DIGITAL SECURITY

Linux Security:

Access Control Mechanisms

This material is based on work supported by the National Science Foundation under Grant No. 0802551

National Science Foundation
WHERE DISCOVERIES BEGIN

Lesson Overview

When working for an institution as a Linux Administrator, you may be required to protect certain information based on its sensitivity. For instance, most organizations will have an internal network in which data contained in certain directories or drives are made public—employees can access the contents.

However, certain kinds of information such as employee salaries, classified research, secret prototypes, health information, military secrets, and private communications are considered highly sensitive and are usually restricted from everyone except very few people authorized to access the data.

In this lesson, we will explore processes, tools, and control lists that make it possible to limit access to sensitive data on a Linux-based system. Understanding this topic is important for any system administrator configuring systems in the workplace that require access control mechanisms.

Objective

You should know what will be expected of you when you complete this lesson. These expectations are presented as objectives. Objectives are short statements of expectations that tell you what you must be able to do, perform, learn, or adjust after reviewing the lesson.

Lesson Objective:

Given the need to secure a Linux server, the student will recommend a set of standard Linux tools such as PAM, Access Control Lists, and TCP Wrappers to effectively secure a Linux system and demonstrate the use of one set of tools for system lock-down.

Lesson Outline

In this lesson, you will explore:

- Access Control Mechanisms
- Discretionary versus Mandatory Control Mechanisms
- Access Control Lists (ACLs)
- Pluggable Authentication Modules (PAM)
- Bastille Linux
- TCP wrappers
- Shadow passwords

Resources and Notes

This lesson uses Ubuntu Linux for demonstration. To complete this lesson successfully, you must have access to:

- Ubuntu Linux on bare metal or as a virtual install
- 10 Gb of hard drive space dedicated to the operating system's use
- A command shell
- Internet for research
- Word processor

Use the resources on the right to configure your system for Ubuntu.

Resources:

- Download Virtualbox
- Using Virtualbox with Ubuntu
- Virtualbox for Linux Hosts
- Virtualbox manual

Access Control Mechanisms

The *Trusted Computer System Evaluation Criteria* or "Orange Book" from the Department of Defense (DOD) requires the following:

Computer systems of interest must enforce a mandatory security policy that can effectively implement access rules for handling sensitive (e.g., classified) information. These rules include requirements such as:

- No person lacking proper personnel security clearance shall obtain access to classified information.
- Discretionary security controls are required to ensure that only selected users or groups of users may obtain access to data (e.g., based on a need-to-know).

As a Linux administrator, data you will be handling may or may not be sensitive (i.e. classified). However, you will still need to implement some access control mechanisms regardless of where you work (government or private industry) to ensure data is available to the right users and not available to others.

Recommended Review

- Trusted Computer System Evaluation Criteria

Access Control Mechanisms

Access Control Mechanisms are either discretionary or mandatory:

Mandatory access controls is defined on page 109 of the Orange Book as:

. . . a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity . . .

On the other hand, discretionary control is defined on page 61 as:

. . . discretionary controls give individuals discretion to decide on which of the permissible accesses will actually be allowed to which users, consistent with overriding mandatory policy restrictions . . .

In Unix/Linux systems, the use of discretionary control mechanisms is considered their traditional/basic control mechanism, but Linux systems also have the capability of providing access control through the use of the Linux Security Module (LSM).

Recommended Review

- Security & SELinux
- DOD Standard

Discretionary & Mandatory Controls

Discretionary Control Mechanisms

Prior to Linux Kernel 2.6, discretionary access control (DAC) was the only security model for Linux. Using this model, security is implemented based on user identity and ownership privileges. A process always inherits the rights or credentials of its parent.

Mandatory Access Control Mechanisms

It was during the 2001 Linux 2.5 Kernel Summit that Peter Loscocco (National Security Agency) presented the mandatory access control (MAC) system design in its SE Linux distribution.

The normal security checks are still performed by Linux, but the Kernel will also make an inquiry to the security model to determine if access should be granted. The MAC design also allows the Linux administrator to choose which model he/she prefers to implement.

