Detekce mobilních zařízení v síťové komunikaci

Technická zpráva FIT VUT v Brně

Petr Matoušek

Technická zpráva č. FIT-TR-2017-08
Fakulta informačních technologií, Vysoké učení technické v Brně

Poslední změna: únor 2018

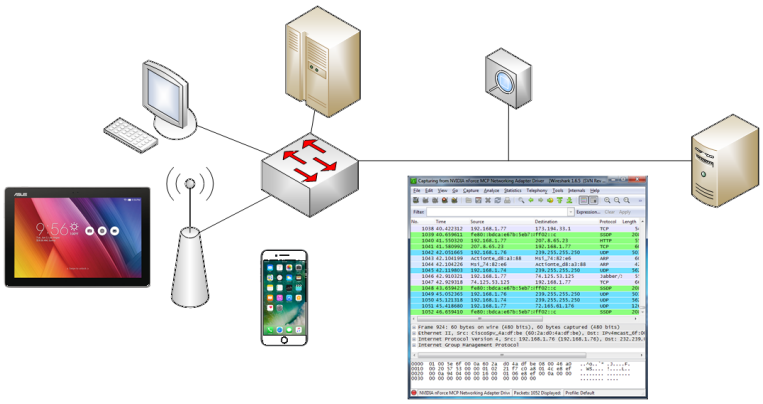
Obsah

1 Úvod

Tato technická zpráva se zabývá analýzou síťové komunikace mobilních zařízení (telefonů, tabletů, chytrých hodinek, apod.), viz obrázek 1, a možnostmi rozlišení těchto zařízení na základě síťové komunikace. Podobně jako u metod pro získávání otisku síťových zařízení vycházíme z dat získaných zachytávání síťové komunikace mobilního zařízení, tedy pasivním monitorováním.

Obrázek 1: Mobilní zařízení

Cílem je vytipovat chování komunikace mobilních zařízení, které by nám umožnilo odlišit tuto komunikaci od ostatní síťové komunikace. Další krokem je získání tzv. otisku zařízení (device fingerprint), který by sloužil k identifikaci daného mobilního zařízení z pohledu vyšetřování kybernetické kriminality, viz obrázek 2.



Obrázek 2: Detekce mobilního zařízení

Výzkum vychází z předpokladu, že mobilní komunikace má určité specifické rysy od komunikace klasických stolních či přenosných počítačů [7]. Tyto rysy se týkají jednak omezenější variabilitou komunikujících aplikací, četností zasílání paketů, i specifickými vlastnostmi týkajícími se nastavení operačního systému a komunikujícího TCP/IP stacku. Na základě těchto softwarových a hardwarových vlastností jsme schopni určit typickou charakteristiku zařízení, tzv. otisk.

2 Způsob získávání digitálních otisků

Pro detekci síťových zařízení se využívá systém získávání tzv. digitálních otisků zařízení (device fingerprint), kdy pomocí komunikaci s daným zařízení či analýzou zachyceného komunikace daného zařízení získáváme charakteristické vlastnosti zařízení. Tyto vlastnosti zvané *atributy* jednoznačně identifikují dané zařízení.

Atributy musí obsahovat hodnoty, které se v čase nemění nebo kde změna není příliš častá. Například IP adresa se mění v závislosti na typu připojení, na druhou stranu typ a verze operačního systému zařízení zůstávají neměnné až do aktualizace celého systému, což v oblasti mobilních zařízení není příliš obvyklé.

Sada hodnot atributů daného zařízení tvoří otisk zařízení. Různé systémy pro vytváření otisků využívají různé atributy a tím vytvářejí i různé otisky. Mezi atributy typické patří výrobce zařízení, model zařízení, operační systém a jeho verze, nainstalovaný software a další.

Otisky známých systémů a zařízení jsou uloženy v databázi otisků. V případě nového zařízení, získáme dostupné atributy a porovnáme je s databází známých otisků. Na základě celkové či částečné shody provedeme klasifikaci neznámého otisku.

V oblasti počítačových sítích jsou populární systémy pro detekci síťových zařízení a služeb, například nástroj Nmap[1] či p0f[2], které využívají atributy získané z hlaviček IP a TCP [10]. Tyto atributy mají různé hodnoty pro různé implementace operačních systémů zařízení či implicitní nastavení TCP/IP modulů v těchto zařízeních.

Další způsob je získávání otisků webového prohlížeče (browser fingerprinting)[4], získávání otisku DHCP serveru (dhcp fingerprinting) a další. Otisky webového prohlížeče se využívají k marketinkovým účelům, například pro sledování vracejících se zákazníků. Otisk operačních systémů a zařízení mají využití při autentizaci, kdy kromě hesla uživatele kontroluje přihlašovací portál typ zařízení, případně jeho nastavení (časovou zónu, jazykové nastavení), což lze využít k detekci zneužití odcizených přihlašovacích údajů.

Kromě obsahu paketů lze vytvářet i otisky založené na statistických datech, jako jsou například velikosti paketů, vzdálenosti mezi pakety, počet paketů, apod [3]. Tento přístup je ale vhodnější pro klasifikaci různých komunikačních protokolů než identifikaci zařízení.

Pro získávání atributů daného systému či zařízení používají systémy pro vytváření otisků buď aktivní přístup, kdy posílají speciální dotazy na zařízení (například systém Nmap, Am I Unique[3]) či pasivní přístup (například systém `p0f`). V následující části si popíšeme základní metody pro získávání otisků.

2.1 Aktivní získávání otisků nástrojem `Nmap`

Jedním z nejvíce rozšířených nástrojů pro získávání otisků síťových zařízení, skenování sítě či detekci operačného systému zařízení je systém Nmap [11]. Tento nástroj se používá zejména pro penetrační testování a bezpečností audit sítě.

Pro svou činnosti využívá Nmap aktivní monitorování. Na zkoumané zařízení posílá speciální IP pakety, které detekují, zda dané zařízení je připojené, jaké služby na něm běží (název a verze aplikací), jaký operační systém a verze je tam nainstalována, zda jsou některé porty blokovány, apod.

Pro získání otisku OS posílá Nmap 16 paketů TCP, UDP a ICMP an otevřené i zavřené porty zkoumaného zařízení. Na základě odpovědí vytváří Nmap otisk zařízení.

2.1.1 Použité atributy

Atributy získávané nástrojem Nmap[4]:

- Podpora a pořadí volitelných položek TCP (TCP options):
  - Konec seznamu voleb (EOL, End of Option List)
  - Prázdná operace (NOP, No operation)
  - Maximální velikost segmentu (MSS, Maximum Segment Size)
  - Měřítko pro zvětšení velikosti plovoucího okna (WS, Window Scale)

[1] Viz http://nmap.org [únor 2018].
[2] Viz http://lcamtuf.coredump.cx/p0f3/ [únor 2018]
[3] Viz http://amiunique.org [únor 2018]
[4] Viz https://nmap.org/book/man-os-detection.html [únor 2018]

  - Hodnota časového razítka (TSval, Timestamp value): průměrný přírůstek za sekundu u po sobě jdoucích paketů
  - Povolení selektivního potvrzování (SACK, Selective ACK)
- Největší společný dělitel (GCD) použitý při inicializaci sekvenčních čísel TCP (ISN, initial sequence number), rychlost čítače ISN (ISR, ISN counter rate), index pravděpodobnosti (SP, predictibility index), algoritmus generování ISN
- Počáteční velikost plovoucího okna (W, TCP initial window size)
- Nastavení zákazu fragmentaci u ICMP (DF, don't fragment)
- Počáteční hodnota TTL (TTL, IP initial time-to-live)
- Nastavení explicitního potvrzení při zahlcení (ECN, Explicit congestion notification)
- Testování sekvenčních čísel TCP a čísel potvrzení TCP
- Test přítomnosti následujících příznaků TCP v odpovědi: Echo (E), Urgent Data (URG), Acknowledgement (ACK), Push (PSH), Reset (RST), Synchronize (SYN), Final (FIN)
- Atributy pro testování IPv6
  - Rychlost čítače ISN (TCP_ISR, TCP ISN counter rate)
  - Velikost obsahu IPv6 (PLEN, Payload Length)
  - Třída provozu IPv6 (TC, Traffic Class)
  - Maximální počet skoků IPv6 (HLIM, Hop limit)
  - Velikost plovoucího okna TCP (Window size)
  - Nastavení příznaků TCP: F, S, R, P, A, U, E, C
  - Použití vyhrazených bitů v TCP (reserved bits)
  - Podpora voleb TCP (options), délka voleb TCP (options lengths)
  - Nastavení TCP voleb: MSS, SACK, WScale

