A-SIT Technology
E-Government Innovationszentrum

31. Oktober 2019 Studie · Privatsphäre & Krypto · Web & Cloud · 13 Min. Lesezeit

Browser-Erkennungsmethoden

Dominik Mocher und Gerald Palfinger



Archiviert. Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

Autorinnen und Autoren	Dominik Mocher, Gerald Palfinger
Stand	2019-10-31
Quelle	https://technology2.a-sit.at/veroeffentlichungen/browser-erkennungsmethoden/

Um Benutzerinnen und Benutzern von Websites personalisierte Angebote anzeigen zu können, müssen Werbenetzwerke Benutzerinnen und Benutzer über verschiedene Webseiten und Sitzungen hinaus wiedererkennen. Manche der für die Identifizierung verwendeten Methoden, wie beispielsweise Cookies, können jedoch durch Einstellungen im Browser oder durch Browsererweiterungen deaktiviert werden. Deswegen werden von den Werbenetzwerken immer neue Möglichkeiten gesucht, um Benutzerinnen und Benutzer zu identifizieren. Diese Studie soll einen Überblick darüber geben, welche Methoden verwendet werden, um Benutzerinnen und Benutzer über Merkmale wie den eingesetzten Browser und seine Einstellungen, das Betriebssystem oder die verwendete Hardware wieder zu erkennen.

1. Einführung

Viele Onlinedienste werden kostenfrei zur Verfügung gestellt. Um diese zu finanzieren, wird in der Regel Werbung geschaltet. Oft werden dazu die Tools großer Werbenetzwerke verwendet. Um die Werbung effektiver zu gestalten und so die Klickraten zu erhöhen, personalisieren diese Werbenetzwerke die angezeigte Reklame. Dazu müssen Nutzerinnen und Nutzer auch über verschiedene Seiten erkennbar bleiben. Um Nutzerinnen und Nutzer wiederzuerkennen, können Cookies eingesetzt werden. Dabei handelt es sich um kurze Textinformationen, die auf dem Gerät gespeichert werden. Durch die Ablage einer eindeutigen ID kann eine Nutzerin oder ein Nutzer damit eindeutig identifiziert werden. Diese Cookies rückten jedoch durch die Verwendung von Werbenetzwerken in den Fokus von Nutzerinnen und Nutzern und können zum Schutz der Privatsphäre deaktiviert bzw. regelmäßig gelöscht werden. Um Nutzerinnen und Nutzer weiterhin erkennen zu können, werden neue Methoden verwendet, die eine Speicherung von Wiedererkennungsmerkmalen auf dem Gerät substituieren. In dieser Studie wird ein Überblick darüber gegeben, wie eine Nutzerin oder ein Nutzer durch Fingerprinting wiedererkannt werden kann und aus welchen Elementen ein solcher Fingerabdruck bestehen kann.

2. Erkennungsmethoden

Die Erkennungsmethoden lassen sich in verschiedene Kategorien einteilen. Die folgenden Kapitel geben einen Überblick über Methoden, die spezifische Browsermerkmale, Hardware- und Betriebssystemeigenheiten sowie Browsererweiterungen als Basis für die Erstellung von sogenannten Fingerabdrücken verwenden.

2.1. Browserbasierte Fingerprints

In einer groß angelegten Studie [1] sammelte die Electronic Frontier Foundation Fingerabdrücke von fast einer halben Million Browsernutzern. Zur Erstellung des Fingerabdruckes werden vor allem Informationen verwendet, die der Browser den aufgerufenen Webseiten zur Verfügung stellt. Darunter fallen die folgende Informationen:

- User-Agent mit Informationen über Browsername und Version
- Gesendete HTTP-Header mit Informationen wie akzeptierten Formaten und Encodings
- Eingesetztes Betriebssystem und Bildschirmauflösung
- Eingestellte Zeitzone und Sprache
- Installierte Plugins (wie Flash oder Silverlight) und Schriftarten
- Information, ob Cookies bzw. Javascript aktiviert sind
- Hash eines mit Hilfe von Canvas bzw. WebGL erstellten Bildes zur Erkennung von Verarbeitungsunterschieden in der verwendeten Grafikhardware bzw. -software
- Weitere Informationen wie Unterstützung von Touchscreens bzw. Verwendung des Do-Not-Track HTTP-Headers

