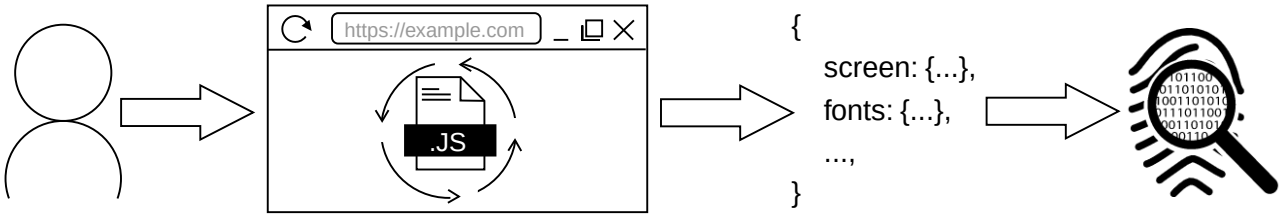
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2021
http://excel.fit.vutbr.cz

Získávání informací o uživatelích na webových stránkách

Tomáš Vondráček*



Abstrakt
Práce se zabývá vytvořením knihovny pro získávání informací o uživatelích na webových stránkách, kde získané informace mohou být použity pro identifikaci uživatelů. Navrhl jsem nové techniky získávání informací a optimalizoval některé aktuálně používané techniky. Implementovány jsou také techniky, kterými lze odhalit vedlejší efekty způsobené webovými rozšířeními, které maskují identitu uživatelů. Na základě analýzy webových rozšíření byly detekovány dříve neznámé informace, které mohou být taktéž použity pro identifikaci uživatele. Analyzovány jsou i získané informace o prohlížeči a zařízení z hlediska míry získané informace, doby jejich získání a stability v čase. Z výsledků analýzy lze optimalizovat množství získávaných informací, čímž lze omezit potenciální zpomalení webových stránek.

Klíčová slova: Identifikace uživatelů — Otisk prohlížeče — Otisk zařízení — Detekce webových rozšíření

Přiložené materiály: N/A

*xvondr23@stud.fit.vutbr.cz, *Fakulta informačních technologií, Vysoké učení technické v Brně*

1. Úvod

K síti Internet je v roce 2021 připojeno na 4.8 miliardy uživatelů a existuje přes 1.8 miliardy webových stránek[1]. Každá webová stránka přitom může o svých uživatelích získat mnoho informací, kde jejich kombinací lze vytvořit potenciálně unikání digitální otisk uživatele. Otisk, resp. informace obsažené v otisku, mohou v praxi například využít:

- reklamní agentury pro cílení reklamy [1],
- protokoly zajišťující autentizaci a autorizaci [2],
- společnosti určující rizikové skóre [3].

Problematika získávání informací je komplexní nikdy nekončící proces. Je tomu tak dáno neustálým vývojem prohlížečů, kdy dochází k přidávání a odebírání webových rozhraní. Správně navržená knihovna pro získávání informací by měla získat co nejvíce autentická data o uživatelově prostředí a neměla by s daty nikterak manipulovat, aby nedocházelo ke zkreslení, nebo změně významu získané informace. Minimální režie knihovny také přispívá k plynulejšímu chodu webových stránek, na kterých je knihovna nasazena.

Uživatelé mají v dnešní době na výběr desítky bezplatných webových rozšíření, které mohou na několik málo kliknutí nainstalovat do svého prohlížeče. Rozšíření nabízí bezplatné služby VPN, nebo zvýšení anonymity uživatele zamaskováním některých informací v prohlížeči. Způsob maskování informací typ-

[1] https://www.internetlivestats.com

icky probíhá buď nahrazením hodnoty, nebo přidáním šumu do výsledku. Konkrétní důsledky implementace maskování informací mohou být ve většině případů odhaleny, čímž dojde k poskytnutí dodatečných informací využitelných pro identifikaci uživatele.