2.1.2 Příklad digitálních otisků

Příklad otisku operačního systému FreeBSD[5]:

```
# FreeBSD 10.0-CURRENT FreeBSD 10.0-CURRENT #0 r251560: Sun Jun  9 19:24:47 EDT 2013 amd64
Fingerprint FreeBSD 10.0-CURRENT
Class FreeBSD | FreeBSD | 10.X | general purpose
CPE cpe:/o:freebsd:freebsd:10.0 auto
SEQ(SP=100-10A%GCD=1-6%ISR=101-10B%TI=RD%CI=RI%II=I%TS=21)
OPS(O1=M5B4NNSNW3NNT11%O2=M5B4NNSNW3NNT11%O3=M5B4NW3NNT11%O4=M5B4NNSNW3NNT11
%O5=M5B4NNSNW3NNT11%O6=M5B4NNSNNT11)
WIN(W1=4000%W2=4000%W3=4000%W4=4000%W5=4000%W6=4000)
ECN(R=Y%DF=Y%T=3C-46%TG=40%W=4000%O=M5B4NNSNW3%CC=N%Q=)
T1(R=Y%DF=Y%T=3C-46%TG=40%S=O%A=S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=N)
T4(R=Y%DF=Y%T=3C-46%TG=40%W=0%S=A%A=S%F=AR%O=%RD=0%Q=)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=3C-46%TG=40%W=0%S=A%A=S%F=AR%O=%RD=0%Q=)
T7(R=N)
U1(DF=N%T=3B-45%TG=40%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=S%T=3B-45%TG=40%CD=S)
```

[5] Níže uvedené příklady jsou z databáze Nmap uloženého v souboru `nmap-os-db.txt`, viz https://svn.nmap.org/nmap/nmap-os-db [únor 2017].

Příklad otisku operačního systému Android:

```
# HTC G1 (T-Mobile GSM/3G phone), Firmware 1.1, Android OS - Linux 2.6.25
# Android 2.1
Fingerprint Android 1.1 (Linux 2.6.25)
Class Google | Android | 1.X | phone
CPE cpe:/o:google:android:1.1 auto
Class Linux | Linux | 2.6.X | phone
CPE cpe:/o:linux:linux_kernel:2.6.25
CPE cpe:/o:google:android:1.1
SEQ(SP=C6-D0%GCD=1-6%ISR=CD-D9%TI=Z%CI=Z%II=I%TS=7|A)
OPS(O1=M5B4ST11NW0%O2=M5B4ST11NW0%O3=M5B4NNT11NW0%O4=M5B4ST11NW0%O5=M5B4ST11NW0)
WIN(W1=16A0%W2=16A0%W3=16A0%W4=16A0%W5=16A0%W6=16A0)
ECN(R=Y%DF=Y%T=41%TG=40%W=16D0%O=M5B4NNSNW0%CC=N%Q=)
T1(R=Y%DF=Y%T=41%TG=40%S=O%A=O|S+%F=AS%RD=0%Q=)
T2(R=N)
T3(R=Y%DF=Y%T=41%TG=40%W=16A0%S=O%A=O|S+%F=AS%O=M5B4ST11NW0%RD=0%Q=)
T4(R=N)
T5(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
T6(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
T7(R=Y%DF=Y%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)
U1(DF=N%T=3B-45%TG=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=3B-45%TG=40%CD=S)
```

Příklad otisku operačního systému Blackberry:

```
# Blackberry OS10.1.0.xxxx
Fingerprint BlackBerry 10
Class RIM | BlackBerry | 10.X | phone
CPE cpe:/o:blackberry:blackberry_os:10.0
SEQ(SP=EE-10C%GCD=1-6%ISR=FF-113%TI=I%CI=I%II=I%SS=S%TS=0|1|3)
OPS(O1=M5B4NW1NNT11SNN|M5B4NW2NNT11SNN|M5B4NW3NNT11SNN|M5B4NW4NNT11SNN|M5B4NW6NNT11SNN
%O2=M5B4NW1NNT11SNN|M5B4NW2NNT11SNN|M5B4NW3NNT11SNN|M5B4NW4NNT11SNN|M5B4NW6NNT11SNN
%O3=M5B4NW1NNT11|M5B4NW2NNT11|M5B4NW3NNT11|M5B4NW4NNT11|M5B4NW6NNT11%O4=M5B4NW1NNT11SNN
|M5B4NW2NNT11SNN|M5B4NW3NNT11SNN|M5B4NW4NNT11SNN|M5B4NW6NNT11SNN%O5=M5B4NW1NNT11SNN
|M5B4NW2NNT11SNN|M5B4NW3NNT11SNN|M5B4NW4NNT11SNN|M5B4NW6NNT11SNN%O6=M5B4NNT11SNN)
WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)
ECN(R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%O=M5B4NW1SNN|M5B4NW2SNN|M5B4NW3SNN|M5B4NW6SNN%CC=N)
T1(R=Y%DF=Y%T=3B-45%TG=40%S=O%A=S+%F=AS%RD=0)
T2(R=Y%DF=N%T=3B-45%TG=40%W=0%S=Z%A=S%F=AR%RD=0)
T3(R=Y%DF=Y%T=3B-45%TG=40%W=FFFF%S=O%A=S+%F=AS%O=M5B4NW1NNT11SNN|M5B4NW2NNT11SNN
|M5B4NW3NNT11SNN|M5B4NW4NNT11SNN|M5B4NW6NNT11SNN%RD=0)
T4(R=Y%DF=N%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%RD=0)
T5(R=Y%DF=N%T=3B-45%TG=40%W=0%S=Z%A=S+%F=AR%RD=0)
T6(R=Y%DF=N%T=3B-45%TG=40%W=0%S=A%A=Z%F=R%RD=0)
T7(R=Y%DF=N%T=3B-45%TG=40%W=0%S=Z%A=S%F=AR%RD=0)
U1(DF=N%T=FA-104%TG=FF%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)
IE(DFI=N%T=FA-104%TG=FF%CD=S)
```

Digitální podpis Nmap obsahuje části týkající popisu zařízení, tj. `Fingerprint`, `Class`, `Common Platform Enumeration (CPE)` , a dále seznam výsledků testů, které se využívají pro porovnání. U některých testů může být pro dané zařízení více možností, což označuje operátor "|".

Položka `Fingerprint` slouží jako identifikátor podpisu. Položka `Class` obsahuje výrobce, rodinu operačních systémů, verzi OS a typ zařízení. Tyto položky jsou také odděleny symbolem "|". Pokud je stejný podpis pro více zařízení, obsahuje podpis více položek `Class` s označením jednotlivých zařízení. Aktuální verze databáze podpisů Nmap obsahuje 1359 různých tříd zařízení a 5654 podpisů.

Druhou část podpisu tvoří výsledky skenování daného zařízení. Pokud u některých zařízení dává výsledek testu různé hodnoty, jsou tyto hodnoty umístěny v podpisu. Nmap používá operátory

nebo ("|"), *interval* ("–") či porovnání numerických hodnot "<"či ">". Například hodnota pro největší společný dělitel GCD může být vyjádřena výrazem `GCD=1-6|64|256|>1024`.

```
S1(P=6000{4}28063cXX{32}0016c1b002bbd213c57562f5a01212e0f8880000020404c4040208
    0a5be177f2ff{4}01030307%ST=0.021271%RT=0.041661)
EXTRA(FL=12345)
```

Otisk obsahuje pole P označuje hodnotu z hlavičky paketu (hodnota XX nahrazují zdrojovou a cílovou adresu, číslo v složených závorkách označuje opakující se počet bytů obsahující danou hodnotu). Každá hodnota pole P reprezentuje nějakou hodnotu atributu (viz seznam atributů IPv6). Pole ST označuje relativní čas odeslání testovacímu paketu, pole RT označuje čas přijetí odpovědi na testovací paket.

