VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

IDENTIFIKACE MOBILNÍCH APLIKACÍ POMOCÍ OTISKŮ TLS

IDENTIFICATION OF MOBILE APPLICATIONS USING TLS FINGERPRINTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **Bc. FILIP KOČICA**
AUTHOR

VEDOUCÍ PRÁCE **Ing. PETR MATOUŠEK, Ph.D., M.A.**
SUPERVISOR

BRNO 2021

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Wireshark, pcap, csv, TLS, otisk, aplikace.

Keywords

Wireshark, pcap, csv, TLS, fingerprint, application.

Citace

KOČICA, Filip. *Identifikace mobilních aplikací pomocí otisků TLS*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr Matoušek, Ph.D., M.A.

Identifikace mobilních aplikací pomocí otisků TLS

Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně.

. . . . . .
Filip Kočica
4. března 2021

Poděkování

—

Obsah

Kapitola 1

Úvod

Tato práce řeší problematiku identifikace mobilních aplikací za pomoci tvorby TLS otisků z komunikace dané aplikace. Pro řešení byly využity moderní způsoby zahrnující výpočet JA3(S) z hlaviček TLS posílaných klientem a serverem při ustavování spojení, které jsou přenášeny v otevřené podobě (nešifrované). V rámci práce bylo také experimentováno s kombinací těchto vlastností a certifikátem posílaným serverem klientovi. Všechny metody byly kvantitativně vyhodnoceny a popsány.

1.1 Identifikace aplikací

Při studiu jsem vycházel z Matoušek et al. [6] a Anderson et al. [5]. V dřívějších letech se pro identifikaci zařízení a aplikací používaly způsoby založené na datech z hlaviček různých nešifrovaných protokolů, jako např. HTTP či IMAP. V posledních letech se značně zvýšila míra šifrované komunikace pomocí SSL/TLS tunelu nad TCP a zmíněné metody identifikace se staly nepoužitelné.

Výzkumníci tedy začali pro identifikaci používat data, které jsou přenášena ještě nezašifrovaná, a sice ustanovení relace TLS, tzv. *TLS handshake*. Specificky to jsou zprávy Client Hello, Server Hello a přenášené certifikáty. Certifikát serveru lze získat ze zprávy `Certificate`, zatímco veřejný klíč klienta je získán ze zprávy `Client Key Exchange`.

Na začátku se provede klasický tří-fázový handshake pro ustanovení TCP spojení. Poté klient žádá o vytvoření tunelu pomocí zprávy Client Hello. Protože existují různé implementace TLS, klient posílá i informaci o verze TLS, dále pak šifrovací sady, či podporovaná rozšíření (m.j.). Server poté m.j. odpovídá vybranou šifrovací sadou a certifikátem. Tyto zprávy tedy obsahují jak data vhodná pro vytváření otisku (verze, sada, ...), tak nevhodná (náhodné bajty). Ze vhodných dat jsou spočteny otisky aplikace. Takto vytvořené otisky jsou poté uloženy do databáze, a je možné do ní přidávat nové otisky (i pro danou aplikaci, např. při nové verzi) či v ní hledat otisky při identifikaci.

Jak bylo zjištěno z práce Matoušek et al. [6], samotný JA3 hash spočtený z Client Hello zprávy není dostatečně jedinečný pro správnou a stabilní identifikaci vzhledem k tomu, že se vyskytují duplicitní JA3 hashe společné pro více různých aplikací. Dle dané práce je do porovnání třeba zanést další vlastnosti, jako jsou JA3S hash (spočtený ze Server Hello) a SNI[1] aplikace, což přináší značně lepší výsledky identifikace. Tato práce si klade za cíl provést obdobné porovnání na vlastní datové sadě, doplněné o další vlastnost a sice certifikát serveru.

[1] Server Name Indication – Indiktátor jména serveru.

Identifikaci aplikací však značně ztěžují různé faktory, jakou jsou například náhodná čísla v hlavičkách TLS handshake, padding (česky výplň), reklamní a sledovací servery, které nejsou přímo spojeny s danou aplikací. Tyto negativně ovlivňující faktory je vhodné odstranit ještě před samotným vytvářením otisků aplikace.

1.2 Vyhodnocení a metriky

Způsob vyhodnocení je takový, že se testovací otisk prochází databáze otisků a výsledek každého porovnání je klasifikován do jedné ze čtyř tříd, a sice [1, 3]:

- ***True positive*** – Otisky jsou predikovány jako **stejné** a ve skutečnosti jsou **stejné**.
- ***False positive*** – Otisky jsou predikovány jako **stejné**, ale ve skutečnosti jsou **různé**.
- ***True negative*** – Otisky jsou predikovány jako **různé** a ve skutečnosti jsou **různé**.
- ***False negative*** – Otisky jsou predikovány jako **různé**, ale ve skutečnosti jsou **stejné**.

