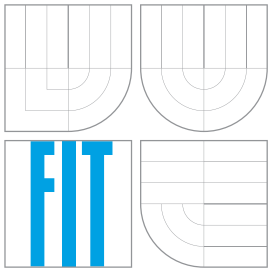
Vysoké učení technické v Brně
Brno University of Technology



Fakulta informačních technologií

Faculty of Information Technology

Projekt číslo 2 - Navodzovanie SSL spojení a validácia certifikátov knižnicou OpenSSL

Kryptografia

Autor práce

Login xsulak04

Brno 05/2017

Úvod

Cieľom tohto projektu bolo vytvoriť nástroj, ktorý bude využívať knižnicu *OpenSSL* pre získanie informácií o SSL certifikátoch. Na základe získaných informácií bude aplikácia vypisovať ku každému serveru mieru dôvery v daný certifikát do 4 kategórií. Číslo 1 určuje najvyššiu mieru zabezpečenia a 4 najnižšiu, kedy je stránka vyslovene nedôveryhodná. Číslo 2 sú systémy, ktorým sa celkom dôveruje, no nie tak ako bankovým systémom a do kategórie 3 spadajú systémy, kde je vyslovene niečo podozrivé. Podrobnejšie informácie sú obsiahnuté v nasledujúcej sekcií.

Posúdenie bezpečnosti

Táto sekcia sa zaoberá jadrom projektu a teda vysvetľuje, ako bola hodnotená miera dôveryhodnosti. Boli posudzované iba certifikáty, samotná stránka síce obsahovala nejaké chyby, no neboli brané do úvahy:

- Kategória 1 - nič podozrivé, kritické alebo škodlivé nebolo nájdené, až na 1 výnimku, pokiaľ využíva koreňová certifikačná autorita kolíznu hešovaciu funkciu *SHA1*. V prípade koreňovej to až tak nemusí vadiť, v prípade ostatných medzi ňou a serverom miera bezpečnosti tiež ešte spadá pod túto kategóriu.
- Kategória 2 - v prípade, že server používa hešovaciu funkciu *SHA1*, prípadne ešte nejakú ako je *MD5* poskytujúcu menšiu mieru zabezpečenia, spadne server do tejto kategórie. Rovnako tak použitie krátkeho verejného kľúča na strane serveru.
- Kategória 3 - pokiaľ server nepracuje so službou TLS vo verzií 1.2, ale so staršou, tak sa síce vykoná rollback a spojenie sa podarí i s následnou analýzou, no už len kvôli tomuto kroku bude server posudzovaný ako nie úplne dôveryhodný a bude v tejto kategórii. Okrem toho ešte aj vtedy, ak má subjekt chybný alebo chýbajúci názov, prípadne ak obsahuje príliš obecný vzor (s použitím wildcardu).
- Kategória 4 - väčšina ostatných chýb, najmä takých, ktoré sú výsledkom verifikácie knižnicou *OpenSSL*. Spadajú sem problémy ako napríklad nemožnosť obdržať certifikát vydavateľa, expirácia certifikátu, ak bol podpísaný sám sebou alebo je neplatný. V tomto projekte sa bolo možné stretnúť väčšinou s týmito problémami, no knižnica *OpenSSL* a teda i toto riešenie dokáže odhaliť množstvo ďalších problémov.

Implementácia

Projekt je implementovaný v jazyku C++ v prostredí 64-bitového operačného systému Fedora 25. V zložke *cert/* sa nachádza certifikát certifikačnej autority laboratória CRoCS FI MU[1]. Okrem tohto certifikátu je nutné načítať ešte aj certifikáty uložené v systéme, čo je realizované následne. Celá práca s knižnicou OpenSSL zaisťuje služba *CryptoService*, ktorá je bázovou triedou samotného klienta implementovaného v súboroch *Client.cpp* s príslušným hlavičkovým súborom.

Aplikácia nepožaduje žiadny parameter, no je možné zadať *-h* alebo *–help* pre vypísanie nápovedy. Je možné meniť logovanie a výpisy pomocou konštánt v *CryptoService.h* a to nasledovným spôsobom:

[1] http://minotaur.fi.muni.cz/crocs-ca.pem

- PRINT_CERT - pri nastavení na true sa bude vypisovať certifikát získaný z každého serveru,
- PRINT_SERVERS_INFO - pri zapnutí tejto možnosti je možné obdržať informácie o každom serveri aj na štandardnom výstupe, nie len vo výstupnom *CSV* súbore. Je to najmä kvôli prehľadnosti a pre testovacie a ladiace účely,
- ANALYZE_GET_REQ - táto možnosť posiela *HTTP GET Request* postupne každému serveru, získava a vypisuje odpoveď. Implementované kvôli možnosti nahliadnuť na server priamo z aplikácie a prípadne analyzovať, čo webový server vracia.

Záver

V tejto práci sa podarilo implementovať automatické vyhodnocovanie SSL certifikátov, ktoré sa ukladá do výstupného *CSV* súboru. Čo sa týka nejakých nedostatkov alebo ďalších vylepšení, tak rozhodne by stálo za to implementovať dodatočný modul kontrolujúci, či certifikát nebol revokovaný. Na to je nutné stiahnuť *CRL*[2], prípadne i *OCSP*[3]. Niektoré certifikáty v rámci tohto zadania boli zneplatnené, no kvôli časovej tiesni táto funkcionalita nebola implementovaná. Pri tomto riešení boli využívané viaceré zdroje informácií, napríklad Stack Overflow[4], SuperUser[5], OpenSSL[6] či iné[7][8].

Dotazník

Táto posledná sekcia obsahuje dotazník presne podľa zadania. Odpoveď 1 znamená rozhodne nie (prípadne nikdy) a číslo 5 znamená rozhodne áno (prípadne veľmi často):

- Používali jste již někdy před tímto úkolem OpenSSL API? **2**
- Chci používat OpenSSL API často. **3**
- OpenSSL API je zbytečně složité. **3**
- OpenSSL API bylo snadné použít. **2**
- Potřebuji podporu více zkušeného vývojáře, aby mohl používat OpenSSL API. **1**
- Funkce v OpenSSL API byly dobře integrovány. **4**
- Vnímám v OpenSSL API příliš mnoho nekonzistence. **2**
- Většina vývojářů se naučí používat OpenSSL API velmi rychle. **3**

[2] Certificate Revocation List, https://www.ietf.org/rfc/rfc5280.txt

[3] Online Certificate Status Protocol, https://tools.ietf.org/html/rfc6960

[4] Nájdenie systémovej zložky s certifikátmi, http://stackoverflow.com/questions/4138139/how-to-find-out-the-path-for-openssl-trusted-certificate

[5] Problematika SHA1 v koreňovej certifikačnej autorite, https://superuser.com/questions/1122069/why-are-root-cas-with-sha1-signatures-not-a-risk

[6] https://wiki.openssl.org

[7] Ukážka získania verejného kľúča z certifikátu pomocou knižnice OpenSSL, http://fm4dd.com/openssl/certpubkey.htm

[8] Ukážka spracovania certifikátu pomocou knižnice OpenSSL, https://zakird.com/2013/10/13/certificate-parsing-with-openssl

- OpenSSL API má velmi těžkopádné k použití. **4**
- Cítil jsem se velmi jistý při použití OpenSSL API. **3**
- Před použitím OpenSSL API jsem se potřeboval naučit spoustu věcí. **1**