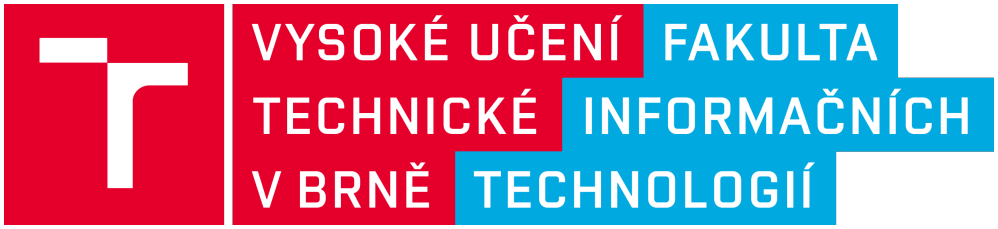

DOKUMENTACE PROJEKTU PŘEDMĚTU ISA

Čtečka novinek ve formátu Atom a RSS s podporou TLS

Tadeáš Kozub [xkozub06]

5. listopadu 2022

Obsah

1 Úvod

Cílem projektu bylo vytvořit konzolovou aplikaci v jazyce C/C++ – RSS / Atom čtečku novinek, která je schopná komunikovat skrze SSL/TLS pomocí knihovny *OpenSSL*.

2 Teorie

2.1 SSL/TLS

SSL (Secure Socket Layer) je kryptografický protokol vydaný v roce 1995 firmou Netscape. Jedná se o nadstavbu nad HTTP, SSL zprostředkovává zejména:[3]

- Ověření identity druhé strany
- šifrování komunikace mezi dvěma stranami

Vzniklo několik verzí SSL (3 major verze)[6] a v roce 1999 vznikl jeho nástupce - TLS (Transport Layer Security). Protokoly jsou si však stále podobné a tak se často hovoří obecně o SSL/TLS.

2.2 OpenSSL

OpenSSL zahrnuje jak CLI nástroj pro různé operace s certifikáty a kryptografickými metodami, tak C/C++ knihovnu pro šifrovanou komunikaci. Knihovna *OpenSSL* má také několik major verzí, těmi nejdůležitějšími jsou 1.1.1 a 3.0.[6] V době psaní této dokumentace přechází nejnovější operační systémy na verzi 3.0 (např. Ubuntu 22, či Fedora 37 / RHEL 9). Pro kompatibilitu se servery Merlin a Eva však byla v projektu použita starší verze 1.1.x – pozor, při kompilaci na modernějších systémech je tedy potřeba downgradovat aktuální verzi knihoven.

Průběh zabezpečené komunikace pomocí knihovny *OpenSSL*:[1][6]

1. inicializace knihovních funkcí
2. vytvoření security kontextu pomocí `SSL_CTX_new(TLS_client_method)`
3. nastavení adresáře s certifikáty
4. vytvoření vstup/výstupní abstrakce pro komunikaci - BIO
5. nastavení SSL komunikace, hostname cílové stanice
6. připojení (BIO se stane zabezpečeným I/O spojením mezi klientem a serverem)
7. ověření platnosti certifikátů
8. zaslání HTTP requestu do BIO
9. přečtení HTTP response od serveru z BIO a uložení
10. (ukončení komunikace a úklid)

2.3 Průběh HTTPS komunikace klient - server

Komunikace začíná v nešifrované podobě, klient zašle na server zprávu Hello a informace o možných šifrovacích sadách, které zná. Server odpoví taky zprávou Hello a následuje výměna klíčů metodou Diffie-Hellman pro symetrickou kryptografii. Díky symetrickým klíčům, které klient i server získali je možné dále komunikovat šifrovanou formou.[4]

Autentizace - komunikující strany si vzájemně vymění certifikáty, které můžou ověřit.[4]

2.4 standard X.509

Digitální certifikát X.509 je dokument ověřující pravost veřejného klíče a příslušnost k danému uživateli. V případě komunikace klient - server (např. surfování na webových stránkách, kde není potřeba autentizace uživatele) se typicky neověřuje identita klienta, ale zejména klient ověřuje indentitu serveru.[4]

3 Technické řešení

3.1 Spouštění programu

Poznámka k překladu: Na školních serverech Merlin a Eva jsou k dispozici statické knihovny a příkaz `make` překládá s flagem `-static-libstdc++`. Je možné, že při spuštění na jiném systému nebudou tyto knihovny dostupné a program make skončí s následujcí chybou:

```
/usr/bin/ld: cannot find -lstdc++
collect2: error: ld returned 1 exit status
make: *** [Makefile:13: all] Error 1
```