View summit picture on next screen

Recommended Reading

- Assessment of ACM

Discretionary & Mandatory Controls

The Linux 2.5 kernel summit

The "Linux 2.5 Kernel Summit" is a two-day affair, held in San Jose, California; the organize
particular. The purpose of this event was to get the core kernel hackers together in one place t
imminent 2.5 development series. The attendee list shows 65 hackers, almost all of whom hav
kernel development. It's an impressive gathering; more than one attendee has been heard to fr

Image source:
http://lwn.net/2001/features/KernelSummit/

Access Control Tools:

ACL, PAM, Bastille, TCP Wrappers, & Shadow Passwords

Access Control Lists

Access Control Lists allow flexibility when applying Unix / Linux file permissions. As you have been instructed in previous lessons, you can implement and/or grant certain permissions to any user and/or group.

The following commands will allow you to manage permissions:

To enable ACL:

- Edit the /etc/fstab file
- Add the ad attribute as an option on the selected partition you want to use ACLs
- Save and remount the partition

You will find the following commands helpful:

- *getfacl* to retrieve ACL information
- *setfacl* to change ACL settings
- *setfacl -m* to add permissions
- *setfacl -b -* to remove all the current ACLs from a file or file system

The next four screens explore the ACL process visually . . .

Required Reading

- Unix Permissions
- Access Control Lists
- Getfacl
- Setfacl

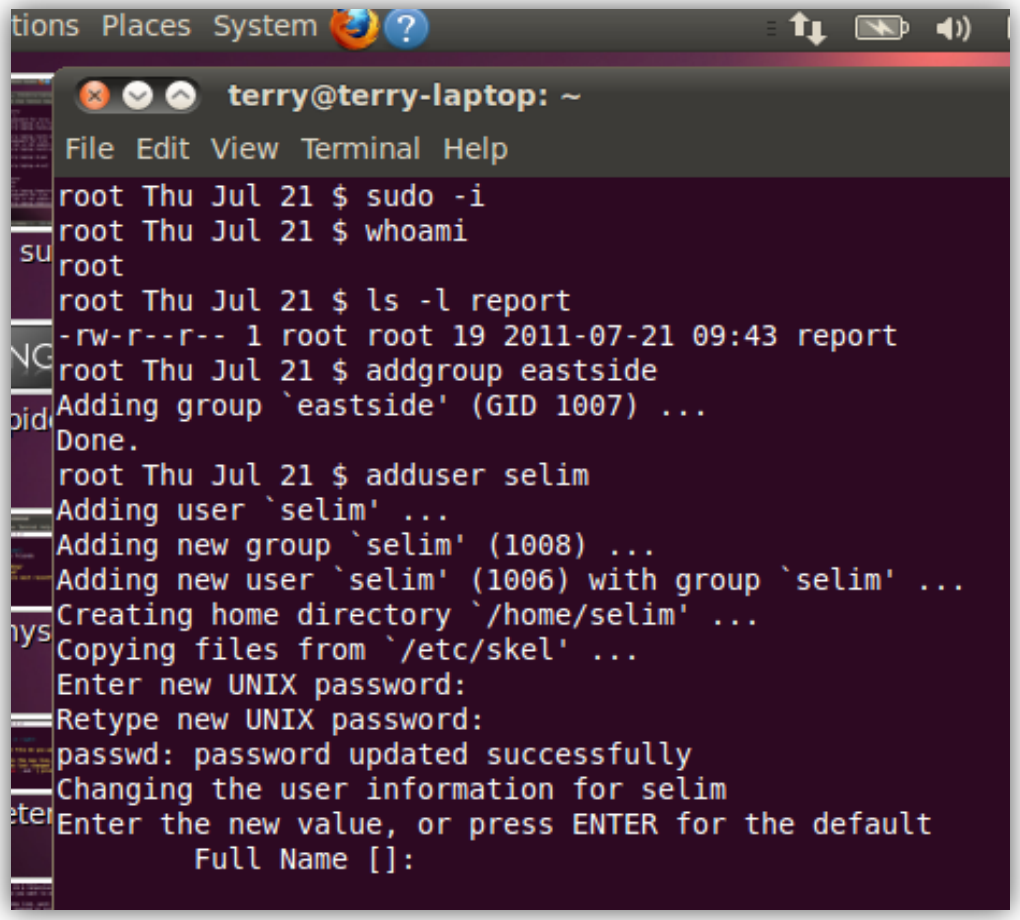
Access Control Lists (Contd)

Explaining the changing ACL entries for a user named Selim who is a member of the Eastside group for a file titled *Report*:

Step 1 - Log on as root.

