Browser Fingerprinting

Jonathan Bräuer

Für das Seminar: Electronic Identity

Browser Fingerprinting

1. Entstehung und Verwendung

2. Browser Fingerprinting Techniken
   - Protokoll Informationen
   - Browser Metadaten
   - Komplexere Methoden
   - Übersicht

3. Schutzmaßnahmen

4. Beispiel und Feldversuch (Henning Tillman, 2013)

(5. Rechtlicher Rahmen)

1. Tracking im WWW

- Mit Tracking im WWW ist gemeint, zwei unabhängige Verbindungen demselben Nutzer zuzuordnen

- Der Zustand der letzten Sitzung kann wiederhergestellt werden (Warenkorb, bereits besuchte Artikel, Interessen, ...)

- Surfverhalten kann aufgezeichnet werden, über verschiedene Services hinweg (Interessen können an andere Services weitergegeben werden)

1. Gründe für Tracking

- Komfort

- Marktforschung

- Personalisierte Werbung

- Und anderes ?

1. Reguläre Trackingverfahren

- HTTP Protokoll beinhaltet dafür „Cookies“

- Andere verwendete Verfahren:

  - „Flashcookies“
    Benötigt Flash-Plugin, aber schwerer kontrollierbar

  - Local-Storage
    Mehr Speicherplatz als „Cookies“, aber weniger Verbreitet

  - ...

1. Cookie Probleme

- Nicht immer Verfügbar (können Deaktiviert werden)

- keine echte Persistenz (Ablaufdatum, veränderbar, löschbar)

- nicht Software- und Geräteübergreifend

→ gibt es da nicht was besseres?

1. Alternative zu „Cookies“

- Es wird also Alternative gesucht, die zuverlässiger ist

- Ansatzpunkt:

→ Erkennung/Speicherung verfügbarer Metadaten über Nutzer

→ Sehr ähnliche Metadaten implizieren denselben Nutzer

=> Browser Fingerprinting

1. Verfahren

- Metadaten von Verbindung zum Server und aus dem Browser sammeln

- Metadaten aus dem Browser zum Server übertragen

- Fingerprint erzeugen

- Abgleich mit bereits existierenden Fingerprints

- Leicht veränderte Fingerprints erkennen und verknüpfen

2. Protokoll Metadaten

- HTTP bzw. HTTPS Protokoll legt einige Metadaten direkt offen, bzw. ermöglicht Tracking durch „falsche“ Verwendung einiger Protokollfunktionen:

  - SSL/TLS Client-Hello Metadaten

  - IP Adresse

  - Request-Header Metadaten

  - Canvas Caching

  - ETAG's

2. Protokoll: SSL/TLS Handshake

- Beim SSL/TLS Client-Hello werden Informationen mitgeschickt, die zum korrekten Aufbau der sicheren Verbindung benötigt werden:

  - SSL/TLS Version

  - Client Zeitstempel (Systemzeit)

  - Liste unterstützter/bevorzugter Cipher-Suites

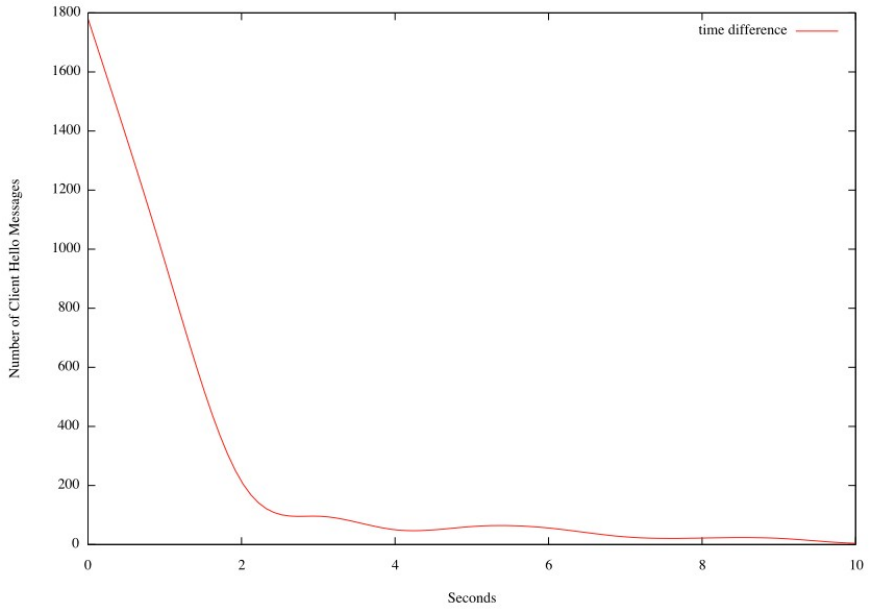
  - Unterstütze Extensions (Kompression....)

