YouTestMe

YTM Deployment Appliance Security Hardening

youtestme

Table of Contents

1 Introduction 4

2 Linux Server Hardening Checklist 4

2.1 Encrypt Data Communication 4

2.2 Avoid Using FTP, Telnet, and Rlogin/Rsh Services 4

2.3 Minimize Software to Minimize Vulnerability 4

2.4 One Network Service Per System or VM Instance 5

2.5 Keep Linux Kernel and Software Up to Date 5

2.6 Turn on SELinux 5

2.7 User Accounts and Strong Password Policy 5

2.8 Password Aging 6

2.9 Restricting Use of Previous Passwords 6

2.10 Make Sure No Non-Root Accounts Have UID Set To 0 6

2.11 Physical Server Security 7

2.12 Disable Unwanted Services 8

2.13 Find Listening Network Ports 8

2.14 Configure Iptables and TCPWrappers 8

2.15 Linux Kernel Hardening 8

2.16 Separate Disk Partitions 8

2.17 Disk Quotas 9

2.18 Turn Off IPv6 9

2.19 Disable Unwanted SUID and SGID Binaries 9

2.20 World-Writable Files 9

2.21 Files Without Ownership 10

2.22 Use A Centralized Authentication Service 10

2.23 Kerberos 10

2.24 Logging and Auditing 10

2.25 System Accounting with auditd 11

2.26 Install and Use Intrusion Detection System 11

2.27 Disable USB/firewire/thunderbolt devices 11

2.28 Backups 11

3 Hardening SSH 12
3.1 Use SSH public key based login 12
3.2 Disable root user login 12
3.3 Disable password-based login where practical 12
3.4 Limit Users' SSH access 12
3.5 Disable Empty Passwords 12
3.6 Use strong passwords and passphrase for users/keys 13
3.7 Firewall SSH TCP port # 22 13
3.8 Change SSH port and limit IP binding 13
3.9 Use TCP wrappers 13
3.10 Thwart SSH crackers/brute force attacks 13
3.11 Account Locking 14
3.12 Configure idle log out timeout interval 14
3.13 Enable a warning banner for SSH users 15
3.14 Patch OpenSSH 15
3.15 Disable host-based authentication 15

1 Introduction

The purpose of system hardening is to eliminate as many security risks as possible. It is done by:

1. Making changes to default system configuration
2. Installing additional security software

2 Linux Server Hardening Checklist

The following instructions assume that you are using CentOS/RHEL based Linux distribution.

2.1 Encrypt Data Communication

All data transmitted over a network is open to monitoring. Encrypt transmitted data whenever possible with a password or using keys/certificates.

1. Use **SCP**, rsync, or **SFTP** for file transfer. You can also mount a remote server file system or your home directory using special sshfs and fuse tools
2. GnuPG allows to encrypt and sign your data and communication, features a versatile key management system as well as access modules for all kind of public key directories
3. OpenVPN is a cost-effective, lightweight SSL VPN. Another option is to try out tinc that uses tunneling and encryption to create a secure private network between hosts on the Internet or private insecure LAN
4. Install and configure Apache SSL HTTPS mod_ssl

2.2 Avoid Using FTP, Telnet, and Rlogin/Rsh Services

Under most network configurations, usernames, passwords, FTP/telnet/rsh commands, and transferred files can be captured by anyone on the same network using a packet sniffer. The standard solution to this problem is to use either **OpenSSH**, **SFTP**, or **FTPS** (FTP over SSL), which adds SSL or TLS encryption to FTP. Type the following “yum” command to delete **NIS**, **rsh** and other outdated services:

```
$ yum erase xinetd ypserv tftp-server telnet-server rsh-server
```

2.3 Minimize Software to Minimize Vulnerability

Avoid installing unnecessary software to avoid vulnerabilities in software. Use the RPM package manager to review all installed set of software packages on a system. Delete all unwanted packages.

```
$ yum list installed
$ yum list packageName
$ yum remove packageName
```

2.4 One Network Service Per System or VM Instance

Run different network services on separate servers or VM instances. This limits the number of other services that can be compromised. For example, if an attacker was able to exploit software such as Apache flow successfully, he or she will get access to the entire server, including other services such as a database, e-mail server, and so on.

