Hardening the Linux server

An introduction to GNU/Linux server security

Jeffrey Orloff
Director of IT/Security
SafeWave, LLC

Skill Level: Intermediate

Date: 17 Dec 2008

Servers—whether used for testing or production—are primary targets for attackers. By taking the proper steps, you can turn a vulnerable box into a hardened server and help thwart outside attackers. Learn how to secure SSH sessions, configure firewall rules, and set up intrusion detection to alert you to any possible attacks on your GNU/Linux® server. Once you've gained a solid foundation in the basics of securing your server, you can build on this knowledge to further harden your systems.

Section 1. Before you start

In this tutorial, you learn the basics of securing a GNU/Linux server and gain a solid foundation on which to build.

About this tutorial

This tutorial takes a basic approach to securing a server running the GNU/Linux operating system. Together with its companion tutorial, "Hardening the Linux desktop," they introduce you to basic security concepts and take you through step-by-step examples of how to protect both the desktop and server environments and the confidentiality, integrity, and availability of the data they contain.

Objectives

In this tutorial, you learn about basic concepts in security administration, including how to secure Secure Shell (SSH) remote logins, create firewall rules, and watch logs for possible attacks.

Prerequisites

This tutorial is written for the beginning GNU/Linux user. You should have some familiarity with operating system installations and the command line. To fully

understand the concepts in this tutorial, you should have gone through the companion tutorial, "Hardening the Linux desktop."

System requirements

To run the examples in this tutorial, you need to install Ubuntu Server Edition on a computer or a virtual machine, such as Sun VirtualBox. You also need an Internet connection to download specific software packages used in the tutorial.

Section 2. Introduction

To understand the basics of hardening a server running GNU/Linux as the operating system, you need to be aware that although many core *concepts* of security apply to both the desktop operating system and the server operating system, the *ways* they're secured are completely different.

The Principle of Least Privilege

A truly secure network makes sure that the Principle of Least Privilege is applied across the enterprise, not just to the servers. The roles taken on by servers and desktops also mandate how the operating system, and the computer itself, should be secured. The desktop may be an attractive target for a script kiddie whose attacks are often thwarted by updated software and malware scanners, but a data center hosting user accounts or credit-card information is a much more attractive target for the skilled attacker who can exploit weaknesses without detection in an environment that hasn't been hardened.

Securing a server is much different than securing a desktop computer for a variety of reasons. By default, a desktop operating system is installed to provide the user with an environment that can be run out of the box. Desktop operating systems are sold on the premise that they require minimal configuration and come loaded with as many applications as possible to get the user up and running. Conversely, a server's operating system should abide by the Principle of Least Privilege, which states that it should have only the services, software, and permissions necessary to perform the tasks it's responsible for.

Section 3. Revisiting the immutable laws of security

In November 2000, Scott Culp of Microsoft drafted what he called the *10 Immutable Laws of Security* (see Resources for a link). There are two versions of these laws:

one for users and one for system administrators. Over the years, these laws have been both revised and despised by people in the security industry. Despite some criticism, the 10 laws for administrators can serve as an excellent foundation for hardening any system if applied correctly.

First, the following law applies to general security practices: Security only works if the secure way also happens to be the easy way. This is the most important law for any system administrator. If a security policy is so tight that people can't perform their job tasks, they're going to find ways to circumvent the security put in place, sometimes creating a greater vulnerability than the policy was put in place to prevent. The best example relates to passwords. Strong passwords should be part of any security policy, but sometimes policies go too far. Requiring users to remember a password that is 15 characters long and that consists of uppercase letters, lowercase letters, numbers, and symbols is asking for a high percentage of users to write their password on a post-it note and attach it to their monitor.

