VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

DETEKCE A ANALÝZA PŘENOSŮ VYUŽÍVAJÍCÍCH PROTOKOLY SSL/TLS
TRAFFIC DETECTION AND ANALYSIS USING SSL/TLS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Jan Hutar

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. David Smékal

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Jan Hutar ***ID:*** 155115
Ročník: 2 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Detekce a analýza přenosů využívajících protokoly SSL/TLS

POKYNY PRO VYPRACOVÁNÍ:

Cílem diplomové práce je popsat a nastudovat problematiku protokolů SSL/TLS a nástrojů pro analýzu zabezpečených spojení pomocí těchto protokolů.
Úkolem je navrhnout postup pro detekci přenosů využívající bezpečnostní protokoly, který bude použitelný v rámci procesu DPI (Deep Packet Inspection).
Dále bude proveden rozbor extrakce metadat, které by mohly sloužit pro další hlubší analýzu a filtraci takto zabezpečených spojení.
Výstupem bude praktická implementace postupu pro detekci SSL/TLS na simplexních/duplexních linkách do softwarové knihovny „OpenDPI“ a vytvoření softwarového nástroje pro analýzu a filtraci spojení s protokolem SSL/TLS.

DOPORUČENÁ LITERATURA:

[1] T. Dierks, E. Rescorla: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, RFC Editor, August 2008. URL https://tools.ietf.org/html/rfc5246

[2] ntop : Open and Extensible LGPLv3 Deep Packet Inspection Library. Ntop [online]. [cit 2016-09-01]. URL http://www.ntop.org/products/deep-packet-inspection/ndpi/

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 24.5.2017

Vedoucí práce: Ing. David Smékal
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Diplomová práce se zabývá detekcí a analýzou zabezpečených spojení elektronické komunikace prostřednictvím protokolů SSL/TLS. V práci je prostudována problematika SSL/TLS protokolů. Následně byla provedena analýza zpráv používaných při navazování zabezpečeného spojení a zpráv poštovních služeb SMTP, POP3 a IMAP při použití STARTTLS. Detekce a extrakce metadat zabezpečených simplexních a duplexních spojení probíhá za použití prostředků hluboké inspekce paketů. Výchozím nástrojem byla použita knihovna funkcí nDPI z projektu Ntop. Knihovna byla rozšířena o detekci zmíněných spojení a extrakci metadat na základě studia a analýzy přenášených zpráv. V závěru práce je provedeno testování na cvičné množině dat a základní analýza získaných metadat.

KLÍČOVÁ SLOVA

SSL, TLS, X.509, DPI, Metadata, SMTP, POP3, IMAP, STARTTLS

ABSTRACT

This diploma thesis deals with a detection and analysis of secure connections of electronic communication through SSL/TLS protocols. The thesis begins with introduction to SSL/TLS protocols. Thereafter, an analysis of messages used to establish secure connections using STARTTLS and postal protocols SMTP, POP3, and IMAP was made. Metadata detection and extraction of secured simplex and duplex connections take place using deep packet inspection tools. The tool of choice is the nDPI library from the Ntop project. The library was extended to detect the connections and extract the metadata based on studies and analysis of transmitted messages. Finally, testing is performed on a training data set and a basic analysis of acquired metadata is made.

KEYWORDS

SSL, TLS, X.509, DPI, Metadata, SMTP, POP3, IMAP, STARTTLS

HUTAR, Jan *Detekce a analýza přenosů využívajících protokoly SSL/TLS*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, Rok. 62 s. Vedoucí práce byl Ing. David Smékal

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Detekce a analýza přenosů využívajících protokoly SSL/TLS“ jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval kolegovi Ing. Romanu Břečkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci, a vedoucímu práce Ing. Davidu Smékalovi. Také bych chtěl poděkovat své manželce Kláře za morální podporu a pomoc, kterou mi poskytla nejen během psaní diplomové práce, ale během celého studia, a které si nesmírně vážím.

Brno ....

podpis autora(-ky)

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

Úvod **11**

1 Popis protokolů SSL a TLS **13**
1.1 Hlavní cíle protokolů SSL/TLS 14
1.2 Konstrukce protokolů 14
1.2.1 Rozbor navázání spojení SSL/TLS 16
1.2.2 SSL/TLS a STARTTLS 16
1.3 Šifrovací metody a pečeti SSL/TLS 17
1.3.1 Výměna klíčů 18
1.3.2 Šifrování přenášených dat 18
1.3.3 Pečeti 18
1.4 Použití protokolů 18
1.4.1 HTTP 19
1.4.2 FTP 19
1.4.3 Přenos pošty 19
1.5 Verze SSL a TLS 20

2 Rozbor metadat spojení SSL/TLS **23**
2.1 Certifikát X.509 23
2.1.1 Získání certifikátu ze zachycených dat 23
2.1.2 Struktura certifikátu X.509[15] 24
2.2 Vybraná metadata z SSL/TLS spojení 24

3 Hluboká inspekce paketů DPI **26**
3.1 Princip DPI 26
3.1.1 Vyhodnocení výsledků 26
3.2 Využití 27
3.2.1 Nástroje hluboké inspekce paketů 29
3.2.2 Porovnání nástrojů 29
3.2.3 Nástroj nDPI 30
3.3 pcapreader 31
3.3.1 Kompilace 31

4 Detekce a analýza SSL/TLS **33**
4.1 Návrh 33
4.1.1 nDPI vnitřní struktura zpracování dat 34
4.2 Detekce 35
4.2.1 Simplexní a duplexní spojení SSL/TLS 36

4.2.2 Detekce poštovních služeb SMTP, POP3, IMAP v případě použití STARTTLS . . . 39
4.2.3 Detekce SSL/TLS spojení poštovních služeb SMTP, POP3, IMAP v případě použití STARTTLS . . . 45
4.2.4 Vyhodnocení zpracovaných metadat . . . 45
4.2.5 Export metadat . . . 46
4.3 Tshark . . . 47
4.4 Analýza získaných metadat . . . 48
4.5 Testování . . . 49

5 Závěr **54**

Literatura **55**

Seznam symbolů, veličin a zkratek **57**

Seznam příloh **59**

A Návod k použití programu **60**

B Obsah přiloženého CD **62**

SEZNAM OBRÁZKŮ

1.1 Zapouzdření aplikačních dat do vrstvy SSL/TLS. 15
1.2 Diagram navázání spojení TLS. Hvězdičky označují údaje, které nejsou nezbytné pro každé spojení a mohou se v závisloti na spojení měnit. . 15
1.3 Ustanovení zabezpečeného TCP spojení mezí Alicí a Bobem. 17
1.4 Přenos pošty protokolem SMTP za použití STARTTLS. 20
3.1 Škálování datového provozu[10]. 27
3.2 Analýza znaků chování přenosu. 28
3.3 Výstup z pcapreader. 32
4.1 Schéma detektoru/analyzátoru. 34
4.2 Vnitřní struktura programu pcapReader. 35
4.3 Hlavička TLS record. 37
4.4 Zpráva Server_Hello TLS record. 37
4.5 Paket Client_Hello, Certificate, Client_Key_Exchange. 38
4.6 Přenos pošty protokolem POP3 za použití STARTTLS. 42
4.7 Přenos pošty protokolem IMAP za použití STARTTLS. 44
4.8 Výstup z pcapreader. 53

SEZNAM TABULEK

4.1 Tabulka zpráv SSL/TLS pro přesnější detekci a extrakci metadat . . 36
4.2 Tabulka podmínek hledání zpráv. 38
4.3 Tabulka rozšíření struktury ndpi_flow->protos.tcp.ssl. 39
4.4 Tabulka zpráv SMTP pro přesnější detekci a extrakci metadat 40
4.5 Tabulka nové Bitmasky ndpi_flow->l4.tcp.smtps_command_bitmask 41
4.6 Tabulka zpráv POP3 pro přesnější detekci a extrakci metadat 43
4.7 Tabulka doplnění Bitmasky
ndpi_flow->l4.tcp.pop_command_bitmask 43
4.8 Tabulka zpráv IMAP pro přesnější detekci a extrakci metadat 44
4.9 Tabulka nové Bitmasky
ndpi_flow–>l4.tcp.imaps_command_bitmask 45
4.10 Tabulka četnosti zachycených cipher suit 48
4.11 Tabulka četnosti zachycených verzí SSL/TLS 49
4.12 Tabulka cvičných vzorků a porovnání výsledků detekce 50
4.13 Tabulka cvičných vzorků a porovnání výsledků detekce 52

ÚVOD

Elektronická komunikace je v současné době jedno z nejvíce rozvíjejících se odvětví internetu. S nárůstem používání elektronické komunikace a s nástupem IoT (Internet of Things) stále více vystupuje do popředí bezpečnost komunikace. Bezpečností komunikace se rozumí nejen zabezpečení proti odposlechu třetí strany, tedy důvěrnost komunikace, ale také autenticita komunikujících stran.

Když se Edward Snowden v červnu roku 2013 v Hongkongu sešel s novinářem z listu The Guardian[1], odstartoval tím novou dobu v bezpečné komunikaci. Povědomost veřejnosti o bezpečné komunikaci byla do té doby minimální a dalo by se říci, že byla i ze strany provozovatelů služeb řešena na minimální úrovni. Od roku 2013 začaly být masivně používány komunikační protokoly SSL/TLS a stejně tak začalo být ve velkém měřítku prováděno penetrační testování elektronických systémů.

Vývoj bezpečnostních rutin ale začal již mnohem dříve společností Netscape Comunication kolem roku 1994. Jako první byl vydán protokol SSL (Secure Socket Layer) 1.0, který nebyl nikdy oficiálně publikován. Za následujících pět měsíců byla vydána další, již oficiálně publikovaná verze SSL 2.0 jako RFC 6176 [1] a s ním i webový prohlížeč Netscape, který tento protokol podporoval. Na tuto skutečnost reagovala i společnost Microsoft vydáním vlastního protokolu PCT (Private Comunication Technology). Protokol SSL byl volně publikován a byl dán k dispozici dalším vývojářským firmám. Velmi brzo se z něj stal standard bezpečné komunikace. Protokol SSL je i nadále vyvíjen. Od roku 1999 je vydáván pod názvem TLS (Transport Layer Security) a 11. června 2016 byl publikován návrh na novou verzi 1.3[2].

Diplomová práce je zaměřena na testování zabezpečení elektronické komunikace a analýzu datových spojení. Nepoužívání protokolů SSL/TLS představuje bezpečnostní riziko, kdy není zajištěna důvěrnost dat ani autenticita komunikujících stran. Přestože jsou protokoly SSL/TLS hojně využívány, stále existují služby, které z důvodu jejich kompatibility a požadavku na jejich dostupnost nepoužívají nejnovější verze těchto protokolů, nebo je nepoužívají vůbec.

První část práce se zabývá teorií a je rozdělena do třech kapitol. První kapitola se zabývá popisem protokolu SSL/TLS. Je zde popsán přehledem vývoj verzí protokolu SSL/TLS a jejich rozdíly, proveden rozbor teorie navázání zabezpečeného spojení a uvedeny příklady služeb využívajících protokoly SSL/TLS včetně použití STARTTLS.

[1] Dostupné z URL: <https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance>

[2] Dostupné z URL: <https://tools.ietf.org/html/draft-ietf-tls-tls13-14>

Druhá kapitola teoretické části práce seznamuje s certifikátem X.509 a vybírá metadata vhodná k uchování z zabezpečeného spojení pro potřeby následné analýzy.

Třetí kapitola teoretické části se zabývá teorií identifikace služeb za použití techniky hluboké inspekce paketů. Součástí kapitoly je provedeno porovnání dostupných nástrojů a seznámení s knihovnou funkcí nDPI vhodnou pro rozvoj do praktické části diplomové práce.

Drůhá část diplomové práce se zabývá návrhem detektoru/analyzátoru SSL/TLS spojení a vlastní praktickou implementací návrhu do knihoven funkcí nDPI. Úvodem je zmíněna vnitřní struktura knihoven nDPI. Následuje analýza obousměrných, jednosměrných zabezpečených spojení a poštovních služeb používajících STARTTLS. Výsledky analýzy jsou použity pro rozšíření a úpravy funkcí jednotlivých detektorů projektu nDPI. Všechny úpravy jsou v práci popsány. Dále je knihovna funkcí nDPI rozšířena o sběr a export požadovaných metadat. Na získaných informacích a metadatech je proveden příklad výsledné analýzy. Závěrem praktické části je vyhodnocení provedeného testování a porovnání výsledků s neupravenou verzí.

1 POPIS PROTOKOLŮ SSL A TLS

Ideologie šifrované komunikace

- **Navázání spojení**
  - **Dohoda na algoritmech**
    - asymetrické šifrování pro výměnu klíčů,
    - symetrické šifrování pro šifrování provozu,
    - hešovací funkce pro kontrolu integrity (MAC).
  - **Předání sdíleného tajemství (asymetrické šifrování) veřejným kanálem**
  - **Autentizace (certifikát serveru, případně klienta)**
- **Přenos aplikačních dat**
  - **Symetrické šifrování**
  - **Zajištuje soukromí i integritu**

Protokoly SSL/TLS jsou primárně navrženy k zajištění integrity a důvěrnosti mezi dvěma komunikujícími aplikacemi. Pro přenos využívají protokol transportní vrstvy TCP (Transmission Control Protocol), nad kterým vytvoří zabezpečené spojení. Spojení je poskytnuto protokolům aplikační vrstvy jako je HTTP (Hypertext Transfer Protocol), SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol) atd. K zajištění důvěrnosti je použita symetrická kryptografie s pomocí blokových šifer jako je RC4 (Bloková šifra – ARCFOUR) a AES (Advanced Encryption Standard). Klíč šifrování pro každé spojení je vždy generován unikátní. Zajištění integrity přenášeného obsahu je prováděno vkládáním heše a pečeti do přenášené zprávy MAC (Message Authentication Code). Protokoly jsou v současné době dostupné ve verzích: SSL 2.0 (rok vydání 1995), SSL 3.0 (rok vydání 1996), TLS 1.0 (rok vydání 1999), TLS 1.1 (rok vydání 2006), TLS 1.2 (rok vydání 2008), z toho již některé není doporučeno používat z důvodu snadné prolomitelnosti. Projekt Thrustyworthy[1] provádí jednou měsíčně testování a hodnocení na asi 150 tisících nejpopulárnějších serverech a jejich podpory protokolů SSL/TLS.

- **Autentičnost** je zajištěna použitím certifikátů, veřejných klíčů asymetrického šifrování např.RSA (Asymetrický šifra – Rivest Shamir Adleman), DSA (Digital Signature Algorithm) atd.
- **Důvěrnost** použitím vhodných šifrovacích algoritmů a infrastrukturou veřejných klíčů PKI (Public Key Infrastructure).
- **Integrita** je výsledkem spojení autentičnosti a důvěrnosti spojení.

