10. OpenSSL: Síťové spojení, verifikace certifikátů, nastavení šifer

Komunikace přes OpenSSL

Za splnění celého zadání dostanete **5 bodů**.

Napište program v jazyce C, který:

- připojí se na webový server (viz url níže)
- ověří certifikát serveru
- nastaví a vypíše typ použité šifry
- stáhne stránku https://fit.cvut.cz/cs/fakulta/o-fakulte do souboru,
- vypíše informace o certifikátu serveru fit.cvut.cz,
- uloží ho do souboru ve formátu PEM.

Dokumentace ke knihovně openssl naleznete na stránkách projektu: http://www.openssl.org/docs/ , https://wiki.openssl.org/index.php/SSL/TLS_Client , návod na práci s certifikáty X.509 např. na https://zakird.com/2013/10/13/certificate-parsing-with-openssl . Nezapomeňte řádně odkazovat na zdroje.

Základní postup

1. Vytvořte TCP spojení na server fit.cvut.cz, port `443` (viz `socket`, `connect`)
2. Inicializujte knihovnu OpenSSL (`SSL_library_init`)
3. Vytvořte nový kontext (`SSL_CTX_new`, použijte metodu `TLS_client_method`)
    a. Zakažte zastaralé a děravé protokoly: `SSL_CTX_set_options(ctx, SSL_OP_NO_SSLv2 | SSL_OP_NO_SSLv3 | SSL_OP_NO_TLSv1 | SSL_OP_NO_TLSv1_1);`
4. Vytvořte SSL strukturu (`SSL_new`)
5. Přiřaďte otevřené spojení (`SSL_set_fd`)
6. Nastavte jméno požadovaného serveru pro mechanizmus SNI: (`SSL_set_tlsext_host_name`)
7. Zahajte SSL komunikaci (`SSL_connect`)
    a. Nyní je navázáno SSL spojení na HTTPS server, můžete poslat požadavek
8. Pošlete HTTP požadavek po zabezpečeném kanálu (`SSL_write`)
    a. Ve smyčce čtěte odpověď po částech, jak přicházejí po síti, a ukládejte výsledná data do souboru (`SSL_read`)
9. Na závěr po sobě uklidíme
    a. Ukončíme SSL session na otevřeném socketu (`SSL_shutdown`)
    b. Zavřeme socket

c. Uvolníme strukturu SSL a kontext (`SSL_free`, `SSL_CTX_free`)

Nezapomeňte přidat do linkování knihovnu ssl `-lssl -lcrypto`.

Použitá šifra

- Zjistěte, na jaké šifře se klient a server dohodnou ve výchozím nastavení
  - po navázání šifrovaného spojení zavolejte `SSL_get_cipher_name`
- Před voláním `SSL_connect` zakažte konkrétní šifru, kterou jste zjistili při prvním spuštění (simulujeme případ, kdy je nalezena zranitelnost šifry a je potřeba ji zakázat), zjistěte, na jaké šifře se dohodne klient a server po této změně.
  - **TLS <= 1.2** Kdybychom např. chtěli zakázat šifru `SEED`, napíšeme `SSL_set_cipher_list(ssl, "DEFAULT:!SEED");`
  - **TLS 1.3** používá jinou sadu šifer (ciphersuites) a jiné funkce API (`SSL_set_ciphersuites`), uvádí se pouze seznam povolených šifer (tu, kterou nechcete, neuvedete)
  - Vypište jméno šifry, kterou jste zakázali, a jméno šifry, která byla zvolena místo ní.
  - Vysvětlete, co znamenají jednotlivé identifikátory, ze kterých se skládá takto získané jméno použité šifry. Zapište svá zjištění.

Verifikace certifikátu serveru

- Po vytvoření nového kontextu (`SSL_CTX`), zavolejte `SSL_CTX_load_verify_locations` nebo `SSL_CTX_set_default_verify_paths` (a zkontrolujte výsledek) – tím nastavíte, kde má knihovna hledat kořenové certifikáty.
- Po vytvoření SSL spojení (`SSL_connect`), získejte výsledek verifikace pomocí `SSL_get_verify_result` a otestujte jej.
- Napište, zda bylo ověření úspěšné.

Verifikaci certifikátu je potřeba provádět při každém navázání spojení.

Přečtení certifikátu ze serveru

- Získejte certifikát od serveru (`SSL_get_peer_certificate`)
- Vypište informace o certifikátu na terminál (např. `X509_get_subject_name`, `X509_get_issuer_name`, `X509_NAME_oneline`)
- a zapište ho do souboru (`PEM_write_X509`).

Příklad kódu pro vytvoření spojení

Známe-li IPv4 adresu serveru, stačí nám pro vytvoření TCP spojení jen několik příkazů.

```
struct sockaddr_in servaddr;
int sockfd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

memset(&servaddr, 0, sizeof(servaddr));
servaddr.sin_family = AF_INET;
servaddr.sin_addr.s_addr = inet_addr("111.111.111.111"); // ip address
servaddr.sin_port = htons(443);                          // port

// Pozor, nektere platformy maji jeste pole sin_len.

if (connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) != 0) {
    error...
}
```

Chcete-li umět překlad jmen (DNS) a zároveň IPv4 i IPv6, doporučujeme `man getaddrinfo`.