Databáze otisků IPv6 má jiný formát než IPv4 a je uložena ve zdrojovém kódu Nmap v souboru FPModel.cc jako množina vektorů. Následující příklad reprezentuje část otisku OS FreeBSD:

Zvláštní formát má otisk IPv6:

```
/* FreeBSD 7.4 - 8.2 */
{ +0.83333333, +0.02430556, +0.00000000, +0.83333333, +0.01736111, +0.00000000, +1.00000000,
  +0.02430556, +0.00000000, +0.83333333, +0.01909722, +0.00000000, +0.83333333, +0.00347222,
  +0.00000000, +0.80000000, +0.01215278, +0.00000000, -1.00000000, -1.00000000, -1.00000000,
  +0.50000000, +0.00000000, +0.00000000, +0.00000000, +0.00000000, +0.00000000, +0.
```

Digitální otisk zařízení je možné získat pomocí příkazu `nmap -O -sV -T4 -d <target>`:

```
TCP/IP fingerprint:
OS:SCAN(V=7.40%E=4%D=2/27%OT=22%CT=1%CU=41734%PV=N%DS=2%DC=I%G=Y%TM=5A953F6
OS:C%P=amd64-portbld-freebsd11.1)SEQ(SP=107%GCD=1%ISR=10C%CI=I%TS=21)SEQ(SP
OS:=103%GCD=1%ISR=107%TI=RI%TS=21)SEQ(SP=102%GCD=1%ISR=10A%TS=21)SEQ(SP=104
OS:%GCD=1%ISR=10B%TI=RI%CI=I%TS=21)OPS(O1=M5B4NW4ST11%O2=M5B4NW4ST11%O3=M5B
OS:4NW4NNT11%O4=M5B4NW4ST11%O5=M5B4NW4ST11%O6=M5B4ST11)WIN(W1=FFFF%W2=FFFF%
OS:W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%DF=Y%T=40%W=FFFF%O=M5B4NW4SLL%CC
OS:=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=40%W=F
OS:FFF%S=O%A=S+%F=AS%O=M5B4NW4ST11%RD=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R
OS:%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=
OS:40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%
OS:Q=)U1(R=Y%DF=N%T=40%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y
OS:%DFI=S%T=40%CD=S)
```

2.1.3 Klasifikace zařízení

Algoritmus pro klasifikaci otisku u nástroje Nmap porovnává soubor získaných atributů neznámého zařízení (tzn. jeho otisk) s každým otiskem v databázi známých otisků `nmap-os-db`. Pro každý test, který je uveden u daného otisku, kontroluje Nmap každou hodnotu. Pokud najde shodu, inkrementuje čítač `PossiblePoints` o hodnotu, která je přiřazena danému testu. Tato váha určuje důležitost atributu v podpisu. Hodnoty vah jsou uloženy na začátku souboru `nmap-os-db`:

```
SEQ(SP=25%GCD=75%ISR=25%TI=100%CI=50%II=100%SS=80%TS=100)
OPS(O1=20%O2=20%O3=20%O4=20%O5=20%O6=20)
WIN(W1=15%W2=15%W3=15%W4=15%W5=15%W6=15)
ECN(R=100%DF=20%T=15%TG=15%W=15%O=15%CC=100%Q=20)
T1(R=100%DF=20%T=15%TG=15%S=20%A=20%F=30%RD=20%Q=20)
T2(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T3(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T4(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T5(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T6(R=100%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
T7(R=80%DF=20%T=15%TG=15%W=25%S=20%A=20%F=30%O=10%RD=20%Q=20)
U1(R=50%DF=20%T=15%TG=15%IPL=100%UN=100%RIPL=100%RID=100%RIPCK=100%RUCK=100%RUD=100)
IE(R=50%DFI=40%T=15%TG=15%CD=100)
```

Například váha hodnoty `Window size (W)` v testu ECN je 15, zatímco u testu T1 je její váha 25.

Během porovnávání se kontroluje hodnoty atributu s ohledem na jeho typ. Pokud je u atributu zadán typ řetězec či číselná hodnota, kontroluje se přesná shoda. Pokud je možných více hodnot pomocí operátorů |, <, >, musí testovaná hodnota být v rozsahu daných hodnot.

Poté, co proběhla kontrolovala všech řádků podpisu (tj. různých testů), spočítá Nmap podíl hodnot `NumMatchPoints` a `PossiblePoints`. Výsledek dává poměr vyjadřující pravděpodobnost, že testovaný otisk odpovídá danému otisku v databázi známých otisků. Hodnota `1.00` odpovídá přesné shodě.

Pokud nedošlo k přesné shodě, vypíše Nmap nejbližší nalezenou shodu, viz následující příklad:

```
Device type: general purpose|storage-misc
Running (JUST GUESSING): Linux 3.X|2.6.X (89%), Synology DiskStation Manager 5.X (87%)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:2.6 cpe:/a:synology:
diskstation_manager:5.2 Aggressive OS guesses: Linux 3.2 - 3.10 (89%),Linux 3.2-3.16 (89%),
%Linux 3.2 - 3.8 (89%), Linux 3.4%(89%), Linux 3.4 - 3.10 (89%), Linux 2.6.32 - 3.10 (88%),
Linux 2.6.32 - 3.13 %(88%), Linux 3.7 (88%), Linux 3.10 (87%),Synology DiskStation Manager
5.2-5644 (87%)
No exact OS matches for host (test conditions non-ideal).
```

Pro klasifikaci OS pomocí IPv6 využívá Nmap techniku strojového učení zvanou *logická regrese* za pomoci knihovny LIBLINER[6]. Klasifikaci předchází natrénování klasifikátoru pomocí anotovaných dat, které jsou tvořeny vektory atributů známých systémů. Vektor reprezentuje souřadnici OS v multidimenzionálním prostoru. Trénovací algoritmus pak spočítá optimální hranice mezi zástupci tříd jednotlivých OS. Tato hranice je opět reprezentována vektorem.

Při porovnávání se počítá kartézský součin mezi každým z hraničních vektorů a daným vektorem atributů. Výsledkem je reálné číslo, které vyjadřuje pravděpodobnost shody (kladná hodnota) či neshody (záporná hodnota) vůči porovnávanému otisku. Pro mapování tohoto čísla do intervalu $(0, 100)$ se používá funkce logické regrese, tj. $f(x) = \frac{100}{1+e^x}$.

Třída OS s nejvyšší hodnotou vyjadřuje nejpravděpodobnější shodu. Pokud se ale jedná o neznámý OS, pak i zde může být shoda vysoká. Z tohoto důvodu se používá algoritmus detekce novosti, který počítá euklidovskou vzdálenost vektoru zkoumaného otisku vůči střední hodnotě vektorů dané třídy OS. Vektory otisků podobných zkoumanému budou mít malou hodnotu novosti, zatímce vektory odlišných otisků budou mít vysokou hodnotu novosti. Pokud hodnota novosti otisku je nižší než 15, pak se použije vypočtená hodnota logické regrese.

Následující příklad ukazuje výsledek klasifikace Mac OS X 10.6.8 pro IPv6:

```
Score Novelty OS class
61.05% 1.00 Apple Mac OS X 10.6.8 - 10.7.0 (Snow Leopard - Lion) (Darwin 10.8.0 - 11.0.0)
10.08% 18.04 Apple Mac OS X 10.7 (Lion) (Darwin 11.1.0)
9.97% 24.06 Apple Mac OS X 10.6.8 (Snow Leopard) (Darwin 10.8.0)
9.43% 19.26 Apple Mac OS X 10.7.2 (Lion) (Darwin 11.2.0)
5.99% 23.63 Apple Mac OS X 10.4.11 (Tiger) (Darwin 8.11.1)
```

2.2 Pasivní získávání otisků nástrojem `p0f`

Nástroj p0f[7] slouží k pasivnímu získávání otisků zařízení z komunikace TCP/IP. Používá se k detekci zařízení za NATem, vyvažování zátěže, detekci klientů a serverů s podvrženými hodnotami `X-Mailer` a `User-Agent`. Pro svou činnost využívá informace přenášené v hlavičkách protokolů IPv4, IPv6, TCP a dalších.

Pro analýzu využívá pakety SYN a SYN-ACK, kde sleduje pořadí voleb TCP, vztah mezi hodnotami MSS a WSS, nárůst hodnot TCP timestamp a další vlastnosti.

