Hannes Federrath, Christoph Gerber, Dominik Herrmann

Verhaltensbasierte Verkettung von Internetsitzungen

Die Analyse der Aktivitäten eines Internet-Nutzers ist über längere Zeit in der Regel nur mit Kenntnis der bei vielen Internet-Zugangsprovidern mindestens einmal täglich wechselnden IP-Adressen möglich. Verwendet der Nutzer zudem einen Anonymisierungsdienst, sollte eine Verkettung der Nutzer-Aktivitäten auch für den Zugangsprovider unmöglich sein. Der Beitrag zeigt, dass eine verhaltensbasierte Verkettung dennoch möglich ist und verbreitete Anonymisierungslösungen in der Praxis einen deutlich geringeren Schutz vor Beobachtung bieten, als bisher angenommen.

Einleitung

In diesem Beitrag stellen wir einen Verkettungsangriff vor, mit dem u. a. Web-Proxy-Server und darauf basierende Anonymisierungssysteme mehrere unabhängige Internetsitzungen ihrer Nutzer ohne deren Wissen und nur durch passives Beobachten verketten können. Dadurch können sie die Online-Aktivitäten ihrer Nutzer über längere Zeiträume als vorgesehen beobachten und umfangreiche Interessenprofile erstellen. Durch eine fortlaufende Verkettung von Internetsitzungen kann mitunter sogar die wahre Identität eines Nutzers aufgedeckt werden, wie die Analyse der Log-Dateien der AOL-Suchmaschine ergeben hat [1].

Die verwendeten Techniken zeichnen sich dadurch aus, dass sie nicht auf die bekannten, expliziten Verkettungsmerkmale wie z. B. Browser-Cookies zurückgreifen. Stattdessen werden implizite Merkmale ausgenutzt, die sich aus dem charakteristischen, gewohnheitsmäßigen Verhalten von Benutzern ergeben.

Prof. Dr. Hannes Federrath

Leiter des Arbeitsbereichs „Sicherheit in Verteilten Systemen" am Fachbereich Informatik der Universität Hamburg
E-Mail: federrath@informatik.uni-hamburg.de

Christoph Gerber, M.A.

Wissenschaftlicher Mitarbeiter am Arbeitsbereich Sicherheit in Verteilten Systemen
E-Mail: gerber@informatik.uni-hamburg.de

Dipl.-Wirtsch.-Inf. Dominik Herrmann

Wissenschaftlicher Mitarbeiter am Arbeitsbereich Sicherheit in Verteilten Systemen
E-Mail: herrmann@informatik.uni-hamburg.de

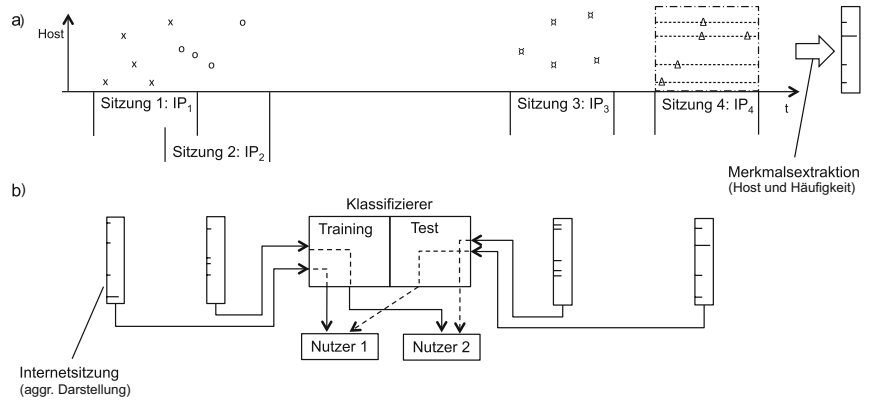
1 Privatsphäre durch dynamische IP-Adressen

