Device Fingerprinting mit dem Web-Browser

Thomas Pieronczyk
Betreuer: Ralph Holz
Seminar Future Internet SS2012
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: thomas.pieronczyk@tum.de

KURZFASSUNG

Nahezu jeder Computer besitzt heutzutage einen Internetanschluss. Web-Browser sind dabei die Hauptschnittstelle zum World Wide Web. Neben all ihren Vorzügen stellen sie jedoch auch ein ernstzunehmendes Sicherheitsrisiko für die Privatsphäre dar. Der Grund dafür ist die Preisgabe von Informationen wie z. B. installierte Plugins, System-Schriftarten, das genutzte Betriebssystem, oder der genutzte Browser. Diese Informationen können genutzt werden um Nutzer im Internet zu identifizieren.
In dieser Arbeit wird die Identifizierung von Nutzern im Internet anhand der von ihnen hinterlassenen digitalen Spuren (sog. Privacy Footprints) erörtert. Es wird aufgezeigt wie digitale Spuren im Internet ausgelesen und gespeichert werden und welche Gefahren dadurch für die Privatsphäre entstehen können. Es werden Methoden vorgestellt, mit denen man aus den gesammelten Spuren digitale Fingerabdrücke (sog. Device-Fingerprints) erzeugen kann. Im Anschluss wird das Thema Device Fingerprinting mit Hilfe von Web-Browsern genauer betrachtet. Basis für diese Betrachtung ist die Publikation „How unique is your web browser“ von Peter Eckersley [6].
Es wird aufgezeigt, wie Internetnutzer über frei zugängliche Informationen relativ eindeutig identifiziert und verfolgt werden können und welche Möglichkeiten existieren, diese Identifizierung zu erschweren.
Peter Eckersley zeigt, dass man einerseits den Web-Browser gut verwenden kann um Internet-Nutzer zu identifizieren, andererseits aber auch, dass die bisher gesammelten Versuchsdaten nicht ausreichend, bzw. nicht geeignet sind, um eine globale Identifizierbarkeit von Internetnutzern bestätigen zu können.

Schlüsselworte

Device Fingerprinting, Web-Browser, Digitale Spuren im Internet, Datenschutz, Privacy

1. EINLEITUNG

In der heutigen Gesellschaft stellen Informationen aller Art ein starkes Machtinstrument zur Verfügung. In den Medien wird regelmäßig über Datenschutzverletzungen und das Erbeuten von Informationen berichtet. Diese Informationen werden gesammelt und beispielsweise für personifizierte Werbung verwendet.
Ein immer größer werdender Teil der alltäglichen Aktivitäten wie z. B. Einkäufe wird heute über das Internet abgewickelt. Damit einhergehend erhöht sich auch der Anteil der preisgegebenen Informationen [10, S. 1]. Obwohl kritische Informationen wie Bank- und Kreditinformationen in der Regel ausreichend vor unbefugten Zugriffen geschützt werden, können vermeintlich unwichtige Informationen von jedem Webserver mühelos ausgelesen und gespeichert werden. Mit IP-Adressen, Cookies oder digitalen Fingerabdrücken ist es möglich, unbedenkliche Informationsschnipsel zu wertvollen Informationen zu kombinieren. Diese Fingerabdrücke werden von Peter Eckersley in der Publikation „How unique is your web browser“ [6] behandelt und sind Hauptgegenstand dieser Arbeit.
Digitale Fingerabdrücke werden aus Informationen des Nutzers generiert, welche unbemerkt und ohne explizite Zustimmung beim Besuch von Webseiten ausgelesen werden können. Mit Hilfe der Identifizierung durch Fingerabdrücke können Internetnutzer verfolgt, ihre Daten gesammelt und diese dann an Drittanbieter weitergereicht werden.
Es werden Möglichkeiten vorgestellt, digitale Fingerabdrücke zu generieren und aufgezeigt, wie eindeutig sich Nutzer im Internet identifizieren lassen, wie beständig digitale Fingerabdrücke sind und welche Gegenmaßnahmen man ergreifen kann um seinen persönlichen digitalen Fingerabdruck zu minimieren.

2. SPUREN IN DIGITALEN MEDIEN

Jeder Aufruf einer Webseite hinterlässt Spuren. Diese Informationen verletzen im Allgemeinen nicht die Privatsphäre des Nutzers. Sie können jedoch über längere Zeit hinweg gesammelt und gespeichert werden und anschließend dafür genutzt werden die Anonymität des Nutzers aufzuheben und die Privatsphäre des Nutzers zu gefährden. In den folgenden Abschnitten wird die Bedeutung von Anonymität erörtert, die Begriffe „Privacy Footprints“ und „Device Fingerprinting“ definiert und auf die Gefahren beider Themen eingegangen.

2.1 Was bedeutet Anonymität im Internet?

Sich anonym im Internet zu bewegen bedeutet, dass man keine Informationen hinterlässt, mit denen man identifiziert und verfolgt werden kann.
Doch warum ist Anonymität im Internet so wichtig? Auf den ersten Blick suggeriert das Verlangen nach Anonymität die Absicht, illegalen Aktivitäten nachzugehen. Anonymität im Internet ist jedoch für jeden Internetnutzer essentiell. Anonymität im Internet schützt die Privatsphäre einer Person. Diese sichert das Recht auf freie Entfaltung der Persönlichkeit, welches eines der Grundrechte im deutschen Grundgesetz beschreibt [13, Art 2. Abs. 1]. Des Weiteren schützt sie das vom Bundesverfassungsgericht erwähnte, im

doi: 10.2313/NET-2012-08-1_04

Grundgesetz jedoch nicht erwähnte „Recht auf informationelle Selbstbestimmung“ [2].
Sie verhindert, dass private Informationen jeglicher Art an Dritte weitergegeben werden, dass Interessen und Nutzungsverhalten aufgezeichnet werden und verhindert damit Konsequenzen wie finanziellen oder sogar physischen Schaden, Rufschädigung oder Diskriminierung [4].
Die im Internet verwendeten Technologien ermöglichen es auf unterschiedliche Art und Weise, Nutzer über ihre Informationen zu verfolgen und damit die Anonymität im Internet auszuhebeln. Die folgenden Abschnitte behandeln diese Problematik.

