Beobachtungsmöglichkeiten im Domain Name System

Angriffe auf die Privatsphäre und Techniken zum Selbstdatenschutz

Handout

Dr. Dominik Herrmann

Universität Hamburg

Vortragsfolien und Handout zum Download:
http://dhgo.to/castfolien, http://dhgo.to/casthandout

Die Dissertation weist auf bislang vernachlässigte Beobachtungsmöglichkeiten im DNS hin und zeigt Möglichkeiten zum Selbstdatenschutz auf.

1. Motivation und Forschungsbedarf

2. Forschungsbeiträge der Dissertation
   a. Erkennung besuchter Webseiten
   b. Identifizierung benutzter Software
   c. Techniken zum Selbstdatenschutz
   d. Verhaltensbasierte Verkettung
   e. Schutz vor verhaltensbasierter Verkettung

3. Schlussfolgerungen & Handlungsempfehlungen

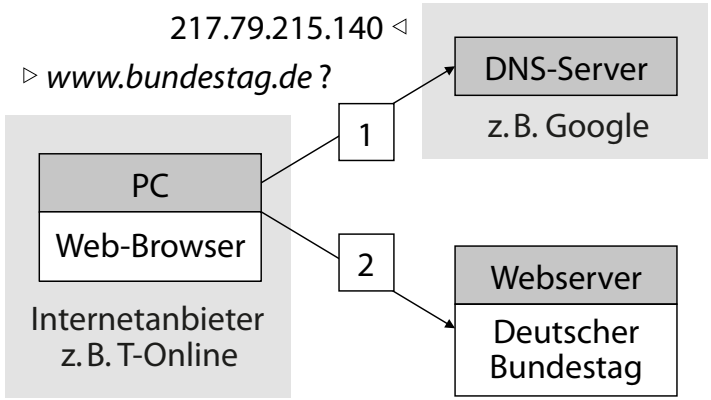
INHALT

Das Domain Name System (DNS) ist ein dezentral organisierter Infrastrukturdienst, der im Internet die Namensauflösung übernimmt.

Das DNS übersetzt (u.a.) die für Menschen leicht zu merkenden Domainnamen in IP-Adressen.

Diese Namensauflösung wird von den sog. rekursiven Nameservern (im Folgenden: **DNS-Server**) durchgeführt. Diese kommunizieren dazu mit sog. „autoritativen Nameservern. Dort ist die Domain-IP-Zuordnung in Resource-Records hinterlegt. Standardmäßig verwenden Internetnutzer einen DNS-Server, den ihr Internetanbieter betreibt.

Der Vortrag konzentriert sich auf die Beobachtungsmöglichkeiten auf den rekursiven Nameservern.



1. MOTIVATION UND FORSCHUNGSBEDARF

Das Schutzziel Vertraulichkeit wurde im DNS bislang vernachlässigt.

Beim Entwurf des DNS lag der Schwerpunkt auf **Verfügbarkeit** und Autonomie, realisiert durch Redundanz und Verteilung.

DNSSEC konzentriert sich auf das Schutzziel **Integrität** bzw. Authentizität.

Das Schutzziel **Vertraulichkeit** wurde jedoch bislang vernachlässigt, da

1. der Internetanbieter ohnehin den vollständigen Datenverkehr einsehen kann und
2. vermeintlich keine sensiblen Inhalte übertragen werden, sondern nur „Metadaten".

„Due to a deliberate design choice, DNSSEC does not provide confidentiality" (RFC 4033)

Der Schutz der Vertraulichkeit bzw. die Möglichkeit der Auswertung von DNS-Anfragen gewinnt wegen drei Trends an Bedeutung.

1. Zentralisierung der Namensauflösung: Internationale Unternehmen (z.B. OpenDNS, Google, Symantec) betreiben öffentliche rekursive Nameserver, die kostenlos genutzt werden können, hohe Performanz und Ausfallsicherheit bieten und die Umgehung von DNS-basierter Internetzensur erlauben.

Googles DNS-Server (8.8.8.8) beantworteten 2013 bis zu 150 Mrd. Anfragen/Tag. Der Anteil der Nutzer solcher **DNS-Fremdanbieter** wird auf 3% geschätzt (Callahan et al., 2013).

vgl. das im Jahr 2009 diskutierte ZugErschwG

2. Im DNS werden in Zukunft **zusätzliche schützenswerte Informationen** hinterlegt.

z.B. im „Object Naming Service“ und mit ENUM (Auflösung von VoIP-Telefonnummern)

3. Die **Auswertung von Verkehrsdaten** zur Aufklärung und Prävention von Straftaten nimmt zu. Die Beschlagnahme und Analyse der im rekursiven Nameserver aufgezeichneten DNS-Anfragen bietet zusätzliche Ermittlungsmöglichkeiten und wäre ein geringerer Eingriff in die Privatsphäre eines Verdächtigen als die TK-Überwachung.

Vorzüge: (1) Weniger Aufwand als bei normaler TK-Überwachung, da DNS-Verkehr nur einen Anteil von 0,05% am übetragenen Datenvolumen hat (Brandhorst & Pras, 2006). (2) Auch nachträgliche Analyse möglich, da DNS-Logs meist für einige Tage aufbewahrt werden (z.B. bei Google 24–48 h).

Ausmaß und Effektivität der Beobachtungsmöglichkeiten sind bislang unbekannt; praxistaugliche Selbstdatenschutz-Werkzeuge fehlen.

Diesen Forschungsbedarf adressiert die Dissertation mit **vier Forschungsfragen**:

1. Welche (nützlichen) Informationen lassen sich anhand von DNS-Anfragen rekonstruieren?

2. Wie können sich Nutzer vor Beobachtung schützen?

3. Wie gut lassen sich die Aktivitäten eines Nutzers nachvollziehen, der unter mehreren dynamischen IP-Adressen auftritt (Tracking)?

4. Wie können sich Nutzer gegen Tracking wehren?

Beim Surfen im WWW werden zahlreiche DNS-Anfragen gestellt. Die Analyse dieser Anfragen ist aus zwei Gründen lohnenswert.

Motivation für die Ermittlung der besuchten Webseiten anhand der DNS-Anfragen:

- Der Betreiber des DNS-Server möchte Verhaltensprofile seiner Nutzer anlegen. Dadurch können er bzw. seine Vertragspartner z.B. zielgerichtete Werbebanner einblenden oder Nutzer aufgrund ihrer Interessen diskriminieren **(Angriff auf Privatsphäre)**.

- Ermittlungsbehörden (oder CERTs) können das DNS-Query-Log eines DNS-Servers beschlagnahmen und auswerten, um die letzten Internet-Aktivitäten eines Straftäters (oder Angreifers) zu rekonstruieren **(IT-Forensik)**.

Bislang werden dazu v.a. Tracking-Techniken eingesetzt, die auf HTTP-Cookies basieren. Die „Internet Society“ vermutet jedoch, dass Werbenetze zum Tracking in Zukunft auch die Informationen einbeziehen werden, die auf DNS-Servern gewonnen werden können (Conrad, 2012).

Ziel: anhand der angefragten Domains möglichst alle Webseiten ermitteln, die tatsächlich besucht worden sind, jedoch keine, die nicht besucht wurden (Risiko der **Fehlinterpretation**).

2A. ERKENNUNG BESUCHTER WEBSEITEN

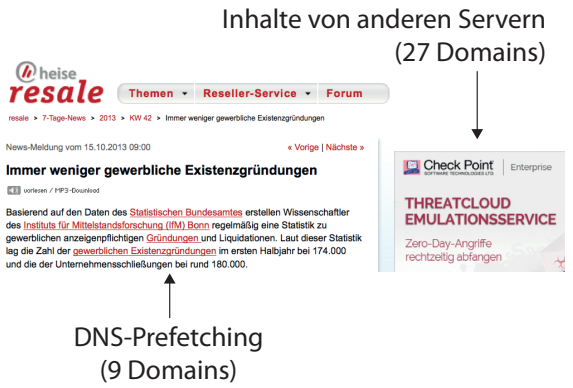
Kann ein DNS-Server ermitteln, welche Webseiten ein Nutzer besucht hat? Allein anhand der DNS-Anfragen gelingt dies zunächst nicht ohne weiteres.