Die Furcht vor unbemerkter Langzeit-Beobachtung im Internet wird auch im Zuge der Berichterstattung über die unmittelbar bevorstehende Einführung von IPv6 deutlich: Noch bevor die Internetzugangsanbieter ihre Realisierungspläne öffentlich vorstellten, wurde bereits vor den neuen Beobachtungsmöglichkeiten gewarnt, die sich womöglich durch die Umstellung ergeben könnten. Im Kern geht es dabei um die Frage, wie lange einem Internetnutzer eine bestimmte IP-Adresse zugewiesen wird. Der Hamburgische Beauftragte für Datenschutz und Informationsfreiheit Johannes Caspar äußerte in einer Pressemitteilung am 07.06.11 die Befürchtung, dass der riesige IPv6-Adressraum die Zugangsanbieter dazu verleiten könnte, jedem Nutzer eine eigene, „lebenslang" gültige IPv6-Adresse – genauer: ein eindeutiges Adress-Präfix – zuzuweisen. Weiterhin sah das IPv6-Protokoll ursprünglich vor, dass sich jedes Endgerät ein weltweit eindeutiges Adress-Suffix zuweist. Da die meisten Router für Heimnetzwerke die Network Address Translation (NAT) bei der Verwendung von IPv6 voraussichtlich nicht mehr unterstützen werden, wären sowohl die eindeutige Präfixe als auch eindeutige Suffixe unmittelbar dazu geeignet, Internetnutzer anhand ihrer IPv6-Adresse wiederzuerkennen. Dienstanbieter könnten dadurch wesentlich leichter als heute Nutzungsprofile erstellen.

Bei den heute noch verbreiteten IPv4-Adressen ist die Wiedererkennung von Nutzern anhand der IP-Adresse hingegen weniger aussichtsreich, da die meisten Zugangsanbieter ihren Kunden aus historischen Gründen bei jeder Einwahl, spätestens jedoch nach 24 Stunden, eine neue zufällige Adresse aus ihrem Adressbereich zuweisen. Außenstehenden ist es somit nicht ohne weiteres möglich, die Aktivitäten einzelner Nutzer über längere Zeit zu beobachten.

Eine Weiterführung der dynamischen Zuteilung der IP-Adresse bei IPv6 sollte nach Auffassung von Caspar am besten gesetzlich erzwungen werden. Einige

Abbildung 1 | Der Verkettungsangriff im Überblick



Zugangsanbieter haben inzwischen allerdings angekündigt, ohnehin an der täglichen Zwangstrennung festhalten zu wollen und jedem Kunden dabei ein neues Adress-Präfix zuzuweisen. Um das Suffix müssen sich die Nutzer selbst kümmern. Mit den sog. IPv6 „Privacy Extensions" [2], die von den meisten Betriebssystemen unterstützt werden, gibt es inzwischen einen Mechanismus, der zufällige und unverkettbare Suffixe erzeugt. Im Ergebnis sind dann nur noch die Aktivitäten innerhalb eines Tages anhand der IPv6-Adresse verkettbar – also genau so wie es heute bei den meisten Zugangsanbietern üblich ist.

Der in diesem Beitrag beschriebene Verkettungsangriff soll zeigen, dass der tägliche Wechsel der IP-Adresse mitunter keinen zuverlässigen Schutz vor Langzeitbeobachtungen bietet. Der Angriff versetzt Außenstehende in die Lage, die einzelnen Internetsitzungen eines Benutzers trotz dynamischer IP-Adressen zu verketten.

2 Angriffsszenario

Der Verkettungsangriff erfolgt rein passiv, d. h. es werden keine Veränderungen an der von den Benutzern verwendeten Software bzw. am Datenverkehr vorgenommen. Voraussetzung für den Angriff ist lediglich, dass der Angreifer die Domain-Namen der Internet-Server erfährt, die seine Nutzer innerhalb einer Internetsitzung kontaktieren. Über dieses Wissen verfügen neben dem Internetzugangsanbieter, der seine Kunden jedoch ohnehin überwachen kann, zum Beispiel die Betreiber von Web-Proxy-Servern bzw. proxy-basierten Anonymisierungsdiensten. Auch die Anbieter diverser Browser-Toolbars (z. B. *Alexa* und *Web of Trust*) haben Zugriff auf die besuchten Web-Server. Eine weitere Gruppe von Anbietern, die den Verkettungsangriff durchführen kann, um Nutzer zu überwachen, stellen alternative DNS-Resolver wie *OpenDNS* oder *Google Public DNS* dar.

Der Angriff basiert auf der These, dass Menschen auch im Internet „Gewohnheitstiere" sind und dort ihre individuellen Vorlieben ausleben. Dies führt zum einen dazu, dass viele Nutzer immer wieder dieselben Webseiten ansteuern. Zum anderen ist zu erwarten, dass jeder Nutzer eine charakteristische Kombination von persönlichen Lieblingsseiten hat, die sonst niemand außer ihm besitzt. Sind die Interessen der Nutzer ausreichend unterschiedlich ausgeprägt und ist die Nutzungsintensität ausreichend groß, könnten sich die Internetsitzungen verschiedener Nutzer allein anhand der besuchten Seiten auseinanderhalten lassen.

