```
openssl s_client -showcerts -connect google.com:443 </dev/null > ./certificate.txt

sudo certutil -d sql:$HOME/.pki/nssdb -L

sudo certutil -d sql:$HOME/.pki/nssdb -A -t "TRUSTARGS" -n certificate -i ./certificate.pem
sudo certutil -d sql:$HOME/.pki/nssdb -A -t "TRUSTARGS" -n certificate -i
./certificate_ROOT_CA.pem
sudo certutil -d sql:$HOME/.pki/nssdb -A -t "TRUSTARGS" -n certificate -i
./certificate_SUB_CA.pem

$ sudo update-ca-certificates
# или
$ sudo update-ca-trust

sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate.pem" -d sql:
$HOME/.pki/nssdb
sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate_ROOT_CA.pem" -d sql:
$HOME/.pki/nssdb
sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate_SUB_CA.pem" -d sql:
$HOME/.pki/nssdb




#!/bin/bash

certname="$1"

for certDB in $(find ~/ -name "cert9.db"); do
        certdir=$(dirname ${certDB});
        for cert_files in ./*.pem; do
                certutil -A -n "${certname}" -t "TCu,Cu,Tu" -i "${cert_files}" -d sql:${certdir}
        done
done



#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import re
```

```
import pathlib

def main():
	cert_file = './certificate.txt'
	with open(cert_file, "r") as f:
		lines = f.readlines()
	onsearch1 = 's:C'
	onsearch2 = 'CN = '
	onsearch3 = 'BEGIN CERTIFICATE'
	onsearch4 = 'END CERTIFICATE'
	fname = ''
	outname = ''
	startname = True
	startcert = False
	endcert = False
	iswrite = False
	for item in lines:
		if onsearch1 in item and startname:
			onstart = re.search(onsearch2, item).end()
			fname = item[onstart:].replace('*','_')
			startname = False
			outname = str(fname  +
'.pem').replace(onsearch2,'').replace('*','_').replace(' ','_').replace('\n','')
			outname = pathlib.Path(outname.replace(' ','_')).resolve()
			iswrite = True
		elif onsearch1 in item:
			onstart = re.search(onsearch2, item).end()
			outname = str(fname + ' ' + item[onstart:]  +
'.pem').replace(onsearch2,'').replace('*','_').replace(' ','_').replace('\n','')
			outname = pathlib.Path(outname).resolve()
			iswrite = True
		if iswrite:
			with open(outname, 'a') as f:
				if onsearch3 in item:
					startcert = True
				if onsearch4 in item:
					startcert = False
					endcert = True
				if startcert:
					f.write(item)
				if endcert:
```

```
                                endcert = False
                                f.write(item)

if __name__ == '__main__':
        main()
```

#bash #Linux #Windows #Certificates #Cert #certDB #openssl #GitBash

Не безопасный веб-сайт - делаем доверенным.

Будь у вас Linux или Windows - вы столкнулись с проблемой - при открытии Российского веб-сайта он ругается на безопасность SSL сертификата. И чтобы вы не делали веб-сайт не становиться доверенным? Вам точно сюда!

Ну а с вами, как всегда Shadow.
Подписывайтесь на канал, ставьте лайки, комментируйте.
Всем Добра и Удачи!

Универсальное добавление сертификатов в доверенные

Итак. Будь у вас **Linux** или **Windows** - вы столкнулись с проблемой - **при открытии Российского веб-сайта** он ругается на **безопасность SSL сертификата**.

Дело в том, что большинство Российских веб-сайтов основывают свои **SSL** сертификаты по **ГОСТ**-у не заботясь о пользователях и обновлении корневых сертификатов в старых **Windows** системах и любых **Linux**-ах.

Вы конечно можете вручную скачать большинство нужных ISGR ROOT сертификатов, **но к сожалению**, только с сайта letsencrypt, т.к. официальный сайт вообще перестал работать. Что на самом деле, даже печально.

Однако, вы скорее всего можете столкнуться с той же проблемой - и у того или иного Российского веб-сайта может быть совершенно другой ROOT центр сертификации, которого нет в **letsencrypt**, хотя на нём же и основываясь. Да, и такое бывает. И сайт опять будет не безопасным.

А хотелось бы универсально добавить Российский веб-сайт в доверенные.

Такой способ конечно существует! Забегу наперёд, и даже не один.

Первое, что вам необходимо понять - это то, что у любого сайта существует не только сам **SSL-сертификат**, но и **сертификат корневого центра** и **сертификат промежуточного центра** сертификации. Вероятно вы это знали. Однако, большинство способов дают лишь скачивание только одного **SSL-сертификата самого сайта**, и не говорят как скачать сертификаты центров.

Это можно сделать 3-мя способами.

1. Воспользоваться браузером **Mozila Firefox**. При открытии страницы барузер также будет ругаться, однако, там можно посмотреть состав сертификата и главное отличие от любого другого браузера - на последней вкладке можно найти ссылки на все сертификаты центров! **Вот это уже интересно.**
2. Можно воспользоваться утилитой **openssl**, которая есть как в **Linux**-е, так и в **git-bash** на **Windows**. Здесь не всё так просто и одновременно проще не бывает.

3. Посмотреть состав сертификата и вручную поискать и скачать сертификаты центров с помощью **Google**. Этот способ вообще не катит, но вполне возможен.

Для начала и примера скачаем все сертификаты (и сайта и центров) в один файл, например сайта **Google**. Почему бы и нет.