Tato práce nabízí knihovnu pro získávání až 127 informací. Knihovna je implementovaná v jazyce TypeScript a pokryta je 142 jednotkovými testy v nástroji Cypress. Knihovna umožňuje získat informace o zařízení, prohlížeči, chování uživatele (pohyby myši, stisky kláves a dotyk) a vedlejších efektů způsobených webovými rozšířeními.

Tato práce představuje některé nové, dříve nepublikované, zdroje informací o zařízení nebo prohlížeči a některé z nich optimalizuje z hlediska potenciálně vyšší míry získané informace. Další významný přínos představuje velká analýza webových rozšíření maskujících identitu uživatele, na kterých jsou úspěšně detekovány jimi způsobované vedlejší efekty v prohlížečích. V neposlední řadě je také provedena analýza získaných informací prostřednictvím implementované knihovny.

2. Existující knihovny pro získávání informací

Nejmodernější a nejpoužívanější knihovna pro získávání informací o uživatelích je FingerprintJS[2], která je aktuálně ve verzi `3.0.6`. Knihovna vznikla již v roce 2015 a nyní nabízí 27 informací o prohlížeči a zařízení. S příchodem 3. verze knihovny skončila podpora rodiny prohlížečů Trident a obecně starších typů prohlížečů, které nepodporují rozhraní `Promise`. Z hlediska kvality získávání informací jsem identifikovat následující nedostatky:

Počet logických procesorů Při získaní počtu logických procesorů není rozlišeno mezi případem, kdy atribut není dostupný (rodina prohlížečů Trident a staré verze prohlížečů), a případem, kdy počet nemohl být získán (dle specifikace[3] má být vráceno číslo 1). V obou případech je vráceno číslo 1, a proto nelze tyto dva případy od sebe rozeznat.

Webové úložiště Detekce podpory webových úložišť[4] je ošetřena pro případ možného vzniku výjimky z důvodu jejich zakázání. Avšak pokud nastane výjimka, informace o ní je ztracena, protože je vrácena hodnota `true` stejně jako v případě, kdy úložiště není zakázané a současně je podporované. Vzhledem k tomu, že informace o podpoře je v obou případech přetypovaná na pravdivostní hodnotu, je ztracena i informace o zakázaní webových úložišť na stránce `about:config` v prohlížeči Firefox.

Rozměry zobrazovacího zařízení Šířka a výška zobrazovacího zařízení je reprezentována jako dvojice. Na dvojici je však vždy uplatněno sestupné řazení. Důsledek této operace je takový, že od sebe není možné odlišit zobrazovací zařízení na výšku a na šířku.

Zajímavé vylepšení, které knihovna FingerprintJS implementuje, spočívá v nápravě techniky zpracování zvukového signálu. Náprava spočívá v opakovaných pokusech o začátek zpracování v případě, že ke zpracování nedojde z důvodu například neaktivity záložky prohlížeče.

Mimo knihovnu FingerprintJS existují i jiné knihovny pro získávání informací. Žádná z knihoven však neimplementuje nové metody, které by knihovna FingerprintJS již nenabízela.

3. Návrh knihovny pro získávání informací

Tato sekce představuje nové metody získávání informací navržené v této práci a také metody, u kterých došlo k zajímavým vylepšením. Knihovna umožňuje získat 127 informací o zařízení a prohlížeči organizovaných ve 21 skupinách, což je až 4× více informací, než je dostupných v knihovně FingerprintJS.

3.1 Nové metody získávání informací

V této sekci jsou představeny pouze zajímavější nové metody, které jsem navrhl a které dle mých znalostí nebyly publikovány v kontextu získávání informací.

Kvalita obrazovky Kvalita obrazovky je získána CSS vlastnosti média `"color-gamut"`. Detekovány jsou barevné gamuty `srgb`, `p3` a `rec2020`.

Detekce letního času Detekce letního času spočívá ve výběru dvou dat, kde jedno datum patří do období letního času a druhé tam nepatří. Pokud je rozdíl posunů zvolených dat nenulový, pak je detekován letní čas.