Zwei Herausforderungen zu überwinden:

1. Der DNS-Server kann grundsätzlich nur die angefragten Domains beobachten. Oft sind jedoch **die URLs** interessant.

2. Beim Abruf einer Webseite werden meist **zahlreiche Domains** angefragt, deren Webseiten **nicht besucht** wurden (Risiko der Fehlinterpretation)

Beispiel:
http://de.wikipedia.org/wiki/Alkoholkrankheit




```
26-Jan-2010 10:23:49.770 client 132.199.9.16#15619: query: www.heise.de IN A +
26-Jan-2010 10:23:50.121 client 132.199.9.16#42151: query: script.ioam.de IN A +
26-Jan-2010 10:23:50.122 client 132.199.9.16#12191: query: ad-emea.doubleclick.net IN A +
26-Jan-2010 10:23:50.510 client 132.199.9.16#33170: query: www.dci.de IN A +
26-Jan-2010 10:23:50.527 client 132.199.9.16#6051:  query: www.ifm-bonn.de IN A +                ...
```

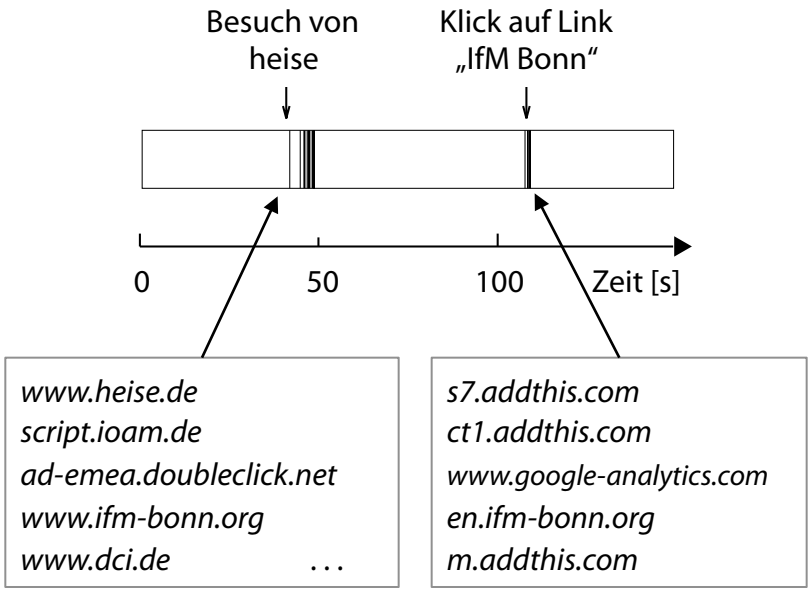
DNS-Query-Log beim Besuch von *http://www.heise.de/-1973600* (Auszug; insg. 37 Domains)

Besuchte und nicht-besuchte Webseiten sind schwer auseinanderzuhalten, u.a. weil moderne Web-Browser das sog. DNS-Prefetching implementieren.

Einfache Beispiel-Heuristiken:

- **H1:** Eine Webseite wurde genau dann besucht, wenn die angefragte Domain die Form *„[www.]domain.tld“* hat (also etwa *heise.de* oder *www.heise.de*, jedoch nicht *script.ioam.de*).
- **H2:** Die Webseite einer beobachteten Domain wurde genau dann besucht, wenn mindestens *t* Sekunden lang zuvor keine DNS-Anfrage gestellt worden ist (z.B. *t*=2).

Fallstudie (s. Abb. rechts):
Zuerst wird *http://www.heise.de/-1973600* besucht. Auf dieser Seite wird ein Link angeklickt, der zur Seite „IfM Bonn“ führt.



Beobachtbare DNS-Anfragen

Ergebnis bei Anwendung der Heuristiken:

H1:	***www.heise.de***, ***www.ifm-bonn.org***, *www.dci.de*, *www.google-analytics.com*
H2:	***www.heise.de***, *s7.addthis.com*

Die Tatsache, dass beim Besuch vieler Webseiten zahlreiche Domains angefragt werden, lässt sich allerdings bei der Erkennung ausnutzen.

Idee des in der Dissertation entwickelten **Website-Fingerprinting-Verfahrens**: Hat eine Webseite ein einzigartiges DNS-Abrufmuster, lässt sich aus dessen Auftreten schließen, dass die Seite besucht wurde.

Abrufmuster: Menge der Domains, die beim Besuch einer Webseite (URL) angefragt wird. Zeitabstände/Reihenfolge der Anfragen werden zur Erhöhung der Robustheit vernachlässigt.

Verwendung: Der DNS-Anbieter erzeugt eine Datenbank mit Abrufmustern, für die er sich interessiert. Dann vergleicht er die DNS-Anfragen eines Nutzers mit den Abrufmustern in seiner Datenbank, um die tatsächlich besuchten Webseiten zu identifizieren.

Abrufmuster von *http://www.ifm-bonn.org*:

{*ct1.addthis.com, en.ifm-bonn.org, m.addthis.com, s7.addthis.com, www.google-analytics.com, www.ifm-bonn.org* }

Für den Vergleich wird fortlaufend ein **Übereinstimmungswert** berechnet, der angibt, welcher Anteil von Muster *M* vom Benutzer angefragt worden ist (Menge *D*): $W_M = |M \cap D| \,/\, |M|$

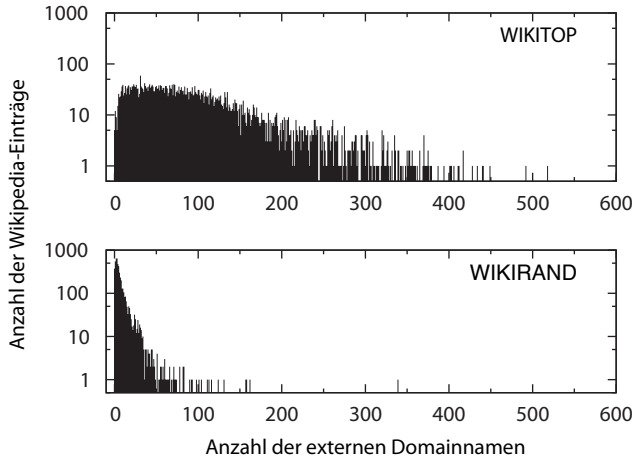
Untersuchungen zeigen: viele Webseiten weisen ein charakteristisches DNS-Abrufmuster auf. Bei einigen Seiten lässt sich sogar die URL rekonstruieren.

Untersuchte Fragestellung: Verfügen typische Webseiten über ausreichend umfangreiche und einzigartige Abrufmuster?

Empirische Überprüfung durch Abruf von
100.000 populären Homepages (ALEXA),
6283 heise-News-Seiten aus 2014 (HEISE),
5000 populären Wikipedia-Seiten (WIKITOP),
5000 zufälligen Wikipedia-Seiten (WIKIRAND).

Evaluation mittels ***k*-Identifizierbarkeit**:
Wert von *k* gibt an, wie viele Webseiten ein bestimmtes Abrufmuster besitzen.

Ergebnisse deuten darauf hin, dass der DNS-Server den Besuch von zahlreiche Webseiten rekonstruieren kann – bei Seiten wie heise und Wikipedia sogar die aufgerufene URL.



Datensatz	Median(# Domains)	k=1	k≤5
ALEXA	8	>99,9%	>99,9%
HEISE	2	63%	76%
WIKITOP	78	>99%	>99%
WIKIRAND	4	80%	94%

Die Tabelle gibt den Anteil der Seiten an, die 1-identifizierbar bzw. höchstens 5-identifizierbar sind. Seiten mit $k = 1$ sind eindeutig identifizierbar (innerhalb des Datensatzes).

Weiterhin geben die DNS-Anfragen eines Nutzers die von ihm eingesetzte Software preis. Dies erleichtert die Vorbereitung gezielter Angriffe.

Motivation für die Identifizierung der benutzten Software mittels DNS-Anfragen:

- Kenntnis der eingesetzten Software verbessert die Erfolgsaussichten gezielter Angriffe (*targeted attacks*). Dies gefährdet die **Sicherheit von IT-Systemen**.

  So müssen zur Infiltrierung eines Systems mit einem Zero-Day-Exploit i.d.R. Betriebssystem und Browser sowie ggf. verwendete Virenscanner bekannt sein.