Z těchto hodnot je pak možné získat metriky pro určování kvality různých systémů a algoritmů. Tyto metriky jsou následující [1, 3]:

- ***Accuracy*** – Udává, jak dobře si model vede ve všech třídách. Vypočítá se jako poměr mezi počtem správných předpovědí a celkovým počtem předpovědí.
- ***Precision*** – Udává přesnost modelu při klasifikaci vzorku jako pozitivního. Počítá se jako poměr mezi počtem správně klasifikovaných vzorků k celkovému počtu vzorků klasifikovaných jako pozitivní.
- ***Recall*** – Udává poměr mezi počtem pozitivních vzorků správně klasifikovaných jako pozitivní k celkovému počtu pozitivních vzorků.

Tyto spočtené hodnoty poté reprezentují jak dobře dokáže model vzorky klasifikovat do správné třídy.

1.3 Struktura textu

Kapitola 1 krátce shrnuje problematiku řešenou v rámci této práce a metriky použité pro vyhodnocení řešení. V další kapitole 2 je nastíněn způsob, jakým se daná problematika v rámci této práce má řešit, k tomu využité aplikace a technologii. Kapitola 3 popisuje implementační detaily poskytnutého řešení. Poté jsou v kapitole 4 prezentovány dosažené výsledky a nakonec v závěru je zhodnocen přínos této práce a možnosti dalšího pokračování.

Kapitola 2

Návrh řešení

Tato kapitola se zabývá návrhem řešení a tedy jak budou data získána, uložena a zpracována. Dále také jaké technologie k tomu budou využity.

2.1 Získání dat

Pro vytvoření mobilní komunikace Android aplikací bude využit nástroj AVD[1]. V nástroji AVD bude vytvořeno virtuální zařízení Pixel 4 běžící na Android 8.0 s verzí API 26 a obsahující předinstalovanou aplikaci Google Play, pomocí které bude možné do virtuálního zařízení jednoduše instalovat aplikace bez nutnosti využití ADB CLI[2].

Dle zadání bude předinstalováno 10 aplikací, a sice následujících: Pinterest v9.5.0, Dáme jídlo v21.02.0, Bolt Food v1.0.0, Signal v5.3.12, ROSSMAN v1.9.1, Zonky v3.4.0, Twitch v10.1.0, Zalando v5.1.2, Discord v61.4 a Reddit 2021.5.0. Popis datových sad lze najít v tabulce 2.1.

Tabulka 2.1: Datové sady pro 10 výše zmíněných testovacích aplikací.

	Trénovací		Testovací	
Aplikace	Počet běhů	TLS spojení	Počet běhů	TLS spojení
Pinterest	6	113	1	73
Dáme jídlo	6	246	1	89
Bolt food	6	87	1	41
Signal	6	58	1	40
ROSSMAN	6	66	1	42
Zonky	6	155	1	56
Twitch	6	91	1	22
Zalando	6	106	1	43
Discord	6	61	1	29
Reddit	6	140	1	33
Celkem	—	1123	—	468
Unikátní	—	223	—	159

[1] Android Virtual Device – Virtuální zařízení android.
[2] Android Debug Bridge Command Line Interface – Rozhraní příkazové řádky pro ladění Android.

2.2 Technologie a formáty

Jako „databáze“ TLS otisků bude pro jednoduchost použita textová forma a sice záznamy uložené ve formátu CSV[3]. Tento formát se pro uchování zpracovávaných dat hodí nejvíce. V případě většího objemu dat (např. tisíce či miliony TLS otisků) by už bylo vhodnější využít sofistikovanějšího způsobu poskytujícího např. indexování pro rychlejší vyhledávání, apod.

Pro implementaci této práce byl vybrán skriptovací jazyk `Bash`, protože disponuje nespočtem funkcí pro zpracování a manipulaci dat (např. `sed`, `awk`, apod.) s možností přímého použití různých linuxových utilit jako jsou např. `tshark` či `md5sum`.

Komunikace mobilní aplikace (běžící na emulátoru AVD) s reálným serverem bude poté odchycena pomocí nástroje Wireshark[4]. Při odchytávání dat nebudou aplikované žádné filtry, data budou předzpracována později pomocí skriptů v Bash.

[3] Comma-Separated Value – Hodnoty oddělené čárkami.
[4] Nástroj pro on-line zachycení paketů ze síťového zařízení – https://www.wireshark.org.

Kapitola 3

Implementace řešení

V této kapitole je popsán způsob extrakce důležitých dat ze zachycených komunikací, způsob výpočtu otisků TLS, experimenty a metodika vyhodnocení.

