Dynamische Trackererkennung im Web durch Sandbox-Verfahren

Tim Wambach

Universität Koblenz-Landau
wambach@uni-koblenz.de

Zusammenfassung

Tracker, Zählpixel und Webbugs sind im Netz unsichtbar und allgegenwärtig. Darüber hinaus ist der Browser durch die Möglichkeit der Erweiterung zu einer Software gewachsen, die nicht mehr vom Benutzer selbst, sondern durch Inhalte aus dem Internet gesteuert wird. In dieser Arbeit wird eine Methode vorgestellt, mit der ohne Rückgriff auf Browsererweiterungen und ohne Selbstgefährdung eine solche Fremdsteuerung zum Zwecke der Besucherverfolgung festgestellt werden kann. Diese Methode beruht auf dem Sandbox-Verfahren, das sich in der Analyse von Schadsoftware bereits bewährt hat. Ziel ist es, eine strukturierte Vorgehensweise zur Analyse von Webseiten vorzustellen, die auch zur Auffindung zukünftiger Trackingverfahren in der Lage ist.

1 Einleitung

Die Analyse des Besucherverhaltens ist zu einer wesentlichen Informationsquelle für Webseitenbetreiber geworden. Die Auswertung der Daten erfordert jedoch einen hohen Aufwand und wird deshalb auch an Drittanbieter ausgelagert. Dieser ist darüber hinaus in der Lage, die gewonnenen Informationen mit den Daten anderer Kunden zu korrelieren. Eine solche übergreifende Besucherverfolgung, als Tracking bekannt, kann durch verschiedene Techniken realisiert werden.

In [MaMi12] findet sich eine Übersicht bzgl. Trackingtechnologien wobei zwischen *stateful* – auch "Supercookies" genannt, und *stateless*-Tracking – auch bekannt als "Fingerprinting" (vgl. [Ecke10]) unterschieden wird. Im Fokus dieser Arbeit stehen aktive Trackingverfahren, welche eine Wiedererkennung des Benutzers durch eine Markierung ermöglichen, die als Evercookies in [Kamk10] umfassend demonstriert werden. Techniken in diesem Zusammenhang sind: *Cookies*, *LSO-Cookies* (Local Shared Objects [MCMC11]) und *HTML5 Storage* (vgl. [w3c13]). Die Schwachstelle *CSS History Knocking* (vgl. [WCJJ11]) ist mittlerweile in aktuellen Browsern behoben und liefert deshalb nur in seltenen Fällen Ergebnisse.

Neben gesetzlichen Reglementierungen besteht die Möglichkeit, durch technische Hilfsmittel ein ungewolltes Web-Tracking zu registrieren und für die Zukunft zu blockieren. Es steht eine breite Auswahl an Browsererweiterungen zur Verfügung: die Addon-Suche in Mozilla Firefox findet 869 Ergebnisse zum Suchwort *Privacy* (Stand: Mai 2015). Erweiterungen, welche Benutzer über Trackingmethoden der besuchten Webseite informieren, wie z.B. Ghostery[1] oder Disconnect[2], können als statisch bezeichnet werden: Sie arbeiten auf einer bestehenden Da-

[1] Ghostery Inc.: Ghostery - `https://www.ghostery.com/de/` [abgerufen am 30.05.2015].
[2] Disconnect: Disconnect - `https://disconnect.me/` [abgerufen am 30.05.2015].

P. Schartner · K. Lemke-Rust · M. Ullmann (Hrsg.) · D•A•CH Security 2015 · syssec (2015) 301-310.

tenbasis und erkennen Tracker anhand einer ihnen typischen, in der Datenbank hinterlegten, Signatur. Tests zeigten, dass diese nicht in der Lage sind Verfahren von unbekannten Anbietern oder neuer Methoden zu registrieren. Zur dynamischen Erkennung kommt die Erweiterung TrackingObserver [RoKW12] in Frage, die Trackingverfahren anhand ihres Verhaltens ohne einer hinterlegten Datenbasis erkennt.

