How to effectively secure a Linux system with YubiKey

One of the most popular solutions to authentication problems is YubiKey. But what is YubiKey and how does hardware authentication work? Can you secure your Linux PC with YubiKey?

It's not just you who are worried about the growing hacking threat. While the 2FA and authentication prompts are enough to deter most potential hackers, thousands of breaches still succeed every day.

One of the most popular solutions to authentication problems is YubiKey. But what is YubiKey and how does hardware authentication work? Can you secure your Linux PC with YubiKey?

Choose the right YubiKey for your system

If you want to use YubiKey for authentication on your Linux computer, there are several outstanding YubiKeys to choose from. The YubiKey 5 and the YubiKey 5 NFC are both classic options respectively that work well with systems with USB-A and USB-C.

If you want to use YubiKey with Linux computers and Android phones, you should consider the YubiKey 5c NFC. If you have a Linux computer and an iPhone, you should consider the YubiKey 5ci as it supports USB-C and Lightning ports.

It is important to note that the YubiHSM Series is not compatible with sudo authentication. Legacy YubiKeys may or may not be compatible with sudo/SSH authentication depending on their specific features.

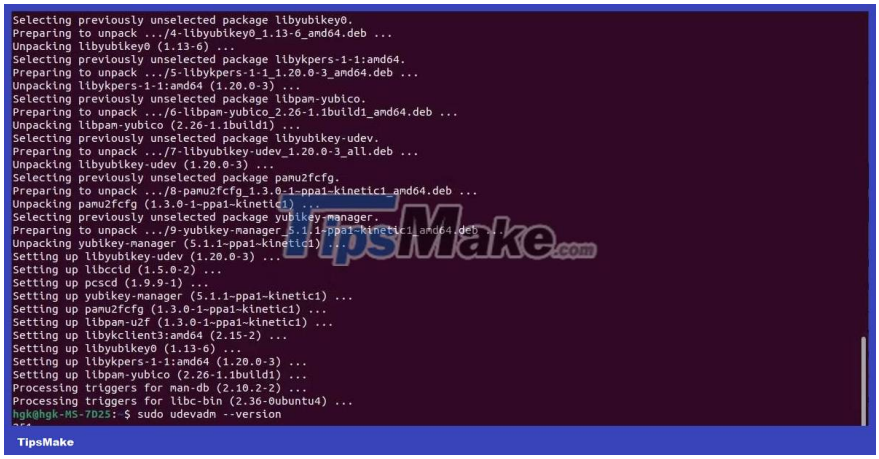
Before getting started with sudo or SSH authentication, you must install the YubiKey PPA. Open terminal and enter the following commands to update packages, then install YubiKey Authenticator and YubiKey Manager:

```
sudo add-apt-repository ppa:yubico/stable sudo apt-get update sudo apt install yu
```

Next, you will need to verify that your system is ready to work with YubiKey. Run the following command in terminal to check your udev version:

```
sudo udevadm --version
```

Terminal will return a number. If the number is 244 or higher, your system is compatible with YubiKey. You can skip the next step in this case.



If not, you will need to configure your system. You should use the following commands to check if udev is installed on your computer - and to install if not:

```
dpkg -s libu2f-udev sudo apt install libu2f-udev
```

Next, check if your YubiKey's U2F interface is unlocked. If you have a YubiKey NEO or a YubiKey NEO-n, plug the YubiKey in, open the YubiKey Manager and navigate to **Interfaces.** Enable **the U2F interface** and click **Save.**

Set up YubiKey for sudo authentication on Linux

sudo is one of the most dangerous commands in the Linux environment. However, it also offers an impressive level of access, enough to get most jobs done. In the wrong hands, the root-level access sudo provides can allow malicious users to exploit or destroy the system.

YubiKeys are great for sudo authentication because their authentication is nearly impossible to replicate without access to the YubiKey itself. Most YubiKey are compatible with sudo authentication, including 5 Series FIPs, Key Series, 4 Series FIPs, Bio Series, 5 Series and 4 Series.

According to Yubico, the first step you need to take to configure sudo authentication is to create a rules file. If your udev version is 188 or higher, install the new U2F rules from GitHub and copy the file **70-u2f.rules** to **/etc/udev/rules.d.**

If your udev version is below 188, install the old U2F rules from GitHub and copy the file **70-old-u2f.rules** to **/etc/udev/rules.d.**

If your udev version is 244 or higher or you have already created the necessary rule files, then you are ready to link YubiKey to your account.

Plug the YubiKey into your computer, open Terminal, and enter the following commands to link the YubiKey to your account:

```
mkdir -p ~/.config/Yubicopamu2fcfg > ~/.config/Yubico/u2f_keys
```

Wait a few minutes until the indicator light on your YubiKey starts flashing. Touch the button on the YubiKey to confirm the device link.