In der Studie konnte festgestellt werden, dass sich die Fingerabdrücke mit der Zeit ändern (z.B. durch Browser-Updates). Durch eine regelmäßige Beobachtung können die Nutzer jedoch weiterhin verfolgt werden, da sich in der Regel nur Teile des Fingerprints ändern und so der Zusammenhang weiterhin festgestellt werden kann. Mehr als 99% der Nutzer können so auch nach einem Browserupgrade wiedererkannt werden. Die Autoren deckten zudem auch auf, dass Erweiterungen, die Fingerprinting verhindern sollten, dieses teilweise sogar vereinfachen können. Dieser Fall tritt vor allem dann auf, wenn diese Module nur von wenigen Nutzern verwendet werden. Darunter fallen zum Beispiel Erweiterungen, welche eine Änderung des User-Agents ermöglichen. Werden hier beispielsweise Werte hinzugefügt oder der User-Agent manuell verändert, so lassen diese Änderungen den eindeutigen Rückschluss auf ein installiertes Plugin bzw. auf den Nutzer zu. Andere Erweiterungen, wie etwa das Javascript-deaktivierende NoScript, verringern die Fingerprintbarkeit des Browsers laut der Einschätzung der Autoren jedoch. Für Anwender ist eine solcher Nebeneffekt meist nicht feststellbar.

In [2] beleuchten Nikiforakis et.al. weitere Mechanismen, die eine Wiedererkennung des Nutzers ohne Cookies zulassen. Im Vergleich zur vorherigen Studie wurden hier die Vorgehensweisen drei kommerzieller Anbieter von Fingerprinting-Software verglichen. In der 2013 durchgeführten Studie verwendeten die meisten der Anbieter das Flash-Plugin, um auf systemspezifische Informationen zuzugreifen. Viele der Flash-Methoden lieferten genauere Informationen als die Browser-Äquivalente, z.B. nicht nur das eingesetzte Betriebssystem, sondern auch Details wie die Kernel-Version oder die Anzahl der eingesetzten Monitore. Inzwischen werden Flash und andere Browser Plugins jedoch kaum mehr verwendet und die Unterstützung für Flash soll bis 2020 eingestellt werden [3]. Andere der eingesetzten Fingerprinting-Methoden sind jedoch auch heute noch praktikabel. So unterscheiden sich verschiedene Browser durch Unterstützung von, teilweise experimentellen, Funktionen. Auch die Liste von installierten Schriftarten wird zur Erkennung beigezogen. Diese kann jedoch nicht über den Browser selbst eruiert werden – das Abrufen der installierten Schriftarten kann auf direktem Weg nur über Plugins wie Flash oder Java WebStart erfolgen. Doch auch wenn kein solches Plugin mehr verwendet wird, können installierte Schriftarten in vielen Fällen durch Ausprobieren erkannt werden. Diese Möglichkeit wurde bereits 2013 von einem der untersuchten Anbieter eingesetzt. Dazu wird ein für den Nutzer unsichtbares Element, welches einen Text enthält, auf der dargestellten Webseite erstellt. Dieser Text wird zuerst mit einer Standardschriftart dargestellt und danach die Größe des Elements gespeichert. In weiterer Folge wird die Schriftart des Texts wiederholt geändert und die Größe des Elements nach jeder Änderung gemessen. Unterscheidet sich diese von der initialen Größe, so ist die Schriftart installiert. Unterscheidet sich die Größe jedoch nicht, so wird davon ausgegangen, dass die Schriftart nicht installiert ist und der Browser auf die Standardschriftart zurückfällt. Die untersuchten Tracker verwendeten auch weitere Informationen, die über Javascript zugänglich sind, wie Zeitzone oder unterstützte Mime-Types. Auf weitere Techniken, wie die Erkennung von HTTP-Proxies mit Hilfe von Flash oder spezielle Active X-Plugins, wird hier aufgrund der fehlenden Unterstützung von Erweiterungen in modernen Browsern nicht weiter eingegangen.