Die Möglichkeiten von Browsererweiterungen sind jedoch eingeschränkt. Externe Bibliotheken (Adobe Shockwave bzw. Flash, Oracle Java, Microsoft Silverlight, etc.), die den Funktionsumfang des Browsers erheblich erweitern, fallen nicht in deren Erfassungsraum. Auch Funktionen der Browser selbst können zu einer ungewollten Datenweitergabe führen – Beispiele hierfür sind:

- die "Phishing and Malware Protection"-Funktion [Mozi15] in Mozilla Firefox stellt, außerhalb des Erfassungsraums von Browsererweiterungen und vom Benutzer unbemerkt, eine Verbindung zu `safebrowsing.google.com` her.
- DNS prefetching ([Mozi14]) in Mozilla Firefox löst Hostnamen verlinkter Inhalte durch eine DNS-Server Abfrage in IP-Adressen auf, noch bevor diese vom Benutzer angewählt wurden.

In dieser Arbeit wird die Forschungsfrage behandelt, ob eine Suche nach Trackern auf Webseiten auch dann durchführbar ist, wenn der Browser und deren Erweiterungen als nicht vertrauenswürdig eingestuft wird und stellt eine neue Form der Privacy-Analyse für Webseiten vor: das Sandboxing-Verfahren. Im Bereich der Malwareanalyse ist eine Sandbox als ein Werkzeug zur dynamischen Analyse von Schadsoftware bereits etabliert. Der Begriff Sandbox stützt sich darauf, dass eine speziell präparierte virtuelle Maschine zum Einsatz kommt, die nach einer erfolgreichen Analyse mittels Sicherungspunkt (Snapshot) in ihren ursprünglichen Zustand zurückgeführt werden kann, wodurch die Gefahr für das Arbeitssystem minimiert wird. In einer solchen Umgebung können ausführbare Dateien, sowie auch Webseiten, gestartet bzw. aufgerufen und anschließend bzgl. ihres Verhaltens analysiert und bewertet werden. Verhalten bezieht sich in diesem Kontext auf Interaktion mit dem Betriebssystem bzw. dessen Ressourcen: Kernel-Aufrufe, erstellte, modifizierte und gelöschte Dateien sowie Änderungen der Registry und Netzwerkverbindungen.

Ziel ist es, neben der Suche nach Tracking-bezogenen Mustern in Sandbox generierten Analysedaten, die Erfassung und Analyse der durch Webseiten verursachten Seiteneffekte auf das System über die Browsergrenzen hinweg. Verwandt dazu ist [RoKW12]: hier werden Trackingverfahren durch ihr Verhalten kategorisiert und können durch eine Browsererweiterung im Chrome Browser auf Webseiten gesucht werden. Dabei kann allerdings nicht zwischen funktionalen Ressourcen und Tracking unterschieden werden. In [ABJB10] werden statische und dynamische Untersuchungen des "private browsing"-Modus durchgeführt. In [ShKa06] und [LDLP+14] werden Verhaltensänderungen von Webseiten bei unterschiedlichen Eingaben analysiert.

Diese Arbeit ist in Kooperation mit der interdisziplinären Projektgruppe "Strukturwandel des Privaten"[3] unterstützt durch die VolkswagenStiftung[4] entstanden.

[3] `http://www.strukturwandeldesprivaten.de`
[4] `https://www.volkswagenstiftung.de`

2 Analysemethodik

Grundsätzlich bietet eine Sandbox die Möglichkeit der Analyse des Netzwerkverkehr bzw. der Kommunikationsbeziehungen. Auf diese Weise kann das vollständige Kommunikationsverhalten z.B. die in der Einleitung erwähnte Verbindung zu `safebrowsing.google.com` durch Mozilla Firefox, protokolliert und analysiert werden. Eine stärkere Einbeziehung dieser Daten steht ebenfalls im Ausblick. Allerdings lässt sich mittels einer solchen statischen Analyse der Verbindungsdaten nicht erschließen, wie der Browser auf die eingehenden Daten reagiert. Aus diesem Grund wird bei dem vorgestellten Verfahren der Fokus auf das Verhalten des Browsers zur Laufzeit – genauer: auf die Interaktion mit dem Betriebssystem bzw. die Schreib- und Leseoperationen gesetzt. Diese Art der Analyse ist neu und in dieser Weise bisher noch an keiner anderen Stelle durchgeführt worden.

