how many replacements the program had to make in all rules of the firewall. Note that the replacement is done not only in the policy rules, but in NAT rules as well.



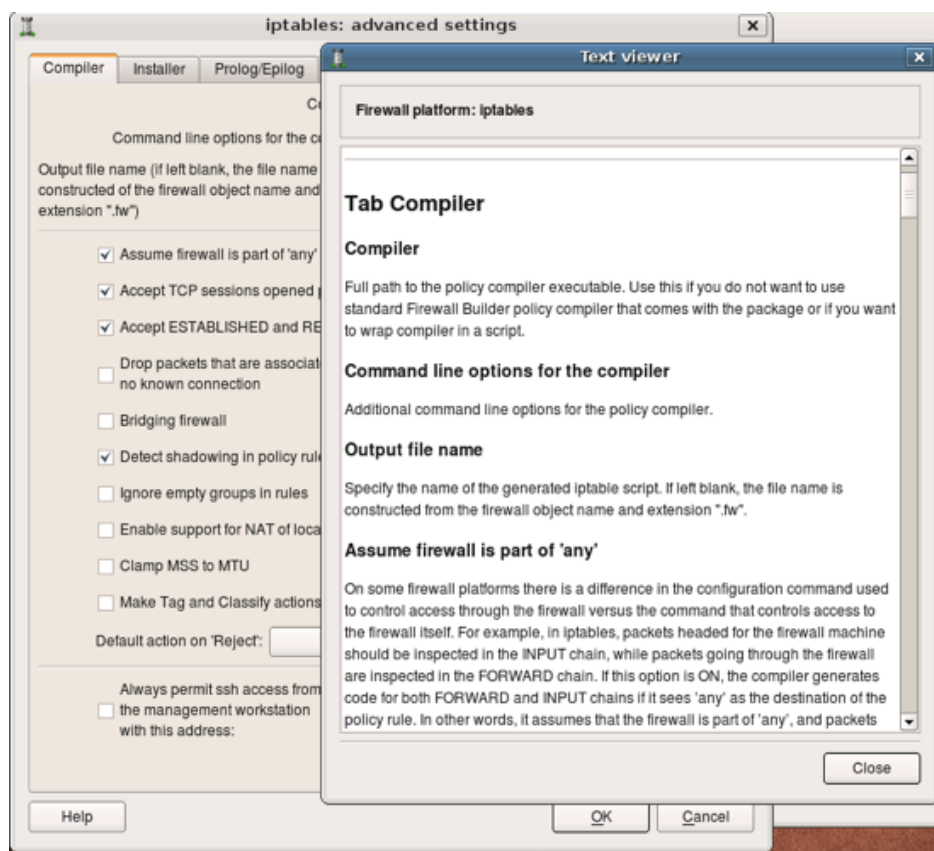
Now that you have created a new object and replaced old network object with new one in all rules, do not forget to save data to a file using menu **File/Save**

Lets inspect properties of the firewall object. Double click on the firewall “guardian” in the tree to open it in the editor panel, then click “Firewall Settings” button in the editor. This opens new dialog that looks like this. Notice button “Help” in this dialog, clicking this button opens help as shown on the next slide.