2.1.1 Privacy Footprints

Privacy Footprints kann man sich als Fußspuren oder Fährten im Internet vorstellen. Sie beschreiben, zu welchem Ausmaß scheinbar unabhängige Webseiten über sogenannte Aggregatorknoten gemeinsamen Zugriff auf nutzerbezogene Informationen erhalten. Ein größerer Footprint bedeutet, dass mehr Informationen auf verschiedenen Aggregatorknoten gespeichert sind. Der Austausch von Informationen wird meist über Cookies bewerkstelligt. Folgende Schritte zusammen mit der Abbildung 1 erläutern anhand eines Beispiels die Funktionsweise von Aggregatorknoten [10, S. 1]:

1. Der Nutzer besucht Seite A und schickt damit Daten X an Server A.
2. Server A speichert Daten X im Aggregatorknoten.
3. Parallel zu Schritt 1. und 2. speichert Server A einen Cookie mit der Adresse des Aggregatorknotens auf dem Computer des Nutzers.
4. Der Nutzer besucht anschließend Seite B und schickt damit Daten Y auf Server B.
5. Server B liest den zuvor gespeicherten Cookie vom Computer des Nutzers und kennt dadurch den Aggregatorknoten.
6. Server B vergleicht Daten Y mit Daten X des Aggregatorknotens und kann den Nutzer reidentifizieren.
7. Server B kann durch die Reidentifikation beispielsweise nutzerbezogene Werbung anzeigen.

Zwei einflussreiche und bekannte Vertreter dieser Aggregatorknoten sind **doubleclick.net** und **google-analytics.com**. In Veröffentlichung [10] wurde in einem Experiment festgestellt, dass unter 1075 unabhängigen Webseiten bei **doubleclick.net** insgesamt 201 (19%) Web-Seiten und bei **google-analytics.com** insgesamt 78 Web-Seiten (7%) über Aggregatorknoten miteinander verbunden waren.

2.1.2 Device Fingerprinting

Unter Device Fingerprinting versteht man die Generierung einer Identifikation (eines Fingerabdrucks) eines Betriebssystems (Software) oder einer Klasse von Geräten (Hardware), ohne die Kooperation der zu identifizierenden Geräte, bzw. Software [17, S. 1]. Dies ist mit unterschiedlichsten Geräten möglich. Es ist z. B. bereits länger bekannt, dass man

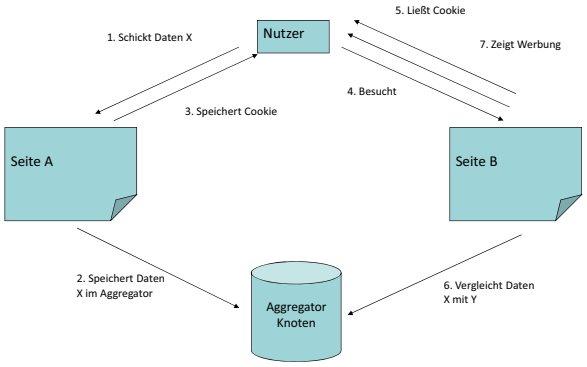

Abbildung 1: Ablauf beim Erzeugen von Privacy Footprints

Schreibmaschinen an den unterschiedlichen Ausfransungen der gedruckten Buchstaben unterscheiden kann [12]. Das Sensorrauschen in Bildern ermöglicht ebenfalls eine Identifizierung von Digitalkameras [8]. Neben der Identifizierung über Hardware können auch Fingerabdrücke mit Hilfe von Software generiert werden [6, S. 1]. Beispiele hierfür sind:

- CSS Font Detector [14]
- CSS History Hack [1]
- Fingerprinting mit dem Browser [6]

Mit den oben aufgeführten Hilfsmitteln lassen sich digitale Fingerabdrücke anfertigen, mit deren Hilfe man Internetnutzer identifizieren und verfolgen kann.

2.2 Gefahren von Privacy Footprints

Die größte Gefahr, die von Privacy Footprints ausgeht, ist die Reidentifikation eines Nutzers [10, S. 1]. Mit ihr können private, vermeintlich anonymisierte Daten mit harmlosen, jedoch personifizierten Daten kombiniert werden und damit persönliche Informationen offengelegt werden [15].
Folgendes Beispiel soll den Ablauf einer Reidentifikation veranschaulichen: Der erste Datensatz ist eine medizinische Akte, die lediglich das Geburtsdatum, das Geschlecht, eine geographische Ortsangabe und medizinische Befunde enthält (anonymer Datensatz). Dieser wird durch Seite A (siehe Grafik 1) repräsentiert und im Aggregatorknoten gespeichert. Der zweite Datensatz enthält Informationen über den Fahrzeughalter eines Kraftfahrzeugs (personifizierter Datensatz). Dieser wird durch Seite B repräsentiert und ebenfalls im Aggregatorknoten hinterlegt. Die Kombination beider Datensätze im Aggregatorknoten führt dazu, dass man die medizinische Akte einer Person namentlich zuordnen kann. Solche Offenlegungen von Informationen können zu finanziellen Schäden oder Rufschädigung der Person führen [10, S. 1].