2.1 Sandbox

Bevor die Methodik genauer dargestellt wird, werden Konfiguration der Sandbox und notwendige Anpassungen näher beschrieben. Eine Sandbox zur Analyse von Schadsoftware verwendet Sicherungspunkte (Snapshots) einer virtuellen Umgebung, um nach der Analyse auf den vorherigen Zustand zurückkehren zu können. Auf diese Weise werden Auswirkungen der Schadsoftware auf die Analyseumgebung verhindert. Diese Vorgehensweise kann sich auch bei einer Analyse von Webseiten als nützlich erweisen, da alle Spuren auf dem System, die ein aktive Wiedererkennung ermöglichen könnten, entfernt werden. Um jedoch ein Tracking zwischen Webseiten verschiedener Domains messen zu können, bedarf es mehrerer Aufrufe im gleichen Kontext. So ist diese Wiederherstellung des Originalzustandes hinderlich, wenn das Verhalten über mehrere Webseitenbesuche hinweg beobachtet werden soll. Im Zuge der Analysen wurde die Sandbox deshalb so verändert, dass diese die Rückkehr zum Sicherungspunkt erst auf explizite Anweisung durchführt. Andernfalls wird der ausgeführte Prozess – der Browser – geschlossen und das System für das nächste Analyseobjekt vorbereitet. In dieser Arbeit wurde sich, aufgrund seines Verbreitungsgrades, Anpass- und Erweiterbarkeit entschieden, die Cuckoo-Sandbox [cuc15] einzusetzen. Innerhalb der Sandbox wird das Betriebssystem Microsoft Windows 7 mit dem Internet Explorer 8 eingesetzt.

Für die Tracking-bezogenen Analysen wurde in der Programmiersprache Python 2.7 ein Framework implementiert, um die Sandbox-generierten Daten unter Verwendung verschiedener Filtermethoden weiterverarbeiten zu können.

2.2 Cross-Domain Datenanalyse

Bei einer standardmäßigen Browserkonfiguration, kann ein Cookie der Webseite der Domain A (z.B. example.COM) bei einem erneuten Besuch nur von A selbst ausgelesen werden. So ist einer anderen Domain B (z.B. example.NET), aus Sicherheits- und Datenschutzgründen, nicht möglich, auf dieses Cookie zuzugreifen. Eine solche Separierung gilt auch für gespeicherte temporäre Dateien z.B. Bilder.

Um eine Domänen-übergreifende Besucherverfolgung zu ermöglichen, müssen die Anbieter der Domänen A und B einen gemeinsamen Drittanbieter C einsetzen (z.B. example.ORG). Bei einem solchen Cross-Domain-Tracking wird von A und B ein Cookie von C gesetzt und ermöglicht die Wiedererkennung des Benutzers – unabhängig davon ob A oder B aufgerufen wird.

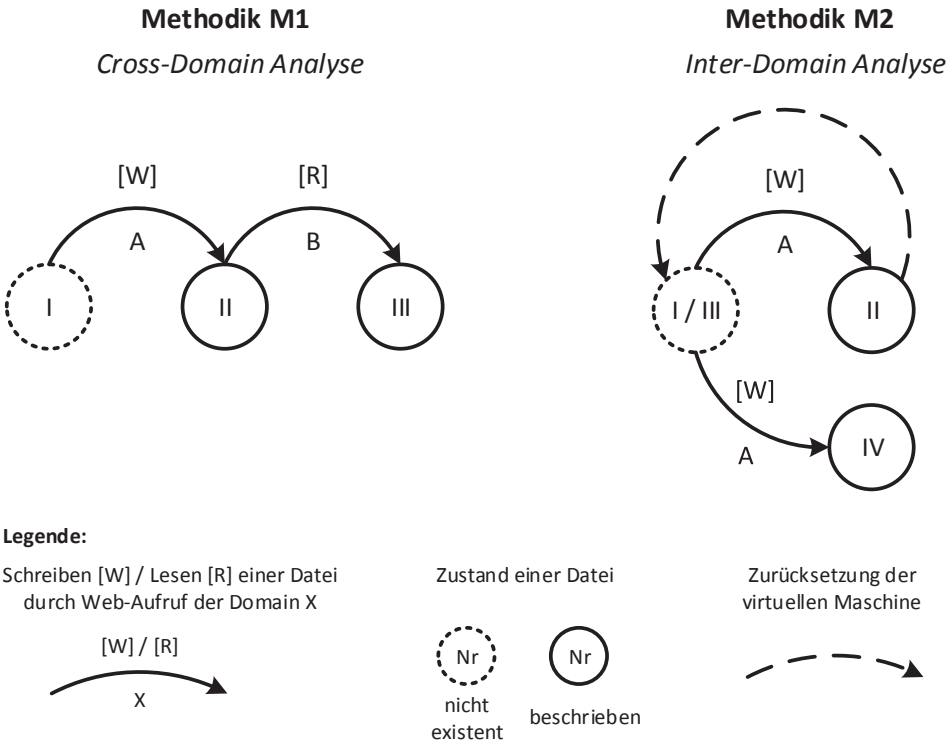

Abb. 1: Übersicht der Analysemethodiken M1 und M2.