3.1 Předzpracování dat

Pro každou z deseti aplikací byly zachyceny dva vzorky komunikace. Jeden tzv. trénovací a druhý pro testování. Tyto vzorky jsou pro každou aplikaci uloženy separátně:

```
data/datasets/test/*.pcapng
data/datasets/train/*.pcapng
```

Prvním krokem je předzpracování souborů se zachycenými komunikacemi pomocí skriptu `src/parseTlsDataset.sh`, který přebírá tři parametry. Prvním je složka, ve které má skript hledat soubory se zachycenou komunikací ve formátu `pcapng`[1], druhým je složka kam má uložit odpovídající soubor ve formátu CSV a posledním je oddělovač. Tzn. pro každou aplikaci je vytvořen jeden odpovídající soubor CSV s extrahovanými daty. Filtrování se provádí na základě jednoduchého pravidla a seznamu hodnot z hlaviček, které se mají uložit oddělené pomocí středníků:

```
tshark -r $filename -T fields -E separator=";" \
    -e ip.src \
    -e ip.dst \
    -e tcp.srcport \
    -e tcp.dstport \
    -e ssl.handshake.type \
    -e ssl.handshake.version \
    -e ssl.handshake.ciphersuite \
    -e ssl.handshake.extension.type \
    -e ssl.handshake.extensions_server_name \
    -e ssl.handshake.extensions_supported_group \
    -e ssl.handshake.extensions_ec_point_format \
    -e ssl.handshake.certificate \
    -R "ssl.handshake.type==1 or ssl.handshake.type==2 or ssl.handshake.certificate" \
    -2
```

[1] Packet CAPture New Generation – Zachycení paketů nové generace.

Pomocí tohoto příkazu tedy lze vyfiltrovat pouze komunikaci TLS *handshake* (ustanovení relace) ze souboru `$filename`. Každý záznam je doplněn o směrovací informace z hlavičky protokolu TCP, aby šlo mimo jiné dohledat odpovědi serveru.

3.2 Výpočet otisků

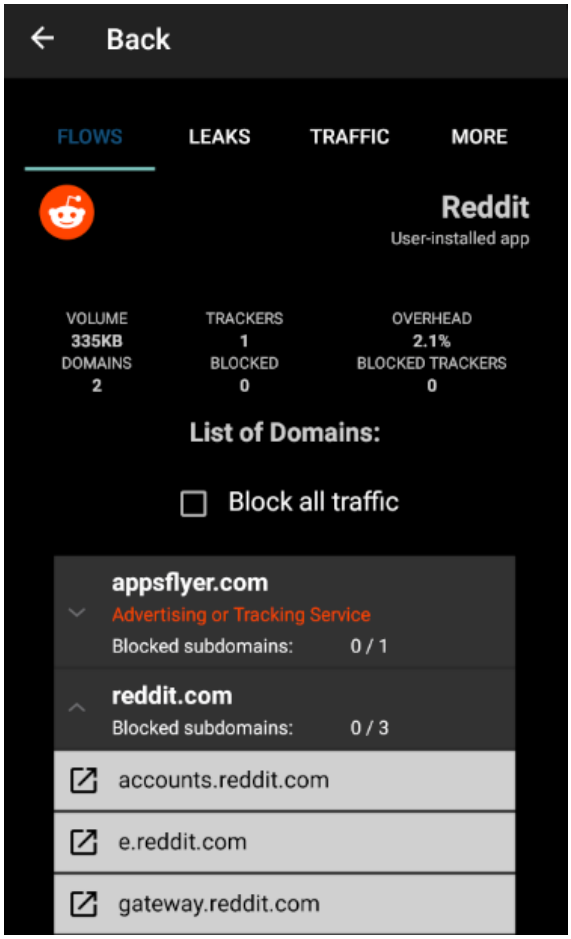
Z předchozího kroku předzpracování vznikne sada souborů CSV s extrahovanou TLS komunikací ustanovení relace. Dále přichází na řadu skript `./src/extractTlsFingerprints.sh`, který slouží ke komplexnějšímu filtrování dat a výpočtu otisků pro každou aplikaci.

Tento skript přebírá tři parametry. Prvním z nich je cesta k souborům CSV. Dalším z nich je cesta k tzv. *whitelist* souborům. Takový soubor existuje pro každou aplikaci (má stejný název) a obsahuje seznam klíčových slov, které se mohou vyskytovat v SNI dané aplikace. Posledním parametrem je pro změnu cesta k souboru *blacklist*. Toto je soubor obsahující seznam klíčových slov, které se zpravidla vyskytují v SNI různých reklamních či analytických serverů, které nemají nic společného s danou aplikací a měli by být odfiltrovány.