→ Goldgrube für einen Fingerprint

2. Protokoll: SSL/TLS Handshake

- Version, Liste der Ciphersuites und Extensions abhängig von Hardware und Software

- Zeitstempel kann abweichen



- Problem: Sichere Konfiguration führt zu eindeutigerem Fingerprint (blacklisting schwacher Algorithmen...)

Quelle: https://isc.sans.edu/forums/diary/Browser+Fingerprinting+via+SSL+Client+Hello+Messages/17210

2. Protokoll: SSL/TLS Handshake

Vergleich zweier Konfigurationen:

Cipher Suites Supported by Your Browser (ordered by preference):

Spec	Cipher Suite Name	Key Size	Description
(cc,14)	ECDHE-ECDSA-CHACHA20-POLY1305-SHA256	128 Bit	Key exchange: ECDH, e
(cc,13)	ECDHE-RSA-CHACHA20-POLY1305-SHA256	128 Bit	Key exchange: ECDH, e
(c0,2b)	ECDHE-ECDSA-AES128-GCM-SHA256	128 Bit	Key exchange: ECDH, e
(c0,2f)	ECDHE-RSA-AES128-GCM-SHA256	128 Bit	Key exchange: ECDH, e
(00,9e)	DHE-RSA-AES128-GCM-SHA256	128 Bit	Key exchange: DH, encr
(c0,0a)	ECDHE-ECDSA-AES256-SHA	256 Bit	Key exchange: ECDH, e
(c0,09)	ECDHE-ECDSA-AES128-SHA	128 Bit	Key exchange: ECDH, e
(c0,13)	ECDHE-RSA-AES128-SHA	128 Bit	Key exchange: ECDH, e
(c0,14)	ECDHE-RSA-AES256-SHA	256 Bit	Key exchange: ECDH, e
(c0,07)	ECDHE-ECDSA-RC4128-SHA	128 Bit	Key exchange: ECDH, e
(c0,11)	ECDHE-RSA-RC4128-SHA	128 Bit	Key exchange: ECDH, e
(00,33)	DHE-RSA-AES128-SHA	128 Bit	Key exchange: DH, encr
(00,32)	DHE-DSS-AES128-SHA	128 Bit	Key exchange: DH, encr
(00,39)	DHE-RSA-AES256-SHA	256 Bit	Key exchange: DH, encr
(00,9c)	RSA-AES128-GCM-SHA256	128 Bit	Key exchange: RSA, enc
(00,2f)	RSA-AES128-SHA	128 Bit	Key exchange: RSA, enc
(00,35)	RSA-AES256-SHA	256 Bit	Key exchange: RSA, enc
(00,0a)	RSA-3DES-EDE-SHA	168 Bit	Key exchange: RSA, enc
(00,05)	RSA-RC4128-SHA	128 Bit	Key exchange: RSA, enc
(00,04)	RSA-RC4128-MD5	128 Bit	Key exchange: RSA, enc

Further information:

User-Agent:	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (K Chrome/38.0.2125.111 Safari/537.36
Preferred SSL/TLS version:	TLSv1
SNI information:	cc.dcsec.uni-hannover.de
SSL stack current time:	The TLS stack of your browser did not send a time value.