2.2. Hardware-/ Betriebssystem Fingerprints

Cao et al. [4] zeigen Methoden auf, die auf Basis von Hardware- bzw. Betriebssystemeigenheiten ein System identifizieren können. Dadurch kann ein Nutzer auch über verschiedene Browser hinweg verfolgt werden. In der Studie werden verschiedene Aufgaben auf der Grafikeinheit mittels WebGL ausgeführt. Durch Unterschiede in der verwendeten Hardware bzw. Treiber unterscheiden sich gerenderte Bilder und Videos auf Pixelebene. Auch wenn es bei manchen Aufgaben nur Unterschiede von wenigen Pixeln gibt, kann durch die Kombination verschiedener Rendering-Aufgaben und Vergleich der Unterschiede dem

System mit hoher Wahrscheinlichkeit ein eindeutiger Fingerabdruck zugeordnet werden. Darüber hinaus kann auch die Web Audio API zur Erstellung eines Fingerabdruckes verwendet werden. So treten auch bei der Verarbeitung von Toninformationen erkennbare Unterschiede zwischen verschiedenen Systemen auf. Ebenso erlaubt die API auch expliziten Zugriff auf Informationen über das eingesetzte Audiogerät, wie z.B. die maximal unterstützte Abtastrate oder die Anzahl der unterstützten Kanäle. Je nach Hardware können sich diese Werte unterscheiden. Zusätzlich kann auch die Anzahl der logischen CPU-Kerne, welche über die Web Workers API abrufbar ist, als weiterer Anhaltspunkt für den Fingerabdruck verwendet werden.

2.3. Fingerprinting von Browser-Erweiterungen

Starov und Nikiforakis [5] stellten fest, dass auch die installierten Browsererweiterungen als Fingerabdruck verwendet werden können. Obwohl die meisten Browser keinen Direktzugriff auf die Liste der installierten Erweiterungen erlauben, konnten die Forscher die durchgeführten Änderungen an der angezeigten Webseite den installierten Plugins zuordnen. Da jedoch nicht jede Erweiterung Änderungen an einer Webseite durchführt, ist eine lückenlose Erkennung mit dieser Methode nicht möglich. 9,2% der 10.000 zum Untersuchungszeitpunkt beliebtesten Plugins führen Änderungen auf jeder beliebigen Webseite durch, während 16,6% der Erweiterungen Änderungen nur auf spezifischen Webseiten ausführen.

In [6] zeigten Sanchez-Rola et.al., dass die installierten Erweiterungsmodule in vielen populären Browsern durch weitere Methoden erkennbar sind. So konnte in Chromium-basierten Browsern (wie z.B. Chrome und Opera) festgestellt werden, dass der Zugriff auf Ressourcen einer Erweiterung unterschiedlich lange dauern, je nachdem ob diese installiert ist oder nicht. Da es nur eine begrenzte Anzahl an unterschiedlichen Erweiterungen gibt, können so alle öffentlich bekannten Plugins durch Ausprobieren entdeckt werden. Um diese Auflistung der installierten Erweiterungen zu verhindern, setzt der Browser Safari randomisierte URIs ein. Es konnte jedoch auch festgestellt werden, dass etwa 40% der getesteten Plugins diese randomisierte URI an die Webseiten verraten. Ist diese URI bekannt, kann diese zur eindeutigen Identifizierung eines Nutzers verwendet werden, da der randomisierte Teil zum Installationszeitpunkt der Erweiterung zufällig generiert und danach nicht mehr geändert wird.

In [7] evaluieren Gulyás et.al., wie akkurat sich Nutzer auf Basis der installierten Browserplugins und aktiven Webseitenlogins erkennen lassen. Dazu erkennen sie mit einer Anzahl von ungefähr 13.000 Erweiterungen etwa 28% aller im Google Chrome Store verfügbaren Erweiterungen. Des Weiteren konnten die aktiven Logins bei circa 60 Webseiten festgestellt werden. Aus der Studie geht hervor, dass etwa 55% der Nutzer, die eine Browsererweiterung verwenden, eindeutig anhand dieser erkannt werden können. Überdies geben die Forscher an, dass beinahe 54% der Nutzer von mindestens einer Erweiterung durch eine Untermenge von 485 Plugins erkannt werden können, wodurch eine schnelle Identifizierung der Nutzer möglich ist. Durch die Kombination mit den erkannten Logins können mehr als 89% der Anwender mit mindestens einer erkannten Erweiterung und einem festgestellten Login eindeutig identifiziert werden.

3. Gegenmaßnahmen

Es wurden eine Vielzahl an Strategien entwickelt, um die Möglichkeit des Browser-Fingerprinting einzuschränken. Eine vermeintlich einfache und naheliegende Strategie ist es, diesen eindeutigen Fingerabdruck zu verändern. Dies kann beispielsweise durch das Ersetzen von dafür relevanten Informationen durch vordefinierte oder randomisierte Werte geschehen. Wie jedoch bereits in Abschnitt 2.1 erwähnt wurde, kann ein solches Vorgehen die Möglichkeit der Erkennung eines Browsers sogar erhöhen.