Abbildung 1a veranschaulicht die Sicht des Angreifers anhand zweier Nutzer. Die Darstellung visualisiert die von den Nutzern besuchten Webseiten im Zeitverlauf. Die Hosts werden auf der y-Achse des Diagramms in einer indizierten Liste dargestellt, die alle Hosts enthält, die irgendwann einmal besucht wurden. Im Diagramm sind vier Sitzungen erkennbar, in denen unterschiedliche Hosts besucht wurden. Der für eine Sitzung verantwortliche Nutzer verbirgt sich aus Sicht des Angreifers hinter einer dynamisch zugewiesenen IP-Adresse. Sitzung 1 und 2 stammen im Beispiel von den Nutzern 1 und 2. Zu einem späteren Zeitpunkt beobachtet der Angreifer zwei weitere Sitzungen 3 und 4, wobei zunächst nicht bekannt ist, wie die korrekte Zuordnung aussieht. Im Beispiel gehört die Sitzung 4 zum Nutzer 1 und die Sitzung 3 zu Nutzer 2. Genau diese Zuordnung wird durch den Verkettungsangriff hergestellt.

3 Stand der Technik

Angriffe auf die Privatsphäre von Benutzern durch Deanonymisierung bzw. Verkettung ihrer Aktivitäten sind Gegenstand zahlreicher Untersuchungen [3, 4, 5, 6, 7, 8]. Häufig werden hierzu Techniken des maschinellen Lernens verwendet. Es gibt bereits einige Arbeiten, die ein ähnliches Ziel verfolgen wie unser Verkettungsangriff, d. h. die Wiedererkennung von Benutzern durch die Analyse ihres Datenverkehrs. Allerdings unterscheiden sie sich in wesentlichen Aspekten von unserem Verfahren und machen keine Aussagen über die Praktikabilität des Angriffs.

Pang et al. [9] versuchen, Benutzersitzungen in 802.11-Funknetzwerken zu verketten. Hierzu ziehen sie jedoch vor allem spezifische Charakteristika von WiFi-Geräten heran, so dass der Angriff lediglich in WLANs funktioniert. Yang [10, 11, 12] verwendet Benutzerprofile, die auf den besuchten Internetseiten basieren, um Betrugsversuche zu erkennen. Sie macht jedoch keine Aussagen dazu, inwiefern ein Angreifer solche Benutzersitzungen verketten kann. Ein mit unserem Angriff vergleichbares Szenario wird von Kumpost [13, 14, 15] untersucht. Während wir uns auf die Verkettung kurzer Internetseitzungen beschränken, arbeitet er mit monatlich aggregierten NetFlow-Log-Dateien, die dem von uns betrachteten Angreifer nicht zur Verfügung stehen.

4 Verkettungsangriff

Um Sitzungen auf Basis der Log-Dateien zu verketten, muss der Angreifer geeignete Merkmale extrahieren, die einerseits die charakteristischen Eigenschaften der Benutzer widerspiegeln, gleichzeitig jedoch generisch genug sind, um auch bei geringfügigen Verhaltensänderungen die Sitzungen eines Nutzers noch zuverlässig wiederzuerkennen.

Unser Verfahren zieht daher ausschließlich die Host-Namen (z. B. www.faz.de) und die jeweils darauf entfallende Anzahl der HTTP-Anfragen innerhalb einer Sitzung eines Nutzers heran. Jede Sitzungen gibt demnach über die Nutzungsintensität einzelner Webseiten Auskunft; die Rei-

henfolge und die genauen Zeitpunkte der Besuche werden jedoch ignoriert. Eine Sitzung s eines Nutzers u lässt sich formal als Multimenge darstellen, in der bei insgesamt n Hosts für jeden Host-Namen h_i die Anzahl f_i der darauf entfallenden HTTP-Anfragen hinterlegt ist:

$$s_u = (h_1 \Rightarrow f_1, h_2 \Rightarrow f_2, ..., h_n \Rightarrow f_n).$$

5 Modellierung

Die Verkettung von zwei Internetsitzungen entspricht in der Terminologie des maschinellen Lernens einem sogenannten Klassifizierungsproblem [16], bei dem Instanzen (Sitzungen) $x \in X$, die durch einen Merkmalsvektor dargestellt werden, mit Hilfe eines Klassifizierers einer von mehreren Klassen (Nutzern) $c \in C$ zugeordnet werden. Dazu erzeugt der Klassifizierer im ersten Schritt anhand von manuell vorklassifizierten Trainingsdaten – im Beispiel in Abbildung 1a wären das die Sitzungen 1 und 2 – ein Modell (Training), das die charakteristischen Eigenschaften der Klassen enthält. Anhand des gelernten Modells werden im zweiten Schritt (Test) die Testinstanzen den jeweils wahrscheinlichsten Klassen zugeordnet.