U aplikačních protokolů využívá pořadí a syntax hlaviček u HTTP či SMTP spíše než hodnotu `User-Agent`.

[6] Viz https://www.csie.ntu.edu.tw/~cjlin/liblinear/ [únor 2018]
[7] Viz http://lcamtuf.coredump.cx/p0f3/ [březen 2018]

2.2.1 Atributy u komunikace TCP/IP

- Počáteční hodnota TTL (ittl, initial TTL)
- Délka voleb IPv4 či rozšiřujících hlaviček IPv6 (olen)
- Hodnota Maximum segment size (MSS)
- Hodnota Windows size (wsize)
- Hodnota Windows scaling factor (scale)
- Seznam voleb TCP včetně pořadí (olayout)
  - Explicitní konec voleb TCP s n byty výplně (eol+n)
  - Povolení volby No operation (nop)
  - Volba Maximum segment size (mss)
  - Volba Windows scaling (ws)
  - Volba Selective ACK permitted (sok)
  - Volba Selective ACK (sack)
  - Timestamp (ts)
  - Neznámá volba n (?n)
- Zvláštní nastavení (quirks) v hlavičkách TCP či IP
  - Nastavená volba Don't fragment (df)
  - Nastavený bit DF ale IPID je nenulový (id+)
  - Bit DF není nastavený, ale IPID je nulový (id-)
  - Podpora ECN (ecn)
  - Nenulová hodnota IPv6 flow ID (flow)
  - Sekvenční číslo je nula (seq-)
  - Hodnota čísla ACK je nenulová, ale příznak ACK není nastaven (ack+)
  - Hodnota čísla ACK je nulová, ale příznak ACK je nastaven (ack-)
  - Ukazatel URG pointer je nenulový, ale příznak URG není nastaven (uptr+)
  - Příznak URG je nastaven (urgf+)
  - Příznak PUSH je nastaven (pushf+)
  - Vlastní časové razítko je nastaveno na nulu (ts1-)
  - Časové razítko obashuje nenulovou hodnotu u druhé strany v paketu SYN (ts2+)
  - Ukončující sekvence segmentu voleb je nenulová (opt+)
  - Násobek Windows scaling je větší než 14 (exws)
  - Špatně formátové volby TCP (bad)
- Obsah paketu (pclass) — SYN paket normálně neobsahuje žádný obsah, ale jsou i výjimky.

Formát otisku TCP/IP u p0f je `{ver:ittl:olen:mss:wsize:scale:olayout:quirks: pclass}`, kde `ver` označuje, že otisk se vztahuje k protokolu IPv4 (4), IPv6 (6) či oba (*). Následující výpis obsahuje ukázky otisků TCP/IP z databáze `p0f.fp`, která je součástí distribuce nástroje p0f[8]. Otisky získané z dotazů TCP SYN (request):

[8] Viz http://lcamtuf.coredump.cx/p0f3/releases/p0f-3.09b.tgz [březen 2018]

```
[tcp:request]
label = s:unix:Linux:3.1-3.10
sig   = *:64:0:*:mss*10,4:mss,sok,ts,nop,ws:df,id+:0
sig   = *:64:0:*:mss*10,5:mss,sok,ts,nop,ws:df,id+:0
sig   = *:64:0:*:mss*10,6:mss,sok,ts,nop,ws:df,id+:0
sig   = *:64:0:*:mss*10,7:mss,sok,ts,nop,ws:df,id+:0

label = s:win:Windows:7 or 8
sig   = *:128:0:*:8192,0:mss,nop,nop,sok:df,id+:0
sig   = *:128:0:*:8192,2:mss,nop,ws,nop,nop,sok:df,id+:0
sig   = *:128:0:*:8192,8:mss,nop,ws,nop,nop,sok:df,id+:0
sig   = *:128:0:*:8192,2:mss,nop,ws,sok,ts:df,id+:0

label = s:unix:MacOS X:10.9 or newer (sometimes iPhone or iPad)
sig   = *:64:0:*:65535,4:mss,nop,ws,nop,nop,ts,sok,eol+1:df,id+:0

label = s:unix:iOS:iPhone or iPad
sig   = *:64:0:*:65535,2:mss,nop,ws,nop,nop,ts,sok,eol+1:df,id+:0

label = s:unix:FreeBSD:9.x or newer
sig   = *:64:0:*:65535,6:mss,nop,ws,sok,ts:df,id+:0

label = s:other:Blackberry:
sig   = *:128:0:1452:65535,0:mss,nop,nop,sok,nop,nop,ts::0

label = s:other:Nintendo:3DS
sig   = *:64:0:1360:32768,0:mss,nop,nop,sok:df,id+:0
```

Otisky získané z odpovědí TCP SYN+ACK (response):

```
[tcp:response]
label = s:unix:Linux:3.x
sig   = *:64:0:*:mss*10,0:mss:df:0
sig   = *:64:0:*:mss*10,0:mss,sok,ts:df:0
sig   = *:64:0:*:mss*10,0:mss,nop,nop,ts:df:0
sig   = *:64:0:*:mss*10,0:mss,nop,nop,sok:df:0
sig   = *:64:0:*:mss*10,*:mss,nop,ws:df:0
sig   = *:64:0:*:mss*10,*:mss,sok,ts,nop,ws:df:0
sig   = *:64:0:*:mss*10,*:mss,nop,nop,ts,nop,ws:df:0
sig   = *:64:0:*:mss*10,*:mss,nop,nop,sok,nop,ws:df:0

label = s:win:Windows:7 or 8
sig   = *:128:0:*:8192,0:mss:df,id+:0
sig   = *:128:0:*:8192,0:mss,sok,ts:df,id+:0
sig   = *:128:0:*:8192,8:mss,nop,ws:df,id+:0
sig   = *:128:0:*:8192,0:mss,nop,nop,ts:df,id+:0
sig   = *:128:0:*:8192,0:mss,nop,nop,sok:df,id+:0
sig   = *:128:0:*:8192,8:mss,nop,ws,sok,ts:df,id+:0
sig   = *:128:0:*:8192,8:mss,nop,ws,nop,nop,ts:df,id+:0
sig   = *:128:0:*:8192,8:mss,nop,ws,nop,nop,sok:df,id+:0

label = s:unix:Mac OS X:10.x
sig   = *:64:0:*:65535,0:mss,nop,ws:df,id+:0
sig   = *:64:0:*:65535,0:mss,sok,eol+1:df,id+:0
sig   = *:64:0:*:65535,0:mss,nop,nop,ts:df,id+:0
sig   = *:64:0:*:65535,0:mss,nop,ws,sok,eol+1:df,id+:0
sig   = *:64:0:*:65535,0:mss,nop,ws,nop,nop,ts:df,id+:0
sig   = *:64:0:*:65535,0:mss,nop,nop,ts,sok,eol+1:df,id+:0
sig   = *:64:0:*:65535,0:mss,nop,ws,nop,nop,ts,sok,eol+1:df,id+:0
```

2.2.2 Otisk MTU

Nástroj p0f umožňuje také získat otisk MTU. Mnohé operační systémy totiž odvozují hodnotu Maximum segment size (MSS) v TCP od nastavení MTU na síťové kartě, což přináší vazbu na linkový protokol použitý u daného zařízení. Hodnota MTU se liší například u PPPoE, IPSec či VPN sítí. Databáze p0f proto obsahuje i otisky systémů podle hodnot MTU.

