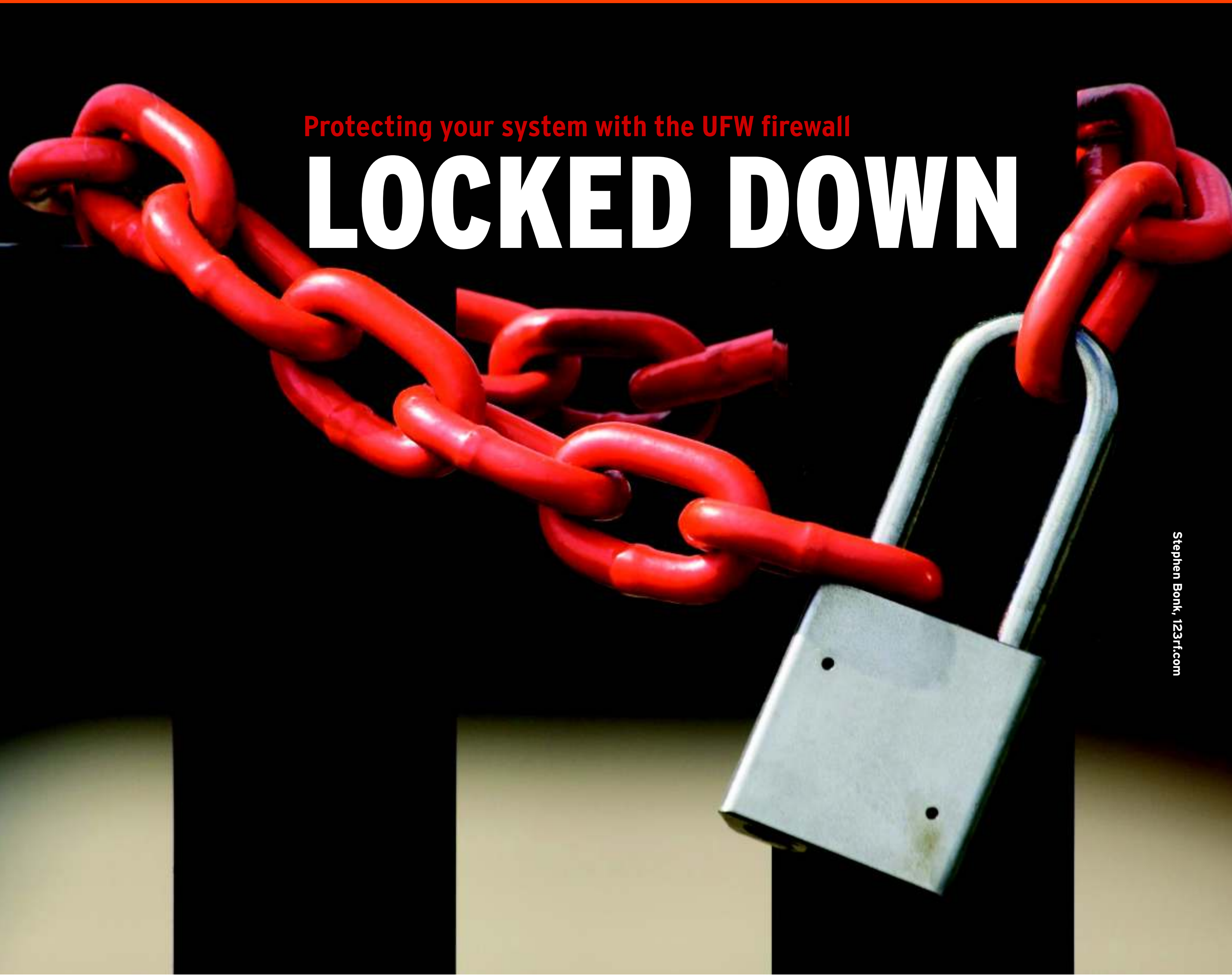


Protecting your system with the UFW firewall

LOCKED DOWN

Stephen Bonk, 123rf.com

Careful users keep an eye on security. Ubuntu's firewall tools will help you keep intruders off your system.