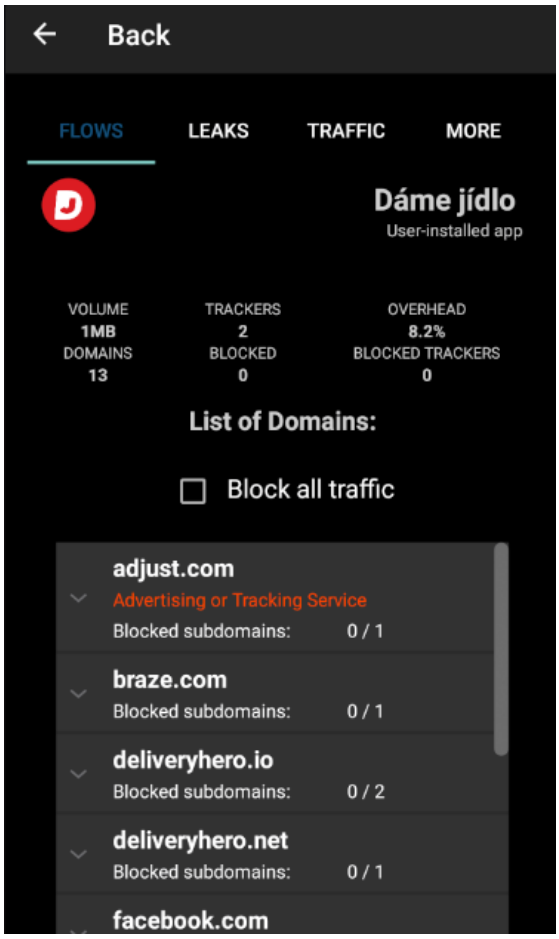
Proces získání *whitelist* klíčových slov byl převzat z práce Matoušek et al. [6], a sice stažení aplikace Lumen[2], zapnutí skenování a používání všech deseti analyzovaných aplikací. V aplikaci Lumen potom lze nalézt jaké monitorovaná aplikace používá domény, subdomény, protokoly, a tak dále. Na obrázcích 3.1 a 3.2 lze vidět příklady dvou aplikací. V aplikaci lze poté nalézt sledovací a reklamní servery zvýrazněné oranžově (které byly umístěny do souboru *blacklist*) a také jaké domény a subdomény používá daná aplikace (umístěny do separátních souborů *whitelist*). Např. pro Reddit stačí použít klíčové slovo „reddit“, avšak pro Dáme jídlo je třeba hledat „deliveryhero“.

Algoritmus poté sestává z procházení skrze jednotlivé CSV soubory, a následně procházení skrze TLS záznamy daného souboru. Pokud je záznam Client Hello, je provedena kontrola zda SNI záznamu obsahuje některé z klíčových slov ve *whitelist* a také neobsahuje nic z *blacklist*. Pakliže záznam tímto filtrem projde, je pro něj dohledána odpověď serveru (Server Hello, certifikát) a spočtou se JA3 a JA3S otisky, které jsou společně s SNI, certifikátem serveru a jménem aplikace spojeny do nového záznamu a uloženy do databáze pro danou aplikaci.

[2]Lumen – https://apkpure.com/lumen-privacy-monitor/edu.berkeley.icsi.haystack



Obrázek 3.1: Investigace datového toku aplikace Reddit pomocí aplikace Lumen.



Obrázek 3.2: Investigace datového toku aplikace Dáme jídlo pomocí aplikace Lumen.

Spočtení otisků aplikace poté sestává z:

1. Odfiltrování hodnot GREASE z extrahovaných položek: šifrovací sady, rozšíření a podporovaných skupin [6, 2].
2. Odfiltrování rozšíření 65281 (tzn. *renegotiation* – nového ustanovení) [6, 4].
3. Odfiltrování rozšíření 21 (tzn. *padding* – výplně) [6, 4].
4. Převodu verze a podporovaných skupin z hexadecimální soustavy do decimální.
5. Sestavení JA3 z Client Hello položek: verze, šifrovací sady, rozšíření a podporovaných skupin. Následně spočtení 32-bitového kontrolního součtu pomocí utility `md5sum`.
6. Sestavení JA3S z Server Hello položek: verze, vybrané šifrovací sady a rozšíření. Následné spočtení 32-bitového kontrolního součtu pomocí utility `md5sum`.
7. Na základě směrovacích informací z hlaviček TCP jsou odpovídající Client a Server Hello (tzn. JA3 a JA3S) spojeny do jednoho záznamu, společně se SNI, hashem certifikátu serveru a aplikací, ze které byly získány.

