VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ MOBILNÍCH APLIKACÍ V SÍŤOVÉM PROVOZU NA ZÁKLADĚ OTISKŮ JA3

DETECTION OF MOBILE APPS IN NETWORK TRAFFIC USING JA3 FINGERPRINTING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JÁN VAVRO**
AUTHOR

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

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Vavro Ján**
Program: Informační technologie
Název: **Monitorování mobilních aplikací v síťovém provozu na základě otisků JA3**
Detection of Mobile Apps in Network Traffic Using JA3 Fingerprinting
Kategorie: Počítačové sítě

Zadání:

1. Prostudujte modifikovanou metodu identifikace mobilních aplikací pomocí otisků JA3 a JA3S dle [1].
2. Navrhněte způsob automatizovaného vytváření otisků mobilních aplikací např. pomocí nástroje Android Virtual Studio. Návrh implementujte a porovnejte výstup s otisky z reálného zařízení.
3. Pro vybranou množinu mobilních aplikací (min. 10) vytvořte databázi otisků.
4. Vytvořte datasety reálné síťové komunikace, která obsahuje mobilní provoz. Popište obsah datasetu a vyznačte, jaké mobilní aplikace obsahuje.
5. Otestujte identifikaci mobilních aplikací v daném provozu na základě databáze otisků. Vyhodnoťte úspěšnost detekce.
6. Zhodnoťte celkový výsledek vaší práce a využitelnost pro monitorování sítí.

Literatura:

1. MATOUŠEK Petr, BURGETOVÁ Ivana, RYŠAVÝ Ondřej a VICTOR Malombe. On Reliability of JA3 Hashes for Fingerprinting Mobile Applications. In Proceedings of ICDF2C 2020, s. 20.
2. van Ede, T., Bortolameotti, R., Continella, A., Ren, J., Dubois, D.J., Lindorfer, M., Cho ness, D., van Steen, M., Peter, A.: FlowPrint: Semi-Supervised Mobile-App Fingerprinting on Encrypted Network Trac. In: NDSS. The Internet Society, 2020.
3. Anderson, B., McGrew, D.: TLS Beyond the Browser: Combining End Host and Network Data to Understand Application Behavior. In: Proceedings of the Internet Measurement Conference. pp. 379-392, 2019.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Matoušek Petr, Ing., Ph.D., M.A.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 2. listopadu 2020

Abstrakt

V posledných rokoch sa stala mobilná komunikácia oveľa bezpečnejšia. Dôvodom je zapúzdrenie internetových dát protokolom TLS, ktorý šifruje prenášané dáta. Bezpečnosť používateľa sa zvýšila, ale na druhej strane táto skutočnosť limituje možnosti monitorovania, keďže obsah komunikácie sa stáva pre monitorovacie systémy neznámy. Táto bakalárska práca skúma možnosti monitorovania mobilných aplikácií v sieťovej komunikácii na základe otlačkov JA3 a JA3S. Práca si dáva za cieľ implementovať nástroje na automatizované vytvorenie databázy otlačkov a následnú detekciu.

Abstract

In recent years, mobile network communication became more secure. The reason is encapsulation with TLS protocol, that encrypts transmitted data. User security was increased, but on the other hand it limits network monitoring possibilites, because the data are encrypted. This thesis reseraches possibilities of monitoring mobile applications in network traffic using JA3 and JA3S fingerprints. The aim is to implement tools for automated creation of fingerprints databse and consecutive detection.

Kľúčové slová

TLS značkovanie, JA3 otlačok, mobilné aplikácie, šifrovaná komunikácia, monitorovanie

Keywords

TLS fingerprinting, JA3 hash, Mobile Application, Encrypted Communication, Monitoring

Citácia

VAVRO, Ján. *Monitorování mobilních aplikací v síťovém provozu na základě otisků JA3.* Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr Matoušek, Ph.D., M.A.

Monitorování mobilních aplikací v síťovém provozu na základě otisků JA3

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Petra Matouška, Ph.D., M.A. Uviedol som všetky literárne zdroje a publikácie, z ktorých som čerpal.

.....
Ján Vavro
10. mája 2021

Poďakovanie

Chcel by som poďakovať pánovi Ing. Petrovi Matouškovi, Ph.D., M.A., ktorý bol vždy ochotný poskytnúť odbornú pomoc pri vypracovaní práce.

Obsah

Kapitola 1

Úvod

V poslednej dekáde sa používanie mobilných zariadení rapídne rozšírilo. V rozmedzí pár rokov sa mobilné zariadenia stali súčasťou každodenného života. Viac ako 44% svetovej populácie (3,5 miliardy) vlastní smartfón [6]. Prevažná väčšina aplikácií potrebuje pre správne fungovanie prístup na internet. Podiel mobilnej internetovej komunikácie je aktuálne 50,44% [2] a očakáva sa, že s nástupom 5G technológií bude ďalej rásť. Neodmysliteľnou súčasťou sieťovej komunikácie je jej bezpečnosť. Jedným z prostriedkov ako ju zvýšiť je monitorovanie siete.

Po udalostiach v uplynulých rokoch, kedy uniklo veľké množstvo citlivých dát, sa začal obsah komunikácie šifrovať. Podiel šifrovanej komunikácie v roku 2020 presiahol 99% [4]. Schopnosť analyzovať šifrovanú komunikáciu je dnes nevyhnutná pre zaistenie sieťovej bezpečnosti. Tradičné monitorovacie techniky prestali fungovať, pretože obsah komunikácie sa stal neznámym. Z tohoto dôvodu bolo nevyhnutné vymyslieť nové spôsoby monitorovania.

Jeden zo spôsobov je vytváranie otlačkov z metadát šifrovanej komunikácie. Pri jej nadväzovaní sa musí klientská a serverová strana dohodnúť na spôsobe a konkrétnych parametroch šifrovania. Tieto údaje sú posielané prostredníctvom nezabezpečeného komunikačného kanálu, a tak sú vhodné na monitorovacie účely. Kombináciou vybraných vlastností z nadväzovania spojenia je možné vytvoriť otlačky, pomocou ktorých sa dajú detegovať, napríklad mobilné aplikácie.

Táto bakalárska práca skúma možnosti ich monitorovania metódou JA3 a JA3S otlačkov. Po preskúmaní relevantných zdrojov informácií v tejto problematike, boli vytvorené dátové sady obsahujúce záznamy so šifrovanou komunikáciou z mobilných zariadení. Boli využité ako virtuálne, tak aj fyzické zariadenia. Následne bol navrhnutý a implementovaný nástroj na extrakciu JA3, respektíve JA3S otlačkov. Jeho výstupom je databáza otlačkov, ktorá slúži ako základ na detekciu mobilných aplikácií v reálnom prostredí. Cieľom tejto práce je schopnosť detegovať všetky mobilné aplikácie, ktoré databáza obsahuje. Otlačky sa však v jednotlivých verziách aplikácií menia, a tak je databázu nutné aktualizovať.

1.1 Štruktúra textu

Druhá kapitola bakalárskej práce popisuje protokol TLS. Text zachádza do detailov iba v miestach, ktoré sú pre vytváranie otlačkov dôležité. V kapitole číslo 3 sa nachádza popis aplikácie na automatickú tvorbu databázy otlačkov a analýza nástrojov použitých na jej tvorbu. Dátovými sadami a detekciou aplikácií sa zaoberá kapitola 4. O evaluácii pojednáva

kapitola 5, ktorá podrobne opisuje výsledky v rôznych testovacích scenároch. Na záver je v práci zhrnutý jej prínos a ďalšie možnosti rozšírenia detekcie.

Kapitola 2

Metóda otlačkov TLS

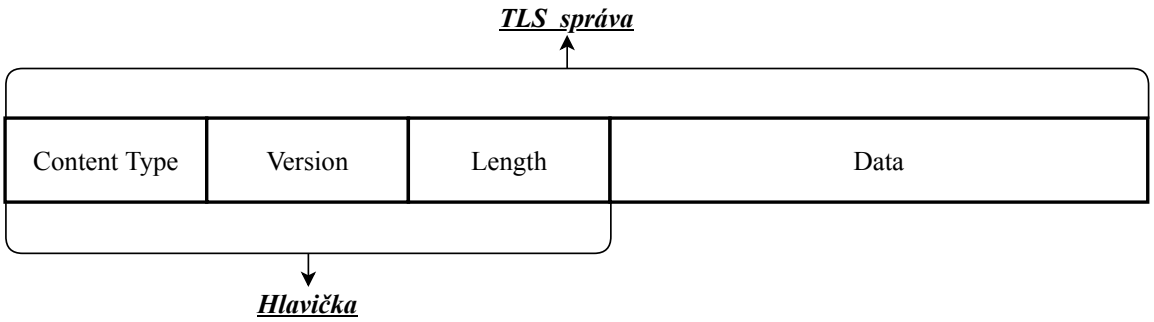
Otlačok TLS slúži na identifikáciu klienta komunikujúceho prostredníctvom zabezpečeného kanálu. Identifikácia prebieha na základe extrakcie parametrov komunikácie, vytvorenia otlačku a následného porovnania s databázou otlačkov. V tejto kapitole je popísaný protokol TLS a vytváranie otlačkov pomocou metód JA3 a JA3S.

2.1 TLS

Informácie v tejto sekcii sú prevzaté z oficiálnej dokumentácie TLS [5], ak nie je uvedené inak. Protokol Transport Layer Security (TLS) poskytuje zabezpečený prenos dát medzi dvoma komunikujúcimi zariadeniami na sieti. Patrí do aplikačnej vrstvy modelu TCP/IP ako je znázornené na obrázku 2.1. Pre fungovanie potrebuje spoľahlivý transportný protokol, napríklad TCP. TLS je nezávislý na vyššom aplikačnom protokole, ktorý ho využíva.

Obr. 2.1: Pozícia TLS v sieťovom modele.

Správy TLS sú typované, čo umožňuje použitie viacerých čiastkových protokolov pomocou tej istej štruktúry a ich vrstvenie. Formát správy je na obrázku 2.2.



Obr. 2.2: Všeobecná štruktúra TLS správ.

`Content Type`: Definuje typ čiastkového protokolu, ktorého dáta správa obsahuje. Čiastkové protokoly sú popísané nižšie 2.1.

`Version`: Špecifikuje verziu TLS. Položka je od verzie 1.3 zastaralá a bude používaná iba kvôli spätnej kompatibilite.

`Length`: Udáva dĺžku správy prenášanej v poli `Data`.

Komunikačný kanál zabezpečený pomocou TLS poskytuje nasledujúce vlastnosti:

- Autentifikácia: Serverová strana sa musí vždy autentifikovať voči klientovi, autentifikácia klienta je nepovinná.
- Dôvernosť: Dáta posielané po úspešnom ustanovení zabezpečeného spojenia sú viditeľné iba pre koncové zariadenia. TLS nešifruje dĺžku prenášanej správy, avšak koncové zariadenia majú možnosť pridať výplň za účelom zvýšenia bezpečnosti.
- Integrita: Dáta posielané po úspešnom ustanovení zabezpečeného spojenia nemôžu byť modifikované útočníkom bez toho, aby to koncové zariadenia detegovali.

Tieto predpoklady sú zachované aj v situácii, keď útočník prebral kontrolu nad celou sieťou, kadiaľ sú správy medzi smerované.

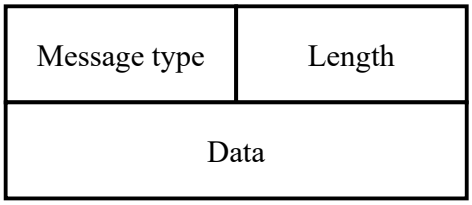
TLS pozostáva z troch základných čiastkových protokolov:

- Protokol na podanie rúk (Handshake protokol): Zabezpečuje autentifikáciu serveru (voliteľne aj klienta), dohodu na kryptografickej metóde a jej parametroch, výmenu verejných kľúčov.
- Protokol na výmenu dát: Slúži ako generická štruktúra pre ostatné čiastkové protokoly.
- Výstražný protokol: Využívaný na signalizáciu problémov.

Podrobnejší popis protokolov sa nachádza v nasledujúcich sekciách. Protokoly sú popísané vo verzii TLS 1.2, pretože väčšina záznamov internetovej komunikácie vytvorených v rámci tvorby bakalárskej práce obsahuje toky v danej verzii. Zmeny vo verzii TLS 1.3 na podstatu detekcie nemajú veľký vplyv.