> *$ openssl s_client -showcerts -connect google.com:443 </dev/null > ./certificate.txt*

Итак, файл **certificate.txt** есть. Надо разделить его на сертификаты сайта и сертификаты центров, иначе вы не сможете добавить их в доверенные, т.к. там содержится вся информация обо всех сертификатах. И даже если вы по инструкции других сайтов оставите только сами сертификаты последовательно друг за другом по типу pkcs7 или списков - вы всё равно не сможете добавить их в доверенные.

Либо мы можем вручную создать нужные файлы и вручную добавить получившиеся сертификаты (начиная со строк «BEGIN CERTIFICATE»и заканчивая строками «END CERTIFICATE») в разные файлы + вручную импортировать в систему.

Либо мы сделаем это автоматически - скриптами. Вот это уже интереснее.

Поэтому создадим 2 скрипта для обработки полученных сертификатов. (Python скрипт будет работать и в Windows).

Для разделения файла на сертификаты создадим скрипт со следующим кодом. пусть это будет файл **convert.py** со следующим содержимым:

```
#!/usr/bin/env python3

# -*- coding: utf-8 -*-

import re

import pathlib

def main():

cert_file = './certificate.txt'

with open(cert_file, "r") as f:

lines = f.readlines()

onsearch1 = 's:C'

onsearch2 = 'CN = '

onsearch3 = 'BEGIN CERTIFICATE'

onsearch4 = 'END CERTIFICATE'

fname = ''

outname = ''

startname = True

startcert = False

endcert = False
```

```
iswrite = False

for item in lines:

if onsearch1 in item and startname:

onstart = re.search(onsearch2, item).end()

fname = item[onstart:].replace('*','_')

startname = False

outname = str(fname + '.pem').replace(onsearch2,'').replace('*','_').replace(' ','_').replace('\n','')

outname = pathlib.Path(outname.replace(' ','_')).resolve()

iswrite = True

elif onsearch1 in item:

onstart = re.search(onsearch2, item).end()

outname = str(fname + ' ' + item[onstart:] + '.pem').replace(onsearch2,'').replace('*','_').replace(' ','_').replace('\n','')

outname = pathlib.Path(outname).resolve()

iswrite = True

if iswrite:

with open(outname, 'a') as f:

if onsearch3 in item:

startcert = True
```

if onsearch4 in item:

startcert = False

endcert = True

if startcert:

f.write(item)

if endcert:

endcert = False

f.write(item)

if __name__ == '__main__':

main()

Отступы проставьте вручную, но я постараюсь прикрепить оба файла скрипта.

Естественно мы не просто так сохраняли в файл **certificate.txt**. Просто, чтобы скрипту было проще - если вам нужно модифицируйте с добавлением аргументов командной строки. Итак, запускаем.

> *$ python ./convert.py*

У вас должно появится 3 новых файла. В случае примера с google - названия начинаются с **_.google.com**. Вот это название и нужно скопировать, оно далее нам понадобиться.

Ручной импорт корневых сертификатов в **Windows** не рассматривается.

Далее вы можете воспользоваться универсальной инструкцией добавления корневых сертификатов в разные браузеры вручную, что для **Windows**, что для **Linux** инструкция будет одинаковой - предоставляется от **Ubuntu. НО для каждого браузера раздельно.**

Либо, **конкретно для Linux**, вы можете пойти по пути наименьшего сопротивления и добавить все сертификаты разом во все базы данных всех браузеров и всех **NSS** баз.

Для последнего воспользуемся небольшим скриптом. Назовём его **CAtoCert9.sh**.

```
#!/bin/bash

certname="$1"

for certDB in $(find ~/ -name "cert9.db"); do

certdir=$(dirname ${certDB});

for cert_files in ./*.pem; do

certutil -A -n "${certname}" -t "TCu,Cu,Tu" -i "${cert_files}" -d sql:${certdir}

done

done
```

Запускаем и не забываем добавить правильное название + обновляем список сертификатов системы:

> *$ chmod +x ./CAtoCert9.sh*
>
> *$ sudo ./CAtoCert9.sh _.google.com*
>
> *$ sudo update-ca-certificates # Для Debian или Fedora*
>
> *# или для Arch-е подобных систем*
>
> *$ sudo update-ca-trust*

Теперь необходимо проверить и удостовериться, что сертификаты добавились во все **NSS** базы:

> *sudo certutil -d sql:$HOME/.pki/nssdb -L*

Далее - на всякий случай - команда ручного добавления сертификатов. НО для одной из NSS баз сертификатов. Т.е. для каждого браузера такую базу надо найти и повторить команды для каждой.

> *sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate.pem" -d sql:$HOME/.pki/nssdb*
>
> *sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate_SUB_CA.pem" -d sql:$HOME/.pki/nssdb*
>
> *sudo certutil -A -n "certificate_name" -t "TCu,Cu,Tu" -i "certificate_ROOT_CA.pem" -d sql:$HOME/.pki/nssdb*

Опять таки, не забываем обновлять сертификаты системы.

> *$ sudo update-ca-certificates # Для Debian или Fedora*
>
> *# или для Arch-е подобных систем*

$ sudo update-ca-trust

Теперь можно запускать любой браузер и проверять, что ваш сайт является доверенным.

Ну, а сегодня на этом всё. Надеюсь я вас сегодня заинтересовал.

Всем удачи, до встречи, Пока-пока!