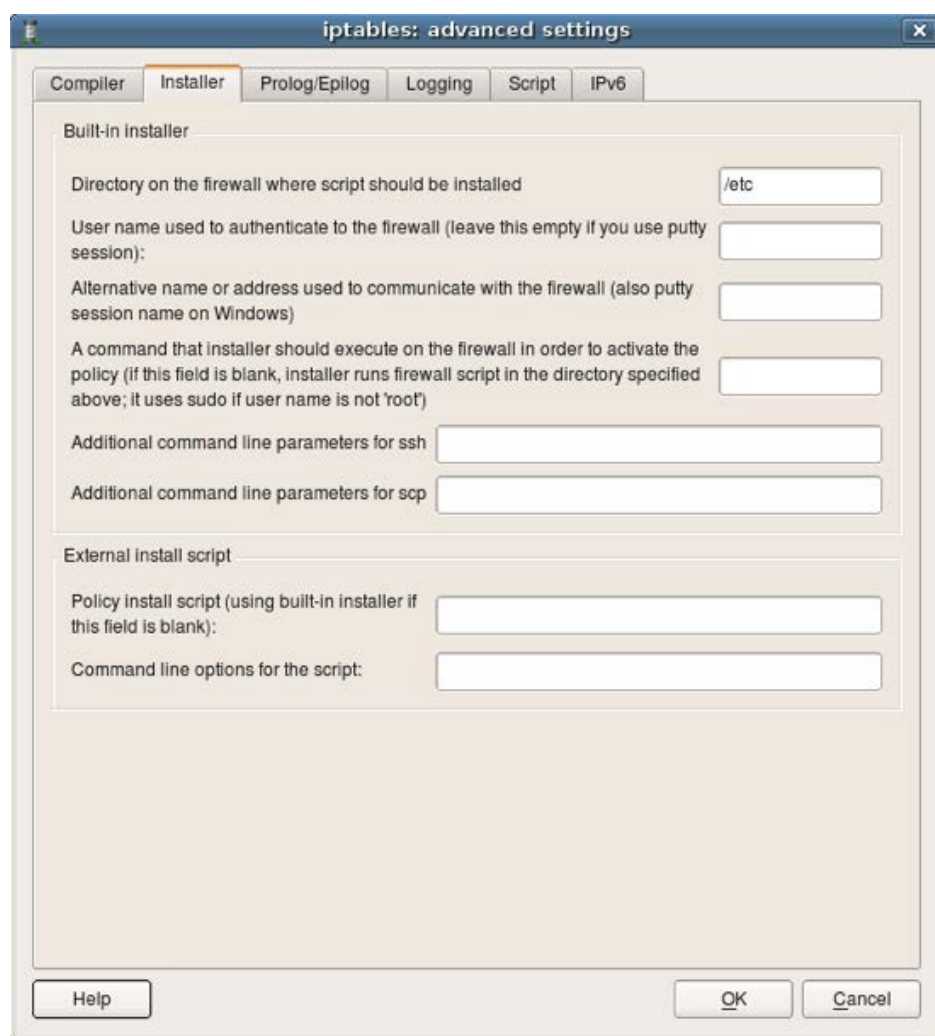


Online help explains all attributes and parameters located in each tab of the firewall settings dialog. I encourage you to explore it as many parameters are important and affect generated iptables script in different ways.