2.5 Keep Linux Kernel and Software Up to Date

Applying security patches is a vital part of maintaining a Linux server. Linux provides all necessary tools to keep your system updated, and also allows for easy upgrades between versions. Use the RPM package manager to apply all security updates:

```
$ yum update
```

2.6 Turn on SELinux

Security-Enhanced Linux (**SELinux**) is an access control security mechanism provided in the kernel.
SELinux provides three basic modes of operation:

- **Enforcing**: This is the default mode which enables and enforce the SELinux security policy on the machine
- **Permissive**: In this mode, **SELinux** will not enforce the security policy on the system, only warn and log actions.
- **Disabled**: **SELinux** is turned off

It can be managed from “/etc/selinux/config” file, where you can enable or disable it.

2.7 User Accounts and Strong Password Policy

Use the **useradd/usermod** commands to create and maintain user accounts. Make sure you have a good and strong password policy. For example, a good password includes at least eight characters long and a mixture of alphabets, number, special character, upper & lower alphabets, etc.

Use tools such as “John the ripper” to find out weak user passwords on your server.

To enforce specific password policy, check the following paragraph: 3.6 Use strong passwords

2.8 Password Aging

The “**chage”** Linux command changes the number of days between password changes and the date of the last password change. This information is used by the system to determine when a user must change his/her password.

The “/etc/login.defs” file defines the site-specific configuration for the shadow password suite, including password aging configuration.
To get password expiration information, enter:

```
$ chage -l username
```

To set the maximum number of days during which a password is valid to 60, the minimum number of days between password changes to 7, and the number of days of warning before a password change is required to 7 for the specific user, type the following command:

```
$ chage -M 60 -m 7 -W 7 userName
```

2.9 Restricting Use of Previous Passwords

You can prevent all users from using or re-using the same old passwords under Linux.

Edit “/etc/pam.d/system-auth” file and update the existing password line and append remember=10 to prevent a user from re-using any of his or her last ten passwords.

password sufficient pam_unix.so use_authtok md5 shadow remember=10

2.10 Make Sure No Non-Root Accounts Have UID Set To 0

Only “root” account has UID 0 with full permissions to access the system. Type the following command to display all accounts with UID set to 0:

```
$ awk -F: '($3 == "0") {print}' /etc/passwd
```

You should only see one line as follows:

```
root:x:0:0:root:/root:/bin/bash
```

If you see other lines, delete them or make sure other accounts are authorized by you to use UID 0.

2.11 Physical Server Security

You must protect the Linux server's physical console access. Configure the BIOS and disable the booting from external devices such as DVD/CD/USB pen. Set BIOS and GRUB boot loader password to protect these settings. You can also apply the following:

- **Enable Authentication for Single-User Mode:**
  Add the following line to "/etc/inittab":
  ~~:S:wait:/sbin/sulogin

- **Disable Interactive Hotkey Startup at Boot:**
  Modify the settings in "/etc/sysconfig/init" file as follows:
  PROMPT=no

- **Setup Screen Locking:**
  To install the "vlock" package, enter:

  ```
  $ yum install vlock
  ```

  vlock is a program used to lock one or more sessions on the Linux console. This is especially useful for Linux machines, which have multiple users with access to the console.

  To lock the console, enter:

  ```
  $ vlock
  ```

  The -an option can be to used lock all console sessions and disable VC switching, enter:

  ```
  $ vlock -a
  ```

- **Disable Ctrl+Alt+Delete:**
  Anyone that has physical access to the keyboard can simply use the "Ctrl+Alt+Delete" key combination to reboot the server without having to log on.