Die in dieser Arbeit vorgestellte Methodik M1, die **Cross-Domain Datenanalyse**, zeigt mittels einer Sandbox den Zugriff auf Ressourcen die durch den Besuch verschiedener Webseiten ausgelöst werden. Auf diese Art können gemeinsame Speicherstellen, z.B. das oben genannte Cookie von C, zwischen den Aufrufen verschiedener Domänen visualisiert werden. Bei dieser Analyse werden allerdings auch Daten erfasst, die möglicherweise nur zur Verbesserung der Browsergeschwindigkeit auf dem System gespeichert und nicht zum aktiven Tracking eingesetzt werden (Cache). Eine genauere Differenzierung zwischen einer Speicherung zum Tracking und als Cache wird die Methode M2, die **Inter-Domain Datenanalyse**, erbringen.

Die erste Analysemethodik M1 beobachtet Effekte auf Systemressourcen beim Besuch mehrerer Webseiten im gleichen Kontext. Im gleichen Kontext ist dabei so definiert, dass zwischen Analysevorgängen die virtuelle Maschine nicht zurückgesetzt wird, sondern in Analyse n die gespeicherte Daten der Analyse $n-1$ noch bestehen. In der ersten Analyse liegt ein vollständig ungenutzter Zustand des Browsers vor.

Der Browser wird zwischen den Webseitenaufrufen geschlossen und der Arbeitsspeicherbereich freigegeben. Aus diesem Grund müssen Daten zur Wiedererkennung zwischen verschiedenen Domänen auf der Festplatte des Systems persistent gespeichert sein. Sofern die Sandbox korrekte Ergebnisse liefert, werden diese Speichervorgänge bei einer passenden Filterung sichtbar. In Abbildung 1 wird dieses Modell skizziert. Die hier verwendete Filtermethode (M1) liefert alle Dateien, die durch ein Besuch der Domain A geschrieben (II) und dem anschließenden Besuch einer anderen Domain B gelesen (III) werden.

2.3 Inter-Domain Datenanalyse

In der im letzten Abschnitt vorgestellten Cross-Domain Datenanalyse werden Ressourcen betrachtet, die zwischen Webseiten genutzt werden. Die Analyse zeigt jedoch nur gemeinsam genutzte Ressourcen auf, die auch tatsächlich zur Analysezeit verwendet werden.

Um diese Schwächen der ersten Methodik zu beheben werden in Methodik M2, der **Inter-Domain Datenanalyse**, gezielt die Möglichkeiten der Sandbox bzw. der virtuellen Umgebung eingesetzt. Anders als in Methodik M1, wird hier jedoch jede Domain für sich alleine separat betrachtet. Im ersten Schritt wird die Webseite besucht (II), analysiert und anschließend mittels eines Snapshots der virtuellen Umgebung der ursprüngliche Zustand wiederhergestellt (III). Daraufhin folgt ein zweiter Besuch (IV), der allerdings durch die Rückführung der virtuellen Maschine wie ein Erstbesuch gewertet werden kann. Durch die Analyse nach Wiederherstellung der ursprünglichen Umgebung wird ein Referenzpunkt geschaffen wie in Abbildung 1 zu sehen. Die Annahme ist, dass durch einen inhaltlichen Vergleich der geschriebenen und gelesenen Daten bei zwei unvorbelasteten Besuchen auf dessen Verwendungszweck geschlossen werden kann. Verglichen werden dabei:

- Namen, Schlüssel und Domänen bei Cookie-Dateien,
- Schlüssel und Wert bei Registry-Inhalten,
- der SHA-256 Hashwert der Datei bei allen anderen Daten.