2.1.1 Protokol na podanie rúk (Handshake protokol)

Handshake protokol slúži na vytvorenie zabezpečeného komunikačného kanálu a na dohodu kryptografických parametrov medzi komunikačnými stranami. Cieľom je dohodnúť sa na verzii TLS, šifrovacích algoritmoch a autentifikovať sa. Správy sú prenášané v sekcii `Data` a ich formát je zobrazený na obrázku 2.3.



Obr. 2.3: Štruktúra správy TLS Handshake.

`Message Type`: Definuje typ handshake protokolu, ktorého dáta správa obsahuje.

`Length`: Veľkosť časti `Data` v správe.

Prvotný handshake

Na začiatku každého TLS spojenia sa uskutoční prvotný handshake. Jednotlivé správy sú znázornené na obrázku 2.4. Prvé dve správy `ClientHello` a `ServerHello` sú podstatné pre detekciu, a preto bude ich obsah popísaný bližšie. Využitie dát z týchto správ je popísané v podkapitole 2.2, resp. 2.3

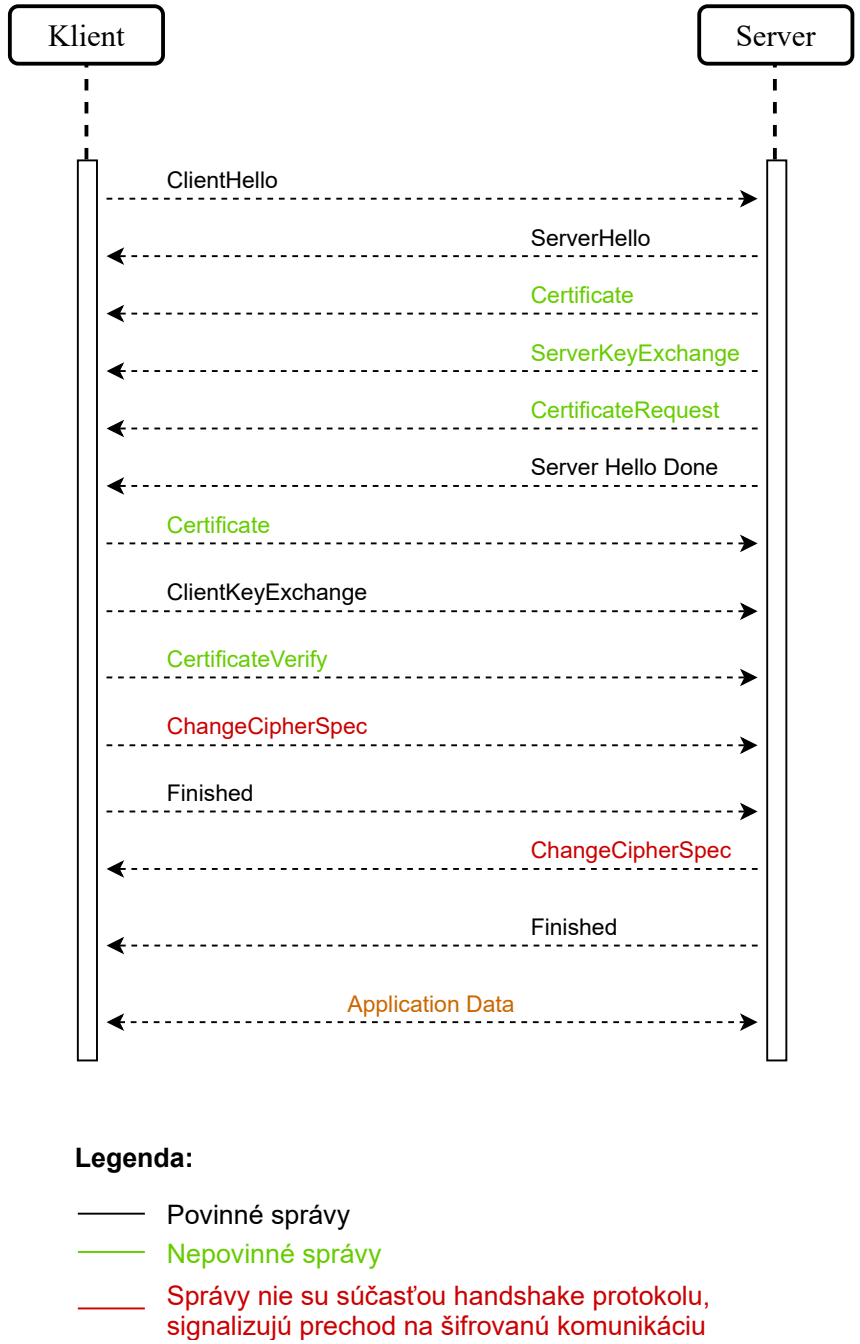
ClientHello - Správa, ktorou klient zahajuje handshake, prípadne pomocou nej dokáže klient zmeniť parametre šifrovania. Obsahuje nasledujúce informácie:

- Client version - najvyššia verzia podporovaná klientom
- Random - štruktúra 28 bajtov náhodne vygenerovaných klientom a časová značka
- Session ID - identifikátor slúžiaci na obnovenie spojenia, nepovinné
- Cipher Suites - obsahuje zoznam kryptografických algoritmov podporovaných klientom
- Compression Methods - obsahuje zoznam kompresných metód podporovaných klientom
- Extensions - slúži na požiadanie o rozšírenú funkcionalitu

ServerHello - Správa, ktorou server reaguje na ClientHello, ak podporuje algoritmy, ktoré ponúka klient. Ak dané algoritmy nepodporuje, odpovedá pomocou výstražného protokolu 2.1.3 správou `HandshakeFailure`, ktorá signalizuje zlyhanie. Nasledovné informácie sú obsiahnuté v správe:

- Server version - najvyššia verzia podporovaná oboma stranami
- Random - štruktúra 28 bajtov náhodne vygenerovaných serverom a časová značka

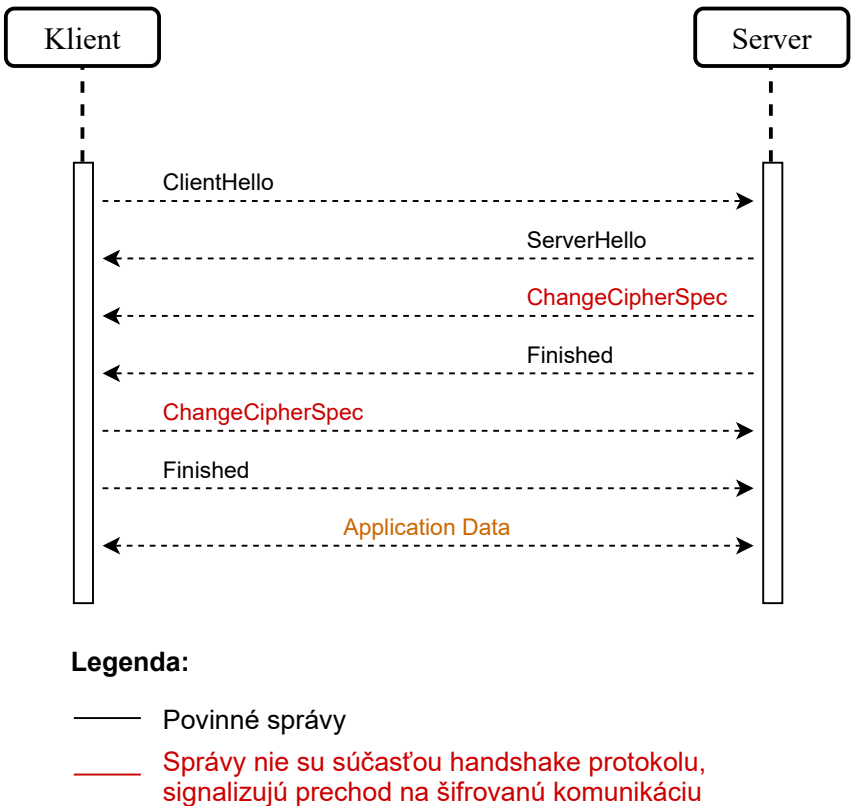
- Session ID - identifikátor vygenerovaný pre aktuálne spojenie
- Cipher Suites - obsahuje jeden kryptografický algoritmus zo zoznamu, ktorý bol poskytnutý klientom
- Compression Methods - obsahuje jednu kompresnú metódu zo zoznamu, ktorá bola poskytnutá klientom
- Extensions - slúži na rozšírenie funkcionality, v prípade ak o ňu klient požiadal



Obr. 2.4: Správy TLS Handshake.

Obnovenie sedenia

Prvotný handshake je výpočtovo náročný kvôli počtu kryptografických operácií, ktoré sa musia vykonať. Kvôli tomu TLS využíva identifikátor sedenia v správach `ClientHello` a `ServerHello` popísaných vyššie. Server si tieto identifikátory ukladá s cieľom šetrenia času a výpočtového výkonu. Ako je vidieť v komunikácii na obrázku 2.5, klient iniciuje spojenie správou `ClientHello`. Do nej vloží identifikátor sedenia, ktorý bol vygenerovaný serverom pre existujúce spojenie. Ak server nájde identifikátor, pošle klientovi správu `ServerHello` s rovnakým identifikátorom. Nasledujú správy `ChangeCipherSpec`, ktoré oznamujú prechod na šifrovanú komunikáciu a `Finished` signalizujúca úspech inicializácie spojenia. Klient odpovedá tými istými správami ako pri prvotnom spojení. Ak sa server rozhodne neobnoviť spojenie, vygeneruje a vloží do správy `ServerHello` nový identifikátor sedenia a komunikácia pokračuje ďalej ako pri prvotnom spojení.



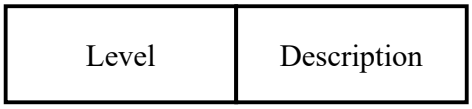
Obr. 2.5: Správy pri obnovení TLS spojenia.

2.1.2 Protokol na výmenu dát (Data protokol)

Protokol na výmenu dát je využívaný ostatnými čiastkovými protokolmi. Slúži na fragmentáciu, šifrovanie a odosielanie dát. Dáta môžu byť voliteľne pred odoslaním aj komprimované. Dátová dĺžka TLS správy je maximálne 2^{14} bajtov. Ak je správa dlhšia, tak je rozdelená do viacerých fragmentov. Naopak, správy toho istého typu s menšou veľkosťou sú vrstvené a prenášané v rámci jedného toku. Prichádzajúce dáta sú overené, dešifrované, poskladané v pôvodnom poradí a následne doručené aplikáciám vyšších úrovní.

2.1.3 Výstražný protokol (Alert protokol)

Výstražný protokol slúži na upozornenie komunikujúcich strán o chybe. Správy sú taktiež šifrované. Formát a popis správy je znázornený na obrázku 2.6



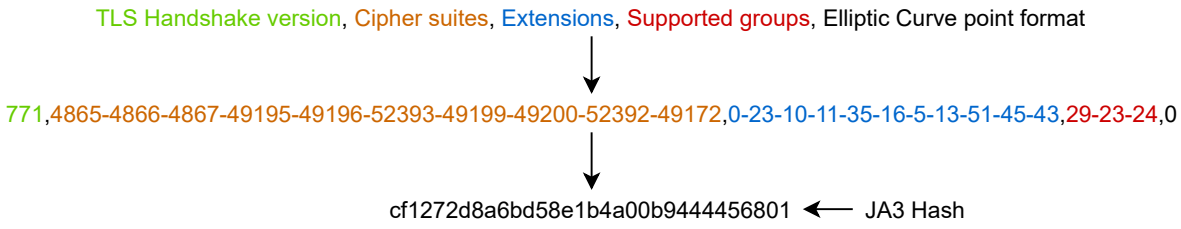
Obr. 2.6: Formát správy TLS Alert.

`Level`: Úroveň výstrahy. Obsahuje jednu z hodnôt - `warning` alebo `fatal`. Správy s úrovňou fatal majú za následok okamžité ukončenie spojenia.

`Description`: Typ výstrahy. Popis jednotlivých typov je dohľadateľný v oficiálnej dokumentácii TLS [5].

2.2 Značkovanie JA3

Značkovanie JA3[1] je metóda na generovanie otlačku, ktorý slúži na identifikáciu klienta komunikujúceho pomocou TLS. Otlačok sa vytvára hešovaním piatich atribútov správy `Client Hello` pomocou algoritmu MD5. Konkrétne sú to TLS Handshake version, Cipher suites, Extensions, Supported groups a Elliptic Curve point format. Vytváranie je znázornené na obrázku 2.7. Výsledkom je reťazec v hexadecimálnej podobe o dĺžke 32 znakov.