[1]Dostupné z URL: <`https://www.trustworthyinternet.org/ssl-pulse/`>

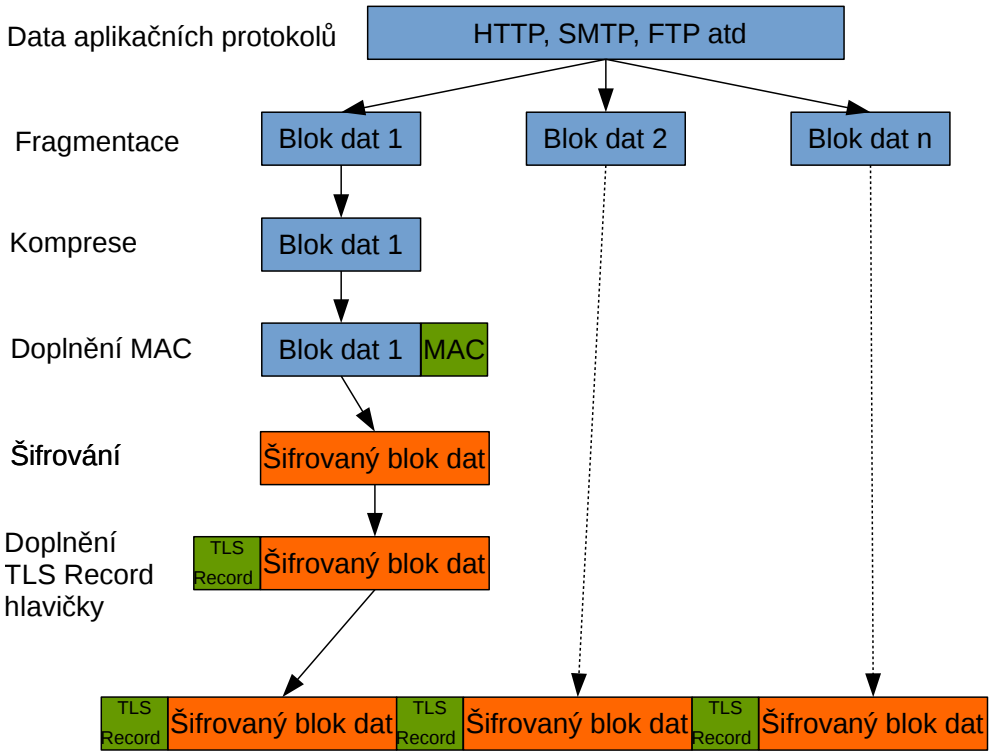
1.1 Hlavní cíle protokolů SSL/TLS

- **Kryptografická bezpečnost** protokolů, které by měly být použity k vytvoření bezpečného spojení mezi oběma komunikujícími stranami.
- **Interoperabilita** (možnost programátorům vytvářet) aplikací nezávisle vytvářených a využívajících protokolů SSL/TLS.
- **Rozšiřitelnost** o nové nezbytné šifrovací metody bez nutnosti tvorby nového protokolu.
- **Relativní efektivnost** snižuje nároky šifrovacích algoritmů na únosnou mez snížením množství potřebných spojení s využitím vhodných schémat ukládání do mezipaměti[2].

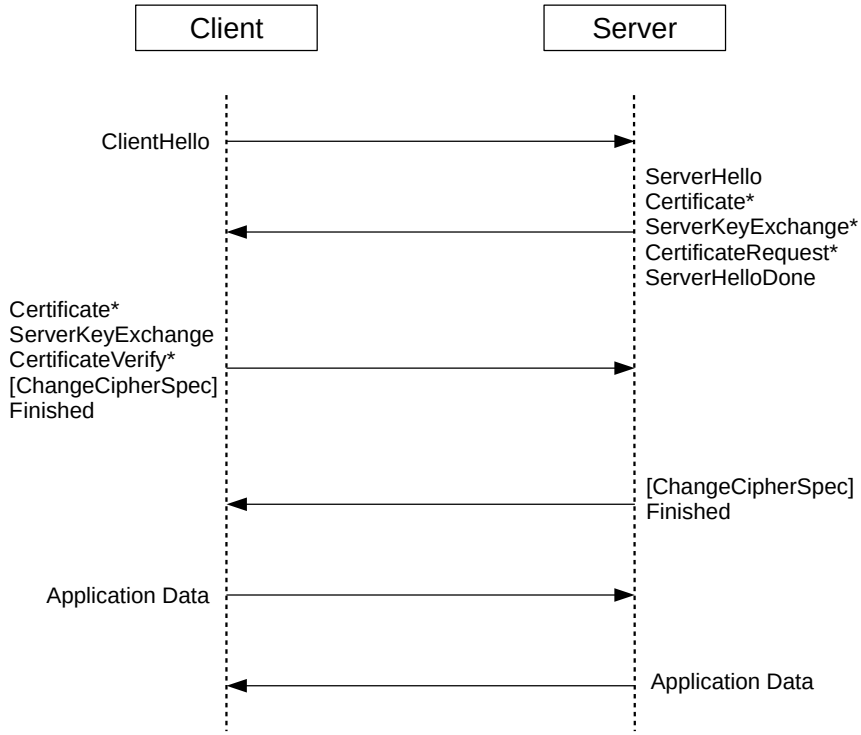
1.2 Konstrukce protokolů

Protokoly SSL/TLS jsou tvořeny spojením dvou vrstev:

- **Record protokol** je umístěn jako první na TCP protokolu a zapouzdřuje protokoly, viz diagram 1.2, vyšších vrstev včetně sady protokolů pro navázání spojení (handshake protokolů) do bloků. Při zapouzdřování je vložena pečeť MAC. Pečeť je počítána z aktuálního nešifrovaného bloku dat. Následně je celý blok včetně MAC zašifrován blokovou šifrou[2].
- **Protokoly navázání spojení (handshake)** jsou čtyři a slouží k ověření identity účastníků spojení, vyjednání šifrovací sady a dalších potřebných parametrů spojení. Navázání spojení s výměnou schopností probíhá v následujícím pořadí viz obr. 1.2.
  - **Handshake protokol** umožňuje účastníkům spojení ověřit identitu a vyjednat nezbytné parametry bezpečného spojení: identifikátor spojení, certifikáty klientů, kompresní metodu, šifrovací algoritmy, hlavní heslo, možnost znovupoužití spojení.
  - **Alert protokol** slouží k signalizaci problémů a komunikaci výjimek.
  - **Change Cipher spec. protokol** umožnuje signalizaci změny šifrovací strategie v průběhu relace. Zprávu může odeslat klient i server. Na tuto zprávu umí okamžitě reagovat Record protokol. Po odeslání této zprávy je další blok zprávy šifrován dle nové strategie.
  - **Application data protokol** bere data z aplikačního protokolu a vkládá je do zabezpečeného kanálu[2].



Obr. 1.1: Zapouzdření aplikačních dat do vrstvy SSL/TLS.



Obr. 1.2: Diagram navázání spojení TLS. Hvězdičky označují údaje, které nejsou nezbytné pro každé spojení a mohou se v závisloti na spojení měnit.

1.2.1 Rozbor navázání spojení SSL/TLS

Zjednodušený rozbor zde uvedený představuje dnes nejpoužívanější způsob navázání spojení, kdy je použit certifikát jen ze strany serveru a ověření identity klienta probíhá až prostřednictvím aplikace. Navázání spojení mezi klientem (Alice) a serverem (Bobem) probíhá níže popsaným způsobem, viz obr. 1.3:

- **1. Client Hello** Alice odešle zprávu Bobovi, že chce začít komunikovat zabezpečeně. Zpráva odeslaná Bobovi obsahuje náhodné číslo R_Alice ze seznamu podporovaných CipherSuite.
- **2. Server Hello** Bob odešle po obdržení zprávy Alici svůj cetifikát a vlastní náhodné číslo R_Bob. Pokud se s jeho CipherSuit shoduje jedna nebo více sad, pošle Alici ten, který pro komunikaci vybral (v současnosti nejčastěji TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256).
- **3.** Alice odesílá zašifrované náhodné číslo $pre\_mastersecret$ pomocí Bobova veřejného klíče získaného z certifikátu. Pro zajištění integrity přiloží hash $master\_secret$. (Pro rozlišení hešů je na straně klienta vkládán ASCII (American Standard Code for Information Interchange) řetězec CLNT pro SSL a pro TLS $client\_finished$). Na straně serveru je tento konstatní řetězec označován SRVR pro SSL a $server\_finished$ pro TLS.
- **4.** Bob po obdržení zprávy vytvoří hash ze všech předchozích zpráv a odešle jej šifrovaně Alici. Tím je ustanovena důvěrnost a integrita mezi oběma komunikujícími stranami.

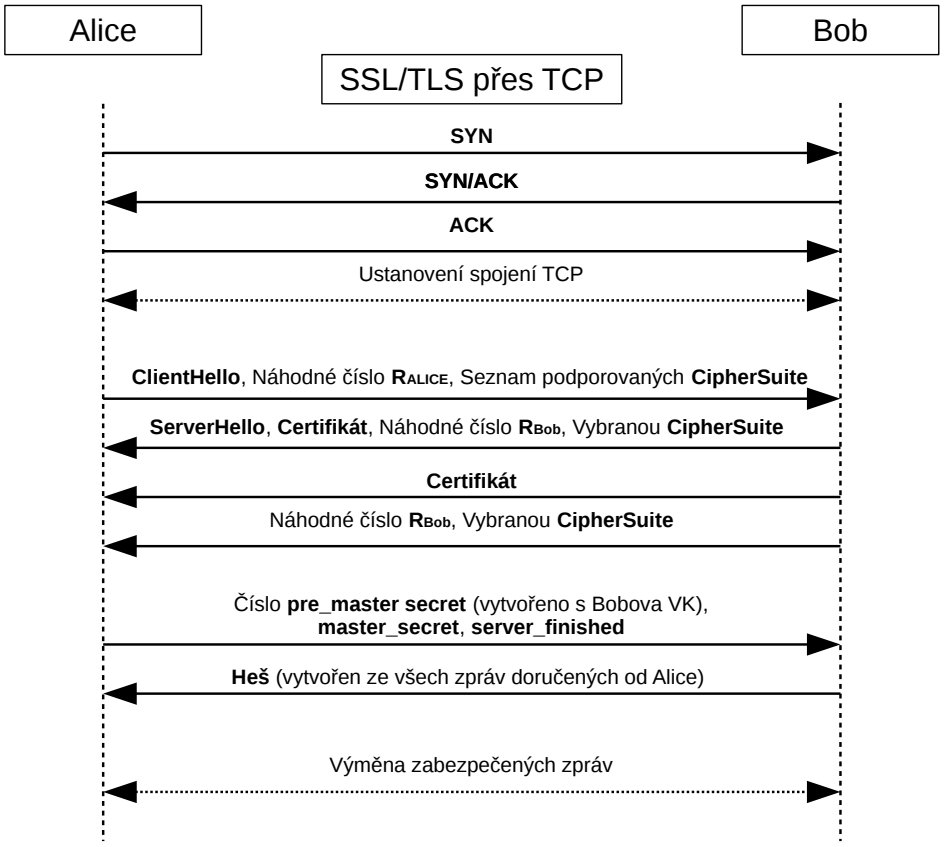
1.2.2 SSL/TLS a STARTTLS

Vytvoření zabezpečené komunikace lze provést dvěma možnými způsoby mezi serverem a klientem.

- **SSL/TLS** – první možností, užívanou např. u HTTP protokolu, je nejprve navázat otevřené spojení na jednom portu (pro http port 80). Zde si klient/server vymění informaci o ustanovení (přesměrování další komunikace) dalšího portu pro komunikaci (pro http port 443), zde je komunikace navázána dle uvedeného schématu (grafu toků) šifrovaně a do vytvořeného spojení jsou dále již vkládaná data z komunikace http.

- **STARTTLS** – další možností použití protokolů SSL/TLS je realizace spojení na jednom portu. Tento druh spojení využívá např. protokol SMTP dle RFC 3207 [3], POP3 a IMAP dle RFC 2595 [4] obr. 1.4. Na jednom portu mezi serverem a klientem je navázáno spojení a probíhá komunikace. V momentě, kdy je potřeba zahájit šifrovaný přenos dat, je poslána zpráva STARTTLS. Od

té doby je spuštěna komunikace dle uvedeného schématu a navázáno šifrované spojení. Vše probíhá stále na jednom portu.



Obr. 1.3: Ustanovení zabezpečeného TCP spojení mezí Alicí a Bobem.

1.3 Šifrovací metody a pečeti SSL/TLS

Každá verze protokolů podporuje určité šifrovací protokoly. V současné době jsou již některé šifrovací algoritmy zastaralé z hlediska jejich prolomitelnosti. Šifrovací metody jsou stanoveny prostřednictvím takzvané Cipher suite, tedy šifrovací sady, která je nabízena při vytváření spojení. Jako první je iniciována sada CipherSuite TLS_NULL_WITH_NULL_NULL = (0x00, 0x00), která nesmí být ustanovena. Sada říká, že nebude použit žádný algoritmus pro šifrování a pečetění[2].
Příklad šifrovací sady z protokolu TLS 1.2:
CipherSuite TLS_RSA_WITH_RC4_128_MD5 = (0x00, 0x04).
Pořadí významů použitých zkratek oddělených podtržítkem je:

- použitý protokol **TLS**,
- šifrovací algoritmus pro výměnu klíčů **RSA**,

- šifrování přenášených dat a délka klíče **RC4_128**,
- hešovací algoritmus pro pečetění bloků dat **MD5**.

Na webových stránkách organizace IANA (Internet Assigned Numbers Authority) [2] jsou uvedeny všechny aktuální šifrovací sady.

1.3.1 Výměna klíčů

Výměna klíčů je prováděna pomocí asymetrického šifrování[3] za použití např. RSA šifry, DH Diffie Helmanova algoritmu, DHE eliptických křivek atd. Aktuální přehled je uveden na webových stránkách organizace IANA[2].

1.3.2 Šifrování přenášených dat

Přenášená data jsou šifrována pomocí proudových šifer[4], jako např. RC4 nebo nově ChaCha20-Poly1305 a blokových šifer[5], jako je AES, Camelliea a několik dalších. Aktuální přehled je uveden na webových stránkách organizace IANA[2].

1.3.3 Pečeti

Do každého bloku zprávy je před šifrováním vložena pečeť MAC. Pečeť MAC je používána s hešovací funkcí jako HMAC (Keyed-hash Message Authentication Code). Slouží k zaručení integrity přenášeného bloku. Aktuální přehled je uveden na webových stránkách organizace IANA[2].

1.4 Použití protokolů

Zabezpečený přenos pomocí protokolů SSL/TLS využívá mnoho aplikací například: HTTP, FTP, SMTP, IMAP (Internet Message Access Protocol), DNS (Domain Name System), Samba (Server Message Block), SIP (Session Initiation Protocol) atd.

[2] Transport Layer Security (TLS) Parameters z URL: <`https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml`>

[3] Asymetrická šifra je taková kryptografická metoda, která pro šifrování a dešifrování používá odlišné klíče (soukromý klíč SK a veřejný klíč VK). Je to základní rozdíl oproti symetrické šifře, která používá k šifrování a dešifrování stejný klíč.

[4] Proudová šifra je typ symetrické šifry, kde vstupní datový tok je kombinován s klíčem nejčastěji pomocí funkce XOR.

[5] Bloková šifra je typ symetrické šifry. Vstupní data jsou šifrována do přesně stanovených bloků.

1.4.1 HTTP

Přenos protokolu HTTP pomocí SSL/TLS je dnes velmi využíván. Základní princip použití SSL/TLS probíhá takto:

- Klient se dotáže na stránku pomocí HTTP protokolu na portu 80.
- Server odpoví, že komunikace bude přesměrována na port 443.
- Klient se připojí na port 443 a zde se již rozběhne navázání zabezpečené komunikace.

Klient si ale tímto způsobem nemůže zabezpečenou komunikaci vynutit a není tedy zabezpečeno, že bude komunikace probíhat podle výše uvedeného scénáře. Tento uvedený scénář je taky vystaven riziku od možného útočníka na síti tzv. MITM (Man-in-the-middle)[6] , který by mohl modifikovat spojení. Proto je zaveden protokol HSTS (HTTP Strict Transport Security)[5], který zajistí vynucení komunikace pomocí SSL/TLS. Pokud tak není provedeno, klienta upozorní prostřednictvím webového prohlížeče.

1.4.2 FTP

Metody přenosu dat FTP pomocí SSL/TLS jsou dvě explicitní nebo implicitní vychází z RFC 4217 [6].

- **Explicitní FTP** – prvotní spojení je provedeno na portu 21 typickém portu pro FTP. Po té je vyjednána žádost o zahájení SSL/TLS spojení. Celé spojení je přesměrováno na port 990, kde je zahájeno SSL/TLS spojení obdobně jako u HTTP přenosu.
- **Implicitní FTP** – spojení je zahájeno rovnou na portu 990 a zabezpečené spojení SSL/TLS je vytvořeno prvotně.