Durch einen Vergleich der geschriebenen Daten, lässt sich die Speicherung zum Zwecke der Pufferung (Cache) von einer Speicherung zum Zwecke von Tracking abgrenzen: Sind die gespeicherten Inhalte identisch, liegt Zwischenspeicherung (Cache) nahe. Sind die Inhalte unterschiedlich, könnte der Inhalt zum Tracking verwendet werden.

Speichert eine Webseite beispielsweise benutzerbezogene Einstellungen (z.B. eine Lautstärkeeinstellung eines Flash-basierten Medienplayers), sind bei einem Wiederbesuch die gleichen Daten zu erwarten. Unterscheiden sich die gespeicherten Informationen allerdings, kann dies als ein Merkmal zur eindeutigen Identifizierung verwendet werden.

Zusammengefasst soll die Analyse und die dabei verwendete Filterung alle Dateien aufzeigen, die bei einem ersten Besuch im Browser gespeichert und sich inhaltlich von einem erneuten Erstbesuch unterscheidet, der durch die Sandbox sichergestellt ist. Durch eine zusätzliche virtuelle Maschine ist es möglich, den Referenzpunkt annähernd parallel zu erheben.

Wie bereits in der Einleitung beschrieben stehen Verfahren zur passiven Wiedererkennung, so genanntes Fingerprinting, außerhalb des Fokus. Allerdings kann deren Einfluss durch Anpassungen der virtuellen Umgebung und/oder den Einsatz eines Proxys zur Änderung der Ursprungsadresse wirksam minimiert werden.

3 Analyseergebnisse

Für die **Cross-Domain Datenanalyse** wurden die Top 100 Domains aus den Alexa Top 500 Deutschland [Alex15] (Stand Mai 2015) entnommen und mittels Sandbox analysiert. Diese Testmenge X kann im Anhang (vgl. Abschnitt 5) eingesehen werden. Die hier vorgestellten Ergebnisse verwenden den Filter aus Abbildung 1 (Methode M1): Eine Ressource gilt als gemeinsam genutzt, also geteilt, wenn eine Domain einen Schreibprozess und mindestens eine

weitere einen Leseprozess durchführt. Die analysierten Daten wurden dabei in drei Gruppen kategorisiert: **Cookies**, Temporäre Internet Daten (**Temp**) und Verschiedenes (**Misc**).

Innerhalb der Analyse der 100 Domains wurden insgesamt 623 Cookies gesetzt. Anzumerken ist jedoch, dass manche Cookies nur kurzzeitig auf dem System verweilen. Von diesen 623 Cookies entsprachen 190 dem beschriebenen Muster. Dies sind im Durchschnitt 1,9 Cookies pro Domain. Es zeigten sich dabei Cookie-Dateien, die von einer Domain gesetzt und von 24 anderen Domains ausgelesen wurden.

Bei Temporären Internet Daten handelt (**Temp**) es sich um einen Speicherort, der vom Browser explizit zum Zwecke der zur Pufferung von Daten vorgesehen ist; im Fall Internet Explorer 8 der Ordner:

```
[profil]\Local Settings\Temporary Internet Files
```

Wobei `[profil]` der Pfad zum Benutzerprofil ist. Es wurden 263 Dateien dieser Kategorie zwischen mindestens zwei Domänen geteilt. Davon:

- 35 Bilder (13,3%, Typen: .gif, .png, .jpg),
- 126 Skripte (47,9%, Typen: .js),
- 1 HTML-Quelltext (0,3%, Typen: .htm, .php, .html),
- 17 CSS-Dateien (6,5%, Typen: .css),
- 4 Schriftart-Dateien (1,5%, Typen: .eot),
- 0 XML-Dateien (0,0%, Typen: .xml),
- 0 Flash-Dateien (0%, Typen: .swf),
- 20 Text-Dateien (7,6%, Typen: .txt) und
- 60 Dateien ohne oder unbekannter Dateiendung (22,9%).

Der Vorteil der Sandbox im Vergleich zum Einsatz von Browser-Erweiterungen stellt die Analyse der letzten Gruppe **Misc** dar und offenbart weitere verwendete Ressourcen des Systems abseits der üblichen Speicherorte des Browsers.