Obr. 2.7: Tvorba JA3.

Nevýhodou tejto metódy je fakt, že dvaja klienti môžu mať rovnaký otlačok. Napriek tomu, že klientské aplikácie sú odlišné, môžu používať tie isté knižnice na implementáciu šifrovanej komunikácie. Preto sa ako doplnok JA3 začali používať otlačky JA3S, popísané v podkapitole 2.3.

2.3 Značkovanie JA3S

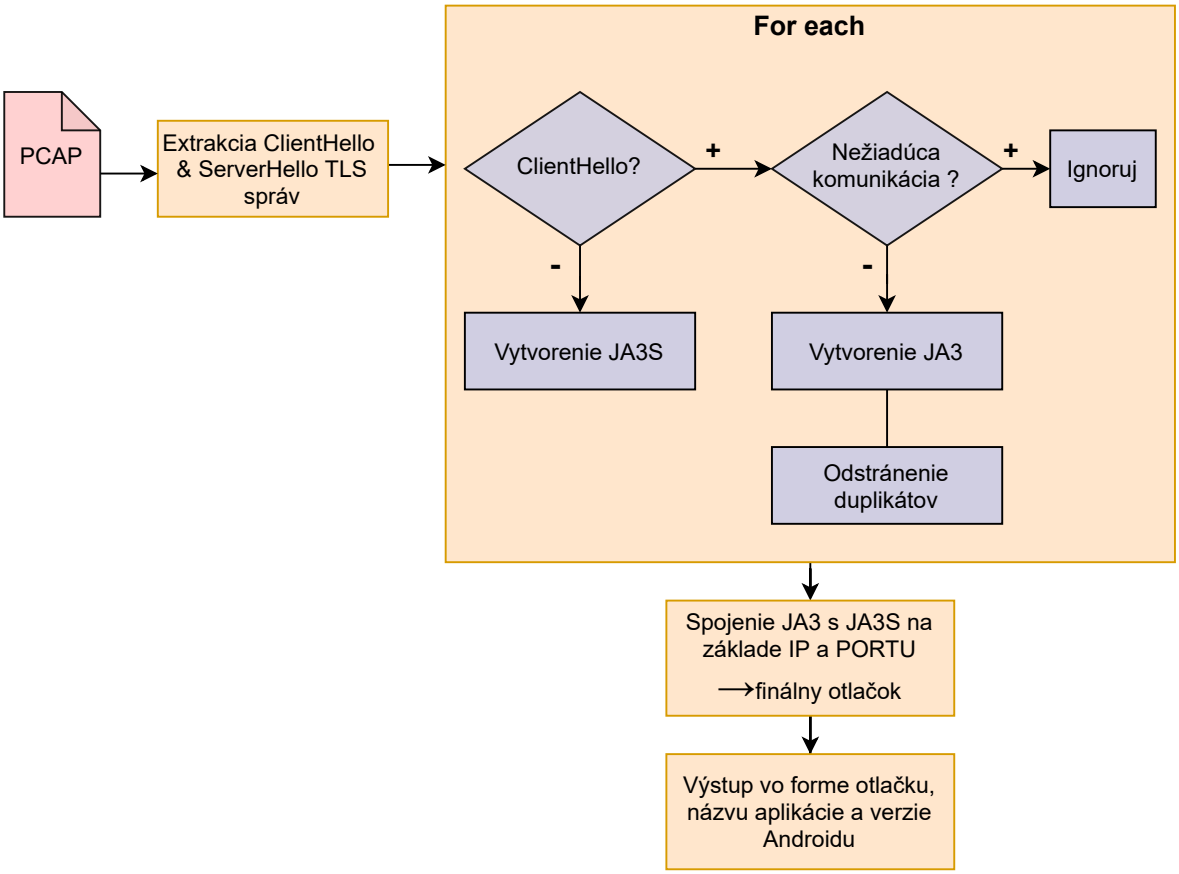
Značkovanie JA3S je metóda, ktorá slúži na identifikáciu serveru vrámci TLS komunikácie. Generovanie otlačku prebieha podobne ako pri JA3. Použité atribúty sú však len `TLS Handshake version`, `Cipher suites`, `Extensions`. Sú extrahované zo správy `ServerHello`. Otlačok sa primárne využíva na spresnenie JA3 otlačku. Párovanie sa vykonáva pomocou IP adresy a portu.

[1] https://github.com/salesforce/ja3

Kapitola 3

Vytváranie databázy otlačkov

Nasledujúca kapitola popisuje návrh aplikácie na automatizované vytváranie databázy otlačkov mobilných aplikácií. Zjednodušený vývojový diagram je na obrázku 3.1. Ako základ pre vytváranie otlačkov treba získať vstupné dáta v podobe záznamov sieťovej komunikácie. Z nej sa vyfiltrujú správy `ClientHello` a `ServerHello` popísané v sekcii 2.1.1. Následne sa z týchto správ odstráni nežiadúca komunikácia, vytvoria sa JA3, resp. JA3S otlačky, ktorých spojením vzniká finálny otlačok aplikácie. Podrobný popis jednotlivých častí sa nachádza v nasledujúcich podkapitolách.

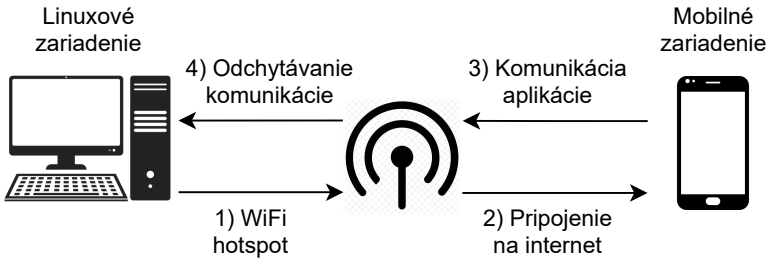


Obr. 3.1: Vývojový diagram.

3.1 Získavanie dát

Prvotnou požiadavkou na vytváranie otlačkov sú záznamy internetovej mobilnej komunikácie. Tie môžu pochádzať z emulátora[1] alebo fyzického zariadenia. Použité boli oba spôsoby s cieľom zistiť, či sa záznamy komunikácie rovnakej mobilnej aplikácie pri rovnakej verzii Androidu budú líšiť. Počas experimentov sme zistili, že medzi fyzickým zariadením a emulátorom sú rozdiely, ktoré budeme diskutovať, avšak nie sú závažné. Otlačky získané na fyzickom zariadení sa nelíšia od tých z emulátora, a tak sú použiteľné pre vytvorenie databázy otlačkov. Na vytvorenie emulátora bol použitý nástroj `AndroidVirtualDeviceManager`[2].

Zoznam aplikácií, ktorých komunikácia bola zachytávaná, je popísaný v kapitole 4. Inštalačné súbory jednotlivých aplikácií pochádzajú z webu apkmirror.com. Pre získanie dát bolo potrebné aplikáciu nainštalovať, spustiť a následne odchytiť jej komunikáciu. Na inštaláciu slúžil nástroj `AndroidDebugBridge`[3]. Odchytávanie komunikácie bolo realizované zreťazením Linuxových nástrojov `tshark`[4] a `timeout`[5]. Topológia získavania komunikácie je na obrázku 3.2. Pri vytváraní záznamov komunikácie z emulátora bolo použité rozhranie spojené s virtuálnym prostredím. V prípade fyzického zariadenia, bola vytvorená WiFi sieť, na ktorú sa zariadenie pripojilo a následne bolo jej rozhranie odpočúvané. Celkovo bolo vytvorených 120 záznamov sieťovej komunikácie. Podrobný popis sa nachádza v kapitole 4.



Obr. 3.2: Topológia odchytávania komunikácie.

3.2 Vytváranie otlačkov

Dáta vo forme `PCAP` súborov získané v prvej fáze sú vstupom pre program vytvárajúci otlačky. Z nich sa vyberú TLS handshake správy. V hlavičke TLS (obrázok 2.2) obsahujú hodnotu `ContentType:` 0x16. Ďalej sa vyfiltrujú správy `ClintHello` a `ServerHello`. Tie sa dajú identifikovať pomocou `MessageType` TLS handshake protokolu (obrázok 2.3) hodnotami 0x1, respektíve 0x2.

Následne sa odstránia správy obsahujúce nežiadúcu komunikáciu, medzi ktorú patria reklamy, dáta analytických nástrojov skúmajúce návštevnosť sledovaného obsahu a iná komunikácia aplikácií so servermi tretích strán. Podrobný popis procesu odstránenia záznamov s nežiadúcou komunikáciou je popísaný v sekcii 3.2.1. Z výslednej množiny správ sa extrahujú parametre `TLS Handshake version, Cipher suites, Extensions, Supported groups a Elliptic Curve point format` nutné pre vytvorenie JA3[S] otlačkov. Z týchto paramet-

[1] https://developer.android.com/studio/run/emulator-commandline
[2] https://developer.android.com/studio/command-line/avdmanager
[3] https://developer.android.com/studio/command-line/adb
[4] https://www.wireshark.org/docs/man-pages/tshark.html
[5] https://man7.org/linux/man-pages/man1/timeout.1.html

rov sú následne odstránené náhodné hodnoty. O ich význame pre TLS protokol a nutnosti ich odstránenia pre detekciu pojednáva sekcia 3.2.2. Extrahované hodnoty sú spojené do jedného reťazca, kde sú jednotlivé parametre oddelené čiarkami. Ak parameter obsahuje viac hodnôt, tie sú oddelené pomlčkami. Výsledný reťazec je zahešovaný funkciou MD5. Jej výstup v hexadecimálnej podobe je nazývaný JA3 otlačok. O otlačku a jeho vytvorení pojednáva podkapitola 2.2. Otlačok JA3S je vytvorený podobným spôsobom. Rozdiel je v tom, že na jeho výpočet sa používajú iba parametre `Handshake version, Cipher suites` a `Extensions`.

Oba otlačky sú nakoniec spojené pomocou IP adresy a portu, a tým vzniká finálny otlačok aplikácie. Spojenie JA3 a JA3S má za úlohu zvýšiť unikátnosť otlačkov. Kombináciou všetkých vyššie spomenutých techník na zlepšenie detekcie bola zvýšená unikátnosť otlačkov a zároveň zmenšený ich celkový počet. O experimentoch s jednotlivými technikami na zlepšenie detekcie pojednáva podkapitola 3.3.

Je nutné podotknúť, že použitý prístup predpokladá, že každá aplikácia má svoju špecifickú komunikáciu a množinou serverov. V prípade prehliadačov tento predpoklad neplatí, a tak ich detekcia použitým prístupom nie je vhodná.

3.2.1 Odstránenie nežiadúcej komunikácie

Veľké množstvo aplikácií na Android ponúka bezplatné verzie aplikácií, ktoré avšak obsahujú reklamy. Tie sú pre detekciu nežiadúce, z dôvodu, že aplikácie komunikujú s reklamnými servermi tretích strán. Reklamné servery sa dynamicky menia, čo je spôsobené reklamnými aukciami, ktoré presmerovávajú aplikáciu z reklamného serveru na konkrétneho poskytovateľa obsahu reklamy na základe výsledku aukcie [3]. Táto skutočnosť spôsobuje dva problémy. Prvým je, že všetky aplikácie obsahujúce rovnakú reklamu by mali rovnaké otlačky. Druhý problém je, že otlačky aplikácie by pribúdali s meniacou sa reklamou.

Zároveň medzi nežiadúcu komunikáciu patrí komunikácia s analytickými nástrojmi skúmajúca návštevnosť sledovaného obsahu a komunikácia aplikácie so sociálnymi sieťami za účelom identifikácie a autorizácie pomocou technológií OAuth 2.0[6] a OpenID Connect[7].

Správa `ClientHello` obsahuje rozšírenie s názvom `ServerNameIndication` (SNI), na základe ktorého dokážeme nežiadúce správy identifikovať. Toto rozšírenie so sebou nesie názov servera, s ktorým sa snaží klient spojiť. Jedno riešenie, ako sa zbaviť problémov definovaných vyššie, je vytvoriť čiernu listinu obsahujúcu zoznam serverov poskytujúcich reklamu alebo analytické nástroje. Táto metóda je časovo nestabilná, keďže s pribúdajúcimi servermi by bolo potrebné aktualizovať aj čiernu listinu. Preto bolo zvolené riešenie, ktoré definuje ku každej aplikácii množinu kľúčových slov. Aby bol pre aplikáciu otlačok vytvorený, musí názov servera, s ktorým klient komunikuje, obsahovať aspoň jedno kľúčové slovo z definovanej množiny.