This connection uses TLSv1.2 with ECDHE-RSA-AES128-GCM-SHA256 and a 128 Bit key for encryp

Cipher Suites Supported by Your Browser (ordered by preference):

Spec	Cipher Suite Name	Key Size	Description
(c0,2b)	ECDHE-ECDSA-AES128-GCM-SHA256	128 Bit	Key exchange: ECDH, encryp
(c0,2f)	ECDHE-RSA-AES128-GCM-SHA256	128 Bit	Key exchange: ECDH, encryp
(00,9e)	DHE-RSA-AES128-GCM-SHA256	128 Bit	Key exchange: DH, encryptio
(cc,14)	ECDHE-ECDSA-CHACHA20-POLY1305-SHA256	128 Bit	Key exchange: ECDH, encryp
(cc,13)	ECDHE-RSA-CHACHA20-POLY1305-SHA256	128 Bit	Key exchange: ECDH, encryp
(c0,0a)	ECDHE-ECDSA-AES256-SHA	256 Bit	Key exchange: ECDH, encryp
(c0,09)	ECDHE-ECDSA-AES128-SHA	128 Bit	Key exchange: ECDH, encryp
(c0,13)	ECDHE-RSA-AES128-SHA	128 Bit	Key exchange: ECDH, encryp
(c0,14)	ECDHE-RSA-AES256-SHA	256 Bit	Key exchange: ECDH, encryp
(c0,07)	ECDHE-ECDSA-RC4128-SHA	128 Bit	Key exchange: ECDH, encryp
(c0,11)	ECDHE-RSA-RC4128-SHA	128 Bit	Key exchange: ECDH, encryp
(00,33)	DHE-RSA-AES128-SHA	128 Bit	Key exchange: DH, encryptio
(00,32)	DHE-DSS-AES128-SHA	128 Bit	Key exchange: DH, encryptio
(00,39)	DHE-RSA-AES256-SHA	256 Bit	Key exchange: DH, encryptio
(00,9c)	RSA-AES128-GCM-SHA256	128 Bit	Key exchange: RSA, encryptio
(00,2f)	RSA-AES128-SHA	128 Bit	Key exchange: RSA, encryptio
(00,35)	RSA-AES256-SHA	256 Bit	Key exchange: RSA, encryptio
(00,0a)	RSA-3DES-EDE-SHA	168 Bit	Key exchange: RSA, encryptio
(00,05)	RSA-RC4128-SHA	128 Bit	Key exchange: RSA, encryptio
(00,04)	RSA-RC4128-MD5	128 Bit	Key exchange: RSA, encryptio

Further information:

User-Agent:	Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTM Chrome/38.0.2125.111 Safari/537.36
Preferred SSL/TLS version:	TLSv1
SNI information:	cc.dcsec.uni-hannover.de
SSL stack current time:	Fri, 09 Nov 2035 02:26:12

This connection uses TLSv1.2 with ECDHE-RSA-AES128-GCM-SHA256 and a 128 Bit key for encryption.

2. IP Adressen:

Use IP Address Data to Detect a Hosting Provider

To show where and how your customers connect to the Internet, Neustar IP Intelligence provides over 30 data attributes for each IP address. We're pleased to announce an addition to the "how" category: a new data field revealing the presence of a computing or Internet hosting service provider. Augmenting the existing network fields, hosting facility detection is another valuable tool in distinguishing real customers from cyber-criminals.