- **Registry-Keys.** Während des Browserbetriebs wurde auf 969 Schlüssel der Registry zugegriffen, wovon 68 nach der beschriebenen Filterung verblieben sind.
- **Adobe Flash Plugin.** Die Flash-Konfigurationsdatei

  ```
  [profil]\AppData\Roaming\Macromedia\Flash Player\
    macromedia.com\support\flashplayer\sys\settings.sol
  ```

  wurde von 38 Domänen verarbeitet. Demnach nutzte keine Domäne Adobe Flash zum Austausch von Daten zwischen Domains. Bei 22 Domains wurden Speichervorgänge durch das Adobe Flash Plugin auf der Festplatte registriert. Zu finden sind diese in Domain-bezogenen Unterverzeichnissen von:

  ```
  [profil]\Application Data\Macromedia\Flash Player\macromedia.com
  ```

- **DOM-Store (HTML5).** Der DOM-Store wurde verteilt auf 5 Ressourcen von 24 Domains als gemeinsamer Speicher genutzt. Die Daten werden dabei in

  ```
  [profil]\Local Settings\Application Data\Microsoft\
    Internet Explorer\DOMStore
  ```

als XML-Datei abgelegt.

- **RecoveryStore.** Diese Daten sind in

  `\AppData\Local\Microsoft\Internet Explorer\Recovery\High\Last Active\RecoveryStore.{[...]}.dat`

  zu finden und dienen der Wiederherstellung einer unerwartet beendeten Sitzung.

Für die **Inter-Domain Datenanalyse** wurden die ersten Top 25 Webseiten aus [Alex15] (Stand: Mai 2015) entnommen (Testmenge A) und durch 25 Webseiten verschiedener Hochschulen (Testmenge B) ergänzt. Die genaue Zusammensetzung kann im Anhang (Abschnitt 5) eingesehen werden. Die Filterung entsprach der Methodik M2 aus Abbildung 1, mit der Ergänzung, dass nur Dateien betrachtet wurden, die dauerhaft auf der Festplatte verblieben sind.

In der Testmenge A wurden im Durchschnitt 15,44, bei Testmenge B 1,88 Cookies gesetzt, die zwischen erstem und zweitem Besuch zugeordnet werden konnten. Sowohl in A, als auch in B änderten sich die Cookies: dies ist leicht einzusehen, da bei Erstellung einer Sitzung bereits ein Cookie mit einer eindeutigen ID gesetzt wird. Die Anzahl der Cookies ist daher ebenfalls ein wichtiger Indikator. In Abbildung 2 findet sich eine Zusammenfassung der 50 Analysen (entlang der Abszissenachse), die das Verhältnis der gesetzten Cookies zu Cookies mit geändertem Inhalt darstellt. Das obere Ende der Linie bzw. die Pfeilspitze markiert die Anzahl der insgesamt gesetzten Cookies, während das untere Ende die Anzahl der unveränderten Cookies markiert; die Länge, Δy, gibt die Anzahl der Cookies mit geändertem Inhalt zwischen zwei unabhängigen Aufrufen an.

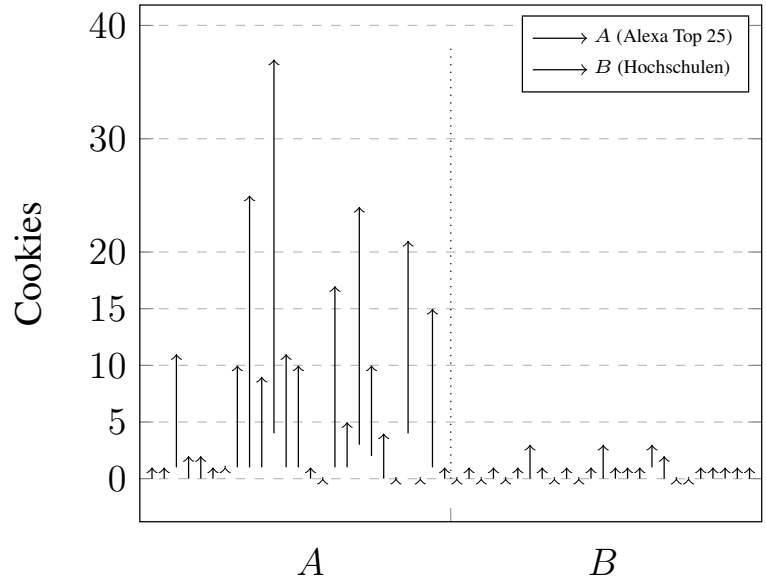

Abb. 2: Verhältnis gesetzter Cookies (Pfeilspitze) zu gesetzten Cookies mit identischem Inhalt (Startpunkt). V.l.n.r: vgl. Abschnitt 5.