Kľúčové slová boli definované na základe skúmania zachytenej internetovej mobilnej komunikácie. V tabuľke 3.1 sú znázornené kľúčové slová aj k nim prislúchajúce názvy serverov skúmaných aplikácií. Pre tento účel bol napísaný skript `SNI-reader`, ktorý dokáže hodnoty `SNI` zo správy `ClientHello` zaznamenávať pre ďalšiu analýzu. Popis skriptu a jeho spustenie je popísané v prílohe B. Zaujímavé štatistiky, ktoré boli pri analýze zozbierané, sú popísané v sekcii 3.2.1. Spustenie skriptu je popísané v prílohe B.

[6] https://oauth.net/2/
[7] https://openid.net/connect/

Štatistiky

Nad extrahovanými dátami, tvorenými multimnožinou hodnôt `SNI`, bola vykonaná analýza. Dáta boli vytvorené zo všetkých záznamov zachytenej komunikácie. Cieľom bolo získať štatistiky z hodnôt, ktoré sa v `SNI` nachádzajú.

Najčastejší názov servera v analyzovaných dátach bol `graph.facebook.com`[8]. Relatívna početnosť tejto hodnoty bola až 7.54%. To znamená, že priemerne každa trinásta `ClientHello` správa smeruje na danú adresu. Tento server slúži ako rozhranie pre získania alebo odoslania dát do sociálnej siete `facebook.com`. Veľký podiel je zapríčinený častou integráciou aplikácií so sociálnou sieťou na autentifikáciu, prípadne na poskytovanie osobných používateľských dát alebo zdieľanie obsahu.

Relatívna početnosť `SNI`, ktoré obsahovali kľúčové slová z tabuľky 3.1 a tým pádom boli použité na generovanie `JA3` otlačkov, je 35.62%. Táto hodnota potvrdzuje tvrdenia zo sekcie 3.2.1 a zároveň prikladá veľkú dôležitosť tejto optimalizácii. Prispieva k spresneniu a zníženiu počtu otlačkov. Zároveň šetrí procesorový čas pri ich vytváraní.

Pri vzájomnom porovnávaní dát z emulátorov a fyzických zariadení boli odpozorované dve skutočnosti. Prvou je, že rozdiely v `SNI` pri správach, ktoré sa následne použijú, sú veľmi malé. Z toho dôvodu možno považovať dáta z emulátora za vhodné na vytváranie databázy otlačkov. Naopak, v hodnotách `SNI` nežiadúcich správ, boli spozorované rozdiely. V emulátoroch nemalo až 19.58% hodnotu `SNI` vyplnenú. Vo fyzických zariadeniach bola táto hodnota prázdna len pri 3.77% správ. Ďalšie rozdiely boli závislé od výrobcu konkrétneho zariadenia. V zariadení od Xiaomi s Androidom vo verzii desať sa nachádzal záznam `tracking.intl.miui.com` až v 8.93% `ClientHello` správ. Na túto službu sa odosielajú údaje o aktivite užívateľa. Ďalšie rozdiely neboli signifikantné.

Aplikácia	Kľúčové slovo	SNI
aliexpress	aliexpress	acs.aliexpress.com
	alibaba	abtest.alibaba.com
	alicdn	dorangesource.alicdn.com
bolt	bolt	user.bolt.eu
discord	discord	dl.discordapp.net
		discord.com
duolingo	duolingo	excess.duolingo.com
		android-api-cf.duolingo.com
linkedin	linkedin	www.linkedin.com
netatmo	netatmo	app.netatmo.net
notecalc	notecalc	calcnote-london.s3.eu-west-2.amazonaws.com
		calcnote-*.*.*.amazonaws.com
reddit	redd.it	preview.redd.it
	reddit	accounts.reddit.com
slack	slack	slack.com
tasty	tasty	api.tasty.co

Tabuľka 3.1: Kľúčové slová priradené aplikáciám.

[8] https://developers.facebook.com/docs/graph-api/using-graph-api/

3.2.2 Odstránenie nežiadúcich hodnôt

GREASE (angl. Generate Random Extensions And Sustain Extensibility) sú náhodne vybrané hodnoty z definovanej množiny. Pridávajú sa do rozšírení TLS, s cieľom zabrániť zlyhaniam pri zväčšení množiny hodnôt používaných v rozšíreniach TLS. Počas TLS handshaku musia komunikujúce strany ignorovať náhodne generované hodnoty. Tie, ktoré tak neurobia, zlyhajú, a tým sa odhalia chyby v implementácii. Podrobnosti aj s množinou definovaných hodnôt sú uvedené v RFC 8701[1]. Nachádzať sa môžu v rozšíreniach `Cipher suites`, `Extensions` a `Supported groups`.

Tabuľky 3.2 a 3.3 demonštrujú dôležitosť odstránenia týchto hodnôt pre vytváranie otlačkov a zároveň aj pre výslednú detekciu. V prvej spomenutej tabuľke sa nachádzajú polia `Cipher suites`, `Extensions` a `Supported groups` extrahované z troch TLS správ `ClientHello`. Analyzované dáta pochádzajú z datasetu vytvoreného pre účely tejto bakalárskej práce. Konkrétne sa jedná o záznamy komunikácie z fyzického zariadenia s Androidom vo verzii 10. Všetky správy prislúchajú mobilnej aplikácii Aliexpress. Červenou sú znázornené GREASE hodnoty v daných správach. Vo štvrtom riadku sú tie isté dáta bez neželaných hodnôt. Ako je vidno, po ich odstránení je obsah polí zo všetkých troch správ totožný, a tak aj výsledný JA3 otlačok bude len jeden. Tabuľka 3.3 znázorňuje JA3 otlačky k dátam z prvej tabuľky.

V prípade neodstránenia náhodných hodnôt by bolo vytvorených príliš veľa otlačkov. Ich presnosť by zároveň bola mizivá, pretože prítomnosť náhodných hodnôt je nepovinná a ich hodnota je náhodná.

Id	Cipher Suites	Extensions	SupportedGroup
#1	64250-4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53	23130-0-23-65281-10-11-35-16-5-13-18-51-45-43-27-14906	31354-29-23-24
#2	39578-4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53	2570-0-23-65281-10-11-35-16-5-13-18-51-45-43-27-6682	10794-29-23-24
#3	64250-4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53	60138-0-23-65281-10-11-35-16-5-13-18-51-45-43-27-43690	14906-29-23-24
#4	4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157-47-53	0-23-65281-10-11-35-16-5-13-18-51-45-43-27	29-23-24

Tabuľka 3.2: Odstránenie náhodných hodnôt.

Id	JA3 otlačok
#1	13fb688942568555951bc1be870c6c53
#2	70b24f44fd26c565c700d265bd4a9451
#3	a22bf879db9072dc1d41f68bbf7ba572
#4	57bb843dc32bd4df2221037af78512a2

Tabuľka 3.3: Vplyv náhodných hodnôt na detekciu.

3.3 Experimenty

Po vytvorení databázy otlačkov bola vykonaná rada experimentov. Cieľom bolo zistiť, aký vplyv mali jednotlivé optimalizačné techniky na obsah výslednej databáze, ktorá je základom detekcie. Konkrétne bolo predmetom skúmania použitie `JA3S` otlačku a odstránenie nežiadúcej komunikácie na základe hodnôt `SNI`.

Experimenty pozostávali z vytvorenia databázy otlačkov z dát, ktoré sú popísané v kapitole 4. Najskôr bola databáza vytvorená iba z `JA3` otlačkov. Potom sa použilo párovanie s `JA3S` otlačkami, ďalej bolo skúmané chovanie `JA3` otlačkov po odstránení nežiadúcej komunikácie pomocou `SNI`. V poslednej fáze boli použité obe metódy spoločne.

Databáza bola vytvorená skriptom `Fingerprint-database-creator` implementovaným pre účely tejto bakalárskej práce. Popis skriptu a jeho spustenie sa nachádza v prílohe B. Výstup zo skriptu je `.csv` súbor. Ten bol importovaný do Postgresql[9] databázy a následne boli na importovaných dátach spúšťané SQL dotazy.

Skúmané veličiny boli celkový počet otlačkov, počet rozdielnych otlačkov, počet unikátnych otlačkov a pomer posledných dvoch veličín udávajúci percentuálnu unikátnosť otlačkov v databáze. Výsledky sú v tabuľke 3.4. Podrobnejší popis sa nachádza v texte pod ňou.

Použité metódy	Celkovo otlačkov	Rozdielne otlačky	Unikátne otlačky	Unikátnosť otlačkov
JA3	1830	181	164	91%
JA3 + JA3S	1830	281	235	84%
JA3 + SNI	581	35	31	89%
JA3 + SNI + JA3S	581	67	65	97%

Tabuľka 3.4: Vplyv JA3S otlačku a SNI na výslednú databázu.

Nasledujúce sekcie podrobne popisujú výsledky z tabuľky. Je v nich vysvetlený princíp každej z použitých metód a následne sú analyzované dosiahnuté výsledky v podobe unikátnosti otlačkov v databáze. Napriek tomu, že táto hodnota nie je vždy smerodajná, dokáže nám pomôcť ohodnotiť kvalitu výslednej databázy. V prípade, že by bola unikátnosť príliš malá, databáza by bola nepoužiteľná bez ohľadu na dáta. Ak sa naopak unikátnosť pohybuje vo vysokých číslach, je nutné zistiť, na základe akých dát bola vytvorená.

3.3.1 Analýza metódy JA3

Prvou analyzovanou metódou na vytváranie databázy otlačkov bolo použitie samotného JA3. Z celkového počtu 1830 otlačkov bolo približne 10% rozdielnych. Unikátnych bolo 91% z nich, čo predstavuje 164 otlačkov.

Pri počte desiatich analyzovaných aplikácií, každá z nich by musela mať priemerne 16 otlačkov. I keď aplikácia môže mať viacero otlačkov, toto číslo je veľmi vysoké a naznačuje nám, že dáta obsahovali veľké množstvo reklám, a tak bolo vytvorených veľa otlačkov, ktoré sú viazané na reklamu a nie konkrétnu aplikáciu. Táto metóda demonštruje, že aj keď je unikátnosť otlačkov vysoká, výsledná databáza nemusí byť kvalitná.

[9] https://www.postgresql.org/docs/

3.3.2 Analýza metódy JA3 a JA3S

Otlačok JA3S slúži na identifikáciu serveru a v tejto práci je použitý na spresnenie JA3 otlačku. Oba otlačky sa spárujú na základe ip adresy a portu. Podrobný popis JA3S otlačku sa nachádza v podkapitole 2.3.

Z tabuľky 3.4 je viditeľné, že pridaním JA3S do výsledného otlačku sa ich celkový počet celkových nezmenil, pretože JA3S len dopĺňa informáciu v JA3 otlačku. Podľa očakávaní sa počet rozdielnych otlačkov zvýšil, a to o 55% v porovnaní s použitím samostatného JA3. Podiel unikátnych otlačkov ale klesol na 84%. Táto skutočnosť je spôsobená tým, že viacero aplikácii môže mať rovnaký JA3 otlačok. Ak tento otlačok vznikol komunikáciou s reklamnými alebo inými servermi tretích strán, tak sa budú otlačky množiť a unikátnosť sa bude zmenšovať. Naopak, ak otlačok vznikol na základe komunikácie so serverom, ktorý je špecifický pre danú aplikáciu, tak pomocou JA3S môžeme vytvoriť unikátne otlačky pre každú aplikáciu. Skúmané dáta obsahovali väčšiu časť nežiadúcej komunikácie, a to malo za následok jemné zníženie unikátnosti otlačkov za cenu spresnenia niektorých, ktoré unikátne neboli.

3.3.3 Analýza metódy JA3 a SNI

`ServerNameIndication(SNI)` je rozšírenie `ClientHello` správy, obsahujúce názov servera, s ktorým klient komunikuje. Pri tvorbe databázy je táto hodnota využívaná na odfiltrovanie nežiadúcej komunikácie so servermi tretích strán. Podrobné informácie sú spísané v sekcii 3.2.1.