3.2.1 Experimenty

Cílem bylo v rámci experimentování využít data o certifikátu a využít je jako další vlastnost při identifikaci otisků. Proto bylo do příkazu `tshark` uvedeného výše přidáno filtrování také na záznamy přenosu certifikátu a byly pro testování vyextrahovány všechny položky vztahující se k certifikátům, jak byly uvedeny na oficiálních stránkách dokumentace aplikace Wireshark[3] (`ssl.handshake.cert*`). Jediný sloupeček výsledného CSV, který nebyl prázdný, byl samotný certifikát, a tedy byl jako jediný použit jako další vlastnost. Při experimentech bylo zjištěno, že více serverů dané aplikace (např. `i.pinimg.com` a `api.pinterest.com`) používá stejný certifikát i přes různé SNI, zatímco stejný certifikát zcela určitě nebudou používat jiné aplikace. To se zdálo jako vhodná vlastnost pro otisk aplikace.

Při připojování JA3S hashe (odpověď serveru) k odpovídajícímu JA3 hashi (Client Hello) byl také na základě směrovacích informací připojen certifikát serveru. Certifikát byl však příliš dlouhý, a tedy z něj byl také spočten MD5 hash. Položky databáze tedy nakonec obsahovaly: JA3, JA3S, certifikát, SNI a jméno aplikace. Vyhodnocení tohoto experimentu lze najít v sekci 4.3.

3.3 Způsob vyhodnocení

Pro vyhodnocení slouží třetí a poslední skript, zvaný `./src/evalTlsFingerprints.sh`. Tento skript má dva argumenty a sice databázi otisků a seznam testovacích otisků k vyhodnocení. Vyhodnocení poté probíhá tak, že je každý testovaný otisk porovnán vůči celé databázi uložených otisků. Na základě predikcí se vytváří matice záměn a z té se poté získají hodnoty TP, FP, TN a FN a z těch se následně spočte úspěšnost.

Celý proces nakonec sestává ze: získání datových sad pro trénování a testování, jejich vložení do odpovídajících složek, vytvoření odpovídajících *whitelist* souborů vyplněných klíčovými slovy dané aplikace, vytvoření souboru *blacklist* a provedení těchto operací:

```
find data/ -name '*.csv' -delete # Smazání předchozích CSV souborů

./src/parseTlsDataset.sh data/datasets/train/ data/datasets/train/ _
./src/parseTlsDataset.sh data/datasets/test/  data/datasets/test/  .

./src/extractTlsFingerprints.sh ./data/datasets/train/ ./data/whitelists/ \
          ./data/sni_blacklist.txt -u > data/database.csv
./src/extractTlsFingerprints.sh ./data/datasets/test/ ./data/whitelists/  \
          ./data/sni_blacklist.txt -u > data/testFingerprints.csv

awk -i inplace '!seen[$0]++' data/database.csv
awk -i inplace '!seen[$0]++' data/testFingerprints.csv  # Unikátnost záznamů

./src/evalTlsFingerprints.sh ./data/testFingerprints.csv ./data/database.csv
```

To provede všechny výše popsané kroky a z matice záměn jsou poté spočteny statistiky úspěšnosti, jak lze vidět např. na snímku 4.1.

[3] https://www.wireshark.org/docs/dfref/s/ssl.html

Kapitola 4

Vyhodnocení řešení

Vyhodnocení proběhlo na datové sadě vytvořené v rámci této práce, čítající dohromady 223 TLS ustanovení komunikace (Client Hello + Server Hello + Server Certificate). Vyhodnocení dle vzoru práce Matoušek et al. [6] zahrnovalo několikanásobné vyhodnocení s použitím různých vlastností TLS a doplňuje vyhodnocení o klasifikaci i na základě certifikátu serveru. Celkové statistiky i s maticemi záměn lze vidět na snímcích 4.1 až 4.4, zachycujících přímo výstup aplikace, nebo v tabulce 4.1, která přehledněji prezentuje spočtené úspěšnosti bez matic záměn.

Tabulka 4.1: Dosažené výsledky při použití různých vlastností TLS.

