superuser

How do you add a certificate authority (CA) to Ubuntu?

Asked 9 years, 1 month ago Active 1 year, 5 months ago Viewed 603k times

239

122

My work has decided to issue their own certificate authority (CA) to handle different aspects of our work securely without paying for certificates.

- Cryptographically sign emails
- Encrypt email contents
- Make access to things like the company IRC client-certificate based.
- Revoke the keys of former employees automatically

They sent me a `.pem` file, and I'm not sure how to add it to my Ubuntu install. The instructions sent were: "*Double-clicking on it on a Mac should install it.*"

How do I proceed? Do I need to do something with OpenSSL to create a `.key`, `.csr`, or `.crt` file?

ubuntu certificate trusted-root-certificates

Share Improve this question Follow

edited Oct 9 '15 at 12:08
ks1322
403 ● 4 ● 19

asked Jun 15 '12 at 16:14
Xeoncross
3,934 ● 8 ● 32 ● 47

7 the comment "The instructions sent were: \"Double-clicking on it on a Mac should install it.\"" made my day – mzoll Sep 19 '19 at 9:43

@mzoll and the way of escaping those quotes – HosseyNJF Sep 25 '20 at 10:39

8 Answers

Active Oldest Votes

319

Installing a CA

Copy your certificate in PEM format (the format that has `----BEGIN CERTIFICATE----` in it) into `/usr/local/share/ca-certificates` and name it with a `.crt` file extension.

Then run `sudo update-ca-certificates`.

Caveats: This installation only affects products that use this certificate store. Some products may use other certificate stores; if you use those products, you'll need to add this CA certificate to those other certificate stores, too. (Firefox Instructions, Chrome Instructions, Java Instructions)

Testing The CA

You can verify if this worked by looking for the certificate that you just added in `/etc/ssl/certs/ca-certificates.crt` (which is just a long list of all of your trusted CA's concatenated together).

You can also use OpenSSL's s_client by trying to connect to a server that you know is using a certificate signed by the CA that you just installed.

```
$ openssl s_client -connect foo.whatever.com:443 -CApath /etc/ssl/certs

CONNECTED(00000003)
depth=1 C = US, ST = Virginia, O = "Whatever, Inc.", CN = whatever.com, emailAddress =
admin@whatever.com
verify return:1
depth=0 C = US, ST = Virginia, L = Arlington, O = "Whatever, Inc.", CN =
foo.whatever.com
verify return:1
---
Certificate chain
 0 s:/C=US/ST=Virginia/L=Arlington/O=Whatever, Inc./CN=foo.whatever.com
   i:/C=US/ST=Virginia/O=Whatever, Inc./CN=whatever.com/emailAddress=admin@whatever.com

... snip lots of output ...

    Key-Arg   : None
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1392837700
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
```

The first thing to look for is the certificate chain near the top of the output. This should show the

The first thing to look for is the certificate chain near the top of the output. This should show the CA as the issuer (next to `i:`). This tells you that the server is presenting a certificate signed by the CA you're installing.

Second, look for the `verify return code` at the end to be set to `0 (ok)`.

Share Improve this answer Follow

edited Jun 12 '20 at 13:48
Community ♦
1

answered Feb 19 '14 at 19:13
Mark E. Haase
3,923 ● 1 ● 16 ● 17

4 this one actually works – Sabareesh Kkanan Aug 4 '15 at 20:14

2 Thanks for noting that firefox / chrome do not use the default cert store. – Tim Strijdhorst Dec 10 '15 at 16:26

7 Note that update-ca-certificates can be very finicky (probably by design). mycert.pem.crt did NOT work, but mycert.crt did. I also think that it needs to be /usr/local/share/ca-certificates, not /usr/share/ca-certificates (despite what comments said in the /etc/ca-certificates.conf). – labyrinth Dec 15 '15 at 17:39