BY JAMES STANGER

When it comes to security, Ubuntu offers several advantages over a system like Windows. For instance, Ubuntu's code is open source, which means hundreds and even thousands of developers review and improve the code. Also, unlike Windows, Linux systems don't have a registry, a feature that has caused Windows systems to succumb to worms, viruses, and various other attacks over the years. Linux systems are also loosely integrated, which means just because one subsystem goes bad, the rest of the system doesn't have to crash or become unstable.

Although Linux is often considered safer than other comparable desktop systems, no operating system is inherently secure – not even Ubuntu. An attacker might try to scan your system to identify an open port, and the application listening on a port might have its own security issues that could lead to compromise.

But regardless of how systems or applications might compare in an abstract sense, the specific reasons really don't even matter: Configuring a personal firewall is important, and practicing safe computing helps you understand your system and brings you a bit more peace of mind.

Introducing UFW

UFW, which stands for "Uncomplicated Firewall," is Ubuntu's official firewall configuration utility [1][2]. The command-based UFW works quite well as a front end for the powerful set of Linux firewall utilities known as iptables [3]. Although UFW appears to the user as a complete firewall system, all it really does is provide a slightly easier syntax for using the iptables.

The iptables utilities, in turn, operate the Netfilter framework found in any Linux implementation. Netfilter and iptables are complicated enough to scare off many end users, so UFW (and its GUI counterpart GUFW) were created to bring the benefits of firewalls to the everyday Ubuntu desktop user.

Getting Started with UFW

To start UFW from any standard Jaunty installation, open a terminal window

What Is a Personal Firewall?

The term *firewall* usually refers to a device located between an internal network (such as a home or corporate network) and a larger, less controlled network such as the Internet. Although modern firewalls come with a wide range of capabilities, the most basic role of a firewall is to restrict access to the internal network by filtering incoming traffic. A personal firewall acts much like a network firewall, except that a personal firewall is typically a software-based component residing on a single system.

A personal firewall can block (or, conversely, allow) traffic based on parameters such as the source IP address, the destination address, or the port number. Services running on your Ubuntu system listen for incoming connections to a specific TCP or UDP port, so the port number provides an indication of the an outside connection is attempting to reach. See Table 1 for a summary of some common service ports.

Many personal firewall can also block or allow traffic based on the network protocol, and several firewall systems also support logging so that you can keep a record of traffic statistics and outside access attempts.

and type *ufw*. As with many of Ubuntu's administrative tools, you need to specify the root password in order to use UFW:

```
$ sudo ufw enable
Firewall is active and ↩
enabled on system startup
```

The result of the preceding command is that UFW will run each time your system starts.

After issuing this command, you can then set a default policy:

```
$ sudo ufw default deny
Default policy changed to 'deny'
(be sure to update your rules ↩
accordingly)
```

This command tells UFW to deny all ICMP packets, as well as the protocols listed in Table 1, automatically. Then you can proceed to create exceptions to this default closed policy. For example, the following commands allow connections to the SSH, web, and VNC servers running your system:

```
$ sudo ufw allow 22
Rule added
$ sudo ufw allow 80
Rule added
$ sudo ufw allow 5900
Rule added
```

You can verify UFW's status by issuing the following command:

```
$ sudo ufw status
```

The output for this command is shown in Listing 1.

Allowing Specific Ports, Addresses, and Ranges

So far, the commands have only specified a port number, not an actual protocol. Filtering for the TCP or UDP protocol is also quite easy:

```
sudo ufw allow 53/udp
Rule added
```

The preceding command allows connections to your system's DNS server, which listens to UDP port 53 by default.

