Create Locally Trusted SSL Certificates with mkcert on Ubuntu 18.04

By koromicha - October 7, 2020

Hello folks, welcome to this very tutorial on how to create locally trusted SSL certificates with mkcert on Ubuntu 18.04.

mkcert is a simple zero-config tool that is used to make locally trusted development certificates. It automatically creates and installs a local CA in the system root store, and generates locally-trusted certificates.

Using certificates from real certificate authorities (CAs) for development can be dangerous or impossible (for hosts like `localhost` or `127.0.0.1`), but self-signed certificates cause trust errors. Managing your own CA is the best solution, but usually involves arcane commands, specialized knowledge and manual steps, but not any more with the availability of mkcert utility.

Without much theory, let us have a look how mkcert can help you on this.

Search

Recent Posts

Analyze PCAP Files using Malcolm Network Traffic Analysis tool
August 4, 2022

Install Malcolm Network Traffic Analysis Tool on Ubuntu 22.04
August 3, 2022

The reasons why spy apps are so popular
July 30, 2022

Easy Way to Extend KVM Virtual Machine Disk Size
July 30, 2022

Monitor Changes to Critical Files on Windows Systems using Wazuh and ELK
July 30, 2022

Create Locally Trusted SSL Certificates with mkcert on Ubuntu 18.04

Installing mkcert on Ubuntu 18.04

As a prerequisite, you are required to install **certutil**, a command-line utility that can create and modify certificate and key databases before you can install mkcert utility.

```
sudo apt install libnss3-tools -y
```

Once the installation of certutil is done, download the current version of mkcert pre-built binary from Github releases page.

As of this writing, the current version of mkcert is v1.4.3

So download the current version and install it as shown below

Categories

Select Category

```
wget https://github.com/FiloSottile/mkcert/releases/download/v1.4.3/mkcert-v1.4.3-linux-amd64
```

```
sudo cp mkcert-v1.4.3-linux-amd64 /usr/local/bin/mkcert
```

```
sudo chmod +x /usr/local/bin/mkcert
```


Generate Local CA on Ubuntu 18.04

Now that the mkcert utility is installed, run the command below to generate your local CA.

```
mkcert -install
```

```
 The local CA is now installed in the system trust store! ⚡
 The local CA is now installed in the Firefox and/or Chrome/Chromium
 trust store (requires browser restart)! 🦊
```


The root CA is stored under **#HOME/.local/share/mkcert**.

You can print the location directory of the root CA path by running the command below.

```
mkcert -CAROOT
```

```
/home/amos/.local/share/mkcert
```

If you encounter the error:

```
 ERROR: no Firefox and/or Chrome/Chromium security databases found
```

Just launch the browsers and re-run the install command.

Create Locally Trusted SSL Certificates with mkcert on Ubuntu 18.04

Now that you have your local CA, run the command below to generate local SSL certificates using mkcert command.

```
mkcert kifarunix-demo.com '*.kifarunix-demo.com' localhost 127.0.0.1 ::1
```

```
Sample command output;
```

```
 Created a new certificate valid for the following names 📜
  - "kifarunix-demo.com"
  - "*.kifarunix-demo.com"
  - "localhost"
  - "127.0.0.1"
  - "::1"

 Reminder: X.509 wildcards only go one level deep, so this won't
 match a.b.kifarunix-demo.com ℹ️

 The certificate is at "./kifarunix-demo.com+4.pem" and the key at
 "./kifarunix-demo.com+4-key.pem" ✅

 It will expire on 31 August 2023 🗓
```

You have the certificate and key in the current working directory;

```
ls -1 ./kifarunix-demo.com+*
```

```
  ./kifarunix-demo.com+4-key.pem
  ./kifarunix-demo.com+4.pem
```

Enable Web Server HTTPS using the Certificates

The certificates are now installed and it is time to enable your webserver to use them for HTTPS connections.

To configure Apache to use these certificates, edit the default ssl configuration file, **/etc/apache2/sites-available/default-ssl.conf** and change the SSL certificate and key file to point to the locally generated cert and key file above.

See the example below. Note the certificates are in my home directory.