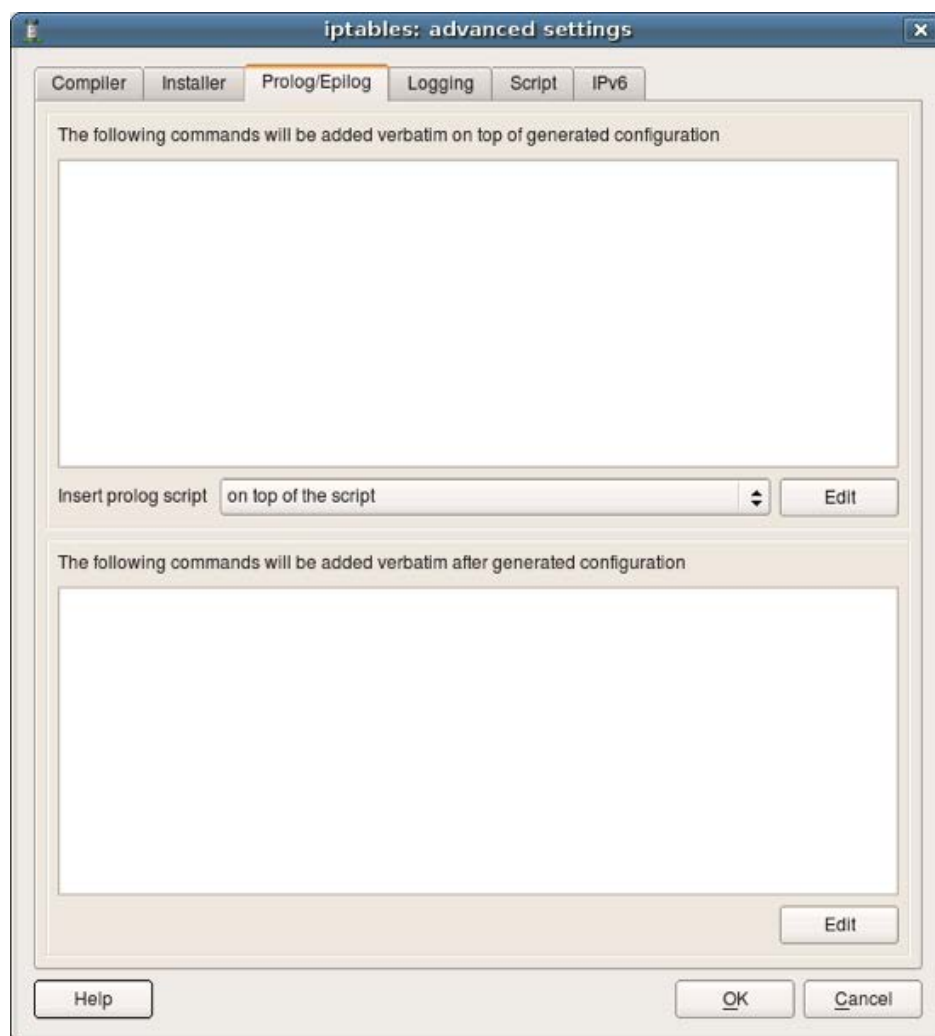
Next few screenshots show other tabs of the firewall settings dialog. You can find detailed explanations of all parameters in the online help.



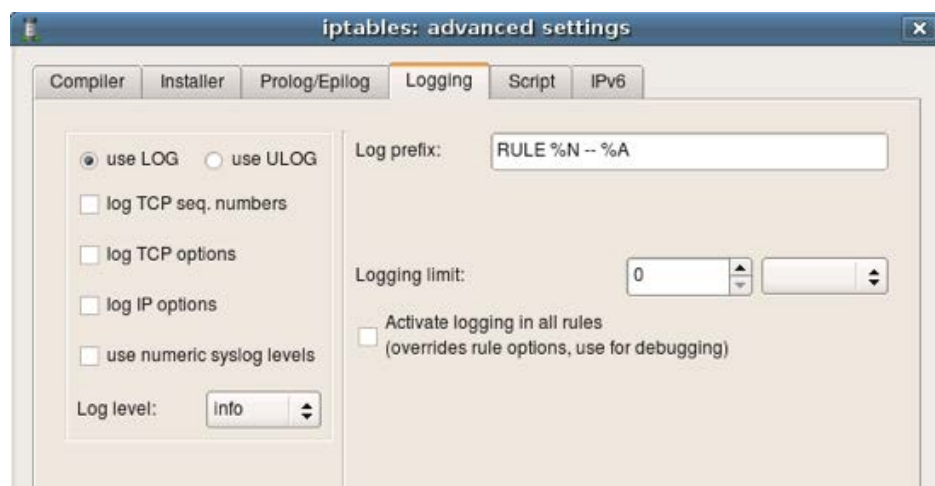
This page defines various parameters for the built-in policy installer. Installer uses ssh client (pscp.exe and plink.exe on Windows) to transfer generated script to the firewall machine and activate it there.



User can define shell commands that will be included in the generated script at the beginning and in the end of it. These commands can do anything you want, such as configure some subsystems, set up routing etc.