If they (ACL entries) do not already exist, add (create):

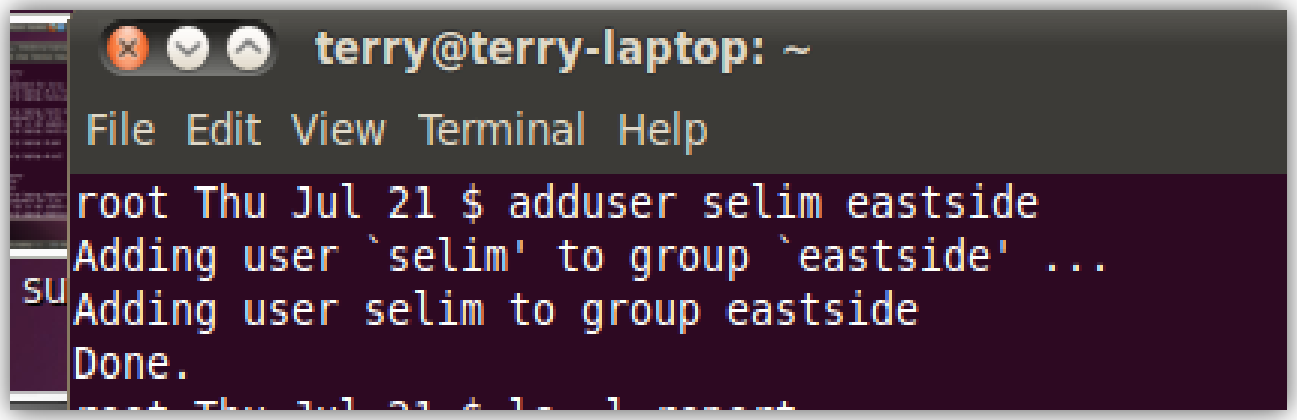
a. the file to be used (report)
b. the user (Selim)
c. the group (eastside)



Access Control Lists (Contd)

Step 2

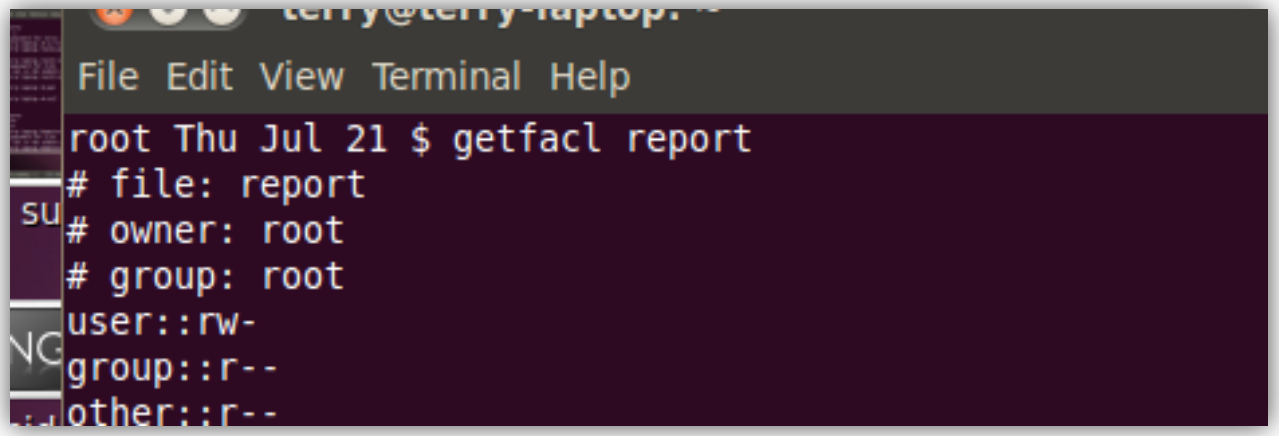
Make sure the user (Selim) is part of the group (eastside)



Access Control Lists (Contd)

