VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí – 4. projekt

Monitoring SSL spojení

18. listopadu 2020 Samuel Krempaský

1 Popis

Cieľom tohoto programu bolo analyzovanie komunikácie SSL(poprípade TLS) či už pomocou súboru typu pcapng alebo načúvaním na sieťovom rozhraní.

2 Teoretické základy pre tvorbu tohoto programu

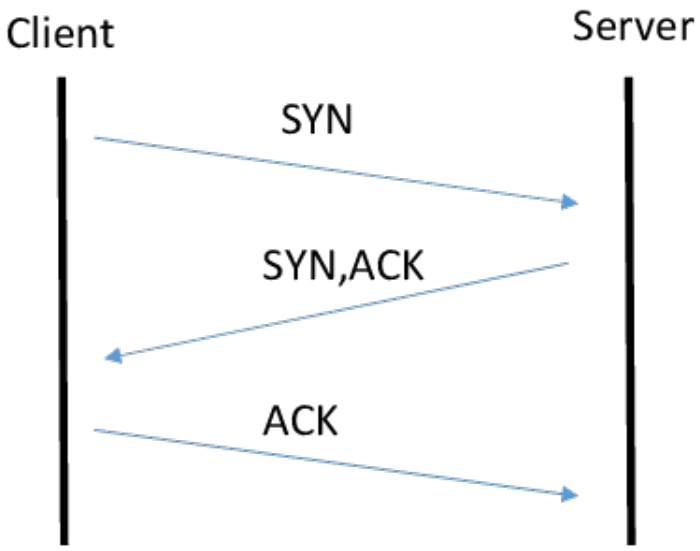
Na vytvorenie toho programu boli potrebné znalosti programovacích jazykov C/C++, poznať základný teoretický model ISO/OSI a model TCP/IP, využívať funkcie knižnice libcap a takisto si naštudovať správanie protokolu TLS/SSL.

2.1 TCP/IP a vzťah s TLS/SSL

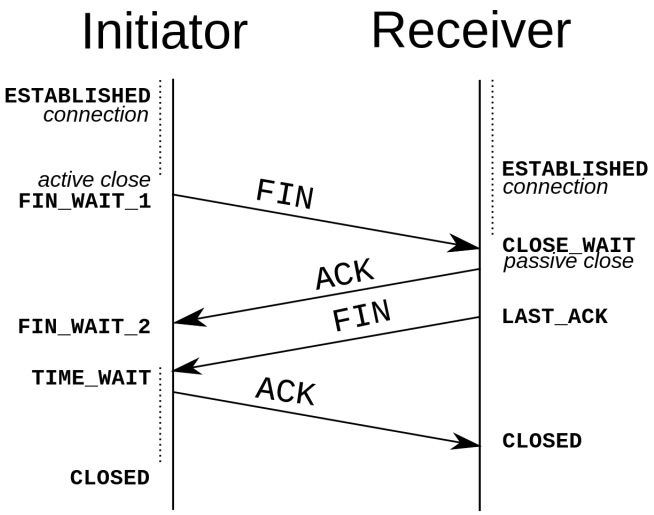
Model TCP/IP vznikal počas studenej vojny z potrieb vytvoriť decentrizovanú a robustnú sieť.[3] Avšak tento model nebol zabezpečený a s komercializáciou internetu vznikla potreba prenášať po sieti citlivé informácie ako napríklad heslá, čísla debetných a kreditných kariet a podobne. Odpoveďou na túto požiadavku sa stal protokol SSL. Ten bol postupne vyvíjaný počas 90. rokov až napokon sa vyvinul do TLS, ktoré bude tento program analyzovať. TLS/SSL je obsiahnutý v TCP payload ktorý bude analyzovaný.

2.2 Priebeh komunikácie klient-server TLS/SSl

Priebeh spojenia prebieha napjrv nadviazaním TCP spojenia tzv. 3-way handshakom a to tak že sa najprv požiadavok SYN odošle klient ten následne prijme prijme SYN+ACK a v poslednom 3. kroku odošle požiadavok ACK.[3]



Následne ukončenie prebieha pomocou požiadavku FIN. Tento krok začína klient. Ten by mal následne prijať ACK+FIN a odoslať ACK. Avšak komunikácia môže byť aj ukončená pomocou požiadavku RST, ako to bolo napr. v referenčnom riešení.