Be sure to replace the paths accordingly.

```
sudo sed -i 's#/etc/ssl/certs/ssl-cert-snakeoil.pem#/home/koromicha/kifarunix-demo.com+4.pem#; s#/etc/ssl/private/ssl-cert-snakeoil.key#/home/koromicha/kifarunix-demo.com+4-key.pem#' /etc/apache2/sites-available/default-ssl.conf
```

To verify this;

```
grep -E "SSLCertificateFile|SSLCertificateKeyFile" /etc/apache2/sites-available/default-ssl.conf
```

```
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      /home/koromicha/kifarunix-demo.com+4.pem
                SSLCertificateKeyFile /home/koromicha/kifarunix-demo.com+4-key.pem
                #   the referenced file can be the same as SSLCertificateFile
```

Enable Apache to use SSL by loading the ssl modules;

```
sudo a2enmod ssl
```

```
sudo a2ensite default-ssl.conf
```

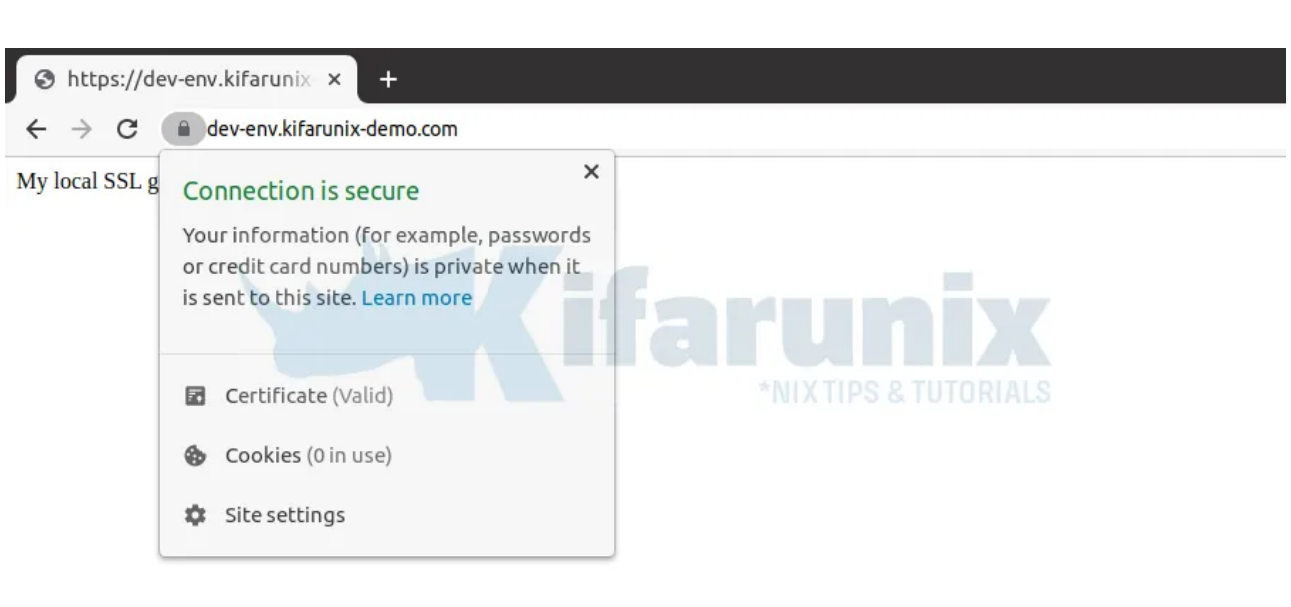
Reload and restart Apache to activate the new configuration

```
sudo systemctl restart apache2
```

Verify Local SSL Certs generated with mkcert

Navigate to the browser and try to access your domain.

I am using local hosts file for my DNS entries.



Enable the Certificates for Nginx Web Server

Create your web page configuration as shown below.

Replace the paths to the ceritificate and key accordingly

```
vim /etc/nginx/sites-available/example.com
```

```
server {
listen 80;
listen 443 ssl;

ssl on;
ssl_certificate /home/koromicha/kifarunix-demo.com+4.pem;
ssl_certificate_key /home/koromicha/kifarunix-demo.com+4-key.pem;

server_name example.com;
location / {
root /var/www/html/example;
index index.html;
}
}
```

Verify that the configuration has no error.

```
nginx -t
```

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Restart Nginx

```
systemctl restart nginx
```

Navigate to the browser and test your ssl for your domain.