Step 3

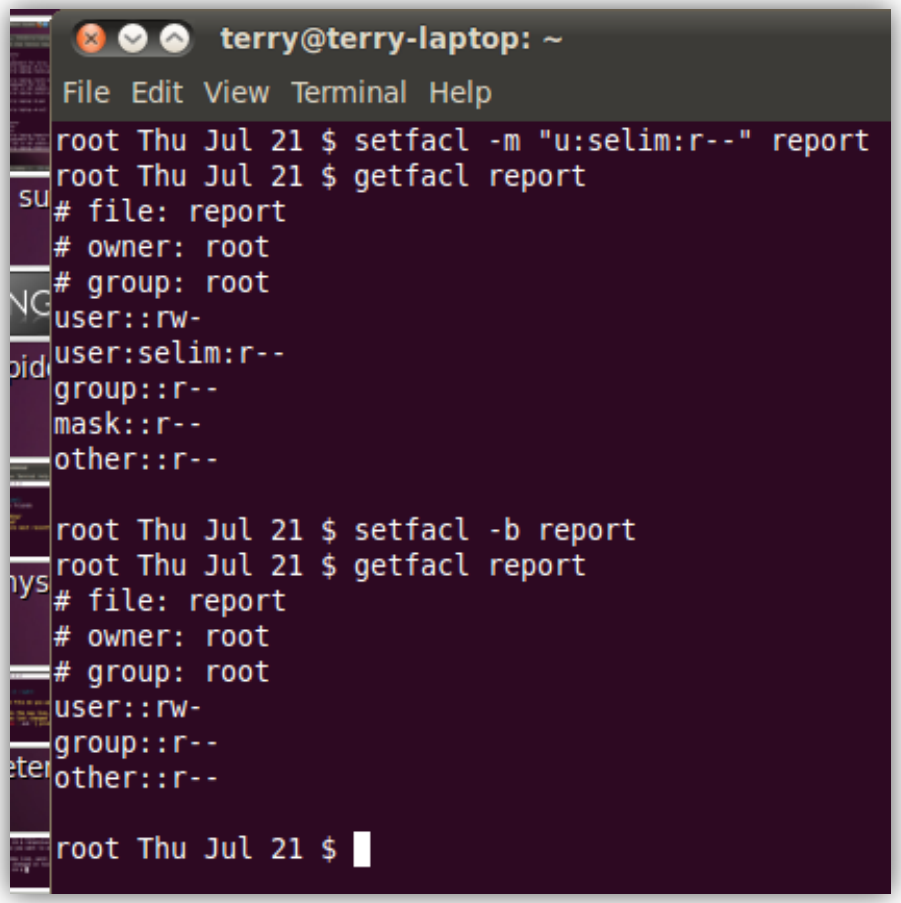
Retrieve the ACL information using the **getfacl** command for the file report.



Access Control Lists (Contd)

Step 4

a. Add permissions using the command setfacl (user Selim for file report)
b. Notice the changes for the file report: the user Selim and the new permission rights using the command *getfacl*.
c. To delete the ACL entries for report use the **setfacl -b** command.
d. You view the changes (again) by using the *getfacl* command.



Pluggable Authentication Modules (PAM)

Years ago, users would traditionally use their password as a means of authentication. However, other methods are now in place that allow a user to be authenticated when accessing resources, devices, or filesystems.

These new "methods" require authentication modules that work with traditional *etc/passwd* file. This is where Pluggable Authentication Modules (PAM), developed by SUN Microsystems, come in handy.

The authentication task of applications is handled by a set of libraries (PAM). This makes it possible for the development of programs that are independent of specific authentication rules. In other words, PAM simplifies the process by granting a standard, flexible interface for the authentication to take place. It is up to the administrator to decide which module he/she would like to implement.

Recommended Reading

- Authentication Howto
- Example PAM Modules
- Linux-PAM Guide
- Linux-PAM Writers' Guide
- Linux-PAM Bug Tracker
- Source for Linux-PAM

Bastille Linux

Bastille Linux is a software tool that allows Unix/Linux administrators to enhance the security of their systems. The danger with Bastille is that an administrator may use it to implement so many restrictions that it makes the system practically useless. Bastille Linux is not for the beginner user, but for the intermediate and/or advanced Unix/Linux user. As a Linux administrator, you need to be very careful using Bastille Linux.