2 Thanks for the `crt` extension comment, that was the secret to getting this work for me, I was given a cert with a `cert` extension and was confused as to why nothing was working. – Ransom Briggs Mar 29 '16 at 16:31

3 One caveat: `s_client` doesn't send SNI by default and the server may need SNI especially if it supports virtual hosts/sites with different certs; for this case add `-servername foo.whatever.com`. Or if it's a *web* server use (modern versions of) `curl` or `wget` which do SNI automatically. – dave_thompson_085 May 14 '16 at 3:40

83

man update-ca-certificates:

```
update-ca-certificates  is a program that updates the directory /etc/ssl/certs to hold
SSL
certificates  and  generates  ca-certificates.crt,  a  concatenated  single-file  list
of
certificates.

It  reads  the  file  /etc/ca-certificates.conf.  Each  line  gives  a  pathname  of  a
CA
certificate under /usr/share/ca-certificates that should be  trusted.   Lines  that
begin
with  "#"  are  comment lines and thus ignored.  Lines that begin with "!" are
deselected,
causing the deactivation of the CA certificate in question. Certificates must have a
.crt
extension in order to be included by update-ca-certificates.

Furthermore  all  certificates  with  a  .crt  extension  found below
/usr/local/share/ca-
certificates are also included as implicitly trusted.
```

From the above, I would infer that the preferred way to get local certificate files into the trusted store is to put them into `/usr/local/share/ca-certificates`, and then run `update-ca-certificates`. You do not need to touch `/etc/ssl/certs` directly.

Share Improve this answer Follow

edited Feb 14 '20 at 13:22
0xC0000022L
5,682 ● 9 ● 41 ● 77

answered Jun 15 '12 at 18:07
Steven Monday
1,615 ● 14 ● 15

30 Naming the certificates with .crt extensions seemed to be required as well. – treat your mods well Mar 5 '13 at 23:12

Thanks for the note @phyzome -- would not have been able to add my cert otherwise. – Seiyria Mar 17 '15 at 14:03

7 I had to add `--fresh` to get it to work. e.g. `update-ca-certificates --fresh` – Elijah Lynn Jun 25 '19 at 4:13

21

The other answers regarding `update-ca-certificates` are correct for applications that read from the system certificate store. For Chrome and Firefox, and probably some others, the certificate must be put in the nssdb, the backend for the Mozilla NSS library.

From https://code.google.com/p/chromium/wiki/LinuxCertManagement:

> For example, to trust a root CA certificate for issuing SSL server certificates, use
>
> certutil -d sql:$HOME/.pki/nssdb -A -t "C,," -n <certificate nickname> -i <certificate filename>

Where `<certificate nickname>` is arbitrary, and `<certificate filename>` is your .pem or .crt file.

Other helpful references:

- General description: https://wiki.archlinux.org/index.php/Network_Security_Services
- `certutil` man page, describing the parameters used above: https://developer.mozilla.org/en-US/docs/NSS_reference/NSS_tools_:_certutil

Share Improve this answer Follow

edited Jun 12 '20 at 13:48
Community ♦
1

answered Oct 10 '13 at 18:46
Johann
541 ● 7 ● 16

1 thanks. It works on Ubuntu 16.04 for Chrome 53.0.2785.143, but Firefox 49 seems to have separate store db and must be added from about:preferences#advanced [View Certiticates] -> [Authorities] -> [Import] More about firefox cert store. askubuntu.com/a/248326/535154 – mauron85 Oct 12 '16 at 12:31

By the way, if you want to install cert *before* first run of Chrome (i.e. while .pki/ dir is still missing), you must first create the nssdb: `mkdir -p $HOME/.pki/nssdb && chmod -R 0700 $HOME/.pki && certutil -d sql:$HOME/.pki/nssdb -N --empty-password` – akavel Dec 15 '16 at 16:21

