Homi👋
LEARN DEVOPS & CLOUD

Linux Server Hardening

PUBLISHED NOVEMBER 4, 2022

Introduction

The most secure server in the world is the one that is powered off and disconnected from the Internet. But if we want to use the server for IT operations, we must make sure that the server is secured properly before making it online.

Server Hardening is the process of making the system secure. Server hardening helps to address the best ways to minimize the points of vulnerability.

In this article, I am going to explain some of the best practices to secure the Linux servers.

Server Hardening can be done in various levels of the system:

1. **Machine Level :** Securing BIOS/drive encryption
2. **System Level:** Password policies
3. **Network Level:** Firewalls/Port configuration
4. **Logging and Auditing Level:** alerting/logs/audits

Server Hardening Checklist

Now, let's look at some of the common and best practices that we can adopt to secure the system:

Regular System Update

This is one of the crucial step in system hardening because most of the flaws in the systems are because of some flaws in the software definition itself. Proper system update helps with reducing the risks related to it.

Depending on the distribution of Linux we use, we can update the system with the help of following commands:

```
sudo apt update -y && sudo apt upgrade -y         // For Ubuntu
sudo yum update -y && sudo yum upgrade -y         // For RHEL/CentOS
```

Block root login access from SSH

Since the root user of the system has administrative access, it should not be permitted for SSH login from remote desktops.

To disable the root access, make necessary changes to the SSH configuration file at **/etc/ssh** directory.

```
cd /etc/ssh/
sudo vi sshd_config
```

Edit the file and search for the pattern:

```
#PermitRootLogin yes
```

We need to uncomment the line and change the value to **no** in order to block the SSH root access.

```
PermitRootLogin no
```

Now, restart the SSH service to reflect the changes:

```
sudo systemctl restart sshd.service
```

Enforce strong password for root

Make sure to set a strong password for the root user. It is better to generate a hash for a plain text password and use the generated hash for the password.

This guarantees more security and reliability of the system.

Stop using default ports

Default ports are standard ports for any service or application. They are known to the world and anyone can enter into the system through the port if it is publicly accessible.

Therefore, we must set the random port for every service we use.

For example:

Instead of using port 22 for SSH, we can use any random port we wish.

Here's how we can change the default SSH port '22' to any other random port. SSH ports are well-known to the world and therefore it is very important that we set this to a random number.

Login as root user and edit the ssh configuration file as:

```
sudo -i
cd /etc/ssh/
vi sshd_config
```

Find the parameter 'PORT 22' on line 17 of the file (usually 17).

```
PORT 22
```

Change the value to the port number you wish to set for SSH and save the file.

Restart the SSH service with the help of following command and you are good to go.

```
systemctl restart sshd.service
```

The port for SSH has been changed successfully.

However, if the linux server is running on cloud, we need to add the new port in the Inbound Rules for the instance in its security group too.

Closed unused open ports

Open ports opens the attacking surface for hackers. Thus, we must not keep any unnecessary service or applications. And if we find one, we must disable the port or remove the application completely.

To check what ports are opened in the server, we can use the command:

```
netstat -tulpn
```

Keep things clean and sorted

We should never forget that *Fewer packages = Fewer Vulnerabilities*

Remove any unnecessary package installed and never install any packages that is not needed. Application packages may contain vulnerabilities that can cause a serious damage to the system.

To remove any package installed, we can use the command:

```
sudo apt-get remove <package-name> --purge        // Ubuntuyum -y remove <package-
name>                      // RHEL/CentOS
```

Also make sure to clean up any old directories and files which are no longer in use.

> **Hardening Linux Kernel – /etc/sysctl.conf**

In order to make a permanent modification, we need to configure the file **sysctl.conf** located inside the **/etc/** directory.

Login as a root user and edit the file as:

```
sudo -i
vi /etc/sysctl.conf
```

It can be used for configuring various security and performance tuning parameters such as:

Control IP packet forwarding:

Servers that acts as routers or gateways need to forward the packets and hence this feature needs to be enabled on such cases. However, in all other servers it is not needed and can be disabled by adding the following lines in **sysctl.conf.**

```
net.ipv4.ip_forward = 0
```

Source IP Address Verification

Helps to check the spoofing attacks. Add the following lines to **sysctl.conf** to enable source IP address verification.

```
net.ipv4.conf.all.rp_filter = 1
```

Exec shield Protection

Exec shield is a security Linux kernel patch to avoid worms and other problems. Add the following lines to **sysctl.conf** to enable exec-shield protection.

```
kernel.exec-shield = 1
kernel.randomize_va_space = 1
```

Prevention from smurf attacks

Smurf attack refers to exploiting the IP broadcast address to create a DoS/DDoS attack through ICMP message packets.

We can stop smurf attacks by disabling the broadcast address as:

```
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

SYN Flood Protection

In SYN Flood attack, the system is flooded with a series of SYN packets.

Then, the system issues an SYN+ACK response and waits for the ACK (3 way handshake).

Since, the attacker never sends back the ACK, the entire resource of the system gets filled, known as backlog queue. And once the queue is full, it ignores the further requests from legit users for various services like http/mail etc.

To prevent the system from SYN Flood Attack, we need to enable the SYN cookies in **sysctl.conf** file by adding the following:

```
net.ipv4.tcp_syncookies = 1
```

Setup Login Banner

Setting up a banner does not contribute in securing the system.

However, banners can be used to show some messages or warnings when someone is establishing SSH connection to the server.

To enable the banner in SSH, first of all we need to create a **banner file** at **/etc/mybanner.** This will be a new file and fill it with desired message and save and exit from the file.

```
sudo vi /etc/mybanner
```

Now, we need to point this banner file by setting the file path to ssh configuration file. Search for the pattern '**Banner**' and uncomment it and set the value to the file path of your banner file as:

```
Banner /etc/mybanner
```

Restart the SSH service:

```
sudo service sshd restart
```

It worked. The banner is displayed above the welcome message as shown above

That's all.

Happy Learning 🙂

CATEGORIES: DevOps Linux

TAGS: linux linux commands linux hardening linuxserver

PREVIOUS POST
Automate backup using Ansible

NO NEWER POSTS
Return to Blog

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name*

Email*

Website

Save my name, email, and website in this browser for the next time I comment.

Post Comment