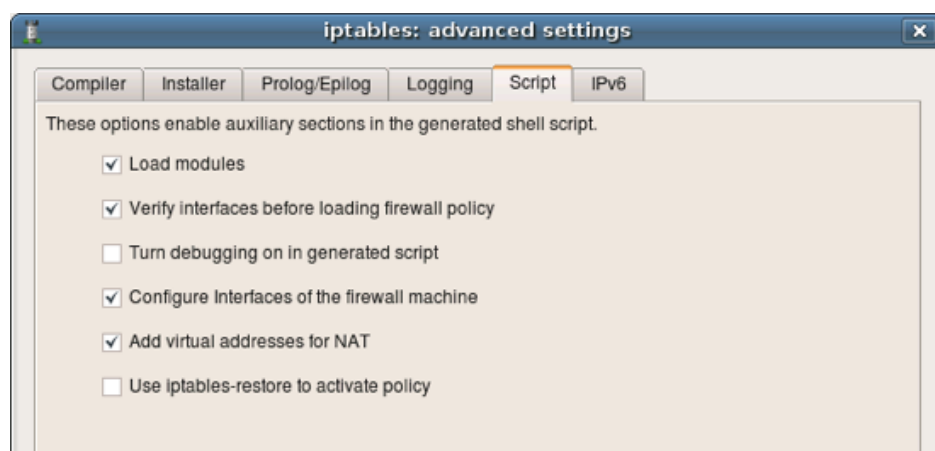


Parameters for logging.

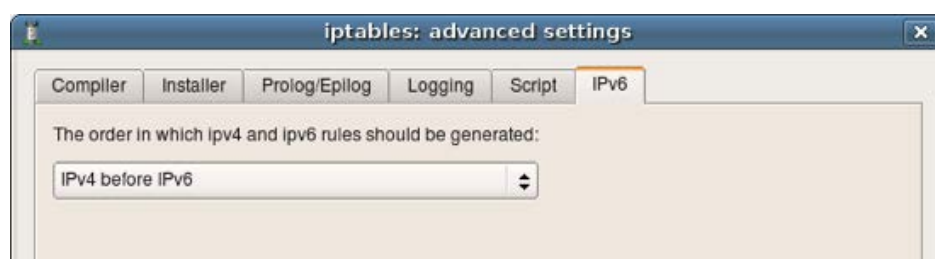


More options for the script generation. Notice that fwbuilder can produce iptables script in two formats: 1) as a shell script that calls iptables utility to add each rule one by one, or 2) it can use iptables-restore script to activate the whole policy at once. Other parameters are explained in the online

help.



Starting with v3.0 Firewall Builder can generate both IPv4 and IPv6 policy. This tab controls the order in which they are added to the script if user defined rules for both address families in the Policy objects of the firewall.



Lets take a look at the policy of the template firewall. These rules are intended to be an example, a starting point to help you create your own policy quicker. Most likely you will want to modify them to suite your requirements. Explanation of the rules given here is rather brief because the goal of this guide was only to demonstrate how to use Firewall Builder.

- Rule 0: this is an anti-spoofing rule. It block incoming packets with source address that matches addresses of the firewall or internal or DMZ networks. The rule is associated with outside interface and has direction set to “Inbound”.
- Rule 1: this rule permits any packets on loopback interface. This is necessary because many services on the firewall machine communicate back to the same machine via loopback.
- Rule 2: permit ssh access from internal network to the firewall machine. Notice service object “ssh” in the column “Service”. This object can be found in the Standard objects library, folder Services/TCP.

guardian / Policy										
	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment	
0	guardian net-192.168.2.0 internal net	Any	Any	outside			Any		anti spoofing rule	
1	Any	Any	Any	loopback			Any			
2	internal net	guardian	ssh	All			Any		SSH Access to firewall is permitted	
3	guardian	internal server	DNS	All			Any		Firewall uses one of the machines	
4	Any	guardian	Any	All			Any		All other attempts to connect to	
5	Any	Any	auth	All			Any		Quickly reject attempts to connect	
6	Any	server on dmz	smtp	All			Any		Mail relay on DMZ can accept	
7	server on dmz	internal server	smtp	All			Any		this rule permits a mail relay	
8	server on dmz	internal net	DNS smtp	All			Any		Mail relay needs DNS and can connect to mail servers on the	
9	net-192.168.2.0	internal net	Any	All			Any		All other access from DMZ to	
10	internal net	Any	Any	All			Any		This permits access from internal net	
11	Any	Any	Any	All			Any			