Bastille Linux comprises a set of Perltk scripts that run as an interactive program, asking questions for each step of the "hardening" process. Some of the steps executed by Bastille include the application of a firewall, system patches, performing a SUID-root audit, and/or the deactivation of unnecessary services. During these steps, the administrator is allowed to understand the security measures to be implemented. Bastille not only provides the option for the administrator to choose which features to implement, but it provides a short explanation and suggestion as to whether or not you should implement the specific feature. The administrator also has the option to install this remotely, if needed.

Some of the distributions on which an administrator may implement Bastille include (but are not limited to): Red Hat, Fedora, SUSE, Debian, Ubuntu, Gentoo, Mandriva, HP-UX, and Mac OS X.

Recommended Reading

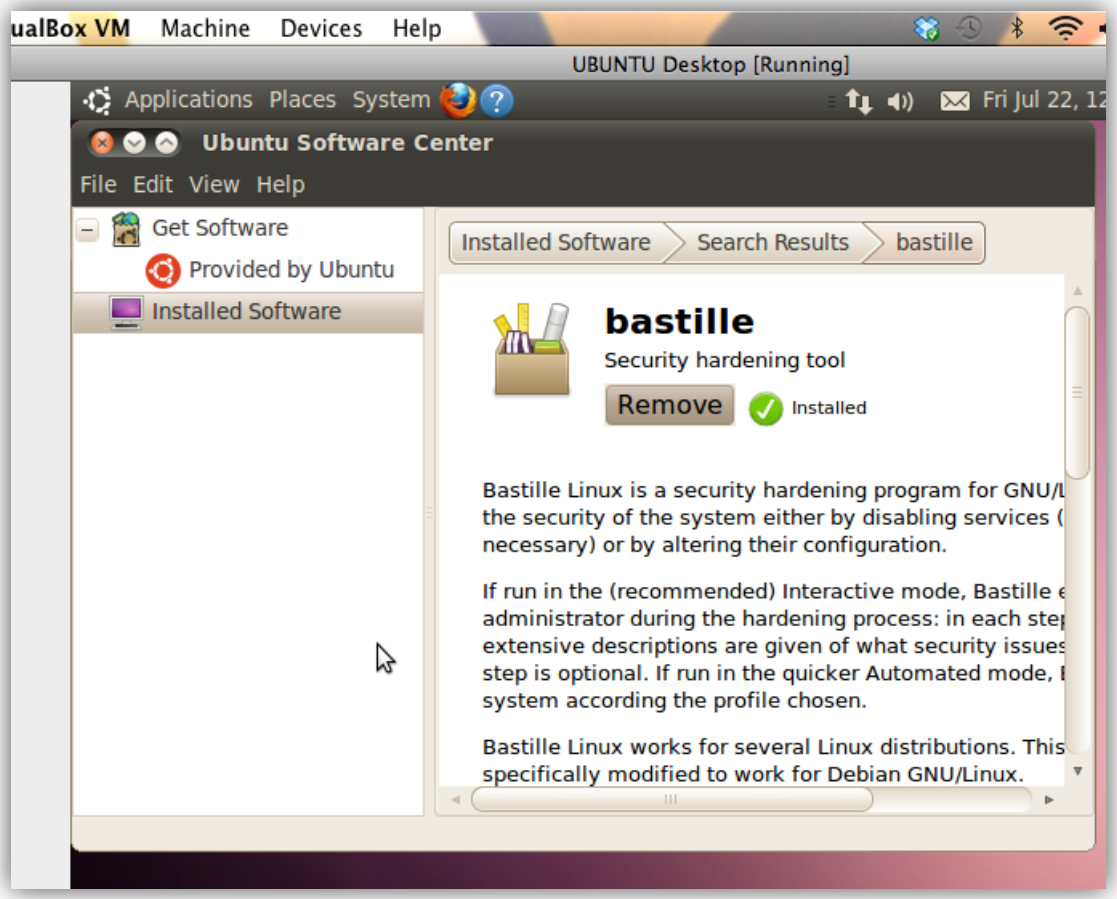
- Peltk
- Bastille Linux Home
- Bastille Linux (Ubuntu)
- Bastille Linux Download
- Bastille Linux Tutorial
- Bastille & CentOS

Installing Bastille Linux

Installing Bastille Unix (formerly known as Bastille Linux)

Step 1 – Log on to Linux as root.