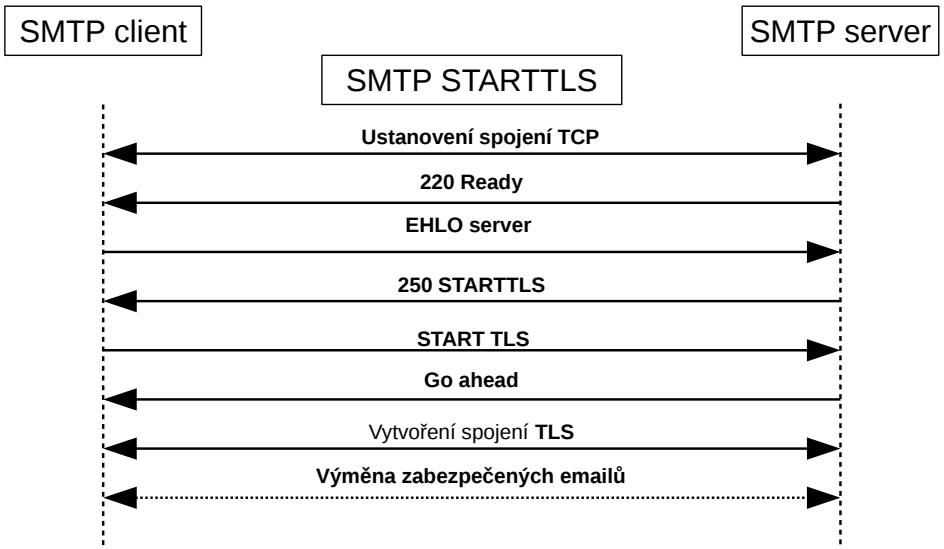
1.4.3 Přenos pošty

Přenos elektronické pošty lze realizovat několika protokoly např. SMTP, POP3, IMAP atd. Elektronická pošta se přenáší nejen mezi klientem a poštovním serverem, ale také mezi poštovními servery různých poskytovatelů. Poštovní služby využívající SSL/TLS mají vyhrazeny porty např. pro SMTP port 465, POP3 port 995, IMAP port 993 atd.

[6]Man-in-the-middle - Muž uprostřed: Název útoku vznikl z basketbalové terminologie, kdy se dva spoluhráči snaží provést přihrávku skrz protihráče. V počítačové terminologii se jedná o neoprávněné vstoupení do komunikace dvou stran podstrčením informací aniž by o tom komunikující strany věděli.

Přenos elektronické pošty velmi často využívá STARTTLS viz odstavec STARTTLS, viz obr. 1.2.2. Příklad průběhu komunikace mezi SMTP klientem a SMTP serverem je zobrazen viz obr. 1.4. Přenášené zprávy SMTP klient a SMTP server:

- 220 Ready: Kladná odpověď připraven.
- EHLO server: Odpověd HELO s oznámením o použití rozšíření SMTP.
- 250 STARTTLS: 250 kladná odpověď na použití STARTTLS.
- STARTTLS: odpověď od klienta použít STARTTLS.
- Go ahead: server posílá klientovi souhlas, že může začít spojení. Jako další zprávu od klienta již očekává ClientHello.



Obr. 1.4: Přenos pošty protokolem SMTP za použití STARTTLS.

1.5 Verze SSL a TLS

SSL 1.0

Protokol nebyl nikdy oficiálně vydán a je zveřejněn pouze návrh z roku 1995[7]. Protokol nebude blíže popisován.

SSL 2.0

- Protokol podporuje pro výměnu klíčů pouze RSA asymetrickou šifru.

[7]The SSL protocol Dostupné z URL: <http://graphcomp.com/info/crypt/ssl_v3.txt>

- Šifrování dat 3DES EDE CBC, IDEA CBC, DES CBC RC2 CBC, RC4 128 a RC4 40. Všechny uvedené šifry, které lze použít pro přenos dat, jsou dnes považovány za slabé.
- Protokol podporuje jednu verzi MAC ve spojení s hešovacím algoritmem MD5.
- Současné verze webových prohlížečů SSl 2.0 nepodporují. Navázání spojení handshake není chráněno proti útoku Man in the Midle[1].

SSL 3.0

- Doplnění kombinace DHE-RSA DHE-DSS (Diffie Hellman) pro výměnu klíčů.
- Zodpovědnost za volbu šifrování je přenesena na stranu serveru oproti SSL 2.0, kde byla na straně klienta[7].

TLS 1.0

Rozdíly oproti protokolu SSL 3.0 nejsou veliké a jsou provedeny především v oblasti použitých šifrovacích algoritmů:

- Rozšíření šifrovacích algoritmů pro předání šifrovacího klíče o metody využívající eliptické křivky ECDHE.
- Rozšíření šifrovacích algoritmů o blokové šifry AES, Camellia, ARIA a SEED.
- Použití pečetící funkce HMAC.
- TLS protokol může doplnění končit v jakékoliv délce, která je násobkem délky bloku, dříve muselo být doplnění co nejkratší.

Pro pečetění bloků je použita funkce HMAC (vylepšená verze MAC)[8].

TLS 1.1

Hlavní změny od TLS 1.0.

- Implicitní IV (inicializačni vektor) je nahrazen explicitním - ochrana proti útokům na CBC schéma kódování[9].

TLS 1.2

Hlavní změny od TLS 1.1[8]:

- Zvětšení flexibility při sjednávání kryptografických algoritmů.
- Nahrazení MD5/SHA-1 ve funkci pseudonáhodného generátoru vlastními funkcemi.
- Použití MD5/SHA-1 v kombinaci s digitálním podpisem prvků bylo nahrazeno jednou funkcí explicitně definovanou.

- Značné odebrání schopností klienta a serveru v akceptovaných hešovacích a podpisových algoritmech.
- Zpřísněná kontrola verze EncryptedPreMasterSecret.
- Délka $Verifidata$ není závislá na šifrovací sadě (výchozí délka zachována).
- Odstraněna možnost Bleichenbacher/Klima útoku.
- Rozšíření Alert protokolu o nové signály.
- Pokud nemá klient k dispozici certifikát, musí poslat prázdný list.
- Jako mandatorní Cipher suit je TLS_RSA_WITH_AES_128_CBC_SHA
- Odstranění zastaralých šifer DES a IDEA.
- Rozšíření šifrovacích algoritmů o další možné.
- Doplnění algoritmů pečetění o HMAC_SHA256.

2 ROZBOR METADAT SPOJENÍ SSL/TLS

Metadata – jsou to data, která poskytují informaci o jiných datech.

Rutiny navázání spojení SSL/TLS přenášejí mnohdy mnoho informací o komunikujících stranách. Analýzou těchto dat lze o komunikujících stranách i o přenášených datech získat charakteristické informace. V průběhu ustanovení spojení SSL/TLS je ve zprávě Client_Hello rozšíření, které obsahuje pole s názvem serveru, od kterého požaduje certifikát. Dále je také obvyklé použit certifikát X.509. Používání certifikátů je založeno na hierarchii Certifikačních autorit CA (Certificate authority). Popis fungování certifikace a struktury PKI (Public Key Infrastructure) popis veřejných klíčů není obsahem této práce a dále se jimi tato práce nezabývá, více lze nálezt zde [14].

2.1 Certifikát X.509

Certifikát X.509 je vyvíjen od roku 1988 a v současné době je používán ve verzi 3 a publikován jako standard v RFC 5280. Vybrané certifikáty vydané důvěryhodnými certifikačními autoritami CA bývají předinstalovány v operačních systémech a webových prohlížečích. Takovéto certifikáty jsou označovány jako kořenové a slouží k ověření pravosti komunikujících stran[15]. V případě, že server nemá od takovéto autority vydán certifikát, je o tom uživatel informován a musí jeho použití odsouhlasit. Neplatné, nebo neznámou autoritou podepsané certifikáty jsou využívány k modifikaci, nebo podstrčení spojení útočníkem na síti, tyto útoky jsou označovány jako MITM 6. Certifikát obsahuje informace o tom, kdo jej vydal a komu, viz příklad 2.1. Informace obsažené v certifikátu velmi dobře slouží jako metadata.

2.1.1 Získání certifikátu ze zachycených dat

Certifikát lze vyextrahovat ze zachycených dat, uložených do formátu pcap pomocí analyzačního nástroje Wireshark[1]. V zachyceném datovém toku je nutné najít pomocí filtru ssl spojení. Dále pak sledováním průběhu ustanovení spojení nalézt v přenášený datech certifikát, označit a uložit. Certifikát v podobě bajtů pak lze snadno převést do textové podoby, například pomocí programu Openssl.

```
openssl x509 -inform der -in vutbr.cert  -text
```

[1]Wireshark je program k analýze síťových protokolů a zachytávání dat s velmi propracovaným grafickým rozhraním.

2.1.2 Struktura certifikátu X.509[15]

- Version: verze certifikátu,
- Serial number: pořadové číslo certifikátu,
- Signature algorithm: algoritmus použití k podpisu,
- Issuer: vydavatel certifikátu,
- Validity: platnost (nepoužívat před datem, nepoužívat po datu),
- Subject: vlastník veřejného klíče,
- Subject public key info: informace o veřejném klíči (algoritmus, klíč),
- Issuer unique identifer: unikátní identifikátor vydavatele,
- X509v3 extension: volitelné rozšíření,
- Signature algorithm: algoritmus elektronického podpisu,
- Certifikát: elektronický podpis.

2.2 Vybraná metadata z SSL/TLS spojení

- Použitá verze SSL/TLS na straně klienta i serveru.
- Ustanovení šifrování Cipher_Suite obsažené ve zprávě Hello_server.
- V případě použití HTTP hostname.
- Rozšíření Server_name ze zprávy Client_Hello.
- Název domény vlastnící certifikát Subject CN.

Výpis 2.1: Příklad úvodní části certifikátu X.509 zdroj: `https://www.vutbr.cz`. Obsah položky seznam DNS a modulus byl odstraněn z důvodu zmenšení. Dále bylo odstraněna část rozšíření, elektronický podpis a veřejný klíč.

```
Certificate:
Data:
    Version: 3 (0x2)
    Serial Number:
    08:1b:57:3f:01:63:da:f4:0f:63:35:ce:d5:66:f0:15
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=NL, ST=Noord-Holland, L=Amsterdam, O=
           TERENA, CN=TERENA SSL CA 3
    Validity
        Not Before: Sep 22 00:00:00 2016 GMT
        Not After : Sep 27 12:00:00 2019 GMT
    Subject: C=CZ, ST=Moravia, L=Brno-střed, O=Vysoké uč
       ení technické v Brně, CN=www.vutbr.cz
    Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                Exponent: 65537 (0x10001)
    X509v3 extensions:
    X509v3 Authority Key Identifier:
        keyid:67:FD:88:20:14:27:98:C7:09:D2:25:19:BB:E9
           :51:11:63:75:50:62
    X509v3 Subject Key Identifier:
        2E:1C:27:D2:4B:AE:90:CF:B8:FF:9C:CD:A6:B1:D7:97:5
           C:9B:1B:DF
    X509v3 Subject Alternative Name:
        DNS:www.vutbr.cz,.............................
```

3 HLUBOKÁ INSPEKCE PAKETŮ DPI

Definice DPI: Hluboká inspekce paketů (DPI) analyzuje všechna data IP paketu, jak ze záhlaví, tak z datové výplně, procházející kontrolním bodem za účelem zjištění, protokolu, aplikace a dalších meta-dat přepravovaného provozu[1].

Hluboká inspekce je používána k monitorování a kontrolování obsahu přenášených dat na aplikační vrstvě modelu ISO/OSI. Diferencování síťového provozu na základě znalosti pouze komunikačního portu je dnes již nedostačující. Pod číslem jednoho portu se může ukrývat velké množství rozdílných služeb. Pro úplnost je nezbytné provádět DPI i nad šifrovanými přenosy, které jsou v současné době významně zastoupeny v datových přenosech. Kontrola přenášených dat je používána v pokročilých firewallech. Pomáhá zabránit šíření škodlivého softwaru (malware) po síti a také k filtraci webových služeb z důvodu uplatnění firemní politiky. Monitorování síťového provozu může být používáno poskytovateli připojení ISP (Internet Service Provider) k účtování poplatků, rozlišení síťové zátěže a tím lepšímu rozložení. DPI je dále využíváno k vládnímu dohledu nad síťovým provozem[10].

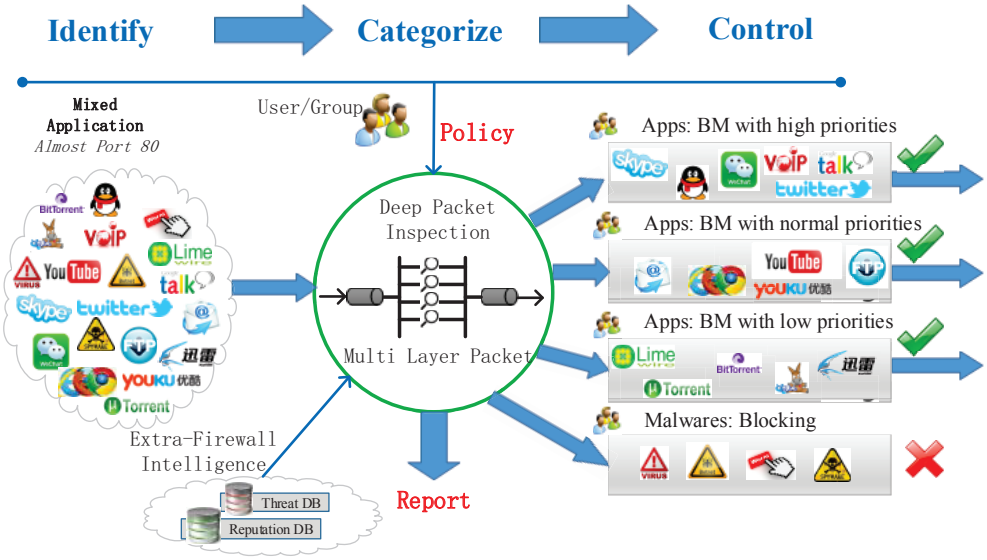
3.1 Princip DPI

Hlavní pointa (jádro, princip) provádění DPI je v porovnávání vzorů. Hluboká inspekce paketů se provádí za provozu v reálném čase bez jakéhokoliv omezení a zásahu do spojení, nebo na datech uložených. Vzory pro porovnávání vznikají z analýzy charakteristických znaků chování síťového provozu dané aplikace. Vlastní vyhodnocování začíná již na 3. a 4. vrstvě ISO/OSI. Každá vrstva modelu sebou nese svou hlavičku a v případě poslední, tedy vrstvy aplikační, není ani zřejmé, kde hlavička přenášeného protokolu jednoznačně končí a kde začíná přenášený obsah. Výchozím bodem pro specifikaci aplikace může být již kombinace částí obsahu IP (Internet Protocol) a TCP hlavičky[10].

3.1.1 Vyhodnocení výsledků

Při provádění DPI je velmi důležitá úspěšnost. Při porovnávání vzorů mohou často vzniknou výsledky False positive. False positive nastává ve chvíli, kdy je na základě podobnosti v průběhu porovnávání vzorů přiřazen nesprávný výsledek. Následně je provoz nesprávně detekován a aplikace pracující s výsledky, například firewall, není

[1] <https://www.symantec.com/connect/articles/perils-deep-packet-inspection>



Obr. 3.1: Škálování datového provozu[10].

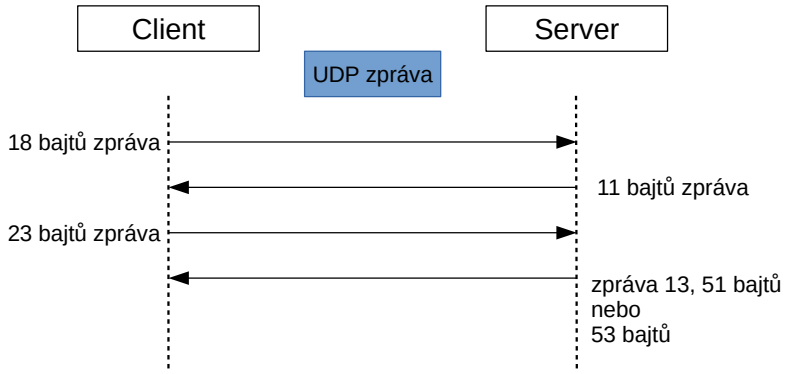
schopná včas případně vůbec detekovat škodlivý software. Pro zamezení vzniku false positive výsledků je nutné neustále udržovat aktuální databázi vzorů[10].

3.2 Využití

Hluboká inspekce paketů je využívána v oboru síťové bezpečnosti např. ke kontrole, verifikaci a filtraci datového provozu jako řada firewalů Sonic Wall od společnosti Dell nebo Cisco NBAR. Dále DPI využívají společnosti k interní kontrole provozu na vlastní síti, stejně tak bezpečnostní agentury pro monitorování národního internetového provozu. DPI je čím dál častěji využívána poskytovateli internetového připojení.