Abbildung 1b veranschaulicht die beiden Schritte des Verfahrens am Beispiel zweier Benutzer. Die Länge der Linien innerhalb eines „Blocks“ (aggregierte Darstellung) in Abbildung 1b entspricht der Auftretenshäufigkeit f des jeweiligen Hosts.

Wir verwenden den Naive-Bayes-Klassifizierer (NB-Klassifizierer), der häufig zur Kategorisierung von Texten (z. B. in einem Spam-Filter) eingesetzt wird [17]. Im Vergleich zu leistungsfähigeren Klassifizierern (z. B. *Support Vector Machines*) weist die von uns eingesetzte multinomiale Variante des NB-Klassifizierers auch bei einer sehr großen Anzahl von Attributen (Hosts) und Klassen (Nutzern) erheblich geringere Laufzeiten auf. Beim multinomialen NB-Klassifizierer ergibt sich das gelernte Modell unmittelbar aus den Auftretenshäufigkeiten der Terme. Die Worthäufigkeitsverteilung unterliegt bei natürlichsprachlichen Texten i. d. R. einem so genannten *Power Law* [18]. In Studien wurde gezeigt, dass auch die Häufigkeiten der Internet-Hosts einer solchen Verteilung unterliegen [19].

Charakteristisch für die Power-Law-Verteilung ist, dass es eine sehr geringe Anzahl von Merkmalen (hier: Hosts) gibt, die überproportional häufig auftreten, und die Auftretenshäufigkeiten der verbleibenden Merkmale extrem schnell (exponentiell) abfallen. Die Häufigkeitsvektoren sind dadurch weitgehend leer *(sparse)*.

Bei m unterschiedlichen Hosts in den Trainingsinstanzen ermittelt der Klassifizierer die Wahrscheinlichkeit, dass eine Instanz x zu einer Klasse c gehört, wie in folgender Formel ersichtlich:

$$P(x \mid c_i) \sim \prod_{j=1}^{m} P(X = h_j \mid c_i)^{f_j}$$

Das Ergebnis $P(x \mid c_i)$ ist proportional zum Produkt der Einzelwahrscheinlichkeiten $P(X = h_j \mid c_i)$, die der Wahrscheinlichkeit entsprechen, dass ein bestimmter Host h_j zufällig aus der aggregierten Multimenge aller Host-Abrufe der Trainingsinstanzen gezogen wird, die für Klasse c_i vorlagen. Diese Einzelwahrscheinlichkeiten werden mit f_j, der Anzahl der Anfragen, die in der Testinstanz für Host h_j enthalten sind, gewichtet. Je häufiger die häufigsten Hosts der Testinstanz x also in der Trainingsinstanz der Klasse c_i auftreten, desto wahrscheinlicher gehört x zur Klasse c_i.

Um die Zuordnung einer Testinstanz zu einer Klasse vorzunehmen, ermittelt der Klassifizierer $P(x \mid c_i)$ für alle trainierten Klassen. Er ordnet der Testinstanz dann die Klasse zu, die den höchsten Wahrscheinlichkeitswert $P(x \mid c_i)$ aufweist. Die Power-Law-Verteilung führt dabei zu einer unerwünschten Verzerrung der einzelnen Instanzen, welche eine korrekte Zuordnung erschwert. Beim *Text Mining* werden daher verschiedene Transformationen auf die absoluten Auftretenshäufigkeiten angewendet, die den Einfluss der Verzerrung abmildern [17]. Die absoluten Abrufhäufigkeiten werden dazu mit der *Term Frequency Transformation* (TF) logarithmiert.

Mit der *Inverse Frequency Transformation* (IDF) wird der Einfluss von Hosts, die von einer Vielzahl von Nutzern besucht werden, reduziert, und das Gewicht der Hosts, die von sehr wenigen Nutzern besucht werden, erhöht. Schließlich hat es sich bewährt, die Merkmalsvektoren mittels *Cosine Normalisation* (N) auf eine einheitliche Länge zu skalieren. Wir ermittelten den Nutzen dieser Transformationen für den Verkettungsangriff im Rahmen einer empirischen Untersuchung (siehe Tabelle 1).