Four of Culp's laws apply directly to the material covered in this tutorial:

- **If you don't keep up with security fixes, your network won't be yours for long.** Attackers find vulnerabilities every day. As a system administrator. you need to make sure your system is updated. But this brings you to a difference between hardening a desktop and hardening a server. Generally, updates to the GNU/Linux desktop should be installed when they're published. When you're dealing with the server, you should test it in a research or development server environment before applying the fix to your production server, to make sure the patch doesn't interfere with the operations of the server or the users.
- **Eternal vigilance is the price of security.** In an effort to make sure your GNU/Linux server is secured, you must constantly check logs, apply security patches, and follow up on alerts. Vigilance is what keeps your system secure.
- **Security isn't about risk avoidance; it's about risk management.** Things happen. There may be a malware outbreak, or your Web site may be attacked. It may be something completely out of your control, such as a natural disaster. At one time or another, the security of your system will be tested. Make sure you've done everything you can to protect your system, and deal with the threat in a way that keeps your server and its resources available to the users who count on it.
- **Technology isn't a panacea.** If there is one law that everyone who deals with technology should know, it's this one. Simply throwing more technology at the security problem won't solve it. Vigilance on the part of the system administrator, buy-in on the part of management, and acceptance on the part of users must all be in place for a security policy to work effectively.

Section 4. Plan the server installation

The first step in hardening a GNU/Linux server is determining the server's function. What you use your server for determines what services need to be installed on the server. For example, if the server in question is used as a Web server, you should install LAMP services. On the other hand, if the server is used for directory services, Linux Apache MySQL PHP/Perl/Python (LAMP) has no business being installed on this machine. The only applications and services that should be permitted to run on your server are those that are required for the task the server is meant to perform. Nothing extra should be installed, for two reasons:

- Installing extra software or running extra services means there is one more door you have to lock. For example, If you're running Lightweight Directory Access Protocol (LDAP) on a server for directory services, you need to make sure that both the operating system and LDAP are up to date with their security fixes and patches so that any known vulnerabilities are plugged. If LAMP were installed on this server, it would require updates and attention, even if it wasn't being used. Its mere existence on the server would provide an attacker another avenue into your system. Likewise, any other software installed on this server must be updated, patched, and monitored to make sure it doesn't provide a vulnerability that an attacker can exploit.
- Installing extra software on a server means someone will be tempted to use that server for something other than its intended use. Not only does using the server for other tasks take resources away from it performing its main task, it exposes the server to threats it would not likely see without the software installed on it.

Among other things, you must decide whether to install a graphical user interface. For years, GNU/Linux admins have held a certain pride in being able to completely administer their networks and servers from a command-line interface. But in recent years, some system administrators have begun administering their GNU/Linux servers through a GUI. The choice to install a GUI such as the X Window System has sparked debate on various forums. On one hand, defenders of the command-line interface bring up the fact that the GUI can tax a system's resources and, because it's an extra service that isn't necessary, provide attackers with additional vulnerabilities. This side also points out that commands can be entered quickly through the command line without the need to search through menus and folders when performing a task.

On the other side of the debate, those who support a GUI environment argue that the GUI process can be killed when no longer in use to save resources and prevent any vulnerabilities from being exploited. They also argue that the GUI makes certain tasks, such as working with a database, much easier for the administrator.

GUI login

Some people who rely on a GUI like Gnome or KDE may be inclined to install a graphical login such as GDM. This isn't necessary because you can log in from the command-line interface just as easily as you would through a GUI-based login screen. The only difference is that you have

to use the `sudo startx` command if you need to administer your server through a GUI.