Podpora dotyku v jazyce CSS Podpora dotyku využívá CSS vlastností média `hover` a `pointer`. Zařízení podporující dotyk splňuje vlastnosti média `hover:none` a `pointer:coarse`.

Snížení přesnosti časových údajů Tor Browser [4] ve výchozím nastavení snižuje přesnost časových údaje na 100 ms. Stejné snížení přesnosti lze

[2] https://github.com/fingerprintjs/fingerprintjs

[3] https://html.spec.whatwg.org/multipage/system-state.html

[4] `sessionStorage` a `localStorage`

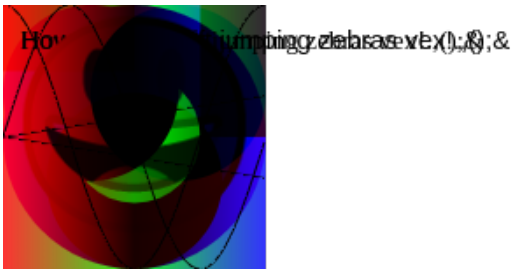
provést i v rodině prohlížečů Gecko na adrese `about:config`. Pokud jsou výsledky volání funkcí, které pracují s časem dělitelné 100, pak velmi pravděpodobně dochází ke snížení přesnosti časových údajů.

3.2 Vylepšené metody získávání informací

V této sekci jsou představeny metody získávání informací, které jsem optimalizoval z hlediska vyšší míry získané informace.

Podpora úložišť Knihovna FingerprintJS umožňuje získat pouze pravdivostní hodnotu o podpoře úložišť. Mimo tu však má implementace knihovny umožňuje získat i hodnoty `undefined`, `null`, nebo název výjimky. Tyto hodnot odpovídají různým stavům úložišť.

Prvek canvas Knihovna FingerprintJS implementuje 4 techniky získávání informací z prvku canvas. Stabilita techniky vykreslující emodži je však mnohem menší než stabilita vykreslení textu v písmu Arial [5]. Pokud tedy například dojde k aktualizaci operačního systému, ve které jsou aktualizovány emodži, bude výsledek informace z prvku canvas díky hašovací funkci velmi odlišný. Navržené vylepšení spočívá v izolaci jednotlivých technik. Pokud dojde ke změně v jedné technice a ostatní zůstanou stejné, je stále možné s určitou pravděpodobností identifikovat uživatele i přes změnu části otisku. Další vylepšení spočívá v implementaci 4 nových technik. Aplikace všech technik je zobrazena na obrázku 1.



Obrázek 1. Vykreslený obrázek v prvku canvas po aplikaci 8 technik zaměřujících se na detekci SW a HW zařízení.

4. Detekce webových rozšířeních

V rámci práce se také zabývám detekcí webových rozšíření, které maskují identitu uživatelů. Hlavním cílem těchto webových rozšíření je zvýšení anonymity uživatelů. Pro dosažení tohoto cíle webové rozšíření používají různé techniky maskování informací. Mezi často používané techniky patří nahrazení hodnoty jinou hodnotou, nebo přidáním šumu do výsledku.

Zásahy webových rozšíření do prohlížečů jsou však velmi často realizovány způsobem, při kterém vznikají nežádoucí vedlejší efekty, které lze detekovat. Vedlejší efekty způsobené webovými rozšířeními lze podle typu získané informace kategorizovat jako anomálie, nebo nekonzistence.

4.0.1 Anomální informace

Anomální informace představuje hodnotu, která za standardní situace nemůže nastat, nebo nestandardní stav objektu daný úpravou jeho vlastností. Příkladem je přepsání nativní funkce prohlížeče webovým rozšířením, nebo zamknutí objektu vůči úpravám. Anomální informace jsou detekovány v následujících kategoriích.

