Локальный HTTPS

Алексей Остапенко @kbakba

Алексей Остапенко

- Telegram: @kbakba
- Twitter: @kbakba
- Mail: conf@kbakba.net

```
brew install openssl mkcert dnsmasq
```

HTTP

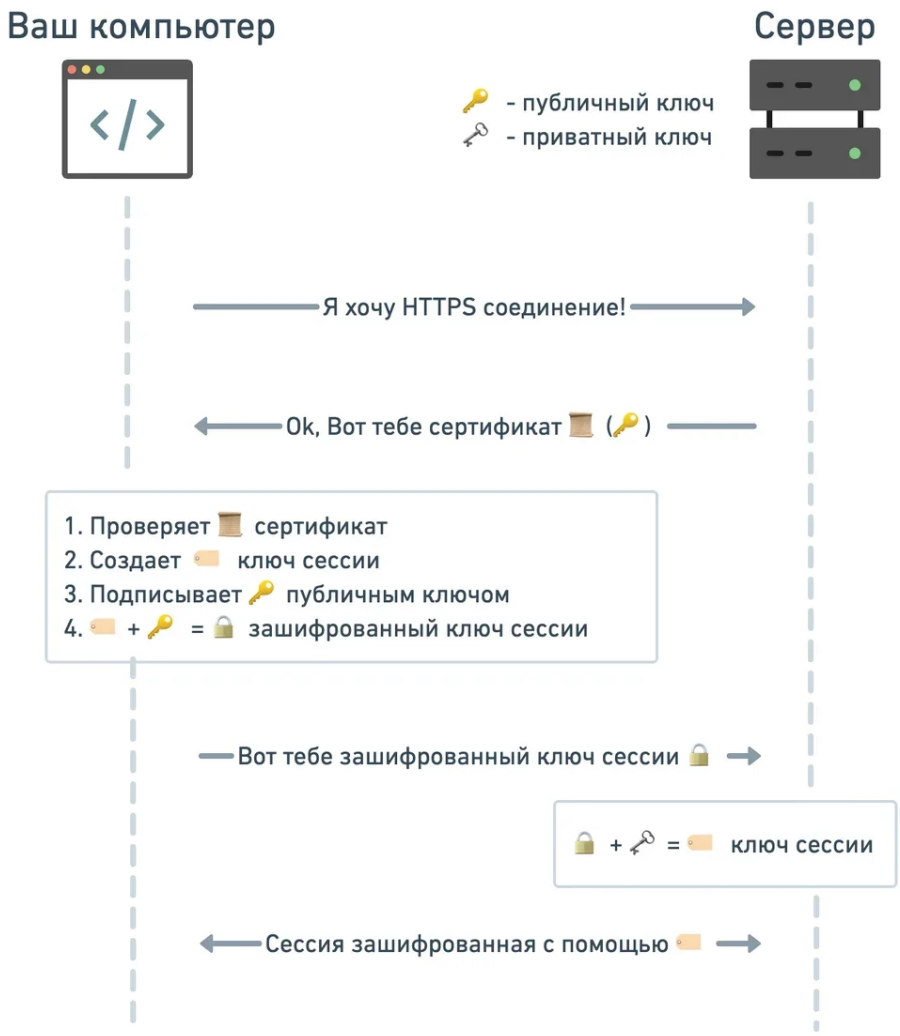
```
npm init -y
npm i -D webpack webpack-cli webpack-dev-server
npx webpack-dev-server
```