V takovém případě, je potřeba spouštět kompilaci bez tohoto flagu, tedy s targetem *nostatic*:

```
$ make nostatic
```

Aplikace se po přeložení spouští příkazem

```
$ ./feedreader <URL | -f <feedfile>> [-c <certfile>] [-C <certaddr>] [-T] [-a] [-u] [-d]
```

Jako podporované zdroje feedů je možné použít jednu URL adresu (http i https adresy jsou podporovány, implicitně s porty :80 pro http a :443 pro https - je však možné specifikovat i vlastní port), **nebo** textový soubor *feedfile* - možné použít s přepínačem -f. Není možné kombinovat URL s feedfile, všechny ostatní přepínače je však možné volitelně kombinovat mezi sebou.

feedfile je obyčejný textový soubor, kde na jednom řádku může být:

(a) komentář - řádek začíná symbolem # (je možné libovolně používat whitespace znaky)

(b) URL adresa ve formátu: `http(s)://www.example.address.com(:PORT)/specific/path/to/feed`

Dále je možné uživatelem specifikovat soubor s certifikátem X.509 (typicky soubory s příponou `.pem, .crt, .cer, nebo .key`), případně také specifikovat adresář s certifikáty (tyto volby je možné kombinovat) příslušnými přepínači -c a -C.

Posledním setem přepínačů, jsou -T, -a, -u pro zobrazení detailů jednotlivých článků a speciální přepínač -d pro spuštění v debug módu. V základním režimu se pro jedntlivé zdroje vypíše pouze nadpis (”Title”) článků, přepínač -T slouží k zobrazení času přídání tohoto článku, případně poslední změny. Přepínač -a zobrazí jméno autora, případně i jeho emailovou adresu (je-li k dispozici) a přepínač -u zobrazí URL odkaz na tento konkrétní článek. Je možné, že ne všechny informace jsou vždy k dispozici - např. RSS feed stránky novinky.cz neuvádí autora, v tomto případě se zobrazí pouze prázdná kolonka autora.

Posledním speciálním přepínačem je volba -d, která umožňuje spouštět aplikaci v debug režimu s tzv. ”verbose” výpisy, které hlásí, co se interně děje a vypisuje detailněji hodnoty, se kterými program pracuje a postupné fáze, kterými prochází.

3.2 Parsování argumentů

O parsování argumentů se stará třída *userArgs*. Ta se stará o validaci a udržování argumentů, kontroluje existenci uživatelem zadaných souborů a správné kombinace přepínačů.

Jednotlivé zdroje feedů jsou uloženy v seznamu URLList, který obsahuje prvek třídy URL, která udržuje pro každý zdroj informace o adrese. Nad každým z těchto prvků v URList se pak dále volají metody třídy *feedFetcher* a *XMLParser*, které z dané adresy získávají a parsují informace - pro detailnější popis vizte kapitoly těchto třídy.

3.3 Získávání dokumentu z URL

O získání HTTP(S) dokumentu ze zadané URL adresy se stará třída *feedFetcher*, která v sobě zahrnuje funkcionalitu pro samotné stažení dokumentu, kde využívá především funkce knihovny *OpenSSL*. V případě, že dojde k problémům k naparsování získaného dokumentu, tak se aktuální zdroj přeskočí a pokračuje se na další.

3.4 Parsování získané odpovědi

K samotnému parsování XML struktury slouží třída *XMLParser*, která nejprve kontroluje formát XML (podporované formáty feedu jsou RSS2 a Atom) a dále prochází XML a vypisuje z něj požadované informace. K parsování je použita knihovna libxml2.

Ke kontrole formátu XML slouží nalezení elementu `<rdf>` pro RSS1 (Tento formát čtečka nepodporuje, takže pouze uživatele informuje, že formát RSS1 není podporován), v případě RSS2[7] hledá element `<rss>` a zároveň kontroluje atribut *"version"*, která musí mít hodnotu *"2.0"*, případně hledá element `<feed>` který značí formát Atom[2]. V jiném případě program zahlásí chybu, že se nejedná o podporovaný formát XML a přeskočí na výpis z dalšího zdroje.