Installing a GUI on your server is entirely a personal choice. Everything in this tutorial is done through the command line; but should you wish to install a GUI, the following instructions show you how to install Gnome as a desktop GUI:

1. Once logged into your system, you should be at the command prompt. To install the Gnome core, type the following: `sudo aptitude install x-window-system-core gnome-core`
2. Press **Enter**. You're asked for the sudo password. Type it, and then press **Enter** again. You're informed about what is being installed.
3. To continue with the installation, type `Y` and then press **Enter**. Doing so installs a scaled-down version of Gnome that keeps the features of the desktop environment to a minimum and saves system resources. To install the full-featured version of Gnome, enter `sudo aptitude install x-window-system-core gnome`
4. After you press **Enter**, you're asked to go through the same process as earlier. Follow along until Gnome is installed on your system.
5. When either package is finished installing, you're still at the command prompt. To open Gnome, type the following: `sudo startx`

Section 5. Securing SSH

SSH provides a user with a connection to a remote computer. As a replacement for Remote Shell (RSH) and Telnet, SSH is commonly used by system administrators to log in to their servers from a remote computer to perform maintenance and administrative tasks. Even though SSH provides a much greater level of security than the protocols that it replaced, you can do some things to make it more secure.

Security by obscurity

One of the most common methods for hardening SSH is to change the port number that is used to access it. The theory is that an attacker using the default port or TCP 22 to establish a connection will be denied access because the service is running on a secure port.

This method of securing SSH is the center of multiple forum debates. Changing the port number won't prevent the SSH port from being found by an attacker with a port scanner who takes the time to scan all of the ports on your server; and for this reason, many system administrators don't bother changing the port. But this approach does prevent script kiddies from attacking SSH with automated tools dedicated to finding open TCP 22 ports, and impatient attackers may grow weary

of scanning your server if they don't find SSH running in the first range of ports they scan.

To change the SSH port address, you need to first install SSH on your server. Type

```
sudo aptitude install openssh-server
```

Press **Enter** and type your password. This command installs openssh to use for remote logins to your server.

Once you have an SSH file to configure, you should copy the file in case something happens when configuring. You can always revert back to the original. Follow these steps:

1. At the command line, type `sudo cp /etc/ssh/sshd_config /ete/ssh/sshd_config.back`
2. Press **Enter** and provide your password to complete the backup of this file.

Install emacs

To install emacs, use `sudo aptitude install emacs` Now, you need to locate the portion of the file where the port number is set. Once you've found this (the default is port 22), you can change it to an arbitrary number. There are more than 65,000 ports; choose something at the upper end of the scale, but a number you'll remember. Remember, skilled attackers know how people think. Changing the port number to 22222 or 22022 is a common mistake—choose a number that isn't easily guessed.

Now, you need to change the permissions for the sshd_config file so you can change it:

1. Type `sudo chmod 644 /etc/ssh/sshd_config`
2. Press **Enter**. Now you can use a text editor like emacs or vi to change the file: `emacs /etc/ssh/sshd_config`

Leave emacs or vi open as you make more changes to this file.

Root login permissions

The root user in all Ubuntu distributions is disabled, but you can activate this account. If you're using SSH, you should deny the root account permission to log in to the server remotely in the event that you or an attacker has activated this account. While you have the editor open, scroll down to the line that reads `PermitRootLogin`. The default is yes.

Whitelist users

Another step you can take to harden SSH on your server is to allow only certain users to use this service. This process is known as *whitelisting*. To create a whitelist,

you first need the usernames of the people who will be allowed to use SSH to remotely access the server. Then, follow these steps

1. Add this line to your sshd_config file:
   ```
   # Allow only certain users
   AllowUsers username username username
   ```
   Substitute usernames from your list in place of the word *username*. Alternately, you can allow groups access to SSH logins by using `# Allow only certain groups AllowGroups group group` Again, substitute your user groups for the word *group* in the example.
2. Save your configuration file, and exit your editor. You need to restart SSH in order for the changes to take effect. You don't need to shut down your computer -- just type `sudo service ssh restart`
3. Press **Enter** and provide your password. The service restarts and tells you `[OK]`.

There are many other ways to further secure SSH that are for more advanced users. When you've had more experience working with GNU/Linux and SSH, you should consider taking these steps.