Falešná nativní funkce Specifikace jazyka JavaScript [6] definuje nativní, běžné a exotické funkce na základě jejich interních slotů. Od interních slotů se také odvíjí řetězcová reprezentace každé funkce. Na základě řetězcové reprezentace je poté možné rozlišit mezi všemi typy funkcí.

Rozšiřitelnost objektu Pokud není definováno jinak, objekt je vždy otevřen vůči rozšiřitelnosti, tj. lze na něj přidat atribut. Rozšiřitelnost lze ověřit funkcí `Object.isExtensible`.

Konfigurovatelnost objektu Pokud je objekt otevřen vůči konfiguraci, je možné přímo na něm smazat atribut. Smazání atributu by vždy mělo proběhnout úspěšně bez vyvolání výjimky. Hodnota atributu je i po smazání atributu dostupná z prototypu daného objektu.

Detekce rozdílných výsledků Detekce rozdílných výsledků spočívá ve dvou spuštění funkcí a následném porovnání jejich výsledků. Pokud výsledky nejsou stejné, je detekována anomálie. Tato detekce má za cíl detekovat zejména webové rozšíření, které přidávají šum náhodným způsobem.

4.0.2 Nekonzistentní informace

Nekonzistentní informace mají platnou hodnotu, avšak tato hodnota nemůže nastat současně s hodnotou jiné informace. Příkladem může být situace, kdy informace z HTTP hlavičky `User-Agent` nekoresponduje s informací získanou z atributu `userAgent` na objektu `navigator`. Nekonzistentní informace jsou detekovány v následujících kategoriích.

Rodina prohlížečů Rodina prohlížečů lze detekovat na základě výjimek a řetězcové reprezentace funkce eval [7]. Další možností je použít názvy pluginů a řetězec `User-Agent`.

Operační systém Operační systém lze detekovat z informací o platformně a grafické kartě [7]. Další možností je opět řetězec `User-Agent`.

Zobrazovací zařízení Na zobrazovacím zařízení se detekují nekonzistence v orientaci, barevné hloubce, poměru rozměrů a podpoře dotyku.

Preferované jazyky První jazyk v poli `languages` by měl odpovídat jazyku v atributu `language`.

Úložiště Úložiště `session` a `local` jsou konfigurovány společně, a tedy informace o podpoře by měla být totožná. Informace o podpoře ukládání souborů cookies by měla korespondovat s hodnotou atributu `cookieEnabled`.

HTTP hlavičky Informace z HTTP hlaviček se porovnávají vůči informacím získaným z jazyka JavaScript. Detekovány jsou nekonzistence v řetězci `User-Agent`, preferovaných jazycích a preferenci „Do Not Track“. Pokud webová stránka explicitně povolí hlavičky Client Hints třetí straně, jsou použity i informace z nich.

5. Analýza webových rozšíření

Analyzoval jsem celkem 19 webových rozšíření, kde některé z nich používají desetitisíce uživatelů. Na všech webových rozšířeních se podařilo detekovat anomální nebo nekonzistentní informace. V některých případech se povedlo i získat originální hodnoty maskovaných informací na základě smazaných atributů. Veškeré dosažené výsledky jsou shrnuty v tabulce 1.

Z výsledků v tabulce 1 je patrné, že na základě počtu detekovaných nekonzistencí a anomálií lze provést odhad typu webového rozšíření. Nejvíce anomálních informací bylo detekováno technikou falešných nativních funkcí a nejvíce nekonzistencí bylo nalezeno v rodinách prohlížečů.

Úspěšně detekované anomálie nebo nekonzistence mohou také zlepšit dlouhodobou identifikaci uživatelů. Pokud bude například detekována anomálie ve funkci `toDataURL`, která slouží k získání informací z prvku canvas, nebudou informace přidány do otisku, čímž se omezí falešně negativní výsledky identifikace.

6. Analýza získaných informací

Implementovaná knihovna byla od 1.3.2021 postupně nasazena na 4 komerční webové stránky. Informace jsou získány vždy pouze jednou v rámci relace, z důvodu omezení počtu duplicitních otisků. Tímto omezením je také sníženo zatížení webových stránek způsobené knihovnou. Za více než měsíc bylo z těchto 4 webů získáno **7 632** otisků.