  To disable "Ctrl+Alt+Delete" update "/etc/inittab" and make sure following line is commented out:

  ca::ctrlaltdel:/sbin/shutdown -t3 -r now

2.12 Disable Unwanted Services

Disable all unnecessary services and daemons (services that run in the background). You need to remove all unwanted services from the system startup. Type the following command to list all services that are started at boot time:

```
$ systemctl list-unit-files
```

To disable a specific service, enter:

```
$ systemctl stop ServiceName
$ systemctl disable ServiceName
```

2.13 Find Listening Network Ports

Use the following commands to list all open ports and associated programs:

```
$ netstat -tulpn
$ nmap -sT -O localhost
```

2.14 Configure Iptables and TCPWrappers

Iptables is a user-space application program that allows you to configure the firewall (Netfilter) provided by the Linux kernel. Use the firewall to filter out traffic and allow only necessary traffic.
Also, use the **TCPWrappers,** a host-based networking **ACL** system, to filter network access to the Internet.

2.15 Linux Kernel Hardening

"/etc/sysctl.conf" file is used to configure kernel parameters at runtime. Linux reads and applies settings from "/etc/sysctl.conf" at boot time.

You can configure various Linux networking and system settings such as:

- Limit network-transmitted configuration for IPv4
- Limit network-transmitted configuration for IPv6
- Turn on ExecShield protection
- Prevent the typical "syn flood attack."
- Turn on source IP address verification
- Prevent a cracker from using a spoofing attack against the IP address of the server
- Log several types of suspicious packets, such as spoofed packets, source-routed packets, and redirects

2.16 Separate Disk Partitions

Separation of the operating system files from user files may result in a better and secure system. Make sure the following filesystems are mounted on the separate partitions:

- /swap
- /boot
- /
- /home

2.17 Disk Quotas

Make sure the disk quota is enabled for all users. To implement disk quotas, use the following steps:

1. Enable quotas per file system by modifying the “/etc/fstab” file
2. Remount the file system(s)
3. Create the quota database files and generate the disk usage table
4. Assign quota policies

2.18 Turn Off IPv6

Internet Protocol version 6 (IPv6) provides a new Internet layer of the TCP/IP protocol suite that replaces Internet Protocol version 4 (IPv4) and offers many benefits. If you are NOT using IPv6, disable it.

1. Edit “/etc/sysctl.conf” and append the following lines:
   net.ipv6.conf.all.disable_ipv6 = 1
   net.ipv6.conf.default.disable_ipv6 = 1
2. To make the settings effective, execute:
   ```
   $ sysctl -p
   ```
3. Add the “AddressFamily” line to “/etc/ssh/sshd_config”:
   AddressFamily inet
4. Restart **sshd** service to apply the changes:
   ```
   $ systemctl restart sshd
   ```

2.19 Disable Unwanted SUID and SGID Binaries

All SUID/SGID bits enabled file can be misused when the SUID/SGID executable has a security problem or bug. All local or remote users can use such a file. It is a good idea to find all such files. Use the find command as follows (it is essential to investigate each reported file):

- See all set user id files:
  ```
  $ find / -perm +4000
  ```
- See all group id files:
  ```
  $ find / -perm +2000
  ```
- Or combine both in a single command:
  ```
  $ find / \( -perm -4000 -o -perm -2000 \) -print
  $ find / -path -prune -o -type f -perm +6000 -ls
  ```

2.20 World-Writable Files

Anyone can modify a world-writable file, resulting in a security issue. Use the following command to find all world-writable and sticky bits set files:

```
$ find /dir -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print
```

You need to investigate each reported file and either set correct user and group permission or remove it.

2.21 Files Without Ownership

Files not owned by any user or group can pose a security problem. Just find them with the following command which does not belong to a valid user and a valid group:

```
$ find /dir -xdev \( -nouser -o -nogroup \) -print
```

You need to investigate each reported file and either assign it to an appropriate user and group or remove it.