Síťová bezpečnost

Využívá jej systém NIDS (Network intrusion detection system) pro odhalení průniku. Jedná se o systém, který si monitoringem provozu na síti klade za cíl odhalit podezřelé aktivity, jako jsou malware, viry, červy, spyware, trojské koně a jiný škodlivý software.



Obr. 3.2: Analýza znaků chování přenosu.

Zpráva šířky pásma (řízení provozu), statistika provozu

Poskytovatelé internetového připojení využívají hlubokou inspekci paketů pro zajištění kvality služeb QoS (Quality of Service) zákazníkovi.

Pro marketingové potřeby

Při sledování provozu konkrétního uživatele jde o pokročilý on-line marketing. Slouží k vysledování zájmů uživatele a na základě této znalosti mu podstrkuje reklamu.

Dodržování vlastnických práv

Nelegální sdílení a distribuce hudby, videí či softwaru lze pomocí hluboké inspekce dat odhalit. Za účelem dodržování a vymáhání autorských práv vznikl registr Audible Magic[2], který má přes 11 milionů otisků dat. Tyto otisky pak slouží k identifikaci dat.

Vládní dozor a cenzura

Pomocí hluboké inspekce paketů mohou vlády, nebo vládní agentury realizovat zákonné sledovaní a záznam provozu na základě znalosti emailu, IP adresy, telefonního čísla atd. konkrétního uživatele. Této metody využívá například vládní projekt PRISM (Planning Tool for Resource Integration, Synchronization, and Management) v USA (United States of America), informace byla odhalena v úvodu zmíněným Edwardem Snowdenem. Stejně tak je možné odhalit nelegální síťový provoz.

[2]Společnost Audible Magic poskytuje službu rozpoznání autora digitálního mediálního obsahu. Využívá se např. k zamezení šíření nelegálních kopií nebo k vyúčtování za poskytnutá autorská práva.

3.2.1 Nástroje hluboké inspekce paketů

V současné době je k provádění hluboké inspekce paketů dostupných několik nástrojů. Dostupných je několik komerčních řešení, od společností je např. Dell (Sonic wall), Rohde & Schwarz (PACE), Cisco NBAR, NetFox Detective, Flowmon atd. Tyto společnosti poskytují svoje produkty pod licencí a své now-how nepublikují. Dále je možné najít několik publikovaných projektů pod licencí open-source, jsou to L7 Filter, nDPI, libprotoident a tstat. Vzhledem k tomu, že realizace hluboké inspekce paketů je velmi komplexní a rozsáhlá, ne všechny open-source nástroje splňují požadované vlastnosti[12].

3.2.2 Porovnání nástrojů

Pro porovnání nástrojů byla použita studie Measuring the Accuracy of Open-Source Payload-Based Traffic Classifiers Using Popular Internet Applications[12].

- **L7 Filter** jedná se o dříve populární nástroj, který provádí klasifikaci na základě předdefinovaných popisů vzorů. Nejnovější popisy jsou z roku 2009, nástroj již není dále vyvíjen a není možné jej použít, protože nereflektuje nástup zabezpečeného SSL/TLS spojení.
- **nDPI** využívá stejně jako L7 Filtr vzory popisů pro klasifikaci. Vzory jsou ručně napsány v jazyce C. Jedná se o stále vyvíjený nástroj. Velkou předností je schopnost identifikovat SSL/TLS spojení a dále jej klasifikovat. Nástroj rozpozná až 180 aplikačních protokolů a lze jej dále rozšiřovat.
- **libprotoident** nástroj je navržen tak, aby jej bylo možné použít tam, kde by hluboká inspekce paketů mohla znamenat narušení soukromí, nebo by byla výpočetně příliš náročná. Proces rozpoznávání je založen na prvních čtyřech bajtech aplikačního protokolu. Dovede klasifikovat až 300 aplikačních protokolů.
- **tstat** je nástroj, který primárně neslouží k provádění hluboké inspekce paketů a je určen pro širší analýzu internetového provozu. Oproti ostatním nástrojům nemá tak širokou podporu aplikačních protokolů.

Vyhodnocení

Vezme-li se v úvahu studie Shane Alcock a Richard Nelson, která byla především zaměřena na rozpoznávání aplikačních protokolů, je zde vyzdvihnut projekt nDPI a libprotoident. Kvalita projektu libprotoident spočívá v tom, že nevyžaduje velký výpočetní výkon. Nevýhodou je, že se hlouběji nezabývá hlubší inspekcí metadat. Výhoda nDPI je návaznost na projekt nTOP, který není jen klasifikátor, ale i komplexní nástroj síťové sondy. Nástroj nDPI má velmi propracované vzory aplikačních

protokolů, kde využívá metadata obsažená v certifikátech komunikujících stran pro klasifikaci aplikačních protokolů[12].

V práci je dále použita knihovna funkcí nDPI. Nástroj nDPI je velmi propracovaný a dobře rozšířitelný a poskytuje dostatečný prostor pro provedení dalších úprav tak, aby mohl plnit funkci detektoru a analyzátoru SSL/TLS spojení. Možnosti knihoven nDPI jsou v práci dále zkoumány a testovány.

3.2.3 Nástroj nDPI

Projekt nDPI vznikl původně z projektu OpenDPI, který se v roce 2011 rozdělil. Nově vznikl projet Ipoque PACE R&S a open-source nDPI. Nástroj nDPI zachoval původní způsob zpracování a využívá některé části z původního projektu OpenDPI. Dále je vyvíjen, rozšiřován a vylepšován. Knihovny projektu jsou napsány v jazyce C. Součástí zdrojových kódů je pcapreader[13], který je funkčním příkladem použití nDPI knihoven. Zpracování pcapreaderu lze rozdělit na tři základní části[11].

První vstupní část zpracovává zachycená data ve formátu pcap [3]. Pakety dekóduje na 2. - 4. vrstvě modelu ISO/OSI. Provádí se extrakce informací, jako je IP adresa, čísla použitých portů, TCP state atd. Pomocí zásuvných disektorů je detekováno asi 100 různých protokolů. Provádí se také první vyhodnocení na základě známých IP adres. Pokud je v této části spojení označeno jako známé(nalezené), již není dále zpracováváno, jen sledováno [11].

Druhá část provádí podrobnější analýzu jak dat z 3. a 4. vrstvy ISO/OSI, tak dat z aplikační 7. vrstvy. K tomu používá vytvořené vzory datových toků, které jsou pro konkrétní aplikační protokoly charakteristické. K prohledávání a porovnávání vzorů aplikačních protokolů je použit Aho-Corasick algoritmus[4][11].

Třetí část vyhodnocení zkouší přiřadit datovým tokům, které nebyly v první a druhé části rozpoznány, protokol dle použitého portu. Takto vyhodnocené protokoly jsou označeny a zobrazeny zvlášť. Funkci třetího hodnocení lze přepínačem -d zakázat.

Detekce a klasifikace šifrovaných spojení nástrojem nDPI probíhá na základě vyhodnocení paketů Client_Hello, Server_Hello. Šifrované spojení lze tedy detekovat, jen pokud je v průběhu zachytávání dat spojení zahájeno. Vyhodnocení výskytu zabezpečeného spojení pouze na základě použitých portů není příliš přesná metoda,

[3] Základní datový formát pro ukládání zachycených síťových dat do souboru více zde <`https://wiki.wireshark.org/Development/LibpcapFileFormat`>

[4] Aho-Corasick algoritmus je vyhledávací algoritmus vynalezený Alfredem Ahem a Margaret J. Corasickovou. Je to druh slovníkového vyhledávacího algoritmu, který ve vstupním textu hledá prvky konečné množiny řetězců. Vyhledává všechny prvky množiny najednou, jeho asymptotická složitost je proto lineární k délce všech vyhledávaných prvků plus délce vstupního textu plus délce výstupu.

ale lze ji použít v třetí části rozpoznání [11].

3.3 pcapreader

Knihovny projektu nDPI, včetně zdrojového kódu pcapreader, jsou dostupné na [5]. Zdrojový kód je napsán multiplatformně, je možné provést kompilaci v operačním systému Linux i Windows.

3.3.1 Kompilace

Kompilace byla provedena v prostředí Linux a provádí se následujícím způsobem. Pokud nejsou nainstalovány knihovny libpcap, je nutné je doplnit. Kompilaci lze provést pomocí spouštění příkazů v shellu prostřednictvím terminálu. V dalším kroku, po přepnutí do adresáře se zdrojovým kódem, je nutné nejprve spustit příkaz ./configure a následně make. Pokud by chyběly nějaké knihovny, bude nás o tom kompilátor informovat. V případě, že má ale všechny, které potřebuje, proběhne kompilace úspěšně a vytvoří binární soubor pcapreader.bin, který již lze spustit.

Základní funkce

Program pcapreader se spouští pomocí parametrů. Základní funkcí je detekce a klasifikace na živých nebo uložených datech. Data jsou programem pcapreader čtena z datového formátu pcap. Živé zachytávání je prováděno pomocí knihovny libpcap nebo winpcap, která zachycená data převádí do formátu pcap.

```
./pcapreader -i eth0 -s 20
```

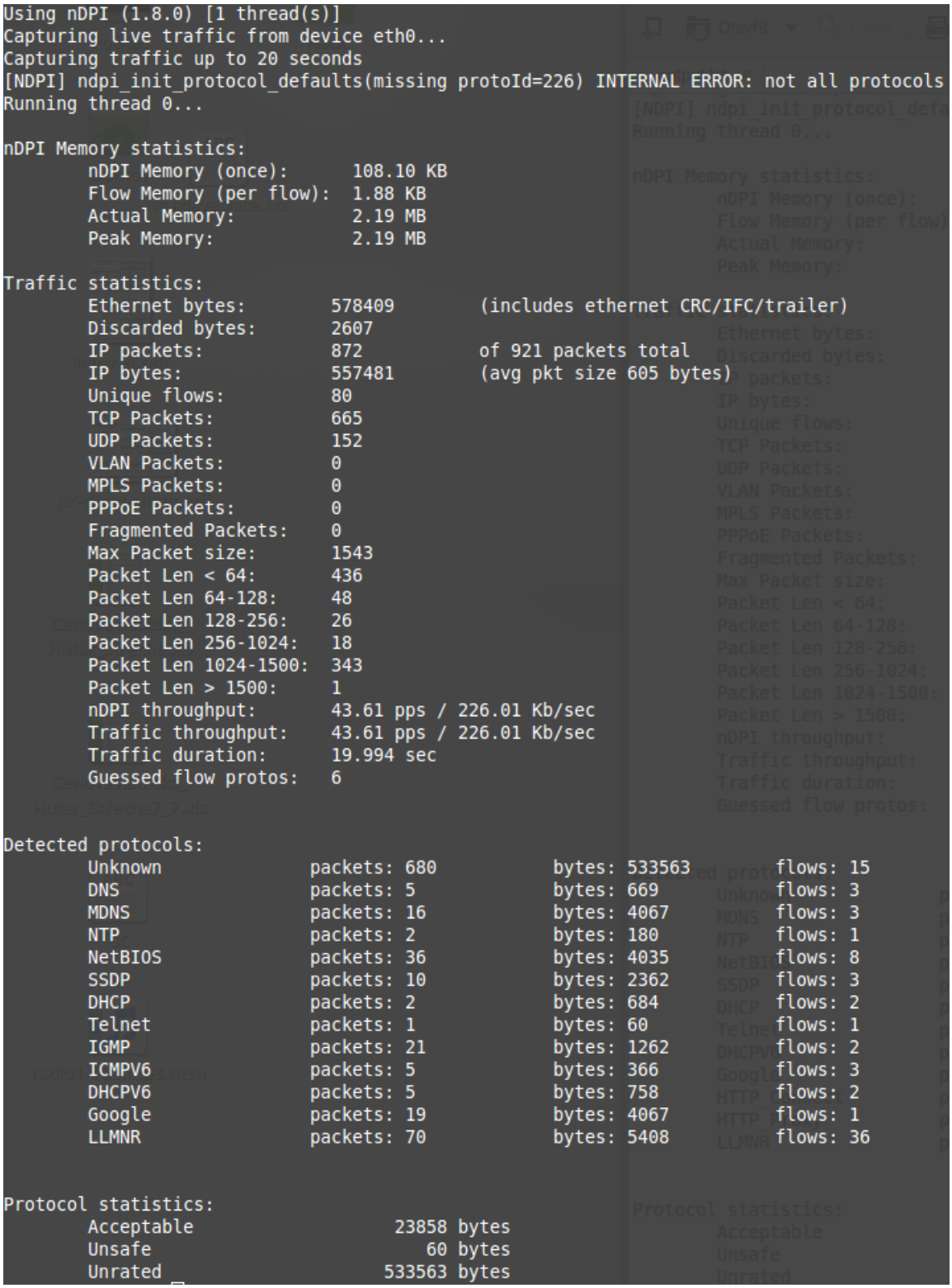
```
./pcapreader -i /var/tmp/capture.pcap
```

Parametr -i určuje místo, odkud budou čtena data, lze zadat síťové zařízení nebo cestu k souboru. Parametr -s určuje dobu záchytu dat v případě živého zpracování dat.

Základní knihovna nDPI podporuje rozpoznání asi 180 možných protokolů a podprotokolů. Tuto množinu lze rozšířit o další možné protokoly pomocí doplnění vlastní specifikace dle uvedeného vzoru v návodu.

Výstup z pcapreader v základu vypadá jako na obr. 3.3. Poskytuje statistiku zpracovaných dat a detekované protokoly. Z celkového počtu 80 datových toků bylo 15 označeno jako Unknown - neznámé nebo nedetekované.

[5] <https://sourceforge.net/projects/ntop/files/nDPI/>



Obr. 3.3: Výstup z pcapreader.

4 DETEKCE A ANALÝZA SSL/TLS

Kapitola Detekce a analýza SSL/TLS se v první části věnuje návrhu softwarového nástroje. Po té následuje podrobnější popis vnitřní struktury a zpracování dat programem pcapReader. V kapitole je provedena analýza Handshake zpráv SSL/TLS protokolů a protokolů poštovních služeb používajících STARTTLS. Dále se věnuje popisu implementace úprav[1] do projektu nDPI a programu pcapReader, postupu extrakce metadat a vyhodnocení výsledků zpracovaných dat. Je zde uvedeno zdůvodnění úpravy návrhu. Na závěr je provedena analýza získaných metadat a provedeno testování upraveného programu.

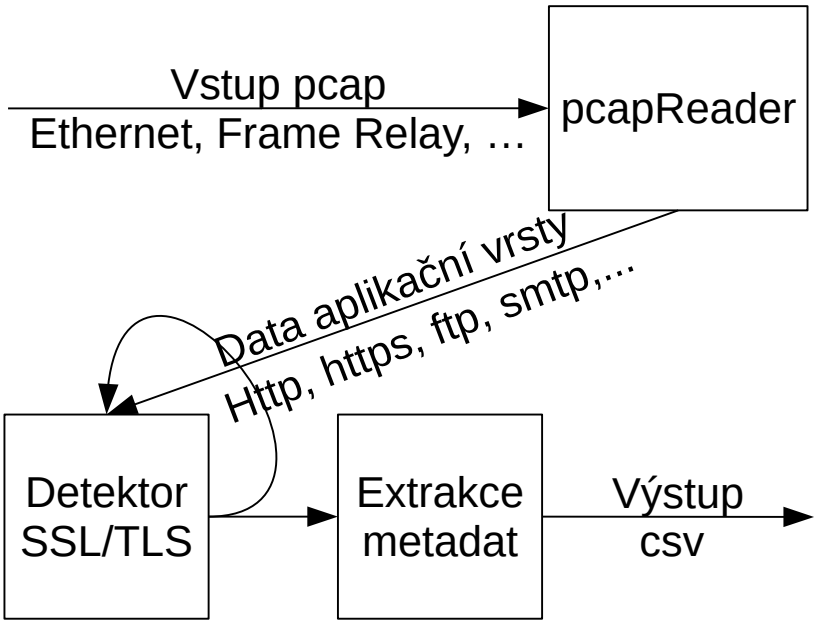
4.1 Návrh

Cílem práce je vytvořit softwarový nástroj, který bude schopen detekovat a extrahovat metadata z SSL/TLS tak, aby je bylo možné použít pro analýzu a filtraci zabezpečených spojení, viz obr. 4.1.