Where are they from?	How did they reach you?	What business or industry?
• Area code • Postal code • City and city confidence factor • State and state confidence factor • Region • Country and country confidence factor • Continent • Defined market area (DMA) • Metropolitan statistical area (MSA) • Latitude/longitude • Time zone • GeoNames identifier	• Proxy detection (level, type and date last detected) • Hosting facility • Autonomous system number (ASN) • Carrier/ISP • Connection speed • Connection type • IP routing type • Registering organization • Second-level domain • Top-level domain	• Home or business indicator • ISIC code • NAICS code • Organization type

2. HTTP-Header Metadaten

- Abhängig von Betriebssystem und Browser werden jeder Anfrage unterschiedliche Header Informationen angehängt:

  - Referer: Welcher Seite startete die Anfrage

  - User-Agent: Eindeutige Browser (-versions) Bezeichnung

  - Accept: Akzeptierte Datentypen
  - Accept-Language: Akzeptierte Sprache
  - Accept-Encoding: Akzeptierte Komprimierung
  - Accept-Charset: Akzeptierte Zeichenkodierung

  - (In manchen Fällen mehr, wie Cache Informationen...)

2. Caching: Canvas

- Moderne Browser unterstützen Caching Funktionen, um das surfen zu beschleunigen

- Dies kann ausgenützt werden, um einzigartige Bilder im Cache eines Browser zu speichern:

- Ablauf:
  - Bild mit eindeutigem Hash wird in verstecktem Canvas geladen
  - Noch nicht im Cache: Bild wird geladen und gespeichert
  - Bereits im Cache: Bild wird aus Cache geladen und an den Server übermittelt

→ Wird ein bekanntes Bild an Server übermittelt, kann Verbindung zugeordnet werden

2. Caching: ETag

- ETag's sind als Versionierung von Resourcen gedacht

- Für jede Version einer Resource wird ein (global) einzigartiger ETag übermittelt

- Bei jedem Abruf einer Resource schickt der Browser ETag Informationen mit

- Erneutes Übermitteln einer bekannten Resource wird verhindert

2. Caching: ETag

- Dies kann jedoch „umgedreht“ werden

- Statt einer Datei einen ETag zuzuordnen, kann einem Benutzter ein einzigartiger ETag zugeordnet werden

- Bei erneutem Aufruf derselben Resource identifiziert sich der Browser durch dieses ETag
Ablauf:

- GET /ETagIdentify
  → Response Header enthält: ETag: ''12345''
- späterer Request: GET /ETagIdentify
  → Request Header enthält: If-None-Match: ''12345''

2. Browser Metadaten

- Das JavaScript Interface der Browser enthält eine große Menge Metadaten für die Anpassungen an die Konfiguration des Benutzers:

  - Navigator/Fenster Informationen

  - Plugin Informationen

  - Graphik API

2. Extrahierbare Informationen

- Einige Metadaten können mithilfe von JavaScript erzeugt werden:
  - Zeitzone
  - Systemfarben
  - Installierte Fonts (CSS Test oder Plugins wie Flash (feste Reihenfolge))
  - WebGL Informationen (Version, Buffergrößen, Extensions, …)
  - Canvas Fingerprinting
  - Plugin/App Erkennung

2. Canvas FP - Idee

- HTML5 bietet „Canvas“-Funktionen an

- Ermöglicht lokale Erstellung und Bearbeitung von Graphiken

- Hardwarebeschleunigt

- Hardware/Software Konfiguration führt zu minimalen, kaum sichtbaren aber messbaren unterschieden bei erzeugten Graphiken

- Abhängig von: Browser, Betriebssystem, Graphiktreiber, Graphikkarte

2. Canvas FP - Verfahren

- Komplexes Bild wird im Browser erstellt und an Server übermittelt

- Minimale Unterschiede durch unterschiedliche Transparenz, Anti-Aliasing, Farbverlauf Erzeugung...