- Ermittlungsbehörden können dadurch Informationen über die von einem Straftäter verwendeten Endgeräte gewinnen (sog. Geräte-Fingerabdruck). Werden dazu passende Endgeräte später sichergestellt, lassen sie sich durch den Fingerabdruck mit der Straftat in Verbindung bringen **(IT-Forensik)**.

  Bisher wird dazu z.B. der **User-Agent-Header** ausgewertet, den der Browser an einen Webserver übermittelt, z.B. „Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv 32.0)“. Anhand der DNS-Anfragen lassen sich vergleichbare bzw. zusätzliche Informationen ermitteln.

Dass die Software-Identifizierung mit DNS-Anfragen gelingt, liegt zum einen daran, dass verräterische Domains aufgelöst werden.

Betriebssystem, Browser und viele Anwendungen verraten sich bzw. ihren Hersteller durch **regelmäßige Anfragen** (für Software-Updates, Zeitsynchronisation, etc.).

ctldl.windowsupdate.com, *su3.mcaffee.com*, *aus3.mozilla.org, time.euro.apple.com*

Systematische Untersuchung von Betriebssystemen, Browsern und Anwendungen:

- Windows XP/7/8, Mac OS X 10.8, Ubuntu 12.04, CentOS 6.3, OpenSuSE 12.2
- Firefox 25, IE 8/10, Safari 6.1, Chrome 31.0
- Avira, Avast, Bitdefender, Kaspersky, McAffee, ICQ, Skype, Dropbox, Twitter, Flash, Java, Acrobat Reader, Eclipse, …

Ergebnis: praktisch alle Systeme bzw. Anwendungen lassen sich anhand von DNS-Anfragen eindeutig erkennen.

Falls ein Nutzer den DNS-Server eines Fremdanbieters **im Betriebssystem einträgt** (statt im Router), erfährt der Server weitere sensible, eigtl. **intern aufzulösende Domains**.

ThinkPadX230.local, hplaserjet2100.fritz.box, *PC1.Speedport_W723_V_Typ_A_1_00_096, www.gppgle.de.intranet.bundestag.de*

Gängige Betriebssysteme und Browser lassen sich jedoch auch anhand ihres charakteristischen *Verhaltens* bei der Namensauflösung identifizieren.

Die RFCs enthalten keine präzise Vorgaben zum Ablauf der Namensauflösung.

These: Software-Hersteller implementieren unterschiedliche Strategien, anhand derer sich ihre Software identifizieren lässt.

Drei ausnutzbare Verhaltensmerkmale:

1. Umgang mit AAAA-Anfragen (IPv6) und Domain-Suffix,
2. Verhalten bei Fehlern (z.B. NXDOMAIN, d.h. „Domain existiert nicht“),
3. Verhalten bei ausbleibender Antwort (Abstand der Retransmissions).

Ergebnis: Identifizierung gängiger Browser und Betriebssysteme möglich, falls diese Merkmale zu beobachten sind.

Beispiel:

Besuch von *http://invalid.name.de* mit Chrome:

Win XP: *invalid.name.de, invalid.name.de.local*

Win 7: *invalid.name.de*

MacOS: *invalid.name.de, invalid.name.de.local, invalid.name.de*

Ubuntu: *invalid.name.de, invalid.name.de.local, invalid.name.de, invalid.name.de.local*

System	Browser	Retransmission-Abstände [s]
Win 7	Firefox	1 1 2 4 6 1 1 2 4
	IE 10	1 1 2 4 4 1 1 2 4
MacOS	Firefox	1 3 9 17 1 3 9
	Safari	1 3 9 27 81
Ubuntu	Firefox	5 5 5 5 5 5 5 5 5 5 …
	Chrome	1 2 5 4 1 4 1 2 …

In der Dissertation werden frühere Vorschläge zur vertraulichen Namensauflösung evaluiert und neue Selbstdatenschutz-Techniken vorgeschlagen.

Motivation für die Fokussierung auf **Selbstdatenschutz-Techniken** zur vertraulichen Namensauflösung:

- Verzicht auf die Verwendung von DNS-Fremdanbietern bietet zwar Schutz vor Beobachtung durch diese, kommt jedoch nicht für alle Nutzer in Frage.

  z.B. nicht für Nutzer, die bewusst einen Fremdanbieter nutzen, um DNS-Sperren zu umgehen

- Anonymitätsdienste zum Surfen im WWW verhindern zwar Beobachtung durch DNS-Server, kommen jedoch ebenfalls nicht für alle Nutzer in Frage.

  z.B. nicht für Nutzer, die mit der Performanz der existierenden Anonymitätsdienste unzufrieden sind.

- Nicht alle Nutzer benötigen vertrauliche Namensauflösung; Selbstdatenschutz-Techniken sind daher ggü. Anpassungen an der DNS-Infrastruktur zu bevorzugen.

  Mit PPDNS (Lu & Tsudik, 2009) existiert zwar ein Vorschlag zur datenschutzfreundlichen Namensauflösung; wegen hoher Einführungshürden ist jedoch nicht mit einer raschen Verbreitung zu rechnen.

Die Analyse existierender Vorschläge zum Selbstdatenschutz identifiziert Defizite im Hinblick auf Performanz und Sicherheit.

Vorschlag von Bortzmeyer (2013):
Nutzer könnten ihre Anfragen **direkt an die autoritativen Nameserver** übermitteln, anstelle sie an einen rekursiven Nameserver zu senden.

Dies würde die **Namensauflösung verlangsamen**, da (1) die Nutzer dann nicht mehr vom gemeinsamen Cache auf dem rekursiven Nameserver profitieren würden, und (2) die Paketumlaufzeit zu den autoritativen Nameservern größer wäre.

Zudem verlagert der Vorschlag die Beobachtungsmöglichkeiten lediglich **zu den Root- bzw. TLD-Servern** (z.B. für „com"), da DNS-Clients die vollständige Domain (FQDN) dorthin übermitteln würden.

Vorschlag von Zhao et al. (2007):
Nutzer könnten ihre beabsichtigten Anfragen durch Senden von Dummy-Anfragen für **zufällig ausgewählte Domains** tarnen (Range Querys). Idee: Bei *n* Anfragen errät der Beobachter die beabsichtigte Anfrage somit nur mit Wahrscheinlichkeit $^1/_n$.

Diese Einschätzung vernachlässigt allerdings die Tatsache, dass beim Abruf vieler Webseiten DNS-Abrufmuster entstehen. Die in der Dissertation durchgeführten Analysen zeigen, dass zufällige Dummy-Anfragen **weniger Sicherheit bieten als angenommen** bzw. dass effektiver Schutz nur mit hohem Aufwand erreicht werden kann.

[siehe Herrmann et al. (2014) für Details]

Das entwickelte DNSMIX-System ermöglicht die weitgehend verzögerungsfreie und unbeobachtbare Namensauflösung

Beobachtungen:

1. Daten im DNS ändern sich relativ selten
2. Nutzerverhalten gehorcht Potenzgesetz (Zipf's law)

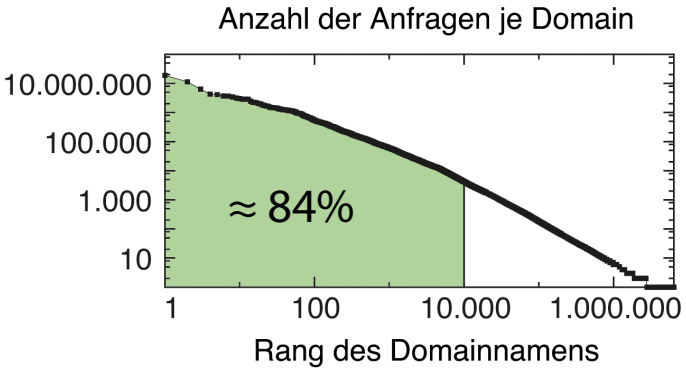
These: Es lässt sich ein Push-System konstruieren, das den Großteil der benötigten DNS-Daten unverlangt an die Nutzer übermittelt.

Die Namensauflösung für die unverlangt übermittelten Domains ist nicht beobachtbar und erfolgt ohne jede Verzögerung. Die übrigen Anfragen werden mittels einer Mix-Kaskade anonymisiert.

Das DNSMIX-System wurde implementiert, um seine Praktikabilität zu evaluieren.

Zu 1: In einer Studie mit 37.000 Domains blieben 87% der Datensätze im Beobachtungszeitraum (57 Tage) gleich (Handley & Greenhalgh, 2005).

