



Fachbereich Informatik und Medien

Masterarbeit

**Nutzeridentifikation im Internet: Evaluation
von Abwehrstrategien**

Vorgelegt von: Christian Labuda
am: 04.08.2011

zum
Erlangen des akademischen Grad

Master of Science

(M.Sc)

Erstbetreuer: Prof. Dr. phil Dipl. Math. Barbara Wiesner
Zweitbetreuer: Dipl.-Inform. Ingo Boersch

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation.....	5
1.2	Ziel dieser Arbeit	7
1.3	Abgrenzung	7
1.4	Aufbau der Arbeit	8
2	Privatsphäre.....	9
2.1	Informationelle Selbstbestimmung	9
2.2	Staatliche Überwachung.....	10
2.3	Überwachung durch Unternehmen.....	11
2.4	Datenverlust und Datenvermeidung	13
3	Nutzeridentifikation.....	14
3.1	Definition	14
3.2	Nutzeridentifikationsprozess	14
3.2.1	HTTP-Requests.....	14
3.2.2	Sessions.....	15
3.2.3	Profile.....	16
3.3	Nutzeridentifikation durch Dritte	19
3.4	Methoden zur eindeutigen Profilzuordnung	20
3.4.1	Passwortbasierte Authentifizierung	20
3.4.2	Cookies.....	20
3.4.3	Hardware ID's	33
3.5	Methoden zur ähnlichkeitsbasierten Profilzuordnung.....	34
3.5.1	Schriftarten	35
3.5.2	UserAgent	35
3.5.3	History.....	36
3.5.4	Autocomplete	38
3.5.5	Ports/Programme	39
3.5.6	IP-Adresse	39
3.5.7	Standortdaten.....	40
3.5.8	Implizites Wissen	41
3.6	Tracking.....	42
3.6.1	Allgemein	42
3.6.2	Informationsleck durch Iframes und Webbugs	43
3.6.3	Cookie-Bibliotheken	44
3.6.4	Tools zum „Tracken“ von Webseitenbesuchern	45
4	Abwehrmaßnahmen.....	46
4.1	Schwachstellen im Identifikationsverfahren	46
4.2	Cookieabwehr	48
4.2.1	Deaktivieren des DOM-Storage	49
4.2.2	Abwehr von Flash-Cookies und Java-Cache-Cookies	49
4.2.3	Cache	51
4.2.4	Verlauf	52
4.2.5	Drittanbietercookies verweigern.....	52
4.2.6	Anti-Cookie Plugins.....	52
4.3	Anonym Surfen	53

4.3.1	Modifizierung von Merkmalen zur Anonymisierung	53
4.3.2	Anonymisierungsdienste	55
4.4	Filterprogramme	56
4.4.1	Werbeblocker	56
4.4.2	Filterproxys	56
4.4.3	Werbefinanziertes Internet & Filterprogramme	56
4.5	Identitätsmanagement	57
5	Entwicklung der Testseite	58
5.1	Anforderungen	58
5.1.1	Client	58
5.1.2	Cookies	59
5.2	Sprachen	59
5.3	Entwicklungsumgebungen	60
5.4	Architektur	60
5.4.1	Server	60
5.4.2	Client	61
5.5	Funktionsablauf	62
5.5.1	Cookies speichern	62
5.5.2	Cookies laden	63
5.6	Systemvoraussetzungen	64
5.7	Installation	64
6	Evaluation von Abwehrstrategien	65
6.1	Vorgehensweise	65
6.2	Cookieabwehr	65
6.2.1	Ziel	65
6.2.2	Testumgebung	66
6.2.3	Durchführung der Tests	67
6.2.4	Auswertung der Ergebnisse	73
6.3	Anonymisierungsdienste	74
6.3.1	Vidalia Bundle	74
6.3.2	Jondonym	75
6.3.3	Vor- und Nachteile beider Dienste	75
6.4	Filterprogramme	76
6.4.1	Werbeblocker	76
6.4.2	Filterproxies Polipo & Squid	78
6.4.3	Filterprogramme als Trackingschutz	78
7	Ausblick	80
7.1	IPv6 ohne dynamische IP-Adressen	80
7.2	WebGL	80
7.3	Biometrie	81
7.3.1	Gesichtserkennung	81
7.3.2	Mausbewegungen	82
8	Fazit	83
	Literaturverzeichnis	86
A	Anhang	89
	Abschließende Erklärung	91

1 Einleitung

1.1 Motivation

Die vielen innovativen Anwendungen des letzten Jahrzehnts haben dazu geführt, dass das Internet immer schneller wächst. Im Vordergrund dieser Entwicklung steht der so genannte „user generated content“. [Dwo10]

Inhalte im Internet werden heutzutage nicht mehr hauptsächlich von Mediengiganten produziert, sondern von den Nutzern des Internets selbst. Die immer stärker werdende Vernetzung und die großen Bandbreiten, sowie neue Anwendungen, die es dem Nutzer erleichtern seine Ideen und Gedanken zu veröffentlichen, haben zu einer Unmenge an personenbezogenen Daten geführt. Blogs und Foren waren bei dieser Entwicklung nur der Anfang.

Die Daten einer einzelnen Webapplikation können ein sehr umfassendes Bild des Nutzers geben. Dies gilt nicht nur für soziale Netzwerke, sondern z.B. auch für einen Onlinekalender bei dem man Termine speichern kann. Diese Termine können Aufschluss über das Alter, den Gesundheitszustand geben, darüber wo sich der Nutzer sich befindet und zu welchen Personen er Kontakt hat. Der Nutzer eines solchen Angebotes ist sich oft nicht bewusst, dass diese Daten durch die Nutzung preisgegeben werden und dass er dadurch identifiziert werden kann. [ScE10]

Der sorglose Umgang vieler Nutzer mit ihren eigenen Daten führt zu weiteren Problemen. Wichtige Passwörter und andere sehr persönliche Informationen werden beispielsweise über Instant-Messenger versendet, da diese Art der Datenübertragung komfortabel und unkompliziert ist. Bei sozialen Netzwerken zeigt sich besonders, dass Nutzer eine Vielzahl von Informationen über sich preisgeben, ohne sich der Risiken bewusst zu sein. Der Schutz der Daten ist bei den meisten dieser Netzwerke jedoch ungenügend. Die Standardeinstellungen von sozialen Netzwerken veröffentlichen mehr als notwendig wäre, um gesuchte Personen (Freunde, Bekannte) zu finden. Außerdem lassen sich Netzwerke dieser Art, nach aktuellem Stand der Technik, nur schlecht oder überhaupt nicht gegen Crawler (Tools zum analysieren und indexieren von Webseiten) schützen.

Durch die wachsende Anzahl von Diensten im Bereich Internet entstehen riesige Informationsmengen, die gezielt einzelnen Personen zugeordnet werden können. Neue technische Möglichkeiten erhöhen ständig den Wert dieser Daten. Informationen aus sozialen Netzwerken und neue Möglichkeiten im Bereich der Biometrie werden zukünftig

nicht nur für Geheimdienste interessant sein. Die Fortschritte im Bereich der Biometrie sind genauso vielfältig, wie ihre möglichen Einsatzgebiete. Es ist sehr wahrscheinlich, dass auch diese Entwicklungen dazu beitragen werden den Nutzer auszuspionieren und seine Aktivitäten zu überwachen.

Die Weiterentwicklung von Handys zu Endgeräten für das Internet hat dazu geführt, dass der Standort des Nutzers dank GPS zu jeder Zeit sehr genau bekannt ist. Es entstanden neue Anwendungen, die genau diese Daten nutzen, um regionsabhängige Dienste anzubieten. Zu diesen zählen z.B. Karten- und Navigationsdienste sowie standortbezogene Werbung. Soziale Netzwerke nutzen die Geokoordinaten der Handys unter anderem, um die Aufenthaltsorte der Freunde anzeigen zu können. Es existieren außerdem Studien [Smi10][WSJ11] die belegen, dass das Verfolgen und Wiedererkennen von mobilen Endanwendern um ein Vielfaches einfacher ist, als dies bei normalen Internetnutzern der Fall ist. Der Grund dafür ist, dass der Zugang zu verschiedenen Webdiensten nicht unbedingt über den Browser sondern über so genannte Apps realisiert wird.

Die Preisgabe großer Mengen personenbezogener Daten ist bereits bekannt und wird kontrovers diskutiert. Da bei der Nutzung eines Webservices jedoch auch noch weitere Daten anfallen, die der Nutzer nicht bewusst und freiwillig angegeben hat, entsteht eine Art Kontrollverlust über die eigenen Daten. Es existieren technische Möglichkeiten Nutzer anhand von Cookies wiederzuerkennen und ihre individuellen Vorlieben durch die Verwendung von Trackern auszuspähen. Hinzu kommt, dass die Möglichkeit, Nutzer über die eigene Seite hinaus zu verfolgen, nicht nur von Werbeagenturen, sondern auch immer stärker von sozialen Netzwerken genutzt wird. Der Umfang eines Profils aus einem sozialen Netzwerk enthält somit neben den sichtbaren Profildaten weitere Daten, die sich durch die Nutzung verschiedenster Webdienste ergeben. Je nach Anzahl der Webseiten, die mit dem jeweiligen sozialen Netzwerk kooperieren, entsteht so ein sehr umfassendes Profil über die Internetnutzung einer Person, ohne dass die jeweilige Person diese Daten kontrollieren kann.

Das Recht auf informationelle Selbstbestimmung [Bun83] wird durch moderne Dienste oft ohne das Wissen der Nutzer in besonderem Maße eingeschränkt. Die einzige Möglichkeit einen Rest an Kontrolle über die eigenen Daten zu behalten, ist die Ergreifung geeigneter Abwehrmaßnahmen. Das Hauptmotiv für die Entstehung dieser Arbeit ist, es jedem Internetnutzer zu ermöglichen, sein Recht auf informationelle Selbstbestimmung wahrzunehmen indem sowohl die Möglichkeiten der Nutzeridentifikation als auch entsprechende Abwehrmaßnahmen untersucht werden.

1.2 Ziel dieser Arbeit

Ziel dieser Arbeit ist es zunächst den aktuellen Stand der Technik im Bereich der Nutzeridentifikation zu untersuchen, um im Folgenden Abwehrmaßnahmen gegen die verschiedenen Methoden der Nutzeridentifikation im Internet bewerten zu können.

Nach aktuellem Stand der Technik verfügen moderne Browser bereits über Möglichkeiten verschiedene Identifikationsmöglichkeiten abzuwehren. Es ist jedoch eine Tatsache, dass eine einzige Lücke in dieser Abwehr es ermöglichen kann den Nutzer wiederzuerkennen, wodurch die anderen Abwehrmaßnahmen ihren Sinn verlieren.

Die Abwehrtechnologien der am stärksten verbreiteten Browser sollen deshalb überprüft werden. Es wird ebenfalls geprüft, ob sich die Abwehrfunktionen der Browser durch eine sinnvolle Auswahl von Plugins und Zusatzprogrammen verbessern lassen. Zur Durchführung der Tests ist eine eigene Testseite zu implementieren, die bekannte eindeutige Identifikationsmechanismen unterstützt. Die Testseite soll außerdem eine Selbstkontrolle ermöglichen, sodass Internetnutzer ihre eigene Abwehr kontrollieren können.

Ein ergänzender Test der 50 am stärksten frequentierten Webseiten soll zeigen, ob die Webseiten trotz Abwehrmaßnahmen noch funktionieren. Dieser Test dient vor allem dazu die Einschränkungen im Komfort, die ein Nutzer durch die Nutzung von Abwehrmaßnahmen hinnehmen muss, einschätzen zu können.

1.3 Abgrenzung

Diese Arbeit wird nicht darauf eingehen, warum Nutzer freiwillig personenbezogene Daten im Internet über sich preisgeben. Diese Arbeit beschränkt sich auf Daten, die der Nutzer allein durch die Nutzung von Webseiten und oft ohne sein Wissen im Internet preisgibt.

Die Bezeichnung mobile Endgeräte bezieht sich im Rahmen dieser Arbeit vor allem auf Smartphones, Handys und Tablets. Diese werden nur am Rand betrachtet, da sich die Problematik der Nutzeridentifikation und auch der Abwehrmechanismen auf diese übertragen lässt. Eine Evaluation der Abwehrmechanismen auf mobilen Endgeräten wird jedoch aufgrund der Vielzahl von Plattformen und Betriebssystemen im mobilen Bereich nicht durchgeführt.

Die Auswertung und Weiterverarbeitung der Nutzerprofile sowie deren kommerzielle Verwertungsmöglichkeiten sind ebenfalls nicht Gegenstand dieser Arbeit.

Es existieren Programme, die verschiedene Onlineidentitäten verwalten, um den Komfort für den Nutzer zu erhöhen. Diese dienen vor allem zur Verwaltung der Daten, die der Nutzer auf einer Webseite eingegeben hat. Sie werden im Rahmen dieser Arbeit nicht evaluiert.

1.4 Aufbau der Arbeit

Diese Arbeit geht zunächst auf das Thema Privatsphäre ein, um ein Bewusstsein für die folgende Problematik zu vermitteln. Darauf folgt ein Kapitel über die Identifikation von Nutzern im Internet. In diesem wird ausführlich der aktuelle Stand der Technik im Bereich der Nutzeridentifikation im Internet geschildert und untersucht, wobei besonders auf die Methoden zum Setzen von Cookies von eingegangen wird.

Um zu demonstrieren, dass neben den vorhandenen Cookiearten in Zukunft auch neue entstehen können, wird außerdem ein Konzept für eine neue Cookieart erstellt und implementiert. Bei dieser neuen Cookieart wird es sich um einen auf JavaApplets basierenden Cookie handeln (siehe 3.4.2.11).

Das vierte Kapitel geht auf aktuelle Strategien ein, die eine Identifikation des Nutzers verhindern sollen. Um diese Strategien und vor allem die Abwehr von Cookies zu testen, wird eine Testseite zur Evaluation erstellt. Im darauf folgenden Kapitel werden diese Strategien untersucht und ausgewertet.

Das Ende der Arbeit gibt einen Ausblick auf Techniken, die in naher Zukunft zur Identifikation von Nutzern eingesetzt werden könnten. Die Arbeit endet mit einem abschließenden Fazit.

2 Privatsphäre

Privatsphäre ist in erster Linie eine Form von Autonomie. Das bedeutet, dass ein Individuum frei und unabhängig entscheiden und sich somit selbst verwirklichen kann. Dem gegenüber steht ein Sicherheitsbedürfnis, welches in unserer Gesellschaft durch Kontrolle und Überwachung befriedigt wird, um jedem Individuum ein Gefühl von Sicherheit und Schutz zu geben.

2.1 Informationelle Selbstbestimmung

Eine Möglichkeit den Begriff Privatsphäre (engl. Privacy) zu umschreiben fanden 1890 Samuel D. Warren und Louis D. Brandeis. Sie definierten Privatsphäre als „the right to be let alone“, beziehungsweise als das Recht allein und unbeobachtet zu sein. [WaB90]

Etwas umfassender wird dieses Recht 1948 von der UNO [UNO48] beschrieben: „Niemand darf willkürlichen Eingriffen in sein Privatleben, seine Familie, seine Wohnung und seinen Schriftverkehr oder Beeinträchtigungen seiner Ehre und seines Rufes ausgesetzt werden. Jede Person hat Anspruch auf rechtlichen Schutz gegen solche Eingriffe oder Beeinträchtigungen.“

Der Begriff Privatsphäre hat also etwas mit der freien Entfaltung der Persönlichkeit in einem nicht öffentlichen Bereich des Lebens zu tun. Die freie Entfaltung der Persönlichkeit, ist durch Art. 2 GG gesetzlich verankert. Der Schutz des nicht öffentlichen Bereich des Lebens teilt sich in verschiedene Grundgedanken.

Einer dieser Grundgedanken ist ein physischer Schutz der Privatsphäre. Der Art. 13 GG stellt die Unverletzlichkeit der Wohnung sicher und schafft somit einen Raum für diesen Schutz. Die Wohnung stellt einen Ort dar, in dem sich ein Mensch als Individuum unbeobachtet von Dritten entfalten darf. Da dies insbesondere für die politische Meinungsfindung von ausschlaggebender Bedeutung ist, ist dieser besondere Schutz der Privatsphäre in allen modernen Demokratien gesetzlich verankert. Ein Eingriff in diesen besonders geschützten Bereich ist deshalb nur im Zuge einer Strafverfolgung möglich.

Jede Art von Kommunikation stellt eine Form des gedanklichen Austausches dar. Die Überwachung einer Kommunikation ist deshalb als ein Eingriff in die persönliche Entfaltung zu sehen, da sich die Personen nicht mehr so ungezwungen verhalten können, wie es ohne Überwachung der Fall wäre. Die Kommunikation zwischen Personen ist durch das Postgeheimnis und das Fernmeldegeheimnis Art. 10 GG geschützt.

Ein weiterer Grundgedanke ist, dass der Schutz der Privatsphäre über die Wohnung hinaus gehen muss, da jede Information über eine Person Rückschlüsse auf diesen geschützten Bereich zulässt. Eine Person sollte deshalb selbst bestimmen können, wer

welche Informationen über diese Person selbst besitzen darf. Informationen dieser Art sind besonders schützenswert und werden auch als personenbezogene Daten bezeichnet. [Bun83]

Mit dem Hessischen Datenschutzgesetz entstand 1970 das erste Gesetz zum Schutz personenbezogener Daten. Dieses Gesetz umfasste dabei nicht nur den Schutz der Daten selbst, sondern vor allem anderen den Schutz der Persönlichkeitsrechte jedes einzelnen. Das Hessische Datenschutzgesetz ist somit das erste Gesetz seiner Art weltweit. Es diente außerdem als Vorlage für das später eingeführte Bundesdatenschutzgesetz.

Erst mit zunehmender elektronischer Datenverarbeitung, wurde Datenschutz zu einem immer wichtigeren Thema. Die für das Jahr 1983 geplante Volkszählung, führte zu Massendemonstrationen, zur Verfassungsbeschwerde und schließlich zum Volkszählungsurteil von 1983 [Bun83] des Bundesverfassungsgerichtes. Das Bundesverfassungsgericht leitete aus dem Art 2 des Grundgesetz ein Recht auf informationelle Selbstbestimmung ab. Dieses Urteil ist von wegweisender Bedeutung für alle späteren Datenschutzgesetze in Deutschland.

Das Recht auf informationelle Selbstbestimmung wurde geschaffen, um einer einzelnen Person mehr Kontrolle darüber zu geben, wer welche Informationen (die Person selbst betreffend) besitzen darf.

2.2 Staatliche Überwachung

Staatliche Überwachung hat zwei grundlegende Ziele. Das erste dieser beiden Ziele ist es möglichst viel zu automatisieren, damit der Staat schnell und wirtschaftlich arbeiten kann. Um dies zu bewerkstelligen, ist es nötig die privaten Informationen der Bürger zusammenzutragen, damit beispielsweise das Finanzamt überhaupt eine Grundlage zum Verrichten der Arbeit hat. Die Daten sollen sicherstellen, dass die Vorgänge schnell und effizient erledigt werden können. Der Vorteil für den Bürger liegt darin, dass die Automatisierung vor allem schnell und kostengünstig ist. Letzteres führt direkt zu einer geringeren steuerlichen Belastung. [Str05]

Das zweite große Ziel staatlicher Überwachung ist es die Sicherheit der Bürger zu garantieren. Ein wichtiger Punkt ist die Aufklärung von Verbrechen, die es nötig macht der Polizei Einsicht in den privaten Bereich verdächtiger Personen zu gestatten. Ohne diese Möglichkeit würde ein rechtsfreier Raum entstehen, der sich negativ auf die Gesellschaft auswirken kann. Auch die Prävention von Verbrechen, wozu unter anderem auch die Terrorabwehr gehört, ist ein Ziel staatlicher Überwachung. Ein Beispiel für einen präventiven Eingriff in die Privatsphäre sind zum Beispiel die Zugangskontrollen an Flughäfen. [Str05]

Dem entgegen steht der Schutz der Privatsphäre jedes einzelnen Individuums. Die Privatsphäre des Einzelnen darf nur dann verletzt werden, wenn der Nutzen für die Allgemeinheit überwiegt. [Bun83] Ob in einem konkreten Fall nun der Schutz der Privatsphäre oder der Vorteil für die Allgemeinheit überwiegt, ist durch die Datenschutzgesetze geregelt. Es kommt jedoch immer auf den einzelnen Fall an, ob Überwachung zulässig ist oder nicht, da gilt es den Grundsatz der Verhältnismäßigkeit zu wahren. Dabei ist außerdem zu beachten, dass immer nur so viele Daten erhoben werden dürfen wie notwendig sind (§3 a BDSG).

Eine weitere Gefahr automatisierter staatlicher Überwachung ist die Einteilung von Personen in Kategorien durch eine automatische Profilerstellung. Es ist dadurch möglich, dass eine Person, die weder definitiv schuldig noch nachweislich unschuldig ist, etwas wie eine „Gefahrenstufe gelb“ [Str05, S. 266] bekommt. Solche Personen werden nun entgegen jeder Rechtsstaatlichkeit mit Misstrauen behandelt und haben mit Benachteiligungen zu rechnen. Die generelle Unschuldsvermutung wird so völlig ausgehebelt.

Die Umkehrung, also zu viel Privatsphäre, führt jedoch dazu, dass die Aufklärung von Verbrechen in bestimmten Bereichen effektiv verhindert wird. Ein Anstieg der Kriminalität ist eine direkte Folge.

Privatsphäre und Überwachung stehen somit in einem ständigen Kampf gegeneinander wobei nur ein ausgewogenes Verhältnis, das der momentanen Situation angepasst ist, den Bedürfnissen der Bürger entspricht. Das Verhältnis kann sich durch hohe Kriminalität, Terroranschläge oder Krieg, also Dinge, die die Sicherheit bedrohen, in Richtung Überwachung verschieben. Frieden und niedrige Kriminalität erhöhen im Gegensatz dazu das Bedürfnis nach mehr Privatsphäre. Die einzelnen Ziele gehen dabei in unterschiedliche Richtungen, schließen sich aber nicht gegenseitig aus. Es ist ein Prozess, der eine ständige Anpassung an die momentane Situation erfordert.

2.3 Überwachung durch Unternehmen

Das Hauptinteresse eines Unternehmens liegt natürlich nicht in der Überwachung seiner Kunden sondern im Erzielen von Profit. In fast jeder Branche ist es jedoch vorteilhaft zu wissen, was ein potentieller Kunde denn überhaupt für Interessen hat und was er sich wünscht. Anhand dieser Informationen kann ein Unternehmen dann gezielt neue Produkte auf den Markt bringen oder interessierte Kunden zielgerichtet mit Informationen über die eigenen Produkte versorgen.

Zwangsläufig sind Unternehmen daran interessiert, ein möglichst genaues Bild ihrer Kunden zu bekommen. Die einfachste Möglichkeit an Daten über potentielle Kunden zu kommen ist der Kauf dieser Daten von einem Unternehmen, das durch sein Geschäftsmodell im Besitz solcher Daten ist. Dies sind im Internet vor allem soziale

Netzwerke und Suchmaschinenanbieter. Diese sind jedoch selbst an Profit interessiert und bieten deshalb an, ihre Nutzer zielgerichtet mit Werbung der jeweiligen Unternehmen zu versorgen. Auf diese Weise verlassen die wertvollen Nutzerdaten das eigene Unternehmen nicht. Ein weiterer Vorteil ist, dass die Datenschutzgesetze vieler Länder durch diese Maßnahme eingehalten werden kann, da keine Weitergabe von Daten an Dritte erfolgt. Dennoch wird eine immer weitergehende Überwachung des Einzelnen durch dieses Modell gefördert, da die Anbieter die Daten erheben auswerten. Im Fall von sozialen Netzwerken ist es sogar möglich die Daten gezielt realen Personen zuzuordnen. Dies war bei den ursprünglichen Werbenetzwerken nur bedingt möglich, da sich die Nutzer dort nicht mit ihrem realen Namen anmeldeten.

Neben diesen Quellen gibt es einen Handel mit Kundendaten. Diese können beispielsweise von insolventen Unternehmen, aus sozialen Netzwerken, Blogs oder aus anderen Quellen stammen. Firmen wie RapLeaf [Rap11] bereiten diese Daten auf, und handeln mit den Informationen.

Die Unternehmen selbst verwalten außerdem Informationen über ihre eigenen Kunden. Heutzutage kommen dabei Customer-Relationship-Management-Systeme (CRM) zum Einsatz. Diese erleichtern es, Kunden mit ihren Interessen und persönlichen Vorlieben, aber auch vorrangegangene Gespräche, zu verwalten. Es vereinfacht das Senden von individualisierten Rundmails und andere Formen der Werbung. Es ist somit einfacher Kunden an das Unternehmen zu binden.

Ein großes wirtschaftliches Interesse an personenbezogenen Daten haben auch Versicherungen. Es ist zum Beispiel unproblematisch und in den meisten Fällen sogar notwendig, dass eine Arztpraxis Informationen über seine Patienten besitzt. Die gleichen Daten können jedoch zu enormen Nachteilen für eine Person führen, falls diese in den Besitz einer Versicherungsgesellschaft gelangen. Für das Funktionieren einer Versicherung ist die Solidarität von großer Bedeutung. Eine Benachteiligung einzelner Personen würde dem Gedanken der Solidarität und somit dem Sinn einer Versicherung widersprechen. Im Fall einer Versicherung stehen sich wirtschaftliche Interessen und die einer solidarischen Gemeinschaft gegenüber.

Ein anderes Beispiel sind die GPS-Informationen eines Autos, oder auch Informationen aus einem Mautsystem [Str05, S. 294]. Da diese Daten Rückschlüsse auf den Fahrstil zulassen, könnten Autofahrer mit aggressivem Fahrstil von den Versicherungen benachteiligt werden, falls diese an die dazu nötigen Daten gelangen. In diesem Fall könnten fehlerhafte Daten sogar dazu führen, dass die Versicherungen sich weigern den Fahrer zu versichern.

Eine weiteres Beispiel für einen Interessenkonflikt im Bereich personenbezogener Daten ist, die Möglichkeit des Auslesens personenbezogener Daten aus einem sozialen Netzwerken durch den Einsatz eines speziellen Crawlers. Diese Daten können dann dann

gewinnbringend in eigenem Interesse genutzt werden. Sie können Informationen darüber enthalten, ob jemand zu übermäßigem Alkoholkonsum tendiert oder Raucher ist und viele weitere Details, die sich negativ auf die Versicherungskosten auswirken können. Dies wäre eine Benachteiligung der Betroffenen zugunsten der Wirtschaftlichkeit der Versicherung. Umgekehrt wäre es natürlich auch eine Bevorzugung gesund lebender Menschen. Die eigentliche Frage ist, an welchem Punkt die Versicherung moralische und gesetzliche Grenzen überschreitet und zum Beispiel teure Patienten mit besonderen Krankheiten ausschließt, um Kosten zu sparen.

2.4 Datenverlust und Datenvermeidung

Das große Problem digitaler Datenansammlungen ist, dass Daten leicht zu kopieren und zu entwenden sind. Dies kann auf sehr unterschiedliche Art und Weise geschehen. Die folgende Tabelle 1 zeigt die 10 größten Verluste personenbezogener Daten.

Tabelle 1 - Datenverlust [OSF11]

Datensätze	Datum	Organisation
130.000.000	20.01.2009	Heartland Payment Systems, Tower Federal Credit Union, Beverly National Bank
94.000.000	17.01.2007	TJX Companies Inc.
90.000.000	01.06.1984	TRW, Sears Roebuck
77.000.000	26.04.2011	Sony Corporation
76.000.000	05.10.2009	National Archives and Records Administration
40.000.000	19.06.2005	CardSystems, Visa, MasterCard, American Express
32.000.000	14.12.2009	Rockyou Inc
26.500.000	22.05.2006	U.S. Department of Veterans Affairs
25.000.000	20.11.2007	HM Revenue and Customs, TNT
24.600.000	02.05.2011	Sony Online Entertainment, Sony Corporation

Diese Tabelle beweist, dass einmal erhobene Daten nicht unbedingt bei dem Unternehmen bleiben müssen, durch das sie erhoben wurden. Es existiert eine Vielzahl von Möglichkeiten durch die Daten in „falsche Hände“ kommen können. Selbst Kreditinstitute sind nicht immer in der Lage, ihre wichtigen Daten zu schützen, wie die Tabelle deutlich zeigt. Es handelt sich bei diesen Vorfällen jedoch keinesfalls um Einzelfälle, sondern nur um die Größten der bekanntgewordenen Vorfälle.

Aus genau diesem Grund ist neben dem Schutz der Daten durch die Unternehmen, die Datenvermeidung eines der wichtigsten Anliegen der Datenschützer. Je weniger Daten erhoben werden, desto geringer ist deren Missbrauchspotential. Es ist jedoch nicht möglich sich allein auf Gesetze und den guten Willen der Unternehmen zu verlassen, sondern es ist nötig, dass auch jeder Einzelne das Entstehen von Daten vermeidet beziehungsweise die Möglichkeit dazu erhält.

3 Nutzeridentifikation

3.1 Definition

Definition: Im Rahmen dieser Arbeit ist Nutzeridentifikation ein Prozess, der das Ziel hat, die bei der Nutzung einer Webseite anfallenden Daten eines einzelnen Nutzers, genau diesem Nutzer zuzuordnen. Der Nutzer ist dabei die Person, die die Webseite im Browser aufruft und die Eingaben macht.

Der Prozess der Nutzeridentifikation besteht aus folgenden Schritten:

1. dem Auslesen der Daten,
2. der Zuordnung der Daten zu einer Session,
3. der Zuordnung der Sessiondaten zu einem abstrakten Profil des Nutzers und
4. der Zuordnung eines Profils zur realen Person.

Der 4. Schritt ist jedoch optional, da es nach der Definition nicht notwendig ist, den realen Nutzer festzustellen. Für eine Nutzeridentifikation genügt es, die anfallenden Daten dem jeweiligen Nutzer zuzuordnen, ohne dessen exakte Identität zu kennen.