- Ändert sich nichts an Hardware/Software Konfiguration, wird identisches Bild erzeugt

- Schnelle Übertragung und einfache Speicherung durch Hash des Bildes

- Wird zB (unter anderem) bei Evercookie verwendet
  (auch von der NSA zum tracken von TOR Nutzern)

2. Plugin/App Erkennung

- Nur MIME Type zugeordnete Plugins sind auflistbar

- Einige Browser unterstützen installierbare Apps

- Durch Tricks lässt sich in manchen Fällen erkennen, ob eine bestimmte App installiert ist

- Beispiel:
    - Adblocker versteckt Elemente, die als Werbung erkannt werden
    - Es ist erkennbar, ob ein Element versteckt wurde
    - Wird also Werbung ausgeblendet, ist vermutlich ein Adblocker installiert

    (Implementierung: https://github.com/viracore/adblockdetect/)

2. Komplexere Methoden

- window.name Tracking

- Performance Tests (CPU+GPU)

- USB Scanning (GamePad API)

- Social Network Login Status

- TCP Port Scanning (auf localhost)

- Verhaltensmuster
    - Tipp/Scroll Verhalten
    - Reihenfolge aufgerufener Seiten
    - Gleichzeitig geöffnete Seiten (3. Party)

- ...

2. Übersicht (Chrome)

Metadaten	JS	Plugin	Deaktivierbar	Änderbar
SSL/TLS Handshake				X
IP Adresse				
HTTP-Header				X
ETag Caching			X	X
Canvas Caching	X		X	X
Browser Metadaten	X			X
Browser Plugins	X			X
CSS Fontserkennung	X			X
Plugin Fontserkennung	X	X	X	X
Zeitzone	X			X
Systemfarben	X			X
WebGL Metadaten	X		X	
Canvas Fingerprinting	X		X	
App Erkennung	X			
Performance Tests	X			

3. Schutz

- Schutz vor Fingerprinting extrem Schwierig
    - Privatsphäre Paradoxon
    - Anonymisierung
    - Randomisierung
    - TOR

3. Privatsphäre Paradoxon

- In vielen Fällen hat der Versuch, die Privatsphäre besser zu schützen einen gegenteiligen Effekt

- Werden Plugins (Flash, Java,...) deaktiviert oder Metadaten versteckt, hilft es nur, wenn genug andere Nutzer dies ebenfalls tun

- Verfügbare Informationen erfüllen einen Zweck und das Browser diese ausblenden ist unrealistisch

→ In der Masse verschwinden („Anonymity Sets“) oder Randomiserung

3. Anonymer Fingerprint

- Möglichst generischer Fingerprint, um in der Masse zu verschwinden

- Problem: Werden falsche Informationen übermittelt, kommt es zu einer falschen Interpretation der Konfiguration

  - Fehlende Plugins verhindern Media Wiedergabe und Programmstarts
  - Fehlerhaft dargestellte Seiten (CSS, Auflösung,....)
  - Eingeschränkte Sicherheit

- Schwierig alle Merkmale zu Anonymisieren

- Arbeitsintensiv oder benötigt 3. Party Software

→ Oder: „The Onion Router“

3. Größte Anonymity Sets 1

Größte Anonymity Set (aus 470,161 Fingerprints)

User Agent	Cookies?	Video, Timezone, Plug-ins, Fonts, Supercook-ies	Frequency
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.7) Gecko/20091221 Firefox/3.5.7	Yes	no javascript	1186
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7D11	No	no javascript	1100
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6	Yes	no javascript	1017
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6	Yes	no javascript	940
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.2) Gecko/20100115 Firefox/3.6 (.NET CLR 3.5.30729)	Yes	no javascript	886
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.2) Gecko/20100115 Firefox/3.6 (.NET CLR 3.5.30729)	Yes	no javascript	788
Mozilla/5.0 (Windows; U; Windows NT 6.1; de; rv:1.9.2) Gecko/20100115 Firefox/3.6	Yes	no javascript	775
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.2) Gecko/20100115 Firefox/3.6	Yes	no javascript	746
Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.1.7) Gecko/20091221 Firefox/3.5.7	Yes	no javascript	702
Mozilla/5.0 (Windows; U; Windows NT 5.1; de; rv:1.9.1.7) Gecko/20091221 Firefox/3.5.7 (.NET CLR 3.5.30729)	Yes	no javascript	618