Tabelle 1 | Evaluationsergebnisse

Transformationsfunktion	Wiedererkennungsrate
ohne	60 %
N	62 %
IDF	65 %
IDF + N	62 %
TF	56 %
TF + N	73 %
TF + IDF	66 %
TF + IDF + N	72 %

6 Evaluation

Um die Wirksamkeit unseres Verkettungsangriffs zu untersuchen, haben wir ihn mit dem Data-Mining-Toolkit Weka [16, 20] implementiert und auf den Datenverkehr von echten Internetsitzungen angewendet. Hierzu wurde im Rahmen einer Studie der Datenverkehr von 28 Nutzern über einen Zeitraum von 57 Tagen aufgezeichnet.

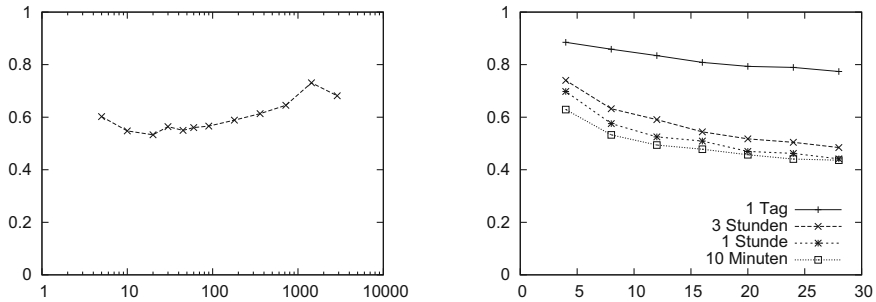
Alle Teilnehmer der Studie zeichneten ihre HTTP-Anfragen mit Hilfe eines lokal installierten Proxy-Servers auf. Um die Privatsphäre der Studienteilnehmer zu wahren, wurden die Host-Namen noch in deren Schutzbereich mittels einer Hashfunktion unkenntlich gemacht. Durch das Hashen wird die Vergleichbarkeit mit Daten anderer Teilnehmer jedoch nicht beeinträchtigt. Von jedem Teilnehmer liegen die gehashten Host-Namen und Abrufzeitpunkte aller HTTP-Anfragen im Beobachtungszeitraum vor. Zur Simulation von Internetsitzungen lassen sich die Datensätze der Teilnehmer in entsprechende Teilmengen zerlegen.

6.1 Evaluation des Angriffs unter Realbedingungen

Zunächst soll untersucht werden, inwiefern ein angreifender Proxy-Server in der Lage ist, die Sitzungen der Studienteilnehmer an aufeinanderfolgenden Tagen zu verketten, wenn die IP-Adresse der Nutzer täglich wechselt. Dazu wird auf den erhobenen Daten die tägliche Zwangstrennung von DSL-Anbietern simuliert, d. h. die Sitzungen aller Nutzer dauern 24 Stunden. Da die tatsächliche Zuordnung der Sitzungen zu den Nutzern bekannt ist, kann die Leistungsfähigkeit des Klassifizierers somit quantitativ bewertet werden.

Zur Evaluation der Effektivität des Verkettungsangriffs wird über alle Nutzer und deren 24-Stunden-Sitzungen iteriert

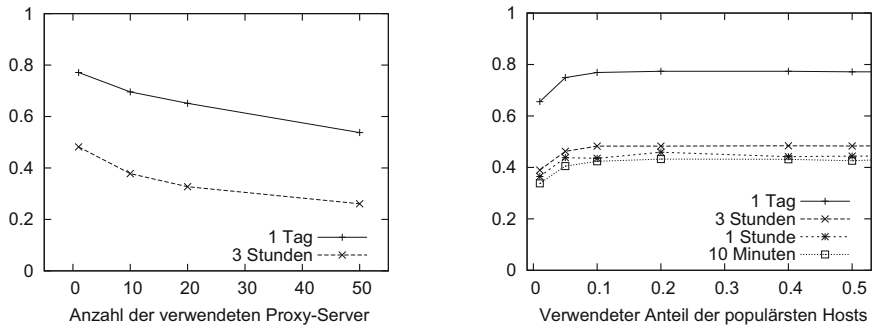
Abbildung 2 | Einflussfaktoren für die Verkettungsgenauigkeit



(a) Sitzungsdauer [Minuten]

(b) Anzahl der aktiven Nutzer

Abbildung 3 | Wirksamkeit von Gegenmaßnahmen



(a) Verteilung der Anfragen

(b) Beschränkung auf populärste Hosts

und geprüft, ob der Betreiber des Proxy-Servers eine Internetsitzung des Nutzers *u* am Tag *t* mit der Internetsitzung desselben Nutzers *u* am darauf folgenden Tag $t+1$ korrekt verkettet hätte. Dabei wird für jeden Nutzer, der an Tag *t* aktiv war, eine Klasse angelegt und der Klassifizierer mit der zugehörigen Internetsitzung von Tag *t* trainiert. Genau eine Internetsitzung gehört also zu dem Nutzer, der in dieser Iteration beobachtet werden soll; die Sitzungen der anderen Nutzer sind in dieser Iteration für den simulierten Angreifer nicht von Interesse. Mit Hilfe des gelernten Modells ordnet der Klassifizierer dann jede Instanz des Folgetages $t+1$ der jeweils wahrscheinlichsten Klasse zu.