Otisk MTU obsahuje dvě položky: `label`, která označuje typ podpisu, a `sig`, což je vlastní otisk (signatura). Příklad MTU otisků u p0f je v následující ukázce:

```
[mtu]
label = Ethernet or modem
sig   = 576
sig   = 1500

label = DSL
sig   = 1452
sig   = 1454
sig   = 1492

label = IPSec or GRE
sig   = 1476

label = IPIP or SIT
sig   = 1480

label = PPTP
sig   = 1490
```

2.2.3 Otisk komunikace HTTP

HTTP otisky získává p0f z hlaviček HTTP dotazů GET a HEAD. Využívá tyto atributy:

- Verze protokolu (ver)
- Seznam hlaviček HTTP (horder)
- Hodnota User-Agent nebo Server (expsw)

Příklad otisků získaných z provozu HTTP u nástroje p0f:

```
[http:request]
label = s:!:Firefox:2.x
sys   = Windows,@unix
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip,deflate],
Accept-Charset=[utf-8;q=0.7,*;q=0.7],Keep-Alive=[300],Connection=[keep-alive]::Firefox/

label = s:!:Firefox:5.x-9.x
sys   = Windows,@unix
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip, deflate],
Accept-Charset=[utf-8;q=0.7,*;q=0.7],?DNT=[1],Connection=[keep-alive],?Referer:Keep-Alive:
Firefox/
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip,  deflate],
Accept-Charset=[UTF-8,*],?DNT=[1],Connection=[keep-alive],?Referer:Keep-Alive:Firefox/
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip, deflate],
Accept-Charset=[UTF-8,*],?DNT=[1],?Referer,Connection=[keep-alive]:Keep-Alive:Firefox/
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip, deflate],
Accept-Charset=[utf-8;q=0.7,*;q=0.7],?DNT=[1],?Referer,Connection=[keep-alive]:Keep-Alive:
Firefox/
sig   = *:Host,User-Agent,Accept=[,*/*;q=],?Accept-Language,Accept-Encoding=[gzip, deflate],
Accept-Charset=[utf-8;q=0.7,*;q=0.7],?Referer,?DNT=[1],Connection=[keep-alive]:Keep-Alive:
```

```
Firefox/

label = s:!:MSIE:7
sys   = Windows
sig   = 1:Accept=[*/*],?Referer,?Accept-Language,UA-CPU,User-Agent,Accept-Encoding=[gzip,
  deflate], Host,Connection=[Keep-Alive]:Keep-Alive,Accept-Charset:(compatible; MSIE

label = s:!:Chrome:51.x or newer
sys   = Windows,@unix
sig   = 1:Host,Connection=[keep-alive],Upgrade-Insecure-Requests=[1],User-Agent,Accept=[*/*],
Accept-Encoding=[gzip, deflate, sdch],Accept-Language:Accept-Charset,Keep-Alive: Chrom

label = s:!:Opera:19.x or newer
sys   = Windows,@unix
sig   = 1:Host,Connection=[keep-alive],Accept=[*/*;q=0.8],User-Agent,
Accept-Encoding=[gzip,deflate,lzma],Accept-Language=[;q=0.]:Accept-Charset,Keep-Alive:OPR/

label = s:!:Android:2.x
sys   = Linux
sig   = 1:Host,Accept-Encoding=[gzip],Accept-Language,User-Agent,Accept=[,*/*;q=0.5],
Accept-Charset=[utf-16, *;q=0.7]:Connection:Android
sig   = 1:Host,Connection=[keep-alive],Accept-Encoding=[gzip],Accept-Language,User-Agent,
Accept=[,*/*;q=0.5],Accept-Charset=[utf-16, *;q=0.7]::Android
sig   = 1:Host,Accept-Encoding=[gzip],Accept-Language=[en-US],Accept=[*/*;q=0.5],User-Agent,
Accept-Charset=[utf-16, *;q=0.7]:Connection:Android

label = s:!:Safari:7 or newer
sys   = @unix
sig   = *:Host,Accept-Encoding=[gzip, deflate],Connection=[keep-alive],Accept=[*/*],
User-Agent,Accept-Language,?Referer,?DNT:Accept-Charset,Keep-Alive:KHTML, like Gecko)

label = s:!:Konqueror:4.7 or newer
sys   = Linux,FreeBSD,OpenBSD
sig   = 1:Host,Connection=[keep-alive],User-Agent,Accept=[*/*],Accept-Encoding=[gzip,
deflate, x-gzip, x-deflate],Accept-Charset=[,*;q=0.5],Accept-Language::Konqueror/

label = s:!:wget:
sys   = @unix,Windows
sig   = *:User-Agent,Accept=[*/*],Host,Connection=[Keep-Alive]:Accept-Encoding,
Accept-Language, Accept-Charset:Wget/

label = s:!:curl:
sys   = @unix,Windows
sig   = 1:User-Agent,Host,Accept=[*/*]:Connection,Accept-Encoding,Accept-Language,
Accept-Charset:curl/
```

2.3 Otisk webového prohlížeče (browser fingerprint)

Získávání otisků webového prohlížeče je jeden z nejběžnějších způsobů, který slouží pro identifikaci uživatele při přihlašování na různé servery. Používá se zejména pro marketinkové účely (cílená reklama pro vracející se uživatele), ale také pro kontrolu, zda nedošlo k podvržení přihlašovacích údajů.

Většina nástrojů pro získávání otisků prohlížeče využívá aktivního přístupu, který spočívá v tom, že uživatel si načte stránku, která obsahuje javacriptový či flashový kód, který se dotáže na nastavení lokálního prohlížeče a toto nastavení uložení na serveru do databáze otisků.

2.3.1 Nástroj Panopticlick

Příkladem nástroje pro získávání webového otisku je Panopticlick[9], který zjišťuje jaké informace o sobě sdělí webový prohlížeč. Tento nástroje využívá aktivní přístup pro získání informací o připojeném prohlížeči[4], viz tabulka 1.

Proměnná	**Hodnota**
User Agent	Přenáší HTTP, loguje server
Hlavička HTTP ACCEPT	Přenáší HTTP, loguje server
Podpora Cookies	Vloženo do HTTP, loguje server
Rozlišení obrazovky	JavaScript AJAX Post
Časová zóna	JavaScript AJAX Post
Pluginy prohlížeče, verze pluginů, typ MIME	JavaScript AJAX Post
Systémové fonty	Flash, Java Applet, sebráno pomocí JavaScript/AJAX
Test supercookkie	JavaScript AJAX Post

Tabulka 1: Získávání otisku prohlížeče nástrojem Panopticlick[4]

Příklad otisku je v tabulce 2. Pro hledání jedinečnosti otisku vypočítává Panopticlick entropii prohlížeče za pomoci hustot pravděpodobnosti jednotlivých atributů:

$$H(F) = -\sum_{n=0}^{N} P(f_n)\log_2(P(f_n))$$

Charakteristika	**Hodnota**
Limited supercookie test	DOM localStorage: Yes, DOM sessionStorage: Yes, IE userData: No
Hash of canvas fingerprint	f8057d0a18ceaf1b0d699e144ddefdb0
Screen Size and Color Depth	1920x1200x24
Browser Plugin Details	Plugin 0: Shockwave Flash; Shockwave Flash 27.0 r0; npwrapper.libflashplayer.so; (Shockwave Flash; application/x-shockwave-flash; swf) (FutureSplash Player; application/futuresplash; spl).
Time Zone	-60
DNT Header Enabled?	False
HTTP_ACCEPT Headers	text/html, */*; q=0.01 gzip, deflate, br cs,en-US;q=0.7,en;q=0.3
Hash of WebGL fingerprint	0d616c55ad81d25267033d5d1854eed7
Language	cs
System Fonts	Andale Mono, Arial, Arial Black, Bitstream Vera Sans Mono, Calibri, Cambria, Comic Sans MS, Courier, Courier New, Georgia, Helvetica, Impact, Times, Times New Roman, Trebuchet MS, Verdana, Wingdings 2, Wingdings 3 (via javascript)
Platform	FreeBSD amd64
User Agent	Mozilla/5.0 (X11; FreeBSD amd64; rv:56.0) Gecko/20100101 Firefox/56.0
Touch Support	Max touchpoints: 0; TouchEvent supported: false; onTouchStart supported: false
Are Cookies Enabled?	Yes

Tabulka 2: Příklad otisku prohlížeče.

Entropie udává míru neurčitosti daného jevu, v tomto případě míru odlišnosti prohlížečů na základě atributů. Entropie jednotlivých položek otisku Panopticlick na základě databáze otisků je v tabulce 3. Čím vyšší je hodnota entropie atributu, tím užitečnější je tato hodnota pro vytvoření jedinečného otisku prohlížeče.

[9] Viz https://panopticlick.eff.org/about [březen 2018].

Proměnná	Entropie (v bitech)
User Agent	10.0
Plugins	15.4
Fonts	13.9
Video	4.83
Supercookies	2.12
HTTP_ACCEPT	6.09
Timezone	3.04
Cookies_enabled	0.353

Tabulka 3: Entropie hodnot jednotlivých atributů

2.3.2 Experiment LetMeTrackYou

Další výzkum otisků webových prohlížečů probíhal v roce 2011 na portále `LetMeTrackYou.org`, kde zkoumali otisky cca 1200 uživatelů [2]. Podobně jako Panopticlick počítali za pomoci entropie množství informace, které lze z jednotlivých atributů otisku.

