# How to get .pem file from .key and .crt files?

Asked 11 years, 4 months ago  Active 7 months ago  Viewed 1.1m times

620

228

How can I create a PEM file from an SSL certificate?

These are the files that I have available:

- `.crt`
- `server.csr`
- `server.key`

ssl openssl pem asn.1 der

edited Oct 11 '17 at 14:48

sdgfsdh
**21.2k** 13 76 156

asked Jun 13 '09 at 23:23

Sergio Rodriguez
**6,566** 3 15 19

2 Other way round: Convert `.pem` to `.crt` and `.key` . – kenorb May 17 '19 at 11:07

## 10 Answers

Active | Oldest | Votes

834

Your keys may already be in PEM format, but just named with .crt or .key.

**If the file's content begins with `-----BEGIN` and you can read it in a text editor:**

The file uses base64, which is readable in ASCII, not binary format. The certificate is already in PEM format. Just change the extension to .pem.

**If the file is in binary:**

For the server.crt, you would use

```
openssl x509 -inform DER -outform PEM -in server.crt -out server.crt.pem
```

For server.key, use `openssl rsa` in place of `openssl x509` .

The server.key is likely your private key, and the .crt file is the returned, signed, x509 certificate.

**If this is for a Web server and you cannot specify loading a separate private and public key:**

```
cat server.crt server.key > server.includesprivatekey.pem
```

I would recommend naming files with "includesprivatekey" to help you manage the permissions you keep with this file.

edited Nov 11 '17 at 12:22

Stevoisiak
**14.3k** 16 91 155

answered Jun 13 '09 at 23:31

maxwellb
**11.3k** 2 21 32

2 Check the format of the server.key. I only assumed it was RSA. But reading the first line of the file will probably tell you that. – maxwellb Jun 13 '09 at 23:32

11 Just a heads up that `cat server.crt server.key > server.pem` won't place the open comment on its own line, which seems to be a requirement. Courier mail gave me hell and it took me hours to figure out what was going wrong. – Graham Walters Feb 12 '14 at 1:36

1 Thanks Graham. Different tools will generate the files differently, and ultimately, some verification is good. When I performed these steps, the files ended with a newline, for example. – maxwellb Feb 19 '14 at 15:25

The tip about concatenating the .crt and .key files together was very helpful. I wanted to use my certificate with stunnel3, but it didn't have a way to specify the key file. Using the concatenation worked. (Actually, since stunnel3 is a Perl program, I added an option to it myself for reading the key file. However, since I saw later the concatenation worked, I've reverted stunnel3 to its original code.) – Mr. Lance E Sloan Jun 25 '14 at 17:31

3 Just a bump to say that cat `server.crt server.key > server.includesprivatekey.pem` is useful for SSL with haproxy 1.5. – jimm101 Sep 18 '14 at 17:49

229

I needed to do this for an AWS ELB. After getting beaten up by the dialog many times, finally this is what worked for me:

```
openssl rsa -in server.key -text > private.pem
openssl x509 -inform PEM -in server.crt > public.pem
```

Thanks NCZ

Edit: As @floatingrock says

With AWS, don't forget to prepend the filename with `file://` . So it'll look like:

```
 aws iam upload-server-certificate --server-certificate-name blah --certificate-
body file://path/to/server.crt --private-key file://path/to/private.key --path
/cloudfront/static/
```

http://docs.aws.amazon.com/cli/latest/reference/iam/upload-server-certificate.html

edited Mar 23 '16 at 16:57

answered Aug 30 '12 at 17:30

slf
**22k** 10 70 98

`server-certificate --server-certificate-name blah --certificate-body file://~/Desktop/server.crt --private-key file://~/Desktop/private.key --path /cloudfront/static/` – FloatingRock Nov 6 '14 at 4:03

1 The second command does nothing if your input is a pem file so assuming it is, you only need the first command – Kristofer Jan 19 '15 at 11:48

Awesome! This is worked well for me! Just converted my .key and .crt files to .pem using your lines of code and then I uploaded (copy/paste) via AWS console. Thanks! – Diego D Jun 30 '16 at 10:52

82

A `pem` file contains the certificate and the private key. It depends on the format your certificate/key are in, but probably it's as simple as this:

```
cat server.crt server.key > server.pem
```

edited Nov 15 '11 at 18:47

answered Jun 13 '09 at 23:30
sth
**192k** 49 259 350

i keep getting `-bash: server.key.pem: Permission denied` – t q Mar 25 '14 at 0:42

2 @tq: That means you aren't allowed to read or write that file. – sth Mar 25 '14 at 1:29

thank you @sth will look for permission. but i was doing this with sudo commands – t q Mar 25 '14 at 2:37

8 @tq: cat server.crt server.key | sudo tee server.pem – dimir Aug 20 '14 at 13:47

2 watchout for missing newlines our your pem file might have garbled lines like -----END CERTIFICATE----------BEGIN CERTIFICATE----- – Wolfgang Fahl Jan 5 '16 at 16:27