Section 6. Write firewall rules

You can deny access to your server through your firewall. Ubuntu Server uses a firewall called Uncomplicated FireWall (UFW), which is actually a management tool for iptables. Iptables filters network packets based on a series of rules written by the system administrator. Iptables can be complicated for beginners, so UFW simplifies it. Using UFW can help you harden your server; but if you're truly interested in server security, learning how to write rules for iptables will let you fine-tune a server's security.

To get started with UFW, you need to install it. Follow these steps:

1. From the command line, type `sudo aptitude install ufw`
2. Press **Enter** and enter your password. Press **Enter** again to install the package.
3. To enable the firewall, type the following: `sudo ufw enable`
4. Press **Enter**. You see the message `Firewall started and enabled on system startup`. Now you can create rules for your firewall.

Remember how you changed the port for SSH earlier? To open the port through UFW by creating a rule, type the following at the command line:

```
sudo ufw allow 65000
```

That command allows access over port 65000 and lets SSH traffic into your server.

To deny access over this port, use the following:

```
sudo ufw deny 65000
```

To allow or deny traffic specifically on TCP port 65000, use the following command:

```
sudo ufw allow 65000/tcp
```

You can also allow or deny traffic according by the protocol it uses. For instance, to block all HTTP traffic, you can use this command:

```
sudo ufw deny http
```

You can create more complicated rules to deny or allow a service based on its IP address. For instance, if your desktop had the IP address 192.168.1.30 and your server had an IP address of 192.168.1.5, you could allow only your computer's IP address the ability to establish an SSH connection:

```
sudo ufw allow proto tcp from 192.168.1.30 to 192.1681.5 port 65000
```

To check which rules you're currently running with UFW, use

```
sudo ufw status
```

You're presented with a list of rules you've written for your firewall. If you see a rule that you wish to delete, type

```
sudo delete [rule]
```

Section 7. Monitor your system

There is a saying in computer security circles that the only way to truly secure a computer is to completely disconnect it and lock it in a box. Not too practical, but the underlying message is that if an attacker really wants into a system, odds are they will find a way in. After you take steps toward intrusion prevention, you need to set up a monitoring system to detect whether an attack against your server has taken place. Then, if you're alerted to an attack, you're better prepared to deal with it early on.

The following sections walk you through the steps of installing and configuring two programs that help to detect intrusions. Tripwire alerts you to unauthorized activity that takes place with system files on your server, and Logwatch is a tool that can be used to create reports for you to analyze.

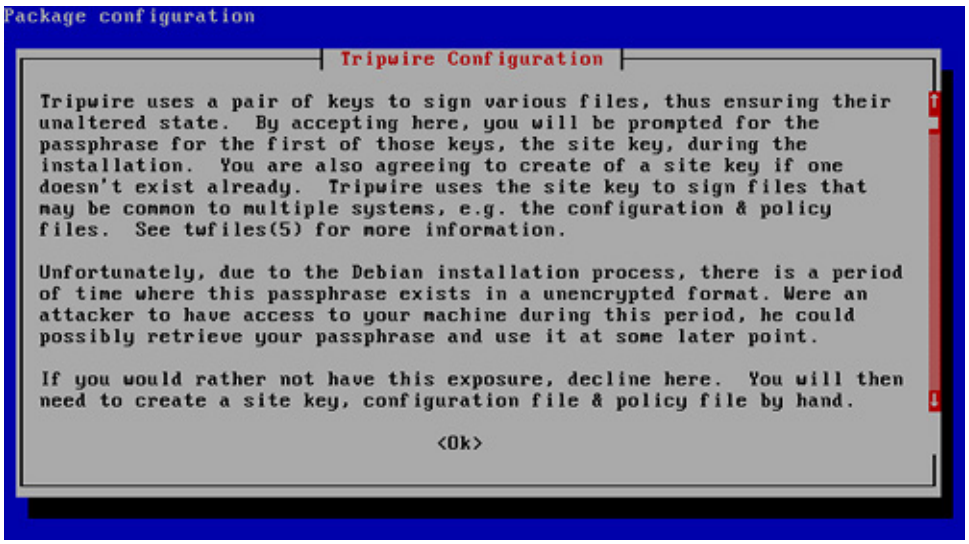
Tripwire

Tripwire is a program that sets up a baseline of normal system binaries for your computer. It then reports any anomalies against this baseline through an e-mail alert or through a log. Essentially, if the system binaries have changed, you'll know about it. If a legitimate installation causes such a change, no problem. But if the binaries are altered as a result of a Trojan horse or rootkit being installed, you have a starting point from which to research the attack and fix the problems.