Kritéria

- **Detekce simplexních a duplexních spojení** – implementace úprav získaných vlastní analýzou paketů do zdrojových kódů projektu nDPI: pcapreader.c, ssl.c, mail_smtp.c, mail_pop.c, mail_imap.c.
- **Detekce poštovních služeb SMTP, POP3, IMAP v případě použití STARTTLS** – implementace úprav získaných vlastní analýzou paketů do zdrojových kódů projektu nDPI: pcapreader.c, ssl.c, mail_smtp.c, mail_pop.c, mail_imap.c.
- **Extrakce vybraných metadat** – použitím programu tshark, po provedení počátečních testů řešení zavrženo, viz odstavec 4.3. Návrh upraven na rozšíření zdrojového kódů ssl.c a pcapreader.c projektu nDPI.
- **Vyhodnocení detekovaných spojení** – rozšířením zobrazených výstupních dat o informace SSL/TLS výstupu pcapreader.
- **Export metadat** – vytvořením strukturovaného souboru oddělovaného čárkou ve formátu csv obsahující získaná metadata.

[1] V částech zdrojového kódu projektu nDPI, kde je provedena úprava je část kódu označena komentářem „Hutar“.



Obr. 4.1: Schéma detektoru/analyzátoru.

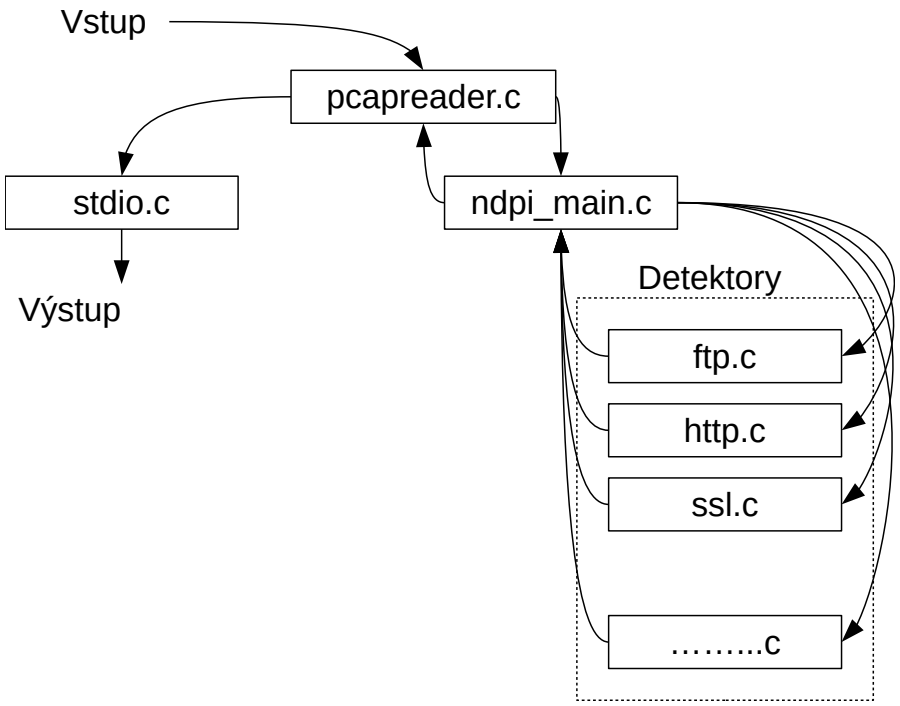
4.1.1 nDPI vnitřní struktura zpracování dat

Program pcapReader, viz obr. 4.2 provádí čtení vstupních dat ve formátu pcap postupně tak, jak jsou uložena v souboru nebo zachycena. Po načtení příchozího paketu provede pcapreader.c načtení dat do prázdného objektu obsahujícího datovou strukturu flow[2] a porovná je s objekty již uloženými. Pokud nalezne shodu, některá data ke spojení aktualizuje daty z paketu, např. velikost paketu, pořadová a sekvenční čísla, atd. V případě, že je v datové struktuře pro konkrétní spojení již přiřazen protokol, tak toto spojení jen sleduje a neposílá dále k detekci. Pokud je spojení nové a není nalezena shoda, vytváří se nový objekt včetně struktury flow naplněné aktuálními daty. Základní limit je nastaven na 2 miliony jedinečných datových toků.

Princip práce knihovny nDPI je následující. Vstupní data jsou načtena ndpi_reader.c do prázdné datové struktury flow. Dále jsou uložená data ve struktuře flow porovnávána s již předchozími načtenými. Pokud není nalezena shoda, je tato struktura uložena jako nová. Pokud shoda je, tak jsou některá data do nalezené stávající struktury nahrána. Dále je prováděna kontrola, zda-li již není detekce ukončena. V takovém případě se takový datový tok pouze sleduje a dále se nezpracovává. Nemá-li příchozí paket detekci ukončenou, pokračuje jeho zpracování v ndpi_main.c.

Funkce v ndpi_main.c zjednodušeně řečeno zkouší poslat paket jednotlivým detektorům na základě porovnání určité charakteristiky uložené ve struktuře flow.

[2]Základní datová struktura objektu obsahuje například: zdrojovou a cílovou IP adresu, zdrojový a cílový port, index detekovaného protokolu, datovou strukturu ndpi_flow.



Obr. 4.2: Vnitřní struktura programu pcapReader.

Detektory reprezentují jednotlivé protokoly nebo služby, které lze detekovat a mají vlastní zdrojový kód, který je dle protokolu nebo služby pojmenován, např. ssl.c, dns, mail_smtp.c. Zdrojový kód je napsaný v jazyce C. Pokud je potřeba upravit vyhledávaní některého protokolu nebo aplikace, provádí se úprava zdrojového kódu jednotlivého detektoru.

Vyhodnocení detektorem se ukládá do flow struktury. V případě úspěchu je detekce ukončena. Pokud není žádným klasifikátorem nalezena shoda, detekce je také ukončena. Na konci vyhledávání jsou data paketu zahozena a zůstávají pouze data uložená ve struktuře flow a na vstup je načten další paket.

4.2 Detekce

Knihovna funkcí nDPI provádí detekci ve třech krocích, jak bylo popsáno v kapitole 3.2.3.. Aby program podrobil všechny pakety druhému kroku analýzy a mohl tak detekovat SSL/TLS spojení, musí být první vyhodnocení přeskočeno.

Vyhodnocení dle množiny známých IP adres provádí detektor tcp_udp.c. Zdrojový kód detektoru byl upraven tak, aby vracel neúspěch detekce. Vyhledávání dále pokračuje podrobnější detekcí.

Druh spojení	Název zprávy
Duplex	Client_hello, Server_hello, Certificate
Simplex server	Server_hello, Certificate
Simplex client	Client_hello, Client_Key_Exchange

Tab. 4.1: Tabulka zpráv SSL/TLS pro přesnější detekci a extrakci metadat

4.2.1 Simplexní a duplexní spojení SSL/TLS

Nejprve byla provedena analýza zdrojového kódu ssl.c. Zdrojový kód obsahuje funkce k rozkladu zprávy Client_Hello a zprávy Server_Hello. Umí také nalézt data rozšíření Server_name zprávy Client_hello. Pokud spojení obsahuje certifikát, lze z něj vybrat název serveru, kterému byl vystaven. Dále provádí další prohledávání na některé další služby používající SSL/TLS spojení, např. whatsapp, jabber atd. Pro potřeby práce to není dále využitelné.

Vyhodnocení je nastaveno tak, že pokud dojde k nálezu zprávy Client_Hello, je spojení označeno jako SSL a detekce pro konkrétní spojení je ukončena. V případě, že první bude nalezen paket Server_Hello, postup je totožný jako v předchozím případě. Taková detekce je pro potřeby detektoru nedostačující a je nutné ji upravit.

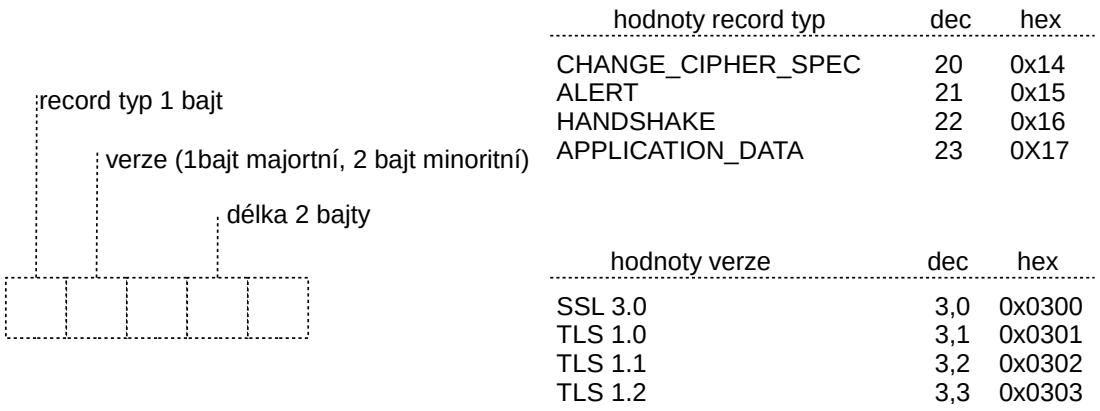
Popis simplex, duplex

- **Duplexní spojení** je takové, které obsahuje pakety z obou stran spojení, tedy jak ze strany klienta, tak ze serveru viz obr. 1.2.
- **Simplexní spojení** je takové, které obsahuje data pouze jedné z komunikujících stran, tedy pakety odesílané jen za strany klienta, nebo jen ze strany serveru.

Pokud chceme pouze detekovat začátek spojení, tak stačí zachytit jeden paket z jakéhokoliv směru. Chceme-li dále určit simplex/duplex a směr, je nutné zachytit pakety minimálně dva.

Analýza SSL/TLS zpráv

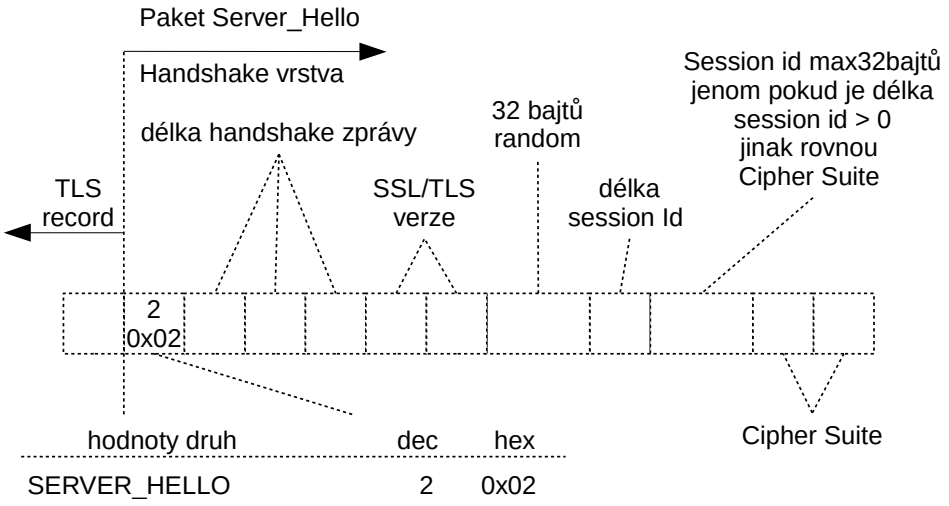
Každý paket SSL/TLS zabezpečeného spojení obsahuje hlavičku TLS Record viz obr. 4.3. Dále zprávy protokolu Handshake obsahují vrstvu handshake, kde jsou definovány konkrétní druhy zpráv v paketu. Na níže uvedených obrázcích je proveden základní rozbor zprávy. Zmíněno je především několik důležitých bajtů, které jsou při rozpoznávání vyhledávány.



Obr. 4.3: Hlavička TLS record.

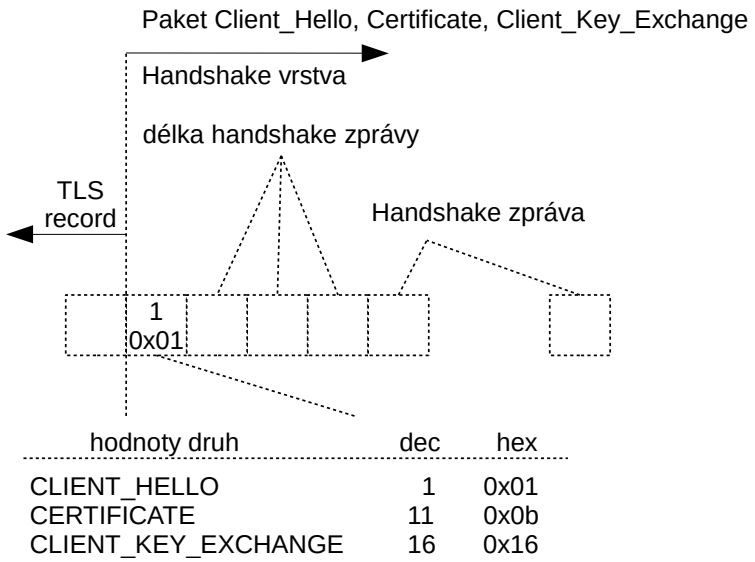
Úpravy zdrojového kódu simplex, duplex

Do zdrojového kódu pcap_reader.c byla doplněna podmínková logika, která zkouší nalézt další doplňující zprávy k danému spojení. Pro vyhodnocování podmínek byla rozšířena datová struktura ndpi_flow–>tcp.protos.ssl o značky viz tab. 4.3, které jsou změněny z hodnoty NULL na TRUE, pokud je daný paket nalezen. Seznam doplněných proměnných client_hello, server_hello, server_hello_certificate, client_key_exchange viz tab. 4.3. Dále byla vytvořena proměnná sslloop viz tab. 4.3 pro počítání následujících zpráv po prvním nalezeném.



Obr. 4.4: Zpráva Server_Hello TLS record.

Níže uvedená tabulka 4.2 slouží jako pravdivostní tabulka, podle které byly doplněny podmínky do zdrojového kódu ndpi_reader.c. Jsou-li splněny dvě podmínky, provede se vyhodnocení druhu spojení.



Obr. 4.5: Paket Client_Hello, Certificate, Client_Key_Exchange.

Zpráva	**Duplex**	**Simplex Server**	**Simplex Client**
Client_Hello	TRUE	NULL	TRUE
Server_Hello	TRUE	TRUE	NULL
Client_Key_Exchange	—	NULL	TRUE
—	sslloop >= 3	sslloop >= 3	sslloop >= 3

Tab. 4.2: Tabulka podmínek hledání zpráv.

Vyhledávací řetězec zprávy Server_Hello, viz obr 4.4, byl doplněn o hledání verze z TLS record hlavičky viz obr. 4.3. Dále byla doplněna podmínka hledání dvou bajtů obsahujících hodnoty použitých šifrovacích algoritmů. Do podmínky vyhledávání byl zahrnut bajt udávající délku session id viz obr. 4.4. Má-li tento bajt nulovou hodnotu, hned za ním následuje dvojice bajtů obsahujících označení šifrovacích algoritmů, jinak je nutné přeskočit čtení o tolik bajtů, kolik udává hodnota bajtu session Id a tam přečíst hledanou dvojici bajtů viz obr. 4.4.

Detektor ssl.c byl rozšířen o datové pole obsahující všechny aktuální šifrovací algoritmy dle IANA publikované zde [3]. Součet dvou čísel šestnáctkové soustavy reprezentujících dva bajty Cipher Suite odpovídá položce v nově vytvořeném datovém poli.

Dle výše uvedené analýzy zprávy SSL/TLS byla do zdrojového kódu ssl.c doplněna vyhledávací podmínka pro zprávu Client_Key_Exchange viz obr. 4.5. Z TLS record hlavičky zprávy byla následně odečtena verze použitého SSL/TLS. Hodnotu

[3]https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

Datový formát	Název proměné
char	server_version[10] client_version[10] cipher_suite[64]
int	sslloop client_hello server_hello client_key_exchange ssl_count

Tab. 4.3: Tabulka rozšíření struktury ndpi_flow->protos.tcp.ssl.

verze v TLS record viz obr. 4.3 u zprávy Client_Hello nelze brát jako použitou, protože je pouze klientem navržena a upřesněna je až serverem ve zprávě Server_Hello.