Analýza dat se bude zabývat zejména mírou získané informace, rychlosti získání skupin informací a také stabilitou informací v čase. Míra získané informace je vyjádřena pomocí normalizované informační entropie a stabilita je posuzována na základě počtu změn hodnot jednotlivých informací v rámci jednotlivých uživatelů.

Tabulka 1. Tabulka demonstruje, kolik anomálních a nekonzistenich informací se podařilo získat z webových rozšířeních pomocí technik popsaných v sekci 4.

Webové rozšíření	Anomálie	Nekonzistence
AudioContext Fingerprint Defender	1	0
Browser Fingerprint Protector	8	2
Canvas Blocker – Fingerprint Protect	2	0
CanvasFingerprint-Block	2	0
Canvas Fingerprint Defender	3	0
CyDec Platform Anti-Fingerprinting	0	6
Don't FingerPrint Me	34	0
Fingerprint Shield	2	0
Fingerprint Spoofing	2	4
Font Fingerprint Defender	1	0
Hide My Back	0	2
JavaScript Restrictor (lvl. 3)	12	1
RandomUA	0	4
Random User-Agent	0	1
Resist Fingerprinting	1	3
Trace – Online Tracking Protection	8	6
User-Agent Switcher	0	4
User-Agent Switcher and Manager	3	3
WebGL Fingerprint Defender	1	1
Celkem: 19 rozšíření	80	37

6.1 Příprava datové sady

Příprava datové sady na analýzu se sestává z následujících kroků[5]:

1. Odstranění otisků s neplatnou hodnotou (10).
2. Odstranění duplicitních otisků uživatelů (3 987).
   (a) Identifikátor u uživatele (1 787).
   (b) Cookie třetí strany (2 200).
3. Odstranění otisků od robotů (282).

[5] V závorkách je uvedeno, kolik otisků bylo odstraněno.

4. Stabilizace informací.
5. Výpočet nových informací.

Duplicitní otisky jsou odstraněny, protože v rámci analýzy entropie (sekce 6.3) jsou důležité informace od unikátních, nikoliv stejných uživatelů. Informace od stejných uživatelů jsou použity pouze pro analýzu stability otisků (sekce 6.4).

6.2 Popis datové sady

Po provedení přípravy dat zbylo potenciálně **3 353** unikátních otisků, na kterých je provedena analýza. Mobilních zařízení se v datové sadě nachází 45 %, osobních počítačů 39 %, notebooků 12 % a zbylá 4 % připadají tabletům.

Normalizovaná entropie všech informací v datové sadě z této práce je rovna 0.99, což je velmi vysoké skóre. Z 3 353 otisků je 96 % otisků rozdílných (vyskytujících se alespoň jednou) a 94 % otisků unikátních (vyskytujících se právě jednou).

6.3 Normalizována entropie a rychlost získání informací

Rychlost získání informací hraje velmi důležitou roli v získávání informací zejména z hlediska snížení zátěže webové stránky a rychlosti následné identifikace uživatele. V rámci analýzy informací jsem proto měřil dobu, jakou trvá získat skupinu informací. Skupina informací obsahuje tematicky seskupené informace. Výsledek analýzy je zobrazen v grafu na obrázku 2.

Nejlepší poměr entropie vůči času získání dosahuje skupina informací z rozhraní WebGL. Nízká doba získání je dána tím, že k získání informací stačí pouhé přečtení atributů. Vysoká entropie je dána zase tím, že zařízení obsahují velké množství různých modelů grafických karet. Druhého nejlepšího poměru dosahují standardní informace o obrazovce, protože informace lze taktéž získat velmi rychle a zejména na mobilních zařízeních jsou rozměry velmi odlišné.