Eine korrekte Zuordnung liegt in der Iteration immer dann vor, wenn der Klassifizierer die Sitzung des Nutzers *u* am Tag $t+1$ der Klasse zuordnet, der auch die Sitzung vom Vortag angehört, und keine weitere Instanz dieser Klasse zugeordnet wird. Falls ein Nutzer am Tag $t+1$ nicht mehr aktiv war, handelt es sich um eine korrekte Zuordnung, wenn der Klassifizierer keine Sitzung von $t+1$ der Klasse des an Tag $t+1$ inaktiven Nutzers zugeordnet hat.

Falls der Klassifizierer genau eine falsche Instanz der Klasse des Nutzers *u* zugeordnet, handelt es sich um einen nicht-erkennbaren Fehler, falls er mehrere Instanzen der relevanten Klasse zuordnet, handelt es sich um einen erkennbaren Fehler.

Wir haben dieses Evaluationsverfahren für alle Kombinationen der Transformationen durchgeführt. Die Ergebnisse sind in Tabelle 1 aufgeführt. Angegeben wird der Anteil der korrekten Zuordnungen (Genauigkeit).

Die Ergebnisse der Untersuchungen zeigen, dass das vorgestellte Verfahren auf Basis des multinomialen Naive-Bayes-Klassifizierers einen Großteil der unmittelbar aufeinanderfolgenden Internetsitzungen der Teilnehmer verketten kann. Bei dem in diesem Abschnitt simulierten Angriffsszenario wurden bis zu 73,1 % der Sitzungspaare korrekt zugeordnet, wenn die TFN-Transformation angewendet wurde. Die Anwendung der IDF-Transformation brachte in diesem Experiment keinen messbaren Vorteil.

6.2 Verifikation mit einem größeren Testdatensatz

Die im vorherigen Abschnitt vorgestellten Ergebnisse beruhen auf einer vergleichsweise kleinen Studie. Sie haben keine Aussagekraft für größere Nutzergruppen. Wir haben den Verkettungsangriff daher zusätzlich auf einem größeren Datensatz durchgeführt. Dabei handelt es sich um die Log-Dateien der DNS-Server einer deutschen Universität, welche die DNS-Anfragen von 3.216 pseudonymisierten Nutzern über einen Zeitraum von 159 Tagen enthalten. Als Merkmal wurde wieder die Häufigkeitsverteilung der Host-Namen verwendet.

Die Ergebnisse lagen in einer vergleichbaren Größenordnung: 66,5 % der Sitzungen wurden bei diesem Datensatz korrekt verkettet, wenn die TFN-Transformation angewendet wurde. Auch Nutzer von DNS-Servern sind also erstens von dem Verkettungsangriff betroffen und zweitens konnten wir auch in einem deutlich größeren Szenario die Ergebnisse der kleineren Studie bestätigen. Jede zweite Internetsitzung wäre selbst bei konsequenter Verwendung eines Proxies oder proxy-basierten Anonymisierers unter dem gegebenen Angreifermodell mit ihrem Nachfolger verkettbar.

6.3 Einflussgrößen

Ausgehend von der Untersuchung der Wirksamkeit unter möglichst realen Bedingungen wurden anhand des kleineren Datensatzes einige Einflussfaktoren in Simulationen genauer untersucht.

Abbildung 2 zeigt den Einfluss der Sitzungsdauer (a) sowie der Anzahl der gleichzeitigen Nutzer (b) in Bezug auf die Verkettbarkeit unmittelbar aufeinanderfolgender Sitzungen. In Abbildung 2a ist – nicht überraschend – ersichtlich, dass sich längere Internetsitzungen tendenziell besser verketten lassen.

Allerdings steigt die Verkettbarkeit auch bei besonders kurzen Sitzungen mit weniger als zehn Minuten wieder an. Dies lässt sich darauf zurückführen, dass bei kurzen Sitzungen die Wahrscheinlichkeit höher ist, dass sich ein Nutzer in zwei aufeinanderfolgenden Sitzungen auf denselben Seiten aufhält.