Zu 2: Für die Dissertation wurde eine Studie durchgeführt, in der die DNS-Anfragen von ca. 4000 Nutzern über einen Zeitraum von 5 Monaten aufgezeichnet wurden (s. Abschn. 2D). In diesem Datensatz entfallen etwa 84% der Anfragen auf die populärsten 10.000 Domains (s. Abb.).



Der Aufwand für das Push-System ist hoch, aber vertretbar. Die DNSMIX-Kaskade ist schneller als der generische Anonymisierungsdienst Tor.

Implementierung des Push-Systems

Der Push-Dienst kennt die populärsten Domains (Annahme!) und ruft regelmäßig deren Resource-Records von den autoritativen Nameservern ab – immer dann wenn die TTL abgelaufen ist (Aufwand A1).

Verbindet sich ein Client mit dem Push-Dienst, erhält er initial die Daten für alle populären Domains (Aufwand A2), danach einen kontinuierlichen Datenstrom mit Änderungen (Aufwand A3).

Evaluation mit 2082 Nutzern über 24 h:

- A1 für Push-Dienst: 352 MB/Tag
- A2 initial je Client: 290 KB
- A3 laufend je Client: 0,8 KB/s

Implementierung der Mix-Kaskade

Die Konstruktion der Mix-Kaskade und das unterstellte Angreifermodell orientieren sich an existierenden Anonymisierungsdiensten, die in der Praxis eingesetzt werden (insbes. AN.ON bzw. Tor).

Es werden Kanäle mit einer Lebensdauer von 60 s geschaltet (RSA 2048 Bit OAEP, AES 128 Bit OFB) und alle Nachrichten haben dieselbe Länge: 57 Bytes (Anfragen) bzw. 89 Bytes (Antworten).

Evaluation mit 2082 Nutzern über 2 h bei 3 Mixen in einem emulierten WAN mit 120 ms Netzwerklatenz. Der Median der Antwortzeiten beträgt dann 171 ms (zum Vergleich: bei DNS über Tor ca. 1400 ms).

Für einen DNS-Server ist es nicht ohne weiteres möglich, die Aktivitäten eines Nutzers länger zu verfolgen, wenn sich dessen IP-Adresse ändert.

Motivation für Konzeption und Untersuchung eines Verfahrens zur verhaltensbasierten Verkettung:

- DNS-Server kann Anfragen verschiedener Nutzer grundsätzlich nur anhand deren IP-Adresse auseinanderhalten.
- Internetanbieter weisen Privatkunden meist dynamische IP-Adressen zu, die sich häufig ändern (z.B. täglich).
- Zielgerichtete Werbung erfordert jedoch Beobachtung über längere Zeiträume. Untersuchung. Bislang ist unklar ob und wie gut Werbenetze Nutzer auch ohne Cookies wiedererkennen können.
- Verkettung von Sitzungen ist auch für **forensische Zwecke** von Nutzen.

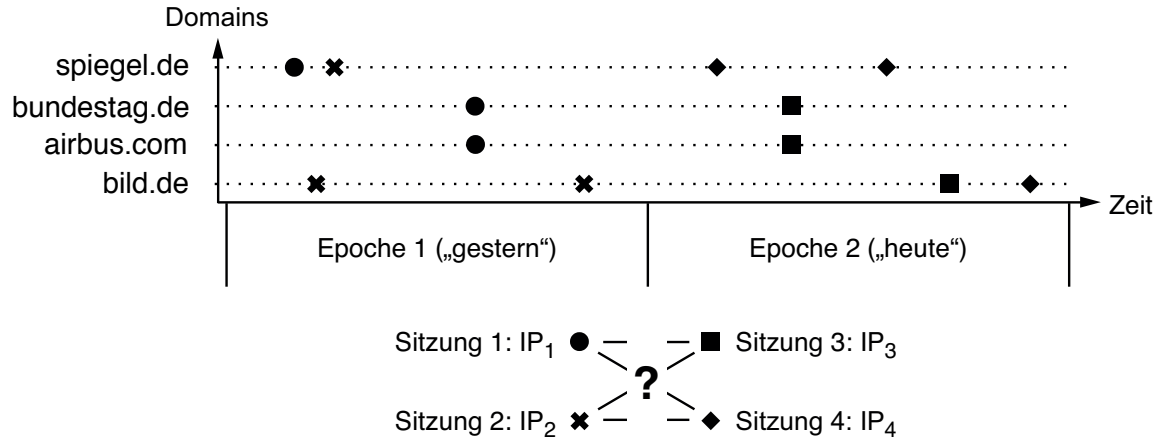
Durch Analyse der DNS-Anfragen könnte ein DNS-Server versuchen, Nutzer mit dynamischen IP-Adressen anhand ihres Verhaltens wiederzuerkennen.

These: Menschen haben individuelle Vorlieben, denen sie regelmäßig nachgehen.

Vorgehen: Betreiber des DNS-Servers trainiert ein Klassifikationsverfahren mit den DNS-Anfragen aus der Vergangenheit, um einen Nutzer damit auch nach dem Wechsel der IP-Adresse wiederzuerkennen.

Verkettungspotential: Die verhaltensbasierte Verkettung gelingt, wenn das Verhalten eines Nutzers im Zeitverlauf konstant bleibt (Stationarität) und er sich anders verhält als andere Nutzer (Individualität).

Fragestellung: Reichen die Anfragen aus einer einzigen Sitzung zur Verkettng aus?



Für das Problem der verhaltensbasierten Verkettung einzelner Sitzungen existieren noch keine geeigneten Lösungen.

Das **Verfahren von Yang (2010)** zielt darauf ab, Nutzer anhand ihres Internet-Surfverhaltens zu authentifizieren.

Merkmal: Menge der besuchten Webseiten

Klassifikation: Extraktion von Mustern mittels Support- und Lift-Maßen; Vergleich von Sitzungen über euklidischem Abstand

Evaluation: Closed-World-Evaluation mit 100 konkurrierenden Nutzern

Ergebnis: 62% der Nutzer werden korrekt wiedererkannt, wenn die Authentifizierung anhand einer einzigen Sitzung erfolgt.
Aber: der Klassifikator wird zuvor mit 200 alten Sitzungen von jedem Nutzer trainiert!

Vorgehensweise in der Dissertation

- Sichtung existierender Ansätze; Auswahl vielversprechender Verfahren
- Adaption des Verfahrens von Yang für die Verkettung einzelner Sitzungen
- Konzeption eines eigenen Verfahrens
- Implementierung der Verfahren mit MapReduce unter Apache Hadoop
- Erhebung eines ausreichend großen und realitätsnahen DNS-Datensatzes
- Kontrollierte Evaluation zum Parameter-Tuning und Benchmarking
- Open-World-Evaluation unter realitätsnahen Bedingungen
- Bestimmung von Einflussfaktoren

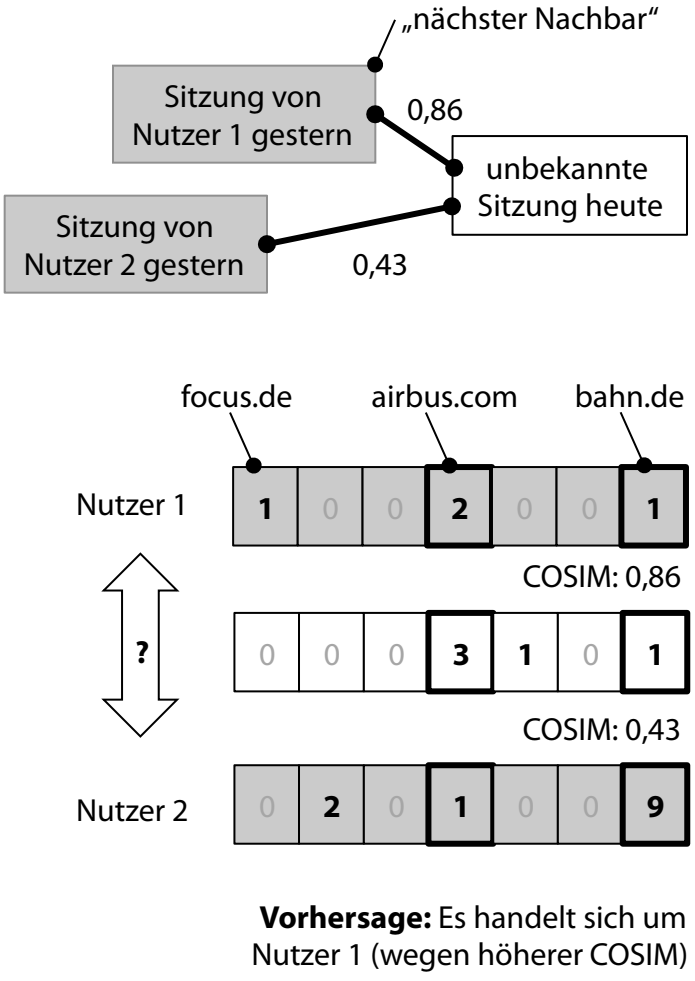
Das konzipierte Verfahren zur verhaltensbasierten Verkettung nutzt eine erprobte Merkmalsrepräsentation und gängige Klassifikationstechniken.