Na základe dát v tabuľke 3.4 je vidieť, že s využitím tejto hodnoty bola väčšia časť komunikácie odfiltrovaná. Konkrétne bol počet všetkých otlačkov zredukovaný z 1830 na 581, čo predstavuje pokles o 68%. Tým pádom klesol aj počet rozdielnych a unikátnych otlačkov. Napriek tomu je výsledná unikátnosť percentuálne mierne znížená. Avšak táto unikátnosť má oveľa väčšiu výpovednú hodnotu, pretože otlačky boli vytvorené iba z komunikácie patriacej konkrétnym aplikáciám.

3.3.4 Analýza metódy JA3, SNI a JA3S

Finálna podoba databázy otlačkov spája výhody všetkých spomenutých metód. Najskôr sa na základe `SNI` odfiltruje nežiadúca komunikácia. Následne sa vytvoria `JA3` heše, ktoré sa spájajú s `JA3S` hešmi. Touto kombináciou vznikajú otlačky najlepšie optimalizované pre detekciu. Kvôli odstráneniu nežiadúcej komunikácie sa výsledný počet otlačkov radikálne zmenší, ale zároveň vďaka `JA3S` je ich presnosť vysoká.

Výsledky prezentované v tabuľke 3.4 jasne potvrdzujú výroky z predchádzajúceho odstavca. Celkový počet otlačkov klesol presne ako pri použití metódy `JA3 + SNI`. Avšak v porovnaní s touto metódou sa počet rozdielnych otlačkov zvýšil o 91% na 67. Táto skutočnosť demonštruje dôsledky použitia `JA3S`. Unikátnych otlačkov je 65, čo predstavuje unikátnosť 97%.

Experimenty popísané v tejto podkapitole ukazujú dôležitosť použitia ďalších techník pri práci s `JA3` otlačkami. Zároveň ukazujú, že kvalita výslednej databáze je podmienená kvalitou vstupných dát.

3.4 Zhrnutie

V tejto kapitole bol popísaný proces tvorby databázy otlačkov. Podrobne boli vysvetlené všetky jeho časti. Pri vytváraní boli použité rôzne techniky, ktoré majú za dôsledok zníženie počtu a zvýšenie kvality otlačkov v databáze. Všetky tieto techniky boli podrobne zanalyzované a ich výsledky boli demonštrované na praktických ukážkach.

Kapitola 4

Popis datasetu

V tejto kapitole je popísaná ako štruktúra a obsah, tak aj štatistiky datasetu vytvoreného v rámci tejto bakalárskej práce. Zároveň sú diskutované otlačky vytvorené z týchto dát.

4.1 Dataset

Dataset slúžiaci ako základ databázy otlačkov je neoddeliteľnou súčasťou detekcie. Je tvorený množinou stodvadsiatich `PCAP` súborov, ktoré obsahujú záznamy internetovej komunikácie vybraných mobilných aplikácií. Táto podkapitola popisuje jednotlivé časti datasetu spolu so štatistikami a zaujímavými faktami z ich analýzy.

Dataset obsahuje záznamy sieťovej komunikácie desiatich aplikácií v troch verziách. Aplikácie boli vybrané z rôznych domén použitia, s cieľom porovnať ich vlastnosti. Komunikácia bola zachytávaná na dvoch fyzických zariadeniach a dvoch emulátoroch. Popis zachytávania komunikácie je popísaný v podkapitole 3.1.

Pre prehľadnosť sú štatistiky každej aplikácie spísané v tabuľke v samostatnej sekcii. Štatistiky v tabuľke obsahujú informácie o počte paketov, počte TLS paketov, percentuálne zastúpenie protokolu TLS v záznamoch, počte TLS Handshake, ClientHello a ServerHello správ. V rámci ClientHello a ServerHello je v zátvorke počet správ, ktoré boli použité na vytvorenie otlačkov.

Každá sekcia obsahuje popis aplikácie, kategóriu v `ObchodPlay`[1], spôsob, akým aplikácia komunikuje do internetu, zaujímavosti zo štatistík a počet unikátnych otlačkov vytvorených pre každú aplikáciu. `ObchodPlay` je oficiálne úložisko a poskytovateľ aplikácií pre zariadenia s operačným systémom Android.

Každý riadok tabuľky predstavuje jeden `PCAP` súbor z datasetu. Názov súborov je v podobe `${názov_aplikácie}-${verzia_aplikácie}.pcap`. Cesta k súborom je vo formáte `pcaps/[emulator|device]/android${verzia}/${názov_aplikácie}`.

Typ zariadenia a verzia Androidu je v tabuľkách zakódovaný v stĺpčeku `Zariadenie`. Tieto hodnoty sú oddelené pomlčkou. Typ `D` (Device) predstavuje fyzické zariadenie a typ `E` predstavuje emulátor. Druhá hodnota znázorňuje verziu Androidu, a to vo formáte `A${verzia_androidu}`. Napríklad hodnota `E-A9` predstavuje emulátor s Androidom verzie deväť.

Všetky záznamy internetovej komunikácie majú dĺžku desať sekúnd. Táto hodnota bola stanovená experimentálne, aby bolo možné vykonať základnú funkčnosť každej aplikácie. V prípade, že by bola dĺžka komunikácie príliš krátka, nastáva riziko, že nebudú vytvorené

[1] https://play.google.com/store/

všetky otlačky. Dôsledok pre detekciu bude jej nepresnosť. Ak bude naopak komunikácia príliš dlhá, budeme mrhať procesorovým časom, pretože sa budú vytvárať dokola tie isté otlačky.

Aliexpress

Prvou aplikáciou ktorej komunikácia bola zachytávaná je Aliexpress. Táto aplikácia patrí do kategórie `Nakupovanie`. Poskytuje rôznorodý sortiment, najmä elektroniku. Tovar pochádza prevažne z Číny. Komunikuje do internetu za účelom získania informácií o ponúkanom tovare a predajcoch.

Aliexpress generoval najviac internetovej komunikácie zo všetkých aplikácií. To je dôsledok veľkého množstva reklám a informácií v podobe predávaných položiek spolu s recenziami a informáciami o predajcoch. Zo štatistík v tabuľke 4.1 možno vidieť, že priemerný počet paketov generovaných emulátorom je nižší ako pri fyzických zariadeniach. Pri emulátoroch je priemerný počet odchytených paketov 2377. Pri fyzických zariadeniach je priemerný počet paketov až 5989. Jeden z dôvodov tak veľkého rozdielu medzi emulátorom a fyzickým zariadením môže byť komunikácia so servermi patriacimi konkrétnym výrobcom mobilných zariadení. Ďalšou príčinou je obrovské množstvo reklám, ktoré sa dynamicky menia. Oba typy komunikácie patria medzi nevyžiadanú a pri vytváraní otlačkov sú odfiltrované, viď sekcia 3.2.1.

Z datasetu bolo pre aplikáciu vytvorených 24 unikátnych otlačkov. Je to najväčší počet zo všetkých aplikácií. Otlačky mali tendenciu pribúdať ako s rôznou verziou Androidu, tak aj s rôznou verziou aplikácie. Časová stabilita týchto otlačkov je pomerne malá, a tak s novou verziou aplikácie i Androidu bude treba získať nové dáta a databázu otlačkov aktualizovať.

Verzia	Zariadenie	Pakety	TLS	TLS[%]	Handshake	ClientHello	ServerHello
7.8.3	D-A8	4454	1003	23	268	52 (40)	52 (40)
	D-A10	10300	3430	33	195	46 (18)	47 (18)
	E-A9	2830	590	21	278	56 (14)	56 (14)
	E-A10	1832	395	22	130	34 (14)	34 (14)
7.7.0	D-A8	4764	1085	23	385	74 (57)	74 (57)
	D-A10	4665	1598	34	220	49 (17)	49 (17)
	E-A9	2695	545	20	244	51 (14)	50 (14)
	E-A10	2127	461	22	164	39 (11)	39 (11)
6.23.1	D-A8	11111	3676	33	251	47 (40)	47 (40)
	D-A10	638	151	24	66	17 (9)	16 (9)
	E-A9	2658	516	19	243	50 (15)	49 (15)
	E-A10	2118	446	21	140	37 (11)	37 (11)

Tabuľka 4.1: Aliexpress.

Bolt

Ďalšou skúmanou aplikáciou je Bolt. Patrí do kategórie `Mapy a Navigácia`. Primárne slúži ako náhrada taxi služieb v mestách. Zároveň poskytuje možnosť krátkodobého prenájmu auta či kolobežky. Komunikácia Boltu tvorí najmä periodickú aktualizáciu polohy áut a kolobežiek.

Zo štatistík v tabuľke 4.2 vidno, že protokol TLS má v dátach najmenšiu relatívnu početnosť. Hodnoty sa tu pohybujú od 16% do 28%. Počet paketov je však priemerný, a

tak sa v dátach nachádza dostatok informácií pre vytvorenie otlačkov a detekciu. Ak by však komunikácie bolo menej a percentuálne zastúpenie protokolu TLS by bolo malé, mohlo by to viesť k ťažkostiam pri vytváraní otlačkov a následne nepresnej detekcii.

Pre aplikáciu boli vytvorené 2 unikátne otlačky. Nebol pozorovaný žiadny konkrétny vzťah medzi otlačkami a verziou aplikácie alebo Androidu.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
5.76	D-A8	1228	342	28	109	20 (1)	20 (1)
	D-A10	2225	511	23	62	27 (1)	26 (1)
	E-A9	1227	321	26	101	21 (1)	21 (1)
	E-A10	1753	279	16	63	22 (1)	22 (1)
6.09	D-A8	1258	396	31	96	19 (1)	20 (1)
	D-A10	1362	318	23	48	21 (1)	21 (1)
	E-A9	1167	307	26	95	20 (1)	20 (1)
	E-A10	1730	276	16	59	21 (1)	20 (1)
6.11	D-A8	1515	423	28	149	30 (1)	30 (1)
	D-A10	1398	305	22	57	25 (1)	25 (1)
	E-A9	1134	302	27	93	19 (1)	19 (1)
	E-A10	1844	306	17	66	22 (1)	22 (1)

Tabuľka 4.2: Bolt.

Pre aplikáciu Bolt bola vykonaná podrobná analýza jednotlivých otlačkov. Špecifikuje pôvod dát, z ktorých sa otlačky vytvorili. Pre aplikáciu boli vytvorené dva JA3 otlačky a dva JA3S otlačky. Tabuľka 4.3 zobrazuje otlačky (nad hrubou horizontálnou čiarou JA3, pod ňou JA3S) a ich výskyt na konkrétnych zariadeniach s konkrétnou verziou Androidu. Kódovanie zariadení v záhlaví tabuľky je rovnaké ako pri tabuľkách so štatistikami. Aplikácia obsahovala vo všetkých troch verziách zhodné otlačky, preto táto informácia nie je v tabuľke uvedená. Značka `X` označuje, že otlačok bol vytvorený z dát aplikácie v špecifikovanej verzii, na špecifikovanom zariadení a so špecifickou verziou Androidu.

Takáto analýza nám vie dať informáciu o tom, ako sa otlačky menili pri rôznych verziách aplikácie a Androidu. V tabuľke vidno, že jeden JA3 otlačok sa objavoval iba v zariadeniach s verziou 8 a 9. Druhý sa objavoval iba v zariadeniach s Androidom verzie 10. Trend závislosti JA3 otlačku na verzie Androidu bol spozorovaný vo viacerých aplikáciách. JA3S otlačky tejto aplikácie sa chovajú obdobne.

Priestorová náročnosť zobrazenia tejto informácie v prípade, že otlačky sa líšia medzi verziami aplikácie je pomerne veľká a s väčším množstvom otlačkov by boli tabuľky neprehľadné. Preto pre ďalšie aplikácie tabuľky s analýzou jednotlivých otlačkov uvádzané nebudú. Výsledky analýzy otlačkov budú pri každej aplikácii stručne zhrnuté.

JA3[S] Otlačok	D-A8	D-A10	E-A9	E-A10
e1330d9d9c9fe3586c1c8c08ffedf63e	X	-	X	-
fada0859379fec2c87b490b8203dc520	-	X	-	X
4eb9934558faa4e61eb16ef5e93574f0	X	-	X	-
15af977ce25de452b96affa2addb1036	-	X	-	X

Tabuľka 4.3: Otlačky aplikácie Bolt.

Duolingo

Aplikácia Duolingo patrí do kategórie `Vzdelávanie`. Konkrétne sa zameriava na zdokonalenie sa v cudzích jazykoch. Každodennými rýchlymi úlohami sa snaží motivovať užívateľa učiť sa hlavne používané frázy a novú slovnú úlohu. Tieto úlohy sú sťahované z internetu. Aplikácia poskytuje možnosť zdieľania úspechov na sociálne siete, čím vzniká ďalšia komunikácia. Štatisticky neboli pri tejto aplikácii pozorované žiadne extrémy.

