


Klíče a certifikáty

 Kompletní přehled pro generování bezpečnostních klíčů, práce s TLS certifikáty, správou SSH přístupu a Git URL.

Generování klíčů & bezpečných hodnot

Základní nástroje pro bezpečné šifrování, tokeny, secrets a root keys.

OpenSSL – (Pro generování náhodných hodnot)

- ▶  Base64 secret (např. JWT, API keys)
- ▶  HEX secret (konfigurační klíče apod.)

Certifikáty & TLS

Moderní způsoby generování certifikátů pro vývoj i servery.

mkcert – Lokální důvěryhodné certifikáty

1. Instalace mkcert

1. Stáhnout `mkcert.exe` z <https://github.com/FiloSottile/mkcert/releases>.
2. Ulož například do `C:\mkcert`.
3. (Volitelné) Přidej tuto složku do systémové proměnné `PATH` pro snadné spouštění z libovolného místa.

2. Instalace lokální certifikační autority (CA)

```
mkcert -install
```

Výsledek:

- CA je nainstalována ve Windows + v prohlížečích (Chrome, Edge...)
- Firefox se musí nastavit ručně (viz níže)

Firefox – ruční přidání CA

1. Otevři Firefox
2. `about:preferences#privacy`
3. **Certificates** → **View Certificates**
4. Tab **Authorities** → **Import**
5. Importuj:

```
C:\Users\<User>\AppData\Local\mkcert\rootCA.pem
```

6. Zaškrtni *Trust this CA to identify websites*

3. Vytvoření certifikátu pro doménu

```
mkcert localhost
```

Výstup:

- `localhost.pem`
- `localhost-key.pem`

Více domén:

```
mkcert localhost 127.0.0.1 myapp.local
```

4. Použití certifikátů

Obvykle použitelné přímo (`.pem`)

- Go
- Node.js
- Nginx
- Caddy
- Docker containers

5. Převod na PFX (.NET / Windows)

```
openssl pkcs12 -export -out server.pfx -inkey localhost-key.pem -in localhost.pem
```

6. Převod na CRT/KEY (Apache, Nginx)

Pouhé přejmenování:

```
localhost.pem → server.crt  
localhost-key.pem → server.key
```

Příklad použití v Go

```
e.StartTLS(":8080", "server.crt", "server.key")
```

SSH – Bezpečné připojení pro GitHub

Co je SSH?

Protokol využívající veřejný a soukromý klíč, bezpečnější než heslo.

Kompletní postup nastavení SSH pro GitHub

- ▶ [1 Generování SSH klíče](#)
- ▶ [2 Zobrazení veřejného klíče](#)
- ▶ [3 Přidání klíče na GitHub](#)
- ▶ [4 Test připojení](#)
- ▶ [5 Klonování pomocí SSH](#)
- ▶ [6 Změna remote URL na SSH](#)
- ▶ [7 Zobrazení URL](#)
- ▶ [8 Oddělené URL pro fetch/push](#)



VPN – Praktický průvodce & tipy



Moderní přehled fungování VPN, výhod, příkladů použití a bezpečnostních doporučení.



Co je VPN?

- **Virtual Private Network** – technologie pro zabezpečené a šifrované připojení přes internet.
- Chrání vaše data, identitu a umožňuje bezpečný vzdálený přístup.
- Umožňuje maskovat IP adresu a obcházet regionální omezení.



Note

VPN je klíčová pro bezpečnou práci na veřejných sítích i pro firemní přístup.



Jak VPN funguje

- ▶ Šifrování dat
- ▶ Tunelování
- ▶ Maskování IP adresy
- ▶ Řízení přístupu



Příklad použití VPN ve firemní síti

- ▶ Zabezpečení dat
- ▶ Přístup k firemním zdrojům
- ▶ Maskování IP adresy
- ▶ Tunelování komunikace