Naopak nejhoršího poměru dosahují 3 skupiny informací ze `IV`. kvadrantu. Pokud by tyto informace knihovna nezískávala, snížila by se unikátnost otisků o 1.3 % na 92.7 %. Současně by však klesla průměrná doba získávání informací o 51 % na 2 496 ms. Nezískáním informací ze `IV`. kvadrantu se tedy mírně sníží unikátnost otisku, ale výrazně se sníží doba získávání otisku.

6.4 Stabilita získaných informací

Stabilita získaných informací představuje další důležitou metriku zejména z hlediska dlouhodobé identifikace. Pokud by informace s vysokou entropií byla vysoce nestabilní, tj. byla by při každém získání jiná, není na jejím základě možné uživatele identifikovat.

V tabulce 2 je zachycen počet změn v rámci skupin informací zaznamenaných u jednotlivých uživatelů. Z tabulky lze pozorovat, že informace s vyšší entropií jsou náchylnější na častější změny v čase. Naopak informace s nižší entropií podléhají méně změnám v čase, a tudíž je jejich stabilita vyšší.

Výjimku však tvoří 3 skupiny informací, které současně dosahují vyšší entropie a je u nich zaznamenáno nižšího počtu změn. Těmito skupinami jsou informace o písmech, rozhraní WebGL a hlaviček protokolu HTTP. Všechny tyto informace lze tedy považovat za velmi kvalitní zdroje informací.

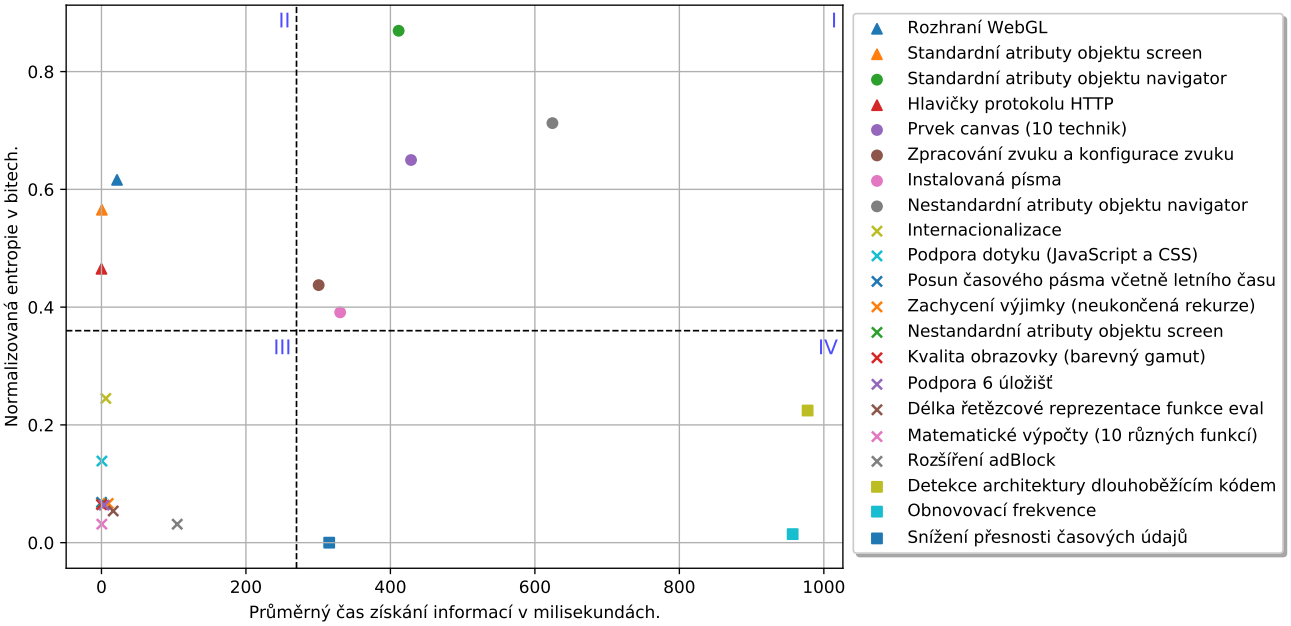
Tabulka 2. Tabulka demonstruje celkový počet změn u jednotlivých skupin informací rozdělených podle typu zařízení, ze kterého informace pochází. Skupiny informací jsou seřazeny podle počtu změn. U každé skupiny informací se nachází příznak, zda skupina patří do kvadrantu `I`, nebo `II`. Tyto dva kvadranty obsahují informace s vyšší entropií. V tabulce není zobrazeno zbylých 10 skupiny informací, kde všechny tyto skupiny patří do kvadrantu `III`, nebo `IV`.