2 There is a way to get Chrome and Firefox to read from the system certificate store. See my answer: superuser.com/a/1312419/506107 – wheeler Apr 10 '18 at 1:05

superuser.com/a/1312419/366107 – wheeler Apr 10 '18 at 1:05

This is fantastic, thank you. Can now use Slack and Teams Preview behind Corporate SSL Decrypt flawlessly. – Bevan Jan 21 '20 at 23:01

1 Firefox allows you to configure security modules... it does not use the same one chrome uses by default... you can add it by loading `/usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so` to the list of security modules under firefox Certificate settings. Super easy once you know. – Ray Foss Sep 16 '20 at 17:06

16

I had same issue, and I had to copy the `.pem` file to `/usr/local/share/ca-certificates`, renaming it as `.crt`. The `.cer` file can easily be converted to `.pem`, with openssl, for example, if you don't have the `.pem`.

After copying the file you must execute `sudo update-ca-certificates`.

Share Improve this answer Follow

answered Apr 30 '14 at 13:39
greuze
303 ●2 ●5

1 `openssl x509 -inform DER -in certificate.cer -out certificate.crt` – webwurst Feb 26 '18 at 11:37

13

For newer builds based on Debian, you may need to run:

```
sudo dpkg-reconfigure ca-certificates
```

NOTE: sudo dpkg-reconfigure ca-certificates calls update-ca-certificates internally

You'll of course still need to copy the certificate (.crt file) to /usr/share/ca-certificates before you do any of this :)

Share Improve this answer Follow

answered Sep 2 '15 at 6:19
missmah
131 ●1 ●2

8

Building on dwmw2's answer, you can actually tell applications that use NSS for its certificate management to use the system trust store.

`libnss3` by default ships with a read-only set of root CA certificates (`libnssckbi.so`), so most of the time you need to manually add them yourself to the local user trust store located in `$HOME/.pki/nssdb`. `p11-kit` offers a drop-in replacement for `libnssckbi.so` that acts as an adapter to the system-wide root certificates installed in `/etc/ssl/certs`.

Edit:

There seem to be more versions of `libnssckbi.so` out there than just in `libnss3`. The following is a script to find them all, back them up, and replace them with links to `p11-kit`:

```
sudo apt-get update && sudo apt-get install -y p11-kit libnss3
find / -type f -name "libnssckbi.so" 2>/dev/null | while read line; do
    sudo mv $line ${line}.bak
    sudo ln -s /usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so $line
done
```

Original instructions:

To do this, install `p11-kit` and `libnss3` (if they are not already instealled):

```
sudo apt-get update && sudo apt-get install -y p11-kit libnss3
```

Then backup the existing `libnssckbi.so` provided by `libnss3`:

```
sudo mv /usr/lib/x86_64-linux-gnu/nss/libnssckbi.so /usr/lib/x86_64-linux-
gnu/nss/libnssckbi.so.bak
```

Finally, create the symbolic link:

```
sudo ln -s /usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so /usr/lib/x86_64-linux-
gnu/nss/libnssckbi.so
```

To confirm that it worked, you can run `ll /usr/lib/x86_64-linux-gnu/nss/libnssckbi.so` and it should show the link:

```
lrwxrwxrwx 1 root root 49 Apr  9 20:28 /usr/lib/x86_64-linux-gnu/nss/libnssckbi.so ->
/usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so
```

Now, if you add a certificate to the CA store using `update-ca-certificates`, those certificates will now be available to applications using NSS (`libnss3`) such as Chrome.