Well, seems up-to that far everything is fine.

And that concludes our guide on how to create locally trusted SSL certificates with mkcert on Ubuntu 18.04.

More `mkcert` usage information.

```
mkcert --help
```

Other Tutorials

Configure Nginx with SSL/TLS certificates on CentOS 8

Monitor SSL/TLS Certificate Expiry with Prometheus and Grafana

Configure Apache with SSL/TLS Certificates on CentOS 8

TAGS create ssl with mkcert Encryption HTTPS locally trusted ssl with mkcert mkcert SSL

Share

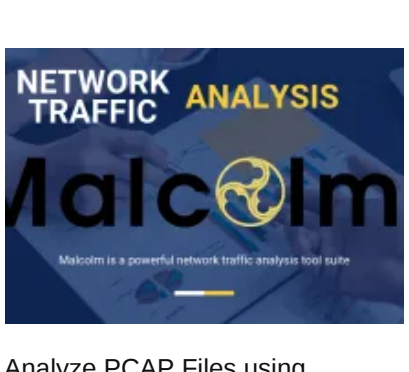
Previous article
How to Automate eCryptfs Mounting Procedure

Next article
How to Install and Use ClamAV Antivirus on Ubuntu 18.04

koromicha

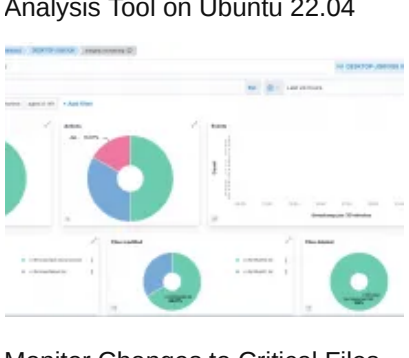
I am the Co-founder of Kifarunix.com, Linux and the whole FOSS enthusiast, Linux System Admin and a Blue Teamer who loves to share technological tips and hacks with others as a way of sharing knowledge as: "In vain have you acquired knowledge if you have not imparted it to others".

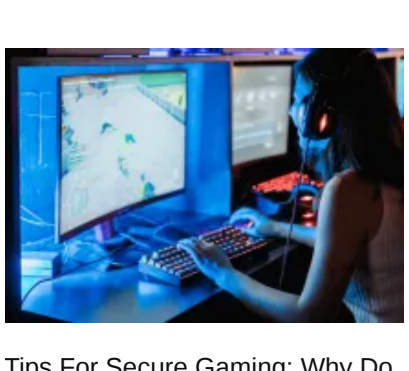
RELATED ARTICLES MORE FROM AUTHOR



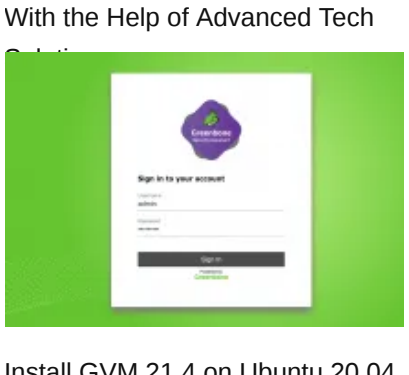
Analyze PCAP Files using Malcolm Network Traffic Analysis tool

Install Malcolm Network Traffic Analysis Tool on Ubuntu 22.04


Monitor Changes to Critical Files on Windows Systems using Wazuh and ELK


Tips For Secure Gaming: Why Do You Need a VPN While Playing?


How to Increase Your Security With the Help of Advanced Tech Solutions


Install GVM 21.4 on Ubuntu 20.04

2 COMMENTS

Pavlo May 26, 2021 At 14:31

Thanks for the article.

I successfully walked through all steps, but my browser writes "Certificate invalid".

I use Chrome 90 on Ubuntu 20.04

Reply

koromicha May 31, 2021 At 23:46

Sounds like you have not installed the local CA. Check the guide create ssl certs with mkcert ubuntu 20.04.

Reply

LEAVE A REPLY

Comment:

Name:*

Email:*

Website:

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

Post Comment