Příklad části zprávy obsahující zprávu Client_Key_Exchange viz obr. 4.5 s označenými vyhledávanými bajty. Červenou barvou je označen bajt record typ, další dva bajty označené modrou barvou značí použitou verzi. Následují dva neoznačené bajty nesoucí hodnotu délky. Červeně označený bajt má v šestnáctkové soustavě hodnotu 10 a v desítkové soustavě má tedy hodnotu 16, značí zprávu Client_Key_Exchange.

```
0020 ...
0030 16 03 03 00 46 10 00 00 42 41 04 5a d6 90 68 17
0040 c8 ff b0 c6 29 8a 1f 2a 75 0d e3 be 8f 79 5a fd
0050 ...
```

4.2.2 Detekce poštovních služeb SMTP, POP3, IMAP v případě použití STARTTLS

Analýzou funkcí pro detekci pošty původního zdrojového kódu bylo zjištěno, že detekce poštovních služeb je ukončena v momentě nalezení počátečních zpráv, tedy paketů, které obsahují ustanovení poštovního spojení (nalezením dvou zpráv). Pro použití rozšíření STARTTLS viz obr. 1.4 je typické, že je použito až po vytvoření základního poštovního spojení a tedy například ve třetí a pozdější zprávě. V době, kdy byl přenášen paket s oznámením STARTTLS, bylo již spojení detekováno. Tím, že STARTTLS probíhá na stejném portu, nebylo tedy ani detekováno SSL/TLS spojení, které bylo následně ustanoveno.

Zdrojové kódy detektorů poštovních služeb jsou mail_smtp.c, mail_pop.c a mail_imap.c. Detektory poštovních služeb využívají pro detekci takzvané Bitmasky, kde probíhá nastavování jednotlivých bitů dle masky nalezených signatur při ana-

Druh spojení	Název zprávy
Duplex	220, EHLO, „250 STARTTLS“ nebo „250 X-ANONYMOUSTLS“, STARTTLS nebo „X-ANONYMOUSTLS“, 220
Simplex server	220, „250 STARTTLS“ nebo „250 X-ANONYMOUSTLS“, 220
Simplex client	EHLO, STARTTLS nebo „X-ANONYMOUSTLS“

Tab. 4.4: Tabulka zpráv SMTP pro přesnější detekci a extrakci metadat

lýze zprávy. Každý typ zprávy má předem definovanou hodnotu tak, aby odpovídal jednomu konkrétnímu bitu z bajtu. Vyhodnocení a určení protokolu je následně prováděno pomocí podmínky určené počtem nastavených bitů na hodnotu jedna.

Nalezení poštovní služby využívající STARTTLS je rozlišeno použitím označení SMTPS, POPS a IMAPS ve výsledném výpisu nalezených protokolů.

Analýza SMTP, POP3 a IMAP za použití STARTTLS

Stejně jako je pro SSL/TLS spojení potřeba upravit detekci tak, aby byla možná pro duplexní i simplexní spojení, platí i pro uvedené protokoly. Níže je uveden rozbor a provedené úpravy pro každý protokol zvlášť.

SMTP

Rozšíření standardního protokolu SMTP o STARTTLS je popsáno v RFC 3207[3]. Schéma realizace spojení viz obr. 1.4. Poštovní služba MS-Exchange využívá vlastní modifikaci nazývanou „X-ANONYMOUSTLS“. Schéma je shodné s použitím zprávy STARTTLS s tím rozdílem, že je použita zpráva označená „250 X-ANONYMOUSTLS“ a „X-ANONYMOSTLS“. Potřebné pakety pro přesnější detekci a extrakci metadat.

Analýza zpráv SMTP mezi klientem a serverem

Zprávy používané mezi komunikujícími stranami mají danou strukturu. Každý řádek zprávy je ukončen znaky CR LR (v šestnáctkové soustavě 0x0d0a). Pomocí těchto znaků lze přenášenou zprávu rozdělit na jednotlivé informační řádky. Zprávy ze strany serveru jsou zpravidla uvozovány kódem trojmístého čísla, např. 220 [4],250 [5] atd. Za číslem následuje pomlčka a zpráva určená klientovi. Pokud za číslem není uvedena pomlčka, ale mezera jedná se o poslední informační řádek.

[4]Service ready

[5]Requested mail action okay, completed

Název zprávy	Hodnota bitu (hex)
SMTPS_BIT_CLIENT_STARTTLS	0x01
SMTPS_BIT_READY_TO_TLS	0x02
SMTPS_BIT_TLS_NOT_AVAILABLE	0x04
SMTPS_BIT_SYNTAX_ERROR_501	0x08
SMTPS_BIT_CLIENT_EHLO	0x10
SMTPS_BIT_CLIENT_X_ANONYMOUSTTLS	0x20

Tab. 4.5: Tabulka nové Bitmasky ndpi_flow->l4.tcp.smtps_command_bitmask

Použití STARTTLS je součástí rozšířené sady služeb ze strany serveru ESMTP (Extended Simple Mail Transfer Protocol)[6]. Sada ESMTP je posílána klientovi vždy, když si o ni požádá dotazem EHLO. Po obdržení této zprávy pošle server v jedné zprávě své schopnosti začínající trojčíslím 250. Klient si vybere požadovanou službu a tou odpoví serveru bez prefixu čísel[7].

Úpravy detektoru mail_smtp.c

Detektor používal pouze jednu Bitmasku umístěnou ve struktuře ndpi_floe–>l4.tcp.smtp_command_bitmask, která byla již plně vyčerpána na detekci zpráv SMTP. Stávající Bitmaska obsahuje jeden bit pro zprávu STARTTLS, tento bit byl využit i pro zprávu „X-ANONYMOUSTLS“. Pro detekci dalších zpráv z sady ESMTP byla struktura ndpi_flow–>l4.tcp rozšířena o novou Bitmasku smtps_command_bitmask. Do této nové struktury byly přidány položky uvedené v tabulce 4.5.

Nová Bitmaska obsahuje také bity s kódem zprávy 454[8] a 501[9] indikující neúspěch STARTTLS dle RFC 3207 [3]. Zpracování těchto bitů je pouze informativní a mohlo by sloužit dalšímu rozvoji detektoru.

Detektor byl dále rozšířen o vyhledávací řetězce „250 X-ANONYMOUSTLS“ a „X-ANONYMOUSTLS“.

Vyhledávací podmínky čísla zprávy byly rozšířeny o hledání odpovědi 220 na klientskou zprávu STARTTLS nebo X-ANONYMOUSTLS. Tato nalezená zpráva značí připravenost serveru na SSL/TLS navázání spojení.

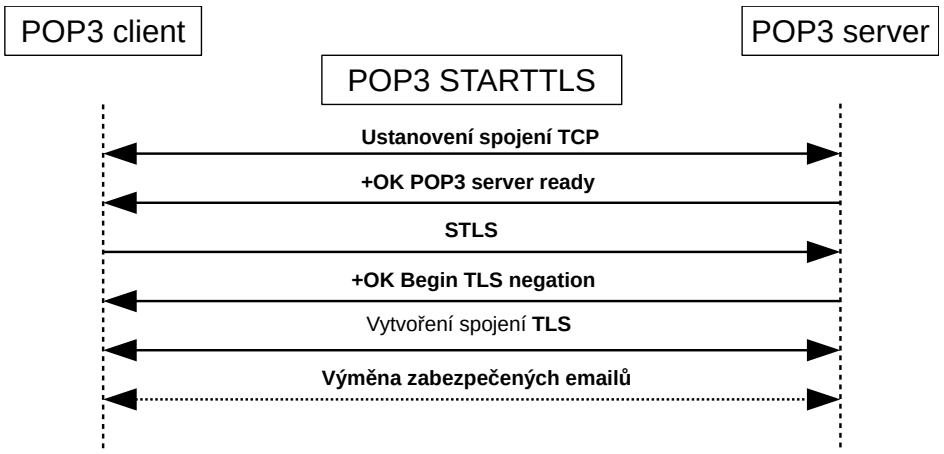
Doplněny byly podmínky vyhodnocení na základě bitů nastavených na hodnotu jedna v obou Bitmaskách. Podmínky vyhodnocení pracují s možností, že by spojení

[6]je rozšířením protokolu SMTP. Zavádí nové zprávy a rozšiřuje práci s protokolem SMTP.
[7]Platí i pro rozšíření X-ANONYMOUSTLS
[8]TLS not available due to temporary reason.
[9]Syntax error (no parameters allowed).

bylo pouze jednosměrné a jsou nastaveny tak, aby poznaly směr spojení. V tabulce 4.4 je uveden přehled zpráv, které určují vyhodnocovací podmínky pro protokol SMTPS.

POP3

Použití rozšíření STARTTLS v protokolu POP3 je uvedeno v RFC 2595 [4], viz obr. 4.6 výměny zpráv.



Obr. 4.6: Přenos pošty protokolem POP3 za použití STARTTLS.

Analýza zpráv POP3

Na základním schématu spojení jsou uvedeny zprávy tak, jak je v doporučení RFC. Základem zpráv od serveru ke klientovi je použití úvodních znaků zprávy a to „+OK“ a „-ERR“. Zprávy klienta jsou rozšířeny o STLS. Tato zpráva značí požadavek klienta na následné použití SSL/TLS spojení. Pokud na tuto zprávu server odpoví „+OK“ případně „+OK Begin TLS“, následuje již jako další zprávy SSL/TLS. Ukončení informačních řádků v protokolu POP3 využívá obdobný způsob jako výše uvedený protokol SMTP.

Analýzou reálného vzorku získaného komunikací s poštovním serverem seznam.cz bylo zjištěno, že je využíváno zprávy ze strany klienta „CAPA“. Na zprávu „CAPA“ server reaguje „+OK Capability“ a seznamem svých schopností. Z takto obdržené zprávy si klient vybere požadovanou, v tomto případě „STLS“. Dále celá komunikace probíhá již jak je uvedeno na obr. 4.6.

Druh spojení	Název zprávy
Duplex	+OK, CAPA nebo STLS, „+OK Begin TLS“,
Simplex server	+OK, „+OK Capability“ obsahující zprávu STLS nebo „+OK Begin TLS“
Simplex client	CAPA, STLS

Tab. 4.6: Tabulka zpráv POP3 pro přesnější detekci a extrakci metadat

Název zprávy	Hodnota bitu (hex)
POP_BIT_Server_STLS	0x0200
POP_BIT_Begin_TLS	0x0600
POP_BIT_OK	0x0800
POP_BIT_ERR	0x1000
POP_BIT_QUIT	0x4000

Tab. 4.7: Tabulka doplnění Bitmaskyndpi_flow->l4.tcp.pop_command_bitmask

Úpravy detektoru mail_pop.c

Detektor poštovní služby POP3 neměl obsazenou celou Bitmasku a tak byla stávající ndpi_flow–>l4.tcp.pop_command_bitmask doplněna, viz tab. 4.7. Byl doplněn vyhledávací řetězec na hledání zprávy „+OK Begin“, hledání „STLS“ v nabídce Capability ze strany serveru. Dle tabulky 4.6 byly sestaveny vyhodnocovací podmínky pro protokol POPS.

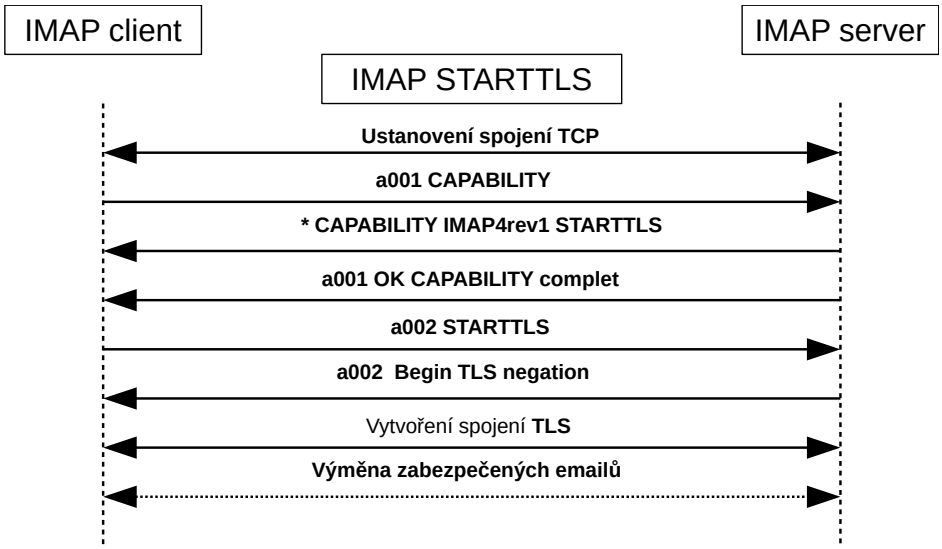
IMAP

Rozšíření protokolu IMAP o STARTTLS je v RFC [4] základní schéma, viz obr. 4.7. Při provádění detekce přenášených zpráv se číslování uvedené v analýze zpráv nebere v potaz a pomocí vhodné funkce je pouze přeskočeno tak, aby byla nalezena a přečtena informace přenášená ve zprávě.

IMAP

Analýza zpráv IMAP

Protokol IMAP podporuje velké množství komunikačních zpráv dle RFC 3501. Pro udržení přehledu používá systém pořadového číslování vyměňovaných zpráv v úvodní části zprávy. Zpravidla pošle klient zprávu serveru s číslem a server svou odpověď na tuto zprávu opatří stejným úvodním číslem, případně doplní hvězdičkou. Za úvodní číselnou zprávou je použit znak mezery a pak je teprve doplněna informace zprávy.



Obr. 4.7: Přenos pošty protokolem IMAP za použití STARTTLS.

Druh spojení	**Název zprávy**
Duplex	CAPABILITY, OK CAPABILITY nebo STARTTLS, Begin TLS,
Simplex server	OK CAPABILITY, Begin TLS
Simplex client	CAPABILITY, STARTTLS

Tab. 4.8: Tabulka zpráv IMAP pro přesnější detekci a extrakci metadat

Při použití STARTTLS se zpravidla dodržuje výměna informačních zpráv, uvedená na obr. 4.7. Klient se zeptá serveru, podporuje-li schopnosti „CAPABILITY“. Pokud ano, server odpoví „CAPABILITY IMAP4rev1 STARTTLS LOGINDISABLED“. Klient posílá serveru vybranou schopnost tedy STARTTLS. Server potvrzuje „OK Begin TLS negotiation“.

Úpravy detektoru mail_imap.c

Detektor nepracoval s použitím masek, ale se značkami, byl nově upraven tak, aby detekce probíhala obdobným způsobem jako u protokolů SMTP a POP za pomocí Bitmasky. Byla vytvořena nová Bitmaska ndpi_flow–>l4.tcp.imaps_command_bitmask, viz tab. 4.9, původní značky imap a imaps byly zachovány a detekce byla zpřesněna použitím Bitmasky. Vyhledávaní bylo rozšířeno o „OK STARTTLS“ a „Begin TLS“. Dle tabulky 4.8 byly sestaveny vyhodnocovací podmínky kombinace Bitmasky a značek pro protokol IMAPS.

Název zprávy	Hodnota bitu (hex)
IMAP_BIT_CAPABILITY	0x01
IMAP_BIT_STARTTLS	0x02
IMAP_BIT_OK_STARTTLS	0x04
IMAP_BIT_OK_Begin_TLS	0x08

Tab. 4.9: Tabulka nové Bitmasky ndpi_flow–>l4.tcp.imaps_command_bitmask

4.2.3 Detekce SSL/TLS spojení poštovních služeb SMTP, POP3, IMAP v případě použití STARTTLS

Funkce ve zdrojovém kódu knihovny nDPI nepočítá s hledáním SSL/TLS spojení u protokolů poštovních služeb, proto bylo potřeba tuto funkci doplnit.