2.3 Realisierungen von Device Fingerprinting

In den folgenden Abschnitten gehen wir auf die in 2.1.2 vorgestellten Fingerprinting-Beispiele etwas genauer ein.

doi: 10.2313/NET-2012-08-1_04

2.3.1 CSS Font Detector

Der CSS Font Detector ist JavaScript-Code, mit dem man – mit Hilfe von Cascading Style Sheets (CSS) – verfügbare Schriftarten in einem Browser identifizieren kann. Der eigentliche Zweck des Detektors ist es, besseres Design von Web-Seiten durch das Setzen von passenden Schriftarten zu ermöglichen. Der Algorithmus nutzt die Gegebenheit aus, dass jedes Zeichen eine unterschiedliche Pixelgröße in unterschiedlichen Schriftarten besitzt (siehe Abbildung 2).

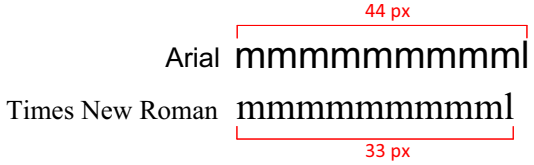

Abbildung 2: Pixellängen für unterschiedliche Schriftarten

Die Funktionsweise ist folgende: Es werden 3 Strings in Monospace, Sans-serif und Sans erzeugt und die Pixelbreite dieser Strings notiert. Anschließend wird ein String mit der zu testenden Schriftart als primäre Schriftart und einer der generischen Schriftarten als Fall-Back Schriftart erzeugt. Die Pixelbreite wird mit den generischen Schriftarten verglichen. Ist die Pixelbreite unterschiedlich, existiert die Schriftart, ist sie gleich, existiert die Schriftart nicht [14].

2.3.2 CSS History Hack

Der CSS History Hack erkennt, wie beim CSS Font Detector mit Hilfe von CSS, ob ein Nutzer bekannte Webseiten besucht hat oder nicht. Dabei besucht der Nutzer eine präparierte Webseite, auf der durch eine festgelegte Menge von Webseiten iteriert wird. Bei jeder Webseite wird überprüft, ob der Nutzer die Seite besucht hat, oder nicht. Der Algorithmus funktioniert folgendermaßen:

1. Es wird eine CSS Regel festgelegt, die einen besuchten Link auf eine festgelegte Farbe setzt. Beispiel:

```
<style>
        a:visited { color: red; }
</style>
```

2. Im nächsten Schritt werden mit JavaScript HTML-Link-Elemente aus einer selbst definierten Menge von Web-Seiten (`www.google.com`, `www.facebook.com`, etc.) erzeugt.

3. Anschließend iteriert man durch die generierten HTML Elemente und vergleicht die Farbe mit der in Schritt 1. festgelegten Farbe für besuchte Links. Entspricht die Farbe des Links der CCS Regel, hat der Nutzer die Web-Seite bereits besucht.

Die zu untersuchenden Elemente müssen nicht sichtbar in die Seite eingebaut werden und werden nach Abschluss des Tests wieder entfernt. Aus diesem Grund kann man den Test auch vom Nutzer unbemerkt durchführen. [1].

3. DEVICE FINGERPRINTING MIT DEM BROWSER

Der CSS Font Detector und der CSS History Hack sind zwei Möglichkeiten, Informationen über einen Nutzer im Internet zu erhalten. Kombiniert man diese mit weiteren Informationen, kann man ziemlich präzise digitale Fingerabdrücke eines Internetnutzers anfertigen. Web-Browser geben viele dieser Informationen ohne Kenntnis oder Zustimmung des Nutzers preis. In den folgenden Abschnitten erörtern wir anhand der Arbeit von Peter Eckersley, wie man mit Hilfe des Web-Browsers digitale Fingerabdrücke erzeugen kann, wie eindeutig diese Fingerabdrücke einen Nutzer identifizieren können, wie stabil diese Fingerabdrücke gegen Veränderungen sind und wie man sich gegen die Verfolgung über digitale Fingerabdrücke schützen kann.

3.1 Methodik

In den folgenden Abschnitten werden wir die Funktionsweise des Browser-Fingerprint-Algorithmus, die mathematischen Grundlagen, die hinter diesem Algorithmus stecken und die Aufbereitung der Datensätze analysieren.

3.1.1 Browser-Fingerprint-Algorithmus

Grundlage der Publikation von Peter Eckersley [6] ist ein selbst konstruierter Algorithmus für die Generierung von digitalen Fingerabdrücken, mit deren Hilfe über die Webseite `http://panopticlick.eff.org` bisher insgesamt 2.102.470 (Stand 25.03.2012) digitale Fingerabdrücke gesammelt wurden.
Beim Besuch der Seite werden zuerst anonymisiert die Konfiguration samt Versionsinformationen von Betriebssystem, Browser und Plugins gespeichert. Anschließend werden die Informationen mit den Datensätzen in der Datenbank verglichen. Im Ergebnis wird dem Nutzer angezeigt, wie eindeutig seine Browserkonfiguration innerhalb des Testdatensatzes ist.