Pro formát RSS2 jsou důležité elementy `title`, `link`, `pubDate` a `author`. Přibližná struktura RSS2 vypadá následovně:[8]

```
<?xml version='1.0' encoding='UTF-8'?\>
<rss version="2.0">
    <channel>
        <title>Novinky - nejnovější články</title>
        <link>https://www.novinky.cz</link>
        <description>Novinky - nejnovější články</description>
        <item>
          <title>Na Václavské náměstí dorazily stovky lidí na podporu Ukrajiny</title>
          <link>https://www.novinky.cz/...</link>
          <pubDate>Sat, 15 Oct 2022 17:18:19 +0200</pubDate>
        </item>
        <item>
            ...
        </item>
        <item>
            ...
        </item>
      </channel>
</rss>
```

Formát Atom se od RSS2 poměrně výrazně liší. Místo elementu channel, ve kterém jsou jednotlivé články jako element item, jsou zde elementy entry uvnitř kořenového elementu feed. Důležitými elementy, které pak čtečka parsuje unvitř jednotlivých elementů entry jsou: `title`, `author`, `link` a `updated`.

Přibližná ukázka formátu Atom:[5]

```
<feed xmlns="http://www.w3.org/2005/Atom">
        <title type="text"> What if </title>
        <entry>
            <title type="text">Transatlantic Car Rental</title>
            <updated>2022-09-06T00:00:00Z</updated>
            <link href="https://what-if.xkcd.com/160/" rel="alternate" type="text/html"/>
```

```
        </entry>
        <entry>
            ...
        </entry>
        <entry>
            ...
        </entry>
    </feed>
```

3.5 Chybové hlášky, zpracování chyb

Všechny chybové hlášky jsou vypisovány na standardní chybový výstup (*stderr*) - k tomuto slouží funkce `ERROR_MSG(string)`. V debug režimu (přepínač -d) jsou na standardní chybový výstup uváděny také informace o aktuální činnosti programu, případně vypsány některé důležité proměnné. Typickým návratovým kódem v případě chyby je rc=1, nebylo nutné rozlišovat velké množství různých chybových stavů. Při chybě je také uveden problém, který nastal, pokud jde o problém v rámci knihovny *OpenSSL*, informace o chybě zajišťuje knihovní funkce `ERR_print_errors_fp(stderr)`

3.6 Testování

K testování projektu byly napsány jednoduché testy v bashi, které se nacházejí ve složce tests/ a je možné je spustit pomocí příkazu

```
$ make test
```

v kořenovém adresáři projektu (`feedreader/`). Testy kontrolují návratové hodnoty testů, zobrazují případné chybové hlášky a také měří běh jednotlivých testů pomocí příkazu `time`.

Odkazy

[1] Kenneth BALLARD. *Secure programming with the OpenSSL API*. [online]. [cit. 2022-11-14]. Srp. 2018. URL: https://developer.ibm.com/tutorials/l-openssl/.

[2] *Introduction to Atom*. [online]. [cit. 2022-11-14]. URL: https://validator.w3.org/feed/docs/atom.html.

[3] Marty KALIN. *Getting started with OpenSSL: Cryptography basics*. [online]. [cit. 2022-11-14]. Čvn. 2019. URL: https://opensource.com/article/19/6/cryptography-basics-openssl-part-1.

[4] Petr MATOUŠEK. *Síťové aplikace a správa sítí: Zabezpečení počítačové komunikace*. [online]. [cit. 2022-11-14]. Říj. 2022. URL: https://moodle.vut.cz/pluginfile.php/509883/mod_resource/content/4/isa-zabezpeceni.pdf.

[5] Randall MUNROE. *Ukázka Atom feedu webu what-if.xkcd.com*. [online]. [cit. 2022-11-14]. Říj. 2022. URL: https://what-if.xkcd.com/feed.atom.

[6] *OpenSSL Manpages*. [online]. [cit. 2022-11-14]. URL: https://www.openssl.org/docs/manpages.html.

[7] *RSS 2.0 Specification*. [online]. [cit. 2022-11-14]. 2003. URL: https://validator.w3.org/feed/docs/rss2.html#syndic8.

[8] *Ukázka RSS2 feedu webu novinky.cz*. [online]. [cit. 2022-11-14]. Říj. 2022. URL: https://api-web.novinky.cz/v1/timelines/section_5ad5a5fcc25e64000bd6e7ab?xml=rss.