Merkmale: Multimenge der in einer Sitzung angefragten Domains inkl. ihrer jeweiligen Anfragehäufigkeit. Anwendung gängiger Transformationsfunktionen aus dem Text-Mining zur Reduktion störender Einflüsse:

- Logarithmierung der einzelnen Anfragehäufigkeiten,
- Multiplikation mit IDF zur Reduktion des Gewichts populärer Domains,
- Normalisierung auf Einheitslänge und
- Erzeugung von N-Grammen, um Abrufreihenfolge einfließen zu lassen.

Klassifikatoren:

- 1-Nächste-Nachbarn (1NN) mit Jaccard- bzw. Cosine-Similarity-Maß (COSIM),
- Multionimial Naive Bayes (MNB).



Vorhersage: Es handelt sich um Nutzer 1 (wegen höherer COSIM)

Es wurde eine Evaluationsumgebung entwickelt, um das Abschneiden der Klassifikatoren anhand von großen DNS-Logs zu analysieren.

Die Evaluationsumgebung und die Klassifikatoren wurden mit dem **MapReduce-Paradigma** implementiert, das von Google zur parallelen bzw. verteilten Verarbeitung großer Log-Dateien entwickelt wurde.

Zur Ausführung wird das MapReduce-Framework **Apache Hadoop** eingesetzt. Alle Experimente wurden auf 18 Desktop-PCs (intel Core i5, 3,1 GHz, 8 GB RAM, 1TB HDD) durchgeführt.

Jedes Experiment durchläuft drei Phasen:

1. Erzeugen des benötigten Datensatzes,
2. Trainining des Klassifikators, Verkettung der Sitzungen und
3. Ermittlung der Vorhersagegenauigkeit.

Die Implementierung umfasst ca. 11.000 Zeilen Code bzw. 70 Java-Klassen. Sie ist Open-Source-Software und wurde unter *https://github.com/hadoop-dns-tracking/* veröffentlicht.

Die Evaluationsumgebung unterstützt zwei Modi: Entweder wird wie bei einer Kreuzvalidation vorgegangen und eine bestimmte Nutzergruppe betrachtet (Closed World) oder die Evaluation iteriert über alle Epochen im Datensatz (Open World).

Zur Untersuchung der Praxistauglichkeit der verhaltensbasierten Verkettung ist ein großer Datensatz mit „echten“ DNS-Anfragen erforderlich.

Ein geeigneter Datensatz war bislang nicht verfügbar.

In Zusammenarbeit mit dem Rechenzentrum einer deutschen Universität wurden die DNS-Anfragen aller Benutzer des Campus-Netzes (Studentenwohnheime und Büro-Arbeitsplätze) über einen Zeitraum von fünf Monaten aufgezeichnet.

IP-Adressen werden im Campus-Netz statisch vergeben, d.h. Verkettungsvorhersagen können überprüft werden.

Es wurden u.a. folgende **Datensätze** gebildet (identischer Zweimonatszeitraum):

2M nur Nutzer in Studentenwohnheimen
2M* zusätzlich Büro-Arbeitsplatznutzer

Überlegungen zum Datenschutz: Um eine ausreichende Größe zu erreichen und Beobachtungseffekte (z.B. Hawthorne-Effekt) zu vermeiden kamen eine Vorabinformation bzw. das Einholen von Einwilligungen bei allen Betroffenen nicht in Frage. Um die Privatsphäre der Nutzer dennoch möglichst gut zu schützen, wurden die IP-Adressen durch Anwendung einer Einwegfunktion (kryptographische Hashfunktion mit konstantem, jedoch unbekanntem Salt) pseudonymisiert.

Die meisten Experimente wurden mit dem 2M-Datensatz durchführt, der eine hohe Qualität und nur wenige Anomalien aufweist.

Kumulative Verteilung relevanter deskriptiver Statistiken im 2M-Datensatz

Deskriptive Statistik	Min.	P_{25}	P_{50}	P_{75}	Max.
Anfragen pro Nutzer	1	24 068	59 367	121 185	28 857 393
Anfragen pro Domainname	1	1	2	4	9 781 157
Domainnamen pro Nutzer	1	2 106	4 184	7 433	303 568
Aktive Nutzer pro Tag	1 107	2 100	2 497	3 092	3 218
Anfragen pro Nutzer und Tag	1	569	1 384	2 969	5 144 359
Domainnamen pro Nutzer und Tag	1	196	372	671	258 110
Aktive Tage pro Nutzer	1	31	43	52	61

Verteilung der RR-Typen der DNS-Anfragen im 2M-Datensatz

Typ	Anfragen		Domainnamen		Nutzer	
	absolut	*relativ*	*absolut*	*relativ*	*absolut*	*relativ*
Total	431 210 371	100,000	5 010 507	100,000	3 862	100,000
A	236 210 050	54,778	3 668 822	73,223	3 860	99,948
AAAA	149 322 427	34,629	2 633 070	52,551	3 170	82,082
PTR	43 060 608	9,986	815 852	16,283	1 934	50,078
SRV	1 497 622	0,347	322	0,006	2 690	69,653
MX	474 827	0,110	252 953	5,048	45	1,165
ANY	281 023	0,065	7	0,000	1 526	39,513
SOA	226 975	0,053	131	0,003	351	9,089
TXT	115 300	0,027	8 715	0,174	680	17,607
NS	12 028	0,003	346	0,007	35	0,906
TKEY	4 518	0,001	2	0,000	1	0,026
NAPTR	4 281	0,001	10	0,000	14	0,363
SPF	512	0,000	236	0,005	1	0,026
CNAME	196	0,000	190	0,004	9	0,233
AXFR	2	0,000	1	0,000	1	0,026
NULL	2	0,000	1	0,000	1	0,026

Der Datensatz 2M im Überblick

Zeitraum: 1. Mai 2010 bis 30. Juni 2010

Anzahl der Nutzer: 3862

Anzahl der Anfragen: 431,2 Mio.

Anzahl der Domains: 5,0 Mio.

Zunächst wurden kontrollierte Experimente durchgeführt, um geeignete Betriebsparameter zu ermitteln und die Klassifikatoren zu vergleichen.

Vorgehensweise in Anlehnung an **klassische 10-fache Kreuzvalidation**:

1. Zufallsauswahl: 3000 Nutzern, die mind. an 20 Tagen aktiv waren.
2. Zufallsauswahl: 20 Sitzungen je Nutzer
3. Trainieren des Klassifikators mit je einer Sitzung pro Nutzer; Klassifikation je einer anderen Sitzung aller Nutzer.
4. Bestimmung der Precision- und Recall-Werte (P bzw. R) zur Ermittlung der Genauigkeit.

Zu beachten: Es handelt sich dabei um eine Closed-World-Evaluation, deren Ergebnisse sich nicht unmittelbar auf die Praxis übertragen lassen.

Zusammenfassung der Ergebnisse:

Die Yang-Verfahren sind zur Verkettung eher ungeeignet (P: 0,52, R:0,30).

Die selbst entwickelten Verfahren sind besser geeignet: 1NN-JACCARD (P: 0,60, R: 0,57) schneidet schlechter ab als 1NN-COSIM und MNB (P: 0,74, R: 0,72). Die Einbeziehung der Abrufhäufigkeiten begünstigt also die Verkettung.

Die höchste Genauigkeit wird erreicht, wenn alle Transformationsfunktionen angewendet werden und die Instanzvektoren mit Bigrammen angereichert werden.