3.1.2 Mathematische Grundlagen

Die mathematischen Grundlagen für die Ermittlung der Eindeutigkeit eines digitalen Fingerabdrucks sind Informationsgehalt und Entropie aus dem Bereich der Informationstheorie nach Claude E. Shannon. Die Identifizierbarkeit einer Browserkonfiguration hängt dabei direkt von der Auftrittswahrscheinlichkeit $P(m)$ einer Messvariablen innerhalb einer endlichen Menge von Messvariablen $m \in M$ ab. Je höher die Wahrscheinlichkeit des Auftretens einer Messvariable, desto geringer ist der Informationsgehalt, desto schwieriger lässt sie sich identifizieren. Umgekehrt sind Messvariablen mit geringer Auftrittswahrscheinlichkeit und dem daraus resultierenden hohen Informationsgehalt sehr leicht zu identifizieren. Dies ist durch die Verwendung von $-log_x(y)$ in den Formeln für Informationsgehalt und Entropie begründet. Informell kann man sagen: je mehr Browser-Konfigurationen eine Messvariable mit hohen Auftrittswahrscheinlichkeiten enthalten, desto schwieriger sind sie zu identifizieren. In den folgenden Abschnitten werden Informationsgehalt und Entropie noch etwas genauer betrachtet.
Formell betrachtet, ist der Informationsgehalt eines Versuchs die Menge an Information, die benötigt wird, um zu wissen, dass ein bestimmtes Ereignis x einer Zufallsvariablen X eingetreten ist [9, S. 23]. In Eckersleys Arbeit gibt die Informationsmenge Auskunft über die Identität eines einzelnen

doi: 10.2313/NET-2012-08-1_04

Web-Browsers $x \in X$. Formel 1 beschreibt den Informationsgehalt für eine Messvariable $P(f_n)$ und für einen Web-Browser $\{F(x) = f_n\}$:

$$I(F(x) = f_n) = -\log_2(P(f_n)) \quad (1)$$

Die Basis für die Informationsmenge ist 2, da die einzelnen Experimente jeweils nur den Ausgang „wahr“ oder „falsch“ haben können (Beispielfrage: Sind Cookies aktiviert?). Die Informationseinheit für Logarithmen zur Basis zwei ist `bit`. Die Entropie liefert den maximalen Informationsgehalt für alle x Ereignisse einer Zufallsvariablen X [9]. In Eckersleys Arbeit ist damit der Informationsgehalt einer Messvariable über alle Browser X gemeint. Formel 2 beschreibt die Entropie für eine Messvariable $P(f_n)$ und für die Menge aller untersuchten Browser:

$$H(F) = -\sum_{n=0}^{N} P(f_n) log_2(P(f_n)) \quad (2)$$

Möchte man den Informationsgehalt von mehr als einer Messvariablen $s \in S$ berechnen, muss man unterscheiden, ob die Messvariablen von einander abhängig sind oder nicht. Im Falle der Unabhängigkeit der Messvariablen wird Formel 3 angewendet:

$$I_s(f_n, s) = -log_2 P(f_n, s) \quad (3)$$

Sind die Messvariablen hingegen voneinander abhängig, wird Formel 4 angewendet:

$$I_{s+t}(f_{n,s}, f_{n,t}) = -log_2(P(f_n, s|f_n, t)) \quad (4)$$

Ein Beispiel für statistisch abhängige Messvariablen wäre die Identifikation eines Flash Block Plugins mit P(Schriftart = „nicht gefunden“ | „Flash“ $\in$ Plugins). Die Entropie für mehrere Messvariablen kann mit Formel 5 berechnet werden:

$$H_s(F_s) = -\sum_{n=0}^{N} P(f_{s,n}) log_2(P(f_{s,n})) \quad (5)$$

Es ist nun bekannt, wie man den Informationsgehalt und die Entropie von Web-Browser-Konfigurationen berechnen kann. Als Nächstes muss geklärt werden, wie die ermittelten Kennzahlen zu interpretieren sind. Man muss herausfinden, wie viel bits an Entropie benötigt werden um einen einzelnen Nutzer (bzw. seinen Web-Browser) im Datensatz eindeutig identifizieren zu können. Dazu wird zuerst die Auftrittswahrscheinlichkeit $P(h)$ einer einzelnen Konfiguration innerhalb des Datensatzes benötigt. Im Datensatz von Panopticlick mit 2.102.470 Einträgen (Stand 25.03.2012) beträgt die Auftrittswahrscheinlichkeit eines Datensatzes: $P(h) = 1/2.102.470$. Der Informationsgehalt beträgt dann:

$$S = -log_2(P(h)) = -log_2(1/2.102.470) = 21.003654\ bits \quad (6)$$

Aufgerundet bedeutet dies, dass ein Nutzer mit einem Fingerabdruck mit 22 bits Informationsgehalt eindeutig im Datensatz von Panopticlick identifizierbar ist. Wendet man diese Erkenntnis auf die Population der Erde mit ca. 7.039.632.000 Menschen (Stand 2012 [16]) aus, erhält man eine Auftrittswahrscheinlichkeit von $P(h) = 1/7.039.632.000$ und damit einen Informationsgehalt von:

$$S = -log_2(P(h)) = -log_2(1/7.039.632.000) = 32,71\ bits \quad (7)$$

Aufgerundet benötigt man also 33 bits an Entropie um eine Person auf der Erde eindeutig zu identifizieren [5].