Dies geschieht dadurch, dass die Kombination an geänderten Werten noch seltener ist als die Standardeinstellungen und deshalb die Zuordnung erleichtert. Ebenso kann durch Änderung einzelner Parameter die Konsistenz zwischen den übertragenen Werten verloren gehen. Wird zum Beispiel das vermeintlich verwendete Betriebssystem nur im User-Agent verändert, so kann das tatsächlich verwendete Betriebssystem dennoch auf anderem Wege wie beispielsweise das JavaScript-Property `navigator.platform` eruiert werden.

FPGuard [8] versucht herauszufinden, ob und wie wahrscheinlich es ist, dass eine Webseite probiert, einen Browser-Fingerabdruck zu erstellen. Dazu wurden neun verschiedene Metriken definiert, die auf entsprechende Aktivitäten hinweisen können. Darin enthalten sind beispielsweise wiederholte Aufrufe der Properties des `navigator` -Objekts, programmatische Zugriffe auf sichtbare oder versteckte Canvas-Objekte, sowie die Anzahl der mittels JavaScript geladenen Schriftarten. In einem weiteren Schritt versucht FPGuard auf verdächtigen Seiten den Inhalt des Fingerabdrucks zu verändern, um dadurch die Identifizierung des Nutzers zu verhindern.

Der Tor Browser [9] inkludiert verschiedene Mechanismen, um Fingerprinting zu verhindern bzw. zu erschweren. Dazu werden viele Werte, wie der User-Agent oder die Größe des Programmfensters, welche eine Auswirkung auf die gemeldete Bildschirmauflösung hat, standardisiert. Ebenso wird nur eine vordefinierte Liste an Schriftarten mitgeliefert. Das programmatische Auslesen von Canvas-Elementen muss von der Benutzerin oder dem Benutzer explizit erlaubt werden. Der Tor Browser deaktiviert Plugins wie Flash und es wird davon abgeraten, zusätzlich zu den bereits mitgelieferten Erweiterungen weitere zu installieren, da diese möglicherweise zur Erstellung eines Fingerprints verwendbar sind. Sollten die standardisierten Werte vom Nutzer verändert werden, sticht man jedoch im Vergleich zu anderen Browsern stärker aus der Masse hervor.

Auch Mozilla als Browserhersteller bietet im Firefox-Browser Schutz vor Fingerprinting und Aktivitätenverfolgung an [10]. Dieser Trackingschutz ist seit Version 69 standardmäßig aktiviert [11]. Damit werden beispielsweise bestimme zur Nutzerverfolgung verwendete Cookies geblockt. Ebenso werden bekannte Skripte unterbunden, die die Rechenleistung des aufrufenden Gerätes zum Minen von Kryptowährungen verwenden. Der Trackingschutz kann auch Fingerprinter blockieren [12], diese Funktion ist jedoch standardmäßig (Stand: Oktober 2019) deaktiviert. Wird der Schutz vor Fingerprintern wie in Abbildung 1 gezeigt durch den Nutzer aktiviert, so werden Anfragen zum Aufruf bekannter Fingerprintingelemente durch den Browser verhindert. Die Aktivierung des Trackingschutzes kann jedoch zu Problemen auf Webseiten führen und bietet nur Schutz vor bekannten Trackingtechnologien.