Policy rules belong to the object “Policy”, which is a child object of the firewall and can be found in the tree right below it. As any other object in Firewall Builder, Policy object has some attributes that you can edit if you double click on it in the tree.

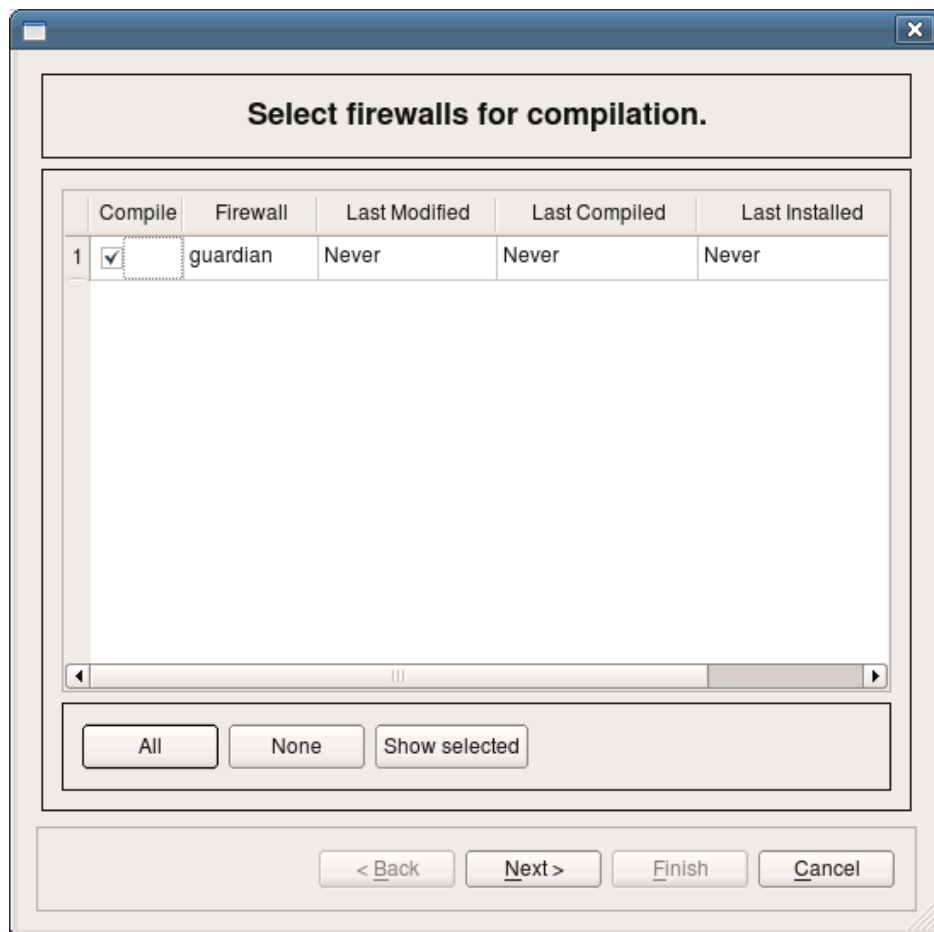
- Policy can be either IPv4, or IPv4 or combined IPv4 and IPv6. In the latter case you can use a mix of IPv4 and IPv6 address objects in the same policy (in different rules) and Firewall Builder will automatically figure out which one is which and will sort them out.
- Policy can translate to only mangle table, or a combination of filter and mangle tables. Again, in the latter case policy compiler decides which table to use based on the rule action and service object. Some actions, such as “Tag” (translates into iptables target MARK) go into mangle table.
- “Top ruleset” means that compiler will place generated iptables rules into built-in chains INPUT/OUTPUT/FORWARD. If policy is not marked as “top ruleset”, generated rules will go into user-defined chain with the name the same as the name of the policy object.