Pre aplikáciu bolo vytvorených 5 unikátnych otlačkov. Otlačky sa menili ako s verziou Androidu, tak aj so zmenou hlavnej (major) verzie aplikácie. Tri otlačky boli získané zo zariadení s aplikáciou vo verzii 3.106.5. Zvyšné dva boli získané z aplikácie vo verzii 4.X.X.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
3.106.5	D-A8	703	321	46	64	17 (2)	17 (2)
	D-A10	669	304	45	43	14 (2)	14 (2)
	E-A9	112	35	31	10	2 (1)	2 (1)
	E-A10	449	150	33	29	10 (2)	10 (2)
4.80.3	D-A8	301	110	37	51	12 (1)	12 (1)
	D-A10	475	142	30	28	11 (1)	11 (1)
	E-A9	215	69	32	33	6 (2)	6 (2)
	E-A10	263	71	27	14	7 (2)	7 (2)
4.81.4	D-A8	306	109	36	51	12 (1)	12 (1)
	D-A10	297	108	36	25	9 (1)	10 (1)
	E-A9	251	82	33	42	8 (2)	8 (2)
	E-A10	284	83	29	16	8 (2)	8 (2)

Tabuľka 4.4: Duolingo.

Discord

Discord je aplikácia na četovanie v dvojiciach alebo tímoch. Patrí do kategórie `Komunikácia`. Pôvodne cielila na hráčov, avšak pre jej bohatú funkcionalitu je posledné roky viac a viac využívaná naprieč rôznymi sektormi na komunikáciu v tíme. Komunikácia s internetom vzniká pri načítaní správ a informácií o užívateľoch a serveroch. V súčasnosti má vlastný Discord server aj fakulta, na ktorej táto bakalárska práca vznikla.

Komunikácia tejto aplikácie je priemerná vzhľadom na komunikáciu ostatných aplikácií v dátovej sade. Jediným faktom stojacim za zmienku je, že vykazuje štatisticky najväčší priemerný percentuálny podiel protokolu TLS o hodnote 38%. Najväčšia hodnota v jednom zázname bola až 51% na fyzickom zariadení s Androidom verzie 8.

Pre aplikáciu boli vytvorené 3 unikátne finálne otlačky. Sú tvorené dvomi JA3 a tromi JA3S otlačkami. Aj pri tejto aplikácii bol JA3 otlačok závislý na verzii Androidu. Jeden JA3 otlačok obsahovala aplikácia vo všetkých verziách, na zariadeniach s Androidom 8 a 9. Druhý JA3 otlačok obsahovala aplikácia iba na zariadeniach s Androidom 10. Pri otlačku JA3S nebola pozorovaná žiadna korelácia s či už s verziou Androidu, aplikácie alebo typom zariadenia.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
40.8	D-A8	330	143	43	59	12 (2)	12 (2)
	D-A10	485	193	40	52	17 (2)	17 (2)
	E-A9	277	98	35	42	8 (2)	8 (2)
	E-A10	254	77	30	14	7 (2)	7 (2)
41.10	D-A8	1385	194	14	46	12 (2)	12 (2)
	D-A10	413	154	37	25	9 (2)	9 (2)
	E-A9	272	114	42	30	6 (2)	6 (2)
	E-A10	278	106	38	12	6 (2)	6 (2)
41.11	D-A8	239	122	51	26	5 (4)	5 (4)
	D-A10	365	161	44	23	9 (2)	9 (2)
	E-A9	371	171	46	40	8 (2)	8 (2)
	E-A10	273	108	40	12	6 (2)	6 (2)

Tabuľka 4.5: Discord.

LinkedIn

LinkedIn je profesijná sociálna sieť, ktorá spája ľudí hľadajúcich prácu a zamestnávateľov po celom svete. Nachádza sa v kategórii `Biznis`. Jej hlavným benefitom je sociálny rozmer, ktorý môže profesijný život veľmi obohatiť. Komunikácia spočíva v načítavaní správ, statusov, informácií o užívateľoch a pracovných ponúk.

Štatisticky táto aplikácia generuje malé množstvo paketov. Tu sa potvrdzuje tvrdenie, že počet paketov je do veľkej miery ovplyvňovaný počtom reklám. Keďže táto aplikácia neobsahuje žiadnu reklamu, tak komunikácie je menej. Počet paketov sa tu pohyboval medzi hodnotami 67 a 244. Vo všetkých troch verziách bolo viac paketov vytvorených fyzickým zariadením. Jediný výrazný výkyv nastal vo verzii 4.1.497 pri fyzickom zariadení s Androidom 8, kde bolo zaznamenaných 1928 paketov. Avšak z nich iba 64 paketov obsahovalo protokol TLS.

Pre aplikáciu bolo vytvorených 5 unikátnych otlačkov. Otlačky sa menili ako s verziou aplikácie, tak aj s verziou Androidu.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
4.0.100	D-A8	186	75	40	15	5 (1)	5 (1)
	D-A10	244	96	39	30	8 (4)	8 (4)
	E-A9	136	52	38	16	4 (4)	4 (4)
	E-A10	150	48	32	16	4 (4)	4 (4)
4.1.236	D-A8	186	75	40	15	5 (1)	5 (1)
	D-A10	233	99	42	1	8 (1)	8 (1)
	E-A9	101	32	32	9	2 (1)	2 (1)
	E-A10	114	35	31	9	2 (1)	2 (1)
4.1.497	D-A8	1928	64	3	8	3 (1)	3 (1)
	D-A10	172	66	38	16	4 (1)	4 (1)
	E-A9	67	22	33	4	1 (1)	1 (1)
	E-A10	126	40	32	13	3 (2)	3 (2)

Tabuľka 4.6: LinkedIn.

NetAtmo

NetAtmo je aplikácia poskytujúca informácie o počasí, a teda patrí do kategórie `Počasie`. Jej komunikácia pozostáva z načítavania aktuálnych meteorologických informácií. V štatistikách neboli zistené žiadne extrémy.

Pre aplikáciu boli vytvorené 5 unikátnych otlačkov. Tvorené boli dvoma JA3 a troma JA3S otlačkami. Podobne ako pri aplikáciách Bolt a Discord, jeden JA3 otlačok prislúchal zariadeniam s Androidom verzie 8 a 9. Druhý otlačok sa objavoval výhradne na zariadeniach s Androidom 10. JA3S otlačky neboli závislé ani na verzii aplikácie, ani na verzii Androidu.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
2.8.1	D-A8	2887	217	8	38	8 (3)	8 (3)
	D-A10	778	316	41	43	12 (7)	12 (7)
	E-A9	136	41	30	19	4 (2)	4 (2)
	E-A10	145	39	27	13	4 (2)	4 (2)
2.8.2	D-A8	496	230	46	41	8 (3)	8 (3)
	D-A10	661	285	43	30	8 (3)	8 (3)
	E-A9	94	26	28	14	3 (2)	3 (2)
	E-A10	90	24	27	11	3 (2)	3 (2)
3.0.0	D-A8	355	198	56	25	5 (1)	5 (1)
	D-A10	413	212	51	21	6 (1)	6 (1)
	E-A9	120	38	32	20	4 (1)	4 (1)
	E-A10	138	39	28	11	4 (1)	4 (1)

Tabuľka 4.7: NetAtmo.

Notecalc

Názov Notecalc napovedá, že sa bude jednať o spojenie kalkulačky a poznámkového bloku. Výpočty sú interaktívne evaluované a je ich možno jednoducho uložiť pre neskoršie použitie. Aplikácia patrí do kategórie `Nástroje`.

Aj keď sa môže zdať, že kalkulačka nemá dôvod pristupovať na internet, táto aplikácia obsahuje druhé najväčšie množstvo paketov. Je to dôsledok masívneho obsahu reklám. Touto cestou sa snažia tvorcovia aplikácie presvedčiť užívateľov, aby si kúpili platenú verziu, ktorá reklamy neobsahuje.

Pre aplikáciu boli vytvorené 3 unikátne otlačky. JA3S otlačok bol vo všetkých finálnych otlačkoch rovnaký. JA3 otlačky boli tým pádom 3. Dva z nich sa objavovali výhradne na zariadeniach s Androidom 10. Posledný sa objavoval v zariadeniach s Androidom verzie 8 a 9.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
2.19.53	D-A8	834	365	44	96	18 (7)	18 (7)
	D-A10	1000	419	42	106	25 (7)	26 (7)
	E-A9	1033	490	47	104	20 (7)	20 (7)
	E-A10	1262	488	39	72	21 (7)	21 (7)
2.20.59	D-A8	1124	186	17	73	16 (4)	16 (4)
	D-A10	1255	499	40	133	32 (9)	31 (9)
	E-A9	1213	534	44	128	24 (9)	24 (9)
	E-A10	1383	514	37	87	24 (9)	24 (9)
2.21.60	D-A8	191	87	73	23	5 (4)	5 (4)
	D-A10	2413	909	38	125	27 (9)	27 (9)
	E-A9	1296	576	44	137	26 (9)	26 (9)
	E-A10	1658	570	34	100	27 (9)	27 (9)

Tabuľka 4.8: Notecalc.

Reddit

Reddit je otvorená sociálna sieť fungujúca na princípe prezerania obsahu užívateľom a jeho následného hodnotenia pomocou hlasovania. To zaručuje, že zaujímavý obsah je zobrazovaný ako prvý. Patrí do kategórie `Sociálne siete`. Z internetu získava aplikácia v podstate všetky zobrazované informácie, kam patria hlavne príspevky s hodnotením a komentármi, ale aj užívatelia a informácie o nich. Štatistiky z komunikácie aplikácie neobsahujú žiadne extrémy.

Pre aplikáciu boli vytvorené 4 unikátne otlačky. Boli tvorené dvoma JA3 a troma JA3S otlačkami. Trend závislosti JA3 otlačku na verzii Androidu sa potvrdil aj tu. Dva otlačky pochádzali zo zariadení s Androidom 10 a dva zo zariadení s Androidom vo verzii 8 a 9. Otlačok JA3S sa menil ako s verziou aplikácie, tak aj s verziou Androidu.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
2020.36.0	D-A8	681	252	37	65	14 (2)	14 (2)
	D-A10	3610	1280	35	73	20 (7)	20 (7)
	E-A9	1122	418	37	102	18 (10)	18 (10)
	E-A10	1616	540	33	65	16 (9)	16 (9)
2020.34.0	D-A8	419	166	40	59	13 (1)	13 (1)
	D-A10	616	225	37	43	16 (2)	16 (2)
	E-A9	842	370	44	96	17 (9)	17 (9)
	E-A10	1195	385	32	60	15 (8)	15 (8)
2020.27.0	D-A8	944	451	48	90	18 (9)	18 (9)
	D-A10	1681	628	37	118	27 (15)	27 (15)
	E-A9	1168	413	35	98	17 (11)	17 (11)
	E-A10	233	56	24	21	6 (2)	6 (2)

Tabuľka 4.9: Reddit.

Slack

Slack je aplikácia určená na komunikáciu v tíme. Jej cieľovou skupinou sú z veľkej časti malé a stredne veľké firmy. Vznikla s cieľom nahradiť neefektívnu emailovú korešpondenciu tam, kde v rámci tímov spolupracujú užšie skupiny ľudí. Patrí do kategórie `Biznis`.

Jej internetová komunikácia obsahuje najmä správy, spolu s informáciách o skupinách a užívateľoch.

Štatisticky táto aplikácia generuje podobne ako LinkedIn malé množstvo paketov. Počet paketov sa tu pohyboval medzi hodnotami 140 a 302. Tak isto ako u LinkedIn viac paketov bolo generovaných fyzickým zariadením.

Pre aplikáciu boli vytvorené 5 unikátnych otlačkov. Opakuje sa ten istý trend ako pri predchádzajúcej aplikácii Reddit. JA3 otlačky sú dva a sú závislé na verzii Androidu. JA3S otlačkov je päť, menia sa ako s verziou Androidu, tak aj s verziou aplikácie.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
19.12.10	D-A8	302	97	32	40	11 (2)	11 (2)
	D-A10	261	94	36	34	11 (3)	11 (3)
	E-A9	153	48	31	25	5 (3)	5 (3)
	E-A10	148	44	30	8	4 (3)	4 (3)
20.08.30	D-A8	293	97	33	45	12 (2)	12 (2)
	D-A10	216	79	37	20	7 (2)	7 (2)
	E-A9	152	50	33	25	5 (3)	5 (3)
	E-A10	142	46	32	8	4 (3)	4 (3)
20.09.10	D-A8	217	79	36	35	9 (2)	9 (2)
	D-A10	294	97	33	30	12 (3)	12 (3)
	E-A9	161	55	34	25	5 (3)	5 (3)
	E-A10	140	43	31	8	4 (3)	4 (3)

Tabuľka 4.10: Slack.