Skupina informací	**Zařízení**		**Kvadrant** `I`, nebo `II`
	Počítač	Mobil	
Nestandardní atributy objektu navigator	215	318	✓
Prvek canvas	102	110	
Standardní atributy objektu screen	157	14	
Standardní atributy objektu navigator	96	48	
Detekce architektury dlouhoběžícím kódem	92	36	✗
Nestandardní atributy objektu screen	45	0	
Zvuk	26	21	✓
Rozšíření adBlock	8	0	✗
Hlavičky HTTP	28	6	✓
Rozhraní WebGL	24	0	
Instalovaná písma	4	0	

7. Závěr

Cílem práce bylo vytvořit knihovnu pro získávání informací na webových stránkách. Vytvořená knihovna je pokryta automatizovanými testy napříč dvěma rodinami prohlížečů a obsahuje 127 informací organizovaných ve 21 skupinách.

Knihovna je aktuálně již více než měsíc nasazená v produkčním prostředí a za celou dobu nebyla zaznamenána žádná běhová chyba. Získáno bylo celkem



Obrázek 2. Na grafu je zobrazena normalizovaná entropie vůči průměrné době získání skupiny informací. Názvy skupiny informací jsou v legendě seřazeny od nejlepšího poměru entropie vůči času získání po nejhorší. Graf jsem rozdělil do 4 kvadrantů, kde kvadranty `I` a `II` obsahují skupiny informací s vyšší entropií a kvadranty `III` a `IV` obsahují skupiny informací s nižší entropií. Skupiny informací v kvadrantech `II` a `III` lze získat velmi rychle a získání skupin informací v kvadrantech `I` a `IV` trvá delší dobu, která je způsobena vyšší výpočetní náročností. Podrobnější popis informací ve skupinách se nachází v příloze 7.1.

7 632 otisků, ze kterých pocházelo 3 353 otisků od potenciálně unikátních uživatelů. Z těchto uživatelů mělo až 94 % zcela unikátní otisk.

Hlavní přínos této práce spočívá ve vytvořené knihovně, která implementuje nové metody získávání informací a optimalizuje aktuálně používané metody. Další přínos spočívá v potvrzení, že webové rozšíření maskující identitu uživatelů přidávají nepřímo zcela nové informace, které mohou být použity k identifikaci uživatelů.

Analýzy provedené v této práci ukazují, že uživatele lze na webových stránkách téměř vždy identifikovat. Autoři webových rozšíření, které maskují identitu uživatelů, by měli na základě výsledků této práce napravit způsoby, kterými dochází k maskování informací.

Poděkování

Velmi děkuji panu Ing. Liborovi Polčákovi, Ph.D. za jeho cenné rady při psaní článku a vedení práce.

Literatura

[1] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. *ACM Trans. Web*, 14(2), April 2020.

[2] J. LeBlanc and T. Messerschmidt. *Identity and Data Security for Web Development: Best Practices*. O'Reilly Media, 2016.

[3] Amin FaizKhademi, Mohammad Zulkernine, and Komminist Weldemariam. Fpguard: Detection and prevention of browser fingerprinting. In Pierangela Samarati, editor, *DBSec*, volume 9149 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2015.

[4] Mike Perry, Erinn Clark, Steven Murdoch, and Georg Koppen. The design and implementation of the tor browser, June 2018. https://2019.www.torproject.org/projects/torbrowser/design.