Výzkumníci využívali data získaná z hlavičky HTTP (pasivní přístup) a dále data získaná za pomoci JavaScriptu (aktivní přístup). Přehled atributů a hodnot entropie je možné vidět v tab. 4.

Zdroj	Položka	Entropie
HTTP	Accept	1.73
HTTP	Accept-Language	4.16
HTTP	Accept-Encoding	1.67
HTTP	Accept-Charset	1.86
HTTP	Connection	0.28
HTTP	User-Agent	6.01
HTTP	DNT	0.51
JavaScript	Enabled	0.87
JavaScript	Version	2.03
JavaScript	Platform	2.19
JavaScript	Charset	1.6
JavaScript	Language	2.23
JavaScript	Cookie support	0.87
JavaScript	Java support	0.98
JavaScript	Timezone	1.57
JavaScript	Plugin versions	5.54
JavaScript	Screen resolution	4.30
JavaScript	Font list	5.55

Tabulka 4: Entropie jednotlivých atributů [2]

Některé charakteristiky prohlížečů se příliš nemění (například název prohlížeče, výrobce, platforma), jiné se mění během upgradů či instalace nových pluginů (například verze prohlížeče, instalované fonty, seznam pluginů). Z tohoto důvodu je potřeba otisk aktualizovat.

2.4 Otisk DNS komunikace

Další možností, jak získat otisk komunikace, je analýza provozu DNS [12]. Tento přístup vychází z předpokladu, že každý operační systém má charakteristické chování při dotazování se DNS serverů, které zahrnuje dotazování se na konkrétní doménová jména týkající se operačního systému (např. pro kontrolu aktualizací), charakteristické druhy dotazů (dotaz na A či AAAA, překlad PTR, apod.), rozložení zasílání dotazů v čase a další.

Kromě dotazů týkající se operačního systému, např. domén `android.clients. google.com`, `micloud.xiaomi.net`, můžeme sledovat i dotazy na domény týkající se používaných aplikací, například `firefox.com`, `facebook.com`, `office365.com` a další.

Při analýze DNS dotazů lze detekovat i testovací body připojení (tzv. captive portals), což jsou obvykle webové stránky, na kterých operační systém či aplikace testují dostupnost a rychlost internetového připojení. Používají se zejména pro mobilní či bezdrátové (WiFi) připojení. Často tyto webové stránky obsahují informace o spojení, autentizaci uživatelů a další. Používají se také pro marketinkové účely. Příkladem domén využívající testovací body připojení jsou například `connectivitycheck.gstatic.com`, `mtalk.google.com` a další.

2.4.1 Chování DNS

- Mobilní operační systémy, např. Android, používá specifické domény, na které se pravidelně dotazují.
- Na určité domény posílá Android OS nejprve dotaz typu AAAA, poté dotaz typu A. Výsledek pak otestuje dotazem PTR.
- Některé operační systémy se pravidelně dotazují na určité domény. Podle frekvence dotazů lze detekovat konkrétní operační systém.

Hodnotu Captive-Portal lze klientovi přidělit o pomocí volby DHCP číslo 160 [8].

2.5 Otisk DHCP komunikace (DHCP fingerprint)

Otisk DHCP vychází z předpokladu, že každá implementace DHCP klienta se mírně liší v nastaveních, které se objevují v hlavičce paketů. Jedná se zejména o rozšířené volby protokolu DHCP, které jsou popsány ve standarech RFC 2132 [1], RFC 4361 [9] či RFC 4578 [6]. Cílem těchto voleb je jednoznačná identifikace bootujícího klienta a jejich operačního systému, aby DHCP server či PXE (Preboot Execution Environment) boot server mohl poslat klientovi správné jméno zavaděče operačního systému (bootstrap image) a také jméno serveru, kde je příslušný zavaděč uložen. Tyto a další informace lze dále využít obecně pro vytvoření otisku klienta. Seznam všech voleb protokolu DHCP lze najít v databázi IANA[10].

Pro vytvoření otisku se využívají zejména informace přenášené v paketech Discover a Request, které posílá klient. V těchto zprávách v části Options je možné najít jméno operačního systému, název zařízení, jméno výrobce a další hodnoty. Tyto informace lze využít k vytvoření otisku zařízení. Otisk lze použít pro identifikaci daného zařízení např. při monitorování zařízení, která žádají o přístup do sítě. DHCP otisky využívají například DHCP servery, WiFi kontrolery a další síťová zařízení.

Otisk DHCP nám umožňuje rozlišit zejména typ zařízení, například zda se jedná o mobilní zařízení, tablet, stolní počítač, server, směrovač, přepínač, herní konzoly, VoIP zařízení, tiskárnu či jiné zařízení. Pro identifikaci musíme vybrat sadu atributů k identifikaci a dále databázi otisků známých systémů a zařízení.

2.5.1 Použité atributy

Pro vytváření DHCP otisku se používají následující atributy:

- IP TTL v paketu DHCP: možnosti 16 (Linux 1), 32 (MS Win 95), 64 (Linux 2), 128 (MS Win $>$ 95), 255 (Mac OS X)
- Nastavení DHCP Options
  - Volba 12: Hostname
  - Volba 55: Parameter List (requested-parameters)

[10] Viz https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml [únor 2018]

  - Volba 60: Class ID (vendor-id)
  - Volba 61: Client ID
  - Volba 77: User-Class Information
  - Volba 81: Client FQDN
  - Volba 82: Relay Agent Information
  - Volba 93: Client System Architecture (PXE boot)
- Počet a pořadí voleb v parametru 55 u daného klienta.

Volby jsou ve zprávě DHCP uloženy ve formát `[Option][Length][Value]`, kde `Option` je jednobytový identifikátor volby, `Length` je celková délka hodnoty volby v bytech, a `Value` je vlastní hodnota.

Otisk dané volby je obvykle uložen ve formátu `[Option][Value]`. Otisk DHCP zprávy je vytvořen ze sekvence otisků jednotlivých voleb, které daný nástroj pro DHCP fingerprinting používá. Jednotlivé části otisku neznámého zařízení se při klasifikaci porovnají s databází známých otisků. Klasifikace využívá porovnání na přesnou shodu hodnot, hledání podřetězce a další podle typu atributu.

Při klasifikaci se vychází z předpokladu, že implementace různých DHCP klientů používají různé volby a různé pořadí voleb, které je typické pro dané zařízení či operační systém. Na základě toho lze klasifikovat neznámé zařízení a přiřadit ho do určité skupiny podle výrobce, typu operačního systému a dalších parametrů.

2.5.2 Příklad otisků DHCP

Otisky DHCP mohou mít různý formát podle druhu implemementace. Například nástroj ArubaOS DHCP Fingerprinting používá k vytváření otisků volby 12, 55, 60 a 81 [13]. Zařízení Infoblox používá pouze dvě volby: 55 (requested-parameters) a 60 (vendor-id)[11].

Pro ukládání otisků používá ArubaOS formát `[Typ][Hodnota]` v hexadecimálním tvaru tak, jak je to uloženo v paketu DHCP. Například volba 55 (Parameter List) obsahující seznam parametrů 1 (subnet mask), 3 (router), 6 (domain server), 15 (domain name), 119 (domain search) a 252 (proxy/autodiscovery) bude mít formát 0x370103060F77FC, kde 0x37 je hexadecimální zápis volby 55.

Příklad otisků využívaných v ArubaOS je v tabulce 5, další lze najít v příloze č.1.

Zařízení	**DHCP volba**	**DHCP otisk (hexa)**
Apple iOS	Option 55	370103060F77FC
Android	Option 60	3C64686370636420342E302E3135
Blackberry	Option 60	3C426C61636B4265727279
Windows 7 Vista	Option 55	37010f03062c2e2f1f2179f92b
Windows XP	Option 55	37010f03062c2e2f1f21f92b
Windows Mobile	Option 60	3c4d6963726f736f66742057696e646f777 320434500
Windows 7 Phone	Option 55	370103060f2c2e2f
Apple Mac OS X do verze 10.6	Option 55	370103060f775ffc2c2e2f
Apple Mac OS X od verze 10.7	Option 55	370103060f775ffc2c2e

Tabulka 5: Příklad ověřených otisků DHCP Aruba [13]

Výhodou DHCP komunikace je, že se šíří broadcastem, proto je jednoduché ji zachytávat a analyzovat. V případě rozsáhlejších sítí je možné využít DHCP relay pro směrování DHCP dotazů na jeden server, kde může běžet i sonda pro vytváření otisků.