Medzi nadviazaním TCP spojenia a jeho ukončením sa takisto nadväzuje spojenie TLS a to podľa tzv. ClientHello a ServerHello.[1][2]. ClientHello inicializuje klient TLS/SSL komunikáciu a ponúka serveru šifry aké môže používať a následne takisto posiela informácie potrebné pre nadviazanie spojenia. Správa prichádzajúca zo servera sa nazýva ServerHello ktorá naďalej pokračuje v nadväzovaní spojenia. Bližšie sú tieto správy rozoberané v sekcii Implementácia.

3 Implementácia

Program sa začína v funkcii main, kde sú spracované argumenty. Popis argumentov je nasledovný:

1. `./sslsniff -r example.pcapng`
2. `./sslsniff -i wlp5s0`

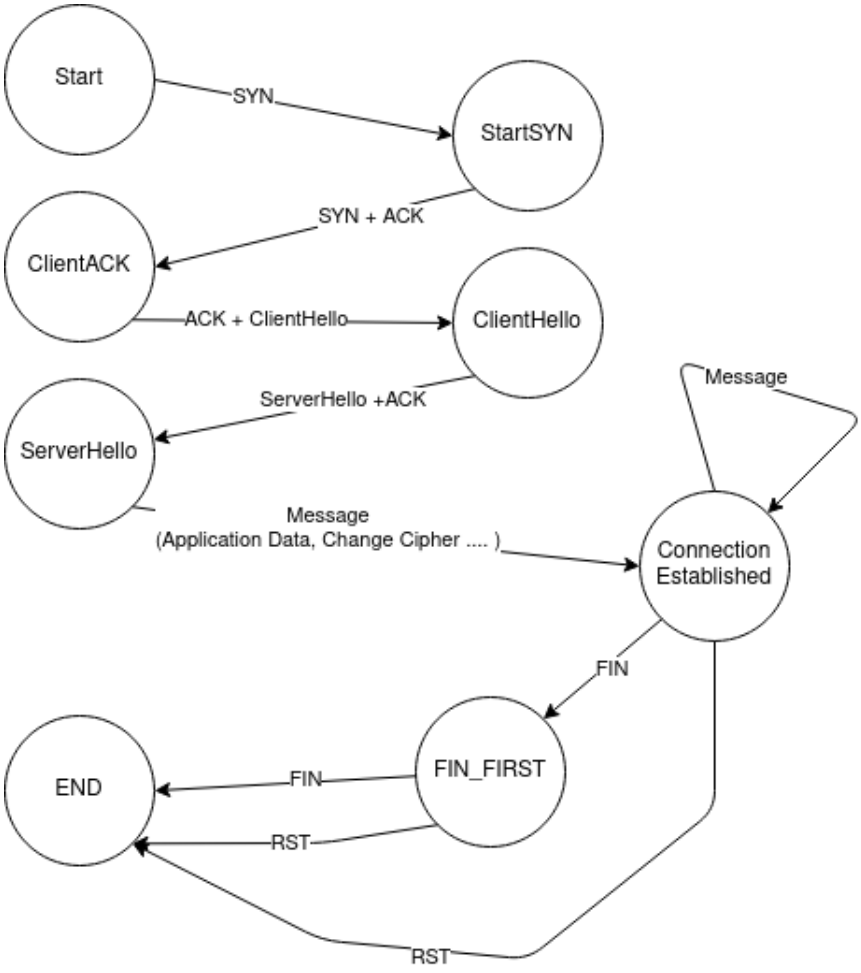
- `./sslsniff` - meno programu
- `-i/-r` - prepínače, -i znamená rozhranie, -r znamená prepínač pre súbor
- `example.pcapng` - meno súboru
- `wlp5s0` - meno rozhrania

Po tejto časti nasledujú funkcie na otvorenie či už súbora alebo rozhrania a jedná sa o `pcap_open_live()` pre načúvanie na rozhraní alebo `pcap_open_offline()` pre otvorenie súbora. Následne sa dostávame do hlavnej slučky kde používame funkcie `pcap_next()` ktorou zaisťujeme chod cyklu a `pcap_next_ex()` torá nám pomocou premmennej `data` poskytuje prístup k dátam celého paketa. tento hlavný cyklus je používaný pri live capture aj pri čítaní zo súboru.

Tento cyklus je jedným z hlavných častí programu. Ďalšou podstatnou časťou je `struct connection`, v ktorom je definovaných niekoľko atribútov ako IP adresy, čísla portov, informácie o čase, momentálny stav pripojenia a takisto ukazovateľ na ďalší `struct`, keďže sa jedná o jednosmerný viazaný zoznam.