To allow users to connect to only TCP port 80, issue the following command:

```
sudo ufw allow 80/tcp
Rule added
```

The previous examples add rules for all IP addresses. You can get much more specific about which addresses you want to associate with your rules. For example, suppose you have a default closed policy enabled on your personal firewall and you want to allow the system at *19.82.44.45* to connect to all ports on your local system. To make this possible, you would issue the following command:

```
sudo ufw allow from 19.82.44.45
Rule added
```

Also, it is possible to specify a range of IP addresses. For example, ICANN, the body that allocates IP addresses on the

Table 1: Common Service Ports

Service Name	Description	Port Number
FTP	File Transfer Protocol server.	TCP 20 and 21 (mostly listed as 21)
POP3	The most popular email mailbox protocol. The POP3 server allows you to log in and download your messages.	TCP 110
IMAP	Internet Message Access Protocol – A newer, more sophisticated mailbox protocol that is still not as popular as POP3.	TCP 143
SMTP	Simple Mail Transfer Protocol – Email protocol used for sending and forwarding outgoing messages	TCP 25
NFS	Network File System – Used in Linux and Unix systems to share files and mount remote directories.	TCP and UDP port 2049; also relies on the Portmapper service, which uses UDP port 111
SSH	A relatively secure way to access and control a system, usually via the command line. Uses encryption and public-key-based authentication. You can also use SSH to tunnel unencrypted protocols.	TCP 22
Telnet	An older protocol (mostly replaced by SSH) used to control remote systems.	Server listens on TCP 23; you can use a Telnet client to connect to any port
VNC	Virtual Network Computing – Protocol that allows you to log in to a remote computer and view the remote X Window session as if you were sitting in front of it.	TCP 5900, by default
IPP	Internet Printing Protocol – Used to connect printers across networks.	UDP and TCP 631.

Default Policy

You can configure a personal firewall to take one of the following default configurations:

- Default open
- Default closed

UFW allows you to choose the default stance. I follow conventional wisdom and choose the default closed configuration, because it’s generally better to disallow everything and then permit only specific traffic. The advantage to this policy is that you have granular control over the types of connections your system allows.

The disadvantage is that you might inadvertently block connections to your system and will have to spend time creating exceptions to this default policy. But as you will learn in this article, creating these exceptions is relatively easy.

Listing 1: ufw Status Output

```
Status: active



To                            Action  From

--                            ------  ----

22                            ALLOW   Anywhere

80                            ALLOW   Anywhere

5900                          ALLOW   Anywhere
```

Internet, created some ranges of private IP addresses that can’t be routed on the Internet. These addresses are often assigned by DHCP servers operating behind a firewall.

To allow all of the RFC 1918 “private IP” addresses to access your system, you would issue the following commands:

```
$ sudo ufw allow from 10.0.0.0/8
rule added
$ sudo ufw allow from ➥
172.16.0.0/12
rule added
$ sudo ufw allow from ➥
192.168.0.0/16
rule added
```

Of course, you can specify any IP address range you like.

Limiting Connections

One of the more powerful features of UFW is the ability to limit connections. For instance, you can specify that your UFW personal firewall system will allow no more than six connection attempts over a period of 30 seconds.

Won’t It Block All Access?

Many users won’t enable a personal firewall because they’re afraid they then won’t be able to visit their favorite video site or download programs. That’s really not going to happen. The developers who create personal firewall software are very smart; they know that you want to get out to all the good stuff on the Internet. They also know there are plenty of bad guys who are trying to get into your computer.

The personal firewall software generally blocks incoming access to your system; it won’t hamper your ability to go out and find what you want. Unless, of course, you specifically tell it to.

Limiting the number of connection attempts can help you thwart the following types of attacks:

- Scanning: If connections are limited, scans are less accurate.
- Denial of service: A denial of service attack involves sending floods of packets or malformed packets to a host in the attempt to either crash the system or overwhelm it. Limiting connections can help the system ignore flooding attempts.
- Brute force: A brute force attack is where an attacker repeatedly tries to guess a user name or password combination. Limiting connections causes a reset to occur, causing the attacker to take far longer to find a successful username/password combination.