Offene Frage: Wie gut funktioniert die Verkettung in der Praxis („Open-World“).

Die besten Klassifikatoren werden unter Realbedingungen evaluiert. Zur Messung der Genauigkeit wird ein Beurteilungsschema entwickelt.

Annahme: Alle Nutzer treten jeden Tag unter einer anderen IP-Adresse auf (d.h. Sitzungen dauern exakt 24 Stunden).

Realitätsnahe Problemstellung:
Der Beobachter versucht, die DNS-Anfragen, die er „heute" sieht, mit den Anfragen von „gestern" zu verketten – also chronologisch von einem auf den jeweils darauffolgenden Tag. Zur Evaluation wird dabei über den ganzen Datensatz iteriert (61 Tage).

Vorgehensweise: Ein Klassifikator wird mit den n Sitzungen der n Nutzer, die „gestern" aktiv waren, trainiert. Dann wird er benutzt, um für jede der m Sitzungen von „heute" die wahrscheinlichste Klasse c_i (die der Sitzung von Nutzer u_i am Vortag entspricht) aus den n Klassen zu bestimmen.

Beurteilungsschema: Enthält Epoche e_1 die Trainingsinstanzen und e_2 die damit zu verkettenden Instanzen, können folgende Fälle auftreten:

- Falls Nutzer u_i in beiden Epochen aktiv war, ist die Vorhersage **korrekt**, wenn nur die Sitzung von u_i aus e_2 der Klasse c_i zugewiesen wird – und keine Sitzungen anderer Nutzer.
- Falls u_i nur in e_1 aktiv war, ist die Vorhersage **korrekt**, wenn der Klasse c_i keine der Sitzungen aus e_2 zugewiesen wird.
- Ordnet der Klassifikator genau eine Sitzung aus e_2 der Klasse c_i zu, gehört diese aber zu einem anderen Nutzer u_a ($a \neq i$) ist es ein für den Beobachter **nicht-erkennbaren Fehler**.
- Ordnet der Klassifikator c_i keine Sitzung aus e_2 zu, obwohl u_i in e_2 aktiv war, ist es ebenfalls ein **nicht-erkennbarer Fehler**.
- Ordnet der Klassifikator c_i mehrere Sitzungen aus e_2 zu, ist es ein **erkennbarer Fehler**.

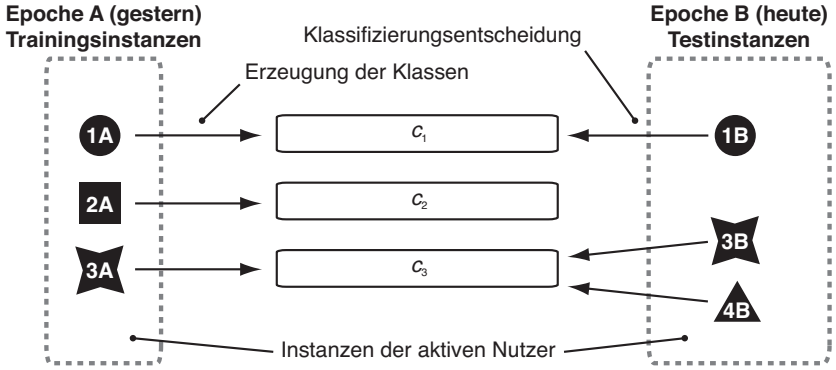
Bei der Open-World-Evalution ist mit einer natürlichen Nutzerfluktuation zu rechnen. Standardverfahren berücksichtigen dies nur unzureichend.

Die 1NN-COSIM- und MNB-Klassifikatoren erreichen im Open-World-Szenario eine Genauigkeit von **74,7%** (= Anteil korrekter Entscheidungen).

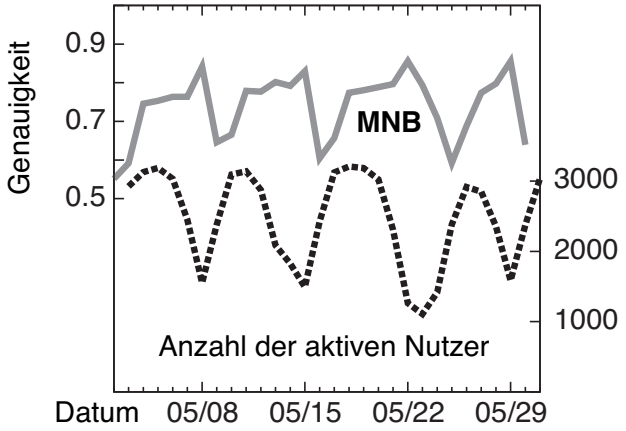
Beobachtung: Mehr als die Hälfte der Fehler sind erkennbare Fehler, also mehrdeutige Entscheidungen.

Diese entstehen, wenn Nutzer „heute“ aktiv sind, aber „gestern“ nicht anwesend waren (in der Abb. rechts bei Nutzer 4). Im Datensatz tritt dies v.a. nach Wochenenden auf.

Ursache: Klassifikator ordnet jede Sitzung von „heute“ stur der am besten passenden Klasse von „gestern“ zu.



Erkennbarer Fehler bei Standardvorgehen

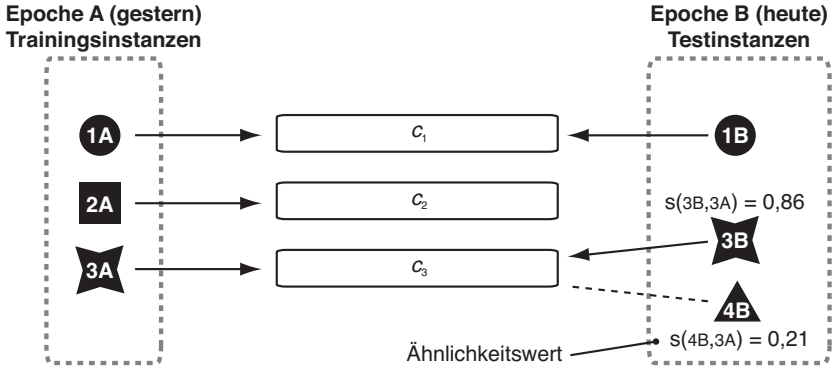


Durch geeignete Anpassungen lassen sich mehrdeutige Zuordnungen auflösen und somit die Genauigkeit des Verkettungsverfahrens erhöhen.

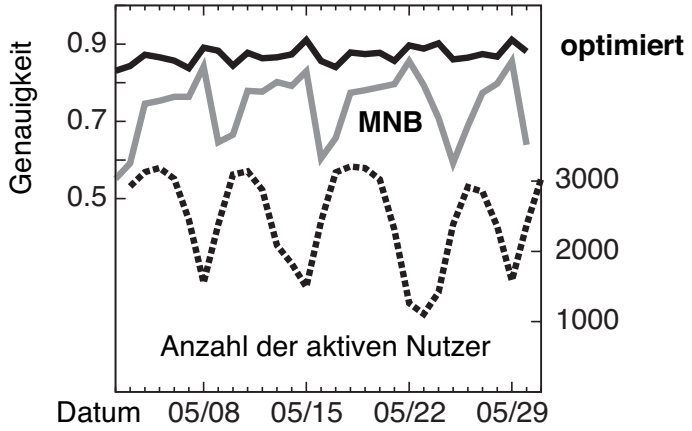
Lösungsansätze:

- Einführung eines **Schwellenwerts**, um bei zu geringer Ähnlichkeit von einer Verkettung abzusehen (sog. „Reject-Rule“; Nachteil: Kalibration erforderlich)
- **Besserer Ansatz:** Paarweise die COSIM-Ähnlichkeitswerte zwischen der betroffenen Trainingsinstanz (3A) von „gestern“ und allen ihr zugeordneten Sitzungen von „heute“ (3B, 4B) bestimmen, um die am besten passende Sitzung zu ermitteln; für die übrigen Sitzungen keine Vorhersage machen.

Wirksamkeit der Optimierung:
Die Genauigkeit steigt sowohl bei 1NN-COSIM als auch MNB auf **86%**.