Here are preconfigured NAT rules.

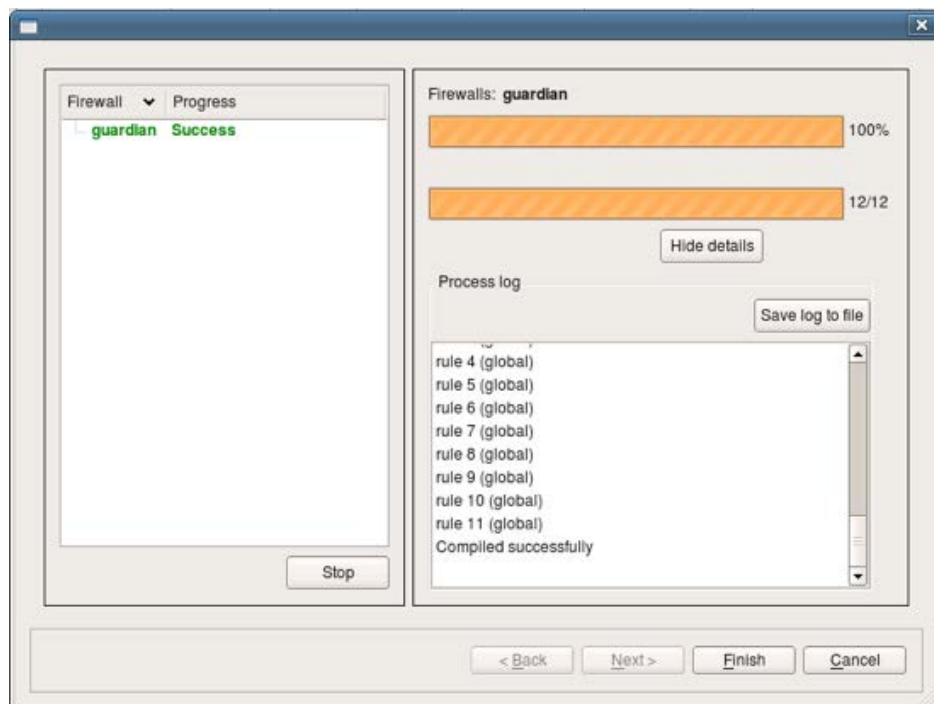
- Rule 0: tells the firewall that no address translation should be done for packets coming from network 192.168.2.0 going to 192.168.1.0 (because Translated Source, Translated Destination and Translated Service are left empty)
- Rule 1: packets coming to the firewall from internal and DMZ networks should be translated so that source address will change and become that of the outside interface of the firewall.
- Rule 2: packets coming from the Internet to the interface “outside” will be translated and forwarded to the internal server on DMZ represented by the host object “server on dmz”.

guardian / NAT								
	Original Src	Original Dst	Original Srv	Translated Src	Translated Dst	Translated Srv	Options	Comment
0	net-192.168.2.0	internal net	Any	Original	Original	Original		no need to translate
1	net-192.168.2.0 internal net	Any	Any	outside	Original	Original		Translate source address for outgoing connections
2	Any	outside	Any	Original		server on dmz	Original	

Now we should be ready to compile policy of the firewall **guardian** and generate iptables script. To do this, select firewall in the tree and click right mouse button. Choose item “Compile” in the pop-up menu. The dialog that appears lists all firewall objects defined in the objects tree and lets you select which ones should be compiled. The firewall **guardian** has just been created and has never been compiled and dialog shows that. Make sure checkbox next to the firewall object **guardian** is checked and click button “Next”.