Falls es doch gelingt den Nutzer mit einer realen Person zu verknüpfen, so wird im Folgenden von einer „vollständigen Nutzeridentifikation“ gesprochen. Eine vollständige Nutzeridentifikation ist nur dann möglich, wenn das gespeicherte Profil mit weiteren Daten in Verbindung gebracht werden kann, die eine Zuordnung erlauben oder der Nutzer sich durch Angabe seiner realen Daten gegenüber dem System authentifiziert. Bei diesem Prozess kann es vorkommen, dass ein Profil auf mehrere Personen passt, oder einer Person mehrere Profile zugeordnet werden können.

3.2 Nutzeridentifikationsprozess

3.2.1 HTTP-Requests

HTTP-Requests sind Anfragen an einen Webserver über das HTTP-Protokoll. Das Auslesen der Daten aus einem Request geschieht auf der untersten Ebene des Nutzeridentifikationsprozesses (Abbildung 1). Bei jedem HTTP-Request, werden Daten an den Server übermittelt, die ausgewertet werden können. Requests auf bestimmte Webseiten können Rückschlüsse auf die Interessen des Nutzers geben. Dies ist möglich, da bei jedem angeklickten Link und bei jedem bestätigtem Formular, Daten an den Server gesendet werden. Die Formulardaten können Aufschluss über die Identität, die Interessen, oder auch die Suchanfragen des Nutzers geben.

Neben diesen aktiv durch den Nutzer ausgelösten Requests, können auch in die Webseite eingebettete Elemente wie JavaScript, Flash und Java-Applets Requests an den Server senden, um zum Beispiel auf dem Client gesammelte Informationen an den Server zu übermitteln. Alle diese Daten sind jedoch unabhängig und müssen zunächst in einer Session zusammengefasst, beziehungsweise einer bestehenden Session zugeordnet werden.

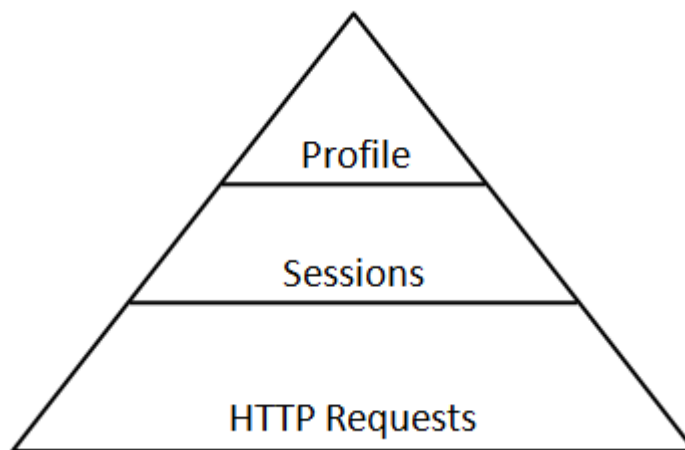


Abbildung 1 - Hierarchie der Zuordnungsebenen

3.2.2 Sessions

Die Zuordnung der Daten eines HTTP-Requests zu einer Session, erfolgt anhand von Cookies, oder anhand einer Sessionvariablen. Für eine sichere Zuordnung zu einer Session wird das Übertragungsprotokoll HTTPS benötigt, da sich sowohl Session-Cookies (siehe 3.4.2) als auch Sessionvariable mitschneiden(engl. sniffen) und fälschen lassen. Bei der Verwendung von HTTPS ist der Einsatz von Session-Cookies oder Sessionvariablen theoretisch nicht mehr nötig, da schon für den kryptographischen Teil des Protokolls eine Zuordnung durchgeführt wird (symmetrischer Session-Schlüssel). Dennoch setzen viele Webservices zusätzlich auf Session-Cookies oder Sessionvariablen.

Eine Session beginnt mit dem ersten Aufruf der Webseite und endet üblicherweise erst, wenn der Browser geschlossen wird oder der Webserver die Session beendet. Beim Speichern der Daten einer Session kann jedoch nicht ohne Weiteres festgestellt werden, ob inzwischen der Nutzer des Clients gewechselt hat. Ein Ausnahme bilden Webseiten die clientseitig biometrische Verfahren einsetzen, um den Nutzer anhand von Mausbewegungen, dem Gesicht oder der Stimme zu erkennen. Im Rahmen dieser Arbeit wurde jedoch keine Webseite gefunden, die eine der Möglichkeiten nachweisbar nutzt.

3.2.3 Profile

3.2.3.1 Definition

Definition: In dieser Arbeit ist ein Profil ein abstraktes Modell, das Informationen über die Interessen, die Verhaltensweisen und typischen Merkmale des Nutzers sowie des benutzten Endgerätes enthalten kann. Die Informationen eines Profils gehören mit hoher Wahrscheinlichkeit zu einem einzigen Nutzer. Ein Profil erleichtert das Zusammenfassen von Sessions des gleichen Nutzers, da es die extrahierbaren Merkmale vorangegangener Sessions beschreibt und so eine Zuordnung ermöglicht.

3.2.3.2 Merkmale

Ein Beispiel für ein extrahierbares Merkmal ist das Betriebssystem des Endegerätes. Hinweise darauf bieten sowohl der UserAgent als auch die installierten Schriftarten. Beide Datensätze können jedoch unterschiedliche Angaben über das Betriebssystem enthalten. Die Information, dass die Angaben unterschiedlich sind, stellt ein besonderes Merkmal dar, dass ebenfalls in das Profil übernommen werden sollte. Die Art und die Menge der Daten in einem neuen Profil kann in Abhängigkeit von den verwendeten Methoden variieren.

Neben dem Betriebssystem sind auch weitere Merkmale (siehe Abbildung 2), wie die Zeiten, in denen sich ein Nutzer oft auf der Webseite aufhält, ein Merkmal, das anhand der Sessiondaten gewonnen werden kann.

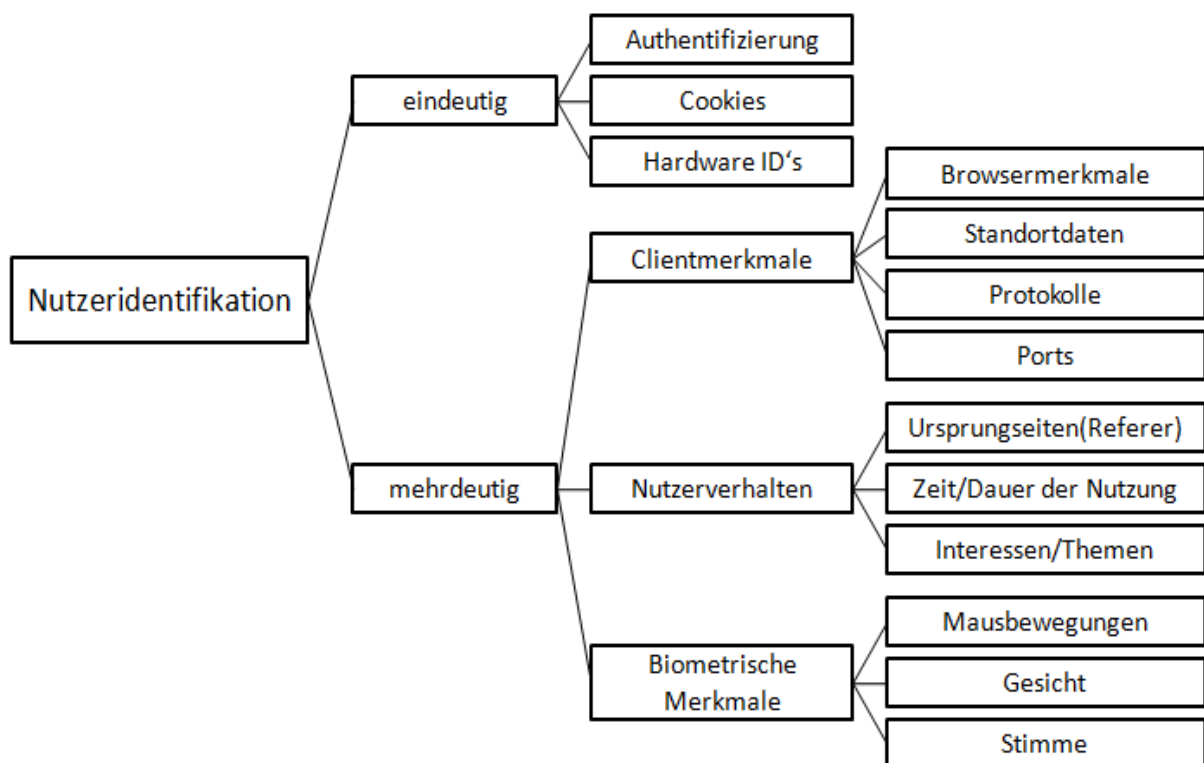


Abbildung 2 - Merkmale

3.2.3.3 Eindeutige Profilzuordnung

Für die Zuordnung von Sessiondaten zu einem Profil ist es möglich, eindeutige oder auf Ähnlichkeit basierende Verfahren einzusetzen. Bei eindeutigen Verfahren können neue Daten sehr leicht mit einem bereits gespeichertem Profil verknüpft werden und dieses dadurch erweitern. Eindeutige Verfahren basieren auf eindeutigen Merkmalen, die bei jedem Besuch ausgelesen werden können. Dafür kann eine Authentifikation des Nutzers nötig sein, aber auch Cookies und das Auslesen von Hardware ID's zählen zu diesen Verfahren (siehe Abbildung 2).

Eine eindeutige Zuordnung einer Session zu einem Profil ist immer dann möglich, wenn das jeweilige eindeutige Merkmal mit dem im Profil übereinstimmt. Im Fall von Cookies wäre das eindeutige Merkmal eine im Cookie gespeicherte Nutzer-ID oder im Fall eines Logins der Nutzername und das Passwort.

Der Einsatz von eindeutigen Verfahren ist jedoch nicht immer möglich. Cookies können nicht immer gesetzt werden, da einige Nutzer den Zugriff sperren. Logins können ebenfalls nicht immer eingesetzt werden, da der Komfort für den Nutzer an einigen Stellen wichtiger ist und auch auslesbare Hardware ID's existieren nicht auf jedem Endgerät.

3.2.3.4 Ähnlichkeitsbasierte Profilzuordnung

Bei mehrdeutigen Merkmalen, ist es ungleich schwieriger die Sessiondaten mit einem Profil zusammenzuführen. Ein mehrdeutiges Merkmal kann bei vielen verschiedenen Nutzern gefunden werden. Die Kombination verschiedener mehrdeutiger Merkmale kann jedoch eine eindeutige Identifizierung ermöglichen. Da sich mehrdeutige Merkmale außerdem verändern können, ist der Grad der Ähnlichkeit für eine Zuordnung entscheidend.

Die Sessiondaten des Webseitenbesuchers können sich von Webseitenbesuch zu Webseitenbesuch unterscheiden. Dies ist zum Beispiel dann der Fall, wenn der Browser ein Update hat, neue Schriftarten installiert wurden, oder der Nutzer von seinen normalen Gewohnheiten abweicht. Die aus den Sessiondaten extrahierten Merkmale sind dann nur noch anhand ihrer Ähnlichkeit mit einem Profil zusammenführbar.[Eck11]

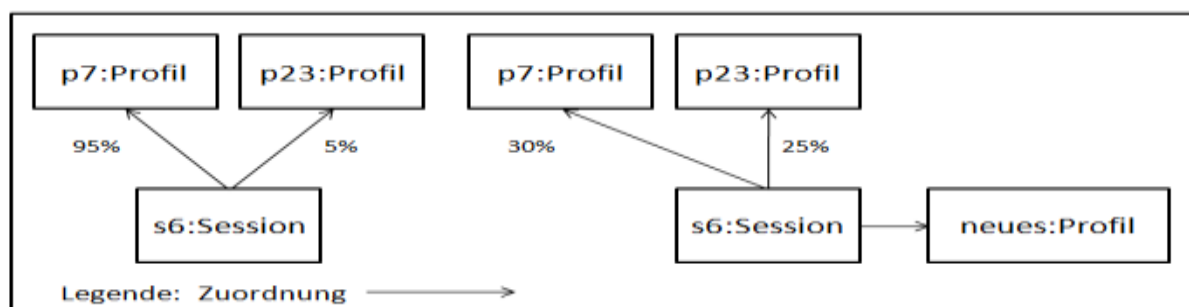


Abbildung 3 – Profilzuordnung (ähnlichkeitsbasierend)

Das Profil, das mit der höchsten Wahrscheinlichkeit zu den Sessiondaten passt, wird um Informationen aus der Session erweitert (Abbildung 3 links). Unterhalb eines bestimmten Schwellwertes kann die Session jedoch auch zur der Erstellung eines neuen Profils führen (Abbildung 3 rechts). Die Gefahr besteht darin, dass die Daten mit einem falschen Profil verknüpft werden (false Positiv) oder dass Sessiondaten, die zu einer bereits erfassten Person gehören, unabhängig bestehen bleiben (false Negativ), beziehungsweise zur Erstellung eines neuen Profils führen. Es ist deshalb auch möglich, dass Profile, die vorher verschiedenen Nutzern zugeordnet wurden, aufgrund neuer Daten zu einem Profil zusammengeführt werden können.

3.2.3.5 Kombination der Verfahren

Der in Abbildung 4 dargestellte Vorgang zeigt die Schritte bis zur Profilzuordnung/Profilerstellung. In diesem Beispiel werden die Möglichkeiten von eindeutigen und ähnlichkeitsbasierenden Verfahren kombiniert. Die eindeutigen Verfahren ermöglichen eine hundertprozentige Wiedererkennung solange die benutzten Merkmale verfügbar sind und verbrauchen nur wenig Rechenleistung, da sie auf einfachen Vergleichsoperationen beruhen. Die ähnlichkeitsbasierenden Verfahren werden auf den Rest der Nutzer angewendet und können so die Nutzer zuordnen, die mit eindeutigen Verfahren nicht hätten zugeordnet werden können. Die für beide Verfahren notwendigen Merkmale müssen jedoch immer ausgelesen werden, da Nutzer, die die eindeutigen Merkmale deaktivieren, sonst nicht mehr erkannt werden können.

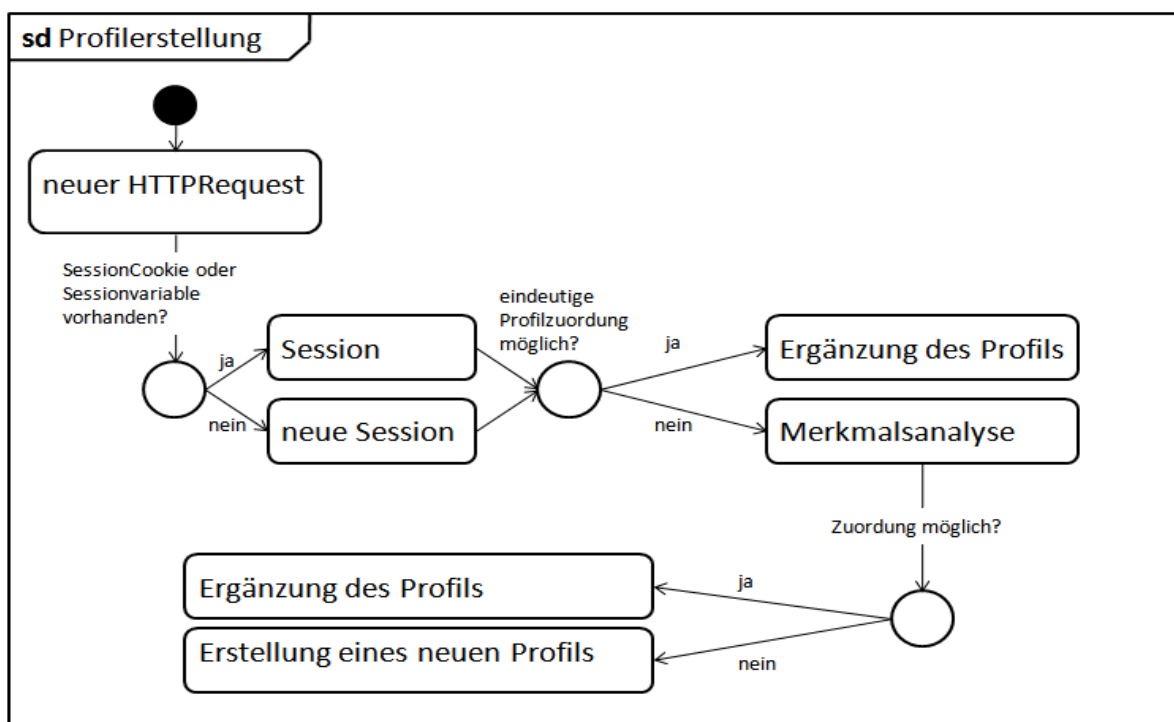


Abbildung 4 – Profilerstellung

Da bei den eindeutigen Verfahren immer alle Merkmale extrahiert werden müssen, ist es möglich diese Daten als Trainingsdaten für ein Künstliches-Neuronales-Netz (KNN) zu verwenden. Für die extrahierten Daten ist durch die eindeutige Zuordnung bereits bekannt zu welchem Profil die Daten gehören. Der sonst aufwendige Prozess zur Erstellung von Trainingsdaten fällt in diesem Fall weg. Das so trainierte KNN könnte dann für die ähnlichkeitsbasierte Profilzuordnung genutzt werden.

3.3 Nutzeridentifikation durch Dritte

Der Unterschied zwischen normalen Webseitenanbietern und Drittanbietern ist, dass Inhalte von Dritten durch den Nutzer nicht direkt aufgerufen werden. Für einen Webseitenanbieter ist es möglich, Inhalte von anderen Anbietern in die eigene Seite einzubinden. Diese eingebetteten Informationen können Grafiken, Iframes, Skripte und andere Objekte sein. Entscheidend ist, dass die Inhalte vom Server eines anderen Anbieters kommen. Wenn der Nutzer eine Webseite aufruft, werden Inhalte von Dritten automatisch mit geladen. Der Vorteil für einen Anbieter, der sich in der Position des Dritten befindet ist die Möglichkeit das Verhalten und die Interessen eines Nutzers über mehrere Seiten zu verfolgen (siehe Abbildung 5). Das entstehende Profil des Nutzers wird dadurch genauer.

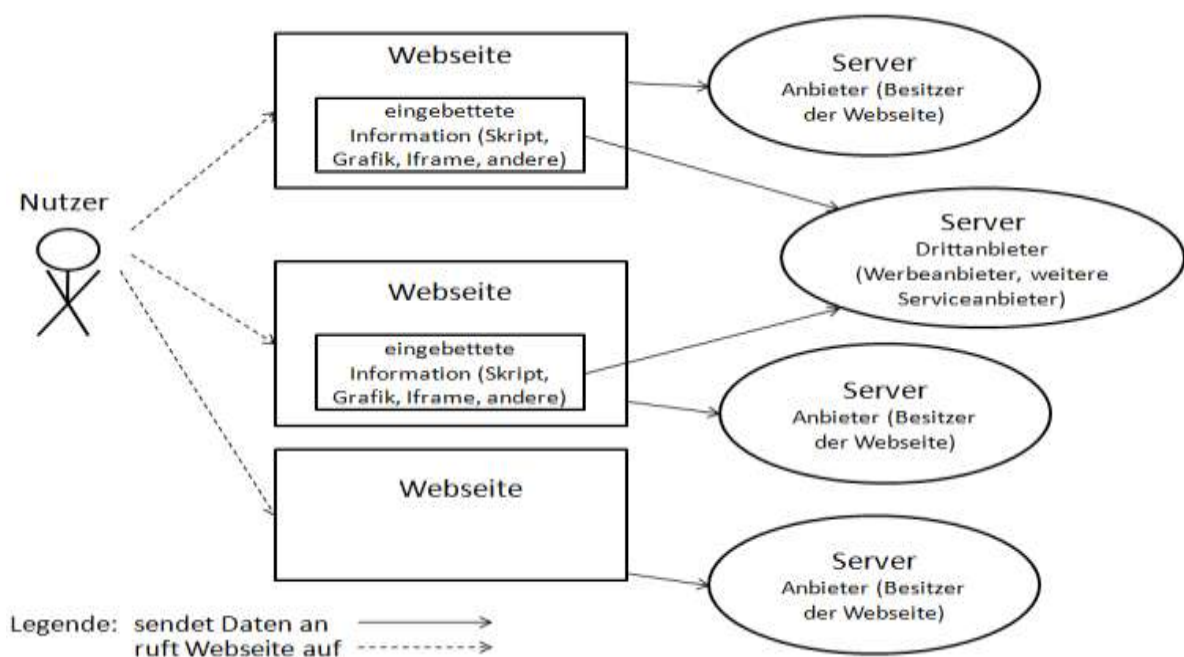


Abbildung 5 – Drittanbieter

Der Aufruf einer Webseite allein genügt um Daten an die jeweiligen Server zu senden. Durch aktive Inhalte auf einer Webseite wie zum Beispiel JavaScript, Flash oder Java können jedoch weitere detaillierte Informationen an die betroffenen Server gesendet werden. Die Same Origin Policy verhindert jedoch eine direkte Kommunikation zwischen dem Webseitenbesitzer und dem Drittanbieter innerhalb des geöffneten Fensters.

3.4 Methoden zur eindeutigen Profilzuordnung

3.4.1 Passwortbasierte Authentifizierung

Die sicherste Variante, um Sessiondaten einem Profil zuzuordnen, ist eine passwortbasierte Authentifizierung. Bei dieser Methode muss sich der Nutzer durch die Eingabe seines Nutzernamens und seines Passwortes gegenüber dem System authentifizieren. Das System vergleicht die Nutzereingaben mit den gespeicherten Daten und erteilt einen autorisierten Zugriff.

Das Profil zur Identifizierung des Nutzers besteht in diesem Fall nur aus dem Nutzernamen und dem Passwort. Eine vollständige Nutzeridentifikation ist jedoch nur dann gegeben, wenn der Nutzer seinen realen Namen angegeben hat und so die Zuordnung des Profils zur realen Person möglich ist.

Eine andere Variante der passwortgestützten Authentifizierung wäre die Anmeldung unter einem Pseudonym. Dies ist das in Internetforen übliche Verfahren. Die Zuordnung zu einer realen Person ist in diesem Fall nicht ohne weitere Daten möglich.

Die Verwendung von Passwörtern ist jedoch nicht immer sinnvoll und wird zu Gunsten des Komfort für den Nutzer durch andere Methoden, wie zum Beispiel Cookies, ersetzt.

3.4.2 Cookies

Cookies wurden eingeführt, da das zum Übertragen von Webseiten benutzte Protokoll (HTTP) zustandslos ist. Dies führt vor allem dazu, dass ein Webserver jeden Request gleich behandelt, egal von welchem Nutzer dieser kommt. Für viele Anwendungsgebiete wie zum Beispiel Webshops, wirkt sich dies jedoch negativ aus, da Sessions nur sehr aufwendig durch spezielle Variablen in den URL's abgesichert werden können. Auf diese Weise können jedoch auch Dritte Zugriff auf den Account erhalten, da beim Kopieren der URL auch die Sessionvariable mit kopiert wird. [Net97]

Cookies sind somit eine Technik, die vor allem aufgrund der erhöhten Sicherheit für den Nutzer eingeführt wurden. Der Inhalt der Sessionvariablen wird auf dem PC des Nutzers in Form eines HTTP-Cookies gespeichert und bei jedem Request im Header mit zum Server gesendet. [Sch00]

Es existieren zwei unterschiedliche Arten von Cookies. Dies sind zum einen, die schon erwähnten Session-Cookies und zum anderen die persistenten Cookies. Während Session-Cookies am Ende einer Session (üblicherweise beim Schließen des Browsers) gelöscht werden, dienen persistente Cookies der Wiedererkennung des Nutzers über einen langen Zeitraum. [Dwo10]

Um den Komfort für den Nutzer zu erhöhen, haben viele Webdienste persistente Cookies eingeführt. Dadurch kann sich der Nutzer ohne erneute Passworteingabe auf einer Webseite anmelden. Außerdem können individuelle Einstellungen des Nutzers in einem solchen Cookie gespeichert werden.

Problematisch sind diese Cookies, wenn zum Beispiel Inhaber eines Werbenetzwerkes persistente Cookies setzen, um die Internetaktivitäten ihrer Nutzer zu verfolgen. Es ist einem Werbeanbieter dadurch möglich, Informationen über die Nutzer zu sammeln, um diese noch gezielter mit Werbung zu versorgen (engl. Behavioral Targeting). Die automatisch angelegten Nutzerprofile über besuchte Seiten lassen, je nachdem welche Seiten tatsächlich besucht und vom Werbeanbieter registriert wurden, Rückschlüsse auf den Familienstand, Hobbies, politische Ansichten und andere Interessen zu. [YLW09]

Um die ungewünschten Eigenschaften von Cookies zu unterdrücken, können diese vom Nutzer selbst gelöscht werden. Es ist außerdem in den meisten aktuellen Browsern möglich seitenspezifische Einstellungen vorzunehmen, um Cookies immer oder niemals anzunehmen. Es hat sich inzwischen bewährt, Drittanbietern das Setzen von Cookies automatisch zu verweigern, da diese oft nur zur Verfolgung der Nutzer (engl. Tracking) dienen. Im IE 9, im Firefox 4, im Safari 5 und im Opera 11 ist diese Funktion bereits enthalten. Die Verwaltung von Cookies kann durch den Einsatz von Cookiemanager wesentlich vereinfacht werden.

Cookies lassen sich außerdem, aufgrund der Art wie sie Informationen speichern, in zwei Kategorien aufteilen. Zu der ersten Kategorie gehören alle Cookies, die vorhandene Mechanismen zum Speichern von Daten nutzen, um eine Identifikationsnummer für den Nutzer (Nutzer-ID) zu hinterlegen.

Die zweite Kategorie beinhaltet Cookiearten, die einen temporären Speicher durch geschickte Zugriffe dazu bringen, ebenfalls eine Nutzer-ID zu speichern, die später wieder ausgelesen werden kann. Grundsätzlich wird bei dieser zweiten Art von Cookies der Umstand ausgenutzt, dass sowohl der Browser als auch die Plugins, Daten temporär zwischenspeichern, um diese bei einem erneuten Besuch der Seite schneller wieder laden zu können.

An PC's, an denen mehrere Nutzer arbeiten, führen Cookies zu einem weiteren Problem. Auf vielen Internetseiten ist es beim Login möglich einen Haken für „angemeldet bleiben“ zu setzen. Die Anmeldeinformationen werden dann in einem Cookie lokal gespeichert, um beim nächsten Besuch der Seite eine automatische Anmeldung zu ermöglichen. Mitbenutzer werden nun auch beim Besuch dieser Seite automatisch angemeldet und haben dadurch vollen Zugriff auf den fremden Nutzeraccount. Da dieses Problem nicht nur zu Hause, sondern vor allem in Internet-Cafés auftritt, verfügen aktuelle Browser über einen privaten Modus, der alle Informationen über die Internetnutzung des Nutzers beim Schließen des Browsers löschen soll.

3.4.2.1 HTTP-Cookie

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	Text
Speicherplatz	80kb (20x 4kb pro Domain)
Zugriff	HTTP, JavaScript
Zugriffsbeschränkung	Domain/Subdomain
Speicherdauer	unbeschränkt

Cookie-Schema 1 - HTTP-Cookie

HTTP-Cookies sind die älteste Cookieart. Diese Cookies wurden entwickelt, um die Sicherheit zu erhöhen und Sessions abzusichern. HTTP-Cookies sind in Textform gespeicherte Informationen, deren Größe auf 4kByte begrenzt ist. Der Client sendet, wenn Cookies gesetzt wurden, diese bei jeder neuen Anfrage mit an den Server. Cookies können so dazu dienen den Nutzer zu identifizieren. Neben einer ID zur Zuordnung der Sessiondaten zu einem Profil werden häufig weitere Informationen im Cookie abgelegt. Das sind zum Beispiel Farbeinstellungen des Nutzers in einem Forum oder im schlimmsten Fall, bei einem Web-Shop, die Kontodaten des Nutzers.[Net97] Ein HTTP-Cookie kann sowohl server- als auch clientseitig gesetzt werden.

```
[JavaScript]
document.cookie="NID=ID12345;expires="+date1.toGMTString()+" ";
[PHP]
setcookie("MultiCookieHTTP", "ID12345", time()+3600*24*100);
```

Quelltext 1

3.4.2.2 DOM-Storage

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	Liste (Text,Zahlen,Objekte)
Speicherplatz	mindestens 200kB browserabhängig
Zugriff	JavaScript
Zugriffsbeschränkung	Domain
Speicherdauer	unbeschränkt (Local-Storage) Dauer einer Session (Session-Storage)

Cookie-Schema 2 - DOM-Storage

DOM-Storage (Document Object Modell Storage) ermöglicht es Daten lokal auf dem Client zu speichern. Man unterscheidet dabei zwei Speichermöglichkeiten. Den so genannten Session-Storage, der beim Schließen des Browsers wieder gelöscht wird und den Local-Storage, der Daten einer Seite dauerhaft speichert.

Der Server hat keinen direkten Zugriff auf die im DOM-Storage gespeicherten Daten. Es ist nur dann möglich den Speicher zu nutzen, wenn der Client das Ausführen von Skripten (meist JavaScript) erlaubt. In diesem Fall können die Daten dann auch über AJAX (asynchronous JavaScript and XML) mit dem Server ausgetauscht werden.

DOM-Storage ermöglicht es Programmierern sehr einfach auf den Speicher zuzugreifen, um individuelle Einstellungen der Nutzer zu speichern. Es ist genau wie bei normalen Cookies auch in diesem Fall möglich eine eindeutige Nutzer-ID in diesem Speicher abzulegen. [W3C10]

```
[JavaScript]
sessionStorage.testText = "ID12345";
localStorage.testText = "ID12345";
```

Quelltext 2

3.4.2.3 Window.Name

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	Text
Speicherplatz	mindestens 2MB browserabhängig
Zugriff	JavaScript
Zugriffsbeschränkung	gleicher Tab (oder gleiches Fenster wenn keine Tabs existieren)
Speicherdauer	Dauer einer Session

Cookie-Schema 3 - Window.Name

Window.Name-Cookies [Fra08] stellen eine Besonderheit unter den Cookies dar. Im Gegensatz zu den meisten anderen Cookies ist über Window.Name ein Austausch von Informationen über Domaingrenzen hinweg möglich. Window.Name bezieht sich direkt auf den Tab beziehungsweise auf das Fenster, falls keine Tabs vorhanden sind. Solange der Tab existiert, bleiben gespeicherte Informationen in diesem Speicher erhalten. Dies ist auch dann der Fall, wenn der Nutzer die Webseite wechselt.

```
[JavaScript]
window.name="ID12345";
```

Quelltext 3

Würde also ein Soziales Netzwerk, die Profilidentifikationsnummer des Nutzers in Window.Name ablegen, so könnte ein Werbeanbieter diese auslesen, um seine eigenen Profildaten mit denen des Netzwerkes zu verknüpfen. Durch diese Maßnahme wäre eine vollständige Nutzeridentifikation möglich. Die folgende Abbildung 6 soll die Zugriffsmöglichkeiten auf Window.Name verdeutlichen.