Table 4. 10 Largest Anonymity Sets

Quelle: https://panopticlick.eff.org/browser-uniqueness.pdf

3. Größte Anonymity Sets 2

Größte Anonymity Set mit JavaScript (aus 470,161 Fingerprints)

User Agent	Cookies?	Video	Timezone	Frequency
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	480	345
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; de-de) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	-60	280
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	360	225
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	0	150
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; de-de) AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7D11	Yes	320x396x32	-60	149
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7D11	Yes	320x396x32	480	149
Mozilla/5.0 (iPod; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	300	145
Mozilla/5.0 (iPhone; U; CPU iPhone OS 3_1_2 like Mac OS X; en-us) AppleWebKit/528.18 (KHTML, like Gecko) Mobile/7D11	Yes	320x396x32	0	114
Mozilla/5.0 (Linux; U; Android 2.0.1; en-us; Droid Build/ESD56) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Mobile Safari/530.17	Yes	480x854x32	300	112
Mozilla/5.0 (iPod; U; CPU iPhone OS 3_1_2 like Mac OS X; de-de) AppleWebKit/528.18 (KHTML, like Gecko) Version/4.0 Mobile/7D11 Safari/528.16	Yes	320x396x32	-60	97

Table 5. 10 Largest Anonymity Sets with Javascript

Quelle: https://panopticlick.eff.org/browser-uniqueness.pdf

3. Randomisierter Fingerprint

- Erstellung zufälliger Fingerprints bei jedem Aufruf einer Seite

- Problem: auch hier wird eigentlicher Zweck verhindert

- Benötigt allerdings Software und Wissen (Datenbank)

- Also: sehr Arbeitsintensiv oder 3. Party Software

3. TOR 1

„The Onion Router“ verfügt über Vorkehrungen gegen Fingerprinting:

- Plugins werden komplett deaktiviert (und Policy: Keine Plugins installieren)

- Canvas Daten können nur über Opt-In ausgelesen werden

- Lokaler Proxy gegen Port Scanning

- Anzahl verwendbarer CSS-Fonts limitiert

- Editierte Auflösungsangaben (über zoom)

- WebGL über Opt-In, reiner Softwarerenderer in Planung

3. TOR 2

- HTTP-Header festgelegt (Windows, Englisch, Firefox, ...)

- Zeitzone festgesetzt

- Zeiterkennung in Javascript quantisiert (gegen Performancetests, Tippen...)

- Platformabhängige Features sind deaktiviert (Battery API, …)

- „New Identity Button“: Vollständiger Reset des Browsers
  (Cookies, Sessions, History, Cache, …)

Quelle: https://www.torproject.org/projects/torbrowser/design/#fingerprinting-linkability

3. Lösungsansätze

- Aufgabe des Browser, die Privatsphäre zu schützen

- Unnötige Informationen entfernen (Systemfarbe, ...)

- Abfrage von MIME-Types und Plugins nicht nötig (IE erlaubt dies zB. nicht)

- Opt-In für Plugin Aufrufe (Flash, Java...)

- Plugins, die auslesen bzw. Upload sensibler Daten verhindern
  (zB Blur/DoNotTrackMe)

4. Feldversuch - Ergebnis

Quelle: http://www.henning-tillmann.de/2013/10/browser-fingerprinting-93-der-nutzer-hinterlassen-eindeutige-spuren/

Vielen Dank

… hier noch ein Beispiel

Links:

- *IP → Geolocation: http://www.whereisip.net/*
- *WebGL Extensions: http://renderingpipeline.com/webgl-extension-viewer/*
- *Unterstütze SSL Ciphersuites: https://cc.dcsec.uni-hannover.de/*