Step 2 – Go to the Ubuntu Software Center and search for Bastille. You may install it from there.



Recommended Reading

- Bastille Linux
- Software Center

Installing Bastille Linux

Installing Bastille Unix

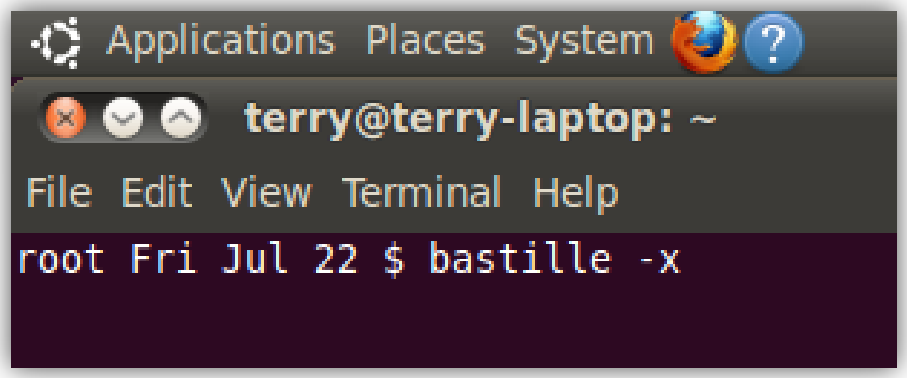
Step 3 - To start Bastille using an interactive GUI interface, you can type:

bastille -x

at the command line (remember, you need to be logged on as root).

Note: You may be prompted to install *perltk* for some distributions. If needed, you would type:

apt-get install perltk



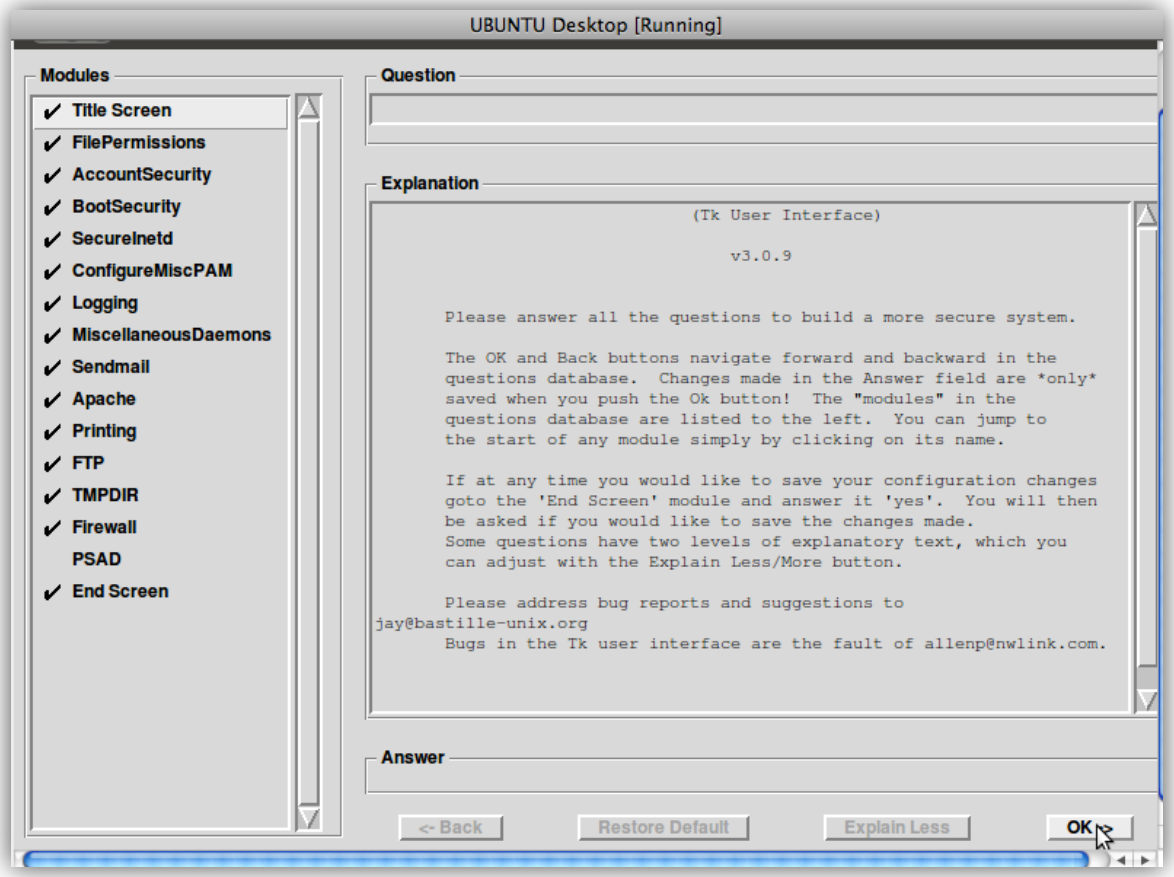
Installing Bastille Linux

Installing Bastille Unix

Step 4

The GUI based interactive Bastille screen will open.

Notice how the tool will allow the administrator to secure the system through the use of a question and answer format.



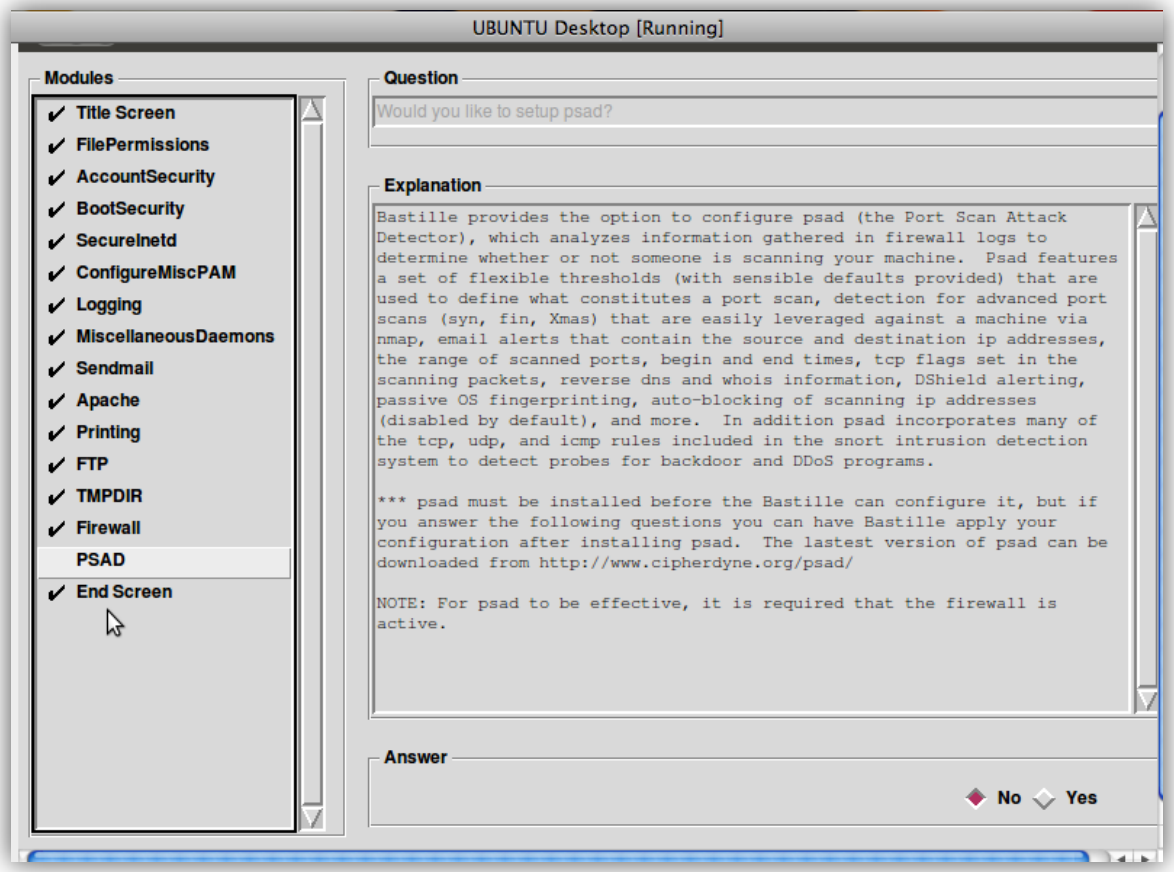
Installing Bastille Linux

Installing Bastille Unix

Step 5

Read carefully! You have the option to select the needed module by clicking on it (on the left of the screen) and a question and explanation will appear at the right side of the screen.