HTTPS

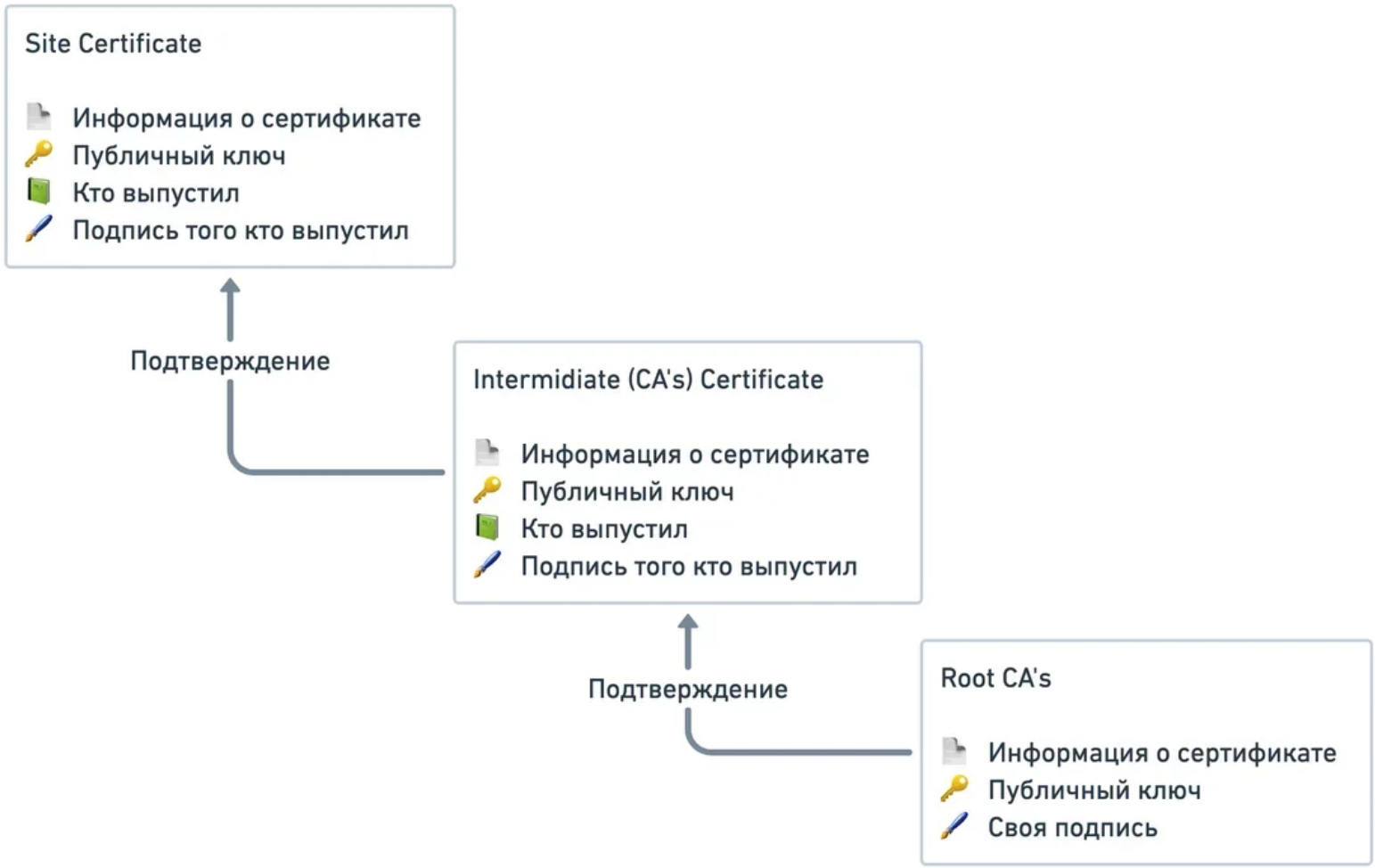
HyperText Transfer Protocol Secure

Man in the middle

Как это работает?


Ваш компьютер
Сервер
- публичный ключ
- приватный ключ
Я хочу HTTPS соединение!
Ok, Вот тебе сертификат (🔑)
1. Проверяет сертификат
2. Создает ключ сессии
3. Подписывает 🔑 публичным ключом
4. + 🔑 = зашифрованный ключ сессии
Вот тебе зашифрованный ключ сессии
+ = ключ сессии
Сессия зашифрованная с помощью

Как проверяется сертификат?



Свой Root Certificate Authority

jamielinux.com/docs/openssl-certificate-authority/

Сгенерировать ключ

```
openssl genrsa -out rootCA.key 2048
```

Сгенерировать Root CA сертификат

```
openssl req -x509 -new -nodes -sha256 -days 1024 \
-key rootCA.key \
-out rootCA.pem
```