2.22 Use A Centralized Authentication Service

Without a centralized authentication system, user auth data becomes inconsistent, which may lead to out-of-date credentials and forgotten accounts that should have been deleted in the first place. A centralized authentication service allows maintaining central control over Linux/UNIX account and authentication data. You can keep auth data synchronized between servers. Do not use the **NIS** service for centralized authentication. Use OpenLDAP for clients and servers.

2.23 Kerberos

Kerberos performs authentication as a trusted third-party authentication service by using shared cryptographic secrets under the assumption that packets traveling along the insecure network can be read, modified, and inserted.

Kerberos builds on symmetric-key cryptography and requires a key distribution center. You can make remote login, remote copy, secure inter-system file copying, and other high-risk tasks safer and more controllable using Kerberos. So, when users authenticate to network services using Kerberos, unauthorized users attempting to gather passwords by monitoring network traffic are effectively thwarted.

2.24 Logging and Auditing

You need to configure logging and auditing to collect all hacking and cracking attempts. By default, syslog stores data in "/var/log/" directory.

Common Linux log files, names, and usage:

- **/var/log/messages**: general message and system related stuff
- **/var/log/auth.log**: authentication logs
- **/var/log/kern.log**: kernel logs
- **/var/log/cron.log**: *crond* logs (cron job)
- **/var/log/maillog**: mail server logs
- **/var/log/yum.log**: *yum* command log file
- **/var/log/utmp** or **/var/log/wtmp**: Login records file
- **/var/log/boot.log**: system boot log
- **/var/log/httpd/**: Apache access and error logs directory
- /var/log/**secure** or **/var/log/auth.log**: authentication log

2.25 System Accounting with auditd

The auditd is used for system auditing. It is responsible for writing audit records to the disk. During startup, the rules in "/etc/audit.rules" are read by this daemon. You can open "/etc/audit.rules" file and make changes such as setting audit file log location and other options. With **auditd** you can answer the following questions:

1. System startup and shutdown events (reboot/halt)
2. Date and time of the event
3. User responsible for the event (such as trying to access /path/to/topsecret.dat file)
4. Type of event (edit, access, delete, write, update file & commands)
5. Success/failure of the event
6. Record events that modify date and time
7. Find out who made changes to modify the system's network settings
8. Record events that modify user/group information
9. See who made changes to a file and so on

2.26 Install and Use Intrusion Detection System

A network intrusion detection system (NIDS) is an intrusion detection system that tries to detect malicious activity such as a denial of service attacks, port scans, or even attempts to crack into computers by monitoring network traffic.

OSSEC (Open Source HIDS Security) is a free, open-source host-based intrusion detection system (HIDS). It performs log analysis, integrity checking, Windows registry monitoring, rootkit detection, time-based alerting, and active response. **OSSEC** has a log analysis engine that can correlate and analyze logs from multiple devices and formats.

2.27 Disable USB/firewire/thunderbolt devices

Type the following command to disable USB devices on Linux system:

```
$ echo 'install usb-storage /bin/true' >> /etc/modprobe.d/disable-usb-storage.conf
```

You can use same method to disable firewire and thunderbolt modules:

```
$ echo "blacklist firewire-core" >> /etc/modprobe.d/firewire.conf
$ echo "blacklist thunderbolt" >> /etc/modprobe.d/thunderbolt.conf
```

Once done, users cannot quickly copy sensitive data to USB devices or install malware/viruses or backdoor on your Linux based system.

2.28 Backups

It cannot be stressed enough how important it is to make a backup of your Linux system. A proper offsite backup allows you to recover from a cracked server i.e., an intrusion. The traditional UNIX backup programs like **dump** and **restore** are also recommended. You must set up encrypted backups to external storage such as **NAS** server or **FreeNAS** server or use cloud computing service.

3 Hardening SSH

OpenSSH is the implementation of the SSH protocol. OpenSSH is recommended for remote login, making backups, remote file transfer via SCP or SFTP, and much more. SSH is perfect for keeping confidentiality and integrity for data exchanged between two networks and systems. The next tips explain how to secure your OpenSSH server running on a Linux system to improve your system security.