Firewall Builder calls policy compiler (which is by the way an external program which can be used on the command line). The next page of the dialog shows compiler progress and result.



Compiler generates iptables script in the file with the name the same as the name of the firewall object, with extension “.fw”. The file is placed in the same directory where the data file .fwb is located.

```
$ ls -la test2.fwb guardian.fw

-rwxr-xr-x 1 vadim vadim 11253 2009-02-16 16:41 guardian.fw

-rw-r--r-- 1 vadim vadim 24696 2009-02-16 16:41 test2.fwb
```

Here is how generated script looks like. This is just a fragment from the middle to show some generated iptables commands.

```
# ===== IPv4
# ===== Table 'filter', automatic rules

$IPTABLES -P OUTPUT DROP
$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP

cat /proc/net/ip_tables_names | while read table; do

$IPTABLES -t $table -L -n | while read c chain rest; do

if test "X$c" = "Xchain" ; then
$IPTABLES -t $table -F $chain
fi
done

$IPTABLES -t $table -X
done

$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A OUTPUT -m state --state ESTABLISHED,RELATED -j
ACCEPT
$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j
ACCEPT

# ===== Table 'nat', rule set NAT
```

```
# NAT compiler errors and warnings:

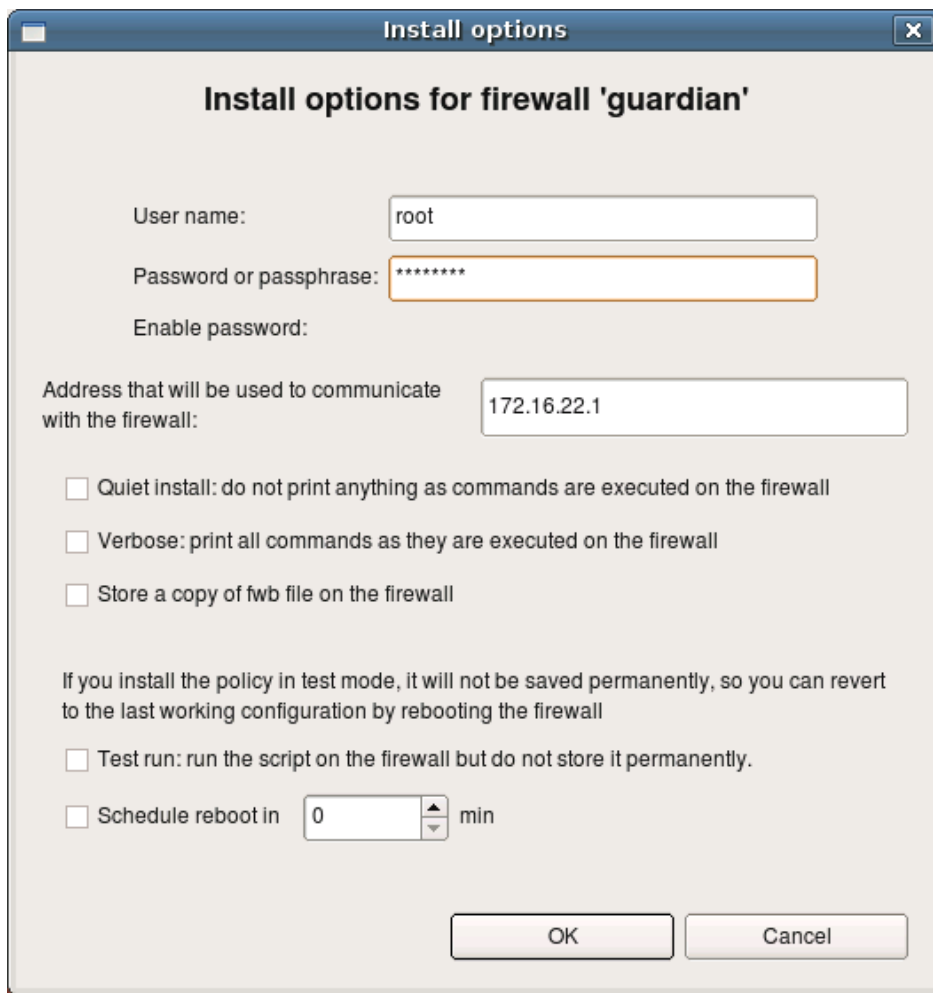
#
#
# Rule 0 (NAT)
#

echo "Rule 0 (NAT)"

#
# no need to translate
# between DMZ and
# internal net

$IPTABLES -t nat -A POSTROUTING -s 192.168.2.0/24 -d
172.16.22.0/24 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -s 192.168.2.0/24 -d
172.16.22.0/24 -j ACCEPT
```