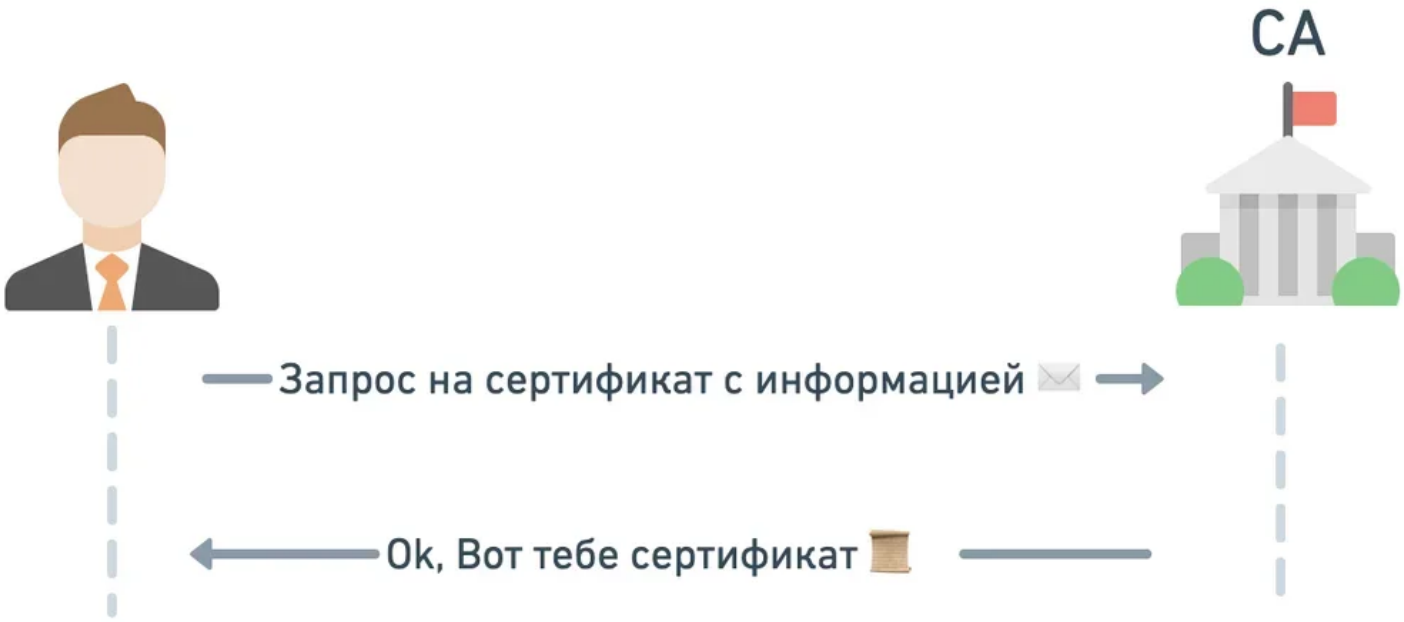
Устанавливаем Root CA сертификат в систему

Firefox

NODE_EXTRA_CA_CERTS

github.com/certifi

Создаем ключ для сертификата сайта и запрос на его создание


CA
Запрос на сертификат с информацией
Ok, Вот тебе сертификат

Генерируем ключ

```
openssl genrsa -out server.key 2048
```

Сертификат на несколько доменов

openssl-csr.conf

```
[ req ]
default_bits = 4096
req_extensions = req_ext
distinguished_name = req_distinguished_name

[ req_ext ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = DNS:localhost, DNS:app.localhost, DNS:*.app.

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
stateOrProvinceName             = State or Province Name (full n
localityName                    = Locality Name (eg, city)
organizationName                = Organization Name (eg, company
```

Создаем запрос для сертификата доменов

```
openssl req -new -key server.key \
-config openssl-csr.conf \
-reqexts req_ext \
-out server.csr
```

Подтверждаем запрос на несколько доменов своим RootCA

```
openssl x509 -days 500 -sha256 -req \
-set_serial 01 \
-extfile openssl-csr.conf \
-extensions req_ext \
-in server.csr  \
-CA rootCA.pem \
-CAkey rootCA.key  \
-CAcreateserial \
-out server.crt
```

Проверяем

А можно проще?

certificatetools.com

mkcert

github.com/FiloSottile/mkcert

```
mkcert -install
mkcert localhost app.localhost '*.app.localhost'
```

SSL Termination

nginx_localhost.conf

```
server {
  listen 443 ssl;

  server_name ~^(?<local_port>\d+)\.app\.localhost$;

  ssl_certificate /YOUR_PATH/server.crt;
  ssl_certificate_key /YOUR_PATH/server.key;

  location / {
    proxy_pass http://127.0.0.1:$local_port;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_f
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Forwarded-Host $server_name;
```

caddyserver.com

github.com/typicode/hotel

Local DNS

dnsmasq

/usr/local/etc/dnsmasq.conf

```
listen-address=127.0.0.1
conf-dir=/usr/local/etc/dnsmasq.d
```

```
mkdir -p /usr/local/etc/dnsmasq.d
tee /usr/local/etc/dnsmasq.d/localhost > /dev/null <<EOF
address=/localhost/127.0.0.1
EOF
```

```
sudo mkdir -p /etc/resolver
sudo tee /etc/resolver/localhost >/dev/null <<EOF
nameserver 127.0.0.1
EOF
```

Вопросы?