3.1 Use SSH public key based login

OpenSSH server supports various authentication. It is recommended that you use public key-based authentication.

1. First, create the key pair using the following “ssh-keygen” command on your local desktop/laptop: `$ssh-keygen -t rsa -b 4096` (default location: ~/.ssh/)
2. Install the public key on the target machine using “ssh-copy-id” command: `$ ssh-copy-id username@target_machine`
3. When prompted, supply the user password. Verify that SSH key-based login is working for you: `$ ssh username@target_machine`

3.2 Disable root user login

Open “/etc/ssh/sshd_config” file and add the following line:
PermitRootLogin no

3.3 Disable password-based login where practical

Whenever practical, password-based logins should be disabled. Only public key-based logins are allowed. Add the following in your “sshd_config” file:
AuthenticationMethods public key
PubkeyAuthentication yes

3.4 Limit Users’ SSH access

By default, all system users can log in via SSH using their passwords or public key. To allow only user “ytmlogin” that privilege, add the following to “sshd_config”:
AllowUsers ytmlogin

3.5 Disable Empty Passwords

You need to explicitly disallow remote login from accounts with empty passwords, update “sshd_config” with the following line:
PermitEmptyPasswords no

3.6 Use strong passwords and passphrase for users/keys

Brute force attack works because users go for dictionary-based passwords. You can force users to use strong passwords of a specified length and special characters.

Navigate to “/etc/pam.d/system-auth” file as “root” user and add the restrictions by modifying the line containing “pam_pwquality.so” PAM module:

password requisite pam_pwquality.so try_first_pass minlen=15 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 local_users_only retry=3 authtok_type=

The presented solution requires an eight-character password with at least one digit, one punctuation-type character, and a minimum one lower-case and upper-case letter.

3.7 Firewall SSH TCP port # 22

To allow SSH connection only from the local network using ***iptables***, for example, 192.168.1.0/24, execute the following command:

```
$ iptables -A INPUT -p tcp -s 192.168.1.0/24 --dport 22 -m conntrack
--ctstate NEW,ESTABLISHED -j ACCEPT
```

3.8 Change SSH port and limit IP binding

By default, SSH listens to all available interfaces and IP addresses on the system. Limit SSH port binding and change SSH port (many brutes forcing scripts only try to connect to TCP port # 22). To bind to 192.168.1.10 and port 300, add or correct the following line in “sshd_config”:

Port 300
ListenAddress 192.168.1.10

3.9 Use TCP wrappers

TCP Wrapper is a host-based Networking ACL (Access Control List) system, used to filter network access to the Internet. OpenSSH does support TCP wrappers. Just update your “/etc/hosts.allow” file as follows to allow SSH only from 192.168.1.2 and 172.16.23.12 IP address:

sshd : 192.168.1.2 172.16.23.12

3.10 Thwart SSH crackers/brute force attacks

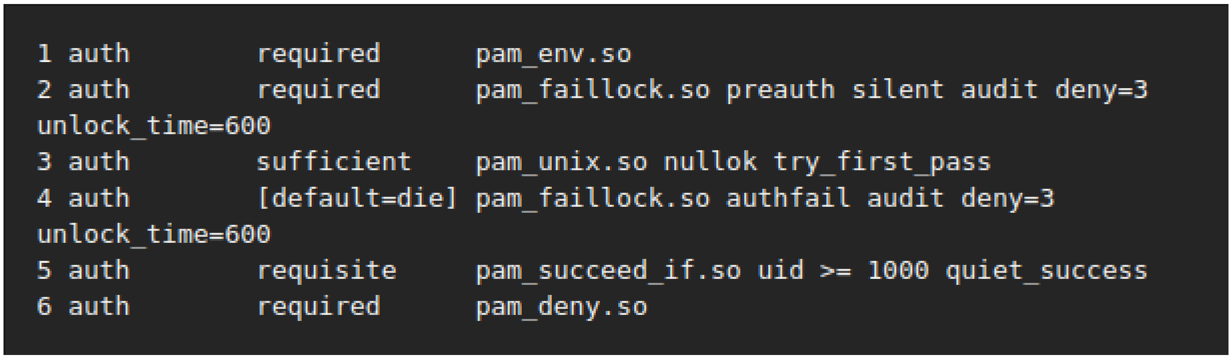
Brute force is a method of defeating a cryptographic scheme by trying a large number of possibilities (the combination of users and passwords) using a single or distributed computer network. To prevent brute force attacks against SSH, use the following software:

1. **DenyHosts** is a Python-based security tool for SSH servers. It is intended to prevent brute force attacks on SSH servers by monitoring invalid login attempts in the authentication log and blocking the originating IP addresses
2. **Fail2ban** is a similar program that prevents brute force attacks against SSH
3. **sshguard** protect hosts from brute force attacks against SSH and other services

3.11 Account Locking

In Red Hat Enterprise Linux 7, the “pam_faillock” PAM module allows system administrators to lockout user accounts after a specified number of failed attempts. Limiting user login attempts serves mainly as a security measure that aims to prevent possible brute force attacks targeted to obtain a user's account password.

1. To lock out any non-root user after three unsuccessful attempts and unlock that user after ten minutes, add two lines to the auth section of the “/etc/pam.d/system-auth” and “/etc/pam.d/password-auth” files. After your edits, the entire auth section in both files should look like this:



```
auth        required      pam_env.so
auth        required      pam_faillock.so preauth silent audit deny=3 unlock_time=600
auth        sufficient    pam_unix.so nullok try_first_pass
auth        [default=die] pam_faillock.so authfail audit deny=3 unlock_time=600
auth        requisite     pam_succeed_if.so uid >= 1000 quiet_success
auth        required      pam_deny.so
```

2. Add the following line to the account section of both files specified in the previous step:
   account required pam_faillock.so
3. To view the number of failed attempts per user, run, as root, the following command:
   `$ faillock`
4. To unlock a user's account, run, as root, the following command:
   `$ faillock --user <username> --reset`

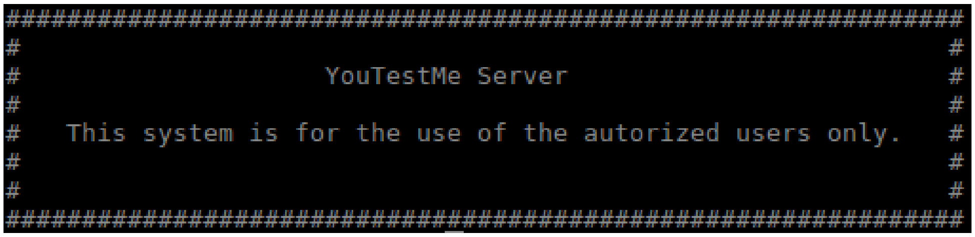
3.12 Configure idle log out timeout interval

A user can log in to the server via SSH, and you can set an idle timeout interval to avoid unattended SSH sessions. Open “sshd_config” and make sure the following values are configured:
ClientAliveInterval 300
ClientAliveCountMax 0

You are setting an idle timeout interval in seconds (300 secs == 5 minutes). After this interval has passed, the idle user will be automatically kicked out (read as logged out).

3.13 Enable a warning banner for SSH users

Set a warning banner by updating “sshd_config” with the following line:
Banner /etc/issue



```
###############################################################
#                                                             #
#                    YouTestMe Server                         #
#                                                             #
#   This system is for the use of the autorized users only.   #
#                                                             #
#                                                             #
###############################################################
```

3.14 Patch OpenSSH

It is recommended that you use tools such as **yum**, **apt-get**, **freebsd-update**, and others to keep systems up to date with the latest security patches.

3.15 Disable host-based authentication

Host-based authentication allows hosts to authenticate on behalf of all or some of the system's users. To disable it, update “sshd_config” with the following option:
HostbasedAuthentication no