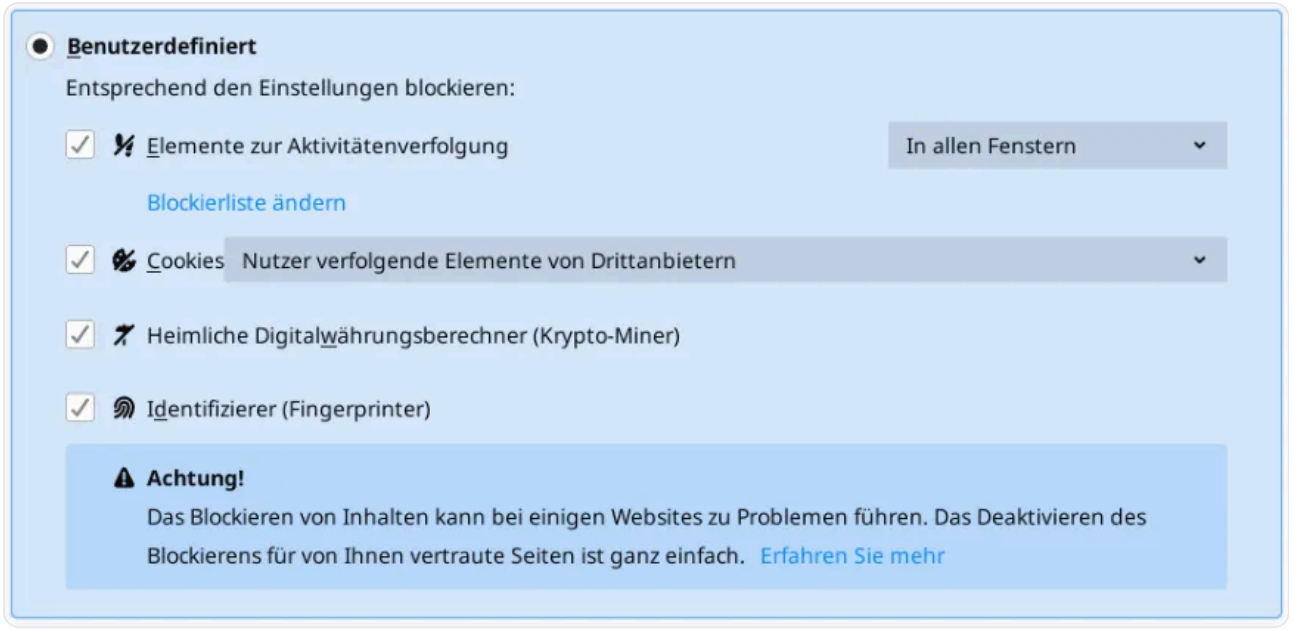


Abbildung 1: Firefox bietet eingebauten Schutz vor verschiedenen Arten von Aktivitätenverfolgung, darunter auch die Möglichkeit Fingerprinter zu blockieren

Durch das Messen von Zeitunterschieden mithilfe von Javascript-Methoden wie `performance.now()` kann das Tippverhalten und auch Mausbewegungen ermittelt werden. Da diese Daten sich zwischen einzelnen Anwenderinnen und Anwendern unterscheiden, können damit auch Benutzerinnen und Benutzer wiedererkannt werden. Auch das Auslesen von URLs, die in anderen Browserfenstern eingegeben wird, ist damit möglich. Um die Anwendung solcher Methoden zu erschweren, wurde die Auflösung entsprechender Zeitmessungen in allen verbreiteten Browsern auf 5µs reduziert [13]. Einzig der Tor-Browser verwendet einen noch konservativeren Wert von 100ms [14]. In [15] wird demonstriert, dass diese Einschränkungen jedoch umgangen werden können, indem eine eigenständige Zeitmessung implementiert wird. Die Autoren empfehlen die Verwendung eines umfangreichen Berechtigungsmodelles im Browser, um die für solche Angriffe notwendigen Schnittstellen abzusichern, da entsprechende Javascript-Methoden auch in Werbung eingebettet werden können.

Browser-Erweiterungen in Form von Adblockern werden genutzt, um werbebasiertes Tracking zu unterbinden. Webseiten, deren Betrieb mithilfe von Werbung finanziert wird, haben ein Interesse daran die Verwendung solcher Plugins zu erkennen. Oftmals wird der Besucherin oder dem Besucher der Inhalt einer Webseite verweigert, um diesen zur Abschaltung des Werbeblockers zu bewegen. Dieser Wettlauf zwischen Adblockern und Anti-Adblocker-Scripts wird in [16] behandelt. Darin wird beschrieben, wie Webseiten mit und ohne Adblocker besucht werden, um unterschiedliche Code-Pfade im Javascript-Code festzustellen. Aufgrund von diesen Unterschieden können neue Erkennungsmethoden für Adblocker aufgezeigt werden, sowie mittels Javascript-Rewriting die Präsenz solcher Erweiterungen verschleiert werden.