3.1.3 Datensammlung und Vorbearbeitung

Die Datensätze, auf die sich die Publikation von Peter Eckersley stützt, wurden im Zeitraum 27.01.2010 – 15.02.2010 über die Webseite `http://panopticlick.eff.org` gesammelt. Es wurden folgende Daten erhoben:

- Ein Browser-Fingerprint
- Eine HTTP-Cookie ID, welche 3 Monate gespeichert wurde
- Ein HMAC-Wert der IP-Adresse
- Ein HMAC-Wert der IP-Adresse mit gelöschtem letzten Oktet (Subnetzadresse)

Die neben dem Fingerabdruck gespeicherte HMAC (Keyed-Hash Message Authentication Code) der IP-Adresse, die HMAC des Subnetzes und die Cookie ID dienten der Verfolgung von Besuchern, welche die Seite mehr als einmal besucht hatten. Die HMACs werden dazu verwendet die IP-Adresse und die Adresse des Subnetzes zu anonymisieren. Die HMAC wird dabei aus einer Nachricht (in dem Fall die IP-Adresse) und einem privaten Schlüssel nach einer festgelegten Hashfunktion berechnet. Das Hashing ohne Schlüssel würde bei der IPv4-Adress-Länge von 2^{32} Bit keine ausreichende Sicherheit bieten [7].
Der Datensatz wurde vor den Messungen auf Doppeleinträge untersucht und bereinigt. Bei Besuchern mit aktivierten Cookies konnten doppelte Datensätze leicht erkannt und entfernt werden. Bei Besuchern mit deaktivierten Cookies wurde angenommen, dass Datensätze mit gleicher IP-Adresse und identischem Fingerabdruck den selben Browser repräsentieren und wurden als Folge dessen auf einen Eintrag reduziert. Bei letzteren gab es jedoch auch eine Ausnahme. Es wurden im Laufe des Experiments identische (Fingerabdruck, IP) Tupel mit unterschiedlichen Cookies registriert. Dies bedeutet, dass ein Besucher mit der gleichen IP-Adresse wiederholt `http://panopticlick.eff.org` mit Cookie A, dann mit Cookie B und anschließend wieder mit Cookie A besucht hatte. Diese Charakteristik wurde bei 3.5% aller IP-Adressen festgestellt.
Zu Beginn der Datensammlung wurden außerdem noch einige Datensätze auf Grund von Fehlern im Algorithmus entfernt.
Aus anfänglich 1.043.426 Datensätzen wurden 470.161 Fingerabdrücke mit minimalen Doppeleinträgen für die Messungen extrahiert [6, S. 7–8].

3.1.4 Erhobene Daten

In Tabelle 1 sind alle Messvariablen samt Quelle aufgelistet, die mit dem Browser-Fingerprint-Algorithmus gesammelt wurden. Alle Informationen werden über den Browser und ohne Zutun des Nutzers den aufgerufenen Webseiten zur Verfügung gestellt.
Die Datensätze könnten um zusätzliche Informationen erweitert werden, wurden jedoch in dem Algorithmus aus folgenden Gründen nicht berücksichtigt [6, S. 7]:

doi: 10.2313/NET-2012-08-1_04

Tabelle 1: Aufschlüsselung der erhobenen Daten [6, S. 5]

Variable	Quelle
User Agent	Über HTTP übertragen, vom Server aufgezeichnet
HTTP Accept Header	Über HTTP übertragen, vom Server aufgezeichnet
Cookies aktiviert?	Aus HTTP abgeleitet, vom Server aufgezeichnet
Bildschirm Auflösung	JavaScript AJAX Post
Zeitzone	JavaScript AJAX Post
Browser Plugins, Plugin Versionen und MIME Typen	JavaScript AJAX Post
System Schriftarten	Flash oder Java Applet, mit JavaScript/AJAX ausgelesen
Supercookie Test	JavaScript AJAX Post

1. Mangelnde Kenntnis oder Mangel an Zeit zur korrekten Implementation (Bsp.: Microsoft Active X)
2. Die Informationen wurden als nicht ausreichend stabil erachtet
3. Die Information kann nur durch explizite Zustimmung des Nutzers erlangt werden

3.2 Ergebnisse

In dem von Peter Eckersley erhobenen Datensatz sind 83,6% der Fingerabdrücke eindeutig identifizierbar, 8,2% teilen sich mit 2 bis 9 Fingerabdrücken die gleiche Konfiguration und 8,1% teilen sich mit 10 Fingerabdrücken die gleiche Konfiguration. Nur ein Bruchteil von unter 0,1% befindet sich in einer Konfigurationsmenge größer 10 (siehe Abbildung 3) [6, S. 9]. Diese Zahlen sind laut Eckersley etwas verfälscht, da er die Hauptnutzer von `http://panopticlick.eff.org` als technisch versiert und sicherheitsbewusst ansieht. Für realistischere Ergebnisse mit durchschnittlichen Nutzern würde die Eindeutigkeit der Fingerabdrücke geringer ausfallen [6, S. 7].

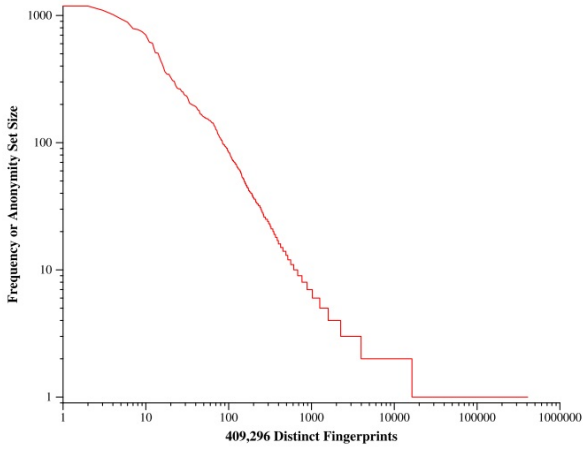

Abbildung 3: Verteilungsfunktion der ermittelten Fingerabdrücke [6, S. 8]

In der Verteilungsfunktion des Informationsgehalts für verschiedene Browser in Abbildung 4 ist leicht zu erkennen, dass der Großteil der Web-Browser in Bezug auf digitale Fingerabdrücke schlecht abschneidet. 90% der modernen Webbrowser sind eindeutig identifizierbar. Gut abgeschnitten haben Web-Browser, bei denen JavaScript deaktiviert ist oder Web-Browser von mobilen Geräten wie iPhone oder Android. Web-Browser von mobilen Geräten sind durch limitierte Pluginfähigkeiten viel einheitlicher und deshalb schwer zu unterscheiden. Ihr Nachteil sind jedoch die mangelnde Kontrolle über angelegte Cookies, welcher die Verfolgung von Nutzern wiederum erleichtert [6, S. 9].