24

Additionally, if you don't want it to ask for a passphrase, then need to run the following command:

```
openssl rsa -in server.key -out server.key
```

edited Nov 15 '11 at 18:48
sth
**192k** 49 259 350

answered Oct 21 '10 at 8:35
rahul
**261** 2 3

9 If you want a file starting with `-----BEGIN RSA PRIVATE KEY-----` and have one that starts with `----BEGIN ENCRYPTED PRIVATE KEY-----` , this is the command you want to use. – Philippe Gerber Feb 16 '13 at 12:50

18

this is the best option to create .pem file

```
openssl pkcs12 -in MyPushApp.p12 -out MyPushApp.pem -nodes -clcerts
```

edited Jan 25 '15 at 16:38

karthikr
**83.8k** 20 174 172

answered May 23 '14 at 21:28

GOrozco58
**1,034** 12 9

4

What I have observed is: if you use openssl to generate certificates, it captures both the text part and the base64 certificate part in the crt file. The strict pem format says (wiki definition) that the file should start and end with BEGIN and END.

> .pem – (Privacy Enhanced Mail) Base64 encoded DER certificate, enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----"

So for some libraries (I encountered this in java) that expect strict pem format, the generated crt would fail the validation as an 'invalid pem format'.

Even if you copy or grep the lines with BEGIN/END CERTIFICATE, and paste it in a cert.pem file, it should work.

Here is what I do, not very clean, but works for me, basically it filters the text starting from BEGIN line:

```
grep -A 1000 BEGIN cert.crt > cert.pem
```

answered Feb 14 '13 at 8:52

adityalad
**121** 4

1 another option is simply to pass the non-strict certificate through `openssl x509` . It will output a valid PEM certificate: `cat certificate.crt | openssl x509 > certificate.pem` – T0xicCode Jul 17 '13 at 15:47

If you want to get everything from "BEGIN" to the end of the file, that's a job for sed. Specifically, you want something like `sed -n '/--BEGIN/,$p' cert.crt` in this case. To explain that: the "-n" tells sed to not print anything by default, and then the range expression `/--BEGIN/,$` makes the `p` command (print) apply to lines between the first line which contains `--BEGIN` and the end of the file ( `$` ). – dannysauer Apr 24 '17 at 17:13

4

I was trying to go from godaddy to app engine. What did the trick was using this line:

```
openssl req -new -newkey rsa:2048 -nodes -keyout name.unencrypted.priv.key -out
name.csr
```

Exactly as is, but replacing name with my domain name (not that it really even mattered)

And I answered all the questions pertaining to common name / organization as www.name.com

Then I opened the csr, copied it, pasted it in go daddy, then downloaded it, unzipped it, navigated to the unzipped folder with the terminal and entered:

```
cat otherfilegodaddygivesyou.crt gd_bundle-g2-g1.crt > name.crt
```

Then I used these instructions from Trouble with Google Apps Custom Domain SSL, which were:

```
openssl rsa -in privateKey.key -text > private.pem
openssl x509 -inform PEM -in www_mydomain_com.crt > public.pem
```

exactly as is, except instead of privateKey.key I used name.unencrypted.priv.key, and instead of www_mydomain_com.crt, I used name.crt

Then I uploaded the public.pem to the admin console for the "PEM encoded X.509 certificate", and uploaded the private.pem for the "Unencrypted PEM encoded RSA private key"..

.. And that finally worked.

edited May 23 '17 at 11:47
Community ♦
1 1

answered Aug 9 '15 at 15:35
user2738183

4

Trying to upload a GoDaddy certificate to AWS I failed several times, but in the end it was pretty simple. No need to convert anything to .pem. You just have to be sure to include the GoDaddy bundle certificate in the chain parameter, e.g.

```
aws iam upload-server-certificate
    --server-certificate-name mycert
    --certificate-body file://try2/40271b1b25236fd1.crt
    --private-key file://server.key
    --path /cloudfront/production/
    --certificate-chain file://try2/gdig2_bundle.crt
```

And to delete your previous failed upload you can do

```
aws iam delete-server-certificate --server-certificate-name mypreviouscert
```

answered Jun 7 '16 at 8:32

skensell
1,317 11 19

This did not work for me `An error occurred (MalformedCertificate) when calling the UploadServerCertificate operation: Unable to parse certificate. Please ensure the certificate is in PEM format.` – Adam Raudonis Sep 1 '18 at 20:19

4

2. Export certificate  from Key chain and  give name (Certificates.p12),
3. Open terminal and goto folder where you save above Certificates.p12 file,
4. Run below commands:

   a) `openssl pkcs12 -in Certificates.p12 -out CertificateName.pem -nodes ,`

   b) `openssl pkcs12 -in Certificates.p12 -out pushcert.pem -nodes -clcerts`

5. Your .pem file ready "pushcert.pem".

edited Jun 28 '17 at 12:05

answered Apr 24 '14 at 6:54

Ashvin Ajadiya

**6,233** 2 23 43

---

1

- Open terminal.
- Go to the folder where your certificate is located.
- Execute below command by replacing name with your certificate.

  openssl pkcs12 -in YOUR_CERTIFICATE.p12 -out YOUR_CERTIFICATE.pem -nodes -clcerts

- Hope it will work!!

answered Jul 27 '17 at 9:40

Dhaval H. Nena

**3,348** 1 26 43

**Highly active question**. Earn 10 reputation in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.