Nevýhodou pro získávání DHCP otisků je, že potřebujeme přístup k lokální síti, kde DHCP provoz probíhá. Pokud monitorujeme komunikaci mimo lokální síť, je tato metoda nepoužitelná.

Známé databáze DHCP otisků jsou PacketFence a Satori.

[11] Viz https://docs.infoblox.com/display/NAG8/Chapter+38+DHCP+Fingerprint+Detection [březen 2018]

2.6 Databáze otisků Fingerbank

Zajímavým projektem je Fingerbank[12], což je databáze otisků konkrétních zařízení. V současné době obsahuje 7.260.097 různých otisků, z toho 2.433.940 otisků typu UserAgent a 4.088 otisků DHCP komunikace.

Otisk zařízení se vytváří z různých vstupů získaných analýzou komunikace:

- User-Agent
- DHCPv4 fingerprint
- DHCPv6 fingerprint
- DHCPv6 vendor
- DHCPv6 enterprise
- MAC address
- mDNS service
- UPnP User Agent
- UPnP Server
- TCP SYN signature, např. 4:128+0:0:1460:8192,2:mss,nop,ws,nop,nop,sok: df,id+:0
- TCP SYN ACK signature, např. 4:128+0:0:1460:8192,2:mss,nop,ws,nop,nop, sok:df,id+:0

2.6.1 Příklad otisků databáze Firebank

Zařízení	User agent	DHCPv4	MAC OUI
Galaxy S7 Edge Version : 7.0	Mozilla/5.0 (Linux; Android 7.0; SM-G935F Build/NRD90M; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/63.0.3239.111 Mobile Safari/537.36 [FB_IAB/FB4A;FBAV/160.0.0.30.94;]	1,3,6,15,26,28,51, 58,59,43 android-dhcp-7.0	Samsung
Windows OS	ARM WinINet Downloader	1,15,3,6,44,46, 47,31,33,121,249, 43 MSFT 5.0	Intel Corporate
LG Stylo 2	Mozilla/5.0 (Linux; Android 5.1.1; LGL82VL Build/LMY47V) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.137 Mobile Safari/537.36	1,33,3,6,15,26,28, 51,58,59 dhcpcd-5.5.6	LG Electronics
HTC Android	A/7.0///QC_Reference_Phone//QCX3/ l51683183228993169 70/99000719/3456 143937/311|480+311|480//63462/63432/-	1,3,6,15,26,28,51, 58,59,43 android-dhcp-7.0	HTC Corporation

Fingerbank nabízí webové rozhraní pro vkládání a vyhledávání v databázi otisků[13], což umožňuje propojit databázi Fingerbank s dalšími systémy pro ověřování. Při dotazování vrací databáze tzv. skóre, tj. hodnotu důvěryhodnosti odpovědi. Skóre uvádí spolehlivost vrácené informace, tj. zařízení a verze. Počítá se z atributů, které se porovnají s uloženými otisky. Otisky mají různou váhu podle spolehlivosti: například informace z DHCP má větší váhu než řetězec User-Agent, který se dá lehko modifikovat. Při hodnocení se používají následující váhy:

- 50 bodů se vztahuje ke vzorům, u kterých byla shoda buď s konkrétním zařízením nebo jeho rodiči (přímých či virtuálních).
- 20 bodů se vztahuje k přímým vzorům, u kterých byla nalezena shoda (bez rodičů).
- 30 bodů se vztahuje k soupeřícím vzorům, u kterých byla nalezena shoda, ale které nejsou příbuzné. Například při shodě se vzorem Android a Windows je skóre sníženo o tuto váhu neboť se snižuje úroveň spolehlivosti, že zařízení je jedním z nalezených vzorů.

[12] Viz https://fingerbank.org/ [únor 2018]
[13] Viz https://api.fingerbank.org/api_doc/2/combinations.html [únor 2018].

2.6.2 Fingerbank Collector – analýza otisků TCP

Nové rozšíření Fingerbank tvoří tzv. Fingerbank Collector, který umožňuje analýzu otisků TCP. Kolektor naslouchá na síti a sbírá informace z komunikace. Zaměřuje se zejména na protokoly ARP, DHCPv4, DHCPv6, DNS, mDNS, HTTP, HTTPs, Radius a TCP. Mezi zajímavé informace patří MAC OUI, otisk DHCPv4, otisk TCP, seznam aktualizací OS z DNS a další.

2.7 Analýza otisků SSL/TLS

Velké množství komunikace na síti je dnes šifrováno pomocí SSL/TLS, což snižuje možnosti získávání otisků zařízení. Nicméně i při navazování komunikace SSL/TLS je možné získat informace, které lze využít k vytvoření otisku [5].

Tato metoda vychází z předpokladu, že různá zařízení používají různé verze a nastavení SSL/TLS knihovny. Při navazování komunikace TLS/SSL dochází k dohadování parametrů mezi klientem a serverem. Pro vytvoření otisku je třeba získat úvodní části komunikace, konkrétně zprávu *Client Hello*, kterou posílá SSL klient SSL serveru. Tuto zprávu lze využít k získání atributů otisku SSL/TLS.

2.7.1 Atributy

Atributy SSL/TLS, které lze získat ze zprávy *Client Hello* jsou:

- Verze protokolu SSL/TLS. Podporované verze protokolů jsou uvedeny v následující tabulce.

Verze protokolu	Kód
SSL 1.0	deprecated
SSL 2.0	deprecated
SSL 3.0	0x0300
TLS 1.0	0x0301
TLS 1.1	0x0302
TLS 1.2	0x0303
TLS 1.3	TBA

- Seznam podporovaných šifrovacích mechanismů (cipher suite list). Každý šifrovací mechanismus je identifikován dvoubytovou hodnotou daného šifrovacího mechanismu. Tyto hodnoty registruje organizace IANA[14]. Při otevírání spojení SSL/TLS nabídne klient seznam podporovaných šifrovacích mechanismů. Příklad některých hodnot je v následující tabulce.

Šifrovací sada (cipher suite)	Kód (hex | dec)	
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xc02b	49.195
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	0xc02f	49.199
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	0xc030	49.200
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	0xcca8	52.392
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	0xcca9	52.393

- Seznam podporovaných rozšíření (TLS extensions). Podobně jako u šifrovacích mechanismů, posílá SSL/TLS klient seznam podporovaných rozšíření. Hlavička obsahuje typ rozšíření, jeho délku a hodnoty. Typy rozšíření definuje IANA[15]. Příklad rozšíření je v následující tabulce.

[14] Viz https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml [březen 2018]
[15] Viz https://www.iana.org/assignments/tls-extensiontype-values [březen 2018]

Rozšíření (extension)	Kód
server_name	0
client_certificate_url	2
status_request	5
supported_groups	10
ec_point_format	11
application_layer_protocol_negotiation	16
extended_master_secret	23
SessionTicket TLS	35

2.7.2 Vytvoření otisku

Otisk může být tvořen výše uvedenými hodnotami. Lze ho například získat z PCAP komunikace pomocí jednoduchého dotazu příkazem tshark[16]: Tento příkaz vyhledá v zadaném soubor PCAP komunikaci SSL s parametrem handshake.type==1 (Client Hello) a vypíše otisk ve formát `IP adresa - verze SSL/TLS - seznam šifrovacích mechanismů - seznam rozšíření`:

```
# tshark -r <file.pcap> -T fields -e ip.src -e ssl.handshake.version
-e ssl.handshake.ciphersuite  -e ssl.handshake.extension.type
-R "ssl.handshake.type==1" -2

147.229.14.156 - 0x00000303 - 49195,49199,52393,52392,49196,49200,
49162, 49161,49171,49172,51,57,47,53,10 - 0,23,65281,10,11,35,16,5,13
```

3 Způsoby detekce komunikujících mobilních zařízení

Mobilní zařízení mají oproti klasickým počítačům odlišné vlastnosti a chování:

- Zařízení jsou optimalizovány na nižší spotřebu energie a omezenější hardware.
- Operační systém mobilních zařízení se liší od klasických operačních systémů typem a možnostmi konfigurace. Uživatel nemá zpravidlamožnost měnit různá nastavení.
- Pro datovou komunikaci využívají zejména mobilní datová síť (GPRS, Edge, LTE, apod.) či bezdrátové spojení WiFi.