4. Fazit

In dieser Studie wurde eine Vielzahl an Möglichkeiten dargestellt, die es ermöglichen einen Fingerabdruck von Browsern zu erstellen. Diese können auch abseits von Cookies verwendet werden, um den Browser sowie Benutzerinnen und Benutzer zu erkennen. Diese reichen von der Aggregation verschiedener, durch

den Browser bereitgestellten Informationen über die Detektion verschiedener Einstellungen und installierter Erweiterungen bis hin zur Erkennung von Browser-, Betriebssystem- und Hardwareeigenheiten wie dem Canvas-Fingerprinting. Darüber hinaus wurden auch Gegenmaßnahmen beleuchtet, die von Browserherstellern oder versierten Nutzern in Form von Erweiterungen eingesetzt werden können um Fingerprinting zu verhindern oder zumindest zu erschweren. Dabei wurde auch das Paradoxon beleuchtet, dass manche, auf den ersten Blick sinnvoll erscheinende, Gegenmaßnahmen die Fingerprintbarkeit jedoch erhöhen können.

Referenzen

[1] P. Eckersley, „How Unique Is Your Web Browser?," 2010.

[2] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens und G. Vigna, „Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting," in *IEEE Symposium on Security and Privacy*, 2013.

[3] Adobe Corporate Communications, „Flash & The Future of Interactive Content," Adobe, 25 7 2017. [Online]. Available: https://theblog.adobe.com/adobe-flash-update/. [Zugriff am 23 7 2019].

[4] Y. Cao, S. Li und E. Wijmans, „(Cross-)Browser Fingerprinting via OS and Hardware Level Features," in *NDSS Symposium*, San Diego, 2017.

[5] O. Starov und N. Nikiforakis, „XHOUND: Quantifying the Fingerprintability of Browser Extensions," in *IEEE Symposium on Security and Privacy*, San Jose, 2017.

[6] I. Sanchez-Rola, I. Santos und D. Balzarotti, „Extension Breakdown: Security Analysis of Browsers Extension Resources Control Policies," in *USENIX*, 2017.

[7] G. G. Gulyás, D. F. Somé, N. Bielova und C. Castelluccia, „To Extend or not to Extend: On the Uniqueness of Browser Extensions and Web Logins," in *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, Toronto, Canada, 2018.

[8] A. FaizKhademi, M. Zulkernine und K. Weldemariam, „FPGuard: Detection and Prevention of Browser Fingerprinting," in *IFIP Annual Conference on Data and Applications Security and Privacy*, 2015.

[9] The Tor Project, „Tor Project | Anonymity Online," [Online]. Available:

https://www.torproject.org/. [Zugriff am 07 08 2019].

[10] G. Kontaxis und M. Chew, „Tracking Protection in Firefox For Privacy and Performance," in *Web 2.0 Security & Privacy*, 2015.

[11] M. Wood, „Today's Firefox Blocks Third-Party Tracking Cookies and Cryptomining by Default," Mozilla Corporation, 3 9 2019. [Online]. Available: https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookiesand-cryptomining-by-default/. [Zugriff am 16 10 2019].

[12] „Lösche deinen digitalen Fingerabdruck und hindere Werbeunternehmen daran, dir durchs Web zu folgen," Mozilla Corporation, 21 5 2019. [Online]. Available: https://blog.mozilla.org/firefox/de/loesche-deinen-digitalen-fingerabdruck-in-firefox/. [Zugriff am 16 10 2019].

[13] I. Grigorik, J. Simonsen und J. Mann, „High Resolution Time Level 2," [Online]. Available:

https://www.w3.org/TR/hr-time-2/#clock-resolution. [Zugriff am 25 10 2019].

[14] M. Perry, „Bug 1517: Reduce precision of time for Javascript," [Online]. Available:

https://gitweb.torproject.org/user/mikeperry/tor-browser.git/commit/?h=bug1517. [Zugriff am 25 10 2019].

[15] M. Lipp, D. Gruss, M. Schwarz, D. Bidner, C. Maurice und S. Mangard, „Practical Keystroke Timing Attacks in Sandboxed JavaScript," in *Computer Security – ESORICS 2017*, Cham, Springer International Publishing, 2017, pp. 191–209.

[16] S. Zhu, X. Hu, Z. Qian, Z. Shafiq und H. Yin, „Measuring and Disrupting Anti-Adblockers Using Differential Execution Analysis," 2018.

[17] M. Schwarz, F. Lackner und D. Gruss, „JavaScript Template Attacks: Automatically Inferring Host Information for Targeted Exploits," in *Network and Distributed Systems Security (NDSS) Symposium*, San Diego, 2019.