Die Verkettbarkeit hängt nur geringfügig von der Anzahl der gleichzeitig aktiven Nutzer ab, wie Abbildung 2b für verschieden lange Sitzungen zeigt. Dieses Ergebnis macht deutlich, dass übliche Anonymitäts- und Unbeobachtkeitsmaße, die auf der Größe der Anonymitätsgruppe beruhen, keine große Aussagekraft hinsichtlich des tatsächlich erreichten Schutzes haben müssen.

Die Verkettung gelingt auch dann, wenn die Sitzungen nicht unmittelbar aufeinanderfolgen. Bei einem Abstand von fünf Tagen beträgt die Genauigkeit noch 65,3 %.

7 Gegenmaßnahmen

Eine nahe liegende Gegenmaßnahme wäre die Verkürzung des Zeitraums, in dem Nutzer unter derselben IP-Adresse agieren. Dies geht jedoch zwangsläufig mit einem gewissen Komfortverlust einher, da bei jedem Wechsel der IP-Adresse existierende TCP-Verbindungen getrennt werden. Zudem gibt es Web-Anwendungen, welche einen Wechsel der IP-Adresse nach der Anmeldung als *Session Hijacking* auslegen und die Sitzung dann sofort beenden. Im Folgenden stellen wir zwei weitere Maßnahmen vor, welche auf den ersten Blick vielversprechend erscheinen.

Die erste Technik basiert darauf, dass die Nutzer ihre Anfragen auf mehrere nicht-kooperierende Proxy-Server (bzw. DNS-Server) verteilen, so dass jeder Server nur noch einen Bruchteil des Datenverkehrs sieht. Im einfachsten Fall könnten die Anfragen zufällig auf alle verfügbaren Proxy-Server verteilt werden. Abbildung 3a zeigt den Anteil der verkettbaren Sitzungen bei Sitzungen der Länge 24 bzw. drei Stunden in Abhängigkeit von der Anzahl der verwendeten Proxy-Server. Die Abbildung zeigt, dass eine Vielzahl von Proxy-Servern nötig ist, um ein hohes Schutzniveau zu erreichen. Mit ausgefeilteren Verteilungsstrategien lassen sich hier möglicherweise weitere Verbesserungen erzielen. Allerdings bedeutet dieser Ansatz zusätzlichen Aufwand für den Benutzer, da dieser mehrere Proxy-Server finden und abwechselnd verwenden muss.

Die zweite hier betrachtete Gegenmaßnahme basiert auf der Annahme, dass die Verkettbarkeit vor allem auf den Hosts beruht, die besonders spezifisch für einzelne Nutzer sind. Demnach sollte ein Benutzer, der explizit solche Hosts meidet und sich nur noch auf populären Seiten bewegt, nicht mehr so leicht an seinem Verhalten wiederzuerkennen sein.

Diese These haben wir in einer Simulation überprüft. Dazu haben wir die Hosts absteigend nach ihrer Aufrufhäufigkeit sortiert, um die populärsten Hosts zu ermitteln. Aus den Internetsitzungen wurden dann alle Hosts entfernt, die nicht in der Liste der populärsten Hosts enthalten waren. Wie Abbildung 3b zeigt, nehmen die Erkennungsraten jedoch nur geringfügig ab: Selbst wenn dem Klassifizierer nur noch die populärsten 1 % der Hosts zur Verfügung standen – was in unserer Studie 251 Hosts entsprach – konnten noch 65 % der Sitzungen verkettet werden. Selbst die Einschränkung, nur noch auf populären Webseiten zu surfen, bietet also keinen wirksamen Schutz; abgesehen davon erscheint sie wenig praktikabel.

7 Schlussbemerkungen

Die Wirksamkeit des dargestellten Verkettungsangriffs wurde anhand eines möglichst realistischen Angreiferszenarios und in Simulationen evaluiert. Unsere Ergebnisse zeigen, dass die Sitzungen vieler Nutzer bereits anhand der Häufigkeitsverteilung der besuchten Hosts erfolgreich verkettet werden können. Die Nutzer von Proxy- und DNS-Servern müssen dem-

nach davon ausgehen, dass ihre Aktivitäten auch dann über Sitzungsgrenzen hinweg beobachtet werden können, wenn sie über dynamisch zugewiesene IP-Adressen verfügen und einen Proxy oder proxy-basierten Anonymisierer verwenden. Schuld an der Verkettungsmöglichkeit ist das individuelle Surfverhalten der Benutzer, welches mittels statistischer Verfahren wie von uns beschrieben ausgewertet werden kann.