Doplnění podmínky v případě detekování protokolů označených jako SMTPS, POPS a IMAPS bylo do funkce packet_processing. Splněná podmínka smaže informaci o ukončené detekci, informaci o nalezeném protokolu uloží do druhé položky datového pole ndpi_flow–>protocol_stack_info a stávající hodnotu prvního pole přepíše na neznámý protokol. Dále byla provedena úprava ve zdrojovém kódu ndpi_main.c. Do funkce ndpi_detection_process_packet, která provádí rozesílání paketů do detektorů, byla zapracována podmínka, která paket označený již nalezenou poštovní službou v druhém poli ndpi_flow–>protocol_stack_info rovnou pošle do detektoru ssl.c. Dále zpracování probíhá jako u běžného zabezpečeného spojení popsaného v kapitole 4.2.1.

Po ukončené detekci SSL/TLS a extrakci metadat je spojení nazpět označeno danou poštovní službou.

4.2.4 Vyhodnocení zpracovaných metadat

Vyhodnocení lze provést pomocí souhrnných informací zobrazených po ukončení detekce a po podrobnějším zpracování dat z csv souboru, kam jsou metadata uložena.

Souhrn zobrazených dat je rozšířen o položku SSL/TLS a Cipher Suite. Položka SSL/TLS obsahuje počet detekovaných zabezpečených spojení a verze použitých SSL/TLS spojení. Položka Cipher Suite obsahuje souhrn použitých šifrovacích algoritmů. Příklad rozšířeného výpisu.

```
--------------SSL\TLS--------------
        SSL flows duplex:  72
        SSL flows simplex: 0
        SSL version TLS 1.0 handshake: 25
        SSL version TLS 1.2 handshake: 47
-----------Cipher Suite-----------
        TLS_RSA_WITH_AES_128_CBC_SHA              25
        TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256   20
        TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256     16
        TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384      3
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384      8
--------------------------------
```

Ze simplexních spojení nelze získat všechna požadovaná metadata. Simplexní spojení ze strany serveru neobsahují informaci požadovaného certifikátu z klientské strany. Nelze tedy porovnat, jestli certifikát posílaný ze strany serveru odpovídá požadovanému certifikátu. Informace v rozšířené části zprávy Client_Hello o názvu serveru, ale není povinná a v testovaných datech se nevyskytovala přibližně v 30% zpracovaných dat.

Dat ze simplexního spojení jen z klientské strany lze získat málo. Není možné získat verzi použitého šifrovacího algoritmu, ani nelze získat certifikát. Posílat certifikát z klientské strany není obvyklé ani povinné. Pokud tedy analyzujeme spojení z klientské strany, získáme jen časovou značku analyzovaného paketu, IP adresy, porty a verzi použitého zabezpečeného spojení ze zprávy Client_Key_Exchange. Verze by bylo možné získat případně i z jakékoliv další zprávy odeslané v tomto zabezpečeném spojení.

Program pcapReader sám o sobě provádí výpis detekovaných protokolů. V této skupině detekovaných dat se nachází i protokol SSL. Hodnota výstupu se ale nemusí shodovat s hodnotou v nově rozšířené částu SSL/TLS a to z důvodů, že program vyhodnocuje jeden druh spojení na konkrétní IP adrese a portu a pokud je navázání spojení SSL/TLS v průběhu opakováno program již nevyhodnocuje, oproti novému rozšíření, které zaznamenává každé spojení.

4.2.5 Export metadat

Všechna získaná metadata jsou okamžitě zpracována a uložena do výstupního csv souboru. Při každém spuštění programu nDPI se generuje nový soubor. Název souboru reprezentuje čas a datum spuštění detektoru a je v následujícím formátu: RokMěsícDen_HodinaMinutaSekunda.csv tedy RRRRMMDD_HHMMSS.csv, například

20170430_194133.csv.

Vybraná metadata jsou uvedena v kapitole 2.2 Vybraná metadata z SSL/TLS spojení. Tato metadata byla rozšířena o hodnotu času nalezeného spojení a rozlišení, o jaký typ spojení se jedná, tedy duplex, simplex klient a simplex server.

Časová značka, kdy bylo spojení realizováno, je velmi důležitá, protože na konkrétní kombinaci IP adresy a portu může probíhat více rozdílných spojení. Pro filtraci jednotlivých spojení je čas výskytu začátku SSL/TLS spojení důležitý. Informace o času každého paketu je uložena v hlavičce datového formátu pcap. Datový formát pcap obsahuje čas záchytu uloženého do 64 bitů ve struktuře timeval. Čas ve struktuře timaval je uložen v sekundách a mikrosekundách. Počátek času se počítá jako unixového času a to od 1. 1. 1970.

Hodnota času je tedy převedena do formátu obdobného pro pojmenování souboru z aktuálního času zpracování. Časová informace je doplněna i o hodnotu mikrosekund. Čas je tedy ve tvaru RRRRMMDD_HHMMSS_Usec.

Samotný export metadat do výstupního souboru probíhá vždy při vyhodnocení nalezeného SSL/TLS spojení. Již v průběhu zpracování dat lze prohlížet obsah souboru. Pokud nebude žádné SSL/TLS spojení detekováno, soubor nebude vytvořen. Soubor je vytvořen do stejné složky, kde je uložený samotný program.

Metadata jsou do csv ukládána v tomto pořadí: čas, IP adresa zdroje, port zdroje, IP adresa cíle, port cíle, verze získaná ze zprávy Server_Hello, verze získaná Client_Key_Exchange, šifrovací algoritmus, informace o typu spojeni (duplex, simplex server, simplex klient). Příklad získaných metadat ze serveru vutbr.cz. Pokud je spojení vyhodnoceno jako duplexní, již není vyhledávána verze SSL/TLS na straně klienta.

```
20170430_090623_996115, 192.168.2.6, 39590, 147.229.2.90,
    443, TLS 1.0,, www.vutbr.cz, www.vutbr.cz
   TLS_RSA_WITH_AES_128_CBC_SHA, Duplex
```

Kompilace programu a úpravy

Verze 1.4 je použitou výchozí verzí knihoven funkcí nDPI. Výsledný program nDPI.exe je 32bitový a je kompilován prostřednictvím Visual Studia 2015 pro operační systém Windows.

4.3 Tshark

Program tshark je verze programu Wireshark určeného do příkazového řádku. Je navržený k zachytávání a zobrazování paketů. Používá knihovny winpcap nebo libpcap

Počet výskytů	Použité Cipher suit
435	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
159	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
127	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
86	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
79	TLS_RSA_WITH_AES_128_CBC_SHA
4	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

Tab. 4.10: Tabulka četnosti zachycených cipher suit

v závislosti na použité platformě. Využívá nastavení, disektory a filtry z programu Wireshark.

Původní návrh softwarového nástroje zahrnoval použití programu tshark. Program nDPI prováděl detekci SSL/TLS spojení a data o spojení předával programu tshark, který prováděl získávání některých metadat o spojení. Při prvních testech na malé množině dat vývojová verze vykazovala dobré výsledky. Pro provedení testu s větším množstvím dat se ukázalo toto řešení jako nevhodné. Program tshark byl spouštěn jako nový proces nezávisle na detektoru nDPI. Pokaždé, když dostal nová data, spustil prohledávaní původního souboru od začátku. Přestože data zpracovávaného souboru pcap byla uložena v operační paměti, doba vyhledání konkrétních paketů zabrala velmi dlouhý čas a zároveň vytěžovala na 100% jedno jádro procesoru. Pokud tedy zpracovávaný soubor obsahoval větší množství SSL/TLS spojení, doba zpracování neúměrně narůstala do řádu minut.

Další nevýhodou byla absence vhodných disektorů pro STARTTLS a simplexní spojení.

Původní návrh obsahující program tshark vykazoval nedostatky a velké množství špatně vyhodnocených paketů, které by musely být dále zpracovávány.

Z těchto důvodů nebyl program tshark do konečného softwarového nástroje použit.

4.4 Analýza získaných metadat

V průběhu testování softwarového nástroje bylo uloženo několik souborů csv se záznamy metadat. Na těchto datech byla provedena níže uvedená analýza. Celkem bylo zaznamenáno 890 zabezpečených SSL/TLS spojení, všechna spojení jsou duplexní. Ve všech případech byl získán název serveru z certifikátu u 302 spojení zpráva Client_Hello neobsahovala server_name.

Všechna zachycená spojení měla použity dostatečné šifrovací algoritmy, viz tab.

Počet výskytů	Verze SSL/TLS
798	TLS 1.2
95	TLS 1.0

Tab. 4.11: Tabulka četnosti zachycených verzí SSL/TLS

4.10 a nikde nebylo zaznamenáno použití již nedoporučovaných. Použití verze SSL/TLS 1.0 a TLS 1.2 je také považováno za bezpečné, viz tab. 4.11.

4.5 Testování

Testování softwarového nástroje pro detekci a extrakci metadat z SSL/TLS spojení probíhalo na vytvořených souborech pcap a živých datech. Pro testování byl použit přenosný počítač HP 15-ax002nc s procesorem Intel i7-6700HQ, 16GB operační paměti a operačním systémem Windows 10.

Testování bylo zaměřeno na:

- Propustnost.
- Hardwarová zátěž.
- Úspěšnost rozpoznání SSL/TLS spojení.

Použitá testovací data

Pro testování propustnosti a hardwarové zátěže byla vytvořena testovací data ze zachyceného síťového provozu pomocí programu Wireshark. Vzorky byly uloženy do souboru ve formátu pcap. Všechny vzorky byly pořízeny na vlastním duplexním provozu.

Pro testování úspěšnosti byly vytvořeny soubory viz tab. 4.12 s jednotlivým TCP spojením obsahujícím SSL/TLS navázání spojení. Jednotlivá spojení byla vyfiltrována pomocí programu Wireshark a uložena. Z takto uložených duplexních vzorků byla následně vytvořena data simplexní rozdělením na serverové a klientské strany a uložena. Byly vytvořeny vzorky samotného TLS spojení a vzorky obsahující protokoly poštovních služeb SMTP, POP3, IMAP při použití STARTTLS včetně následně navázaného SSL/TLS spojení. Data poštovních služeb byla zachycena z vlastního poštovního účtu za použití klienta Thunderbird na poštovní server seznam.cz.

Propustnost a hardwarová zátěž

Měření propustnosti a hardwarové zátěže bylo prováděno souběžně. K měření propustnosti byla vyhodnocena měření, která poskytuje program nDPI.

Název	**Velikost**	**Počet paketů**	**Původní verze**	**Upravena verze**
duplex_tls.pcap	6495 kB	16	SSL	SSL
server_tls.pcap	4548 kB	8	SSL	SSL
client_tls.pcap	1971 kB	8	SSL	SSL
duplex_smtp_starttls.pcap	8697 kB	40	Mail_SMTP	Mail_SMTPS
server_smtp_starttls.pcap	5987 kB	20	Mail_SMTP	Mail_SMTPS
client_smtp_starttls.pcap	2734 kB	20	Mail_SMTP	Mail_SMTPS
duplex_pop_starttls.pcap	8550 kB	43	Mail_POP	Mail_POPS
server_pop_starttls.pcap	5860 kB	21	Mail_POP	Mail_POPS
client_pop_starttls.pcap	2714 kB	22	Unknown	Mail_POPS
duplex_imap_starttls.pcap	8753 kB	39	Mail_IMAP	Mail_IMAPS
server_imap_starttls.pcap	6343 kB	16	Mail_IMAP	Mail_IMAPS
client_imap_starttls.pcap	2434 kB	23	Unknown	Mail_IMAPS

Tab. 4.12: Tabulka cvičných vzorků a porovnání výsledků detekce

Program pcapReader je vybaven měřením propustnosti paketů a zobrazuje ve dvou jednotkách pps [10] a b/s [11]. Pokud je program spuštěn na živých datech, tak měření propustnosti neprovádí.

Další měření prováděná programem nDPI jsou již zobrazována v obou režimech, tedy počet IP paketů zpracovaných a jejich celkový počet, celkový počet bajtů a počet unikátních spojení.

```
pcap file contains
        IP packets: 389460 of 390277 packet total
        IP bytes: 210733503
        Unique flows: 27936
        nDPI throughout: 935.60 K pps / 3,77 Gb/s
```

K měření hardwarové zátěže byly použity systémové nástroje MS Windows a to Správce úloh a Sledování prostředků. Byly sledovány parametry vytížení systému, procesoru a operační paměti.

Měření propustnosti bylo provedeno i na výchozí verzi, aby mohli být porovnány hodnoty po provedení úprav.

[10] Packet per second - jednotka propustnosti síťového zařízení, neboli zpracované pakety za sekundu.

[11] bit za sekundu - jednotka přenosové datové rychlosti.

Vyhodnocení Hardwarová zátěž

Provedeny byly celkem tři testy. Jeden na zachytávání živého provozu po dobu dvou hodin. Další tři testy byly provedeny na dvou pcap souborech uvedených v tabulce. Soubory byly o různé velikosti a reprezentovaly běžný osobní datový provoz na internetu.

Při živém zachytávání dat byl procesor vytížen programem maximálně na 0,5% a operační paměť byla vytížena po 2 hodinách provozu na 5MB. Během živého zachytávání bylo zachyceno 2439743 paketů, 2439569 z toho bylo zpracováno, celkem přeneseno 2,8GB. Bylo zaznamenáno 691 unikátních datových spojení. Zachyceno bylo celkem 245 SSL/TLS spojení. Celý záznam měření viz obr. 4.8.

Výpočet využití paměti v případě zpracovávání hustšího síťového provozu. Použijeme-li hodnotu obsazení paměti pro 691 unikátních spojení, což je přibližně 5MB, pak jedno spojení spotřebuje asi 7kB paměti. Za jednu hodinu internetového provozu 50 uživatelů vznikne 67 tis. unikátních spojení(hodnota unikátních spojení získána z výpisu proxy serveru malé firmy). Pro takovýto provoz by bylo potřeba asi 47MB paměti za hodinu zpracovaného síťového provozu. Pro dlouhodobější nasazení by bylo vhodné program odladit tak, aby po určitém časovém intervalu, nebo dle obsazení paměti provedl její vyprázdnění například opětovným spuštěním programu.

Využití jednoho procesoru vzrostlo při zpracování dat ze souboru na 20% po dobu než, byl soubor zpracován. Doba zpracování ze souboru je úměrná jeho velikosti. S výslednou propustností viz tab. 4.13.

Při zpracování živých dat je využití procesoru závislé na rychlosti přísunu paketů a unikátních spojení. V případě, že by byl jeden procesor využit na 100% a nestíhal by včas zpracovávat data, začala by klesat propustnost. Pro zpracování ze souborů bylo dosaženo velmi dobrých výsledků propustnosti, viz Vyhodnocení měření propustnosti. Rychlost zpracování je tedy závislá na taktu a výkonu procesoru. V případě potřeby nasadit detektor na vysokorychlostní síť by však musel být zdrojový kód upraven tak, aby uměl vytěžovat více než jeden procesor.

Vyhodnocení měření propustnosti

Měření probíhalo obdobným způsobem jako měření hardwarové zátěže s tím rozdílem, že byly sledovány jiné parametry.

Měření propustnosti při zpracování souborů v upravené / neupravené verze viz tab. 4.13. Při měření neklesla rychlost propustnosti pod 708 kpps a 4,64 Gb/s a celé měření je uvedeno v souhrnné tabulce. Hodnota propustnosti při větším množství zabezpečených spojení oproti původní verzi klesla na polovinu.

Na základě měření propustnosti lze konstatovat, že detektor SSL/TLS včetně zpracování metadat ze souborů je rychlostně dostačující. Pro zpracování na živých

Velikost	Unikátních spojení	SSL/TLS spojení	Propustnost pps	Propustnost b/s
673MB	799	72	1,10 G /1,04 M	10, 19 G /9,58 G
262MB	2222	532	1,41 M / 708 k	9,25 G / 4,64 G

Tab. 4.13: Tabulka cvičných vzorků a porovnání výsledků detekce

datech hraje velmi důležitou roli množství unikátních datových spojení. Pro potřeby nasazení na živých zpracováních by musel být testován na reálných spojích.

Úspěšnost rozpoznání SSL/TLS spojení

Testování úspěšnosti detekce bylo prováděno na vybrané množině vytvořených cvičných dat. Cvičná data sloužila již při vývoji a provádění úprav zdrojových kódů programu.

Testování detekce poštovních služeb používajících STARTTLS probíhalo jen na omezené množině dat.