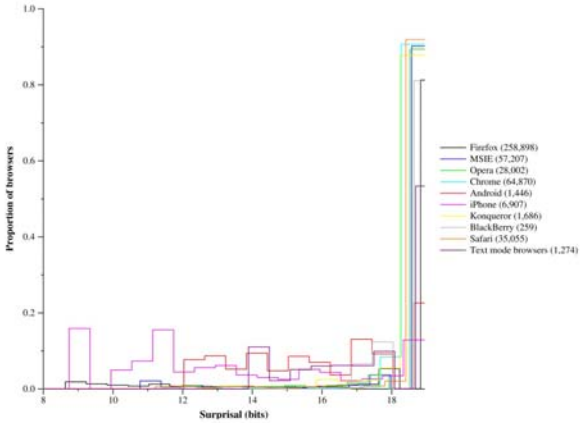

Abbildung 4: Informationsgehalt für verschiedene Web-Browser [6, S. 9]

Eckersley berechnete auch die Fingerabdrücke der Web-Browser für jede einzelne Messvariable (siehe Tabelle 2). Den höchsten Grad an Identifizierbarkeit zeigen Plugins und Schriftarten, gefolgt von User-Agent-Informationen (String mit Informationen über Betriebssystem, Browser, und deren Versionen), HTTP Accept Header (nennt dem Browser den Medientyp des HTTP-Response), Bildschirmauflösung, Zeitzone, Supercookies (Cookies für den Adobe Flash Player [3]) und Cookies.

3.2.1 Fingerabdruck-Charakteristiken

Neben den direkt ablesbaren Charakteristiken zum Identifizieren von Nutzern hat Peter Eckersley noch weitere, subtilere Charakteristiken entdeckt, mit denen sich Nutzer leicht identifizieren lassen. Ein Beispiel ist die JavaScript-Konfiguration eines Browsers. Bei deaktiviertem JavaScript werden Standardwerte für Video, Plugins, Schriftarten und Supercookies festgelegt. Die Präsenz dieser Werte zeigt, dass JavaScript deaktiviert ist.
Ein weiteres Beispiel sind Browser, die Flash in den Plugins auflisten, bei denen es jedoch nicht möglich ist, die Systemschriftarten auszulesen (System-Schriftarten werden im Browser-Fingerprint-Algorithmus mit Hilfe von Flash aus-

doi: 10.2313/NET-2012-08-1_04

Tabelle 2: Durchschnittliche Entropie einzelner Variablen [6, S. 17]

Variable	Durchschn. Eigeninformation
User-Agent	10,0
HTTP-Accept-Header	6,09
Cookies aktiviert?	0,353
Bildschirm-Auflösung	4,83
Zeitzone	3,04
Browser Plugins, Plugin Versionen und MIME-Typen	15,4
System Schriftarten	13,9
Supercookie verfügbar	2,12

gelesen). Diese Charakteristik ist ein eindeutiges Indiz für die Nutzung eines Flash Blockers und verstärkt die Eindeutigkeit von Fingerabdrücken.
Das letzte Beispiel handelt von User-Agent-Spoofing-Plugins, mit denen sich User-Agent-Informationen wie z. B. Betriebssystem oder Browser verfälschen lassen. In der Untersuchung wurden Datensätze entdeckt, welche sich als iPhone ausgaben, jedoch auch das Flash-Plugin installiert hatten (iOS unterstützt zur Zeit kein Flash). Die geringe Anzahl der Fingerabdrücke, die diese Charakterstiken aufzeigen, verstärkt die Identifizierbarkeit [6, S. 4].

3.2.2 Globale Extrapolation

Peter Eckersley hat in seiner Publikation auch die Frage behandelt, ob sich der über `http://panopticlick.eff.org` erhobene Datensatz auf globaler Ebene extrapolieren lässt.
Er geht auf die Aussage von Mayer [11] ein, dass sich mit einer Stichprobe auf Grund der Multinomialverteilung keine Aussagen über die globale Eindeutigkeit von Web-Browser-Fingerabdrücken machen lassen. Er unterstützt Mayers These, behauptet jedoch, dass sie im Bereich Privatsphäre etwas zu optimistisch ausgelegt sei. Eckersley geht davon aus, dass man mit einer geeigneten Stichprobe von Fingerabdrücken und Monte-Carlo-Simulation mindestens ein globales Mengenverhältnis von Eindeutigkeiten ermitteln könnte.
Dieser Ansatz ist jedoch für eine globale Extrapolation mit dem Panopticlick Datensatz nicht umsetzbar, weil er hauptsächlich aus Datensätzen von technisch versierten, sicherheitsbewussten Internetnutzern besteht. Diese würden ein verzerrtes Ergebis globaler Fingerabdrücke wiedergeben. Der Datensatz müsste um neutrale Einträge ausgeweitet werden, um ein realistisches Ergebnis zu liefern [6, S. 10–11].