To install and configure Tripwire through the command line, follow these steps:

1. Enter the following command:
   ```
   sudo aptitude install tripwire
   ```
2. Press **Enter** and type your password, and Tripwire downloads and installs.
3. You're presented with the configuration screen shown in Figure 1.

   Figure 1 Tripwire Configuration.

   This screen informs users of Debian-based systems about a potential scenario in which an attacker could obtain the passphrase used with Tripwire while it's unencrypted. Advanced users may opt to stop here and create the site key and configuration files on their own. Beginners should select **OK** and move forward.
4. The next screen asks if you wish to create your passphrase during installation. Select **Yes**, and press **Enter**.
5. The next screen informs you about how Tripwire works. The program creates a text file that stores an encrypted database of the systems configuration. This text file is the baseline. If any changes are made to the system configuration, Tripwire sees the change and creates an alert. In order for you to make legitimate changes to the system, you create a passphrase. Select **Yes** and press **Enter** to begin building the configuration file.
6. The following screen explains the same thing, but this time you're building the Tripwire policy file. Again, select **Yes** and press **Enter**.

7. Once the files are built, you're prompted to enter the site-key passphrase. You need to remember this passphrase. Select **OK**, and then press **Enter**. You're prompted to enter your passphrase again on the next screen.
8. You're brought to the local passphrase screen. This passphrase is required for the local files on the server. Enter your local passphrase, select **OK**, and then press **Enter**. You need to re-enter this passphrase again as well.
9. Now that Tripwire has been installed, you're told the location of the database and the binaries. With **OK** selected, press **Enter** again to complete the configuration process.

You can run the database initialization by typing

```
sudo tripwire --init
```

Press **Enter**. You're asked to provide the local passphrase you created during the Tripwire installation. Provide the passphrase and again press **Enter**. Now, Tripwire has created the baseline snapshot of your file system. This baseline will be used to check for changes to critical files. If such a change is detected, an alert will be sent.

You can check run an integrity check at any time by following these steps:

1. Type
   ```
   sudo tripwire --check
   ```
2. Press **Enter**. You're provided with a report that is saved in the reports directory. To view this report, use the `twprint` command:
   ```
   sudo twprint --print-report -r\
   ```
3. Press **Enter**, and type the `sudo` password. You're given a different type of prompt that looks like this:
   ```
   >
   ```

   At this prompt, type the location and filename of the report you wish to print:
   ```
   > /var/lib/tripwire/report/<server name>-YYYYMMDD-HHMMSS.twr| less
   ```

   If you don't know the exact time you ran your report, navigate to the directory /var/lib/tripwire/reports to see the complete filename.

As your skills advance, you can look to twadmin to further fine-tune the capabilities of Tripwire. You can also set a cron job to e-mail you a copy of this report each day or configure Tripwire to e-mail you if an anomaly is reported.

Logwatch

Logwatch is a great tool for monitoring your system's log files. This program requires a working mail server on your network to e-mail the logs to you. If you wish to change the .conf file, you need to open /usr/share/logwatch/default.conf/logwatch.conf and look for the line that reads `MailTo`. Change user.name.domain.tld to your e-mail address.