Tasty

Tasty je aplikácia obsahujúca recepty rozličných jedál. Je zaradená do kategórie `Jedlo a pitie`. Komunikácia obsahuje najmä recepty a ich detaily.

Štatistiky nevykazujú žiadne extrémy. Počet paketov je mierne nižší oproti priemeru. Opäť prevažuje trend, že fyzické zariadenie obsahuje viac komunikácie ako emulátor.

Pre aplikáciu boli vytvorené 2 unikátne otlačky. Čo sa týka JA3 otlačkov, opakuje sa ten istý trend ako pri predchádzajúcich aplikáciách Reddit a Slack. Jeden z dvojice JA3 pochádza zo zariadení s Androidom 10, druhý zo zariadení s Androidom 8 a 9. Pozorovaný bol iba jeden JA3S otlačok.

Verzia	Zariadenie	Pakety	TLS	% TLS	Handshake	ClientHello	ServerHello
1.19	D-A8	426	148	35	56	13 (1)	13 (1)
	D-A10	306	103	34	47	13 (1)	13 (1)
	E-A9	236	78	33	42	8 (1)	8 (1)
	E-A10	335	90	27	29	9 (1)	9 (1)
1.38	D-A8	385	131	34	46	11 (1)	11 (1)
	D-A10	437	160	37	37	13 (1)	13 (1)
	E-A9	247	88	36	47	9 (1)	9 (1)
	E-A10	245	77	31	24	8 (1)	8 (1)
1.39	D-A8	930	340	37	58	12 (1)	12 (1)
	D-A10	406	163	40	40	12 (1)	12 (1)
	E-A9	252	90	36	47	9 (1)	9 (1)
	E-A10	281	87	31	26	9 (1)	9 (1)

Tabuľka 4.11: Tasty.

4.2 Zhrnutie

V tejto kapitole bol popísaný vytvorený dataset spolu s jeho štatistikami. Dáta boli analyzované a zaujímavé fakty pre detekciu boli vypísané v jednotlivých sekciách podkapitoly 4.1. Bola diskutovaná potrebná dĺžka odchytenej komunikácie a problémy, ktoré môžu nastať pri jej nesprávnom zvolení. Zároveň bolo experimentami zistené, že stabilita JA3 a teda aj finálnych otlačkov je viac závislá na verzii Androidu, ako na verzii aplikácie. Aplikácie mali vo väčšine prípadov rovnaké otlačky pri všetkých troch verziách. Avšak s verziou Androidu sa otlačky menili.

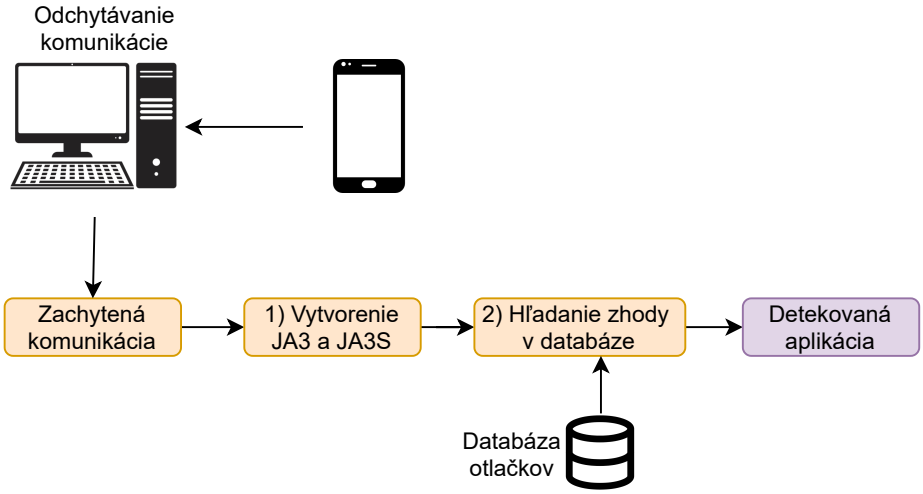
Kapitola 5

Detekcia aplikácií a testovanie

V tejto kapitole je vysvetlený princíp detekcie aplikácií. Následne sú diskutované problémy, ktoré môžu nastať a ich riešenie. Ďalej je popísané testovanie ako na známych, tak aj na neznámych dátach.

5.1 Princíp detekcie aplikácií

Detekcia aplikácií pozostáva zo zachytenia komunikácie, extrakcie dát z protokolu `TLS`, vytvorenia `JA3` a `JA3S` otlačkov a následného porovnania hodnôt s databázou otlačkov. Diagram detekcie je na obrázku 5.1.



Obr. 5.1: Diagram detekcie.

Otlačky sa vytvárajú takým istým spôsobom ako pri tvorbe databázy popísanej v kapitole 3. Zhoda v databáze sa hľadá na základe dvojice otlačok (JA3-JA3S) a názov servera (SNI). Dôvody použitia tejto dvojice a nie len otlačku samotného sú popísané v sekcii 5.1.1. Ak sa v databáze nájde zhoda, na výstup putuje detegovaná aplikácia. Ak sa zhoda nenájde, je komunikácia označená ako neznáma. Všetky TLS spojenia iniciované reklamami a aplikácie, ktoré databáza neobsahuje, budú označené ako neznáma komunikácia.

5.1.1 Problémy pri detekcii

V databáze aplikácií môže nastať situácia, kedy viacero aplikácií bude mať rovnaký otlačok. Tomuto javu sa snažíme predchádzať, avšak otlačky sú tvorené parametrami TLS komunikácie, ktoré môžu byť rovnaké. Preto sa pri detekcii porovnáva aj hodnota `SNI`.

Táto situácia nastala aj pri dvoch aplikáciách v databáze vytvorenej v rámci tejto bakalárskej práce. Inicializácia TLS spojenia aplikácií Reddit a Tasty obsahovala rovnaké parametre, a tak pri ich detekcii nastávala nejednoznačnosť. Obe aplikácie majú dva rovnaké otlačky. Konkrétne otlačky sú zakódované vo formáte JA3-JA3S:

- fada0859379fec2c87b490b8203dc520-05c6c275b10b37e3d5561f48e80d659c
- e1330d9d9c9fe3586c1c8c08ffedf63e-05c6c275b10b37e3d5561f48e80d659c

Je nutné podotknúť, že JA3S otlačky, ktoré majú za úlohu spresniť JA3 otlačky, sú v oboch finálnych otlačkoch rovnaké, a tak v tomto prípade detekcii vôbec nepomáhajú.

Pridaním hodnoty `SNI` do detekcie bola docielená schopnosť detegovať aj aplikácie s rovnakým otlačkom. Tabuľka 5.1 zobrazuje hodnoty `SNI` prislúchajúce zhodným otlačkom.

Aplikácia	SNI
Reddit	gql.reddit.com
	www.reddit.com
	gateway.reddit.com
	www.redditstatic.com
	styles.redditmedia.com
	preview.redd.it
Tasty	api.tasty.co

Tabuľka 5.1: Hodnoty SNI pre kolízne aplikácie.

5.2 Testovanie

Táto podkapitola popisuje testovanie aplikácie a úspešnosť detekcie. Testovanie je rozdelené na dve časti, podľa pôvodu vstupných dát. V prvej sa bude testovať na videných dátach, z ktorých sa vytvorila aplikácia otlačkov. V druhej časti bude identifikácia aplikácii vykonávaná na nevidených dátach, pochádzajúcich z reálneho prostredia.

Na testovanie slúžil skript `Application-detector`. Jeho popis a použitie sa nachádza v prílohe B. Skript má dva parametre: databázu otlačkov a dáta, v ktorých chceme detegovať mobilné aplikácie. Detekcia prebieha podľa postupu popísaného v podkapitole 5.1.

Výsledky testovania budú zobrazené v tabuľkách v sekcii 5.2.1 a 5.2.2. Aplikácie sú kvôli úspore priestoru kódované rímskymi číslami v abecednom poradí. Tabuľka 5.2 zobrazuje mapovanie aplikácií a rímskych čísel.

Označenie	Aplikácia
I	Aliexpress
II	Bolt
III	Discord
IV	Duolingo
V	LinkedIn
VI	NetAtmo
VII	Notecalc
VIII	Reddit
IX	Slack
X	Tasty
XI	Neznáma aplikácia

Tabuľka 5.2: Kódovanie aplikácií v tabuľkách.

Pre určenie úspešnosti detekcie budú použité štyri rôzne hodnoty. Prvé dve hodnoty značia úspech detekcie, druhé dve jej neúspech. Ich definícia je nasledovná:

- True positive (TP): Aplikácia bola súčasťou dát a bola detegovaná.
- True negative (TN): Aplikácia nebola súčasťou dát a nebola detegovaná. Táto hodnota predstavuje počet, koľko krát bola úspešne detegovaná neznáma komunikácia.
- False positive (FP): Aplikácia nebola súčasťou dát a bola detegovaná.
- False negative (FN): Aplikácia bola súčasťou dát, ale nebola detegovaná.

Tieto hodnoty sa použijú pre výpočet presnosti (accuracy), precíznosti (precision) a úplnosti (recall) detekcie. Presnosť určuje aký podiel otlačkov sme detegovali správne.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precíznosť udáva aká spoľahlivá je detekcia. Zjednodušene je to podiel správne detegovaných aplikácií ku všetkým detegovaným aplikáciám.

$$Precision = \frac{TP}{TP + FP}$$

Úplnosť určuje aký podiel všetkých výskytov aplikácií odhalíme.

$$Recall = \frac{TP}{TP + FN}$$

5.2.1 Testovanie na známych dátach

Testovanie na známych dátach má za úlohu zistiť, či identifikácia aplikácií funguje spoľahlivo na dátach, z ktorých bola vytvorená databáza otlačkov. Zároveň takéto testovanie dokáže spoľahlivo odhaliť nejednoznačnosť pri detekcii.

Výsledky testovania sú v tabuľke 5.3. Tá obsahuje maticu zámen [1] (confusion matrix). Každý riadok predstavuje vstupné dáta s komunikáciou jednej aplikácie vo všetkých troch verziách. Stĺpce tabuľky predstavujú detegované aplikácie. Čísla v tabuľke určujú, koľko krát bola aplikácia detegovaná.

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI
I	260	-	-	-	-	-	-	-	-	-	303
II	-	12	-	-	-	-	-	-	-	-	255
III	-	-	24	-	-	-	-	-	-	-	78
IV	-	-	-	19	-	-	-	-	-	-	97
V	-	-	-	-	22	-	-	-	-	-	27
VI	-	-	-	-	-	24	-	-	-	-	45
VII	-	-	-	-	-	-	90	-	-	-	175
VIII	-	-	-	-	-	-	-	85	-	-	121
IX	-	-	-	-	-	-	-	-	32	-	57
X	-	-	-	-	-	-	-	-	-	12	114

Tabuľka 5.3: Identifikácia aplikácií na známych dátach - matica zámen.

Celkový počet TP pre všetky aplikácie je 585. Počet FP je nula. Dôkazom toho je, že matica zámen má pri zanedbaní detekcie neznámych aplikácií (stĺpec XI) diagonálny tvar. To znamená, že žiadna aplikácia nebola identifikovaná nesprávne. Počet FN je opäť nulový. To sa dá overiť súčtom použitých ClientHello správ (hodnoty v zátvorke) z tabuliek v podkapitole 4.1 a následným porovnaním s hodnotou z matice zámen.

Tabuľka 5.4 zobrazuje výslednú presnosť, precíznosť a úplnosť. Všetky hodnoty sa rovnajú 100%. Tento výsledok bol očakávaný, keďže vstupom pre detekciu sú dáta, z ktorých bola vytvorená databáza otlačkov. Táto skutočnosť nám potvrdzuje, že metóda identifikácie pracuje spoľahlivo, ak databáza obsahuje všetky otlačky, ktoré sa v dátach nachádzajú.

Accuracy	Precision	Recall
100%	100%	100%

Tabuľka 5.4: Výsledky identifikácie aplikácií na známych dátach.