[5] Anna Kobusińska., Jerzy Brzeziński., and Kamil Pawulczuk. Device fingerprinting: Analysis of chosen fingerprinting methods. In *Proceedings of the 2nd International Conference on Internet of Things, Big Data and Security - Volume 1: IoTBDS,*, pages 167–177. INSTICC, SciTePress, 2017.

[6] Jordan Harband and Kevin Smith. Ecmascript® 2020 language specification. Technical report, Ecma International, 2020. https://262.ecma-international.org/11.0.

[7] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. Fp-scanner:

The privacy implications of browser fingerprint inconsistencies. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 135–150, Baltimore, MD, August 2018. USENIX Association.

Přílohy

7.1 Skupiny informací

Tato sekce popisuje skupiny informací zobrazené na obrázku 2 a tabulce 2, které obsahují více informací.

7.1.1 Objekt navigator a screen

Výčet získaných informací z těchto objektů včetně rozdělení na standardní a nestandardní je zachycen v tabulce 3.

Tabulka 3. Rozdělení atributů v objektech `navigator` a `screen` na standardní a nestandardní.

Typ	Atributy
`navigator`	
Standardní	`userAgent`, `platform`, `appVersion`, `language`, `languages`, `plugins`, `hardwareConcurrency`, `mimeTypes`, `mediaDevices`, `appCodeName`, `appName`, `cookieEnabled`, `maxTouchPoints`, `doNotTrack`, `javaEnabled`, `onLine`, `product`, `productSub`, `vendor`, `vendorSub`
Nestandardní	`oscpu`, `cpuClass`, `userLanguage`, `browserLanguage`, `msDoNotTrack`, `msMaxTouchPoints`, `battery`, `connection`, `keyboard`, `gamepads`, `bluetooth`, `xr`, `deviceMemory`, `wakeLock`
`screen`	
Standardní	`devicePixelRatio`, `width`, `height`, `availWidth`, `availHeight`, `colorDepth`, `pixelDepth`, `orientation`
Nestandardní	`availLeft`, `availTop`

7.1.2 Úložiště

Podpora úložišť je detekována pro `caches`, `cookie`, `indexedDB`, `localStorage`, `sessionStorage` a `webSQL`.

7.1.3 Kvalita obrazovky

Detekována je podpora barevného gamutu `srgb`, `p3` a `rec2020`.

7.1.4 Podpora dotyku

Podpora je detekována pomocí jazyka CSS i JavaScript. V jazyce JavaScript je testována možnost vytvoření události `"TouchEvent"` a také existence atributu `ontouchstart` na objektu `window`. V jazyce CSS se používají vlastnosti média `hover` a `pointer`.

7.1.5 Internacionalizace

Z rozhraní `DateTimeFormat` jsou získány atributy `locale` a `timeZone`.

7.1.6 Časové pásmo

Z rozhraní `Date` jsou získány dvě časové pásma funkcí `getTimezoneOffset`.

7.1.7 Matematické funkce

Použity jsou matematické funkce `asinh`, `acosh`, `atanh`, `expm1`, `log1p`, `sinh`, `cosh`, `tanh`, `tan` a `cos`.

7.1.8 Prvek canvas

Použity jsou techniky vykreslení textu v písmu Arial, vykreslení textu v záložním písmu, emodži, míchání barev, vyplňování oblastí, matematické funkce, lineární barevný přechod, transformace objektů, vyhlazování písma a kombinace všech technik (obrázek 1).

7.1.9 Zpracování zvuku a konfigurace zvuku

Zpracování vygenerovaného zvukového signálu je realizováno rozhraním `OfflineAudioContext` a konfigurace zvuku je získána z rozhraní `AudioContext`.

7.1.10 Rozhraní WebGL

Z rozhraní WebGL jsou získány informace o grafické kartě, podporovaných rozšířeních a verzi rozhraní.