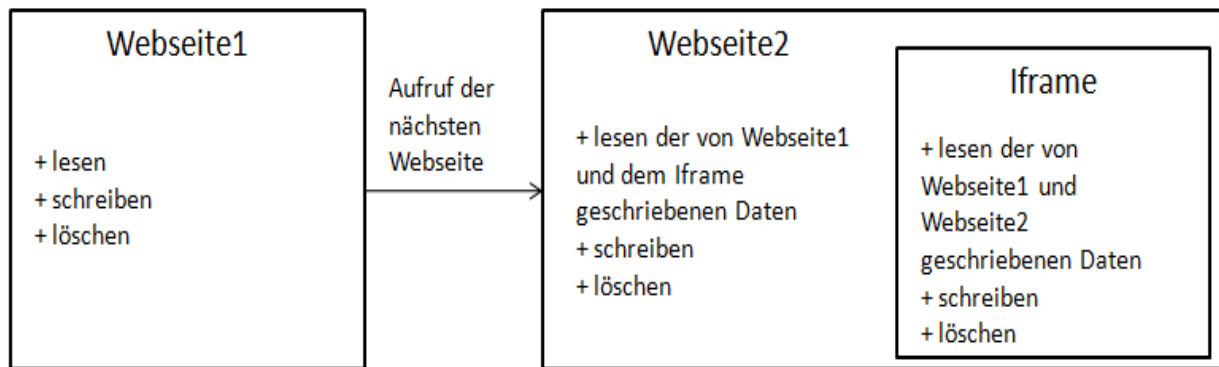


Abbildung 6 - Window.Name

3.4.2.4 IE-UserData

Browser	Internet Explorer (Version 5 - 8)
Speicherort	Browser
Art der Speicherung	Text
Speicherplatz	unbekannt
Zugriff	JavaScript
Zugriffsbeschränkung	Domain
Speicherdauer	unbeschränkt

Cookie-Schema 4 - IE-UserData

Dieser Speicher des Internet Explorer, besteht bereits seit der Version 5. Er wurde eingeführt, um die Eingaben von Nutzern auf einem Formular clientseitig zu speichern. Dieser Speicher kann jedoch genutzt werden, um beliebige Daten zu speichern. Dadurch ist es möglich eine Nutzer-ID in diesem Speicher abzulegen, um sie später wieder auszulesen und an den Server zu senden. Um auf diesen Speicher zuzugreifen, ist es nötig, dass das Ausführen von Skripten im Internet Explorer erlaubt ist.[Mic11]

```

[JavaScript]
el.setAttribute("multicookie", "ID12345");
el.save("multicookie");

```

Quelltext 4

3.4.2.5 Global-Storage

Browser	Firefox
Speicherort	Browser
Art der Speicherung	Liste (Text,Zahlen,Objekte)
Speicherplatz	unbekannt
Zugriff	JavaScript
Zugriffsbeschränkung	Domain/Subdomain
Speicherdauer	unbeschränkt

Cookie-Schema 5 - Global-Storage

Der Global-Storage ist ein Speicher, der nur im Firefox zur Verfügung steht. Mozilla zählt diesen Speicher zum DOM-Storage. Nach der W3C-Spezifikation gehören dazu jedoch nur der Local- und der Session-Storage. Im Gegensatz zum Local-Storage und zum Session-

Storage handelt es sich beim Global-Storage um eine Liste. Der Zugriff auf diese Liste kann entweder auf die eigene Domain oder auf eine Subdomain begrenzt werden. Ein Zugriff auf den Speicher fremder Seiten ist ausgeschlossen. [Moz11]

```
[JavaScript]
globalStorage['meinSeite.de'].name = "ID12345";
globalStorage['sub.meinSeite.de'].name = "ID12345";
```

Quelltext 5

Das Speichern einer Nutzer-ID kann nach dem Beispiel (Quelltext 5) erfolgen. Der Speicher wird beim Beenden des Firefox nicht automatisch gelöscht, sodass dieser auch beim nächsten Webseitenbesuch noch vorhanden ist, falls er nicht vom Nutzer gelöscht wurde.

3.4.2.6 Database-Storage

Browser	Chrome, Safari
Speicherort	Browser
Art der Speicherung	SQL-Datenbank
Speicherplatz	unbekannt
Zugriff	JavaScript
Zugriffsbeschränkung	Domain/Subdomain
Speicherdauer	unbeschränkt

Cookie-Schema 6 - Database-Storage

Database-Storage [Web07] ist ein besonderer Speicher, der bei einigen auf WebKit [Web11] basierenden Browsern verfügbar ist. Dies betrifft vor allem Safari und Chrome jedoch nicht den ebenfalls auf WebKit basierenden Android Browser. Im Gegensatz zu den anderen Speichermöglichkeiten, handelt es sich bei diesem Speicher um eine vollwertige SQLite-Datenbank. Diese wurde eingeführt, damit Webseiten ihre Funktionen auch offline zur Verfügung stellen können.

Eine clientseitige Datenbank hat den Vorteil, dass Bilder und andere Daten lokal gespeichert werden können und dadurch auch dann verfügbar sind, wenn keine Internetverbindung besteht. Dies hat für einige Webservices große Vorteile. Ein Beispiel wären Onlinenotizen oder Terminkalender, die lokal gespeichert werden und sich beim nächsten Besuch der Webseite synchronisieren.

Genau wie die anderen clientseitigen Speicher lässt sich auch der Database-Storage dazu nutzen, um einen Cookie auf dem Endgerät des Nutzers zu hinterlassen. Da es sich in diesem Fall jedoch um eine SQLite-Datenbank handelt, ist statt einer einfachen Zuweisung eine INSERT Operation nötig. Der folgende Quelltext 6 demonstriert wie eine Nutzer-ID in dieser Datenbank gespeichert werden kann.

```
[JavaScript]
tx.executeSql("
    INSERT OR REPLACE INTO multicookie (name,testText)
    VALUES(?,?)",
    ['test', "ID12345"],
    [],
    function (tx, result) {console.log(result)},
    function (tx, err) {console.log(err) }
);
```

Quelltext 6

3.4.2.7 WebCache

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	Bilder, Textdateien(HTML, XML, JSON, CSS, ...)
Speicherplatz	browserabhängig
Zugriff	HTTP
Zugriffsbeschränkung	Domain
Speicherdauer	unbestimmt (kann bei Platzmangel automatisch überschrieben werden)

Cookie-Schema 7 - WebCache

WebCache [Kam10] ist vor allem eine Möglichkeit des HTTP-Protokolls Daten temporär zu speichern, damit diese nicht immer wieder erneut geladen werden müssen. Diese temporäre Speichermöglichkeit bezieht sich sowohl auf den Client als auch auf Proxyserver, die sich im Kommunikationsweg zwischen Client und Server befinden.

Entscheidend für die Nutzung des WebCache sind die Informationen, die im Header eines HTTP-Requests oder einer Antwort übertragen werden. Der Wert Expires im Header der Antwort legt beispielsweise fest, wie lange eine Datei maximal zwischengespeichert werden darf. Da dieser Wert serverseitig festgelegt werden kann, gibt es somit keine Beschränkung für die Speicherdauer des Cookies.

Um den WebCache zum Speichern einer Nutzer-ID zu bewegen, wird auf dem Client mit Hilfe von JavaScript eine Datei vom Server angefordert (xmlHttpRequest), die die Nutzer-ID enthält. Falls der Browser die Datei bereits im Cache hat, so wird er diese laden und die Information kann mit dem Script ausgelesen werden. Andernfalls wird eine neue Datei vom Server geladen und im Cache des Clients gespeichert.

Einige Browser laden zunächst den Header der jeweiligen Datei vom Server, um zu überprüfen, ob diese verändert wurde. Die Datei kann dann gegebenenfalls aktualisiert werden. Aus diesem Grund ist es wichtig serverseitig unterscheiden zu können, ob der Client nun eine Datei aus dem Cache laden möchte oder eine neue Nutzer-ID zum Speichern anfordert. Dies kann dadurch geschehen, dass der Server GET und POST unterscheidet und entsprechend der Anfrage entweder antwortet, dass die Datei nicht

modifiziert wurde, oder die Datei mit der Nutzer-ID zurück sendet. Clientseitig muss das JavaScript dementsprechend die richtige Anfrage stellen. Durch geschickte Anwendung dieser Technik ist auch ein Überschreiben der im Cache gespeicherten Daten möglich. Die genaue Verwaltung des Caches ist jedoch von Browser zu Browser unterschiedlich, sodass WebCache als Speicher nicht besonders zuverlässig ist. Hinzu kommt, dass der Speicher für temporäre Dateien in den meisten Browsern stark begrenzt ist. Dadurch kann es leicht passieren, dass gespeicherte Daten aus Platzgründen überschrieben werden.

3.4.2.8 PNG-Caching

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	Bilder
Speicherplatz	browserabhängig
Zugriff	HTTP
Zugriffsbeschränkung	Domain
Speicherdauer	unbestimmt (kann bei Platzmangel automatisch überschrieben werden)

Cookie-Schema 8 - PNG-Caching

PNG-Caching stellt eine besondere Form der normalen WebCache-Methode dar. Es handelt sich dabei um eine neue (seit 2010) und weitestgehend unbekannte Methode, die zusammen mit der Cookie-Bibliothek Evercookie veröffentlicht wurde. [Kam10] Bei diesem Verfahren wird eine Nutzer-ID in Form von Farbwerten in eine PNG-Grafik eingebettet und auf dem Client gespeichert, um die ID später wieder auslesen zu können.

Es wird zunächst eine ganz normale Webseite im Browser des Nutzers geladen. Ein Script prüft dann auf dem Client, ob sich bereits eine präparierte Grafik im Cache befindet. Zu diesem Zweck wird ein Canvas-Objekt erstellt, in dem die Grafik geladen und die Farbwerte ausgelesen werden können. Diese können dann sofort an den Server zurückgesendet werden. Falls keine Grafik im Cache enthalten ist, wird eine neue Grafik vom Server angefordert, die dann automatisch vom Browser im Cache gespeichert wird. Serverseitig muss im Fall, dass eine neue Grafik angefordert wird, die Nutzer-ID in eine Reihe von Farbwerten umgewandelt werden. Durch die Darstellung von Farben in Form von RGB-Werten, lassen sich 3 Byte pro Pixel speichern.

Die Technik des PNG-Caching macht sich genau wie die WebCache-Methode zu nutze, dass der Browser Daten zwischenspeichert, um diese bei einem erneuten Seitenaufruf schneller laden zu können. Diese neue Cookie-Variante funktioniert jedoch nur in Browsern, die auch Canvas unterstützen. Ähnliche Techniken wären jedoch auch im Internet Explorer mit ActiveX möglich, sodass eine browserunabhängige Implementierung möglich wird. In älteren Browsern, die weder ActiveX-Steuerelemente noch Canvas unterstützen, funktioniert die Technik jedoch weiterhin nicht.

Der einzige Schutz besteht darin, beim Schließen des Browsers den Cache zu löschen. Dies geschieht bei aktuellen Browsern automatisch, wenn man die entsprechende Funktion aktiviert.

3.4.2.9 CSS-History-Cookie

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Browser
Art der Speicherung	ID (Text mit begrenztem Zeichensatz)
Speicherplatz	2048 Byte (abzüglich der Länge der URL)
Zugriff	HTTP
Zugriffsbeschränkung	keine
Speicherdauer	abhängig von der Speicherdauer des Webseitenverlaufs

Cookie-Schema 9 - CSS-History-Cookie

CSS-History-Cookies nutzen zum Speichern von Informationen den Fakt, dass Browser besuchte Links farblich von anderen noch nicht besuchten Links unterscheiden. Es ist somit möglich festzustellen, ob ein bestimmter Link bereits besucht wurde.

Durch den Einsatz von JavaScript ist es außerdem möglich Links zu besuchen, indem diese mit einem XMLHttpRequest aufgerufen werden. Um dieses Verhalten zum Speichern einer ID zu nutzen, werden zum Beispiel folgende Links nacheinander aufgerufen, um die ID 4192 zu speichern.

```
http://meineseite.de/4  
http://meineseite.de/41  
http://meineseite.de/419  
http://meineseite.de/4192
```

Um die Daten wieder auszulesen, werden nun zunächst die folgenden Links geprüft.

```
http://meineseite.de/0  
http://meineseite.de/1  
http://meineseite.de/2  
http://meineseite.de/3  
http://meineseite.de/4
```

Da `http://meineseite.de/4` bereits besucht wurde, beginnt der Algorithmus nun die nächste Ziffer zu überprüfen.

```
http://meineseite.de/40  
http://meineseite.de/41
```

Dieser Vorgang wird wiederholt, bis die gespeicherte Zahl vollständig ausgelesen ist. Eine auf den Ziffern 0-9 basierende ID kann somit mit nur 11 Zeichen 10^{11} verschiedene Codes bilden. Es ist möglich diese Technik zu nutzen, um NutzerIDs zu speichern.

Um die Anzahl der Requests an den Server zu senken, wäre es möglich die Anzahl der möglichen Symbole auf 2 zu verringern. Dadurch würden die URL's länger werden, jedoch wäre der Server entlastet, da die Anzahl der Requests stark abnimmt. [JJ06]

Der folgende Quelltext 7 zeigt die Funktionsweise des CSS-History-Cookie. Eine Nutzer-ID wird in einzelne Symbole zerlegt und gespeichert. Um den Link zu besuchen, wird ein Iframe genutzt.

```
[JavaScript]
var x='12345'; //Nutzer-ID
var url='empty.php?';
for(i=0;i<x.length;i++){
    url+= x[i];
    createIframe_CSSHistory(url);
}
```

Quelltext 7

3.4.2.10 Flash-Cookie

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Flash
Art der Speicherung	Text
Speicherplatz	100kB (Standard)
Zugriff	Flash (browserübergreifender Zugriff)
Zugriffsbeschränkung	Domain
Speicherdauer	unbeschränkt

Cookie-Schema 10 - Flash-Cookie

Der Flash-Cookie wird auch Local-Shared-Object (kurz: LSO) genannt und nach Angaben des Herstellers Adobe ist der aktuelle Flash Player 10 auf über 97% aller PC's installiert. [Ado10] Genau wie die normalen Cookies ermöglichen es Flash-Cookies den Nutzer eindeutig zu identifizieren. Da viele Internetnutzer nicht wissen, dass Flash-Cookies existieren, eignen sich Flash-Cookies zur Überwachung von Internet Nutzern besser als die normalen HTTP-Cookies. Flash-Cookies werden in einem speziellen Verzeichnis auf der Festplatte gespeichert und sind deshalb unabhängig vom jeweils verwendeten Browser. Es ist also sinnlos verschiedene Browser zu nutzen, um die Internetspuren auf diese Art zu verschleiern, da die Cookies zum Beispiel beim Surfen mit den IE gesetzt und nach einem Wechsel zum Firefox auch dort vorhanden sind. [Sch10] Der folgende Quelltext 8 zeigt wie der Flash-Cookie gespeichert wird.

```
[haXe C#]
so.data.MultiCookie = cookie;
```

Quelltext 8

Eine Untersuchung [SCM09] hat gezeigt, dass mehr als 50% der am meisten besuchten Seiten Flash-Cookies einsetzen. In diesen ist fast immer eine eindeutige ID enthalten, um den Nutzer zu identifizieren. Auffällig ist dabei, dass oft die selbe ID wie in den HTTP-Cookies gesetzt wird. Die Cookies stammen jedoch nicht ausschließlich von der Seite selbst sondern von Dritten (Werbeanbietern). Beim Besuch einer einzigen Seite können somit mehrere Flash-Cookies von der Seite selbst und von allen Werbeanbietern auf der Seite gesetzt werden.

Eine nutzerfreundliche Lösung dieses Problems wäre, wenn sich das Flash-Plugin an die Privacy-Einstellungen des verwendeten Browsers halten würde. Die Unabhängigkeit des Plugins wird von Sicherheitsexperten und Datenschützern seit Jahren kritisiert. [SCM09][McK08]

Seit der Version 10.3 gibt es Bemühungen die Privatsphäreneinstellungen verschiedener Browser zu übernehmen. Adobe hat dazu eine entsprechende Bibliothek veröffentlicht, die es einem Browser ermöglichen soll auf die Speicherung der Flash-Cookies Einfluss zu nehmen.

3.4.2.11 Java-Cache-Cookie

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Java
Art der Speicherung	Text, beliebige Dateien
Speicherplatz	1GB (Standard)
Zugriff	Java (browserübergreifender Zugriff)
Zugriffsbeschränkung	Domain
Speicherdauer	unbeschränkt

Cookie-Schema 11 - Java-Cache-Cookie

Im Rahmen dieser Arbeit wurde untersucht, ob es möglich ist, den Speicher des JRE als Speicher für eine Nutzer-ID zu nutzen. Zu diesem Zweck wurde ein Konzept erstellt, nach dem sich die Nutzer-ID in diesem Speicher ablegen und auslesen lässt. Der Java-Cache-Cookie ist die funktionierende Umsetzung dieses Konzeptes in Form einer neuen Cookieart und entstand ebenfalls im Rahmen dieser Arbeit. Der Java-Cache-Cookie wird in den noch folgenden Tests die Abwehrmechanismen und Privacy-Einstellungen der Browser auf die Probe stellen.

Java ist eine plattformunabhängige Programmiersprache. Die Sprache ermöglicht es, sogenannte Applets zu erstellen. Applets sind Programme, die direkt im Browser ausgeführt werden können. Damit ein Applet ausgeführt werden kann, benötigt man ein Java-Runtime-Environment (JRE). Dies ist eine virtuelle Maschine, die den Code übersetzt, damit er auf dem jeweiligen Endgerät ausgeführt werden kann. Außerdem wird ein Plugin für den jeweiligen Browser benötigt.

Da Applets zunächst auf dem Server liegen und vom Browser herunter geladen werden müssen, hat man einen lokalen Cache in das JRE integriert. In diesem Cache können Applets und JAR-Archive gespeichert werden, damit diese bei einem erneuten Besuch der Internetseite schneller geladen werden können.

JAR-Archive sind ein auf dem ZIP-Format basierendes Archivformat, das eingeführt wurde, um das Applet zusammen mit allen benötigten Ressourcen an den Client zu senden. Hauptziel dieser Entwicklung war es, den Datentransfer zum Client zu beschleunigen.

Damit Informationen dauerhaft gespeichert werden können, muss es sich entweder um eine ausführbare `.class` Datei oder um ein JAR-Archiv handeln. Da in einem JAR-Archiv neben Programmen auch Ressourcen gespeichert werden können, wurde das JAR-Archiv gewählt, um darin eine Textdatei zu speichern, welche die Nutzer-ID enthält. Das JAR-Archiv ist somit der Cookie. Die Java-Spezifikation zum Zwischenspeichern von JAR-Archiven sieht vor, dass der HTML-Headertag Expires für die Dauer der Speicherung entscheidend ist. Die Dauer ist somit unbeschränkt, da dieser Wert serverseitig festgelegt werden kann.

Das Speichern des Java-Cache-Cookies, erfolgt dann nach dem in Abbildung 7 dargestellten Schema.

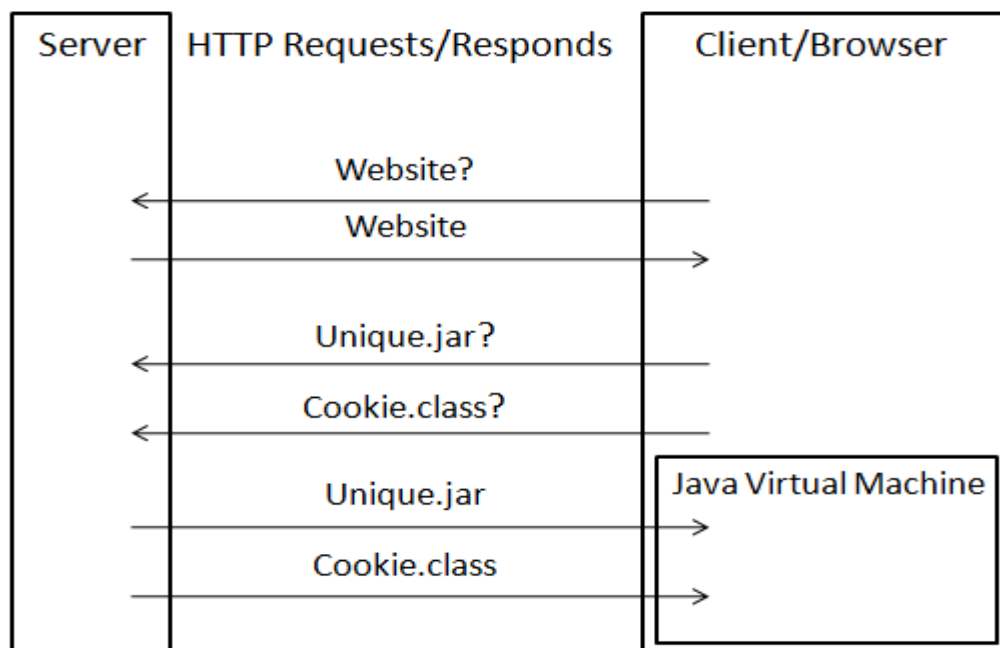


Abbildung 7 - Java-Cache-Cookie Save

Der Webseitenbesucher sendet mit dem Browser einen Request zum Webserver. Dieser gibt eine HTML-Seite zurück. Daraufhin werden die Requests, nach den in die HTML-Datei eingebetteten Ressourcen, zum Server gesendet. Dies sind in diesem Fall, die Unique.jar-Datei und die Cookie.class-Datei. Das gekennzeichnete JAR-Archiv wird

automatisch im Cache des JRE gespeichert, wenn es im Applet-Parameter `cache_archiv` angegeben wird. Das JAR-Archiv wird somit getrennt von dem eigentlichen Applet an den Client gesendet.

Das Laden des Java-Cache-Cookies, erfolgt dann nach dem in Abbildung 8 dargestellten Schema.

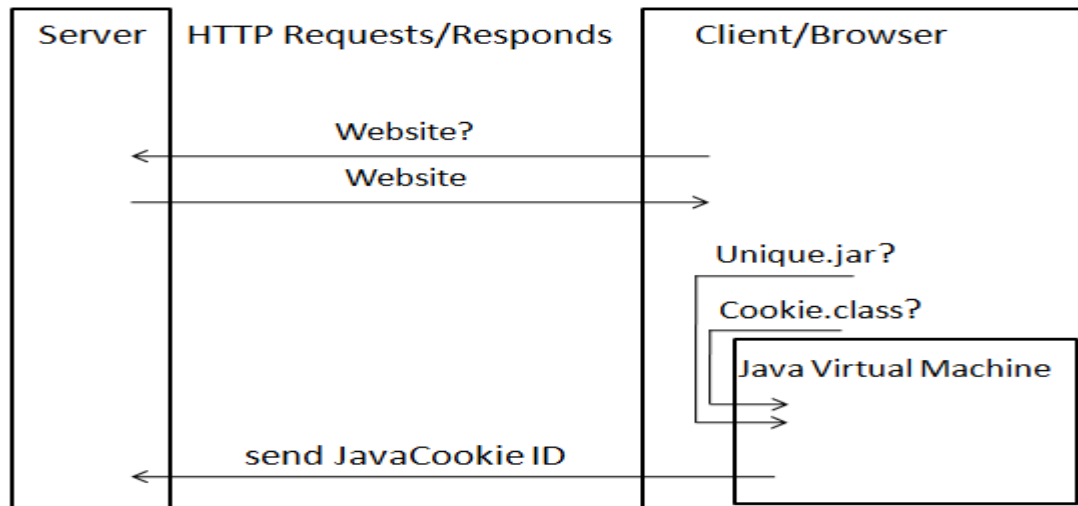


Abbildung 8 - Java-Cache-Cookie Load

Ähnlich wie beim Speichern wird auch in diesem Fall erst die HTML-Datei der Webseite geladen. Die Requests nach den Ressourcen werden jedoch umgelenkt, falls diese schon auf dem Client vorhanden sind. In dem Fall, dass die Dateien schon vorhanden sind, werden diese direkt aus der virtuellen Maschine geladen und das Applet ausgeführt. Dieses liest die Nutzer-ID des JAR-Archives aus und sendet die Nutzer-ID sofort an den Server zurück.

Serverseitig muss der gesamte Vorgang durch den Einsatz von Sessions gesichert werden, damit eine Zuordnung von Nutzer und Java-Cache-Cookie möglich wird. Der Einsatz von Sessionvariablen, die an die Requests angehängt werden, ist in diesem Fall nicht möglich. Die Sicherung der Session, muss somit über HTTP-Cookies erfolgen. Der Grund dafür ist, dass beim Cachen des JAR-Archiv die genaue URL berücksichtigt wird. Sessionvariablen würden in diesem Fall dazu führen, dass die Archive nicht als identisch erkannt werden und der Mechanismus zum Speichern und Auslesen des Java-Cache-Cookies nicht mehr funktioniert.

Der Java-Cache-Cookie beweist, dass neben dem Flash-Cookie auch weitere Plugins Möglichkeiten zum Speichern einer Nutzer-ID bieten können. Ein Application Programming Interface (API) zum Speichern von Cookies ist dazu nicht notwendig. Die einzige Voraussetzung ist, dass das Plugin einen eigenen beschreibbaren Speicher beinhaltet, der auch nach dem Beenden des Browsers nicht gelöscht wird.

3.4.2.12 Google-Gears

Browser	Chrome, Firefox, Internet Explorer, Opera, Safari
Speicherort	Gears (ist bei Chrome vorinstalliert)
Art der Speicherung	SQL-Datenbank
Speicherplatz	unbekannt
Zugriff	Gears (browserübergreifender Zugriff)
Zugriffsbeschränkung	Domain
Speicherdauer	unbeschränkt

Cookie-Schema 12 - Google Gears

Ähnlich wie der Database-Storage funktioniert auch die in das Google-Gears-Plugin integrierte SQL-Datenbank (Structured Query Language). Google-Gears ist im Chrome-Browser standardmäßig vorinstalliert und kann auch in anderen Browsern nachgerüstet werden. Gears soll vor allem Vorteile für Googles eigene Dienste bringen und diese offline verfügbar machen. Die Weiterentwicklung des Gears-Projekt durch Google wurde jedoch eingestellt, da es sich nicht wie erhofft durchsetzen konnte und die Entwicklung sich stattdessen auf HTML5 konzentriert. [Fet10]

```
[JavaScript]
geardb.execute("
    INSERT OR REPLACE INTO multicookie2(name,testText)
    VALUES(?,?)",
    ['test', "ID12345"]
);
```

Quelltext 9

3.4.3 Hardware ID's

Auf jedem Endgerät können eindeutige Merkmale wie zum Beispiel die Mac-Adresse und andere Identifikationsnummern der verbauten Hardware existieren. Das gezielte Auslesen dieser eindeutigen Merkmale wird jedoch durch verschiedene Sicherheitsmaßnahmen verhindert. Durch Sicherheitslücken kann es jedoch gelingen diese eindeutigen Merkmale dennoch auszulesen.

Im Bereich der mobilen Endgeräte (Smartphones, Handys, Tablets) hat man festgestellt, dass die Sicherheitsmaßnahmen durch das Installieren von Apps umgangen werden können. [Smi10] Das Problem ist das Design der Schnittstelle zwischen einer App und dem Betriebssystem. Statt dem Nutzer die Möglichkeit zu geben einer App den Zugriff auf verschiedene Funktionen zu erlauben, hat der Nutzer nur die Wahl entweder alle Zugriffe der App zu akzeptieren oder diese nicht zu benutzen.

Ein Beispiel wäre eine App zur Anzeige der aktuellen Aktienkurse, die gleichzeitig Geräteerkennung und GPS-Daten an den Anbieter schickt. Der Anbieter weiß nun, dass es sich sehr wahrscheinlich um eine vermögendere Person handelt und wo dieser Nutzer

sich aufhält. Unter Umständen sind sogar die Email-Adresse und der Name, sowie die Anschrift des Nutzers über die App ermittelbar. Der Nutzer bezahlt das Angebot somit durch die Preisgabe personenbezogener Informationen.

3.5 Methoden zur ähnlichkeitsbasierten Profilzuordnung

Ein völlig anderer Ansatz zur Nutzeridentifikation ist die Zuordnung einer Session zu einem Profil anhand von mehrdeutigen Merkmalen. Für einen Webseitenbetreiber hat diese Technik den Vorteil, dass keine Schreibrechte auf dem Endgerät des Nutzers vorhanden sein müssen. Die Merkmale des Nutzers und des verwendeten Endgerätes können verwendet werden, um diese zu einem Profil zusammenzufügen. Um eine solche Zuordnung zu ermöglichen, müssen jedoch zunächst die Merkmale ausgelesen oder aus den Daten extrahiert werden.

Für jedes einzelne Merkmal ist es möglich die Entropie des Merkmals zu bestimmen. Die Entropie ist ein Maß für die Unordnung (Zufälligkeit) in einem System. Es gilt: je höher die Unordnung, desto höher ist die Entropie. In einem System mit nur einem Element ist die Entropie immer 0, da keine Unordnung möglich ist. [Koc00]

Bezogen auf die Nutzeridentifikation gilt, je höher die Entropie eines bestimmten Merkmals ist, desto besser ist es geeignet Nutzer voneinander zu unterscheiden. Da die Merkmale jedoch untereinander korreliert sind, ist es nicht ohne Weiteres möglich die Entropie aller erfassten Merkmale zu bestimmen, oder die Merkmale unabhängig voneinander zu betrachten.

