5 ways to improve Linux user account security

The first and most important step to secure Linux servers and systems is to prevent unnecessary access by malicious parties. Proper user account control is one of many ways to increase the security of your system.

A well-protected user account is able to prevent common attack methods. Therefore, as a Linux system administrator, it is also your responsibility to protect your server through effective security techniques.

This article covers some basic user account security controls to prevent unnecessary access and fix vulnerabilities that can occur when a system is compromised.

1. Restrict root account access

By default, every Linux system installation establishes a root account that can be accessed by anyone from the outside via SSH. However, accessing the root account via SSH or multiple users accessing inside the system can cause problems.

For example, an attacker could log in as the root user and gain access to the system.

To restrict unnecessary root access from within or outside your Linux system, you can:

1. Add another user and give that user root privileges
2. Disable SSH root login

Create a new superuser

To grant sudo or root privileges to a regular Linux user account, add the user to the sudo group as follows:

```
usermod -aG sudo username
```

Now switch to the user account with the su command and verify its root privileges by issuing the command that only the root user can reach:

```
su - username sudo systemctl restart sshd
```

Enabling sudo permissions provides some nice security benefits, such as:

1. You don't need to share the root password with regular users.
2. It helps you to check all commands run by common users, which means it stores details about who, when and where execute the command in the file /var/log/secure.
3. In addition, you can edit the /etc/sudoers file to limit the superuser privileges of normal users. You can use the command su -l to check the current root privileges of the user.

Disable SSH root login

To disable SSH root access on the system, first open the main configuration file.

```
sudo vim /etc/ssh/sshd_config
```

Now, uncomment the following line to set the root login permission to no:

```
PermitRootLogin no
```

Save the file and restart the sshd service by typing:

```
sudo systemctl restart sshd
```

Now, whenever you try to SSH into the system as the root user, you will get the following error message:

```
Permission denied, please try again.
```

2. Set expiration date on account

Another effective way to control unnecessary access is to set expiration dates on accounts created for temporary use.

For example, if an intern or an employee needs access to the system, you can set an expiration date during account creation. It's a precaution in case you forget to manually delete an account after they leave the organization.

Use the chage command with the grep utility to fetch account expiration date details for the user:

```
chage -l username| grep account
```

Output:

```
Account expires : never
```

As shown above, the output says there is no expiration date. Now use the usermod command with the -e flag to set the expiration date in YYYY-MM-DD format and verify the change with the chage command above.

```
usermod -e 2021-01-25 username chage -l username| grep account
```

3. Improved security for account passwords

Enforcing a strong password policy is an important aspect of user account security, as weak passwords allow attackers to easily break into your system through brute-force attacks, Dictionary or Rainbow Table.

Choosing an easy-to-remember password can offer some convenience, but it also opens up many opportunities for attackers to guess passwords with the help of tools and word lists available online.

Set password expiration date

Linux provides several default options inside the /etc/logins.defs file that allow you to set an account password expiration date. Use the chage command and record the password expiration details as follows:

```
chage -l username | grep days
```

Variable Default ValueUses Ideal ValuePASS_MAX_DAYS9999 The default number of days to use a password depends on your account setup type40PASS_MIN_DAYS0Prevent users from changing their password immediately5PASS_MIN_LEN5Force users to set a password of a certain length15PASS_WARN_AGE0User Alerts change your password before being forced to do so7

For existing accounts, you can control password aging with the help of the chage command to set PASS_MAX_DAYS, PASS_MIN_DAYS and PASS_WARN_AGE to 4, 5 and 7.

```
chage -M 40 -m 5 -W 7 username
```

Password Hash

Another way to enhance account password security is to store password hashes inside /etc/shadow. Hashes are one-way math functions that take a password as input and output an irreversible string.

Previously, on Linux systems, whenever a user entered their password to log in, the system generated its hash and cross-checked it with the code stored in the file /etc/passwd.

However, there is a problem with the passwd file permissions, anyone with system access can read the file and crack the hash using the rainbow table.

Therefore, Linux now stores hash functions inside the /etc/shadow file with the following set of permissions:

```
ls -l /etc/shadow
---------- 1 root root 1626 Jan 7 13:56 /etc/shadow
```


You can still install Linux with the old ways of storing hashes. You can modify that by running the pwconv command, which will automatically save the password hashes to the file /etc/shadow. Similarly, you can enable another method (file /etc/passwd) with the pwunconv command.

4. Delete unused user accounts

Bad guys can exploit unused and expired accounts in the system, by renewing it and making it appear as a legitimate user. To delete an inactive account and associated data whenever a user leaves the organization, first find all files associated with the user:

```
find / -user username
```

Then disable the account or set an expiration date as discussed above. Don't forget to back up user-owned files. You can choose to assign files to a new owner or delete them from the system.

Finally, delete the user account with the userdel command.

```
userdel -f username
```

5. Restrict remote access to a specific group of users

If you are hosting a web server on your Linux machine, you may need to allow only specific users to have remote SSH access to the system. OpenSSL allows you to limit users by cross-checking if they belong to a specific group.

For that, create a user group named ssh_gp, add the users you want to give remote access to the group and list the user group information as follows:

```
sudo groupadd ssh_gp
sudo gpasswd -a username ssh_gp
groups username
```

Now, open the main OpenSSL configuration file to include the ssh_gp allowed user group.

```
sudo vim /etc/ssh/sshd_config
AllowGroups ssh_gp
```

Remember to uncomment the line to ensure successful group inclusion. When done, save and exit the file and restart the service:

```
sudo systemctl restart sshd
```

Given the importance of setting up, managing, and securing user accounts is a fundamental challenge faced by Linux administrators. This article has listed some security measures that an account administrator must take to protect the system from potential threats caused by unprotected user accounts.

You finished reading the article "**5 ways to improve Linux user account security**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.