VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISA Projekt

Čtečka novinek ve formátu Atom s podporou TLS

29.09.2018 Filip Kočica <xkocic01@fit.vutbr.cz>

Obsah

1 Úvod

V dnešní době, kdy velké množství internetové komunikace je zabezpečené, je dobré znát alespoň základy vytvoření zabezpečeného připojení a následné komunikace. Technická zpráva pojednává o použitých technologiích, návrhu, implementaci a testování aplikace pro čtení novinek.

2 Použité technologie

2.1 RSS a Atom

RSS (Rich Site Summary) je rodina XML formátů určených pro čtení novinek na webových stránkách a obecněji syndikaci obsahu. [1] Slouží pro uživatele k odběru novinek. Atom je standard pro publikaci novinek, nahrazující RSS.

2.2 OpenSSL

OpenSSL (Open Secure Sockets Layer) šifruje data ještě předtím, než opustí počítač a jsou dešifrována až v cílovém bodě. [2] SSL může být použito pro šifrování jakéhokoli protokolu, v našem případě HTTP. Není to jen knihovna pro šifrování dat, ale skládá se také z digitálních certifikátů a podpisů.

2.3 XML

XML (eXtensible Markup Language) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Používá se pro serializaci dat. [3]

2.4 HTML

HTML (Hypertext Markup Language) je v informatice název značkovacího jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. [4]

2.5 HTTP

HTTP (Hypertext Transfer Protocol) je internetový protokol určený pro výměnu hypertextových dokumentů ve formátu HTML. Používá obvykle port TCP/80.

HTTPS (Secured) je syntakticky identické jako HTTP, pouze přidává signalizaci prohlížeči, aby použil šifrovací metodu SSL/TLS k přenosu dat. SSL je vhodné pro HTTP, protože dokáže poskytnout ochranu přenosu, i když je pouze jedna strana komunikace ověřená. Typicky je ověřen pouze server (např. uživatel potvrdí certifikát). [5]

2.6 User-agent

Hypertext Transfer Protocol (HTTP) identifikuje zejména klientský software, který inicioval požadavek, pomocí záhlaví uživatelského agenta, i když klient není obsluhován uživatelem. [6]

3 Implementace

3.1 Vstup

Pro kontrolu a zpracování parametrů programu jsem si na začátku vybral známou funkci `getopt()`, protože práce s ní je jednoduchá a dostačující. Kvůli problémům na serveru eva (více v sekci Testování) jsem ale musel napsat vlastní modul pro zpracování argumentů. Program podporuje zadání URL i souboru feedfile zároveň (musí být zadán alespoň jedna z možností, jinak chyba). Stejně tak certifikátu a cestou k certifikátům, oba jsou předány funkci `SSL_CTX_load_verify_locations()` zároveň. Program nepodporuje zadání stejného parametru vícekrát, protože to zde nedává smysl.

3.2 Vytvoření spojení

K vytvoření zabezpečeného připojení jsem si vybral doporučenou knihovnu OpenSSL, která je nejznámější open-source knihovnou pro vytváření zabezpečené komunikace. Tato knihovna používá abstraktní knihovnu BIO k vytvoření komunikace různých druhů. V našem případě jsem ji využil pro vytvoření soketů (česky přípojek). Poté je potřeba ověřit platnost certifikátu předloženého serverem a čtení/zápis na server je podobný (API) jako u běžných BSD soketů. Rozhraní BIO jsem použil i pro nezabezpečenou komunikaci. V případě, že server neodpoví HTTP 200 (Ani přesměrování se neřeší), je vypsána chyba na standardní chybový výstup a program se ukončí s příslušných chybovým kódem.

Zabezpečené připojení na port 443 se provádí pouze v případě, že je specifikován protokol HTTPS. Pokud je použit HTTP protokol, nebo není specifikován, používá se implicitně nezabezpečené připojení na port 80. Pokud tedy není port explicitně uveden v adrese. Poté se použije tento specifikovaný port.

3.3 Zpracování obsahu novinek

Ke zpracování XML souboru jsem také využil doporučenou knihovnu libxml2, se kterou jsem pracoval poprvé. Při zpracování se vytváří v paměti strom reprezentující strukturu XML souboru, jež se poté prochází a vybírají se podstatné informace.

3.4 Struktura programu

Program je členěn do modulů, obstarávající jednotlivé úkoly, které se nakonec spojí do výsledného spustitelného programu. Zdrojový kód je komentován v univerzálním jazyce pro tvorbu dokumentace Doxygen. Program zanechává malé množství neuvolněné paměti někde v OpenSSL, kterou se mi nepodařilo uvolnit.

3.5 Výstup programu

Dodatečné informace k jednotlivým zdrojům se vypisují v tomto pořadí: URL, čas, autor. V případě, že některá z těchto informací chybí, je hodnota dané položky prázdná (s výjimkou autora, u kterého se vypíše autor celého souboru (feedu), pokud existuje). Pokud chybí jméno (title) některého zdroje (entry), je vypsáno "Neznámý název". Pokud existuje u některého zdroje více autorů, jsou vypsáni všichni.