In der Kategorie **Temp** wurden bei Bildern keine besonderen Auffälligkeiten entdeckt. Eine Änderung der Webseitenbeschreibung (HTML) ist allgemein zu erwarten. Es zeigt sich jedoch, dass Javascript-Dateien in einigen Fällen bereits vom Server, vor Auslieferung an den Browser, angepasst wurden. In der Testmenge A trat dies in zwölf Fällen, in der Testmenge B in zwei Fällen auf.

Bei **Misc** konnten bzgl. Registry-Keys keine ungewöhnlichen Ergebnisse festgestellt werden. Auch wenn durchschnittlich in Testmenge A 398 und in B 368 verschiedene Registry-Keys verwendet wurden, findet sich pro Überprüfung nur zwischen 1-4 Schlüssel eine Differenz bei Besuch und Wiederbesuch. Diese wurden manuell geprüft und wurden zufällig gewählt oder hatten einen Bezug zur Systemzeit. In 9 Fällen bei A und einem Fall bei B wurde auf Adobe Flash Speicher zugegriffen, allerdings ohne dabei eine langfristige Speicherung durchzuführen. In 19 der 25 Fälle der Testmenge A wurde der DOM-Store verwendet wobei in elf Fällen mit unterschiedlichen Inhalten. In Testmenge B wurde der DOM-Store in zwei Fällen verwendet wobei sich der Inhalt zwischen Besuch und Wiederbesuch nicht unterschieden hat.

4 Zusammenfassung und Ausblick

Es wurde gezeigt, wie durch Einsatz einer Sandbox eine Analyse von Webseiten durchgeführt werden kann, ohne dabei Erweiterungsfähigkeiten des Browsers zu verwenden und ohne die umliegende Arbeitsumgebung zu gefährden. Während Methodik M1 die verschiedenen genutzten Speicherformen und Möglichkeiten des Browsers aufzeigte, wurden in Methodik M2 die Vorteile der Virtualisierung für einen inhaltlichen Vergleich der gespeicherten Daten verwendet. Im praktischen Test konnten durch Methodik M1 die verschiedenen Trackingverfahren und deren Speicherstellen gefunden werden. Über Methodik M2 konnte evaluiert werden, wie diese Speichermöglichkeiten für eine Wiedererkennung des Benutzers verwendet werden.

Eine Fortführung des Ansatzes würde eine stärkere Einbeziehung der Netzwerkkommunikation zur Auffindung von Trackingverfahren vorsehen. Hierbei ergibt sich jedoch das Problem der Analyse von Inhaltsdaten in verschlüsselten Verbindungen. Weitere mögliche Einsatzgebiete dieser Technik wären die Sicherheitsanalyse von Browsererweiterungen (Addons, Plugins) oder der ”private browsing” Modus verschiedener Browser.

5 Anhang

Zusammensetzung der Testmengen; auf Protokollangaben und optionale third-level domains wurde verzichtet.

$X = \{$google.de, facebook.com, amazon.de, ebay.de, youtube.com, google.com, wikipedia.org, web.de, t-online.de, gmx.net, bild.de, yahoo.com, spiegel.de, googleadservices.com, xhamster.com, mobile.de, paypal.com, chip.de, focus.de, live.com, streamcloud.eu, wetter.com, gutefrage.net, immobilienscout24.de, bahn.de, twitter.com, blogspot.de, otto.de, idealo.de, xing.com, youporn.com, chefkoch.de, pornhub.com, ask.com, rtl.de, kicker.de, bing.com, heise.de, postbank.de, 1und1.de, welt.de, sueddeutsche.de, kinox.to, msn.com, booking.com, autoscout24.de, dhl.de, ok.ru, telekom.com, mpnrs.com, tumblr.com, dict.cc, microsoft.com, movie4k.to, de.wordpress.com, zalando.de, wetteronline.de, sport1.de, fiducia.de, leo.org, zeit.de, redtube.com, amazon.com, bs.to, linkedin.com, stern.de, outbrain.com, computerbild.de, adobe.com, wow.com, adscale.de, arbeitsagentur.de, t.co, onclickads.net, instagram.com, apple.com, adcash.com, wetter.de, n-tv.de, faz.net, deutsche-bank.de, xvideos.com, twitch.tv, ikea.com, tubecup.com, new.livejasmin.com, imgur.com, aol.de, tchibo.de, transfermarkt.de, pinterest.com, duden.de, mydealz.de, commerzbank.de, ebay.com, freenet.de, holidaycheck.de, telekom.de, sky.de, lidl.de$\}$