To enable connection limiting, use the following syntax:

```
ufw limit service_name/ ➥
protocol
```

For example, to limit connections for your web server, you could issue the following command:

```
$ sudo ufw limit www/tcp
rule added
```

Removing a Rule

If you want to remove a rule, specify the port. For example, to remove the rule allowing SSH, issue the following command:

```
$sudo ufw delete allow 22
Rule deleted
```

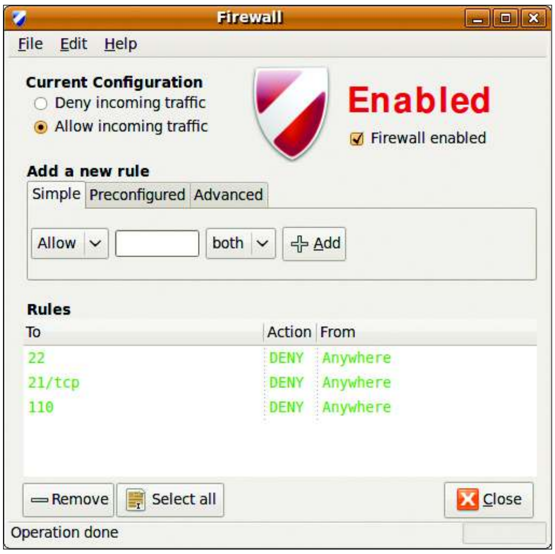
If you want to change the default stance of your firewall, simply issue the following command:

```
$ sudo ufw default allow
```

However, understand that all of the existing rules you have created will still exist, so it is best if you stick with a default deny policy, then create rules to allow specific traffic.

GUFW

Operating your firewall from the command line gives you the most control over your personal firewall, but you don’t have to use the command line if



Firewall

Current Configuration

○ Deny incoming traffic
● Allow incoming traffic

Enabled

☑ Firewall enabled

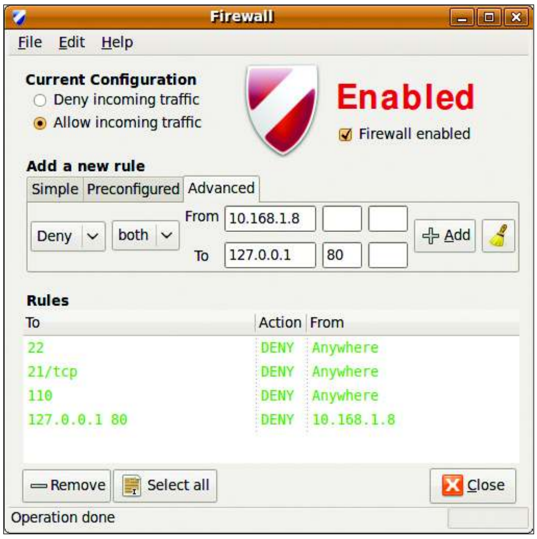
Add a new rule

Rules

To	Action	From
22	DENY	Anywhere
21/tcp	DENY	Anywhere
110	DENY	Anywhere

Operation done

Figure 1: UFW, showing a simple firewall configuration.



Current Configuration
○ Deny incoming traffic
● Allow incoming traffic

Enabled
☑ Firewall enabled

Add a new rule
Simple | Preconfigured | Advanced

Deny | both | From 10.168.1.8 | To 127.0.0.1 80

Rules

To	Action	From
22	DENY	Anywhere
21/tcp	DENY	Anywhere
110	DENY	Anywhere
127.0.0.1 80	DENY	10.168.1.8

Operation done

Figure 2: Specifying advanced port features in GUFW.

you don't want to. Jaunty provides a fairly robust GUI application known as GUFW (Figure 1). To access GUFW, go to *System* | *Administration* in the Ubuntu main window, then select *Firewall configuration*. Just as with UFW, you will need to provide your root password to launch GUFW.