[1] https://en.wikipedia.org/wiki/Confusion_matrix

5.2.2 Testovanie na neznámych dátach

Pre testovanie na neznámych dátach bol vytvorený záznam komunikácie desiatich skúmaných aplikácií. Na konci tohto záznamu je komunikácia jednej aplikácie, ktorá sa v databáze otlačkov nenachádzala. Záznam pochádza z fyzického zariadenia s Androidom vo verzii 10. Beh aplikácií na pozadí bol na mobilnom zariadení zakázaný.

Po získaní komunikácie ju bolo nutné anotovať. Každá aplikácia komunikovala 10 sekúnd a medzi jednotlivými aplikáciami bola 5 sekundová pauza. Aplikácie boli spúšťané v abecednom poradí. Pomocou týchto informácií bolo možné identifikovať komunikáciu patriacu každej aplikácii. Tieto údaje budú využité neskôr pri evaluácii kvality detekcie.

Výsledky testovania sú v tabuľke 5.5. Tá obsahuje maticu zámen. Každý riadok predstavuje komunikáciu jednej aplikácie. Stĺpce tabuľky predstavujú detegované aplikácie. Čísla v tabuľke určujú, koľko krát bola aplikácia detegovaná. Čísla v zátvorke udávajú predpokladaný počet detekcií. Tieto čísla boli získané pri anotovaní zachytenej komunikácie.

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI
I	6 (8)	-	-	-	-	-	-	-	-	-	10
II	-	1 (1)	-	-	-	-	-	-	-	-	21
III	-	-	2 (2)	-	-	-	-	-	-	-	11
IV	-	-	-	1 (11)	-	-	-	-	-	-	40
V	-	-	-	-	2 (2)	-	-	-	-	-	16
VI	-	-	-	-	-	1 (1)	-	-	-	-	5
VII	-	-	-	-	-	-	-	-	-	-	15
VIII	-	-	-	-	-	-	-	10 (10)	-	-	10
IX	-	-	-	-	-	-	-	-	2 (2)	-	7
X	-	-	-	-	-	-	-	-	-	1 (1)	18
XI	-	-	-	-	-	-	-	-	-	-	15

Tabuľka 5.5: Identifikácia aplikácií na neznámych dátach - matica zámen.

Z matice vidno, že aplikáciu Notecalc (stĺpec VII) sa nepodarilo detegovať ani raz. Avšak aj očakávaný počet detekcií je nula. Po podrobnejšej analýze komunikácie tejto aplikácie bolo zistené, že komunikácia podľa ktorej je možné aplikáciu detegovať sa vyskytuje iba tesne po prvom spustení po inštalácii. Následne obsahuje komunikácia iba šum. Táto skutočnosť demonštruje, že aplikácie, pre ktoré nie je typická komunikácia do internetu, nie sú vhodné na detekciu použitou metódou.
Ak zanedbáme detekciu neznámej komunikácie (stĺpec XI), vidíme že matica zámen je diagonálna. Z toho vyplýva, že počet FP je nula. To znamená, že žiadna aplikácia nebola identifikovaná nesprávne, a tak hodnota precíznosti (precision) bude veľká. Celkový počet

TP pre všetky aplikácie je 26. Počet FN je 12. Toto číslo udáva koľko krát sme aplikáciu nedetegovali.

Tabuľka 5.6 zobrazuje výslednú presnosť, precíznosť a úplnosť. Výsledná presnosť 94% značí, že detekcia je do vysokej miery úspešná. Veľmi vysoká precíznosť s hodnotou 100% udáva, že nástroj nikdy nedeteguje nesprávnu aplikáciu. Hodnota úplnosti bola iba 68%. Táto hodnota udáva, aký podiel pozitívnych výskytov aplikácií bolo odhalených.

Relatívne nízka hodnota úplnosti je dôsledok časovej nestability otlačkov, a teda zvýšením počtu FN. Analýza ukázala, že vo väčšine prípadov nedetegovaná komunikácia obsahovala nové JA3S otlačky. So starnúcimi dátami v databáze sa bude úplnosť ďalej zmenšovať. Aktualizáciou databázy otlačkov sa hodnota naopak zvýši. Ak zoberieme do úvahy veľmi vysokú precíznosť, výsledná detekcia bude úspešná hneď po prvom identifikovaní aplikácie. Tým pádom je aj nižšia hodnota úplnosti postačujúca.

Accuracy	Precision	Recall
94%	100%	68%

Tabuľka 5.6: Výsledky identifikácie aplikácií na neznámych dátach.

5.3 Zhrnutie

Na začiatku kapitoly bol vysvetlený princíp detekcie a riešenie možného problému nejednoznačnosti otlačkov.

Následne bolo v dvoch fázach vykonané testovanie detekcie. Prvá časť testovala detekciu na videných dátach, z ktorých bola vytvorená databáza otlačkov. Cieľom bolo zistiť, či je metóda identifikácie mobilných aplikácií na základe JA3 funkčná, a či sú nástroje na vytváranie databázy a detekciu správne implementované. Skúmanými veličinami bola presnosť (accuracy), precíznosť (precision) a úplnosť (recall). Výsledok všetkých troch hodnôt bol 100%. Metóda identifikácie sa tak ukázala ako funkčná.

Druhá časť testovania skúmala identifikáciu mobilných aplikácií na nevidených anotovaných dátach. Cieľom bolo nasimulovať reálne prostredie a zistiť, ako dobre detekcia funguje. Výsledná presnosť bola 94%, precíznosť 100% a úplnosť 68%. Nižšia hodnota úplnosti je dôsledkom starnutia dát v databáze otlačkov. Avšak vysoká hodnota precíznosti nám zaručuje, že detekcia aplikácií bude úspešná, napriek tomu, že databáza neobsahuje všetky otlačky.

Kapitola 6

Záver

Cieľom tejto práce bolo vytvoriť nástroje umožňujúce detekciu prítomnosti mobilných aplikácii v sieťovej komunikácii na základe otlačkov JA3. Výsledné riešenie je funkčné a spĺňa všetky body formálneho zadania.

Na začiatku práce som preštudoval možnosti identifikácie mobilných zariadení na základe otlačkov JA3 a JA3S. Následne som navrhol spôsob automatizovaného vytvárania otlačkov mobilných aplikácií. Finálne otlačky aplikácií sú tvorené pomocou JA3, JA3S a názvu servera (SNI). Experimentálne bolo ukázané, že táto trojica dokáže najspoľahlivejšie identifikovať mobilné aplikácie.

Pri ich vytváraní som diskutoval dve optimalizácie prispievajúce k zlepšeniu detekcie. Konkrétne sa jednalo o odstránenie šumu a nežiadúcich hodnôt z komunikácie. Medzi šum patrí najmä reklama, ale aj komunikácia s analytickými nástrojmi skúmajúca návštevnosť sledovaného obsahu. Využitie a dôležitosť daných optimalizácií bola ukázaná na príkladoch z reálnych dát.

Následne som vytvoril datasety obsahujúce reálnu mobilnú sieťovú komunikáciu. Využil som ako emulátor, tak aj fyzické zariadenie. Cieľom bolo zistiť rozdiely v komunikácii. Odlišnosti boli diskutované, avšak pre detekciu nemali významnú rolu. Otlačky získané z fyzického zariadenia a emulátora boli rovnaké. Z datasetov bola vytvorená databáza otlačkov.

Evaluácia vytvoreného riešenia prebehla v dvoch fázach. V prvej bola detekcia spustená nad videnými dátami, z ktorých bola databáza otlačkov vytvorená. Nástroj na týchto dátach detegoval všetky aplikácie správne, a tak možno konštatovať, že implementácia ako aj metóda detekcia je korektná. Následne boli odchytené nevidené dáta obsahujúce komunikáciu skúmaných aplikácií. Táto komunikácia bola anotovaná a následne na nej prebehla detekcia. Výsledky ukázali čiastočné nepresnosti spôsobené starnutím databáze a časovou nestabilitou otlačkov. Napriek týmto nepresnostiam dokázal vytvorený systém spoľahlivo mobilné aplikácie odhaliť.

Vytvorené riešenie je použiteľné pre monitorovanie sietí, napr. s pomocou technológie IPFIX[1]. Uvažujme architektúru, kde exportéry posielajú IPFIX správy s tokmi na kolektor. Na kolektore sa nachádza databáza otlačkov. Toky obsahujúce protokol TLS sú následne posielané na vstup detekčného skriptu. Na základe týchto informácií by bolo možné regulovať aplikácie, ktoré v sieti komunikujú.

[1] https://en.wikipedia.org/wiki/IP_Flow_Information_Export

Literatúra

[1] Benjamin, D. *Applying Generate Random Extensions And Sustain Extensibility (GREASE) to TLS Extensibility* [online]. Január 2020 [cit. 2021-003-28]. Dostupné z: https://tools.ietf.org/html/rfc8701.

[2] Clement, J. *Mobile internet traffic as percentage of total web traffic* [online]. statista.com, máj 2020 [cit. 2020-11-17]. Dostupné z: https://www.statista.com/statistics/306528/share-of-mobile-internet-traffic-in-global-regions/.

[3] Kwakyi, G. *How Do Mobile Advertising Auction Dynamics Work? Incipia* [online]. statista.co, 2018 [cit. 2021-03-28]. Dostupné z: https://incipia.co/post/app-marketing/how-do-mobile-advertisingauction-dynamics-work/.

[4] Matoušek, P., Burgetová, I., Ryšavý, O. a Victor, M. On Reliability of JA3 Hashes for Fingerprinting Mobile Applications. In:. Springer International Publishing, 2021, sv. 351, s. 1–22. DOI: 10.1007/978-3-030-68734-2_1. ISBN 978-3-030-68733-5.

[5] Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3* [online]. August 2018 [cit. 2020-11-22]. Dostupné z: https://tools.ietf.org/html/rfc8446.

[6] Turner, A. *HOW MANY SMARTPHONES ARE IN THE WORLD?* [online]. bankmycell.com, november 2020 [cit. 2020-11-17]. Dostupné z: https://www.bankmycell.com/blog/how-many-phones-are-in-the-world.

Príloha A

Obsah priloženého pamäťového média

```
/
├── res
│   ├── doc ..... Zdrojové súbory textovej práce
│   ├── pcaps ..... Záznamy internetovej komunikácie na vytvorenie databázy
│   ├── pcap_eval ..... Záznamy internetovej komunikácie na detekciu
│   ├── script_outputs ..... Súbory ktoré boli výstupom skriptov
│   └── database ..... SQL skript pre vytvorenie a naplnenie databázy
└── src
    ├── fingerprint_database_creator.py .. Skript na vytváranie databázy otlačkov
    ├── application_detector.py ..... Skript na detegovanie mobilných aplikácií
    ├── get_sni.py ..... Skript na extrakciu hodnôt SNI
    └── FinalFingerprint.py ..... Pomocná trieda
```

Príloha B

Použitie programu

B.1 Installation

Requirements:

- tshark, version 3+
- python, version 3.7+
- vitrualenv, version 16+
- libxml2, libxslt - packages needed for installing pyshark

Before running script create virtual environment and download dependencies:
`virtualenv .env && source .env/bin/activate && pip install pyshark`

> ***NOTE:*** If you are activating virtual environment from other shell then `sh` or `bash` you need to activate it using correct script. E.g. `.env/bin/activate.csh` for C-shell

B.2 Usage

Fingerprint-database-creator Script for generating fingerprints in CSV format and debug information.

`python fingerprint_database_creator.py [--help] --data <data_source>`

Example: `python fingerprint_database_creator.py --rootdir ../pcaps/`

Arguments:

- –help show help message and exit
- –data
  - - pcap or directory with pcaps

Output files:

- fingerprintsDatabase.csv - CVS file with final fingerprints
- fingerprintsDatabase.txt - debug information about creating fingerprints

Application-detector Script for detection mobile application from pcap file
`python application_detector.py [--help] --data <data_source> --database <csv_file>`

Example: `python application_detector.py --data pcap_eval/eval_comm.pcap --database fingerprintsDatabase.csv`

Arguments:

- –help show help message and exit
- –data
  - - pcap or directory with pcaps
- –database
  - csv file containing database of fingerprints

Output files:

- sni.txt - file with SNI values

SNI-reader Script for obtaining SNI from Client Hello packets for further analysis
`python get_sni.py [--help] --data <data_source>`

Example: `python get_sni.py --data ../pcaps/`

Arguments:

- –help show help message and exit
- –data
  - - pcap or directory with pcaps

Output files:

- detection_results.txt - file with detected apps