Entscheidend für die Zuordnung zu einem Profil ist der Grad der Ähnlichkeit. Jedes einzelne Merkmal einer Session kann mit den in den Profilen gespeicherten Merkmalen verglichen werden. Am Ende lässt sich ein Korrelationsfaktor bestimmen, der ausschlaggebend dafür ist, ob die Sessiondaten zu dem jeweiligen Profil gehören. Eine Zuordnung der Sessiondaten zu einem Profil muss jedoch nicht sofort geschehen. Änderungen am Profil, oder neue Technologien können auch zu einem späteren Zeitpunkt zu einer Zuordnung der Daten führen.

Der größte Nachteil der ähnlichkeitsbasierenden Profilzuordnung ist, dass gerade im Fall von Firmen, in denen auf allen PC's die gleiche Software läuft, eine Erkennung einzelner Nutzer nahezu unmöglich ist. In diesem Fall weisen alle PC's dieselben Merkmale aus. Ein Merkmal, das innerhalb einer Firma nur eine einzige Ausprägung hat, hat dann eine Entropie von 0 in dieser Firma. Dies bedeutet jedoch nicht, dass dieses Merkmal außerhalb der Firma auch gleich 0 ist. Außerhalb können weitere Ausprägungen des Merkmals existieren, wodurch auch die Entropie größer als 0 ist.

3.5.1 Schriftarten

Die installierten Schriftarten können sowohl mit einem JavaApplet als auch mit Hilfe von Flash ausgelesen werden. Das Ergebnis ist eine unsortierte Liste von Schriftarten. Bei der Untersuchung der Schriftarten ist nicht nur entscheidend welche Schriftarten installiert sind, sondern auch deren Reihenfolge.

Die Reihenfolge der Schriftarten hängt von deren Installationszeitpunkt ab und ändert sich nur, wenn neue Schriftarten hinzukommen oder entfernt werden. Einzige Ausnahme sind Betriebssysteme für den Mac. Dort existieren Programme, die unter Umständen die installierten Schriftarten überschreiben und dadurch ebenfalls deren Reihenfolge ändern. Das Besondere ist, dass die Liste der Schriftarten eine Entropie von 13,9Bit besitzt. Damit ist dieses Merkmal sogar aussagekräftiger als der UserAgent. [Eck11]

Neben dem Auslesen der Schriftarten über Plugins ist es möglich mit Hilfe von JavaScript zu testen, ob eine einzelne Schriftart installiert ist. Die Entropie des Ergebnisses ist jedoch kleiner, da es nicht möglich ist alle existierenden Schriftarten zu überprüfen und auch die Reihenfolge nicht bestimmbar ist.

3.5.2 UserAgent

Ein besonderes Merkmal zur Nutzeridentifikation ist der UserAgent. Der UserAgent ist eine Zeichenkette, die bei jedem HTTP-Request an den Server gesendet wird. Ein aktueller UserAgent könnte wie folgt aussehen:

```
Mozilla/5.0 (Windows; U; Windows NT 6.1; de; rv:1.9.2.8)
Gecko/20100722 Firefox/3.6.8 (.NET CLR 3.5.30729).
```

Tabelle 2 - UserAgent

Ausdruck	Beschreibung
Mozilla	Heute: Name der Anwendung Früher: Mozilla = Kennzeichen für die Unterstützung von Frames
5.0	Version der Anwendung
Windows	Betriebssystem
U	Sicherheitsstandard (N=kein, I=schwach, U=stark)
Windows NT 6.1	Betriebssystemversion: Windows 7
de	Sprache: deutsch
rv:1.9.2.8	Gecko Version
Gecko	JavaScript Engine
20100722	Erstellungsdatum des Browsers
Firefox	Browsersname
3.6.8	Browserversion
.NET CLR 3.5.30729	.NET Runtime und Versionsnummer

Die Tabelle 2 zeigt den Aufbau eines UserAgent mit den daraus extrahierbaren Informationen. Der UserAgent enthält eine Menge von Informationen, über die verwendete Software und deren Versionen. Die Entropie des Merkmals UserAgent ist rund 10Bit. [Eck11] Dies genügt, um kleinere Mengen von Nutzern allein anhand ihres UserAgent voneinander zu unterscheiden.

Der Informationsgehalt des UserAgent kann durch die Installation bestimmter Software noch größer werden. Toolbars, wie zum Beispiel die Ask-Toolbar, tragen sich beispielsweise in den UserAgent ein und erhöhen somit dessen Informationsgehalt, wodurch der Nutzer leichter identifizierbar wird. Die aktuelle .Net-Runtime verändert ebenfalls den UserAgent wie in Tabelle 2 zu sehen ist.

Um einen beliebigen Menschen eindeutig zu identifizieren, ist ein Informationsgehalt von rund 33Bit [Eck10] nötig. Deshalb ist es nicht möglich, jedem Menschen eindeutig einen einzigen UserAgent zu zuordnen, da dieser einen mittleren Informationsgehalt von nur 10Bit besitzt. Dennoch ist der UserAgent ein Merkmal, dass die Gruppe von Nutzern stark eingrenzt und dadurch deren Verfolgung erleichtert. Für viele Anwendungsgebiete ist es jedoch nicht problematisch, dass eine Merkmalskombination auf eine Gruppe von Individuen passt.

3.5.3 History

3.5.3.1 WebTimingAttack

Eine Möglichkeit mehr über die Gewohnheiten eines Webseitenbesuchers zu erfahren, sind so genannte WebTimingAttacks. Dabei macht man sich zunutze, dass der Browser Bilder von Webseiten im Cache behält, um diese später schneller wieder laden zu können. Durch dieses Verhalten entsteht eine deutliche Zeitdifferenz beim Laden der Bilder. Bilder von besuchten Seiten werden deutlich schneller geladen als Bilder von Seiten die der Nutzer noch nicht besucht hat. Anhand der Ladezeiten eines Bildes lässt sich somit feststellen, ob eine Seite besucht wurde. Es wurde bereits gezeigt [FeS02], dass dieser Angriff mit einem Java-Applet durchgeführt werden kann.

Im Rahmen dieser Arbeit wurde untersucht, ob sich das Verhalten der Browser auch mit reinem JavaScript nachvollziehen lässt. Die Untersuchung hat gezeigt, dass ein deutlicher Zeitunterschied mit Hilfe von JavaScript messbar ist. Die Untersuchung hat außerdem gezeigt, dass die Wahl größerer Bilder zu größeren Differenzen bei den Ladezeiten führt. Bei einer hohen Übertragungsgeschwindigkeit und einem zu kleinen Bild ist der Test nicht mehr aussagekräftig genug, um Rückschlüsse zu auf den Besuch der Webseite zu ziehen. Die Bilder selbst müssen für den Besucher der eigentlichen Webseite nicht sichtbar sein. Es genügt diese im Hintergrund mit Hilfe des Imageobjektes zu laden.

Um dieses Verhalten des Browser auszunutzen, kommen mehrere Szenarien in Frage. Die einfachste Variante ist, dass der Webseitenbesitzer selbst wissen möchte, welche Vorlieben seine Kunden haben und entsprechende Messungen auf seiner Seite vornimmt.

Weitere Möglichkeiten entstehen durch Iframes, wodurch Dritte diesen Test durchführen können. Eine weitere Möglichkeit sind HTML-E-mails, die mit Hilfe von Iframes Messungen vornehmen und Daten zurücksenden.

Für die Erkennung von Nutzern ist diese Technik deshalb interessant, weil es möglich ist für jeden Nutzer eine Menge an besuchten und nicht besuchten Seiten zu bestimmen. Diese Mengen können auch als weiteres Merkmal für Zuordnung des Nutzers zu einem Profil dienen.

3.5.3.2 CSS-HistoryAttack

Ähnlich wie die WebTimingAttack funktioniert auch die CSS-HistoryAttack. Bei dieser Technik wird der Umstand ausgenutzt, dass Browser besuchte Links farbig markieren. Um nun bei einer größeren Menge an Webseiten zu testen, ob der aktuelle Nutzer diese besucht hat ist es nur notwendig die Links zu generieren und deren Farbwert auszulesen. Die generierten Links können dabei für den Nutzer unsichtbar sein, sodass das Ausspähen unbemerkt bleibt.

Im Gegensatz zur WebTimingAttack werden keine Verbindungen zu anderen Servern hergestellt, wodurch sich in kurzer Zeit wesentlich mehr Seiten testen lassen, da nur im Verlauf des Browsers anhand der Farbe geprüft wird, ob eine Seite besucht wurde oder nicht.

Es ist sogar möglich diesen Angriff auf den Browserverlauf ohne JavaScript durchzuführen. Dazu werden die zu prüfenden Links serverseitig in das HTML-Dokument eingebettet. Durch den Einsatz von CSS ist es möglich diese für den Benutzer unsichtbar zu machen. Jeder Link bekommt zusätzlich ein individuelles Hintergrundbild nach folgendem Schema (Quelltext 10). [Han11]

```
[CSS]
#links a:visited span.span1 {
display:block;
background: url(CSS-history.cgi?linkId=1);
}
#links a:visited span.span2 {
display:block;
background: url(CSS-history.cgi?linkId=2);
}
```

Quelltext 10

Für das Funktionieren dieses Angriffs ist die bedingte Aktivierung von `a:visited` entscheidend. Nur wenn ein Link besucht wurde, wird ein Request an den Server gesendet, um das Hintergrundbild für den Link zu laden. Da jeder Link einen anderen Request an den Server sendet, kann der Server die besuchten Seiten anhand der Request feststellen.

Auch bei dieser Technik lässt sich die Menge der besuchten und der nicht besuchten Seiten bestimmen, wodurch eine Zuordnung des Webseitenbesucher zu einem bereits bestehenden Profil ermöglicht werden kann.

Diese Lücke wurde jedoch in den Browsern Internet Explorer, Firefox, Chrome, Safari und Opera zwischen 2010 und Mitte 2011 geschlossen, sodass die jeweils neuesten Versionen der Browser immun gegen diese Art des Auslesens von Informationen sind.

3.5.4 Autocomplete

In einigen älteren Browsern ist es möglich, im Hintergrund ein verstecktes HTML-Formular zu erstellen und durch geschickte Auswahl der Namen der Inputfelder den Browser dazu zu bringen, das Formular automatisch auszufüllen. Um den Browser zum Ausfüllen des Formulars zu bewegen sind nur 2 Schritte nötig. Das jeweilige Inputfeld muss den Focus bekommen und es muss ein künstliches Tippen auf den Pfeiltasten simuliert werden. Beides ist durch die Nutzung von JavaScript möglich, ohne dass der Nutzer dies bemerkt. Durch diese Technik lassen sich die Nutzereingaben auslesen, die der Nutzer bei einem Inputfeld, welches die gleiche ID besitzt, zuvor eingegeben hatte. Dies gelingt auch, wenn das ursprüngliche Inputfeld zu einer anderen Webseite gehört. Es ist dabei durchaus möglich Merkmale wie Name, Anschrift, aber auch die Email-Adressen auszulesen. [Gro10][GrJ10]

Bei neueren Browsern, die über eine Autocomplete-Funktion verfügen, ist es nicht möglich diesen Angriff automatisch ablaufen zu lassen. Es ist die Hilfe des Nutzers nötig, der dazu gebracht werden muss, die entsprechenden Tasten selbst zu drücken. Da sich die meisten Autocomplete-Funktionen über die Pfeiltasten steuern lassen, reicht es den Nutzer ein Spiel spielen zu lassen, bei dem die Tasten „Pfeil Nach Unten“ gefolgt von „Enter“ besonders oft kombiniert werden. Dies genügt, um im Hintergrund die gewünschten Informationen auszulesen.

3.5.5 Ports/Programme

Es ist möglich ein sogenanntes Fingerprinting durchzuführen und den Client nach offenen Ports zu scannen. Die offenen Ports selbst und die Antworten der jeweiligen Programme können Informationen enthalten, die es erleichtern einen Nutzer eindeutig zu identifizieren.

Problematisch bei diesem Ansatz zur Nutzeridentifikation ist jedoch, dass dieses Vorgehen auch als Vorbereitung einer Straftat nach StGB §202c angesehen werden kann, da sich auf diese Weise verwundbare Lücken in der Abwehr des Clients entdecken lassen. Da nicht eindeutig klar ist, in wie weit der Fingerprint bereits als unerlaubtes Ausspähen von Daten oder dessen Vorbereitung gewertet werden kann, ist ein solches Vorgehen in Deutschland zumindest fragwürdig.

Die Grundidee des Portscans bei einem Nutzer ist, dass jeder einzelne PC abhängig von der installierten Software ein eigenes Profil geöffneter und geschlossener Ports aufweist. Es ist außerdem möglich, dass bestimmte Programme aufgrund ihres Protokolls eine eindeutig Identifizierung des Nutzers ermöglichen.

Da es heutzutage üblich ist, dass private Netzwerke hinter einem Router mit NAT-Technologie liegen, ist die Wahrscheinlichkeit auf offene Ports zu stoßen eher gering. Unternehmen schotten ihre internen Netze durch den Einsatz von Proxyservern ab, sodass auch in diesem Fall ein Fingerprinting wenig erfolgsversprechend ist.

3.5.6 IP-Adresse

Man unterscheidet 2 Arten von IP-Adressen. Das sind zum einen statische Adressen, die fest an einen bestimmten Server gebunden sind und zum anderen dynamische Adressen, die häufig den Besitzer wechseln. Statische Adressen werden vor allem im Intranet von Firmen oder aber von wichtigen Servern (zum Beispiel DNS-Servern) im Internet genutzt.

Da Internetnutzer häufig eine neue dynamische IP-Adresse zugewiesen bekommen, ist die Adresse allein nur wenig interessant. Anhand der IP-Adresse lassen sich jedoch weitere Informationen ermitteln, die sich wesentlich besser für die Wiedererkennung eignen. Dies sind vor allem der Internetprovider und der ungefähre Standort des Nutzers.

3.5.7 Standortdaten

Die Standortdaten können dazu beitragen, einen mobilen Internetnutzer einem bereits gespeicherten Profil zuzuordnen. Um an die Standortdaten eines Nutzers zu gelangen, gibt es mehrere Möglichkeiten, die im Folgendem betrachtet werden.

3.5.7.1 IP-Adressen basierte Standortbestimmung

IP basierte Dienste zur Geolokalisation sind bereits stark verbreitet. Die Dienste machen sich zunutze, dass IP-Ranges auf regionale Internetserviceprovider (ISP) verteilt sind. Dadurch wird es möglich, eine grobe Regionszuordnung durchzuführen. Da die jeweiligen ISP ihre IP-Adressen ebenfalls in Bereiche aufteilen, um diese dann bestimmten kleineren Gebieten zuzuordnen, ist es möglich eine genauere Positionsbestimmung anhand der IP-Adresse durchzuführen. In den meisten Fällen lässt sich anhand der IP-Adresse die Stadt bestimmen, in der der Besitzer der IP-Adresse sich aufhält.

Die Frage, ob IP-Adressen personenbezogene Daten sind, ist noch immer nicht eindeutig geklärt. Es gibt jedoch deutliche Tendenzen in Richtung einer Bejahung dieser Frage. Um die Zuordnung von IP-Adressen zu einer Person zu vermeiden, ist es möglich diese durch das Löschen von Bits am Ende der Adresse teilweise zu anonymisieren. Die Bestimmung des Standortes ist selbst mit einer teilweise anonymisierten IP-Adresse möglich. Untersuchungen haben gezeigt, dass die letzten 8 Bits einer IP-Adresse für die Bestimmung des Ortes eine sehr geringe Relevanz besitzen.[Küh09]

3.5.7.2 Funkzellen zu Standortbestimmung

Die Funkzelle, in der sich ein Nutzer befindet, bietet eine weitere Möglichkeit zur Bestimmung des Standortes. Dies ist vor allem dadurch möglich, dass sich ein Mobiltelefon laufend bei einer Funkzelle meldet, damit Anrufe zu dem Mobiltelefon durchgestellt werden können. Für den Betreiber des Funknetzes ist es somit möglich, den Standort des Nutzers auf 100 bis 5000m (je nach Größe der Funkzelle) genau zu bestimmen. Die Nutzung dieser Daten für Services ist in Deutschland jedoch nicht ohne die Erlaubnis des Endnutzer erlaubt (TKG §98). Da die Erlaubnisanfrage bei potentiellen Kunden in dem betroffenen Gebiet jedoch eine Standortbestimmung einschließen würde, wurde diese Möglichkeit der Standortbestimmung bisher hauptsächlich für Notfälle benutzt.

3.5.7.3 LBS und GPS

Location Based Services (LBS) stellen eine weitere Möglichkeit dar, um an Standortdaten zu gelangen. LBS sind Dienstleistungen die auf ein bestimmtes Gebiet begrenzt sind. Damit diese Services funktionieren, ist es jedoch wichtig, die Position der jeweiligen Nutzer zu kennen. Dies geschieht hauptsächlich über Anwendungen für Smartphones (Apps), die auf das in vielen Smartphones integrierte GPS-Modul (Global Positioning System) zugreifen.

Durch das GPS-Modul ist eine sehr genaue Bestimmung des Standortes möglich. Die GPS-Module arbeiten passiv und können die Positionsdaten nur empfangen. Um die Daten dennoch zum Anbieter des Webservices zu bekommen, werden die Apps eingesetzt. [Sch07, S. 56ff]

Dem Nutzer steht ein vielfältiges Angebot an Diensten, wie zum Beispiel Navigationsdienste zur Verfügung. Ein Stadtplan wird durch diese Dienste überflüssig. Dies geschieht jedoch zu dem Preis, dass jede Bewegung durch den Dienstanbieter verfolgt werden kann. Um dem Kunden Kartendaten und weitere Informationen zur Verfügung zu stellen, werden die GPS-Daten an einen Webservice gesendet, der dann das Kartenmaterial an das Mobiltelefon überträgt oder bereits auf dem Gerät gespeichertes Material lädt. Zur Finanzierung dieser Dienste wird häufig ortsbezogene Werbung genutzt, indem zum Beispiel Restaurants aus der näheren Umgebung angezeigt werden.

Inzwischen existieren auch Webservices, die nicht nur die eigene Position auf einer Karte anzeigen, sondern auch die von registrierten Freunden. Zu den Anbietern von Diensten dieser Art gehören vor allem soziale Netzwerke. Neben dem inzwischen sehr bekannten Dienst „Places“ von Facebook existieren weitere Dienste, die Ähnliches anbieten. Darunter sind zum Beispiel Google mit seinem Dienst Latitude, und die sozialen Netzwerke Foursquare und Mologogo, bei denen das Ermitteln des Aufenthaltsortes von Freunden und Bekannten sogar ein zentraler Bestandteil ist.

3.5.8 Implizites Wissen

In jedem einzelnen Nutzerprofil ist neben den sicheren Merkmalen auch Wissen enthalten, welches unsicher ist, wofür sich jedoch eine Wahrscheinlichkeit berechnen lässt. Die für die Berechnung der Wahrscheinlichkeit verwendeten Algorithmen stammen aus dem Bereich der Empfehlungssysteme.

Zur Verdeutlichung des Prinzips soll ein praktisches Beispiel dienen. Ein Nutzer, der in einem Onlineshop eine High-End-Grafikkarte gekauft hat, interessiert sich häufig auch für Prozessoren, Computerspiele oder CAD-Programme. Anhand bestimmter Merkmale des

Nutzerprofils, in diesem Fall der Kauf einer Grafikkarte, ist es möglich auf weitere Interessen und Merkmale zu schließen, beziehungsweise dem Nutzer weitere Produkte und Dienstleistungen zu empfehlen.

Grundlage für die Berechnung sind die Profile der Nutzer. Nutzer die zuvor ähnliche Merkmale und Interessen gezeigt haben, wie der aktuelle Nutzer, haben auch ein weiteres Interesse gezeigt. Es ist also wahrscheinlich, dass der aktuelle Nutzer auch dieses Interesse zeigen wird. Dabei geht man von der Annahme aus, dass die Wahrscheinlichkeit auf weitere gemeinsame Interessen um so größer ist, je ähnlicher zwei Nutzerprofile sind. [BHK98]

Die Datenbasis ist in den Merkmalen und Interessen beziehungsweise dem Verhalten der Nutzer implizit enthalten und kann mit den Techniken des Data-Mining extrahiert werden. Zur Extraktion dieser Daten werden sowohl speicherbasierte als auch modellbasierte Algorithmen eingesetzt. Speicherbasierte Algorithmen nutzen die gesamte Datenbasis, um Empfehlungen für den aktuellen Nutzer zu generieren. Modellbasierte Algorithmen erstellen anhand der Daten zunächst ein Modell, welches später zur Generierung der Empfehlungen genutzt werden kann. [KNN08][BHK98]

Die Empfehlungen stellen eine Form von neuem Wissen über den Nutzer dar. Das durch diese Verfahren generierte zusätzliche Wissen kann wiederum dazu dienen, das Profil des Nutzers zu ergänzen. Da dadurch mehr Merkmale des Nutzers bekannt sind, kann die Nutzeridentifikation auf diese Weise verbessert werden. Es wäre möglich, dass die Fehlerrate einer ähnlichkeitsbasierten Zuordnung durch dieses zusätzliche Wissen gesenkt werden kann. Entsprechende Untersuchungen stehen jedoch noch aus.

3.6 Tracking

3.6.1 Allgemein

Tracker sind Werkzeuge, die den Schritt der Profilerzeugung und den der Profilzuordnung (Wiedererkennung) automatisieren. Dabei können sie mehrere Methoden kombinieren, um höhere Erkennungsraten zu erzielen. Neben der Automatisierung der Profilerzeugung und Profilzuordnung, bieten viele Tracker die Möglichkeit die erhobenen Daten statistisch auszuwerten.

Tracker werden vor allem im Rahmen von Analysewerkzeugen für Webseitenadministratoren eingesetzt. Einem Webseitenadministrator wird es so ermöglicht genaue Nutzerzahlen zu ermitteln. Da viele Internetangebote sich allein durch Werbung finanzieren, sind diese Nutzerzahlen ausschlaggebend für die Einkünfte des Betreibers. Je mehr Nutzer eine Seite hat, desto höher sind im Allgemeinen auch die Einkünfte durch die Werbebanner.

Die Analysewerkzeuge können einem Administrator dabei helfen, die Werbung besser zu platzieren oder eine andere Optimierung an der Seite vorzunehmen, um beispielsweise mehr Nutzer anzulocken. Da zu den erhobenen Informationen auch Bildschirmauflösung und Farbtiefe gehören können, lassen sich Webseiten so anpassen, dass die Mehrheit der Nutzer eine optimierte Seite zu sehen bekommt. Damit eine statistische Auswertung jedoch möglich wird, muss jeder Nutzer wiedererkannt werden, damit er nicht doppelt gezählt wird. Die Wiedererkennung erfolgt durch die Zuordnung der aktuellen Session des Nutzers zu einem bereits bestehenden Profil.

Werbefirmen, die Banner auf Webseiten schalten, benutzen ebenfalls Tracker, um ihre Nutzer wiederzuerkennen. Dies geschieht vor allem mit Hilfe von Iframes und Webbugs. Ein Phänomen der letzten Jahre ist, dass neben den Werbeanbietern auch soziale Netzwerke anfangen diese Techniken einzusetzen, um ihre Nutzer außerhalb des eigenen Angebotes zu verfolgen.

Ein bekanntes Beispiel für einen solchen Tracker ist der „Gefällt mir“ Button von Facebook. Dieser Button ist in Wirklichkeit ein Iframe der mit Hilfe von Cookies Informationen mit dem Profil des Nutzers verknüpft. Damit Daten an Facebook gesendet werden, ist nur der Besuch einer Seite mit „Gefällt mir“ Button nötig und nicht das Anklicken des Buttons. Facebook kann somit jedem Nutzer eine Menge an besuchten Seiten zuordnen. Neben Facebook versuchen auch andere soziale Netzwerke ähnliche Trackingsysteme zu etablieren. Google versucht dies durch die Einführung eines „+1“ Button, der nach dem selben Prinzip funktionieren soll.

3.6.2 Informationsleck durch Iframes und Webbugs

Es ist bekannt, dass Werbenetzwerke und soziale Netzwerke über die eigene Seite hinaus Informationen über ihre Nutzer sammeln. Dabei kommen zwei verschiedene Techniken zum Einsatz. Zum einen sind das Webbugs und zum anderen Iframes. Als Webbugs bezeichnet man Grafiken, die oft nur 1x1 Pixel groß sind. Ein Iframe ist im Gegensatz zum Webbug keine Grafik, sondern eine vollständige HTML-Seite, die in eine andere Webseite eingebettet wird. Sowohl Webbugs als auch Iframes müssen nicht vom Server des Anbieters stammen sondern können einem Dritten gehören. Dieser Dritte ist dann in der Lage Informationen über das Surfverhalten der Nutzer zu sammeln.

Der Grund dafür, dass Dritte an diese Informationen gelangen ist, dass das HTTP-Protokoll. Es schreibt vor, dass bei jeder Anfrage an einen Server bereits gesetzte HTTP-Cookies mit gesendet werden müssen. Dies hat zur Folge, dass jeder Request gezielt einem Nutzer zugeordnet werden kann, falls eine entsprechende Nutzer-ID im Cookie enthalten ist.

Diese Funktionsweise machen sich Werbeanbieter und soziale Netzwerke zunutze, indem sie ihre Werbung (oder Like Buttons) auf einer ganzen Reihe von Webseiten einbinden, wodurch die Anfragen nach der Grafik oder dem Iframe an ihre eigenen Server gehen. Um als Dritter unterscheiden zu können von welcher Seite der Zugriff erfolgte, wird einfach die Seite des eigentlichen Anbieters als Variable in der URL übergeben.

```
[HTML]

<iframe "src=www.dritter.de/?b=besuchteseite"></iframe>
```

Quelltext 11

Alternativ ist es auch möglich, den Referer (die URL der verweisenden Domain) aus den Headerinformationen des HTTP-Request auszulesen. Dieser sollte ebenfalls die URL der Seite, in die der Iframe/Webbug eingebettet ist, enthalten.

Durch die Nutzung eines Iframes stehen dem Werbeanbieter sämtliche Möglichkeiten zum Setzen von Cookies oder zum Auslesen von Nutzerinformationen offen. Im Fall von Webbugs können nur HTTP-Cookies und die Headerinformationen des Requests ausgelesen werden. Beim Setzen neuer Cookies hängt es von den Einstellungen des Browsers ab, ob der Drittanbieter den Cookie setzen darf. In diesem Fall haben soziale Netzwerken einen Vorteil gegenüber den meisten Werbeanbietern. Besucht ein Nutzer die Seite eines sozialen Netzwerkes, so ist diese Seite kein Drittanbieter und kann somit Cookies setzen, um den Nutzer auch über die eigene Seite hinaus zu verfolgen. Werbeanbieter sind jedoch immer Dritte, solange der Nutzer nicht direkt die Seite eines Werbeanbieters besucht.

3.6.3 Cookie-Bibliotheken

Grundlage für das Funktionieren eines Trackers können Cookie-Bibliotheken (auch Cookie-APIs genannt), einzelne Cookiearten oder auch auf Ähnlichkeit basierende Verfahren sein. Cookie-Bibliotheken erleichtern eine eindeutige Profilzuordnung in dem sie Funktionen zum Setzen, oder Auslesen von Cookies bereitstellen. Beispiele für solche Cookie-Bibliotheken sind iLocalStorage [Dmi09] und Evercookie [Kam10]. Beide Bibliotheken ermöglichen das gleichzeitige Setzen von verschiedenen Cookiearten.

Evercookie stellt unter den Cookie-Bibliotheken eine Besonderheit dar. Die meisten bekannten Bibliotheken nutzen nur 1 oder 2 Cookiearten, wobei HTTP und Flash-Cookies am häufigsten verwendet werden. Evercookie unterstützt ganze 14 verschiedene Cookiearten und ermöglicht dadurch eine sehr zuverlässige und eindeutige Profilzuordnung. Diese Cookie-Bibliothek stellt einen evolutionären Sprung dar, da versucht wird die normalen Abwehrmechanismen gegen Cookies, durch das redundante Setzen von Cookies in verschiedene clientseitige Speicher zu umgehen. Gelöschte Cookiearten werden bei Evercookie automatisch neu gesetzt, solange auch nur eine einzige Cookieart übrig ist.

3.6.4 Tools zum „Tracken“ von Webseitenbesuchern

Die Techniken, die von den einzelnen Werkzeugen eingesetzt werden, können sich sehr voneinander unterscheiden. Das weit verbreitete Analysewerkzeug GoogleAnalytics erhebt eine Vielzahl von Daten und setzt zu dem auch noch Cookies. Eine Wiedererkennung des Nutzers ohne Cookie, wird von dem Tool jedoch nicht unterstützt. Im Rahmen der Untersuchung verschiedenster Tools konnte auch kein einziges gefunden werden, welches eine Erkennung ohne Cookies ermöglicht. Es ist jedoch nicht auszuschließen, dass Tracker in Zukunft auch eine Wiedererkennung ohne Cookies ermöglichen werden.

Tabelle 3 - Tools

Tracker	benötigt JavaScript	ermöglicht Wiedererkennung ohne Cookies	Besonderheit
Google Analytics	ja	nein	nur HTTP-Cookies, zahlreicher weitere Informationen (Plugins Fenstergröße Farbtiefe), Daten werden an Google gesendet
Facebook „Gefällt mir“ Button	nein, in manchen Fällen um den IFrame zu generieren	nein	Iframe, Cookie, übergibt die URL der Seite in die der IFrame eingebettet ist, Daten werden an Facebook gesendet
Piwik	ja(nein bei geringfügiger Modifikation)	nein	unterstützt anonymisierte IP-Adressen, Daten bleiben beim Webseitenanbieter

Die drei Tools (Tabelle 3) zum Tracken wurden ausgewählt, um die Unterschiede verschiedener Trackingtools aufzuzeigen. Die Untersuchung der Tools ergab, dass keiner der Tracker in der Lage ist den Nutzer ohne Cookies wiederzuerkennen.

Da es sich bei Google Analytics um ein JavaScript handelt, funktioniert dieser Tracker auch nur, wenn JavaScript für Google erlaubt ist. Der Facebook-Button kommt hingegen grundsätzlich ohne JavaScript aus, jedoch wird auf vielen Seiten ein JavaScript zum einbetten des Buttons genutzt, wodurch dieser dann ohne JavaScript nicht mehr erscheint. Piwik benötigt grundsätzlich JavaScript zum Tracken der Nutzer. Durch eine kleine Modifikation ist es jedoch auch möglich ohne JavaScript Nutzer zu erfassen.