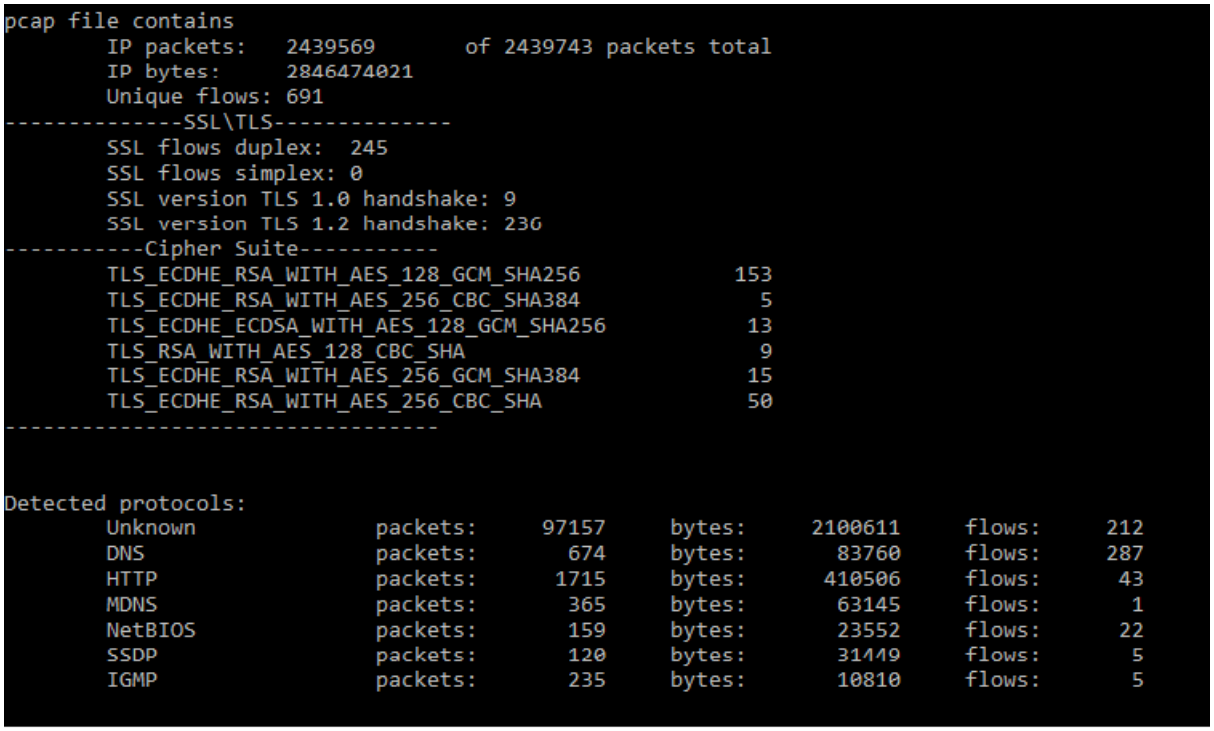
Při testování bylo sledováno rozpoznání simplex, duplex a správnost určení použitého protokolu u poštovních služeb. Dále byla ověřována správnost získaných metadat pomocí programu Wireshark a hexaeditoru a bylo provedeno porovnání výstupních výsledků s neupravenou verzí programu viz tab. 4.12.

V tabulce nejsou uvedena všechna výstupní data jako je počet zpracovaných paketů, přenesených bajtů a počet unikátních spojení. Tyto hodnoty byly v obou případech shodné.

Vyhodnocení úspěšnosti rozpoznání SSL/TLS spojení

Testování rozpoznání samotného SSL/TLS spojení probíhá s velkou úspěšností v použitých množinách. Pro přesnější výsledky by musela být provedena dlouhodobější testování na reálných datových provozech a výsledky porovnávány s výsledky jiného detektoru. Funkce detektoru bude vždy nutné občasně kontrolovat vlastní analýzou zachycených dat. Také provádět dodatečné zapracování nových šifrovacích algoritmů a změny v předpisech RFC.

Výsledky testování poštovních služeb jsou ovlivněny nedostatečným množstvím testovacích dat. Testovací data byla pouze cvičná, vývojová. Pro podání přesnějších výsledků detekce STARTTLS u poštovních služeb by bylo nutné provést více testování na reálných datech a nové poznatky následně zapracovat. Detekce je naprogramována dle doporučení konkrétních RFC a poznatků získaných vlastní analýzou. Samotná implementace protokolů poštovních služeb je často ovlivněna zvyklostmi



Obr. 4.8: Výstup z pcapreader.

programátorů, kteří do zpráv zahrnují vlastní úpravy. Pro konkrétní nasazení musí být předem provedena analýza konkrétního datového spojení a případně rozšířen zdrojový kód detekce poštovních služeb.

5 ZÁVĚR

Cílem diplomové práce bylo popsat a nastudovat problematiku protokolů SSL/TLS a princip detekce zabezpečených spojení pomocí hluboké inspekce paketů dostupnými nástroji. Na základě získaných poznatků navrhnout a realizovat implementaci do vybrané knihovny funkci nDPI tak, aby výsledný softwarový nástroj byl schopný detekovat simplexní, duplexní zabezpečené spojení a z detekovaných spojení extrahovat vybraná metadata.

Praktický přínos diplomové práce je implementace poznatků z provedené analýzy do programu pcapReader a úprava detektorů zabezpečeného spojení a poštovních služeb. Program nově dovede detekovat celé navázání zabezpečeného spojení SSL/TLS a získat z něj vybraná metadata pro duplexní i simplexní spojení. Dále je nově schopen rozpoznat použití zabezpečeného spojení při použití STARTTLS u poštovních služeb. Ze všech detekovaných spojení používajících SSL/TLS protokoly, provádí vyhodnocení a extrakci metadat do csv souboru. Exportovaná data jsou dále použitelná k analýze a filtraci spojení.

V průběhu vývoje byl nástroj testován na přesnost, propustnost a hardwarovou zátěž a porovnán s neupravenou verzí. Úpravy se projevily na přesnosti a to tím, že detekce byla rozšířena o zabezpečené poštovní spojení, nově označené jako SMTPS, POPS a IMAPS.

Snížila se hodnota propustnosti, která je závislá na množství zpracovávaných spojení. Rozdíl v propustnosti je dán množstvím vyskytujících se SSL/TLS spojení.

Před výsledným vyhodnocením detekce se vyžaduje poslat do detektoru větší množství paketů a tím vzrůstá čas průchodu paketů zabezpečeného spojení oproti původní verzi.

Výsledky úspěšnosti rozpoznání a extrakce simplexních a duplexních zabezpečených spojení splnily předpokládaná očekávání, navíc na základě získaných poznatků byly rozšířeny o rozpoznání poštovních služeb používajících STARTTLS.

Byla otestována implementace programu Tshark a na základě výsledků bližšího zkoumání nebyl oproti původnímu návrhu program Tshark použit.

Byla provedena analýza kvality zabezpečení na získaných metadatech.

Závěrem lze konstatovat, že nově vzniklý nástroj může dobře fungovat jako pomocný nástroj při analýze dat a penetračním testování zabezpečení datových spojení. Pro zvyšování přesnosti vyhodnocení by bylo nutné nástroj testovat na větší množině reálných dat a dle konkrétních zjištění případně upravit konkrétní detektor. Přeci jen RFC jsou pouze doporučení a konkrétní implementace mohou doznat vlastních úprav. Dle mého názoru má knihovna funkce nDPI velmi velký potenciál pro další vývoj a rozšiřování schopností.

LITERATURA

[1] TURNER, S.; POLK, T. *Prohibiting Secure Sockets Layer (SSL) Version 2.0* [online]. 2011, poslední aktualizace 5. 2011 [cit. 7. 12. 2016]. Dostupné z URL: <https://tools.ietf.org/html/rfc6176>.

[2] DIERKS, T.; RESCORLA, E. *The Transport Layer Security (TLS) Protocol Version 1.2* [online]. 2008, poslední aktualizace 8. 2008 [cit. 7. 12. 2016]. Dostupné z URL: <https://tools.ietf.org/html/rfc5246>.

[3] HOFFMAN, P. *SMTP Service Extension for Secure SMTP over Transport Layer Security* [online]. 2002, poslední aktualizace 5. 2002 [cit. 7. 12. 2016]. Dostupné z URL: <https://www.ietf.org/rfc/rfc3207>.

[4] NEWMAN, C. *Using TLS with IMAP, POP3 and ACAP* [online]. 1999, poslední aktualizace 6. 1999 [cit. 28. 04. 2017]. Dostupné z URL: <https://www.ietf.org/rfc/rfc2595>.

[5] HODGES, J.; JACKSON, C.; BARTH, A. *HTTP Strict Transport Security (HSTS)* [online]. 2012, poslední aktualizace 11. 2012 [cit. 7. 12. 2016]. Dostupné z URL: <https://tools.ietf.org/html/rfc6797>.

[6] FORD-HUTCHINSON, P. *Securing FTP with TLS* [online]. 2005, poslední aktualizace 10. 2005 [cit. 7. 12. 2016]. Dostupné z URL: <https://tools.ietf.org/html/rfc4217>.

[7] FREIER, A.; KARLTON, P. *The Secure Sockets Layer (SSL) Protocol Version 3.0* [online]. 2011, poslední aktualizace 8. 2011 [cit. 7. 12. 2016]. Dostupné z URL: <https://tools.ietf.org/html/rfc6101>.

[8] DIERKS, T.; ALLEN, C. *The TLS Protocol version 1.0* [online]. 1999, poslední aktualizace 2. 1999 [cit. 7. 12. 2016]. Dostupné z URL: <https://www.ietf.org/rfc/rfc2246.txt>.

[9] DIERKS, T.; RESCORLA, E. *The Transport Layer Security (TLS) Protocol version 1.1* [online]. 2006, poslední aktualizace 2. 2006 [cit. 7. 12. 2016]. Dostupné z URL: <https://www.ietf.org/rfc/rfc4346.txt>.

[10] XU, C.; CHEN, S.; SU, J.; YIU, S. M.; HUI, L. C. K. *A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms* [online]. 2014, poslední aktualizace 2016 [cit. 1. 12. 2016]. Dostupné z URL: <http://ieeexplore.ieee.org/document/7468531/>. ISSN 1553-877X.

[11] DERI, L.; MARTINELLI, M.; BUJLOW, T.; CARDIGLIANO, A. *nDPI: Open-source high-speed deep packet inspection* [online]. 2002, poslední aktualizace 5. 2014 [cit. 12. 4. 2017]. Dostupné z URL: <http://ieeexplore.ieee.org/document/6906427/>. ISSN 2376-6492.

[12] ALCOCK, S.; NELSON, R. *Measuring the accuracy of open-source payload-based traffic classifiers using popular Internet applications* [online]. 2013, poslední aktualizace 2013 [cit. 20. 4. 2017]. Dostupné z URL: <http://ieeexplore.ieee.org/document/6758538/>. ISBN 978-1-4799-0540-9.

[13] *Open and Extensible LGPLv3 Deep Packet Inspection Library* [online]. Poslední aktualizace 2016 [cit. 7. 12. 2016]. Dostupné z URL: <http://www.ntop.org/products/deep-packet-inspection/ndpi/>.

[14] BURDA, Karel. *Bezpečnost informačních systémů* [elektronicky]. Vyd. 1. V Brně: Vysoké učení technické v Brně Fakulta elektrotechniky a komunikačních technologií, 152 s poslední aktualizace 2013 [cit. 5. 3. 2017]. ISBN 978-80-214-4890-2

[15] COOPER, D.; SANTESSON, S.; FARREL S.; BOEYEN, S.; HOUSLEY, R.; POLK, W. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile* [online]. 2008, poslední aktualizace 5. 2008 [cit. 12. 4. 2017]. Dostupné z URL: <https://tools.ietf.org/html/rfc5280>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

IoT	Internet of Things
SSL	Secure Socket Layer
TLS	Transport Layer Security
PCT	Private Comunication Technology
DPI	Deep Packet Inspection
MAC	Message Authentication Code
TCP	Transmission Control Protocol
HTTP	Hypertext Transfer Protocol
SMTP	Simple Mail Transfer Protocol
FTP	File Transfer Protocol
RC4	Bloková šifra – ARCFOUR
AES	Advanced Encryption Standard
RSA	Asymetrický šifra – Rivest Shamir Adleman
DSA	Digital Signature Algorithm
PKI	Public Key Infrastructure
ASCII	American Standard Code for Information Interchange
IANA	Internet Assigned Numbers Authority
HMAC	Keyed-hash Message Authentication Code
IMAP	Internet Message Access Protocol
Samba	Server Message Block
DNS	Domain Name System
SIP	Session Initiation Protocol
MITM	Man-in-the-middle
HSTS	HTTP Strict Transport Security

ISP	Internet Service Provider
IP	Internet Protocol
PRISM	Planning Tool for Resource Integration, Synchronization, and Management
USA	United States of America
CA	Certificate authority
QoS	Quality of Service
NIDS	Network intrusion detection system
ESMTP	Extended Simple Mail Transfer Protocol

SEZNAM PŘÍLOH

A Návod k použití programu **60**

B Obsah přiloženého CD **62**

A NÁVOD K POUŽITÍ PROGRAMU

Verze programu je kompilovaná 32 bitově pro operační systém Windows.

Použití programu pro práci se soubory

Počítač pro analýzu dat ze souboru nemusí být vybaven síťovou kartou. Musí mít, ale nainstalovaný program Winpcap[1]. Pokud je již na počítači nainstalován program Wireshark včetně Winpcap není třeba již pro danou pracovní stanici nic dodatečně instalovat.

Nejvhodnější způsob práce s programem je pomocí dávkového souboru. Do adresáře, kde se nachází program pcapReader.exe vytvoříme dávkový soubor a do něj zadáme cestu k programu a parametry. Pokud chceme uložit i výstup zobrazený po ukončení, lze pomocí parametru » výstup přesměrovat do definovaného souboru. Příklad dávkového souboru pcapReader.cmd.

```
pcapReader.exe -d -i C:\Users\Hutar\Data\ssl_1.pcap >>
   vystup.log
```

Použití programu při záchytu ze síťové karty

Při použití programu ze síťové karty je potřeba pracovní stanici se síťovým rozhraním. Dále je potřeba nainstalovaný program Winpcap, Windump[2] nebo program Wireshark. Pomocí programu Windump nebo programu Wireshark zjistíme adresu síťového rozhraní, které chceme použít k záchytu dat.

Zjištění názvu síťového rozhraní pomocí Windump.exe včetně výstupu.

```
windump.exe -D
1.\Device\NPF_{DA0608F8-A0DC-4A9E-860B-731ACB85E4AB} (
   Microsoft)
2.\Device\NPF_{52011853-ED78-4B81-AF1C-E931C07793C7} (
   Realtek PCIe GBE Family Controller)
```

Zjištění názvu síťového rozhraní pomocí Wireshark.

```
Capture-->Options-->Manage Interfaces-->Interface Name
```

Zjištěný název rozhraní použijeme místo cesty k souboru a postup je obdobný jako u výše popsaného návodu práce se soubory.

[1]Dostupné z URL: <https://www.winpcap.org/>

[2]Dostupné z URL: <https://www.winpcap.org/windump/install/default.htm>

Příklad dávkového souboru livepcapReader.cmd.

```
pcapReader.exe -d -s 100 -i \Device\NPF_{52011853-ED78-4
   B81-AF1C-E931C07793C7}
```

Pro živé zachytávaní lze použít parametr -s a za tento parametr uvést čas v sekundách po který chceme zachytávat. Po uplynutí doby se program sám ukončí. Pokud nezadáme parametr s hodnotou času záchytu, bude probíhat do doby, než bude program ručně ukončen ručně pomocí kláves Ctrl + c. Následně bude zobrazen výstup programu.

B OBSAH PŘILOŽENÉHO CD

Na přiloženém CD

- /data_pcap/
  – testovací pcap soubory.
- /nDPI_zdroj_kody/
  – veškeré zdrojové kódy potřebné pro kompilaci programu pcapReader.
- /pcapReader/
  – program pcapReader.exe.
- /pdf/
  – elektronická verze diplomové práce.

Pro provádění úprav na knihovně funkcí bylo použito vývojové prostředí programu Visual Studio 2015.

Návod k používání programu pcapReader.exe je uveden v příloze A.