3.3 Stabilität von Fingerabdrücken

Die Mitverfolgung von wiederkehrenden Besuchern von `http://panopticlick.eff.org` (siehe 3.1.3) zeigt, dass sich digitale Fingerabdrücke schnell ändern können. Diese Veränderungen werden z. B. durch Updates des Browsers, Updates der Plugins, Deaktivieren von Cookies, Installation von neuen Fonts, oder durch das Anschließen eines externen Monitors (Veränderung der Auflösung) hervorgerufen. Veränderte Fingerabdrücke eines Nutzers werden mit Hilfe der gespeicherten HTTP-Cookie ID (siehe 3.1.3) und einem einfachen Algorithmus erkannt. Der Algorithmus erkannte Veränderungen bei 65% aller Fälle, lag bei 0,56% falsch und war durch zu große Umstellungen der Browser-Konfiguration bei 35% aller Fälle nicht in der Lage, eine Veränderung zu erkennen. Insgesamt 37,4% der wiederkehrenden Besucher mit aktivierten Cookies besaßen einen veränderten Fingerabdruck. Die Änderungsrate bei `http://panopticlick.eff.org` fällt laut Eckersley etwas höher aus als in der realen Welt, da das Experiment die Besucher zum Ändern ihrer Konfigurationen animiert. Dieses Experiment zeigt, dass Veränderungen der Konfiguration nicht zuverlässig vor Identifikation oder Verfolgung schützen [6, S. 11–13].

3.4 Mögliche Schutzmaßnahmen

in diesem Abschnitt werden Möglichkeiten aufgezeigt, wie man sich gegen das Erzeugen von digitalen Fingerabdrücken schützen kann. Es ist anzumerken, dass man immer einen Kompromiss aus Schutz und Bequemlichkeit / Nutzbarkeit eingehen muss. Je mehr Schutzfunktionalitäten beim Surfen im Internet verwendet werden, desto langsamer werden die Internetseiten aufgebaut und desto mehr Nutzerinteraktion ist für das Freischalten von Inhalten notwendig. Teilweise können Inhalte nicht korrekt bzw. teilweise oder überhaupt nicht dargestellt werden. Man kann dies sehr leicht feststellen, indem man z. B. `http://www.facebook.com/` mit deaktiviertem JavaScript aufruft.
Darüber hinaus ist noch anzumerken, dass nicht alle Schutzmaßnahmen auch zu dem gewünschten Ergebnis führen, nicht identifizierbar zu sein. Es existieren Produkte, die Informationen eines Nutzers im Internet verschleiern, welche das Identifizieren eines Nutzers durch ihre geringe Verbreitung sogar erleichtern. Kontraproduktive Verschleierungstechniken haben wir mit Flash Blocker und User Agent Spoofing bereits in Kapitel 3.2.1 beschrieben. Beispiele für Produkte mit kontraproduktiven Verschleierungstechniken sind der Privoxy Web Proxy (`http://www.privoxy.org`), welcher in Eckersleys Untersuchungen durchschnittlich 15.5 bits Entropie aufwies und der Privacy-Browser Browzar (`http://www.browzar.com/`), bei dem alle Datensätze eindeutig identifizierbar waren.
Es existieren jedoch auch Lösungen, die den digitalen Fingerabdruck im Internet tatsächlich minimieren. Zwei Werkzeuge, die das Erzeugen von Fingerabdrücken im Experiment erheblich erschwert haben waren einerseits das Anonymisierungsnetzwerk TOR[1] und das NoScript[2] Plugin für den Firefox Browser. Mit Hilfe des Tor Projektes kann effektiv das Hinterlassen von Spuren im Internet verringert werden. Das NoScript Plugin verhindert das Ausführen von JavaScript, Java und anderen Plugins auf besuchten Seiten. Der Nutzer muss bei jeder Webseite explizit die Ausführung von Skripten erlauben.
Neben den in Browsern verfügbaren Schutzmaßnahmen führt Eckersley noch Funktionalitäten in Browsern auf, die große Teile von Systeminformationen freilegen und damit das Erzeugen von Fingerabdrücken stark vereinfachen. Diese Funktionalitäten lassen sich zur Zeit nicht durch den Nutzer beeinflussen oder deaktivieren. Dazu gehören z. B. das Auflisten von Systemschriftarten (mit Ausnahme der kompletten Deaktivierung des Flash Plugins) oder Plugins samt Versionsinformationen. Die hohe Aussagekraft von Plugins ist

[1] `https://www.torproject.org/`
[2] `https://addons.mozilla.org/de/firefox/addon/noscript/`

doi: 10.2313/NET-2012-08-1_04

mit einer durchschnittlichen Entropie von 15,4 bits und von Schriftarten mit einer durchschnittlichen Entropie von 13,9 bits (siehe Tabelle 2) im Gegensatz zu den restlichen Charakteristika deutlich zu erkennen. Der eigentliche Nutzen der API Methoden zum Auflisten von installierten Plugins samt Versionshinweisen liegt in der Vereinfachung der Software-Entwicklung. Die Auflistungen sind deshalb lediglich Software-Entwicklern von Vorteil und sind für den Endnutzer von keinerlei Interesse. Laut Peter Eckersley sollten System-Schriftarten und Plugins nur über Zustimmung des Nutzers aufgelistet werden.
Ein weiteres Problem existiert mit den detaillierten Versionsangaben in User Agent Strings. Hier werden oft Mikroversionen angegeben (Java 1.6.0_17 statt Java 1.6). Diese detaillierten Versionsangaben führen zu hohen Entropien und damit zu besseren Fingerabdrücken. Die Mikroversionen werden von Softwareentwicklern für Fehleranalysen genutzt. Hier sollte der Schwerpunkt weg von der Annehmlichkeit der Entwickler in Richtung Schutz der Privatsphäre des Nutzers verlagert werden, indem man lediglich Hauptversionen im User Agent String anführt.
Eine weitere, die Entropie steigernde Gegebenheit ist, dass Plugin- und Schriftart-Auflistungen unsortiert zurückgegeben werden (Peter Eckersley hat in seinen Experimenten nicht den CSS Font Detector (Kapitel 2.3.1) verwendet, bei dem die Reihenfolge der zu testenden Schriftarten selbst definiert wird). Dadurch können Nutzer mit gleicher Konfiguration unterschieden und damit auch identifiziert werden. Eine einfache Sortierung der Auflistungen würde dieses Problem beseitigen [6, S. 13–15].