Ukázka výstupu programu:
Open-source boffins want to do for the IoT edge what Kubernetes did for containers
URL: http://go.theregister.com/feed/www.theregister.co.uk/2018/09/27/...
Aktualizace: 2018-09-27T09:44:01Z
Autor: Richard Chirgwin

4 Licence

Kódy přejaté z tutoriálu IBM i Q/A fóra StackOverflow jsou označeny dle licenčních podmínek. [7][8]

5 Testování

Testování jsem prováděl jak na serveru merlin, tak později na FreeBSD serveru eva. Na evě mi ale nefungovalo zpracování argumentů pomocí getopt, a proto jsem musel provést zpracování ručně. Proto nefunguje "spojené"zadání argumentů (např. -uaT), ale je třeba argumenty zadat jednotlivě (např. -u -a -T). Stejně tak jsem musel trochu upravit Makefile, kvůli linkování externích knihoven a nefunkčnosti wildcard. Asi největší (začátečnická) chyba byla vytvoření souboru feedfile na systému Windows, který používá zalamování řádků CRLF, a poté vývoj a testování na systému Linux/FreeBSD, které pro zalamování řádků používají pouze LF.

Při testování adresy z příkladů "https://tools.ietf.org/agenda/atom", mi server odpovídal na HTTP požadavek odpovědí HTTP 403 – Forbidden. Spolužák mi poté poradil, ať použiji u HTTP požadavku "User-Agent". Poté už všechno fungovalo a server odpovídal HTTP 200 s požadovaným obsahem.

5.1 Testy

Dle zadání se po zadání příkazu ‘make test‘ mají spustit námi napsané testy. Vzhledem k tomu, že obsah informačních kanálů se často mění bylo potřeba čas od času testy aktualizovat. V době testovaní už pravděpodobně některé neprojdou kvůli jinému obsahu zdrojů. Testy jsou napsány jako skript v bash a obsahují referenční výstupy jednotlivých testů ve složce RefResutls, se kterými je porovnáván výstup programu. Aplikace je otestována na serverech merlin i eva.

6 Omezení

Žádná – aplikace je plně funkční. Aplikace podporuje RSS 0.91, RSS 1.0 (Modul Dublin Core), RSS 2.0 i Atom. Je také robustní a dokáže odolat nevalidním vstupům, stejně jako nevalidnímu XML. Dokáže si také poradit s kombinací verzí RSS, např. použití elementu <dc:creator> v RSS 2.0. V takovýchto případech jsou data vypsána na standardní výstup, jako kdyby byly ve správném elementu.

7 Příklady použití

Použití:

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

Příklady (server merlin.fit.vutbr.cz):

```
feedreader https://www.theregister.co.uk/headlines.atom \
                    -c /etc/ssl/certs/ca-bundle.crt -u -a -T
feedreader https://www.theregister.co.uk:443/headlines.atom
feedreader https://www.theregister.co.uk/headlines.atom -u -a -T
feedreader -f feedfile
feedreader https://www.theregister.co.uk/headlines.atom -f feedfile
```

8 Závěr

Výsledkem práce je robustní otestovaná aplikace pro zpracování novinek uváděných na serverech v různých formátech. Pro příště bych snad jen trochu víc přemýšlel nad důsledkem kombinace operačních systémů Linux a Windows. Stejně tak bych začal testovat na serveru eva používající operační systém FreeBSD dříve (nejlépe od začátku), abych nemusel spoustu věcí nakonec přepisovat. Obecně se budu více soustředit na přenositelnost vytvořených programů mezi různými platformami.

9 Literatura

[1]*RSS: Rich Site Summary. Wikipedia* [cit. 2018-09-30]. Dostupné z: https://cs.wikipedia.org/wiki/RSS

[2]*OpenSSL: Secure programming with the OpenSSL API* [cit. 2018-09-30]. Dostupné z: https://www.ibm.com/developerworks/library/l-openssl/

[3]*XML: Extensible Markup Language. Wikipedia* [cit. 2018-09-30]. Dostupné z: https://cs.wikipedia.org/wiki/Extensible_Markup_Language

[3]*HTML: Hypertext Markup Language. Wikipedia* [cit. 2018-11-09]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Markup_Language

[4]*HTTP: Hypertext Transfer Protocol. Wikipedia* [cit. 2018-11-09]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[5]*User agent: User agent. Wikipedia* [cit. 2018-10-09]. Dostupné z: https://en.wikipedia.org/wiki/User_agent

[6]*Attribution required: StackOverflow* [cit. 2018-10-10]. Dostupné z: https://stackoverflow.blog/2009/06/25/attribution-required/

[7]*IBM License Agreement: IBM* [cit. 2018-10-10]. Dostupné z: https://www.ibm.com/developerworks/apps/download/index.jsp?contentid=11410&filename=intro-openssl.zip&method=http&locale=