You can install Logwatch with this command:

```
sudo aptitude install logwatch
```

To e-mail the logs to yourself, type

```
logwatch --mailto email@youraddress.com --range All
```

Pressing **Enter** sends a copy of the report to the e-mail address specified. If you aren't running a mail server on your network but would still like to see a Logwatch report, the following command provides it on your screen:

```
logwatch --range All --archives --detail Med
```

The output spans several screens; press **Shift-Page Up** to move to the beginning of the report.

Section 8. Users and groups

GNU/Linux handles groups and permissions differently than the Microsoft® Windows® operating system does. You can organize users into groups for easy administration, but you also need to provide access to files and folders through permissions. No blanket "power user" gives users access to almost everything on a computer or network. The GNU/Linux system was designed to be more secure; it works off a *3x3* system for granting permissions:

Don't run as root

Anyone who knows anything about GNU/Linux security will tell you never, never, never to run anything as the root user. Logging in as administrator in a Windows network is common, but doing so is discouraged in the GNU/Linux community. This is why whenever you need to run something as root, you use the sudo command. Any system administrator can use the sudo command if you give them the password. To see how and when the sudo password is being used, check out /var/log/messages. Because you're looking for all uses of sudo, use the `grep` command to find them.

- **File permissions** -- Read (r), write (w), and execute (x). Each of these permissions is also given a number: read = 4, write = 2, and execute = 1.
- **Directory-level permissions** -- Enter, which gives permission to enter the directory; show, which gives permission to see the contents of the directory; and write, which gives permission to create a new file or subdirectory.
- **How permissions are assigned** -- Permissions are assigned in three ways: by user level, group level, and other level. The user level defines the user who created the file or directory, the group level defines the group the user is in, and the other level is for any user outside of the user's group.

The user permissions are granted first: for example, r/w/x means the user can read, write, and execute the file or files in the folder. You can apply the number value to each permission. Thus if a user can read, write, and execute, you add the corresponding numbers 4, 2, and 1, for a total of 7. Next come the group permissions. For instance, the other members of the user's group may be able to read and execute, but not write. Adding up the corresponding values gives you 5. Those in the others category can only read the files, so their numerical value is 4. Thus, the permissions for the file or folder are 754.

When permissions are set to 777, everyone is given the ability to read write and execute. The `chmod` command changes permissions for files and directories. If you wish to change ownership of a user, use the `chown` command. To change group ownership of a file or directory, use the `chgrp` command.

Section 9. Encryption

Encryption is the process of taking data stored on a computer and scrambling it in a manner that makes it unreadable to anyone who doesn't possess the key to re-create the data in its original form. Data that has been encrypted can be stored on the local computer, stored on a network share, or transmitted to other users and computers.

It's possible to encrypt an entire hard disk or the partitions of the disk. This should be done at installation. You can also secure data through encryption by creating a directory and encrypting it. For example, if you've set up a file server, you may want to encrypt a directory that holds sensitive information.

Before you go forward with protecting your data, you need to install eCryptfs from the Ubuntu repositories by typing

```
sudo aptitude install ecryptfs-utils
```

Press **Enter**, and type your root password.

Encrypt a directory

The next step is to create a directory to encrypt. The example uses a directory called secure, but you can name it anything you wish. Follow these steps:

1. Enter the following command:
   ```
   mkdir ~/secure
   ```
2. Just to keep others from snooping around, change the permissions to 700:
   ```
   chmod 700 ~/secure
   ```
3. Mount the new directory with the eCryptfs file system:
   ```
   sudo mount -t ecryptfs  ~/secure ~/secure
   ```

4. You're asked a series of questions. Be sure you remember the answers, because you'll need them when you remount. The first question asks which type of key you'd like to use. Make your selection by typing the number that corresponds to your choice. Next, select the cipher you wish to use and the size of the key.
5. Once you've answered all the questions, your directory is ready to add files and other subdirectories to. When you're ready to secure your directory, unmount it with

```
sudo unmount ~/secure
```

Section 10. Additional security steps

Now that you've created a solid foundation for hardening your server, you should take a few steps to further enhance the security measures you've put into place. These last few tips introduce some of the extra points to keep in mind when hardening your GNU/Linux server.