If another YubiKey is available, you should add it as a backup device by entering the following command and completing the same process:

```
pamu2fcfg -n >> ~/.config/Yubico/u2f_keys
```

Finally, you will need to configure the sudo command to require YubiKey authentication. You should start by entering the following command to open the sudo configuration file:

```
sudo vi /etc/pam.d/sudo
```

Once the configuration file is open, paste the following line just below the **@include common-auth** line to configure sudo to require YubiKey authentication:

```
auth required pam_u2f.so
```

Save and exit the file by pressing **Escape** , typing **:wq** and pressing **Enter** , but keep the terminal open. You won't be able to reverse the changes you've made to sudo authentication if Terminal is closed.

Open a second Terminal and run the following command after pulling out the YubiKey, then enter your password:

```
sudo echo testing
```

Authentication will fail. Plug in your YubiKey, re-enter your command and password. When the YubiKey indicator light starts flashing, press the button on the YubiKey. It will validate the command. If so, your YubiKey is fully set up for sudo authentication.

How to set up YubiKey for SSH authentication

You can also use YubiKey for SSH authentication! Several YubiKey series are compatible with SSH, including 5 FIPS Series, 5 Series, 4 FIPS Series, and 4 Series. Using YubiKey to authenticate connections will allow you to make each SSH login much more secure.

The best method for setting up YubiKey has been outlined by an experienced user on GitHub. You will need SSH 8.2 or later and YubiKey with firmware 5.2.3 or later. You can check your OpenSSH version - and update if necessary - with the following commands:

```
ssh -V sudo apt update && sudo apt upgrade
```

Next, you will need to configure SSH to accept your YubiKey. Enter the following command to open the vi editor and edit the configuration file:

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

Add the following line to the configuration file to get your YubiKey accepted:

```
PubkeyAcceptedKeyTypes sk-ecdsa-sha2-nistp256@openssh.com,sk-ssh-ed25519-cert-v0
```

Save and exit the file by pressing **Escape** , typing **:wq** and pressing **Enter.** Finally, restart the SSH service with the following command to get your new configuration working:

```
sudo service ssh restart
```

Finally, you are ready to generate the key pair that you will use for SSH authentication. Navigate to the SSH directory and generate a new SSH key with the following commands:

```
cd home/username/.ssh ssh-keygen -t ed25519-sk
```

Two files will be created in the ~/.ssh/ directory **.** Note that you may need to use **ecdsa-sk** instead of **ed25519-sk** if your system is not compatible and Terminal prompts for failed key registration.

Next, you will need to add the public key to your server with the following command:

```
ssh-copy-id -i ~/.ssh/id_ed25519_sk.pub username@server
```

You should also add yourself to the sudoers file to maintain permissions after disabling root login. Access the file and open it with visudo.

Warning : Do not open the sudoers file with a regular text editor.

Below the line that says **root ALL=(ALL:ALL) ALL** , add the following:

```
username ALL=(ALL:ALL) ALL
```

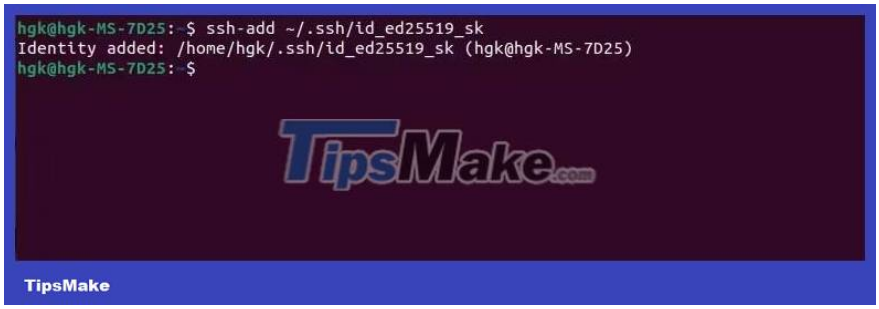
Open the file **/etc/ssh/ssd_config** and add the following lines to disable root login and password-based login:

```
ChallengeResponseAuthentication noPermitRootLogin no
```

Finally, enter the following command to load your key into the SSH agent for the duration of the session:

```
ssh-add ~/.ssh/id_ed25519_sk
```

You can now use your YubiKey for SSH authentication. You will need to plug the YubiKey into your computer when prompted and press the button when the indicator light flashes. With this new authentication method, SSH access to your remote server will be significantly more secure.



You finished reading the article "**How to effectively secure a Linux system with YubiKey**" 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.