Aus datenschutztechnischer Sicht schneidet Piwik am besten ab, da im Gegensatz zu den beiden anderen Trackern keine Daten an Dritte gesendet werden und nur der Webseitenbesitzer selbst Einsicht in die Daten erhält.

4 Abwehrmaßnahmen

Die Browser besitzen schon in der Standardinstallation ohne Plugins Mechanismen, die dazu beitragen sollen die Privatsphäre des Nutzers zu schützen. Das Ziel dieser Maßnahmen ist es vor allem eine Wiedererkennung des jeweiligen Nutzers zu verhindern, oder zumindest wesentlich zu erschweren. Zu diesen Maßnahmen zählen unter anderem Löschfunktionen für Cookies und den Browsercache. Ob die Löschfunktionen ausreichend gut funktionieren und welche weiteren Mechanismen die Browser besitzen, sollen Tests (siehe Kapitel 6.2.3) im Rahmen dieser Arbeit zeigen.

4.1 Schwachstellen im Identifikationsverfahren

Das Identifikationsverfahren beruht im Wesentlichen aus den folgenden vier Schritten:

1. dem Auslesen der Daten,
2. der Zuordnung der Daten zu einer Session,
3. der Zuordnung der Sessiondaten zu einem abstrakten Profil des Nutzers und
4. der Zuordnung eines Profils zur realen Person.

An jedem dieser Punkte kann angesetzt werden, um den Prozess der Nutzeridentifikation zu erschweren oder sogar zu verhindern. Es ist jedoch nicht an jeder Stelle günstig das Verfahren zu stören. Das liegt vor allem daran, dass ein Ansatz an der falschen Stelle dazu führen kann, dass eine große Menge von Webseiten nicht mehr ordnungsgemäß funktioniert.

Das Auslesen der Daten kann zum Beispiel nicht vollständig verhindert werden, solange überhaupt eine Art von Kommunikation zwischen dem Nutzer und dem Webserver stattfinden soll. Es ist jedoch möglich die auslesbaren Daten zu manipulieren. Dies wirkt sich vor allem auf den dritten Schritt des Nutzeridentifikationsverfahrens aus.

Die Zuordnung der HTTP-Requests zu einer Session ist ein Schritt, der zwar verhindert werden kann, aber der nicht verhindert werden sollte. Ein großer Teil aller Webseiten baut darauf auf, dass die Zuordnung eines Request zu einer Session funktioniert. Andernfalls stehen viele Funktionen einer Webseite nicht mehr zur Verfügung. Daher sollten weder Sessionvariablen noch Session Cookies (HTTP) durch Abwehrmaßnahmen beeinflusst werden.

Der dritte Schritt der Nutzeridentifikation, die Zuordnung der Sessiondaten zu einem bestehenden Profil kann verhindert werden, wenn die zur Profilzuordnung wesentlichen Merkmale verändert oder entfernt werden. Eine Profilzuordnung kann entweder anhand eindeutiger Methoden oder ähnlichkeitsbasierender Methoden durchgeführt werden. An dieser Stelle ist es möglich verschiedene Abwehrmaßnahmen einzusetzen.

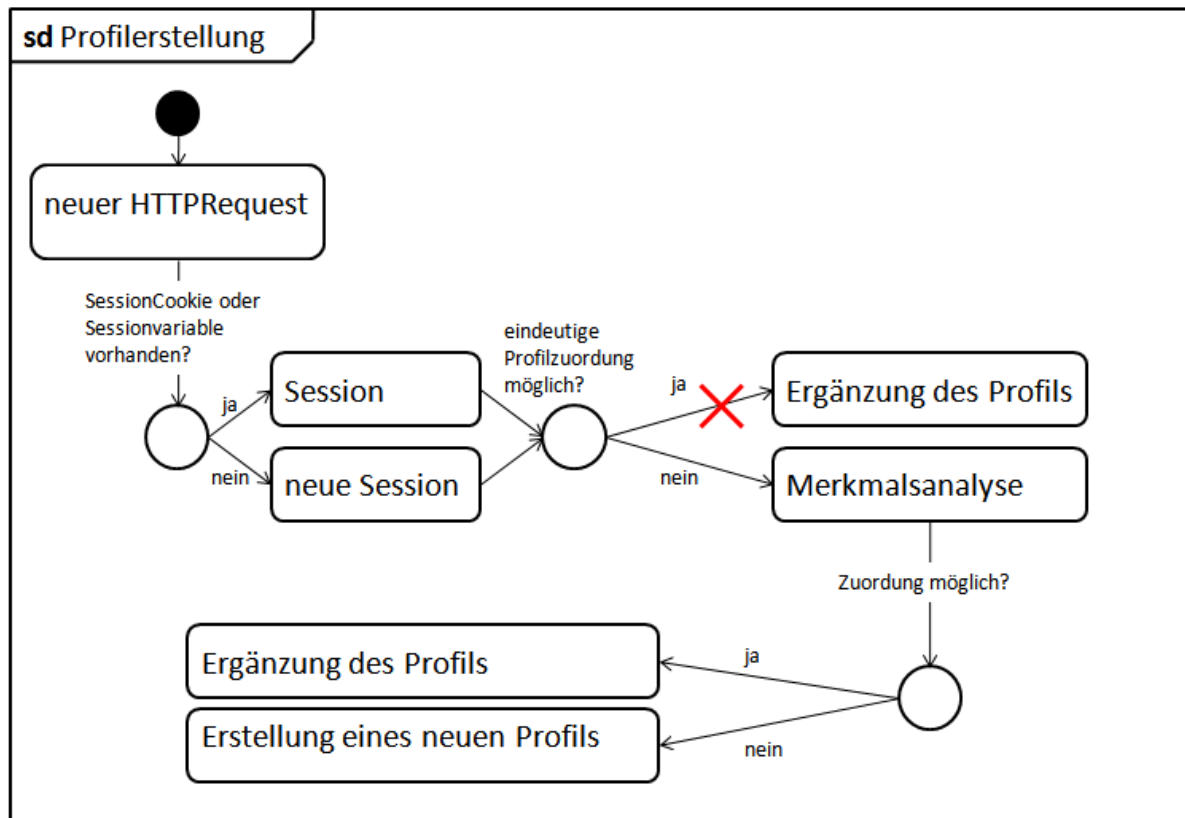


Abbildung 9 - Ansatz eindeutiges Verfahren

Eindeutige Verfahren eignen sich am besten zur Profilzuordnung, da diese ohne größeren Aufwand eingesetzt werden können. Dies ist auch ein Ansatz (Abbildung 9) für Abwehrmaßnahmen, die eine eindeutige Zuordnung verhindern. Aufgrund der Tatsache, dass bei Webdiensten vor allem eindeutige Merkmale wie Cookies zur Profilzuordnung genutzt werden, sollen die Löschmöglichkeiten für Cookies im Folgenden besonders betrachtet werden.

Anonymisierungsdienste können ebenfalls dazu beitragen, die Zuordnung des neuen Profils durch verschiedene Verfahren effektiv zu verhindern. Der Randomize-Ansatz zielt darauf ab, dass nach der Analyse aller Merkmale keine Übereinstimmung mit einem bekannten Profil stattfindet und eine Ergänzung des richtigen Profils (Abbildung 10) verhindert wird. Dies führt zu einem „false negativ“, also zu einer falschen Erstellung eines neuen Profils.

Der Mimikri-Ansatz bewirkt, dass viele Nutzer einem einzigen Spezialprofil (Abbildung 10) zugeordnet werden, obwohl jeder Nutzer einem eigenständigen Profil zugeordnet werden müsste. Ein Beispiel wäre das JonDoFox-Profil für den Firefox. Beim JonDoFox werden die Browsermerkmale durch andere Merkmale ausgetauscht.

Um die einzelnen Verfahren auch einem normalen Nutzer zur Verfügung zu stellen, werden von einigen Anbietern Anonymisierungspakete bereitgestellt, die einen effektiven Rundumschutz bieten sollen.

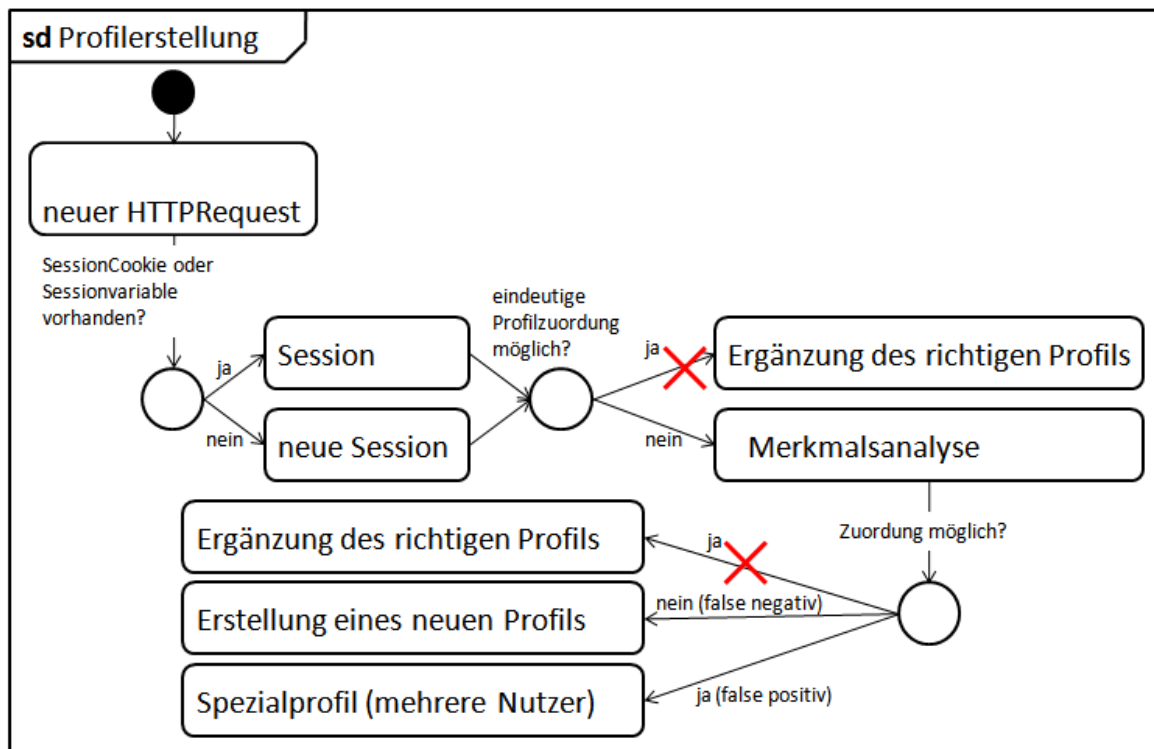


Abbildung 10 - Ansatz ähnlichkeitsbasierendes Verfahren

Der letzte Schritt der Nutzeridentifikation ist die Zusammenführung eines Profils mit dem realen Nutzer. Durch das Einloggen bei einem bestimmten Anbieter ist unter Umständen eine Zuordnung des Profils zur realen Person möglich. Um zu vermeiden, dass Plattformen, die die reale Identität des Besuchers kennen, wie zum Beispiel Amazon, Ebay oder Facebook eine Zusammenführung mit dem Profil eines Dritten durchführen können, ist Identitätsmanagement ein wichtiger Ansatz.

4.2 Cookieabwehr

Cookies stellen eine besonders einfach zu realisierende und eindeutige Form der Profilzuordnung dar. Deshalb ist der Einsatz von Cookies auch die am häufigsten genutzte Technik der Profilzuordnung.

Die Schwachstelle eines Cookies ist, dass dieser auf dem PC des Nutzers gespeichert wird. Da der Nutzer selbst volle Kontrolle über dieses Gerät hat, ist er in der Lage alle Cookies zu löschen. Die zuverlässigste Methode ist die Nutzung einer virtuellen Maschine, die nach dem Surfen im Internet auf einen früheren Speicherpunkt zurückgesetzt wird. Dadurch gehen alle Änderungen die während des Surfens entstanden sind verloren und die Cookies sind entfernt. Der Nachteil ist, dass wirklich alle Änderungen (auch Bookmarks, gespeicherte Seiten, ...) gelöscht sind, wodurch der Komfort für den Nutzer leidet.

Die meisten Browser verfügen jedoch über eine einfache Möglichkeit Cookies zu löschen. Diese Möglichkeit findet man im Menü des jeweiligen Browsers. Die Browser unterstützen neben den Standard HTTP-Cookies auch weitere Cookiearten. Es ist in einigen Fällen allerdings unbekannt, ob ein spezieller Cookie von der Löschfunktion betroffen ist. Ein Test im Rahmen dieser Arbeit wird zeigen, welche Cookies von der Löschfunktion der Browser erfasst werden. (siehe 6.2.3)

4.2.1 Deaktivieren des DOM-Storage

Der Firefox bietet als einziger der getesteten Browser die Möglichkeit Global-Storage, Local-Storage und Session-Storage zu deaktivieren, in dem man in der Adresszeile `about:config` aufruft, und dann den Eintrag `dom.storage.enabled` auf `false` ändert [Moz06]. Durch diese Änderung ist es nicht mehr möglich, den Speicher auszulesen oder zu beschreiben. Diese Einstellungsmöglichkeit ist jedoch schwer erreichbar und somit für normale Anwender ungeeignet.

Die Deaktivierung des DOM-Storage birgt für den Nutzer keine Nachteile. Die im Rahmen der Arbeit getesteten Seiten (siehe Anhang A) nutzen ausschließlich den Session-Storage als zusätzlichen Speicher. Es konnten auch nach dem Deaktivieren des DOM-Storage keine Änderungen in der Funktionalität der Seiten festgestellt werden.

4.2.2 Abwehr von Flash-Cookies und Java-Cache-Cookies

Die sicherste Möglichkeit diese Cookies abzuwehren ist es, Java und Flash in den einzelnen Browsern zu deaktivieren. Dies führt jedoch dazu, dass viele Internetseiten bestimmte Inhalte nicht mehr darstellen können. Die bessere Alternative ist es deshalb, den lokalen Speicher dieser Plugins zu deaktivieren.

Es ist möglich, sowohl die Flash-Cookies als auch den Java-Cache global für alle Browser zu deaktivieren. Durch die globale Deaktivierung wird das Setzen dieser Cookies effektiv verhindert. Im Fall von Flash kann das jedoch zu Problemen führen, da einige Flashapplikationen (vor allem Spiele) auf den Speicher angewiesen sind und ohne die Cookies nicht richtig arbeiten können.

Adobe, der Hersteller von Flash, bietet zum Einsehen, Löschen und Deaktivieren der Cookies eine spezielle Webseite an:

http://www.macromedia.com/support/documentation/de/flashplayer/help/settings_manager.html

Über diese Webseite ist es möglich auch von Smartphones die Flash-Cookies zu entfernen. Die wenigsten Nutzer wissen jedoch von dieser Seite oder machen von dem angebotenen Service regelmäßigen Gebrauch, um das Erstellen von Persönlichkeitsprofilen zu vermeiden. Die Webseite beinhaltet außerdem eine Option, um Webseiten zu verbieten mit Hilfe von Flash lokal Informationen zu speichern. Dazu muss der Speicherplatz, wie in Abbildung 11 zu sehen ist, auf „Keinen“ geändert werden.

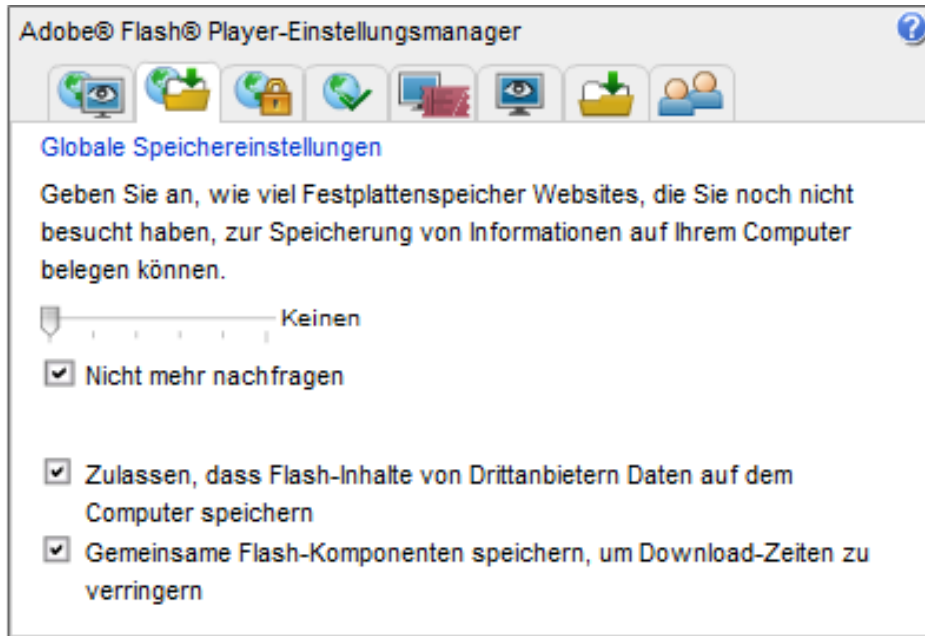


Abbildung 11 - Adobe Einstellungsmanager

Das Ändern der Standardeinstellung auf „Keinen“ hatte, auf den im Rahmen dieser Arbeit überprüften Seiten (siehe Anhang A), keinen Einfluss auf die Darstellung der Inhalte und Videos. Bei Flash-Spielen, kann es jedoch passieren, dass diese nicht ausgeführt werden können, da für das Speichern von Spielständen die Flash-Cookies genutzt werden.

Damit Flash richtig funktioniert, ist es notwendig Drittanbietern das Speichern von Daten zu erlauben, da Videos und andere Flash Inhalte sich häufig auf einem anderen Server als die Webseite selbst befinden. Die Option zum Sperren der Drittanbieter-Daten betrifft auch den temporären Speicher, wodurch die Flash Inhalte von Dritten nicht mehr geladen werden könnten. Die Speicherplatzeinstellungen für die Cookies wird jedoch nicht beeinflusst. Das heißt, dass falls der Regler für den Speicherplatz auf „Keinen“ steht, auch Dritte keine Cookies speichern können, selbst wenn diese berechtigt sind Daten zu speichern.

Flash verfolgt standardmäßig einen Blacklistansatz, bei dem man bestimmten Seiten das Setzen von Cookies verbieten kann. Durch die Begrenzung des Speichers auf „Keinen“ wird das Verhalten des Flash Players umgekehrt und man kann den wenigen Seiten, die wirklich auf die Cookies angewiesen sind, erlauben Cookies zu setzen (Whitelist).

Der Java-Control-Panel bietet ebenfalls eine Möglichkeit, den lokalen Speicher für alle Browser zu deaktivieren. Dazu muss im Java-Control-Panel unter der Registerkarte Allgemein im Bereich „Temporäre Internet-Dateien“ eine Einstellung wie in der folgenden Abbildung 12 geändert werden.

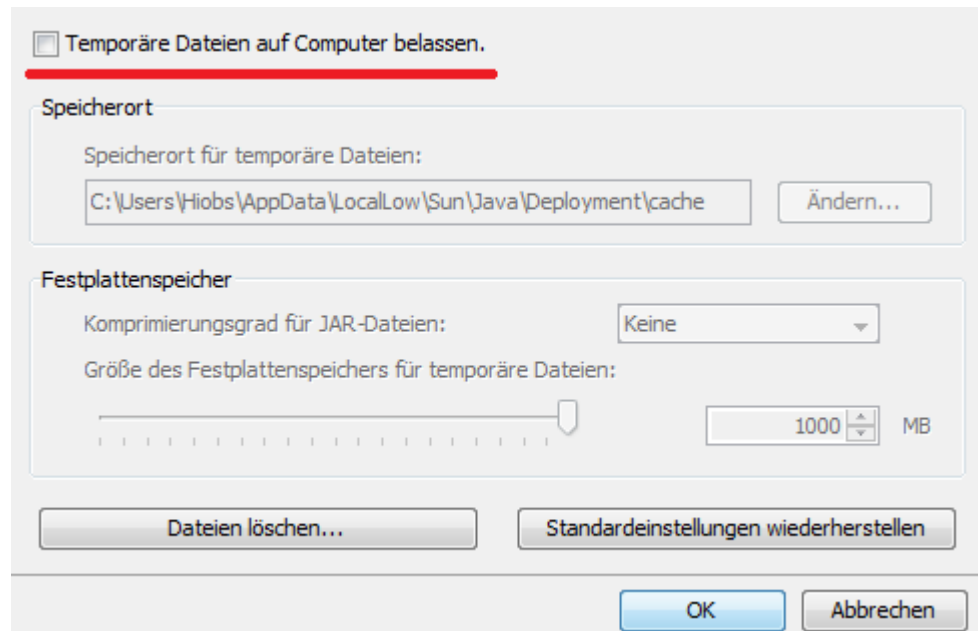


Abbildung 12 - Java-Control-Panel

Durch diese Änderung, werden temporäre Daten beim Beenden des Browser sofort gelöscht. Standardmäßig ist die Option aktiv, um das Laden von Java-Applets zu beschleunigen, wenn diese erneut aufgerufen werden. Auf die Ausführbarkeit von Applets, hat die Deaktivierung jedoch keine Auswirkungen.

Die bereits gespeicherten Dateien können ebenfalls an dieser Stelle gelöscht werden, indem der „Dateien löschen...“ Button betätigt wird.

4.2.3 Cache

Als Cache bezeichnet man spezielle Speicher des Browsers, die das erneute Laden einer Webseite beschleunigen sollen. Diese lassen sich genau wie die Cookies über das Menü löschen. Da diese Speicher auch von einigen Cookies genutzt werden, handelt es sich um eine Löschfunktion, die als Abwehrmaßnahme gegen Cookies eingesetzt werden kann.

Der Cache lässt sich außerdem nutzen, um ein Webseitenprofil auszulesen. Dies kann eine ähnlichkeitsbasierende Profilzusammenführung verbessern und sollte daher vermieden werden, in dem der Cache regelmäßig gelöscht wird.

4.2.4 Verlauf

Der Verlauf des Browsers bietet die Möglichkeit CSS-History-Cookies zu setzen und kann durch eine sogenannte History-Attack ausgelesen werden. Eine Funktion zum Löschen des Verlaufs ist deshalb ein nützliches Hilfsmittel gegen Cookies und das Ausspähen von individuellen Merkmalen. Einige Browser sind inzwischen immun gegen das Auslesen des Verlaufs. Das Löschen des Verlaufs ist deshalb in diesen Browsern nicht notwendig. Dies betrifft die Browser Internet Explorer 9, Firefox 5, Safari 5 und Chrome.

4.2.5 Drittanbietercookies verweigern

Es ist möglich, Drittanbietern das Speichern von Cookies generell zu verbieten. Da bei Dritten Informationen von mehreren Webseiten zusammenfließen, liefern diese Informationen ein viel detaillierteres Bild des Webseitenbesuchers.

Aufgrund der detaillierten Informationen wird unter Umständen auch eine vollständige Nutzeridentifikation einfacher. Aus diesem Grund sollte diese Funktion aktiviert werden, falls sie vorhanden ist. Beim Aktivieren dieser Funktion ist zu beachten, dass sie nicht für alle Cookies gültig ist. Ein Beispiel dafür sind Window.Name Cookies.

4.2.6 Anti-Cookie Plugins

Anti-Cookie Plugins sollen die Verwaltung und das gezielte und regelmäßige Löschen von Cookies vereinfachen. Sie erhöhen somit hauptsächlich den Komfort, da der Nutzer sich nicht selbst um die Löschung verschiedener Cookiearten kümmern muss. Bei einigen Browsern können bestimmte Cookiearten ausschließlich mit Hilfe eines Plugins gelöscht werden, da diese keine Option zum Löschen dieser Cookies besitzen.

Der Nachteil bei der Nutzung von Plugins ist jedoch, dass auch eine Webseite die installierten Plugins ermitteln kann. Dies kann dazu führen, dass der Nutzer leichter einem Profil zugeordnet werden kann. Viele Abwehrplugins werden nicht von einer breiten Masse an Nutzern genutzt und die Nutzung eines solchen Plugins wird somit zu einem besonderen Merkmal. Ein echtes Plugin zur Anonymisierung sollte für den Webseitenbetreiber unsichtbar bleiben und nicht mit Hilfe von JavaScript oder Ähnlichem ermittelt werden können. [Eck11]

4.3 Anonym Surfen

Um zu verstehen, was anonymes Surfen bedeutet, ist es notwendig zu wissen, dass mit jedem Aufruf einer Webseite Daten übertragen werden, durch die man unter Umständen eindeutig einem Profil zugeordnet werden kann.

Neben den Cookies existieren weitere Merkmale, wie zum Beispiel der UserAgent oder der HttpAcceptHeader, die zusammengekommen eine eindeutige Identifikation auch ohne Cookies ermöglichen können. Die Gesamtheit aller erfassbaren Merkmale ergibt ein Identifikationsmuster. Die Information, dass ein bestimmtes Merkmal nicht gesendet wird, ein Zugriff nicht erlaubt ist, oder sich der Nutzer sich für Computerspiele interessiert, ist ebenfalls Teil des Identifikationsmuster. Je mehr Daten erfasst werden, die den Nutzer von anderen Nutzern unterscheiden, desto besser eignet sich ein Identifikationsmuster, um den Nutzer später wiederzuerkennen.

Genau an diesem Punkt setzt die Anonymisierung an. Das bedeutet, dass möglichst alles getan wird, um eine Profilzuordnung und somit die Wiedererkennung zu verhindern. Ein erster Schritt ist das Löschen von Cookies, da Cookies die Technologie sind, die am häufigsten zur Profilzuordnung bei Internetnutzern eingesetzt wird. Die Verwendung von Cookies erlaubt es, einen Nutzer mit einem eindeutigen Muster zu versehen. Andere Techniken arbeiten passiv und versuchen die Profile anhand ähnlichkeitsbasierender Methoden zuzuordnen. Anonymes Surfen bedeutet hauptsächlich, dass die individuellen Merkmale modifiziert werden, um eine Wiedererkennung zu verhindern. Es geht somit über das reine Löschen der Cookies hinaus.

4.3.1 Modifizierung von Merkmalen zur Anonymisierung

Grundsätzlich existieren mehrere Ansätze, die eine Anonymisierung ermöglichen. Zwei dieser Ansätze (Mimikri und Randomize) werden im Folgenden erörtert. Da beide Ansätze sowohl Vor- als auch Nachteile beinhalten.

4.3.1.1 Mimikri

Die Idee des Mimikri-Ansatz ist die Nachahmung eines sehr weit verbreiteten Identifikationsmusters. Dadurch wird eine Zuordnung unmöglich, da sich der einzelne Nutzer nicht mehr von der Masse der Nutzer unterscheidet. Dieser Ansatz birgt jedoch gleich mehrere Probleme in sich. Zunächst muss ein Muster gefunden werden, dass verbreitet genug ist, um eine Identifikation zu verhindern. Die dazu notwendigen statistische Daten schwanken von einer Webseite zur Nächsten und sind zu dem auch noch regionsabhängig. Dies bedeutet nicht nur, dass im UserAgent eine Information über die bevorzugte Sprache enthalten ist, sondern auch, dass in anderen Ländern andere Browser bevorzugt und andere Internetseiten besucht werden. Erschwerend kommt hinzu, dass die Aktualität der Einstellungen ständig überprüft werden müsste. Aktuelle

Trends im Internet können schnell dazu führen, dass sich die am stärksten verbreiteten Identifikationsmuster ändern. Das führt zu einem enormen Aufwand für den Nutzer. Es ist jedoch möglich ein solches System zu automatisieren. Dazu wäre nur eine vertrauenswürdige Quelle mit den notwendigen statistischen Daten nötig. An dieser Stelle wird jedoch wiederum die Regionsabhängigkeit problematisch, da ein Nutzer durchaus Webdienste aus anderen Regionen nutzen könnte. Grundsätzlich gilt, je mehr Nutzer diesem Ansatz folgen und mit dem gleichen Identifikationsmuster im Internet surfen, desto sicherer ist dies für die Privatsphäre des Einzelnen.

4.3.1.2 Randomize

Der Randomize-Ansatz basiert darauf, dass bei jedem Browserstart ein neues zufälliges Identifikationsmuster erzeugt wird. Dies ist sehr leicht zu realisieren und der Nutzer kann nach einem Browserstart nicht mehr vom Webserver zu einem Profil zugeordnet werden. Regionsspezifische Trends müssen bei dieser Methode auch nicht Berücksichtigt werden. Der einzige Nachteil dieser Methode ist, dass ein gezieltes Tracking unter Umständen einfacher wird, da der Nutzer immer ein außergewöhnliches Identifikationsmuster aufweist. Da jedoch alle verbreiteten Trackingsysteme darauf ausgelegt sind breite Nutzermassen zu identifizieren, ist diese Vorgehensweise dennoch erfolgsversprechend. Auch bei diesem Ansatz führen mehr Nutzer zu mehr Anonymität.

4.3.1.3 Praktischer Einsatz

Beide Möglichkeiten lassen sich entweder durch spezielle Plugins oder anonymisierende Proxysysteme, die die Informationen modifizieren praktisch umsetzen. Die Nutzung von Proxyservern, die nicht lokal installiert sind, hat den Vorteil, dass neben den gesendeten Informationen auch die IP-Adresse versteckt werden kann. Dies verhindert, dass der Standort des Nutzers ermittelt wird.

Wie bei Proxyservern und anderen Anonymisierungsdiensten ist es auch im Firefox möglich die meisten Informationen die der Browser an eine Webseite sendet zu verändern. Dies kann jedoch zu Darstellungsfehlern führen, wenn beispielsweise der UserAgent des Firefox auf einen UserAgent des Internet Explorers geändert wird. Es existieren immer noch genügend Webseiten gibt die browserspezifische Seiten ausliefern, da sich die Webseitendarstellung des weit verbreiteten Internet Explorer 6 stark von der standardisierten Norm unterscheidet.

Das speziell angepasste Firefoxprofil JonDoFox ist ein Beispiel dafür, wie weit sich der Firefox durch Mimikri-Techniken modifizieren lässt, um anonym zu surfen. Die anderen Browser bieten nicht die Möglichkeit ihre Informationen so einfach zu verändern, wie es im Firefox möglich ist.