4. ZUSAMMENFASSUNG

Peter Eckersley zeigt exemplarisch, wie man Internet-Nutzer anhand der Informationen ihres Web-Browsers identifizieren kann. Der entwickelte Web-Browser-FingerprintAlgorithmus konnte einen Großteil der Web-Browser-Konfigurationen im Testdatensatz eindeutig identifizieren. Sehr gut ist auch, dass nicht nur die einzelnen Fingerabdrücke an sich, sondern auch das Reidentifizieren von veränderten Fingerabdrücken behandelt wird.
Die Wahl der Messvariablen im Algorithmus (siehe Tabelle 1) brachte ausreichend hohe Entropiewerte um Browser-Konfigurationen im Testdatensatz in über 83,6% der Fälle eindeutig zu identifizieren. Die Kennzahlen lassen sich jedoch aus den in Kapitel 3.2.2 beschriebenen Gründen nicht auf die globale Ebene extrapolieren. Eine Möglichkeit die Extrapolation der Daten zu ermöglichen, wäre die Messvariablen um weitere Informationen wie Active-X, Microsoft Silverlight, Adobe Flex, oder JavaFX Komponenten zu erweitern und damit die Entropiewerte nochmals zu erhöhen. Eine weitere Möglichkeit wäre die Sammlung von Fingerabdrücken von weniger voreingenommenen, technisch unversierteren Internet-Nutzern.
Eckersley zeigt neben der Problemstellung auch unterschiedliche und mehr oder weniger effektive Möglichkeiten auf, sich gegen das Erheben von digitalen Fingerabdrücken zu schützen. Er zeigt darüber hinaus auch auf, dass es in diesem Bereich noch einigen Spielraum für Verbesserungen gibt.
Abschließend ist zu sagen, dass digitalen Fingerabdrücken in Zukunft im Bereich Privatsphäre die gleiche Relevanz zugeschrieben werden muss, wie es bereits bei Cookies und IP-Adressen der Fall ist.

5. LITERATUR

[1] What the internet knows about you. *http://whattheinternetknowsaboutyou.com*.

[2] BVerfGE, Urteil des Ersten Senats, 65, 1. *http://sorminiserv.unibe.ch:8080/tools/ainfo.exe?Command=ShowPrintText&Name=bv065001*, Dezember 1983.

[3] Website-Speichereinstellungen. *http://www.macromedia.com/support/documentation/de/flashplayer/help/settings_manager07.html*, März 2012.

[4] J. Appelbaum. The Tor Project. *https://www.torproject.org/about/overview.html.en#whyweneedtor*.

[5] P. Eckersley. A Primer on Information Theory and Privacy. *https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy*, Januar 2010.

[6] P. Eckersley. How unique is your web browser? pages 1–18, LNCS 6205/2010, 2010. Proc. 10th Privacy Enhancing Technologies Symposium (PETS2010).

[7] R. C. H. Krawczyk, M. Bellare. HMAC: Keyed-Hashing for Message Authentication. Februar 1997.

[8] Jan Lukás, Jessica Fridrich, and Miroslav Goljan. Digital Camera Identification from Sensor Pattern Noise. *IEEE Transactions on Information Forensics and Security 1*, 2, 2006.

[9] R. Johannesson. *Informationstheorie – Grundlagen der (Tele-)Kommunikation*. Addison-Wesley Publishing Company, 1992.

[10] B. Krishnamurthy. Generating a Privacy Footprint on the Internet. *Proc. 6th ACM SIGCOMM Conf. on Internet Measurement (IMC'06)*, pages 1–6. ACM, Oktober 2006.

[11] J. R. Mayer. Any person... a pamphleteer - internet anonymity in the age of web 2.0. *Undergraduate Senior Thesis, Princeton University*, 2009.

[12] Ordway Hilton. The Complexities of Identifying the Modern Typewriter. *Journal of Forensic Sciences 17*, 2, 1972.

[13] Parlamentarischer Rat. *Grundgesetz für die Bundesrepublik Deutschland*. Bonn, 1949. Stand: September 2010.

[14] L. Patel. JavaScript/CSS Font Detector. *http://www.lalit.org/lab/javascript-css-font-detect/*, März 2007.

[15] S. Schoen. What Information is Personally Identifiable? *https://www.eff.org/deeplinks/2009/09/what-information-personally-identifiable*, September 2009.

[16] Stiftung Weltbevölkerung. Die Weltbevölkerungsuhr. *http://www.weltbevoelkerung.de/oberes-menue/publikationen-downloads/zu-unseren-themen/weltbevoelkerungsuhr.html*, 2012.

[17] A. B. Tadayoshi Kohno and K. Claffy. Remote Physical Device Fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, Juni 2005.

doi: 10.2313/NET-2012-08-1_04