Z pohledu detekce a vytváření otisků zařízení nás nejvíce zajímá komunikace na vyšších vrstvách modelu TCP/IP, tj. síťová, transportní a aplikační, ke které budeme přistupovat. Tato komunikace je obvykle stejná bez ohledu na připojení, tzn. zda přistupujeme přes mobilního operátora či využíváme WiFi připojení přes poskytovatele internetového spojení ISP.

V současné době je většina mobilní komunikace na vyšších vrstvách šifrovaná, což do značné míry limituje použití techniky pro detekci a vytváření otisků. Na druhou stranu lze získat i dostatek informací z nešifrované komunikace.

Při vytváření otisků musíme vycházet z toho, že přesnost otisku se během času mění, například aktualizací softwaru, přidáním nové aplikace, změnou konfigurace a podobně. Z tohoto důvodu popíšeme v následujícím textu faktory, které mohou způsobit změnu atributů, které se používají k vytváření otisků.

3.1 Struktura mobilní komunikace

Na základě experimentů, viz kapitola **??**, jsme zjistili, že přes 90% mobilní komunikace je šifrovaných. Nešifrovaný provoz tvoří jednak komunikace na lokálních sítích typu ARP, EAP, ICMP, IGMP, dále pak služby nad UDP, například DNS, DHCP, NTP, Netbios, či TCP, zejména HTTP

[16] Viz https://isc.sans.edu/forums/diary/Browser+Fingerprinting+via+SSL+Client+Hello+Messages/17210 [březen 2018]

či MQTT. Pokud nemáme k dispozici SSL/TLS proxy zařízení (útoky typu MITM), můžeme pro detekci využít pouze nešifrovanou komunikaci.

3.2 Získávání atributů z komunikace mobilních zařízení

3.2.1 Komunikace HTTP

- Položka `User-Agent` v hlavičce HTTP, která obsahuje název a typ webového prohlížeče, název typ operačního systému, model telefonu, podporované jazyky a další.
- Položka `Accept` popisuje podporované datové typy.
- Položka `Accept-Encoding` obsahuje seznam podporovaných kódování prohlížeče.
- Položka `Accept-Charset` obsahuje podporované sady znaků.

3.2.2 Analýza komunikace DNS

3.2.3 Analýza komunikace TCP/IP

Zde můžeme využít atributy, které využívají nástroje typu Nmap či p0f, viz section 2.1.1 a 2.2.

Zajímavé může být použití IP adres, které mohou prostřednictvím služby Whois či geolokace upřesnit místo připojení. Toto je ale možné pouze u veřejných adres IPv4 a IPv6. V dnešní době většina mobilní komunikace využívá privátních adres přidělených poskytovatelem připojení, které jsou na straně poskytovatele překládany na veřejnou adresu. Je možné, že se časem rozšíří podpora veřejných IPv6 adres i na mobilní zařízení.

Literatura

[1] Alexander, S.; Droms, R.: *DHCP Options and BOOTP Vendor Extensions*. IETF RFC 2132, March 1997.

[2] Broenink, R.: Using Browser Properties for Fingerprinting Purposes. In *16th Twente Student Conference on IT*, January 2012.

[3] Crotti, M.; Dusi, M.; Gringoli, F.; aj.: Traffic Classification Through Simple Statistical Fingerprinting. *SIGCOMM Comput. Commun. Rev.*, ročník 37, č. 1, Leden 2007: s. 5–16, ISSN 0146-4833, doi:10.1145/1198255.1198257.
URL http://doi.acm.org/10.1145/1198255.1198257

[4] Eckersley, P.: How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, Berlin, Heidelberg: Springer-Verlag, 2010, ISBN 3-642-14526-4, 978-3-642-14526-1, s. 1–18.
URL http://dl.acm.org/citation.cfm?id=1881151.1881152

[5] Husák, M.; Čermák, M.; Jirsík, T.; aj.: HTTPS Traffic Analysis and Client Identification Using Passive SSL/TLS Fingerprinting. *EURASIP Journal on Information Security*, ročník 2016, 2016, ISSN 1687-4161, doi:http://dx.doi.org/10.1186/s13635-016-0030-7.
URL http://www.jis.eurasipjournals.com/content/2016/1/6

[6] Johnston, M.; Venass, S.: *Dynamic Host Configuration Protocol (DHCP) Options for the Intel Preboot eXecution Environment (PXE)*. IETF RFC 4578, November 2006.

[7] Khanna, V. K.: Remote fingerprinting of mobile phones. *IEEE Wireless Communications*, ročník 22, č. 6, December 2015: s. 106–113, ISSN 1536-1284, doi:10.1109/MWC.2015.7368831.

[8] Kumari, W.; Gudmundsson, O.; Ebersman, P.; aj.: *Captive-Portal Identification Using DHCP or Router Advertisements (RAs)*. IETF RFC 7710, December 2015.

[9] Lemon, T.; Sommerfeld, B.: *Node-specific Client Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)*. IETF RFC 4361, February 2006.

[10] Lippmann, R.; Fried, D.; Piwowarski, K.; aj.: Passive Operating System Identification from TCP/IP Packet Headers. In *Proceedings Workshop on Data Mining for Computer Security (DMSEC)*, 2013.

[11] Lyon, G. F.: *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Nmap Project, 2009.

[12] Matsunaka, T.; Yamada, A.; Kubota, A.: Passive OS Fingerprinting by DNS Traffic Analysis. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, March 2013, ISSN 1550-445X, s. 243–250, doi:10.1109/AINA.2013.119.

[13] Networks, A.: *ArubaOS DHCP Fingerprinting*. Technical report, Aruba Networks, 2011.
URL http://www.arubanetworks.com/assets/vrd/AOS-DHCP-FingerPrint-AppNote.pdf

Příloha 1: Přehled otisků DHCP

Následující seznam otisků DHCP pochází z Aruba Networks[17]. Typ volby je uložen v položce ID v decimálním tvaru.

Zařízení	Volba	Otisk
Android 2.x	55	37017921030
Samsung Galaxy S, Android 2.355	55	3701792103061c333a3b
Blackberry	55	370103060F
iPad	55	370103060F77FC
Apple Mac Book	55	370103060F775FFC2C2E2F
Nokia N900, Maemo OS	55	370103060c0f111c28292a
Nintendo DS	55	37010306
Playstation 3	55	equals 3701031c060f
Nokia N98, Symbian OS	55	370C060F01031C78
HTC, Win Mobile 6.x	55	370103060f2c2e2f
Windows XP	55	37010f03062c2e2f1f21f92b
Windows Vista	55	37010f03062c2ef1f2179f92b
Windows 7 (Korean edition)	55	37010f03062c2ef1f2179f92b
Windows 7 (English edition)	55	37010f03062c2ef1f2179f92b
Windows (Multi-version)	55	37010F03062C2E2F1
Cisco 1750 VPN	55	3701060F2C0321962B
Debian/Linux 2.6 generic	55	37011C02030F0677
Linux (unknown)	55	37011C02030F06770C2C2F1A792A
Linux Debian 2.6.35	55	37011c02030f06770c2c2f1a
Palm PDA	55	37011C02030F060C
Samsung s8000	55	3701020304050607080 90C0D0F1011171A1C2A2C3 233353638
Windows CE Casio Scanner	55	370103060F2C2E2F
Windows CE Symbol Scanner	55	370103060F2C2E2F4243
Windows phone 7	55	370103060f2c2e2f
Android 2.x (multiple)	60	3c6468637063642034
BlackBerry	60	3c426c61636b4265727279
Nokia N900, Maemo OS	60	3c756468637020302e392e39
Windows CE	60	3c4d6963726f736f66742057696e646f777320434500
Windows (Multiple)	60	3c4D53465420352E30

[17] Viz http://community.arubanetworks.com/t5/Community-Tribal-Knowledge-Base/Mobile-Device-Signatures-DHCP-Fingerprints/ta-p/16749 [březen 2018]