Updates

A production server should never have updates and patches installed unless they were first tested on a test, or development, server. Because a GUI may not be installed on your server, you have to download any updates and patches through the terminal. When you're ready to install updates, enter the command `sudo apt-get update` and then `sudo apt-get dist-upgrade`. In some cases, you need to restart your server.

Malware

Many system administrators find installing antivirus software on a server running GNU/Linux to be a waste of resources because no viruses in the wild can attack the GNU/Linux operating system. But any GNU/Linux administrator who is running SAMBA to share Windows files should definitely make sure an antivirus scanner like ClamAV is installed to make sure infected files don't spread throughout your system.

Although viruses don't pose as much of a threat to the GNU/Linux server, rootkits can cause you a headache. *Rootkits* are tools that attackers use to gain root-level permissions to a system, capture passwords, intercept traffic, and create other vulnerabilities. To combat this threat, you should install tools such as RKHunter and chkrootkit on the server (see instructions in the "Hardening the Linux desktop" tutorial).

Backup and recovery

Servers that house gigabytes of information, corporate Web sites, or catalogs for directory services need to have a backup and recovery strategy in place. Most

corporate networks can afford redundancy through multiple servers, and smaller networks can find peace of mind through virtualization and back-up and recovery software.

If you're planning to run backup and recovery software from the Ubuntu repositories, Sbackup is an excellent choice because it can be run from either the command line or a GUI. When backing up server data on a corporate network, it's important that your backup files be stored outside the server. Portable storage devices provide large amounts of storage space at extremely reasonable prices, and they're excellent options for storing backed-up files and directories.

Passwords

As the system administrator, you're required to set passwords for your server's root account and possibly other sensitive accounts in your organization such as MySQL databases or FTP connections. You can't force strong passwords for your users with Ubuntu Server, but you can be sure you train users on how to create a strong password.

Network password policy

If you're running directory services like OpenLDAP, you have the option to enforce strong passwords across your network with some of the configuration options available.

Make sure your users' passwords contain at least three of the following: an uppercase letter, a lowercase letter, a number, or a symbol. To further strengthen the password, make it a policy that all passwords are at least eight characters long.

One way to teach users to use strong passwords but keep them from writing down complex passwords on sticky notes is to have them use passphrases. Something like Myf@voritecolorisBlue! is much easier to remember than M$iuR78$, and both meet minimal complexity standards.

Section 11. Conclusion

Having completed this tutorial and "Hardening the Linux desktop," you should have a solid knowledge base on the topic of system security. Keep in mind that these tutorials are aimed at beginners, to provide a foundation for learning more about GNU/Linux security.

Resources

Learn

- Also by Jeffrey Orloff, "Hardening the Linux desktop" (developerWorks, November 2008) is a step-by-step guide to securing a GNU/Linux desktop computer.
- Scott Culp's 10 Immutable Laws of Security boils down the important facts of security for users, and his follow-on article, 10 Immutable Laws of Security Administration gives similar guidance for administrators.
- In the developerWorks Linux zone, find more resources for Linux developers (including developers who are new to Linux), and scan our most popular articles and tutorials.
- See all Linux tips and Linux tutorials on developerWorks.
- Stay current with developerWorks technical events and Webcasts.

Get products and technologies

- Download Ubuntu Server Edition to follow along with the lessons in this tutorial.
- Download Sun VirtualBox to create a virtual machine so that you can practice with the lessons in this tutorial.
- With IBM trial software, available for download directly from developerWorks, build your next development project on Linux.

Discuss

- Get involved in the developerWorks community through blogs, forums, podcasts, and spaces.

About the author

Jeffrey Orloff

Jeffrey Orloff serves as the Director of IT and Security for SafeWave, LLC. He also works as the technology coordinator for the School District of Palm Beach County's Department of Alternative Education/DJJ.