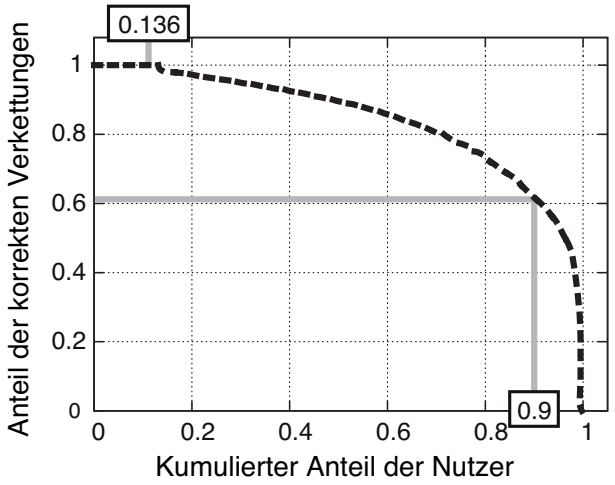
Anpassung: Auflösung mehrdeutiger Verkettungen



Die Verkettung gelingt allerdings nicht bei allen Nutzern gleichermaßen. In weiteren Experimenten wurden Einflussfaktoren analysiert.

Bei 13,6% der 3862 Nutzer traf das angepasste MNB-Verfahren ausschließlich korrekte Verkettungsvorhersagen. Bei 90% der Nutzer waren mindestens 60% der Vorhersagen korrekt (s. Abb.).

Als nächstes zu klären: Welche Einflussfaktoren bestimmen darüber, ob sich die Sitzungen eines Nutzers verketten lassen oder nicht?



Ergebnisse der Zusammenganganalyse mittels Spearman-Rangkorrelationskoeffizienten:

Die Genauigkeit, die bei einem Nutzer erzielt wird, steht in

schwach positivem Zusammenhang mit (1) der mittleren Anzahl der aktiven Stunden pro Tag, (2) der Anzahl der aktiven Tage, (3) der mittleren Anzahl der DNS-Anfragen pro Tag und (4) der mittleren Ähnlichkeit unmittelbar aufeinanderfolgener Sitzungen des Nutzers,

keinem Zusammenhang mit der mittleren Anzahl der unterschiedlichen Domains, die der Nutzer pro Tag auflöst, und

negativem Zusammenhang mit der Ähnlichkeit der Sitzung des Nutzers und Sitzungen anderer Nutzer, also bei geringer Individualität.

Einige Nutzer wären wegen benutzerspezifische Domains besonders leicht wiedererkennbar. Dadurch könnte die Genauigkeit weiter erhöht werden.

Eigenschaften benutzerspezifischer Domains:

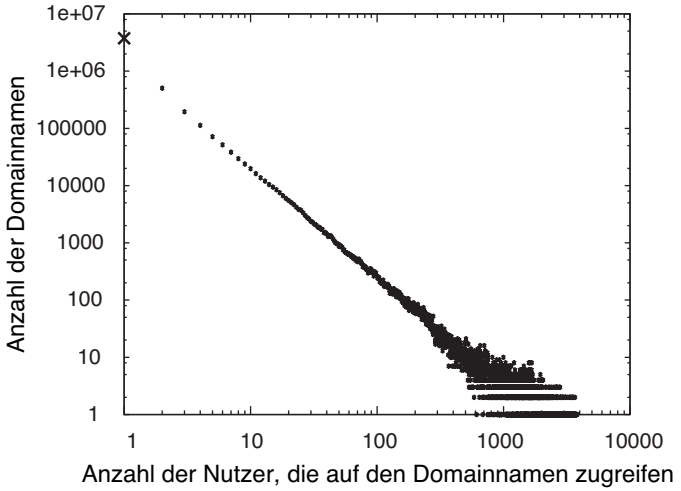
E1: Im gesamten Datensatz stellt ausschließlich **ein einziger Nutzer** Anfragen für diese Domain.

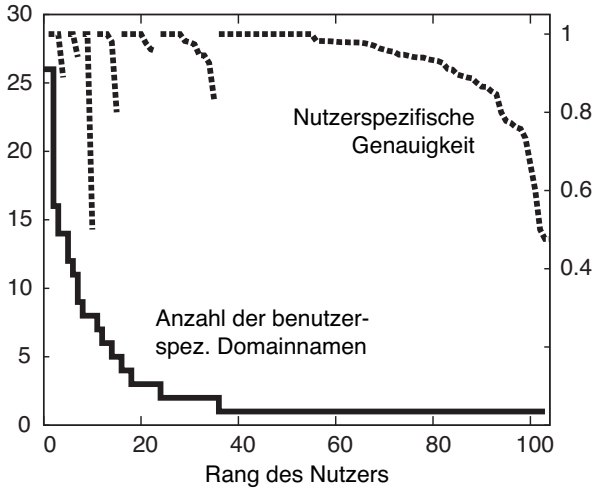
E2: Der Nutzer stellt **in jeder seiner Sitzun-gen** mind. eine Anfrage für die Domain.

Folge: Anhand einer **benutzerspezifischen Domain** kann ein Beobachter unmittelbar alle Sitzungen eines Nutzers verketten.

Mehr als 3,7 Mio. Domains in 2M erfüllen E1 (Abb. oben), 273 Domains E1 und E2. **103 Nutzer (3%) lösen mind. 1 benutzersp. Domain auf.** Beispiele: *BAUMAX.COM*, *api.gubble.com*, *mosesonline.com*

Klassifikator weiß jedoch nicht, um welche Domains es sich handelt; daher gelingt ihm die Erkettung dieser Nutzer trotzdem nicht zuverlässig (Abb. unten).





Verhaltensbasierte Verkettung ist robust und funktioniert auch bei größeren Nutzergruppen – entscheidende Eigenschaften für den Praxiseinsatz.

Wie gut gelingt die Verkettung, wenn nur die populärsten Domains zur Verfügung stehen? Wenn der DNS-Server nur die 100 bzw. 1000 am häufigsten angefragten Domains erfährt, beträgt die Genauigkeit bereits 61,7% bzw. 82,4%.

Wie gut gelingt die Verkettung bei größeren Datensätzen? Wiederholung des Open-World-Experiments mit dem Datensatz 2M* (12.015 Nutzer): Es wird immer noch eine Genauigkeit von **75,9%** erreicht.

Wie gut gelingt die Verkettung, wenn Training mehr als 1 Tag zurückliegt? Genauigkeit sinkt in den ersten Tagen nur geringfügig; nach 28 Tagen werden noch 76,3% erreicht.

Das vorgestellte Verkettungsverfahren ist nicht auf DNS beschränkt, sondern **auf andere Anwendungen übertragbar**, etwa

- HTTP-Proxy-Server, Anonymisierungsdienste, öffentliche WLAN-Hotspots,
- Werbenetze (Google, Facebook, etc.), die Nutzer nach dem Löschen von Cookies wiedererkennen wollen.

Szenario in der IT-Forensik: Ermittler haben mittels TKÜ den Datenverkehr einer Sitzung aufgezeichnet, in der eine strafbare Handlung begangen wurde. Allerdings ist die Identität des Täters noch unbekannt. Gelingt den Ermittlern die Verkettung mit einer anderen Sitzung, die einem Verdächtigen zugerechnet werden kann, kann dies die Überführung begünstigen.

Abschließend werden in der Dissertation Selbstdatenschutz-Techniken vorgeschlagen, um die verhaltensbasierte Verkettung zu unterbinden.

Die vorgestellte Selbstdatenschutz-Technik DNSMIX sowie Anonymitätsdienste wie Tor können grundsätzlich ebenfalls vor verhaltensbasierter Verkettung schützen.

Im Folgenden werden jedoch insbesondere Techniken untersucht, **die ausschließlich vor der verhaltensbasierter Verkettung** von Sitzungen schützen.

Motivation: Vielen Internetnutzern ist heute durchaus bewusst, dass ihre Aktivitäten *innerhalb* einer Sitzung von Dritten beobachtet werden können. Sie sind jedoch nicht dazu bereit, den Kontrollverlust hinzunehmen, der aus der Verknüpfung von Informationen über mehrere Sitzungen hinweg resultieren würde.