$A = \{$google.de, facebook.com, amazon.de, ebay.de, youtube.com, google.com, wikipedia.org, web.de, t-online.de, gmx.net, bild.de, yahoo.com, spiegel.de, googleadservices.com, xham-

ster.com, mobile.de, paypal.com, chip.de, focus.de, live.com, streamcloud.eu, wetter.com, gutefrage.net, immobilienscout24.de, bahn.de}

$B = \{$tu-dresden.de, uni-saarland.de, hochschule-trier.de, uni-luebeck.de, uni-potsdam.de, uni-mannheim.de, uni-koblenz-landau.de, uni-klu.ac.at, uni-regensburg.de, tugraz.at, th-nuernberg.de, tu-ilmenau.de, www3.uni-bonn.de, ruhr-uni-bochum.de, htwg-konstanz.de, uni-halle.de, fh-ooe.at, fh-salzburg.ac.at, uni-giessen.de, hs-fulda.de, uni-goettingen.de, fh-zwickau.de, uni-flensburg.de, uni-kassel.de, w-hs.de}

Literatur

[ABJB10] G. Aggarwal, E. Bursztein, C. Jackson, D. Boneh: An Analysis of Private Browsing Modes in Modern Browsers. *In: Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, USENIX Association, Berkeley, CA, USA (2010), 6–6.

[Alex15] Alexa: Top 500 Sites in Germany. `http://www.alexa.com/topsites/countries/DE` (2015), abgerufen am 30.05.2015.

[cuc15] Automated Malware Analysis - Cuckoo Sandbox. http://www.cuckoosandbox.org/ (2015), abgerufen am 30.05.2015.

[Ecke10] P. Eckersley: How Unique is Your Web Browser? *In: Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, Springer-Verlag, Berlin, Heidelberg (2010), 1–18.

[Kamk10] S. Kamkar: evercookie - virtually irrevocable persistent cookies. http://samy.pl/evercookie/ (2010), abgerufen am 30.05.2015.

[LDLP+14] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, R. Geambasu: XRay: Enhancing the Web's Transparency with Differential Correlation. In: *CoRR*, abs/1407.2323 (2014).

[MaMi12] J. R. Mayer, J. C. Mitchell: Third-Party Web Tracking: Policy and Technology. *In: Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, IEEE Computer Society, Washington, DC, USA (2012), 413–427.

[MCMC11] A. M. Mcdonald, L. F. Cranor, A. M. Mcdonald, L. F. Cranor: A Survey of the Use of Adobe Flash Local Shared Objects to Respawn HTTP Cookies (2011).

[Mozi14] Mozilla: Controlling DNS prefetching - HTTP | MDN. `https://developer.mozilla.org/en-US/docs/Web/HTTP/Controlling_DNS_prefetching` (2014), abgerufen am 30.05.2015.

[Mozi15] Mozilla: How does built-in Phishing and Malware Protection work? https://support.mozilla.org/en-US/kb/how-does-phishing-and-malware-protection-work (2015), abgerufen am 30.05.2015.

[RoKW12] F. Roesner, T. Kohno, D. Wetherall: Detecting and Defending Against Third-party Tracking on the Web. *In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, USENIX Association, Berkeley, CA, USA (2012), 12–12.

[ShKa06] U. Shankar, C. Karlof: Doppelganger: Better Browser Privacy Without the Bother. *In: Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, ACM, New York, NY, USA (2006), 154–167.

[w3c13] w3c: Web Storage - W3C Recommendation. `http://www.w3.org/TR/webstorage/` (2013), abgerufen am 30.05.2015.

[WCJJ11] Z. Weinberg, E. Y. Chen, P. R. Jayaraman, C. Jackson: I Still Know What You Visited Last Summer: Leaking Browsing History via User Interaction and Side Channel Attacks. *In: Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, IEEE Computer Society, Washington, DC, USA (2011), 147–161.