4.3.2 Anonymisierungsdienste

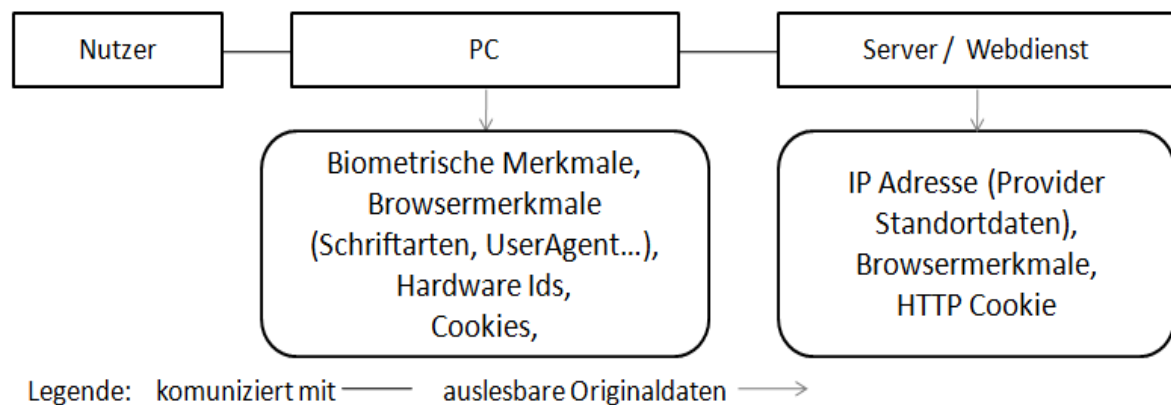


Abbildung 13 - Nicht Anonymisiert

Ausgangspunkt eines Anonymisierungsdienstes ist eine nicht anonymisierte Verbindung zum Server (Abbildung 13). Durch den Anonymisierungsdienst sollten alle Merkmale verändert werden, damit eine Profilzuordnung nicht mehr möglich ist. Dabei ist zu beachten, dass eindeutige Markierungen wie Cookies entfernt werden müssen.

Um eine Anonymisierung durchzuführen, kann ein Proxyserver eingesetzt werden, wodurch die IP-Adresse nicht mehr ermittelbar ist. Außerdem werden Cookies beim Browserstart, oder am Ende einer Session gelöscht. Zusätzlich können die Browsermerkmale verändert werden, um eine Erkennung anhand ähnlichkeitsbasierender Methoden zu verhindern. Durch die Deaktivierung von Plugins kann verhindert werden, dass Schriftarten (siehe 3.5.1) ausgelesen werden.

Einen schwierigen Punkt im mobilen Bereich stellen auslesbare Hardware-IDs dar. Diese können über Apps oder Plugins ausgelesen werden. Ein Anonymisierungsdienst für solche Endgeräte muss den Zugriff auf derartige Hardware-IDs deshalb blockieren.

Ziel ist es die Profilzuordnung wie in Abbildung 14 zu verhindern. Bei Merkmalen, die eine ähnlichkeitsbasierende Zuordnung ermöglichen, ist zu beachten, dass diese verändert aber nicht blockiert werden sollten. Das Blockieren könnte dazu führen, dass das Identifikationsmuster des Nutzers einen hohen Informationsgehalt erhält, da dieses spezielle Identifikationsmuster einzigartig sein könnte.

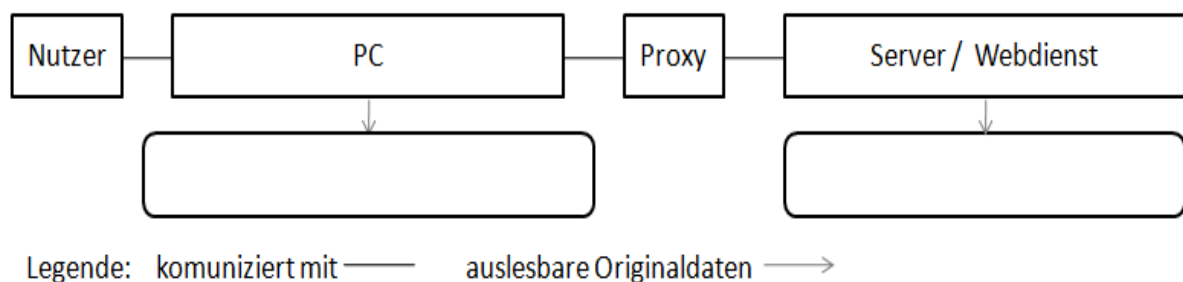


Abbildung 14 Anonymisiert

4.4 Filterprogramme

4.4.1 Werbeblocker

Die bekanntesten Filterprogramme sind Werbeblocker. Ein Werbeblocker verhindert, dass HTTP-Requests an bestimmte URL's oder auch gesamte Server gesendet werden. Werbung wird nach der Filterung ebenfalls nicht mehr angezeigt. Eine Profilzuordnung durch Dritte, kann durch eine solche Maßnahme effektiv verhindert werden. Zu beachten ist aber, dass diese Systeme ausnahmslos mit Blacklists arbeiten. Es ist somit immer möglich, dass ein Werbeanbieter einfach eine neue Domain nutzt, um seine Werbung zu verbreiten und den Nutzer wiederzuerkennen. Der Werbeanbieter hat dabei jedoch das Problem, dass auf die meisten Cookiearten nicht domainübergreifend zugegriffen werden kann.

Bei der Nutzung von Werbeblockern ist zu beachten, dass der Verfasser der Filterlisten neben Werbung auch andere Filter in die Liste schreiben könnte, um eine Art Zensur durchzuführen. Der Nutzer eines Werbeblockers muss also entweder einen vertrauenswürdigen Anbieter finden oder die Liste manuell erweitern. Da manuelle Filterlisten oft kürzer sind, entsteht ein kleiner Geschwindigkeitsvorteil beim Laden der Webseite. Dies ist vor allem auf mobilen Endgeräten spürbar.

4.4.2 Filterproxys

Filterproxys dienen vor allem der Filterung bestimmter Inhalte. Dies sind zum Beispiel Cookies bestimmte URL's, bestimmte Dateitypen oder auch Malware. Ein Proxyserver lässt sich dadurch auch nutzen, um weitestgehend anonym im Internet zu surfen, ohne einen Anonymisierungsdienst zu beanspruchen. Der entscheidende Vorteil ist die höhere Geschwindigkeit, die ein eigener Proxy bietet. Der Nachteil ist, dass diese Vorgehensweise eine Menge KnowHow vom Anwender voraussetzt, da Einrichtung und Konfiguration mit dem Ziel der Anonymisierung nicht einfach sind. [Egg06]

4.4.3 Werbefinanziertes Internet & Filterprogramme

Die Nutzung von Programmen wie Werbeblockern und Filterproxys führt zu einer neuen Problematik, die in Zukunft an Bedeutung gewinnen wird. Eine großer Teil der Webseiten im Internet finanziert sich allein durch die Einblendung von Werbung. Wenn der Anteil der Nutzer, die Filterprogramme nutzen, über ein bestimmtes Maß hinaus steigt, werden sich einige Angebote nicht mehr finanzieren lassen. Die fehlenden Einnahmen stellen ein Problem dar, dass letztendlich auf die Nutzer der Webseiten zurückfällt, wenn die Betreiber ihre Angebote aus Kostengründen einstellen müssen. Es existieren inzwischen konkrete Studien, die sich mit dieser Problematik auseinandersetzen. [Fai10]

4.5 Identitätsmanagement

Es ist unumgänglich, dass man bei einigen Diensten im Internet, wie zum Beispiel einem Onlineshop, seine wahre Identität preisgibt, um diesen zu nutzen und etwas zu bestellen. In dem Augenblick, in dem der Nutzer seine Identität preisgibt, wird der Webseitenbesitzer versuchen diese Identität mit dem aktuellen Profil zu verknüpfen. Der Nutzer hat darauf jedoch keinen Einfluss. Als Nutzer hat man jedoch die Möglichkeit sich vor der Preisgabe der Identität ein neues Profil zuzulegen, indem beispielsweise ein anderer Browser genutzt wird oder ähnliche Maßnahmen ergriffen werden. Entscheidend ist, dass sich das individuelle Identifikationsmuster stark ändert und alle eindeutigen Merkmale (wie Cookies) entfernt werden.

Der Wechsel zu einem anderen Browser bewirkt in diesem Fall, dass die meisten Cookies nicht mehr existieren, dass eventuell andere Plugins installiert sind und dass sich die Browsermerkmale ändern. Zu beachten ist dabei, dass Cookies, die auf einem Plugin basieren, extra gelöscht werden müssen und dass die Schriftarten trotz allem gleich bleiben. Plugins sollten in dem Browser, mit dem der Nutzer seine Identität preisgibt, deshalb deaktiviert werden. Dadurch können auch keine Schriftarten mehr ausgelesen werden.

Identitätsmanagement bezeichnet die Verwaltung verschiedener Onlineidentitäten. Dies ist ein aufwendiger Prozess, der jedoch durch Softwarelösungen vereinfacht werden kann. Eine Software für Identitätsmanagement muss das Ändern der Profile und das Löschen aller eindeutigen Merkmale automatisieren. Ein einziges eindeutiges Merkmal könnte die Zuordnung der verschiedenen Profile eines Nutzers ermöglichen und den Nutzer somit enttarnen. Ein Nutzer kann die Software anhand von Cookie-Testseiten und Anonymitätstest, die die entsprechenden Merkmale anzeigen überprüfen. Dabei ist zu beachten, dass die Tests dem aktuellen Stand der Technik entsprechen müssen, da neue Merkmale sonst nicht geprüft werden können.

5 Entwicklung der Testseite

5.1 Anforderungen

5.1.1 Client

Um die Cookie-Abwehrfunktionen der einzelnen Browser evaluieren zu können, soll eine spezielle Webseite erstellt werden. Die Testseite muss eine Eingabemaske enthalten, über die man verschiedene Zahlenwerte als Cookie speichern kann. Dabei ist zu beachten, dass die Eingabemaske auch bei ausgeschaltetem JavaScript richtig funktioniert. Die eingegebene Zahl soll zum Server gesendet werden, wo dann das Speichern der Cookies vorbereitet und gestartet wird.

Die Eingabemaske soll außerdem einen Button zum Laden der Cookies enthalten. Dieser Button wird den Befehl zum Auslesen der Cookies zum Server senden. Die ausgelesenen Daten sollen in Form einer Tabelle auf der Testseite ausgegeben werden.

Anhand der Anforderungen ergeben sich die in der folgenden Abbildung 15 dargestellten Anwendungsfälle.

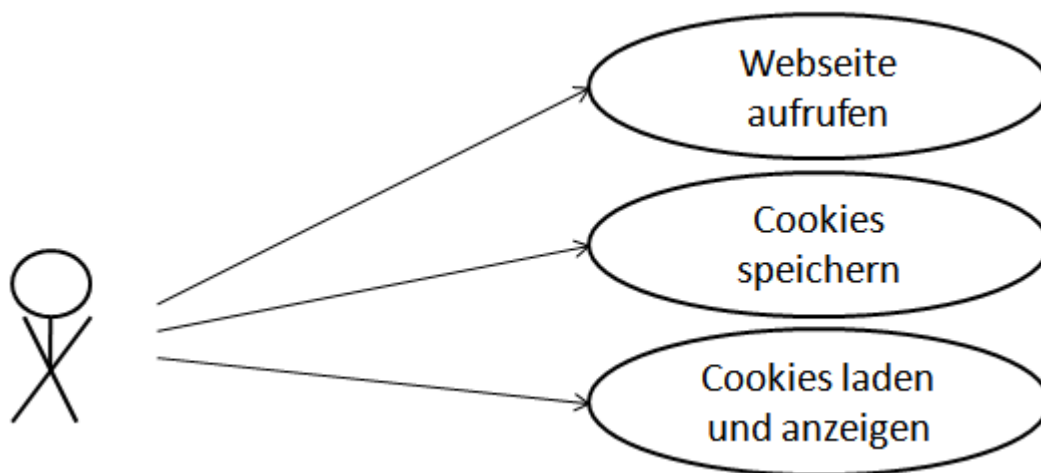


Abbildung 15 - Anwendungsfalldiagramm

Beim Aufruf der Seite wird zunächst nur die Eingabemaske gezeigt. Durch die Eingabe einer Zahl und die Betätigung des Buttons zum Speichern werden die Cookies gespeichert. Die eingegebene Zahl ist sowohl client- als auch serverseitig zu validieren. Die gespeicherten Cookies müssen jederzeit durch den Laden-Button geladen werden können.

Um auch einen Test auf mobilen Endgeräten zu ermöglichen, soll das Design der Testseite möglichst kompakt sein und auf überflüssige Elemente weitestgehend verzichten. Die geringe Auflösung mobiler Geräte ist dabei zu beachten.

5.1.2 Cookies

Die folgenden Cookiearten müssen auf der Testseite untersucht werden können:

- HTTP-Cookie,
- Session-Storage,
- Local-Storage,
- Window.Name,
- IE-UserData,
- Global-Storage,
- Database-Storage,
- WebCache,
- CSS-History-Cookie,
- Flash-Cookie,
- Java-Cache-Cookie und
- Google Gears DB.

Die Cookies sind anhand ihrer Spezifikationen (falls vorhanden) zu implementieren. Bei der Implementierung sollen dass die Anforderungen an den Client so gering wie möglich gehalten werden. Dies betrifft sowohl den Einsatz von clientseitigen Skripten als auch den Einsatz von ergänzenden Cookies zur Absicherung der Session.

Die Funktionalität der einzelnen Cookiearten ist durch geeignete Tests sicherzustellen. Dabei ist zu berücksichtigen, dass nicht jede Cookieart in jedem Browser funktionieren kann.

Der PNG-Caching Cookie wird nicht extra getestet, da dieser Cookie den selben Speicher wie das WebCache Cookie nutzt und durch die Nutzung des Canvas-Elementes nicht mit allen Browsern kompatibel ist. Die Testergebnisse des WebCache Cookie sind auf den PNG-Caching Cookie übertragbar, falls der jeweilige Browser das Canvas-Element unterstützt.

5.2 Sprachen

Die Skriptsprache PHP 5 soll serverseitig für die Verwaltung der Cookies eingesetzt werden. Als weitere Sprachen kommen JavaScript für verschiedene Cookiearten sowie Java und haXe (für Flash) für die auf den jeweiligen Plugins basierenden Cookies zum Einsatz.

5.3 Entwicklungsumgebungen

Als Entwicklungsumgebungen werden Eclipse für die Entwicklung des Java-Cache-Cookies, FlashDevelop für die Entwicklung des FlashCookies und Notepad++ für die Entwicklung der PHP-Skripte und der JavaScript-Cookies eingesetzt.

5.4 Architektur

Die Struktur der Testseite orientiert sich an einer klassischen Client-Server-Architektur. Die Übertragung der Daten zwischen Client und Server erfolgt mittels HTTP. Die Cookiedaten, die mit Hilfe der verschiedenen Techniken gespeichert und gelesen werden sollen, werden immer in Form eines Strings an die jeweilige Funktion übergeben. Dies ist nötig, da für die Komponenten des Systems 4 verschiedene Programmiersprachen genutzt werden und Komplikationen bei der Übertragung der Werte vermieden werden sollen.

5.4.1 Server

Der Server besteht wie in Abbildung 16 zu sehen ist aus 4 Komponenten. Die Ausgabe der Testseite wurde in der Klasse HtmlBuilder untergebracht, um die Ausgabe von den anderen Funktionen zu trennen. Der für die Plugins notwendige Code zum browserabhängigen einbetten von Java und Flash ist in den Funktionen addJava() sowie addFlash() enthalten.

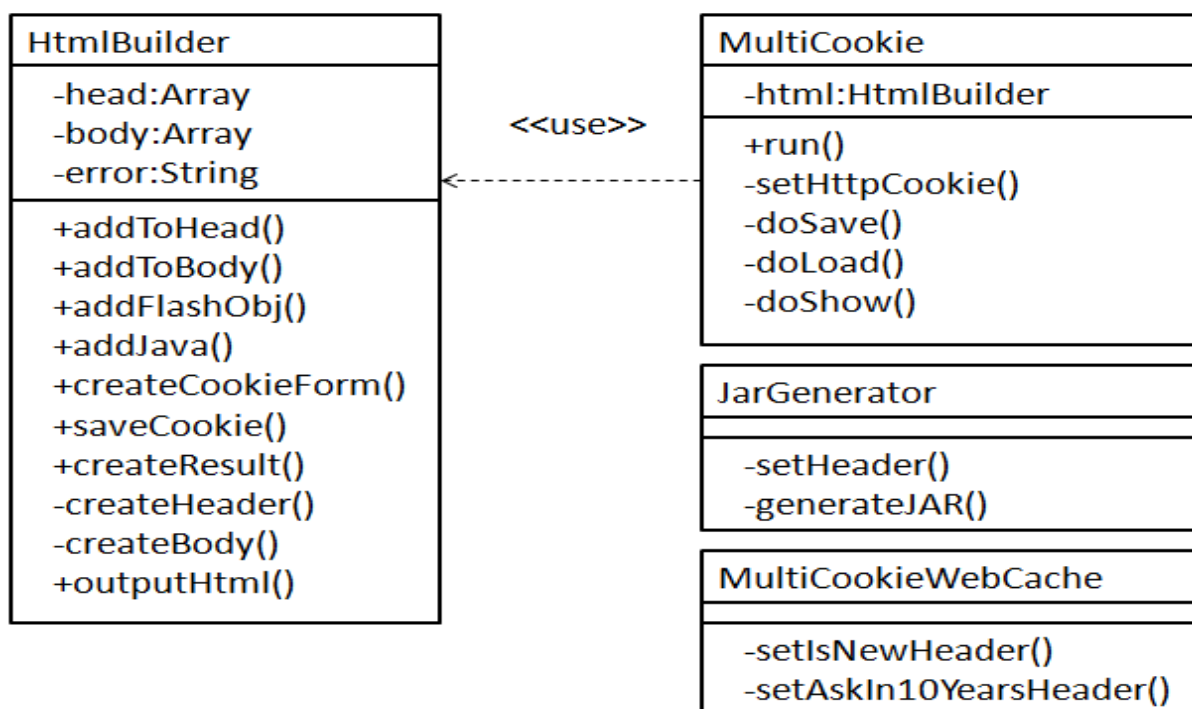


Abbildung 16 - Klassendiagramm Server

Die Klasse MultiCookie dient der zentralen Verwaltung. Sie ist vor allem für das Laden und Speichern der Cookies verantwortlich und steuert die einzelnen Funktionen zum Speichern der verschiedenen Cookiearten.

Das MultiCookieWebCache-Skript stellt Funktionen für den WebCache Cookie bereit und das JarGenerator-Skript dient zur Generierung eines JAR-Archivs. Das JAR-Archiv ist ein modifiziertes Archiv, das die Cookiedaten des Java-Cache-Cookie enthält.

5.4.2 Client

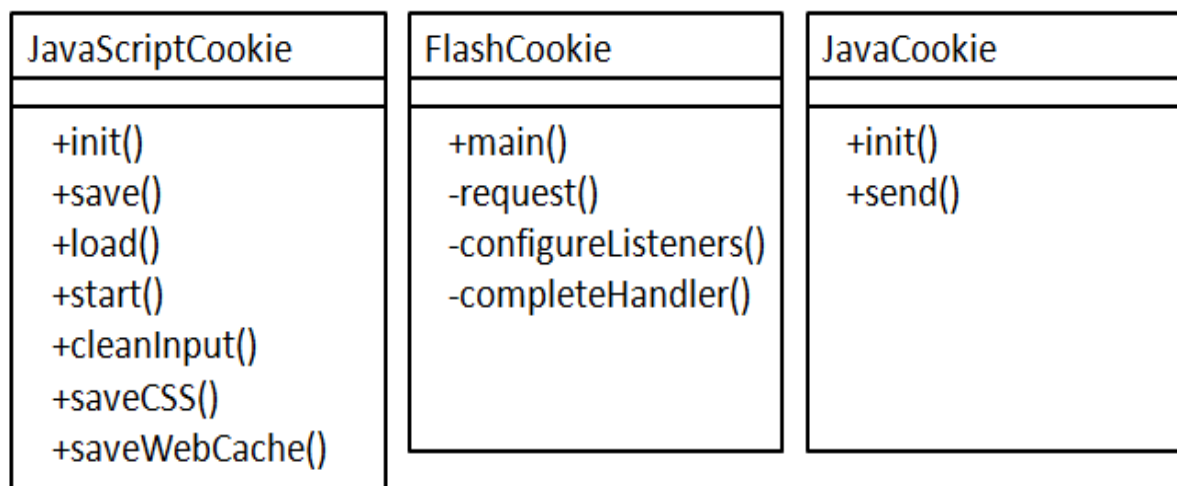


Abbildung 17 - Klassendiagramm Client

Die in Abbildung 17 dargestellten Klassen stellen die Funktionen zum Speichern und Laden verschiedener Cookiearten zur Verfügung. Die Klasse JavaScriptCookie dient zum Laden und Speichern der auf JavaScript basierenden Cookiearten.

Die Klassen FlashCookie und JavaCookie sind weitestgehend unabhängig und werden durch ihre Initialisierungsparameter gesteuert. Beide Klassen senden automatisch Cookiedaten zum Server an die Klasse MultiCookie falls ein Cookie gefunden wurde. Eine besondere Funktion des Java-Cache-Cookie ist, dass dieser automatisch wiederhergestellt wird, wenn der dazugehörige HTTP-Cookie noch vorhanden ist.

Da sowohl Java als auch der FlashCookie ihre Daten zunächst an den Server senden, kann es notwendig werden, dass die Seite zum Anzeigen der Cookies nach dem ersten Ladevorgang aktualisiert werden muss, um diese Cookies auch anzuzeigen. Die Initialisierung der beiden Plugins benötigt Zeit, deshalb sollte zwischen dem ersten und dem zweiten Ladevorgang eine kurze Pause sein.

5.5 Funktionsablauf

5.5.1 Cookies speichern

Das Speichern von Cookies geschieht nach folgendem Schema (Abbildung 18). Nach der Eingabe einer beliebigen Zahl und dem Click auf den „Save MultiCookie“-Button, wird zunächst die Eingabe an den Server gesendet, um dort validiert zu werden. Falls die eingegebenen Zeichen die Validierung bestehen, beginnt der Speichervorgang. Als erstes wird ein HTTP-Cookie gesetzt, da dieser Teil des Übertragungsprotokoll ist und an dieser Stelle gesetzt werden sollte. Der folgende Schritt ist die Erstellung eines neuen HtmlBuilder-Objektes, um den HTML-Code der Testseite zu erzeugen. Mit dem Aufruf der Funktion `outputHtml()` wird die Ausgabe generiert und an den Client gesendet. Dort beginnt automatisch die Speicherung des Flash-Cookies, des Java-Cache-Cookies und der JavaScript-Cookies. Der Aufruf der entsprechenden Funktionen zum Speichern erfolgt in diesem Fall indirekt. Der Befehl zum Speichern der JavaScript Cookies wird in die Testseite eingebettet. FlashCookie und JavaCookie starten automatisch die jeweiligen Funktionen sowie diese geladen sind.

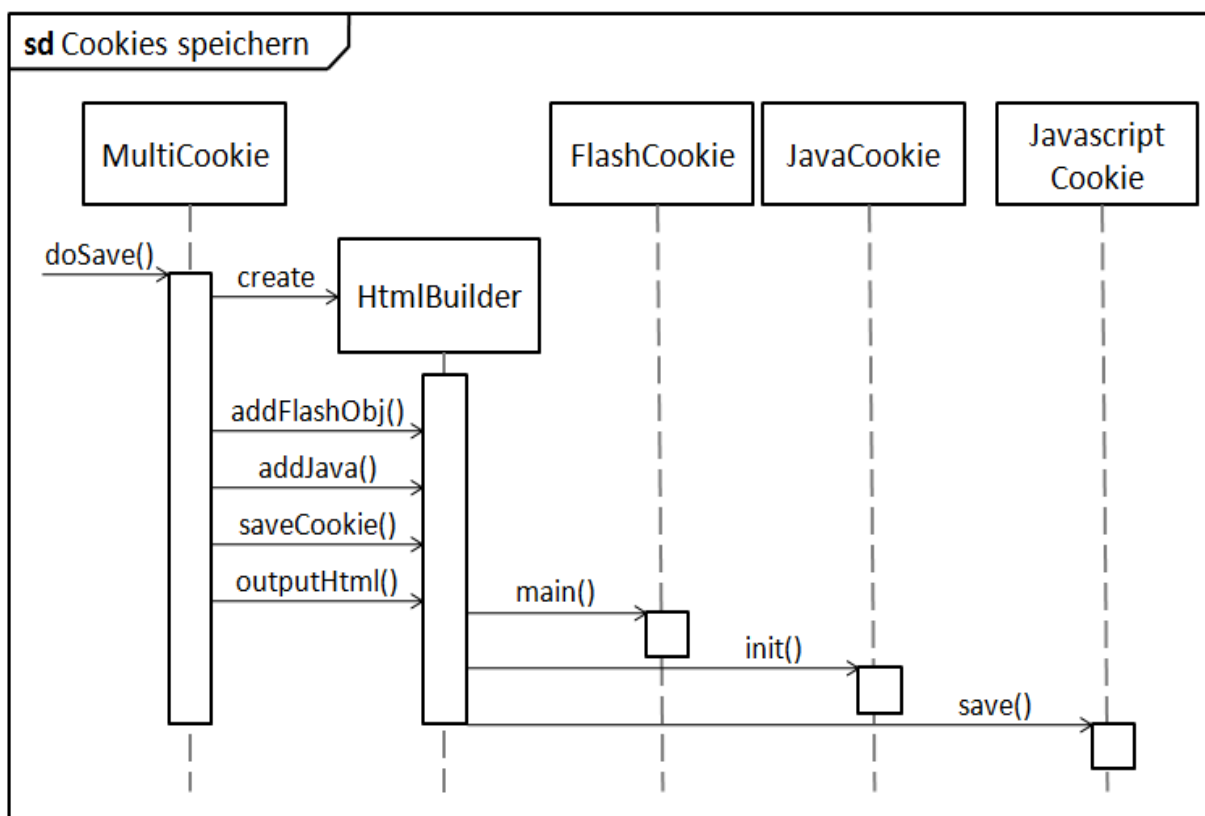


Abbildung 18 - Sequenzdiagramm Cookies speichern

5.5.2 Cookies laden

Die folgende Abbildung 19 zeigt den Ladevorgang eines Flash-Cookies. Der erste Schritt ist das Laden der Testseite und die Ausführung des Plugins. Das Plugin liest automatisch den Cookie aus und sendet, falls ein Cookie gefunden wurde, diesen zum Server. Dabei wird die Klasse MultiCookie aufgerufen, welche die im Cookie enthaltene Nutzer-ID auf dem Server speichert. Bei einem erneuten Aufruf der Testseite mit dem „Load MultiCookie“ Button wird die Nutzer-ID mit ausgegeben. Die Nutzer-ID wird dabei direkt in die Ausgabe der Testseite integriert.

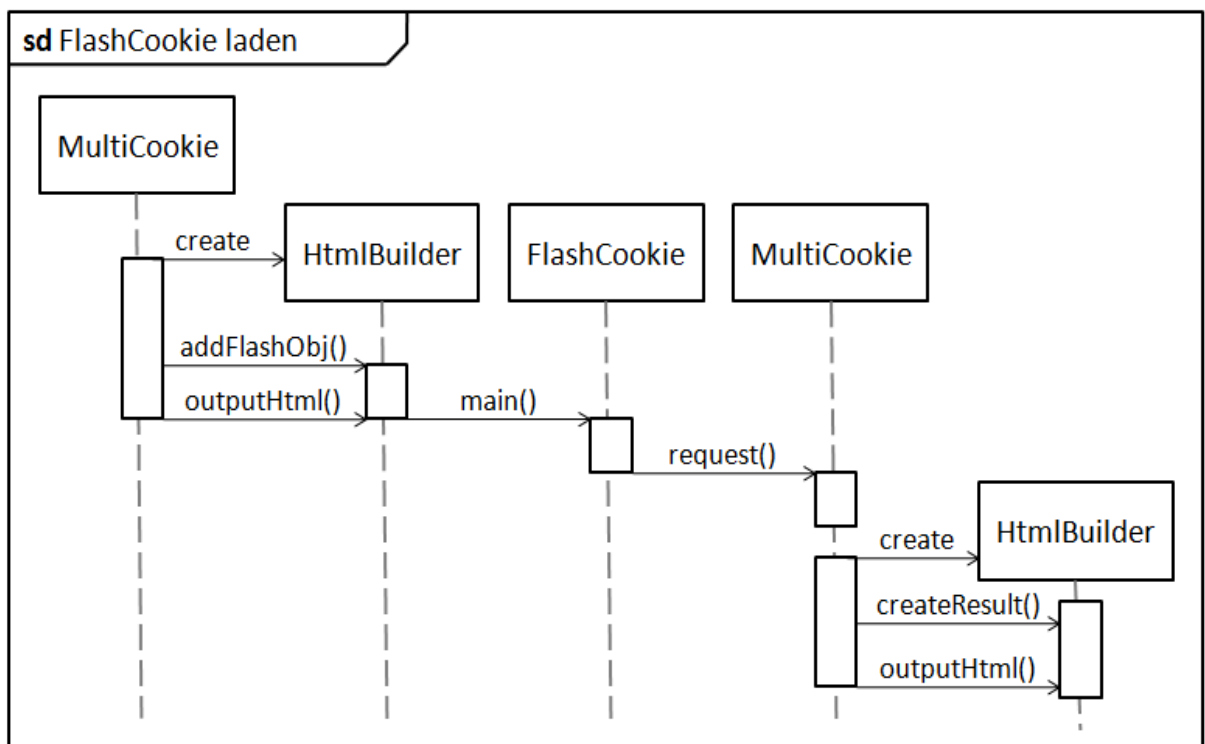


Abbildung 19 - Sequenzdiagramm Flash-Cookie laden

Der Ladevorgang des Java-Cache-Cookies ist mit dem Ladevorgang des Flash-Cookie weitestgehend identisch. Die JavaScript-Cookies unterscheiden sich von diesen beiden Cookies, da sie sofort auf der Testseite angezeigt werden können, ohne dass ein Zwischenschritt über den Server notwendig ist (Abbildung 20).