Unsere Ergebnisse sind auch für Nutzer von Anonymisierungsdiensten, die den Datenverkehr über einen HTTP-Proxy-Server weiterleiten, sowie die Nutzer alternativer Anbieter von DNS-Servern von Bedeutung. Üblicherweise wurde bisher meist angenommen, dass solche Dienstanbieter die Anfragen eines Nutzers in der Regel nur verketten können, solange diese von derselben IP-Adresse ausgehen bzw. wenn Tracking-Techniken wie Cookies verwendet werden.

Die hier vorgestellten Gegenmaßnahmen erschweren zwar die Profilbildung, verursachen jedoch erheblichen Mehraufwand bei den Nutzern. Da eine bewusste Änderung des Benutzerverhaltens zur Unterbindung von Verkettungsangriffen von den Nutzern nicht erwartet werden kann, bleibt es weiteren Arbeiten vorbehalten, effizientere Gegenmaßnahmen zu entwickeln.

Literatur

[1] Michael Barbaroand, Tom Zeller: A Face is Exposed for AOL Searcher No. 4417749. The New York Times, August 9, 2006.

[2] T. Narten, R. Draves: Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 3041, 2001.

[3] Justin Brickell, Vitaly Shmatikov: The Cost of Privacy: Destruction of Data-mining Utility in Anonymized Data Publishing. In: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '08), ACM, New York, NY, USA, 2008, 70–78.

[4] Peter Eckersley: How Unique Is Your Web Browser? In: Proceedings of the 10th International Symposium on Privacy Enhancing Technologies (PETS 2010), LNCS 6205, Springer, Heidelberg, Berlin, 2010, 1–18.

[5] Dimitris Koukis, Spyros Antonatos, Kostas G. Anagnostakis: On the Privacy Risks of Publishing Anonymized IP Network Traces. In: Proceedings of the 10th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security, LNCS 4237, Springer, Heidelberg, Berlin, 2006, 22–32.

[6] Bradley Malin, Edoardo Airoldi: The Effects of Location Access Behavior on Re-identification Risk in a Distributed Environment. In: Proceedings of the 6th International Workshop on Privacy Enhancing Technologies, LNCS 4258, Springer, Heidelberg, Berlin, 2006, 413–429.

[7] Arvind Narayanan, Vitaly Shmatikov: Robust De-anonymization of Large Sparse Datasets. In: 2008 IEEE Symposium on Security and Privacy (S&P 2008), IEEE Computer Society, 2008, 111–125.

[8] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel: A Practical Attack to De-anonymize Social Network Users. In: 31st IEEE Symposium on Security and Privacy, S&P 2010, IEEE Computer Society, 2010, 223–238.

[9] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, David Wetherall: 802.11 User Fingerprinting. In: Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking (MobiCom '07), ACM, New York, NY, USA, 2007, 99–110.

[10] Balaji Padmanabhan, Yinghui Yang: Clickprints on the Web: Are there Signatures in Web Browsing Data? Working Paper Series, 2007. Available at http://ssrn.com/abstract=931057

[11] Yinghui Yang, Balaji Padmanabhan: Toward User Patterns for Online Security: Observation Time and Online User Identification. Decision Support Systems 48 (2010) 548–558.

[12] Yinghui Yang: Web User Behavioral Profiling for User Identification. Decision Support Systems 49 (2010) 261–271.

[13] Marek Kumpošt: Data Preparation for User Profiling from Traffic Log. In: Proceedings of the International Conference on Emerging Security Information, Systems, and Technologies (SECUWARE 2007), 2007, 89–94.

[14] Marek Kumpošt: Context Information and User Profiling. PhD thesis, Faculty of Informatics, Masaryk University, Czeck Republic, 2009.

[15] Marek Kumpošt, Vašek Matyáš: User Profiling and Re-identification: Case of University-Wide Network Analysis. In: Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business (TrustBus '09), Springer-Verlag, Berlin, Heidelberg, 2009, 1–10.

[16] Ian H. Witten, Eibe Frank: Data Mining: Practical Machine Learning Tools and Techniques. Elsevier, San Francisco, 2005.

[17] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: Introduction to Information Retrieval. Cambridge University Press, Cambridge, UK, 2008.

[18] George Kingsley Zipf: The Psycho-biology of Language – An Introduction to Dynamic Philology, 2nd edition, M.I.T. Press, Cambridge/Mass., 1968.

[19] Lada Adamic, Bernardo Huberman: Zipf's Law and the Internet. Glottometrics 3/1 (2002) 143–150.

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11/1 (2009) 10–18.