Share Improve this answer Follow edited Apr 10 '18 at 1:45 answered Apr 10 '18 at 1:00

wheeler
231 ●3 ●7

I've been fighting Ubuntu 18.04 to try and get this to work for the past 3 days and it won't work for whatever reason. I link the p11-kit-trust.so to the libnssckbi.so but when I do that there are *no* certificates at all any longer. Any website I go to thats https enabled (which is basically all of them) prompt that there is a security issue. Is there something obvious I'm missing? – Kevin Vasko Oct 25 '19 at 21:34

This is brilliant and worked a charm for Firefox anyhow. The script suggested is dangerous mind you. I'd suggest trying the find alone first on your system to see what it finds. I have a huge systems and it turns up stuff in timeshift backups for exmple and I also have a few sshfs mounts at any one time depending on what I'm working on and it trundles off onto those as well. I'd use `locate` in any case to find them and patch them one by one or write a script that loops through specified files only. – Bernd Wechner Aug 20 '20 at 12:19

@KevinVasko I'm on 20.04 and no trouble. Only tried Firefox but it worked fine. I take notes when I do things like this and keep them, and noted for myself, that these are .so files, so binary object files, and to be sure Firefox is using the new on you have to shut down Firefox and restarted it. But you are alas, not very clear what you mean by "there are no certificates at all" - in what context? In `/etc/ssl/certs` ? – Bernd Wechner Aug 20 '20 at 12:24

4

As noted, various applications using NSS have their own certificate store. As things stand on Ubuntu, you have to manually use `certutil` to add your CAs for each application, for each user.

In other distributions like Fedora, this kind of thing Just Works™ and you should file a bug against any applications which doesn't automatically trust the CAs you install with `update-ca-trust`.

You can fix this in Ubuntu too by installing the `p11-kit-modules` package and then replacing the NSS *built-in trust roots* module with `p11-kit-trust.so`, by making a symbolic link for example from `/usr/lib/firefox/libnssckbi.so` to `/usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so`

Then you *will* get the system's configured trust roots, not some hard-coded ones. Note that Ubuntu ships multiple *different* copies of that libnssckbi.so library with the hard-coded trust roots, and you have to replace all of them!

cf. https://bugs.launchpad.net/ubuntu/+source/nss/+bug/1647285

Share Improve this answer Follow

edited Apr 25 '18 at 10:11
C2H5OH
123 ● 7

answered Dec 12 '16 at 12:36
dwmw2
191 ● 1 ● 2

When I did `sudo find / -type f -name "libnssckbi.so"`, It found `libnssckbi.so` in three places: `/usr/lib/thunderbird/`, `/usr/lib/firefox/`, and `/usr/lib/x86_64-linux-gnu/nss/`. So you are saying that I should link the `libnssckbi.so` in all three of those folders to `p11-kit-trust.so` ? – wheeler Apr 10 '18 at 0:06

1 Okay, just confirmed that linking `/usr/lib/x86_64-linux-gnu/nss/libnssckbi.so` -> `/usr/lib/x86_64-linux-gnu/pkcs11/p11-kit-trust.so` worked like a CHARM. I was able to add a certificate into `/usr/local/share/ca-certificates`, run `sudo update-ca-certificates`, and PRESTO, Chrome started to accept the self-signed certificates. – wheeler Apr 10 '18 at 0:32

@dwmw2 I've been fighting Ubuntu 18.04 to try and get this to work for the past 3 days and it won't work for whatever reason. I link the p11-kit-trust.so to the libnssckbi.so but when I do that there are no certificates at all any longer. Any website I go to thats https enabled (which is basically all of them) prompt that there is a security issue. Is there something obvious I'm missing? – Kevin Vasko Oct 25 '19 at 21:44

Seriously stupid answer to add here, but I had spent 2 hours going back and forth with certutils in linux... I was sure everything was correct:

1

```
hutber@hutber-mint /var/www/asos-mvt-framework $ certutil -L -d sql:${HOME}/.pki/nssdb

Certificate Nickname                                         Trust Attributes
                                                             SSL,S/MIME,JAR/XPI

anyproxy                                                     CT,,
rootCA                                                       CT,,
myasos                                                       CT,,
```

But still, in chrome nothing was working. I tried everything, in the end....

`Restarting Chrome`

Was the key to my success after following: Steven Monday's advice

Share Improve this answer Follow

answered Jan 2 '18 at 12:58

Jamie Hutber

345 ● 2 ● 9 ● 23

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.