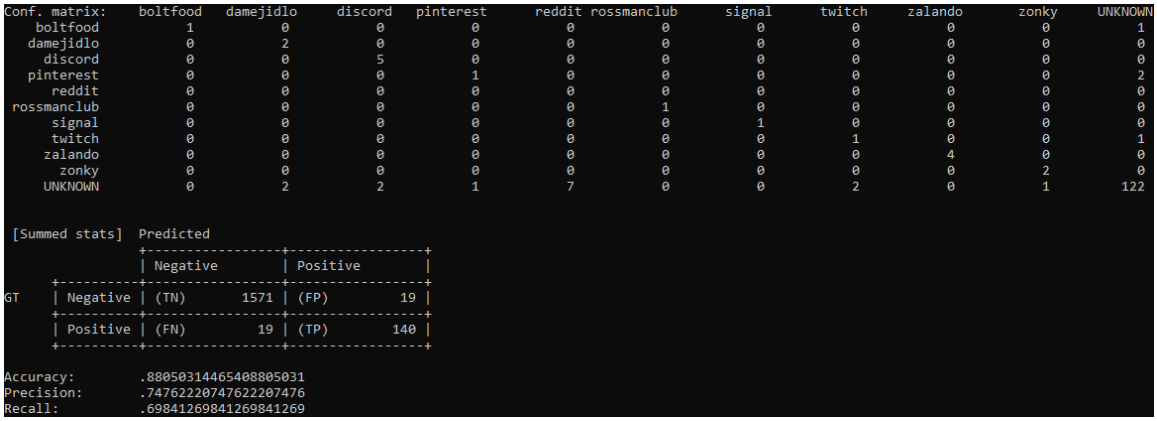
TLS vlastnosti	Celkem	Accuracy	Precision	Recall
JA3	159	0.80	0.16	0.18
JA3+JA3S	159	0.88	0.75	0.70
JA3+JA3S+Cert	159	0.94	**0.99**	0.80
JA3+JA3S+SNI	159	**0.97**	**0.99**	**0.93**

```
Conf. matrix:    boltfood   damejidlo     discord   pinterest      reddit rossmanclub      signal      twitch     zalando       zonky     UNKNOWN
    boltfood           0           0           0           0           0           0           0           0           0           0           0
   damejidlo           0           0           0           0           0           0           0           0           0           0           0
     discord           0           0           0           0           0           0           0           0           0           0           0
   pinterest           0           0           0           0           0           0           0           0           0           0           0
      reddit           0           0           0           0           0           0           0           0           0           0           0
 rossmanclub           0           0           0           0           0           0           0           0           0           0           0
      signal           0           0           0           0           0           0           1           0           0           0           0
      twitch           0           0           0           0           0           0           0           0           0           0           0
     zalando           0           0           0           0           0           0           0           0           0           0           0
       zonky           0           0           0           0           0           0           0           0           0           0           0
     UNKNOWN           1           4           7           2           7           1           0           3           4           3         126


 [Summed stats]  Predicted
                 +----------------+----------------+
                 | Negative       | Positive       |
      +----------+----------------+----------------+
GT    | Negative | (TN)       1558 | (FP)         32 |
      +----------+----------------+----------------+
      | Positive | (FN)         32 | (TP)        127 |
      +----------+----------------+----------------+

Accuracy:        .79874213836477987421
Precision:       .16340621403912543153
Recall:          .18181818181818181818
```

Obrázek 4.1: Snímek obrazovky s výstupem skriptu při použití pouze vlastnosti JA3, obsahující matici záměn a spočtenou úspěšnost. Na vodorovné ose jsou predikce a na svislé pravdivé štítky (ang. *ground-truth*). Dále je vidět tabulka sumarizovaných hodnot TP, FP, TN a FN pro všechny aplikace a spočtená úspěšnost pomocí tří metrik.



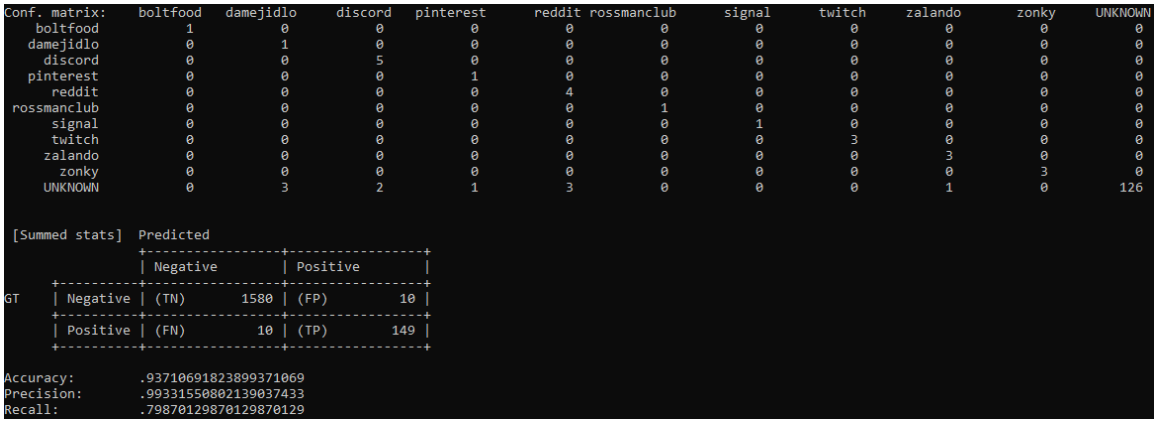
Conf. matrix:	boltfood	damejidlo	discord	pinterest	reddit	rossmanclub	signal	twitch	zalando	zonky	UNKNOWN
boltfood	1	0	0	0	0	0	0	0	0	0	1
damejidlo	0	2	0	0	0	0	0	0	0	0	0
discord	0	0	5	0	0	0	0	0	0	0	0
pinterest	0	0	0	1	0	0	0	0	0	0	2
reddit	0	0	0	0	0	0	0	0	0	0	0
rossmanclub	0	0	0	0	0	1	0	0	0	0	0
signal	0	0	0	0	0	0	1	0	0	0	0
twitch	0	0	0	0	0	0	0	1	0	0	1
zalando	0	0	0	0	0	0	0	0	4	0	0
zonky	0	0	0	0	0	0	0	0	0	2	0
UNKNOWN	0	2	2	1	7	0	0	2	0	1	122