V tele cyklu sa následne dostaneme k hlavičkám protokolov ethernet a IP, z ktorých zistíme o akú verziu IP. Následne sa dostávame k protokolu TCP z ktorého získavame dáta o veľkosti, časové údaje a nakoniec k dátam v TCP v ktorom sa nachádza SSL/TLS. Tento packet sa dostáva do druhej najdôležitejšej funkcii `FSmach()`, kde sa nachádza konečný automat na spracovanie paketov a ich správ.

3.1 Implementácia konečného automatu komunikácie



Jedná sa o o približný konečný automat implementovaný v zdrojovom kóde programu. Nachádzame sa v stave `Start` kedy klient posiela správu SYN Serveru a dostávame sa do stavu `StartSYN` kedy Server príma túto správu. Následne posiela posiela server klientovi SYN+ACK. Klient môže následne poslať ACK buď samostatne alebo aj spolu s CLientHello. V tomto okamihu už považujeme komunikáciu TCP/IP za stanovenú a je potrebné sa pozrieť na stanovenie komunikácie pomocou TLS/SSL.

Analýzu TLS/SSL komunikácie začíname v stave CLientHello, a stará sa nám o tom funkcia s obdobným názvom a to `analyzeClientHello()`. Vďaka tejto funkcii zistíme tzv. SNI, čiže hostname a takisto si zapíšeme aj počet bytov a ďalšie potrebné informácie do `struct connection`. Ak prebehlo všetko správne tak nasleduje správa zo Servera tzv. ServerHello. Táto správa je spracovávaná v obdobnej funckii a to `analyzeServertHello()`. Po úspechu tejto funkcii t.j. táto správa splnila všetky požiadavky ServerHello sa následne ešte TCP paket v ktorom je táto správa obsiahnutá ešte ďalej analyzuje na prípadné ďalšie TLS/SSL hlavičky, keďže môže nastať problém s tzv. reassemblingom paketov t.j. keď TLS hlavičky a ich payload je obsiahnutý vo viacerých TCP paketoch.

Tento problém som vyriešil tak že po každej detekovanej TLS hlavičke som ešte kontroloval či za jej payloadom sa nenachádza ďalšia TLS hlavička s obsahom. Toto kontrolovanie prebieha iteratívne a stále ak sa detekuje hlavička tak sa preskočí niekoľko počet bytov daného payloadu a aktualizuje sa počet bytov v hlavnej štruktúre `connection`.

Po správnom priebehu správ ServerHello a ClientHello sa TLS/SSL komunikácia dostane do stavu Connection Established, kedy len príjma TCP pakety a tieto pakety sú obdobne analyzované ako v predošlých stavoch. Túto časť skúma funkcia `analyzeEstablishedConnectionHeadears()`, ktorá takisto analyzuje pakety ako v predošlých funkciách pre reassembling.

Ak v tomto stave príde TCP paket s FIN tak sa dostane do tzv. stavu FIN_FIRST kedy vyčkáva na ďalší FIN a ten nakoniec toto spojenie ukončí a dostane sa do Stavu END. Avšak takisto môže prísť aj správa RST do stavu ConnectionEstablished alebo do FIN_FIRST a po prijatí RST sa ukončenie ukončí bez medzistavu. V stave END sa nakoniec vypíše ukončené spojenie podľa špecifikácie zadania a spočíta dĺžku trvania spojenia.

4 Splnenie požiadavkov

Mnou implementovaný program dokáže zachytiť komunikáciu s použitím IPV4 a IPV6, live capture a takisto analyzovať súbor formátu `.pcapng`. Avšak pri testovaní neboli vždy zachytené správne počty bytov a počty packetov ale vo všetkých základných testoch mi to fungovalo avšak pri obsiahlejších testoch tam boli drobné odchýlky. Tento program som napísal sám, avšak niektoré časti boli prebrané z dema od garanta tohoto predmetu a to hlavne pri hlavnom cykle kde získavame pakety zo súbora alebo z rozhrania. Inak všetko ostatné je implementované mnou.

Literatúra

[1] Dierks, T.: The Transport Layer Security (TLS) Protocol Version 1.2. https://tools.ietf.org/html/rfc5246#section-7.4.1.2, last accessed 2020-11-18.

[2] Driscoll, M.: TLS Illustrated. https://tls.ulfheim.net/, last accessed 2020-11-18.

[3] Matoušek, P.; učení technické v Brně, V.: *Sít'ové aplikace a jejich architektura*. VUTIUM, 2014, ISBN 9788021437661. Dostupné z: https://books.google.sk/books?id=LpnhrQEACAAJ