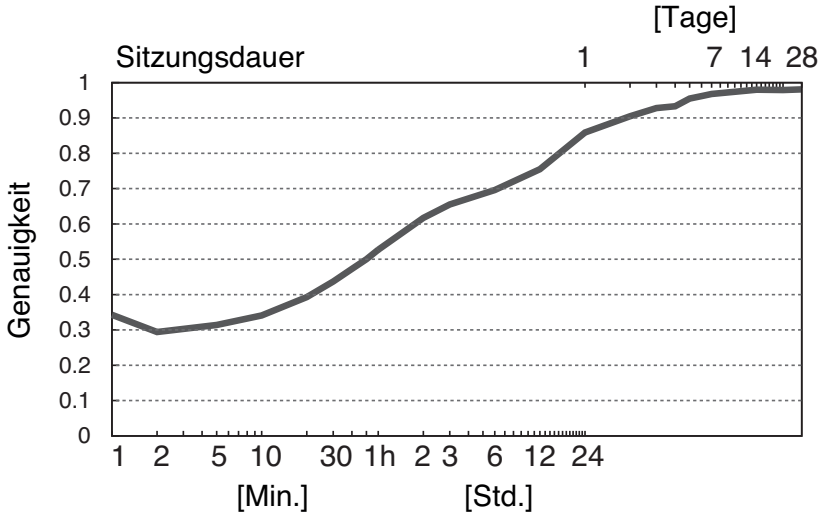
Während zahlreiche Nutzer Cookies im Browser löschen und Privacy-Tools nutzen, um eine langfristige Verkettung ihrer Aktivitäten zu verhindern, nehmen nur wenige Nutzer die Komforteinbußen und die Verzögerungen in Kauf, die bei der Verwendung eines Anonymitätsdienstes entstehen, der die Beobachtung durch den Internetanbieter oder Dritte verhindern würde.

Es werden drei Techniken untersucht. Insbesondere die Verkürzung der Sitzungsdauer durch häufiges Wechseln der IP-Adresse ist vielversprechend.

1. Das Übermitteln von Dummy-Anfragen für zufällige Domains erweist sich im Test als wenig wirksam: Die Genauigkeit der Verkettung sinkt nur um 1 Prozentpunkt. Ursache: Die IDF-Transformation eliminiert den störenden Einfluss der zufälligen Anfragen.

2. Clients könnten die TTL ignorieren und die **abgefragten Informationen länger als vorgesehen zwischenspeichern**, um die Anfragehäufigkeiten vor dem Beobachter zu verbergen. Diese Maßnahme zeigt Wirkung: bei einer Caching-Dauer von 24h werden nur noch 25,2% der Anfragen an den DNS-Server übermittelt und die Genauigkeit sinkt auf 46,4%. Nachteil dieses Ansatzes: mitunter werden veraltete und daher nicht mehr erreichbare IP-Adressen verwendet.

3. Die Verkürzung der Sitzungsdauer hat den größten Effekt. Bei stündlichem Wechsel waren noch 55% der 1-stündigen Sitzungen verkettbar; wird die IP-Adresse alle 5 Minuten gewechselt, sinkt die Genauigkeit weiter auf 31,4%. Bei den verbreiteten IPv4-Internetanschlüssen geht der Wechsel der IP-Adresse allerdings mit einer kurzen Verbindungsunterbrechung einher.



Die Ergebnisse der Dissertation sind für Nutzer von DNS-Fremdanbietern, Software-Hersteller, IT-Forensiker und Datenschützer von Bedeutung.

Ergebnisse

Wegen charakteristischer Abrufmuster bzw. unterschiedlicher Implementierungen kann ein DNS-Server u.U. die von einem Nutzer **besuchten Webseiten** bzw. **eingesetzte Software** identifizieren.

Das entwickelte Verfahren zur verhaltensbasierten Verkettung ermöglicht die **Nachverfolgung von Nutzeraktivitäten** über lange Zeiträume.

Unbeobachtbarkeit ist im DNS mit einem Push-System für einen Großteil der Domains mit vertretbarem Aufwand realisierbar. Mit **auf DNS abgestimmten Mix-Systemen** können die übrigen Domains mit geringer Verzögerung aufgelöst werden können. Die Verschleierung der beabsichtigten Anfragen mittels **Range Querys** ist hingegen weniger effektiv als bislang angenommen.

Schlussfolgerungen

DNS-Server verfügen über **Beobachtungsmöglichkeiten**, welche die Privatsphäre der Nutzer bzw. die Sicherheit ihrer IT-Systeme bedrohen. DNS-Logs eignen sich für **forensische Analysen**.

Täglich wechselnde IP-Adressen bzw. tägliches Löschen von Cookies bieten **weit weniger Schutz** vor Tracking im Internet als bisher angenommen. Problematisch erscheint, dass im Unterschied zu heute verbreiteten Tracking-Techniken (z.B. Browser-Fingerprinting) die **verhaltensbasierte Verkettung clientseitig nicht erkennbar** ist.

Wie das DNSMIX-System zeigt, kann sich neben der Verbesserung generischer Systeme wie Tor die Erforschung und Entwicklung von **dienstspezifischen Datenschutz-Techniken** lohnen.

Aus der Dissertation lassen sich praxisrelevante Handlungsempfehlungen und Ansätze für zukünftige Forschungsvorhaben ableiten.

Empfehlungen für Nutzer, die großen Wert auf ihre Privatsphäre legen

- DNS-Fremdanbieter möglichst meiden; sonst: möglichst im Router und nicht im Betriebssystem eintragen, um unnötige Preisgabe interner Domains zu vermeiden,
- Rechnername und Domain-Suffix so wählen, dass sie nicht die eigene Identität preisgeben,
- präemptive Namensauflösung im Browser deaktivieren; ggf. Adblock-Plugin nutzen,
- DSL-Router mit eigenem DNS-Cache einsetzen,
- IP-Adresse häufig wechseln (gleichzeitig Cookies und Browser-Cache löschen).

Bei hohem Schutzbedarf: eigenen (internen) DNS-Server betreiben, der so konfiguriert ist, dass sensible Namen nicht an fremde DNS-Server übermitteltwerden (Response Policy Zones).

Empfehlungen an Software-Entwickler, Internet-Provider und Standardisierer

- Mit IPv6-Internetzugängen ließe sich das häufige Wechseln der IP-Adresse benutzerfreundlich realisieren und somit ein effizienter Schutz vor verhaltensbasierter Verkettung erreichen. Dabei müssen **gleichzeitig Präfix und Interface Identifier** ersetzt werden (vgl. Empfehlungen in der IPv6-Orientierungshilfe der Landesdatenschutzbeauftragten). Hersteller und Internet-Provider sollten dazu ermuntert werden, entsprechende Produkte anzubieten.
- Bei der Standardisierung von neuen Protokollen sollte überprüft werden, ob unerwünschte Seitenkanäle existieren bzw. ob durch Fingerprinting-Verfahren sensible Informationen gewonnen werden können.

Literaturverzeichnis des Handouts

S. Bortzmeyer, DNS privacy problem statement, draft-bortzmeyer- perpass-dns-privacy-01, RFC Editor, 2013.

C. Brandhorst and A. Pras, DNS: A Statistical Analysis of Name Server Traffic at Local Network-to-Internet Connections, in: EUNICE 2005, IFIP International Workshop on Networked Applications. Proceedings, 2006, 255–270.

T. Callahan, M. Allman, and M. Rabinovich, On Modern DNS Behavior and Properties, in: Computer Communication Review 43.3 (2013) 7–15.

D. Conrad, Towards Improving DNS Security, Stability, and Resiliency, Internet Society, 2012, URL: http://www.internetsociety.org/sites/default/files/bp-dnsresiliency-201201-en_0.pdf

M. Handley and A. Greenhalgh, The case for pushing DNS, in: 4th ACM Workshop on Hot Topics in Networks (Hotnets-IV). Proceedings, 2005.

D. Herrmann, M. Maaß, H. Federrath: Evaluating the Security of a DNS Query Obfuscation Scheme for Private Web Surfing, in: IFIP SEC 2014, AICT 428, Springer, 2014, 205–219.

Y. Lu, G. Tsudik. Towards Plugging Privacy Leaks in Domain Name System, 2009 URL: http://arxiv.org/abs/0910.2472

Y. Yang, Web user behavioral profiling for user identification,in: Decision Support Systems 49 (2010) 261–271.

F. Zhao, Y. Hori, and K. Sakurai, Analysis of Privacy Disclosure in DNS Query, in: International Conference on Multimedia and Ubiquitous Engineering (MUE 2007). Proceedings, IEEE, 2007.