[Summed stats] Predicted

GT		Negative	Positive
	Negative	(TN) 1571	(FP) 19
	Positive	(FN) 19	(TP) 140

Accuracy: .88050314465408805031
Precision: .74762207476222207476
Recall: .69841269841269841269

Obrázek 4.2: Výsledky kombinace vlastností JA3+JA3S.



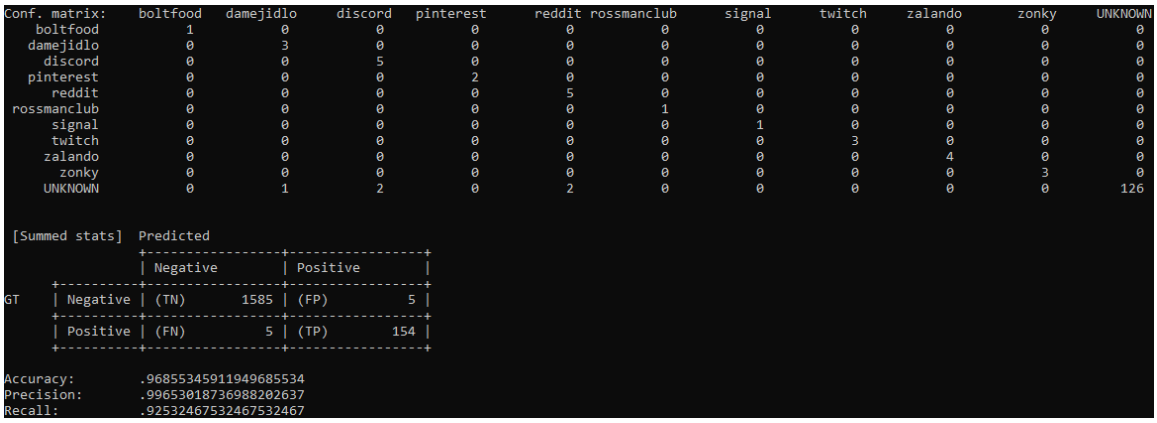
Conf. matrix:	boltfood	damejidlo	discord	pinterest	reddit	rossmanclub	signal	twitch	zalando	zonky	UNKNOWN
boltfood	1	0	0	0	0	0	0	0	0	0	0
damejidlo	0	1	0	0	0	0	0	0	0	0	0
discord	0	0	5	0	0	0	0	0	0	0	0
pinterest	0	0	0	1	0	0	0	0	0	0	0
reddit	0	0	0	0	4	0	0	0	0	0	0
rossmanclub	0	0	0	0	0	1	0	0	0	0	0
signal	0	0	0	0	0	0	1	0	0	0	0
twitch	0	0	0	0	0	0	0	3	0	0	0
zalando	0	0	0	0	0	0	0	0	3	0	0
zonky	0	0	0	0	0	0	0	0	0	3	0
UNKNOWN	0	3	2	1	3	0	0	0	1	0	126

[Summed stats] Predicted

GT		Negative	Positive
	Negative	(TN) 1580	(FP) 10
	Positive	(FN) 10	(TP) 149

Accuracy: .93710691823899371069
Precision: .99331550802139037433
Recall: .79870129870129870129

Obrázek 4.3: Výsledky kombinace vlastností JA3+JA3S+Certifikát.



Conf. matrix:	boltfood	damejidlo	discord	pinterest	reddit	rossmanclub	signal	twitch	zalando	zonky	UNKNOWN
boltfood	1	0	0	0	0	0	0	0	0	0	0
damejidlo	0	3	0	0	0	0	0	0	0	0	0
discord	0	0	5	0	0	0	0	0	0	0	0
pinterest	0	0	0	2	0	0	0	0	0	0	0
reddit	0	0	0	0	5	0	0	0	0	0	0
rossmanclub	0	0	0	0	0	1	0	0	0	0	0
signal	0	0	0	0	0	0	1	0	0	0	0
twitch	0	0	0	0	0	0	0	3	0	0	0
zalando	0	0	0	0	0	0	0	0	4	0	0
zonky	0	0	0	0	0	0	0	0	0	3	0
UNKNOWN	0	1	2	0	2	0	0	0	0	0	126

[Summed stats] Predicted

GT		Negative	Positive
	Negative	(TN) 1585	(FP) 5
	Positive	(FN) 5	(TP) 154

Accuracy: .96855345911949685534
Precision: .99653018736988202637
Recall: .92532467532467532467

Obrázek 4.4: Výsledky kombinace vlastností JA3+JA3S+SNI.