Select "yes" or "no" according to the selection:



TCP Wrappers

TCP Wrappers is a program providing firewall services to networked UNIX/Linux users by monitoring incoming packets. TCP Wrappers will determine if access is authorized by granting or denying access to either other nodes in the network or nodes that are outside the network.

To determine whether to grant or deny access to a network, TCP Wrappers use Access List Rules that are included in the /etc/hosts.allow or /etc/hosts.deny files. If the access list does not find a rule to accept the connection, then it checks the deny file, to see if the request needs to be denied.

TCP Wrappers are capable of monitoring and filtering incoming requests for several network services including:

- SYSTAT
- FINGER
- FTP
- TELNET
- RLOGIN
- RSH
- EXEC
- TFTP
- TALK

Recommended Reading

- Intro to TCP Wrappers
- TCP Wrappers on Linux
- TCP Wrappers Mac OS X
- TCP Wrappers Source

Shadow Passwords

Passwords are used as a means to authenticate a user to a system, node, or network. In a Unix/Linux environment, passwords were not shadowed (unshadowed) before ~1990's. This meant that even though the password information could only be changed by the super user, it could be easily read by an intruder (hacker). However, through the implementation of shadow passwords intruders are unable to view passwords during a "brute force attack."

The etc/passwd file contains the following user password information: username, encrypted password (consisting of 13 characters - the first two characters are the "salt" and the remaining characters are the hash), password expiration information, UID, GID, full name, home directory path, and login shell.

For shadow passwords, the /etc/shadow file usually contains the following user information: User login name, salt, number of days since last password change, number of days until change allowed, number of days before change required, number of days warning for expiration, number of days before account inactive, number of days since account expires.

Most newer Unix-like operating systems use shadow passwords.

Recommended Reading

- Why Shadow Passwords
- /etc/passwd file format
- /etc/shadow file
- Linux Shadow Password

Example 1: Shadow Passwords

Example of an entry using a shadowed password in the "/etc/passwd" file for a user Mike Dillon:

dillonm:x:691:691:Mike Dillon:/home/dillonm:/bin/bash
(a) (b) (c) (d) (e) (f) (g)
(a) dillonm - username
(b) :x: - shadowed password
(c)691: - UID
(d)691: - GID
(e)Mike Dillon: - Full name
(f)/home/dillonm: - home directory
(g)/bin/bash - shell

Example 2: Shadow Passwords

Example of an entry in the" / etc / shadow" file which contains password and account expiration information for a user by the name of P Valdez:

valdezp:Fp5mckrOMBhF.:10063:0:99999:5:::
(a) (b) (c) (d) (e) (f)(g,h)

(a) valdezp - username
(b) Fp5mckrOMBhF. - this represents the password which is 13 characters and encrypted. If it is blank (::), it means a password is not needed to login. If there is an asterisk, it means the password has been disabled (:*:)
(c) 10063 - number of days (since January 1, 1970) since the password was last changed
(d) 0 - number of days since the password has been changed. A zero means the password can be changed anytime.
(e) 99999 - number of days left before a password must be changed
(f) 5 - number of days to warn user of a password that is about to expire (7 = one week)
(g) :: - number of days (since January 1, 1970) that an account has been disabled
(h) :: - reserved field for future use

Lesson Summary

In this lesson, we discussed various ways of controlling which users have access to certain files and processes on a computer network with sensitive data.

We identified and explored five access control tools or mechanisms useful for Linux systems to control user access. These five tools include:

1. Access Control Lists (ACL)
2. Pluggable Authentication Modules (PAM)
3. Bastille Linux
4. TCP Wrappers
5. Shadow Wrappers

We also examined the differences between discretionary access control (DAC) and mandatory access control (MAC).

Use the *recommended* and *required reading* resources provided throughout this lesson to learn more about each access control tool discussed and consult your instructor if you encounter difficulties.

Additional Resources

- Mandatory Access Control
- Discretionary Control