Getting Started with GUFW

Once you've launched GUFW, you will see that three tabs are available:

- Simple: Allows you to add a particular port number or service quickly.
- Preconfigured: Instead of requiring you to specify a service, GUFW provides a ready-made list for you.
- Advanced: Allows the creation of more sophisticated rules, including source and destination ports and IP addresses.

Figure 2 shows how you can add an advanced rule that denies both TCP and UDP traffic from the system at IP address 10.168.1.8 to the local system.

Logging

Logging is a vital function of any personal firewall. To enable logging in GUFW, go to *Edit* | *Preferences*, and then select the *Enable ufw logging* checkbox.

To enable logging from the command line, enter:

```
ufw logging level
```

If you use the GUFW or the *ufw* command to turn logging on without any arguments, the logging level will automatically be set to low. To set the level to full (the highest level of logging), issue the following command:

```
$ sudo ufw logging full
logging enabled
```

Specifying the full or even the *high* setting will cause your system to log an enormous amount of connections, which might fill up your hard drive. I recommend specifying low logging, which is the default.

Logging isn't just an all or nothing proposition. You might want to log only specific connections. The following command tells ufw to log connections to your VNC server (port 5900):

```
ufw allow log 5900/tcp
```

The following command disables all logging:

```
$ sudo ufw logging off
logging disabled
```

Some Final Tips

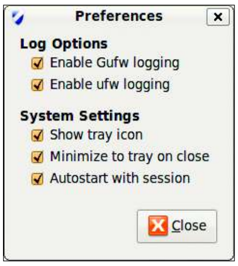
Most users configure GUFW to start when they log on to the GUI desktop system. To ensure that this happens, start GUFW, then go to *Edit* | *Preferences* to open the Preferences dialog box, shown in Figure 3.

Under the *System Settings* section, select all three checkboxes, as shown. Next time, when you boot your system and log in, GUFW will be running automatically, nicely tucked way in your status bar.

Once you have created a rule set that you like, you might want to save a backup. To do so, go to *File* | *Export rules*, and then save the configuration as a simple text file. Then you can load this configuration back into your system – or onto another system – by going to *File* | *Import rules*.

Conclusion

My advice to you is to play around with this firewall a bit. One person's perfect firewall is another's perfect nightmare. I guess that's why they call it a "personal firewall."



Log Options
☑ Enable Gufw logging
☑ Enable ufw logging

System Settings
☑ Show tray icon
☑ Minimize to tray on close
☑ Autostart with session

Figure 3: Setting up GUFW to run automatically in Jaunty.

If I could propose a one-size-fits-all solution, I'd offer up a text file you could just import into the firewall, and you'd be finished.

But it really doesn't work that way for a couple of reasons. First of all, your needs are different than mine. Second, it's important for you to take responsibility over your computer and customize it for your needs. If you work on learning these settings, we'll all be one step closer to a safer Internet. ■

Sidebar: Installing UFW and GUFW

If, for some reason, UFW and GUFW aren't installed on your Ubuntu system by default, make sure that the Universe repository is enabled, and then issue the following commands:

```
sudo apt-get install ufw
sudo apt-get install gufw
```

Other Firewalls

If you don't want to use UFW, you don't have to. Other firewall creation utilities have existed for years, including IPCop [4], SmoothWall [5], Endian [6], or Arno-iptables [7]. Some of these products are pretty robust and provide more features than a typical end user really needs. Still, it's good to know about these tools in case you ever need to use them in the future.

INFO

[1] The Ubuntu UFW Wiki: *https://wiki.ubuntu.com/UbuntuFirewall*

[2] The Ubuntu UFW Forum: *http://ubuntuforums.org/showthread.php?t=823741*

[3] Netfilter/iptables: *http://www.netfilter.org/*

[4] The GUFW home page: *http://gufw.tuxfamily.org/index.html*

[5] IPCop: *http://www.ipcop.org*

[6] SmoothWall: *http://www.smoothwall.org*

[7] Endian: *http://www.endian.com*

[8] Arno-iptables: *http://freshmeat.net/projects/iptables-firewall/?topic_id=151*