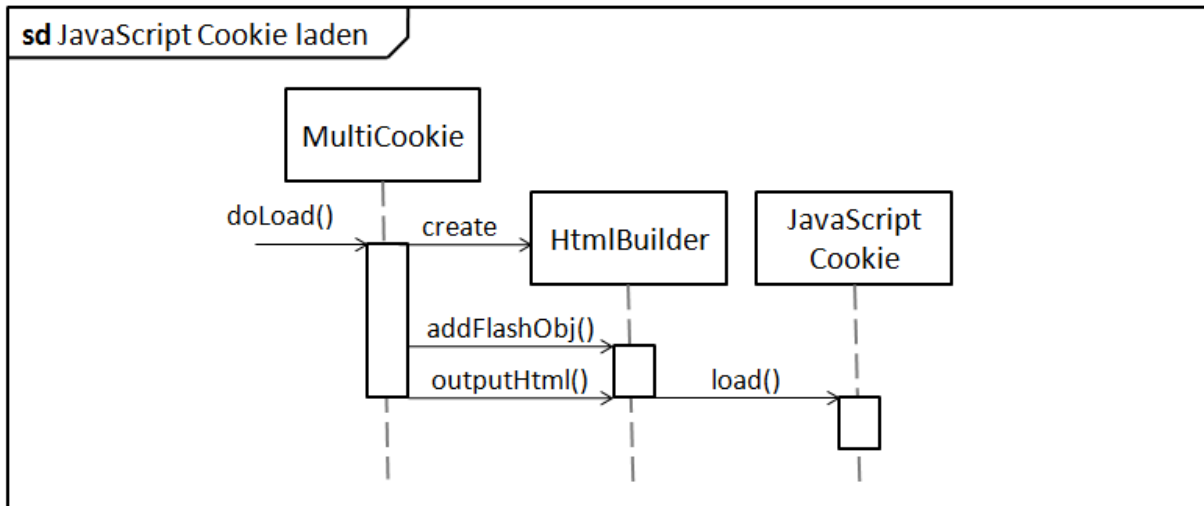


Abbildung 20 - Sequenzdiagramm JavaScript-Cookie laden

Das Senden der ausgelesenen Daten der einzelnen auf JavaScript basierenden Cookies zum Server ist für die vollständige Funktionalität der Testseite nicht erforderlich. Die Testergebnisse müssten für einen vollständigen Cookie nach dem Auslesen mit einem XMLHttpRequest an den Server gesendet und dort validiert werden. Dieser Schritt ist jedoch für die geplanten Tests nicht notwendig, da die Daten nur angezeigt werden sollen.

5.6 Systemvoraussetzungen

- Apache Webserver (mit dem Modul mod_rewrite)
- PHP 5

5.7 Installation

Um die Testseite zu installieren, muss das Archiv mit der Testseite in ein freigegebenes Verzeichnis auf dem Webserver extrahiert werden. Eine automatische Weiterleitung sorgt dafür, dass der Nutzer vom Pfad dieses Verzeichnisses zur MultiCookie Klasse geleitet wird.

Bei der Installation ist zu beachten, dass die `.htaccess` Datei im Ordner `obj` angepasst werden muss. Diese leitet mit Hilfe des Apache Module `mod_rewrite` Anfragen zu der Datei `Test.jar` nach `JarGenerator.php` um. Ohne diese Umleitung kann der Java-Cache-Cookie nicht funktionieren.

6 Evaluation von Abwehrstrategien

6.1 Vorgehensweise

Aufgrund vorangegangener Studien [SCM09] ist bereits bekannt, dass nicht alle Techniken zur Wiedererkennung eines Nutzers erfolgreich durch die Standardinstallation eines Browsers verhindert werden können. Anhand der folgenden Tests soll überprüft werden, ob es möglich ist, Cookies allein mit den Funktionen der Browser zu löschen.

Da moderne Browser neben zahlreichen Funktionen zum Löschen von Daten auch einen privaten Modus besitzen, soll dieser gesondert untersucht werden. Der private Modus verspricht, dass alle Daten, die während einer Sitzung gespeichert werden, beim Beenden des Browser gelöscht werden, um eine Identifikation des Nutzers beispielsweise durch Cookies zu verhindern. Da sich Personen, die um ihre Privatsphäre besorgt sind, besonders auf diesen Modus verlassen werden, soll überprüft werden, ob eine Wiedererkennung des Browsers trotz der Nutzung des privaten Modus möglich ist.

Es soll außerdem überprüft werden, ob es möglich ist die Speicher, die nicht durch den Browser gelöscht werden können, auf andere Art und Weise zu löschen (manuell, oder durch AntiCookie-Plugins).

Da das anonyme Surfen beinhaltet, dass auch ähnlichkeitsbasierende Verfahren den Nutzer nicht dem richtigen Profil zuordnen können, werden 2 Anonymisierungspakete untersucht werden.

6.2 Cookieabwehr

6.2.1 Ziel

Das Ziel der folgenden Tests ist die Prüfung der Abwehrmechanismen gegen Cookies. Dabei soll unter anderem getestet werden, ob die Löschfunktionen zum Löschen der Cookies allein genügen, oder ob es sinnvoll ist, weitere Löschfunktionen zu nutzen, um die Cookies zu entfernen. Zusätzlich soll der Private Modus der Browser untersucht werden, da dieser anonymes Surfen im Internet verspricht. Im Idealfall sollten alle Cookies durch die Nutzung der Cookie-Löschfunktion oder durch die Nutzung des Privaten Modus gelöscht werden.

6.2.2 Testumgebung

6.2.2.1 Server

Serverseitig wird die im Rahmen dieser Arbeit erstellte Testseite (Kapitel 5) eingesetzt, um die Abwehrfunktionen gegen verschiedene Cookiearten zu untersuchen.

6.2.2.2 Betriebssystem

Die Tests werden mit dem Betriebssystem Windows 7 (Servicepack 1) durchgeführt, da die am stärksten verbreiteten Browser alle über eine Windows Version verfügen. Ein Test auf einem anderen Betriebssystem könnte zu abweichenden Ergebnissen führen, da sich die Browser in ihrer betriebssystemspezifischen Implementierung leicht unterscheiden.

6.2.2.3 Browser

Die Browser wurden Anhand ihrer Verbreitung bei den Nutzern ausgewählt. Die Auswahl der Browser umfasst den Internet Explorer 8.0 und 9.0, Firefox 3.6 und 5.0, Chrome 9.0 und 12.0 Safari 5.0.2 und 5.0.5 und Opera 11.01 und 11.11. Da im Browserbereich ständig Updates nachgerüstet werden, um Sicherheitslücken zu schließen, kann diese Untersuchung nur ein Blick auf den aktuellen Stand der Technik sein. Es ist durchaus möglich, dass einige der Lücken im Bereich Cookies bereits in der jeweils nächsten Version behoben sind. Aus diesem Grund sind die Testergebnisse nur mit den Browsern in den bereits genannten Versionen sicher reproduzierbar.

6.2.2.4 Plugins

Da Flash von mehr als 95% aller Internetnutzer genutzt wird, soll das Flash-Plugin in den folgenden Tests berücksichtigt werden. Zusätzlich zu diesem Plugins wird auch JAVA eine Rolle spielen, da im Rahmen dieser Arbeit bereits festgestellt wurde, dass es möglich ist Cookies für den Java-Cache zu entwickeln. Das Java-Plugin ist in 78% aller Browser aktiv. [Sta11] Durch diese starke Verbreitung ist ein Java-Cache-Cookie eine ernst zu nehmende Gefahr für die Privatsphäre.

Google Gears ist für alle Browser der folgenden Tests als Plugin erhältlich. Es wird in den Tests jedoch nur beim Chrome Browser eine Rolle spielen, da das Plugin dort fester Bestandteil des Browsers ist. Der Google-Gears-DB-Cookie wird deshalb nur im Chrome Browser funktionieren.

6.2.2.5 Sonstige Tools

Als zusätzliches Tool kommt der Netzwerksniffer Wireshark zum Einsatz, um zu überprüfen, ob die Ergebnisse der Testseite mit den tatsächlich gesendeten Daten übereinstimmen. Durch den Einsatz von Wireshark kann auch untersucht werden, ob sonstige Daten an den Webservice gesendet werden, die eine Identifikation des Nutzers ermöglichen.

6.2.3 Durchführung der Tests

Die folgenden Tests sollen zeigen, in wie weit es möglich ist, Cookies zu löschen oder abzuwehren. Ein gefundener Cookie wird in den Tabellen mit einem „c“ gekennzeichnet, während ein „-“ signalisiert, dass kein Cookie gefunden wurde.

Die erste Testreihe wurde am 23. und 24. März 2011 durchgeführt die entsprechenden Browserversionen und Testergebnisse stehen in Klammern. Der zweite Test wurde am 26. und 27. Juni 2011 durchgeführt.

6.2.3.1 Browser mit Plugins

Die folgende Tabelle 4 zeigt, welche Cookies nach dem einfachen Löschen der Cookies, über die jeweilige Browserfunktion noch gefunden werden. Dieser Test bezieht sich ausschließlich auf die Option „Cookies löschen“. Die Optionen zum Löschen des Caches und des Browserverlauf werden nicht genutzt.

Tabelle 4 - Cookies löschen

Cookie	IE 9.0.8112 (8.0.6)	Firefox 5.0 (3.6.8)	Chrome 12.0 (9.05)	Safari 5.0.5 (5.0.2)	Opera 11.11 (11.01)
HTTP-Cookie	c (c)	- (-)	- (-)	- (-)	c (-)
Session-Storage	c (c)	c (c*)	c (c)	c (c)	c (c)
Local-Storage	c (c)	c* (c*)	- (-)	c (c)	c (c)
Window.Name	c (c)	c (c)	c (c)	c (c)	c (c)
IE-UserData	- (-)	- (-)	- (-)	- (-)	- (-)
Global-Storage	- (-)	c* (c*)	- (-)	- (-)	- (-)
Database-Storage	- (-)	- (-)	- (-)	c (c)	- (-)
WebCache	c (c)	c (c)	- (-)	- (-)	c (c)
CSS-History-Cookie	- (c)	- (c)	- (-)	- (-)	c (c)
Flash-Cookie	- (c)	- (c)	- (c)	c (c)	c (c)
Java-Cache-Cookie	c (c)	c (c)	c (c)	c (c)	c (c)
Google Gears DB	- (-)	- (-)	- (c)	- (-)	(-)

*Diese Cookies werden nur gelöscht, wenn als Option für den Zeitraum „Alles“ angegeben wird. Bei allen anderen Optionen bleiben die Cookies erhalten.

Durch diesen Test soll gezeigt werden, welche Cookies nach einem normalen Löschvorgang durch einen Nutzer noch vorhanden wären. Dieser Test ist besonders deshalb wichtig, da Nutzer die Löschoption ohne Neustart nutzen könnten. Falls in

diesem Fall eine Seite aufgerufen wird, die Cookies gesetzt hat, kann die Seite den Nutzer wiedererkennen und anhand der noch vorhandenen Cookies alle nicht mehr vorhandenen Cookies neu setzen.

Bei keinem der getesteten Browser waren nach dem Aufruf der Löschfunktion alle Cookies gelöscht. Im Internet Explorer und im Opera sind nach dem Löschen der Cookies sogar die HTTP-Cookies noch vorhanden. Dieses Ergebnis ist für einen normalen Nutzer nicht nachvollziehbar und entspricht nicht dem erwarteten Verhalten. Nach dem Aufruf der Löschfunktionen sollten die Cookies auch verschwinden, unabhängig von einem Neustart. Es existiert außerdem kein Hinweis darauf, dass ein Neustart erforderlich ist.

Die nachfolgende Tabelle 5 zeigt, welche Cookies nach dem Löschen und einem anschließendem Neustart des Browsers noch vorhanden sind.

Tabelle 5 - Cookies löschen und Browserneustart

Cookie	IE 9.0.8112 (8.0.6)	Firefox 5.0 (3.6.8)	Chrome 12.0 (9.05)	Safari 5.0.5 (5.0.2)	Opera 11.11 (11.01)
HTTP-Cookie	- (-)	- (-)	- (-)	- (-)	- (-)
Session-Storage	- (-)	- (-)	- (-)	- (-)	- (-)
Local-Storage	- (-)	c *(c*)	- (-)	c (c)	c (c)
Window.Name	- (-)	- (-)	- (-)	- (-)	- (-)
IE-UserData	- (-)	- (-)	- (-)	- (-)	- (-)
Global-Storage	- (-)	c* (c*)	- (-)	- (-)	- (-)
Database-Storage	- (-)	- (-)	- (-)	c (c)	- (-)
WebCache	c (c)	c (c)	c (-)	- (-)	c (-)
CSS-History-Cookie	- (c)	- (c)	- (-)	- (-)	- (c)
Flash-Cookie	- (c)	- (c)	- (c)	c (c)	c (c)
Java-Cache-Cookie	c (c)	c (c)	c (c)	c (c)	c (c)
Google Gears DB	- (-)	- (-)	- (c)	- (-)	- (-)

*Diese Cookies werden nur gelöscht, wenn als Option für den Zeitraum „Alles“ angegeben wird. Bei allen anderen Optionen bleiben die Cookies erhalten.

Nach dem Neustart der Browser ist deutlich zu sehen, dass ein großer Teil der Cookies gelöscht wurde. Es ist jedoch auch erkennbar, dass nicht alle Cookies von der Löschfunktion erfasst wurden, sodass der Nutzer weiterhin einem Profil zugeordnet werden kann.

Der nächste Test soll zeigen, ob es möglich ist auch die anderen Cookies mit Hilfe der Löschfunktionen des Browsers zu entfernen. Zu diesem Zweck werden zusätzlich zu den normalen Cookies noch der Verlauf gelöscht und der Cache geleert. Da der vorangegangene Test bereits gezeigt hat, dass die Cookies erst nach einem Neustart gelöscht sind, wird auch beim folgenden Test ein Neustart nach dem Aufruf der Löschfunktionen durchgeführt.

Tabelle 6 - Cookies, Verlauf, Cache löschen und Neustart

Cookie	IE 9.0.8112 (8.0.6)	Firefox 5.0 (3.6.8)	Chrome 12.0 (9.05)	Safari 5.0.5 (5.0.2)	Opera 11.11 (11.01)
HTTP-Cookie	- (-)	- (-)	- (-)	- (-)	- (-)
Session-Storage	- (-)	- (-)	- (-)	- (-)	- (-)
Local-Storage	- (-)	c* (c*)	- (-)	c (c)	- (-)
Window.Name	- (-)	- (-)	- (-)	- (-)	- (-)
IE-UserData	- (-)	- (-)	- (-)	- (-)	- (-)
Global-Storage	- (-)	c* (c*)	- (-)	- (-)	- (-)
Database-Storage	- (-)	- (-)	- (-)	c (c)	- (-)
WebCache	- (-)	- (-)	- (-)	- (-)	- (-)
CSS-History-Cookie	- (-)	- (-)	- (-)	- (-)	- (-)
Flash-Cookie	- (c)	- (c)	- (c)	c (c)	c (c)
Java-Cache-Cookie	c (c)	c (c)	c (c)	c (c)	c (c)
Google Gears DB	- (-)	- (-)	- (c)	- (-)	- (-)

*Diese Cookies werden nur gelöscht, wenn als Option für den Zeitraum „Alles“ angegeben wird. Bei allen anderen Optionen bleiben die Cookies erhalten.

Dass die Löschfunktion des Firefox nur mit der Option „Alles“ zuverlässig funktioniert, ist ein bereits bekannter Fehler, der in kommenden Versionen des Browsers sicher behoben wird. Es ist außerdem ersichtlich, dass der Safari Browser den Local-Storage und den Database-Storage nicht löscht.

Das Ergebnis dieses Dritten Tests zeigt außerdem, dass vor allem Plugins die Privatsphäreneinstellungen der Browser untergraben und dadurch zu Lücken in der Abwehr führen. Der Flash-Cookie kann in den neueren Browserversionen gelöscht werden. Dies ändert jedoch nichts daran, dass Plugins von den Privatsphäreneinstellungen der Browser unberührt bleiben.

6.2.3.2 Browser mit Plugins ohne JavaScript

Da JavaScript in einigen Unternehmen und bei verschiedenen Nutzern aus Sicherheitsgründen deaktiviert ist, soll dieser Test zeigen, ob sich das Deaktivieren von JavaScript positiv auf die Abwehr von Cookies auswirken kann.

Der nachfolgende Test (Tabelle 7) zeigt die Cookies, die ohne JavaScript und nach einem einfachen Aufruf der Cookielöschfunktion mit anschließendem Neustart noch gefunden werden konnten. Cache und Verlauf wurden bei diesem Test nicht gelöscht.

Ein großer Teil der Cookies konnte allein durch das Deaktivieren von JavaScript nicht mehr gefunden werden. In den Browsern Chrome und Opera führt das Deaktivieren von JavaScript auch zu einer Deaktivierung von Java, wodurch der Java-Cache nicht mehr beschrieben und ausgelesen werden kann.

Tabelle 7 - Cookies, Verlauf, Cache löschen und Neustart

Cookie	IE 9.0.8112 (8.0.6)	Firefox 5.0 (3.6.8)	Chrome 12.0 (9.05)	Safari 5.0.5 (5.0.2)	Opera 11.11 (11.01)
HTTP-Cookie	- (-)	- (-)	- (-)	c* (c*)	- (-)
Session-Storage	- (-)	- (-)	- (-)	-* (-*)	- (-)
Local-Storage	- (-)	- (-)	- (-)	-* (-*)	- (-)
Window.Name	- (-)	- (-)	- (-)	-* (-*)	- (-)
IE-UserData	- (-)	- (-)	- (-)	-* (-*)	- (-)
Global-Storage	- (-)	- (-)	- (-)	-* (-*)	- (-)
Database-Storage	- (-)	- (-)	- (-)	-* (-*)	- (-)
WebCache	- (-)	- (-)	- (-)	-* (-*)	- (-)
CSS-History-Cookie	- (-)	- (-)	- (-)	-* (-*)	- (-)
Flash-Cookie	- (c)	- (c)	- (c)	-* (-*)	c (c)
Java-Cache-Cookie	c (c)	c (c)	- (-)	-* (-*)	- (-)
Google Gears DB	- (-)	- (-)	- (-)	-* (-*)	- (-)

*Der Safari stürzte während des Tests ohne JavaScript ab, sodass ausschließlich der HTTP-Cookie an den Server übertragen werden konnte.

Auf Flash-Cookies hat die Deaktivierung von JavaScript jedoch keine Wirkung. Eine Untersuchung verschiedener Seiten hat jedoch gezeigt, dass Flash häufig mit Hilfe einer JavaScript-Datei eingebunden wird. Da die verschiedenen Browser einen unterschiedlichen HTML-Code benötigen, um den Flash Player zu initialisieren, wird häufig ein Skript namens `swfobject.js` eingebunden, um den Entwicklern eine serverseitige browserabhängige Codegenerierung zu ersparen. In diesem Fall würde das Deaktivieren von JavaScript auch das Setzen von Flash-Cookies verhindern. Flash-Inhalte, die auf diese Weise eingebettet wurden, können allerdings ebenfalls nicht mehr gestartet werden.

6.2.3.3 Privater Modus

Tabelle 8 - Privater Modus

Cookie	IE 9.0.8112 (8.0.6)	Firefox 5.0 (3.6.8)	Chrome 12.0 (9.05)	Safari 5.0.5 (5.0.2)	Opera 11.11 (11.01)
HTTP-Cookie	- (-)	- (-)	- (-)	c* (-)	- (-)
Session-Storage	- (-)	- (-)	- (-)	- (-)	- (-)
Local-Storage	- (-)	- (-)	- (-)	c* (c*)	- (-)
Window.Name	- (-)	- (-)	- (-)	- (-)	- (-)
IE-UserData	- (-)	- (-)	- (-)	- (-)	- (-)
Global-Storage	- (-)	- (-)	- (-)	- (-)	- (-)
Database-Storage	- (-)	- (-)	- (-)	c* (c*)	- (-)
WebCache	- (-)	- (-)	- (-)	- (-)	- (-)
CSS-History-Cookie	- (-)	- (-)	- (-)	- (-)	- (-)
Flash-Cookie	- (-)	- (-)	- (-)	c* (-)	- (-)
Java-Cache-Cookie	c (c)	- (-)	c (-)	- (-)	- (-)
Google Gears DB	- (-)	- (-)	- (c)	- (-)	- (-)

*Die im nicht privaten Modus gespeicherte Daten konnten gelesen werden, aber das Speichern war nicht möglich, sodass Information nach dem Beenden des privaten Modus nicht mehr vorhanden waren beziehungsweise nur die Daten enthielten, die sie schon vor der Sitzung enthielten.

Der Test des privaten Modus (Tabelle 8) zeigt, dass nur Firefox und Opera die Daten einer Sitzung komplett löschen. Die anderen Browser konnten nicht alle Daten löschen wobei der IE und der Chrome Browser nur den Java-Cache-Cookie der erst im Rahmen dieser Arbeit entwickelt wurde, nicht löschen konnten.

Im Safari war es möglich Daten auszulesen, die im normalen Modus gespeichert wurden. Dies führt dazu, dass der Nutzer von allen Seiten, die einen solchen Cookie bereits gesetzt haben, auch im privaten Modus noch identifizierbar ist. Im Safari 5 selbst ist es nicht möglich Local-Storage und Database-Storage zu löschen. Auch im „Private Modus“ des Safari lassen sich weiterhin Daten im Local-Storage und im Database-Storage ablegen und in späteren Sessions wieder auslesen. In einem umfangreichen Test [McK08] zur Sicherheit der Browser im Jahr 2008 war das Löschen des Local-Storage im Safari 4 noch über ein Untermenü in den Einstellungen möglich. Um die beiden Speicher dennoch zu leeren, ist es möglich folgende Ordner von der Festplatte zu löschen.

```
C:\Users\X\AppData\Local\Apple Computer\Safari\Databases  
C:\Users\X\AppData\Local\Apple Computer\Safari\LocalStorage  
„X“ ist durch den jeweiligen Nutzernamen zu ersetzen.
```

Der Google Browser scheiterte in der Version 9 noch an einem Speicher, der ausschließlich in Google Chrome vorhanden ist. Diese Lücke im private Modus des Chrome Browsers führte dazu, dass der Nutzer weiterhin identifizierbar war. In der aktuellen Version ist dieses Problem jedoch behoben.

Da Adobe das Flash Plugin ab der Version 10.3 mit einer API ausgestattet hat, die es dem Browsern ermöglichen soll auch die Flash-Cookies zu löschen, ist bei den neueren Tests bereits in den Browsern IE, Firefox und Chrome zu sehen, dass die Flash-Cookies verschwinden. Dass die Funktion in Safari und Chrome noch nicht enthalten ist, kann daran liegen, dass dies erst für den nächsten Major-Release geplant ist. Bei den anderen Browsern gab es bereits einen Major-Release.

6.2.3.4 Mobile Browser

Ein Test mit Android-x86 2.2 und dem dort integrierten Browser ergab, dass es eine Möglichkeit gibt, Cookies, Verlauf und Cache zu löschen. Eine Überprüfung dieser Löschoptionen im Rahmen dieser Arbeit zeigte jedoch, dass weder Local-Storage noch CSS-History Cookies innerhalb der Anwendung gelöscht werden können.

Die einzige Möglichkeit diese dennoch zu löschen, ist über Settings / Applications / Manage Applications / All den Browser auszuwählen, und dessen Daten mit Clear Data zu entfernen. Der Nachteil dieser Vorgehensweise ist, dass gleichzeitig alle gespeicherten Passwörter, Formulardaten und Bookmarks gelöscht werden.

Aufgrund der Vielzahl von Endgeräten und Browsern im mobilen Bereich lässt sich nur vermuten, dass andere Endgeräte ebenso wie der Android Browser über Lücken im Schutz gegen Cookies verfügen und die Nutzer ihr Recht auf informationelle Selbstbestimmung daher nicht wahrnehmen können.

6.2.3.5 Anti-Cookie Plugins

Da es nur wenige Webseiten gibt, die die Cookietechnologien (außer HTTP-Cookies) zur Erweiterung ihres Angebotes nutzen, sollte ein Privacy Plugin alle Cookies beim Beenden des Browsers löschen. Ausgenommen davon sind HTTP-Cookies. Für HTTP-Cookies sollte eine editierbare Whitelist definieren, welche Webseiten Cookies permanent setzen dürfen und welche Cookies nur bis zum Beenden des Browsers setzen dürfen. Durch diese Maßnahmen wäre der Schutz des Nutzers vor ungewünschtem Tracking gewährleistet, ohne dass der Komfort übermäßig eingeschränkt wird. Zusätzlich wäre eine Anleitung hilfreich, die die einzelnen Funktionen des Programms erklärt.

Tabelle 9 - Cookieabwehrplugins

Plugin	Browser	entfernte Cookies	Anmerkungen
BetterPrivacy	Firefox	SessionStorage, Local-Storage, Global-Storage, Flash-Cookies	automatisierbar, einfache Bedienung, Whitelist für Flash-Cookies, keine Anleitung
Cookie Monster 3.47	IE, Firefox, Chrome, Safari, Opera	HTTP	einfache Bedienung, Whitelist, keine Anleitung
MAXA CookieManager (Standard)	IE, Firefox, Chrome, Safari, Opera	HTTP	automatisierbar, einfache Bedienung, Whitelist, keine Anleitung, (die Pro Version kann weitere Cookiearten löschen)
Cookie Safe Lite 3.05	Firefox	HTTP, SessionStorage, Local-Storage, Global-Storage	Blacklist/Whitelist, temporär erlauben, hoher Aufwand am Anfang (bei Whitelist), keine Anleitung
Ghostery	IE, Firefox, Chrome, Safari	Flash-Cookies, Silverlight	löschen beim Beenden des Browsers, Trackerabwehr durch Blacklist
PrivacySuite 0.7	Firefox	HTTP, SessionStorage, Local-Storage, Global-Storage, WebCache, Flash-Cookies	Management für mehrere Identitäten, Trackerabwehr durch Blacklist

Tabelle 9 zeigt eine Reihe von Plugins, die die Cookieabwehr der Browser verbessern, oder ergänzen sollen. Die Tabelle zeigt deutlich, dass ergänzende Plugins hauptsächlich für den Firefox existieren. Die anderen Browser können durch externe Cookiemanager bequem verwaltet werden. Der Funktionsumfang der einzelnen Abwehrsysteme ist jedoch sehr unterschiedlich. Die meisten löschen nur HTTP-Cookies, wodurch der Nutzer Einschränkungen im Komfort hinnehmen muss, da automatische Logins nicht funktionieren. Einen Schutz gegen aggressives Tracking, bei dem Cookie-Bibliotheken wie zum Beispiel Evercookie eingesetzt werden, bieten diese Systeme jedoch nicht.

Die Spalte „entfernte Cookies“ gibt die tatsächlich entfernten Cookies an und kann daher von der Produktbeschreibung variieren. Alle Produkte in der Tabelle wurden auf der Testseite zum Überprüfen der Cookieabwehr mit dem Firefox getestet.

Der Tabelle ist zu entnehmen, dass eine Kombination aus Ghostery und Cookiesafe einen ausreichenden Schutz bieten würde. Die Tabelle erhebt jedoch keinen Anspruch auf Vollständigkeit und es ist durchaus möglich, dass weitere Kombinationen oder einzelne Plugins existieren, die einen vollwertigen Schutz bieten.

Das größte Problem dieser Plugins und Anwendungen ist die Erstellung einer individuellen Whitelist. Die Frage, ob eine Seite auf eine Whitelist gehört, ist nicht einfach zu beantworten. Es hängt hauptsächlich davon ab, ob der Nutzer dem Anbieter vertraut und ob der Vorteil durch die Nutzung des Cookies die Nachteile überwiegt. Da eventuelle Nachteile jedoch nicht einschätzbar sind, lässt sich die Problematik auf das Vertrauen in den Anbieter reduzieren.

Cookies zu deaktivieren kann ebenso problematisch sein, da man davon ausgehen muss, dass Cookies bei der Mehrheit der Nutzer aktiv sind. Diese regelmäßig zu löschen oder nur temporär zu erlauben, sind jedoch gute Möglichkeiten das Problem zu umgehen.

6.2.4 Auswertung der Ergebnisse

Das Entfernen von Cookies ist für einen normalen Anwender nur schwer durchführbar. Ein großes Problem besteht schon allein darin, dass beim Aufruf der Löschfunktionen der Browser die Cookies nicht sofort gelöscht werden, sondern ein Neustart notwendig ist, auf den jedoch nirgends hingewiesen wird. Ein weiteres Problem ist, dass einige Browser nicht alle browserinternen Speicher auch wirklich löschen. Dies ist für den Nutzer jedoch nicht ersichtlich, da entsprechende Anzeigen für den Inhalt der Speicher fehlen. Allein der private Modus konnte bei allen getesteten Browsern außer dem Safari überzeugen. Dieser konnte einige browserinterne Speicher weder löschen noch im privaten Modus verbergen.

Plugins bergen ein zusätzliches Risiko, da jedes Plugin eine Möglichkeit bieten kann Cookies zu setzen. Der im Rahmen dieser Arbeit entwickelte Java-Cache-Cookie demonstriert, dass die Speichermöglichkeiten von Plugins nicht ausreichend überwacht werden. Eine Einschränkung der Zugriffsrechte von Plugins (außer für Flash) fehlt in allen Browsern völlig.

Die getesteten Anti-Cookie Plugins konnten vor allem den Komfort für den Nutzer wesentlich erhöhen. Dies gelingt den Plugins vor allem durch die automatische Löschung von Cookies beim Beenden des Browsers und durch eine zielgerichtete Menüführung, die die Verwaltung der Cookies erleichtert. Erst bei den erweiterten Funktionen wäre eine Hilfe in Form einer Anleitung nützlich gewesen.

6.3 Anonymisierungsdienste

6.3.1 Vidalia Bundle

Im Rahmen dieser Arbeit wurde das Anonymisierungspaket Vidalia Bundle für den Firefox analysiert. Das Vidalia Bundle besteht aus dem lokalen Proxy Polipo, Vidalia, einem Firefox Plugin namens TOR-Button und einem TOR-Client. TOR ist ein Netzwerk aus Servern, das die eigene IP verschleiert und die übertragenen Daten durch Verschlüsselung schützt.