Now you can transfer it to the firewall and execute it there to install iptables rules. However it is much more convenient to use built-in policy installer to do this. To use installer, click right mouse button on the firewall object in the tree and use menu item **Install**. Firewall Builder will compile the policy if necessary and then open dialog where you can configure parameters of the installer. Here you need to enter password to authenticate to the firewall. Once you click **OK**, installer will connect to the firewall using ssh client. First, it will copy generated script to the directory **/etc** on the firewall (or different one, if configured in the **Installer** tab of firewall settings dialog), then it will run this script and check for errors. Its progress will be visible in the panel of the installer wizard, just like the progress of policy compiler.



The screenshot shows a dialog box titled "Install options" for a firewall named "guardian". The dialog has a title bar with a close button. The main content area is titled "Install options for firewall 'guardian'". It contains several input fields and checkboxes:

- User name:** A text box containing "root".
- Password or passphrase:** A text box containing "*****".
- Enable password:** A checkbox that is currently unchecked.
- Address that will be used to communicate with the firewall:** A text box containing "172.16.22.1".
- Options:** Three checkboxes:
 - Quiet install: do not print anything as commands are executed on the firewall
 - Verbose: print all commands as they are executed on the firewall
 - Store a copy of fwb file on the firewall
- Test mode:** A checkbox labeled "Test run: run the script on the firewall but do not store it permanently." which is unchecked.
- Schedule reboot:** A checkbox labeled "Schedule reboot in" followed by a spin box containing "0" and the text "min".

At the bottom of the dialog are two buttons: "OK" and "Cancel".

This guide walked you step by step through the process of creating of a firewall object, making some minor changes in its parameters and policy rules, compiling the policy and activating it on the firewall machine. This guide did not touch advanced topics such as built-in revision control system, working with multiple data files, working with multiple firewall objects, IPv6. You can find documentation and guides on these topics and more on our project web site at <http://www.fwbuilder.org>.

Category: Security Ubuntu Tags: firewall , fwbuilder , Security

← Nice Ubuntu Wallpapers

Do You Have A How-To You'd Like
To Share Here? →

7 thoughts on "Getting Started with Firewall Builder"



Kurt von Finck
2009/06/04

Couldn't you just paste

http://www.fwbuilder.org/slideshows/tutorial_3/sl...

into the blog entry and be done with it?

Reply ↓



rtafcott
2009/06/04

Kurt, hurtling toward you fate in Albuquerque...I can appreciate that...I'd definitely prefer to be hurtling elsewhere!

Reply ↓



rtafcott
2009/06/04

should be YOUR fate

rt

Reply ↓



alifaan
2009/06/04

will installation of firewall builder erase my existing iptables configuration?

Reply ↓



Daniel
2012/09/15

Do I need to install a desktop GUI on my ubuntu server 11.04, before I can use fwbuilder?

Regards

Reply ↓



Daniel
2012/09/15

Hope someone can help me

Reply ↓



Me
2013/02/28

[@Daniel](#)

Yes you need a desktop GUI for it

Reply ↓

Leave a Reply

Copyright 2006-2018

Christer Edwards

Iconic One Theme | Powered by Wordpress