4. Feldversuch

- 2013 durchgeführt von Henning Tillmann

- Fast 24.000 Fingerprints wurden analysiert

- Etwa 16.000 einzigartige Fingerpints

- Versuch mit extrahierten Metadaten und Graphik Caching

4. Feldversuch - Fingerprint

$M := \{$ http_user_agent, http_accept, http_accept_charset, http_accept_encoding, http_accept_language, navigator_appCodeName, navigator_appName, navigator_cookieEnabled, navigator_language, navigator_platform, color_depth, devicePixelRatio, timezone_offset, java_enabled, color_activeborder, color_activecaption, color_appworkspace, color_background, color_buttonface, color_buttonhighlight, color_buttonshadow, color_buttontext, color_captiontext, color_graytext, color_highlight, color_highlighttext, color_inactiveborder, color_inactivecaption, color_inactivecaptiontext, color_infobackground, color_infotext, color_menu, color_menutext, color_scrollbar, color_threeddarkshadow, color_threedface, color_threedhighlight, color_threedlightshadow, color_threedshadow, color_window, color_windowframe, color_windowtext, plugin_flash, plugin_adobe_acrobat, plugin_silverlight, plugins, mimetypes, fonts $\}$.

4. Feldversuch - Auswertung

- Plugin Versionsnummern am Aussagekräftigsten

- Viele Fingerprints änderten sich leicht

- Fehlerhafte oder Manipulierte Fingerprints problematisch

- Ohne fehlerhafte und doppelte Einträgen: 92,57% einzigartig

- Nur mit Plugins, MIME-Types, Fonts und „User-Agent“: 86,73% einzigartig

- Ohne JavaScript und Flash sinkt Anzahl der eindeutigen Datensätze deutlich

5. Rechtlicher Rahmen

- Keine rechtliche Grundlage

- keine 100% Wiedererkennung, damit schwierig rechtlichen Rahmen personenbezogener Daten anzuwenden

- Bei IP-Adressen:
    - Statische sind Personenbezogen Daten
    - Dynamische gelten für Provider als Personenbezogene Daten
    - Für Serviceanbieter ist Personenbeziehbarkeit unterschiedlich, da sie mit weiteren Daten verknüpft werden kann (zB in Niedersachsen: Ja)

- Fingerprints sind vergleichbar mit dynamischen IP Adressen

5. Rechtlicher Rahmen

- Bei Cookies: Sehr undurchsichtig

- EU Richtlinie verlangt Hinweis über Verwendung von Cookies

- Browser Einstellung übernimmt Einverständniserklärung

- In Deutschland aber noch nicht verpflichtend umgesetzt

Quelle: http://www.it-recht-kanzlei.de/cookies-einwilligung-datenschutz.html 2014

5. § 14 BDSG (kurz)

- Speicherung, Veränderung und Nutzung Personenbezogener Daten nur erlaubt, wenn:

1) Erforderlich für Verantwortliche Stelle (für den erforderten Zweck)
2) Für andere Zwecke, nur wenn:
    - Gesetze es verlangen
    - der Betroffene eingewilligt hat
    - es offensichtlich im Interesse des Betroffenen liegt
    - es allgemein zugängliche Daten sind
    - …

…

5. Rechtlicher Rahmen: Problem

- Werden Fingerprints als personenbezogene Daten und als nicht im Interesse des Betroffenen eingestuft, ist das aktuelle bestehende HTTP-Protokoll nicht mehr funktionsfähig

- Selbst wenn vor jeder Homepage eine Einverständniserklärung verlangt wird, so ist auch durch Verbindungsaufbau ein Fingerprint vorhanden

- Lösung: Allgemein Einverständniserklärung bei der Installation des Browsers
  oder Änderung des Protokolls

2. Navigator/Fenster

- Auflösung und verfügbare Fenstergröße

- Detaillierte Informationen über Browser

- Ladezeiten

- Sprachen

- Platform

- Cookies Aktiviert

2. Plugins

- Liste installierter Plugins mit zugehörigen MIMETypes

- Ermöglichen nativ unbekannte Resourcentypen korrekt anzuzeigen (Audio, Video...)

- Erlauben öffnen externer Programme (Downloadmanager, …)

- Detaillierte Versionsinformationen für Debugging