4.1 Použití JA3

Při zkoumání, kolik aplikací má stejný JA3 hash, bylo zjištěno, že až 8 z 10 aplikací mělo jeden z JA3 hashů stejný – např. `fada0859379fec2c87b490b8203dc520` pro Discord, Pin-

terest, atd. To při vyhodnocení vytvářelo velké množství falešně pozitivních detekcí, jak lze vidět na snímku 4.1. Pouze aplikace Signal správně klasifikovala jediný unikátní získaný vzorek. Ostatní se často zaměňovaly s neznámým provozem.

4.2 Použití JA3+JA3S

Poté, co bylo porovnání doplněno o další vlastnost (JA3S), se úspěšnost značně zlepšila. Zde bylo zjištěno, že duplicity se nachází maximálně u 2 z 10 aplikací. Tímto se značně snížil počet falešně pozitivních detekcí, viz 4.2.

4.3 Použití JA3+JA3S+Cert

Výsledky použití této kombinace vlastností se nachází někde na pomezí JA3+JA3S a JA3+JA3S+SNI. Tato kombinace je dle mého názoru velmi podobná JA3+JA3S+SNI. Avšak občas se stalo, že aplikace měla stejný hash certifikátu jako reklamní/analytický server, což způsobovalo falešné detekce a bylo dosaženo mírně horších výsledků (např. hash `d41d8cd98f00b204e9800998ecf8427e` aplikace Dáme jídlo a analytického serveru google `ssl.google-analytics.com`). Viz 4.3.

4.4 Použití JA3+JA3S+SNI

Zde byl do porovnání zahrnut i identifikátor serveru, což zajistilo, že téměř každá pozitivní predikce byla správná. Viz 4.4.

Kapitola 5

Závěr

V rámci práce bylo implementována identifikace mobilních aplikací pomocí TLS otisků ve skriptovacím jazyce `Bash` a pomocí kombinace vlastností TLS handshake zpráv dosaženo relativně dobrých výsledků identifikace aplikací. Řešení bylo vyhodnoceno na 159 testovacích a 223 trénovacích ustanoveních získaných z 10 populárních aplikací.

Vytvořené skripty jsou poměrně pomalé a vyhodnocení vytvořené sady trvá i několik jednotek či desítek minut. To je způsobeno zejména spouštěním tzv. *sub-shellů* v cyklech. `Bash` byla celkově velká chyba protože nebylo možné nalézt žádnou knihovnu či použitelný kus kódu a vše bylo třeba implementovat ručně. Příště bych si pro implementaci vybral kompilovaný programovací jazyk blízký síťovému zpracování, jako např. `C++`, popřípadě skriptovací jazyk `Python`.

Vzhledem k velikosti datové sady `pcapng` cca. 300MB byla tato sada umístěna na google disk a do README umístěn odkaz. Dále je tam popsáno kam je třeba datovou sadu umístit pro možné otestování řešení. Také se omlouvám za strany s poděkováním apod., nepodařilo se mi je odstranit ze šablony.

V rámci budoucí práce by bylo zajímavé vyhodnocení na základě komplexnějších klasifikátorů než pouhé porovnání extrahovaných vlastností. Zajímavé by bylo reprezentovat extrahované vlastnosti jako jakýsi n-ární vektor a tedy mít „prostor otisků“ a např. pomocí jednoduchého učení vymezovat hranice mezi těmito třídami v prostoru.

Literatura

[1] *Accuracy, Precision, Recall or F1?* [https://towardsdatascience.com/331fb37c5cb9]. Naposledy navštíveno: 2021-02-24.

[2] *Applying GREASE to TLS Extensibility* [https://tools.ietf.org/html/draft-davidben-tls-grease-01]. Naposledy navštíveno: 2021-02-23.

[3] *Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall* [https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/]. Naposledy navštíveno: 2021-02-24.

[4] *Transport Layer Security (TLS) Extensions* [https://www.iana.org/assignments/tls-extensiontype-values/tls-extensiontype-values.xhtml]. Naposledy navštíveno: 2021-02-23.

[5] Anderson, B., Paul, S. a McGrew, D. A. Deciphering Malware's use of TLS (without Decryption). *CoRR*. 2016, abs/1607.01639.

[6] Matoušek, P., Burgetová, I., Ryšavý, O. a Victor, M. On Reliability of JA3 Hashes for Fingerprinting Mobile Applications. In: *Digital Forensics and Cyber Crime. ICDF2C 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer International Publishing, 2021, sv. 351, s. 1–22. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. DOI: 10.1007/978-3-030-68734-2_1. ISBN 978-3-030-68733-5.