Tabelle 10 - Cookie-Test Vidalia Bundle

Cookie	Firefox 3.6.17 + Vidalia Bundle
HTTP-Cookie	c (temporär: bis zum Schließen des Browsers)
Session-Storage	-
Local-Storage	-
Window.Name	c (temporär: bis zum Schließen des Browsers)
IE-UserData	-
Global-Storage	-
Database-Storage	-
WebCache	c (temporär: bis zum Schließen des Browsers)
CSS-History-Cookie	-
Flash-Cookie	-
Java-Cache-Cookie	-
Google Gears DB	-

Das Vidalia Bundle erreicht somit einen Schutz, der die meisten Eigenschaften des Nutzers verschleiert. Einige Browsermerkmale sind jedoch weiterhin vorhanden und können ausgelesen werden.

6.3.2 Jondonym

Ein weiterer im Rahmen dieser Arbeit getesteter Anonymisierungsdienst ist Jondonym mit dem JonDoFox. Jondonym ist ein Netzwerk aus Proxyservern, das ähnlich wie TOR funktioniert, die IP-Adresse versteckt und den Inhalt der übertragenen Daten durch Verschlüsselung schützt. JonDoFox ist ein spezielles Profil für den Firefox, dass die verschiedenen Browsermerkmale auf weit verbreitete Merkmale ändert.

Cookies dürfen grundsätzlich nur gesetzt werden wenn der Nutzer das erlaubt. Dazu wird das Anti-Cookie-Plugin CookieMonster 1.05 verwendet. Auf der Testseite war es selbst mit Erlaubnis nur möglich HTTP-Cookies zu setzen. Durch die Aktivierung von JavaScript konnten jedoch auch ein Windows.Name Cookie gesetzt werden. Andere Cookiearten funktionierten weiterhin nicht.

Die Ausführung von JavaScript wird durch das NoScript-Plugin verwaltet. Das Deaktivieren von JavaScript verhindert das Auslesen einiger Cookiearten, verschiedener Browsermerkmale und eine Erhebung biometrischer Daten durch Mausbewegungen. Es führt jedoch auch dazu, dass einige Webseiten nur eingeschränkt oder gar nicht funktionieren.

Die Browsermerkmale sollen bei Jondonym außerdem durch das spezielle JonDoFox-Profil stark geändert werden. Dies konnte anhand der Testseiten von Jondonym und PanoptiClick überprüft und bestätigt werden.

6.3.3 Vor- und Nachteile beider Dienste

Durch die Nutzung eines Proxynetzwerkes wird die Geschwindigkeit beim Surfen im Internet beeinträchtigt. Dieses Problem kann jedoch umgangen werden, indem ein kostenpflichtiges Proxynetze anstelle des voreingestellten benutzt wird. Jondonym bietet einen solchen Service auch an, um den Komfort bei der Nutzung des Dienstes zu erhöhen.

Beide Services setzen zur Anonymisierung ihrer Nutzer auf den Mimikri-Ansatz und nutzen einen UserAgent mit dem Länderkürzel der USA. Dies hat den Nachteil, dass ein deutscher Nutzer, der sich hauptsächlich über deutsche Seiten bewegt, sich vor allem auf kleineren Seiten von anderen Nutzern unterscheidet. Dieser Unterschied wird durch die Nutzung des Proxynetzes noch verstärkt. Das Proxynetze, das die IP des Nutzers verschleiern soll, endet häufig mit einem Proxy in Europa. Dies hat zu Folge, dass der „anonymisierte“ Besucher einer kleinen deutschen Webseite nun einen UserAgent aus der USA hat und eine IP in irgendeinem europäischen Land besitzt. Durch diese beiden Punkte unterscheidet sich der Besucher extrem von den anderen Nutzern. Bei einem erneuten Besuch der Webseite würde der UserAgent gleich bleiben, während die IP wieder aus irgendeinem europäischen Land stammt. Dies ergibt für den Besucher eine

einzigartige Kombination, wodurch er beim Besuch kleiner Seiten anhand der Ähnlichkeit zugeordnet werden kann. Dies funktioniert jedoch nur dann eindeutig, wenn der Webseitenbesucher der einzige Besucher ist, der diesen Anonymisierungsdienst nutzt.

Sowohl beim Vidalia Bundle als auch bei Jondonym werden alle Firefox Plugins/Addons deaktiviert, da diese eine Identifikation des Nutzers ermöglichen könnten. Durch den Einsatz von Plugins, wie Flash und Java, ist es zum Beispiel möglich die installierten Schriftarten auszulesen, wodurch ein sehr aussagekräftiges Merkmal gewonnen wird. Es ist außerdem möglich eine Verbindung zum Webserver aufzubauen, ohne dass diese den Proxyserver nutzt, der die IP versteckt. Der Internetprovider und die Standortdaten des Nutzers könnten dadurch ermittelt werden. Der Nutzer wird durch die Deaktivierung besser geschützt, muss jedoch auch darauf verzichten, Flash, Java und ähnliche Plugins nutzen zu können.

Durch den Einsatz von CookiesMonster und NoScript bietet Jondonym mehr Flexibilität für den Nutzer. Dieser kann nun selbst bestimmen, welche Seiten trotz der Anonymisierung auch permanente Cookies setzen dürfen, da der Nutzer dem Anbieter vertraut.

6.4 Filterprogramme

6.4.1 Werbeblocker

6.4.1.1 Adblock Plus

Einer der im europäischen Raum am häufigsten genutzten Werbeblocker ist Adblock Plus. [Fai10] Dieser Werbeblocker kann genutzt werden, um eine Verbindung zu bekannten Werbenetzwerken zu verhindern. Dadurch, dass die Verbindung zu bestimmten URL verhindert wird, wird auch eine Profildzuordnung durch die Werbeanbieter verhindert. Ein Werbeblocker stellt somit einen einfachen Trackingschutz vor Dritten dar. Die vom Nutzer besuchten Webseiten sind allerdings auch weiterhin in der Lage den Nutzer wieder zu erkennen.

Die Möglichkeit eigene Filter zum Blocken von Werbung zu erstellen, führt zu einem weiteren Vorteil. So ist es zum Beispiel möglich die „Gefällt Mir“-Buttons von Facebook zu filtern, wodurch Facebook weniger Daten über den Nutzer sammeln kann. Im Fall des „Gefällt Mir“-Buttons würde ein Filter dann wie folgt aussehen.

```
|http://www.facebook.com/plugins/like.php?*
```

Das Erstellen eigener Filter wird durch die Möglichkeit Wildcards benutzen zu können wesentlich erleichtert.

6.4.1.2 IE - Tracking Protection List(TPL)

Im Internet Explorer 9 ist ein Werbeblocker in Form eines Trackingschutzes[Mic10] integriert. Genau wie bei Adblock Plus ist es möglich Filterlisten zu importieren und diese um eigene Filter zu erweitern. Die Erstellung eigener Filterlisten ist jedoch kompliziert, da extra eine eigene Liste in einem auf XML basierenden Format erstellt werden muss. Es ist jedoch möglich häufig auftretende Elemente über das Menü zu sperren. Dies ist sehr anwenderfreundlich, doch ein gezieltes Blocken ist durch diese Option nicht möglich .

6.4.1.3 Opera Adblocker

Opera besitzt genau wie der IE einen integrierten Werbeblocker. Dieser arbeitet ebenfalls mit importierbaren Filterlisten, wobei die Filter manuell erweitert werden können. Auch die Filterlisten von Opera unterstützen Wildcards, sodass das Erstellen von eigenen Filtern nicht besonders schwierig ist.

6.4.1.4 Vergleich der Werbeblocker

Tabelle 11 - Werbeblocker Vergleich

Kriterium\Werbeblocker	Adblock Plus	IE TPL	Opera Adblock
importierbare Filterlisten	+	+	+
externe Filterliste (Aktualität der Filter)	+	+	-
eigene Filter	+	+	+
gezieltes Filtern bestimmter Teil-URLs	+	+	+
einfaches filtern häufiger Elemente	-	+	-
reguläre Ausdrücke zum Filtern	+	-	-
Filterlisten Editor	+	-	-
erforderliches Know How zum erstellen eigener Filter	gering	gering	mittel
Komfort	+	+	-

Beim Vergleich der Werbeblocker schneidet das Browser Plugin Adblock Plus am besten ab. Dicht darauf folgt die bisher nur wenig genutzte Werbeblockfunktion des Internet Explorer 9. Der Werbeblocker von Opera kann besonders in Hinblick auf den Komfort nicht überzeugen, da eigene Filter in eine Textdatei eingegeben werden müssen.

6.4.2 Filterproxies Polipo & Squid

Polipo [ChD10] und Squid [Egg06] sind so genannte Webcaching-Proxys. Ihre Hauptaufgabe besteht darin häufig genutzten Webseiten, Grafiken und andere Objekte zwischenspeichern, um so eine Seite, die wiederholt aufgerufen wird, schneller laden zu können. Beide Proxys sind somit in der Lage das Surfen im Internet zu beschleunigen. Dies ist vor allem bei der Nutzung einer langsamen Internetverbindung, oder der Nutzung eines Anonymisierungsnetzwerkes von Vorteil. Das Vidalia Bundle hat deshalb Polipo als festen Bestandteil der Anonymisierungslösung integriert.

Beide Proxys können genutzt werden, um Header-Informationen aus dem HTTP-Protokoll zu verändern. Dadurch werden zum Beispiel HTTP-Cookies nicht an den Nutzer weitergegeben und die Privatsphäre des Nutzers bleibt geschützt.

Eine wichtige Funktion von Polipo ist, dass der Proxy so konfiguriert werden kann, dass Referer nur dann gesendet werden, wenn die aktuelle Second Level Domain mit der angeforderten Second Level Domain übereinstimmt. Dies führt dazu, dass beim Wechsel von einer Webseite auf die nächste nicht festgestellt werden kann, vom welchem Server der Nutzer gekommen ist.

Neben diesen Funktionen bieten Polipo und Squid die Möglichkeit, als Werbeblocker zu dienen und mittels Filterlisten bestimmte URLs zu blockieren. Die Nutzung von Wildcards in den URLs ist bei beiden Proxys erlaubt, um die Erstellung von Filtern zu vereinfachen. Der Vorteil eines filternden Proxyserverns ist, dass alle Nutzer des Proxys keine Werbeblocker mehr benötigen.

6.4.3 Filterprogramme als Trackingschutz

Die Untersuchung der Filterprogramme ergab, dass es mit einem Filterprogramm allein nicht möglich ist Tracking völlig auszuschließen. Dies liegt vor allem daran, dass die Filterlisten nur die zu sperrenden URLs enthalten. Für einen besseren Schutz wäre es notwendig die Verfahren zur Nutzeridentifikation zu erkennen und die Übertragung der Daten zu verhindern.

Da die Filterlisten jedoch hauptsächlich Werbeanbieter enthalten, die den Nutzer über mehrere Seiten verfolgen und somit ein umfangreiches Interessenprofil des Nutzers erstellen können, tragen Filterprogramme wesentlich zum Schutz der Privatsphäre ihrer Nutzer bei.

Um die Privatsphäre des Nutzer wirklich zu schützen, ist zusätzlich die Nutzung eines Browsers im privaten Modus sinnvoll, da nicht alle Cookiearten gefiltert werden können. Auch ein spezielles Browserprofil wie JonDoFox kann den Nutzer zusätzlich schützen. Letztendlich kann ein richtiger Schutz nur durch ein vollständiges Anonymisierungspaket wie Vidalia Bundle oder Jondonym erreicht werden.

Die Kombination eines Filterprogramms mit einem Anonymisierungspaket kann die Geschwindigkeit des Seitenaufbaus spürbar erhöhen, da durch die Filterung von Werbung weniger Daten übertragen werden müssen und die geringe Bandbreite der Proxynetze so besser ausgenutzt wird. Bei den Anonymisierungspaketen Vidalia Bundle und Jondonym sind bereits Filterprogramme gegen Werbung enthalten, sodass eine zusätzliche Installation nicht mehr nötig ist.

7 Ausblick

7.1 IPv6 ohne dynamische IP-Adressen

Da die IPv4 Adressen nahezu aufgebraucht sind, ist es sehr wahrscheinlich, dass IPv6 innerhalb der nächsten Jahre eingeführt wird. Das Problem ist, dass es dann keinen Grund mehr dafür gibt, IP-Adressen dynamisch zu vergeben, da in diesem Fall ausreichend Adressen vorhanden wären.

Falls IPv6 mit statischen Adressen eingeführt werden sollte, bedeutet dies das Ende der Anonymität im Internet, da jeder Nutzer anhand seiner statischen Adresse eindeutig identifizierbar wäre.

Der Schutz der Privatsphäre kann in diesem Fall nur noch durch die Nutzung eines Proxymnetzwerkes wie TOR oder Jonym sichergestellt werden. Eine gesetzlicher Zwang zur Vergabe dynamischer Adressen an Privathaushalte wäre deshalb wünschenswert, um die Privatsphäre der Internetnutzer auch weiterhin zu schützen.

7.2 WebGL

Es ist bereits bekannt, dass die WebGL Schnittstelle, die in einigen Browsern bereits enthalten ist, zu Sicherheitsproblemen führen kann. [Eik11]

Der direkte Zugriff auf eine komplexe Hardware wie die Grafikkarte könnte in Zukunft jedoch auch völlig neue Möglichkeiten bieten, um Identifikationsmuster zu erstellen oder Cookies zu hinterlegen. Die WebGL-Spezifikation sieht frei programmierbare Shader vor. Dies könnte man nutzen, um die Ausführungsgeschwindigkeiten für verschiedene Operationen zu messen und ein Identifikationsmuster zu erstellen. Es wird sich zeigen wie hoch die Entropie dieses neuen Merkmals ist. Die große Vielfalt an Grafikkarten, lässt jedoch den Schluss zu, dass eine Entropie über 5Bit realistisch ist. Dabei ist jedoch zu beachten, dass auch die Grafikkarte mit anderen Merkmalen korreliert. So läuft beispielsweise OSX nur auf spezieller Hardware, wodurch auch nur bestimmte Grafikkarten von diesem Betriebssystem unterstützt werden.

7.3 Biometrie

7.3.1 Gesichtserkennung

Es ist bekannt, dass es Möglichkeiten gibt, Menschen anhand von biometrischen Merkmalen wie dem Fingerabdruck oder dem Gesicht zu identifizieren. Neue Verfahren führen besonders im Bereich der Gesichtserkennung ständig zu besseren Erkennungsraten bei gleichbleibend niedriger Fehlerquote.

Datenschützer bemängeln seit Jahren die zunehmende Anzahl an Webcams in öffentlichen Bereichen. Eine Auswertung der Daten in Form von Gesichtserkennung findet momentan jedoch nur in sicherheitsrelevanten Bereichen als zusätzliche Zugangskontrolle statt (z.B. am Eingang der Bundesdruckerei in Berlin). Da die Auswertung der Daten und das Erstellen einer Datenbank mit den nötigen biometrischen Vergleichsdaten sehr aufwendig ist, wird diese Technik noch nicht im größeren Maßstab eingesetzt.

Es gibt einen Weg eine Datenbank mit Vergleichsbildern für die biometrische Gesichtserkennung von Personen zu generieren. Die dazu nötigen Informationen könnten aus sozialen Netzwerken extrahiert werden. Nutzer dieser Netzwerke sollten es daher vermeiden detaillierte Bilder des eigenen Gesichtes als Profilbild in sozialen Netzwerken zu veröffentlichen. Auch Bilder, auf denen die Person verlinkt ist, könnten sich eignen, um die für eine Erkennung notwendigen biometrischen Daten zu extrahieren. Es gilt auch in diesem Fall der Grundsatz der Datensparsamkeit, da man nicht voraussagen kann, ob diese Informationen nicht irgendwann automatisiert ausgewertet können, um dann missbräuchlich verwendet zu werden.

Um Nutzer im Internet anhand ihres Gesichtes zu identifizieren, benötigt der Webseitenbetreiber neben den Vergleichsbildern, Zugriff auf die Webcam des Nutzers. Die Webseite muss also so gestaltet werden, dass eine Webcam benötigt wird, oder zumindest Vorteile für den Nutzer birgt. Ein Beispiel für einen Dienst der so gestaltet ist, dass eine Webcam benötigt wird, ist zum Beispiel der populäre Dienst Chatroulette. Für den Betreiber eines solchen Dienstes wäre es möglich die Techniken der Gesichtserkennung zu nutzen, um die Nutzer wiederzuerkennen.

Aktuell wird diese Technik bereits bei der Gesichtserkennung von Facebook angewendet. Facebook versucht bei jedem neu hochgeladenen Bild alle Personen zu erkennen, um diese zur Verlinkung vorzuschlagen. [Bra11]

7.3.2 Mausbewegungen

Neben den klassischen biometrischen Merkmalen, wie Fingerabdruck, Körpergröße und Gesicht, lassen sich weitere biometrische Merkmale erheben. Diese Merkmale ermöglichen die Identifizierung eines Menschen durch sein individuelles Verhalten.

Auf modernen Internetseiten kommen heutzutage Statistiktools zum Einsatz, die es den Webseitenentwicklern und Designern erleichtern, die Webseiten an das Verhalten der Benutzer anzupassen und damit die Usability (Nutzerfreundlichkeit) zu verbessern. Bei diesen Statistiken werden heutzutage nicht nur die angeklickten Links ausgewertet, sondern zunehmend auch die Bewegungen der Maus und die Mausklicks aufgezeichnet, um so genannte Heatmaps zu erstellen. Durch eine Heatmap kann optisch verdeutlicht werden, in welchen Bereichen der Webseite sich der größte Anteil an Nutzern bewegt und auf welche Elemente besonders häufig geklickt wird.

Die Daten, die für diese Statistiken genutzt werden, können auch genutzt werden, um Nutzer wiederzuerkennen. Im Bereich der Biometrie gibt bereits umfangreiche Studien im Bereich der Handschriftenerkennung. Es ist möglich aus den Bewegungsgeschwindigkeiten des Stiftes und den Pausen, die durch das Absetzen des Stiftes entstehen, charakteristische Merkmale zu extrahieren. Diese ermöglichen es dann die Person wiederzuerkennen. [Vie05]

Nach dem gleichen Prinzip ist es möglich die Nutzer einer Webseite anhand ihrer Mausbewegungen zu erkennen. Dies wurde bereits im Jahr 2003 wurde in einer Studie [GFJ07] bewiesen. Aufgrund der Erkennungsraten ist das System allein jedoch nicht ausreichend, um Millionen von Webseitennutzern zu unterscheiden. Eine Erweiterung dieses Systems um Informationen aus dem UserAgent und weitere Browsermerkmale würde zu wesentlich besseren Erkennungsraten führen. Es ist also möglich, dass auch biometrische Systeme dieser Art in Zukunft als Hilfsmittel zur Wiedererkennung von Internetnutzern genutzt werden könnten.

Problematisch ist die direkte Abhängigkeit dieser Technik von den verwendeten Eingabegeräten. Die Benutzung eines Touchscreens führt dazu, dass keine Mausbewegungen aufgezeichnet werden und der Nutzer deshalb nicht richtig erkannt werden kann. Die Nutzung verschiedener Eingabegeräte wäre deshalb eine Möglichkeit, um biometrische Merkmale zu verändern, da diese stark von den vorhandenen Sensoren (Maus, Touchscreen, Joystick) abhängig sind.

8 Fazit

Die im Rahmen dieser Arbeit untersuchten Cookies demonstrieren die Vielzahl von Möglichkeiten, die existieren, um einen Nutzer eindeutig einem Profil zuzuordnen. Durch die Entwicklung des Java-Cache-Cookies im Rahmen dieser Arbeit ist die Annahme, dass es in Zukunft noch weitere Cookiearten geben wird durchaus berechtigt. Sehr wahrscheinlich ist, dass die neue Möglichkeit Flash-Cookies zu löschen umgangen wird, indem ein Flash-Cache-Cookie nach dem Vorbild des Java-Cache-Cookies entwickelt wird.

Die Abwehrmaßnahmen der Browser weisen im Moment noch Lücken gegen die eine oder andere Cookieart auf, jedoch sind bei allen Browserherstellern Bestrebungen zu erkennen diese Lücken zu schließen. Die Tests aus den Monaten März und Juni 2011 beweisen, dass bestimmte Lücken in der Cookieabwehr geschlossen wurden. Dabei ist jedoch zu beachten, dass einige der Lücken, wie zum Beispiel die des CSS-History-Cookies, bereits seit 2002 bekannt sind.

Ein gravierendes Problem aller Browser ist, dass nach dem Aufruf der Löschfunktionen ein Neustart notwendig wird, auf den an keiner Stelle hingewiesen wird. Ein Internetnutzer der also seine Cookies löscht, um von der nächsten Seite nicht erkannt zu werden, wird aufgrund des fehlenden Neustarts dennoch erkannt. Der Nutzer ist somit trotz Nutzung der Abwehrmaßnahme nicht geschützt.

Der Private Modus ist eine vorbildliche Lösung zum Schutz der Privatsphäre des Nutzers, da der Nutzer einfach durch einen Knopfdruck zum anonymen Surfen umschalten kann. Dazu ist weder zusätzliches Wissen über Anonymisierung noch zusätzliche Software notwendig. Ähnlichkeitsbasierende Methoden werden durch den Modus jedoch nicht gestört, sodass der Nutzer nur gegen einen Teil der Identifizierungsverfahren geschützt wird. Es ist jedoch nicht erwiesen, dass ähnlichkeitsbasierende Verfahren beim Tracking von Nutzern im Internet im Moment überhaupt eine Rolle spielen. Eindeutige Methoden wie Cookies sind für die meisten Internetnutzer im Moment vollkommen ausreichend. Unter dieser Annahme ist der private Modus des jeweiligen Browsers völlig ausreichend, um weitestgehend anonym im Internet zu surfen. Da es momentan keine Untersuchungen über den praktischen Einsatz von ähnlichkeitsbasierender Profilzuordnung gibt, müssen zukünftige Untersuchungen prüfen, ob bereits Webseiten existieren, die eine Profilzuordnung anhand der Ähnlichkeit durchführen oder ob entsprechende Tools beziehungsweise Frameworks existieren.

Eine im Mai 2011 in Kraft getretene Richtlinie der EU [EPE09] soll den rechtlichen Rahmen für das Setzen von Cookies grundlegend ändern. Der Dienstanbieter ist demnach verpflichtet den Nutzer über das Setzen von Cookies zu informieren. Dabei existiert nur eine Ausnahme: „Ausnahmen von der Informationspflicht und der Einräumung des Rechts auf Ablehnung sollten auf jene Situationen beschränkt sein, in denen die

technische Speicherung oder der Zugriff unverzichtbar sind, um die Nutzung eines vom Teilnehmer oder Nutzer ausdrücklich angeforderten Dienstes zu ermöglichen.“ [EPE09] Das bedeutet, dass ausschließlich Session Cookies gesetzt werden dürfen, da andere Cookies nicht unverzichtbar sind, um die Funktionalität eines Dienstes zu gewährleisten. Persistente Cookies dürften somit ausschließlich mit der Einwilligung des Nutzers gesetzt werden. Die Umsetzung dieser Richtlinie in konkrete Gesetze hätte jedoch auch zur Folge, dass ähnlichkeitsbasierende Verfahren für die Profilzuordnung von Nutzern schlagartig an Bedeutung gewinnen würden.

Ähnlichkeitsbasierende Verfahren stellen eine Möglichkeit der Profilzuordnung dar, die auch dann funktioniert, wenn eindeutige Verfahren nicht anwendbar sind. Der größte Nachteil dieser Verfahren ist, dass diese komplexer und rechenintensiver als eindeutige Verfahren sind. Als Abwehrmaßnahme kommen in diesem Fall nur Anonymisierungspakete in Frage, die das individuelle Identifikationsmuster des Nutzers manipulieren. Ob der von den beiden getesteten Anonymisierungspaketen gewählte Mimikri-Ansatz zur Manipulation des Identifikationsmusters von Vorteil ist, ist gerade bei kleineren und nicht aus den USA stammenden Webseiten zweifelhaft. Der Nutzer könnte durch dieses Identifikationsmuster auch extrem auffallen, statt in der Masse unterzugehen. Durch die Nutzung des Randomize-Ansatzes würde der Nutzer hingegen immer auffallen, ist jedoch dadurch, dass nie 2 Sessions mit dem gleichen Identifikationsmuster entstehen, geschützt.

Der Komfort für den Nutzer ist entscheidend dafür, ob ein Anonymisierungsdienst genutzt wird. Die im Rahmen dieser Arbeit getesteten Anonymisierungspakete ließen sich ohne größeres technisches Know How installieren und nutzen. Größtes Problem ist die Geschwindigkeit dieser Dienste und die Deaktivierung von Plugins wie Flash. Für Nutzer, die auf Flash und schnelle Reaktionszeiten der Webseiten verzichten können, sind diese Pakete gut geeignet, um anonym im Internet zu surfen. Für die übrigen Nutzer wäre es denkbar auf die Nutzung eines Proxynetzes zu verzichten und stattdessen die IP häufiger zu wechseln. Die Aktivierung von Flash ist gerade in Hinblick auf die auslesbaren Schriftarten und deren Entropie jedoch nicht zu empfehlen. Falls sich in Zukunft das HTML5 Video-Tag durchsetzen sollte, könnte auch Flash für viele Nutzer überflüssig werden.

Das Bundesdatenschutzgesetz erlaubt einem Nutzer seine personenbezogenen Daten löschen zu lassen. Da ein Nutzer jedoch häufig gar nicht weiß, dass Daten gespeichert werden, oder dass diese sogar weitergegeben werden, wird diese Möglichkeit häufig nicht genutzt. Hinzu kommt, dass Nutzerprofile, solange der Nutzer seinen wahren Namen nicht selbst angegeben hat, weitestgehend anonym sind. Rechtlich wird das Speichern und Weitergeben anonymer Daten noch immer als unbedenklich angesehen und ist deshalb völlig legal. Eine aktuelle Studie kommt jedoch zu folgendem Schluss: „Data can be either useful or perfectly anonymous but never both.“ [Ohm09] Die Studie

verdeutlicht, dass es durch eine Zusammenführung verschiedener anonymisierter Datensätze möglich ist, die wahre Identität der Nutzer festzustellen. Aus genau diesem Grund sollte ein Nutzer darauf achten, dass ein Profil nie so umfangreich wird, dass seine Identität feststellbar wird.

Im Internet ist bereits eine größere Anzahl von Testseiten verfügbar, auf denen ein Nutzer seine Anonymität prüfen kann. Diese Tests lesen dazu hauptsächlich Browsermerkmale aus, um diese zu bewerten. Dass im Internet hauptsächlich Cookies zur Nutzeridentifikation genutzt werden, wird von diesen Tests jedoch außer Acht gelassen. Die im Rahmen dieser Arbeit entwickelte Webseite ermöglicht es einem Nutzer seine Abwehr gegen eine größere Anzahl verschiedener Cookiearten zu testen. Es ist dem Nutzer dadurch möglich, die eigene Abwehr anhand des Testergebnisses anzupassen. Dies ist die Grundvoraussetzung dafür, dass der Nutzer sein Recht auf informationelle Selbstbestimmung wahrnehmen kann. Ohne die Möglichkeit die eigene Abwehr auch testen zu können, wäre der Nutzer auf die Angaben anderer angewiesen.

Zusammenfassend kann man sagen, dass es möglich ist das Recht auf informationelle Selbstbestimmung wahrzunehmen. Um dieses Ziel jedoch zu erreichen, ist der Nutzer gezwungen geeignete Maßnahmen zum Schutz der eigenen Daten durchzuführen. Die Hauptverantwortung für die im Internet preisgegebenen Daten trägt der Nutzer somit selbst. Dies gilt besonders bei der Nutzung von sozialen Netzwerken, und beim OnlineShopping, da der Nutzer dort eine Vielzahl von Daten preisgibt, die sich unter Umständen mit den vorhanden Nutzerprofilen verknüpfen lassen. Die Anonymisierungspakete bieten einen effizienten Schutz der Nutzer muss jedoch darauf achten, dass er sich während der Nutzung eines solchen Dienstes nicht selbst authentifiziert.

Literaturverzeichnis

Bücher

- [BHK98] John S. Breese, David Heckerman, Carl Kadie: Empirical Analysis of Predictive Algorithms for Collaborative Filtering - In: Proc. 14th Conf. Uncertainty in Artificial Intelligence (Madison WI, 1998).
- [Koc00] Karl-Rudolf Koch: Einführung in die Bayes-Statistik (Heidelberg, 2000).
- [Sch00] Bruce Schneier: Secrets & Lies - Digital Security in a Networked World (New York, 2000).
- [Sch07] Peter Schaar: Das Ende der Privatsphäre - Der Weg in die Überwachungsgesellschaft 2.Auflage (München, 2007).
- [Str05] Pär Ström: Die Überwachungs MAFIA - Das gute Geschäft mit unseren Daten (Wien, 2005).
- [Vie05] C. Vielhauer: Biometric User Authentication for IT Security – From Fundamentals to Handwriting (New York, 2005).

Onlinequellen

- [Ado10] Adobe: Flash Player Version Penetration
http://www.adobe.com/products/player_census/flashplayer/version_penetration.html
(Stand:26.08.2010).
- [Bra11] Herbert Braun: Heise Online - Facebook erkennt Gesichter
<http://www.heise.de/newsticker/meldung/Facebook-erkennt-Gesichter-1256586.html> (Stand:12.07.2011).
- [Bun83] Bundesverfassungsgericht: Das Volkszählungsurteil 1983
http://zensus2011.de/fileadmin/material/pdf/gesetze/volkszaehlunsurteil_1983.pdf (Stand:20.05.2011).
- [ChD10] Juliusz Chroboczek, Christopher Davis: Polipo
<http://www.pps.jussieu.fr/~jch/software/polipo/polipo.pdf> (Stand:24.06.2011).
- [Dmi09] Dmitry Shkinev: iLocalStorage <http://code.google.com/p/ilocalstorage/wiki/API>
(Stand:24.05.2011).
- [Dwo10] Roland Dworschak, Bundesamt für Sicherheit in der Informationstechnik: Web2.0
https://www.bsi.bund.de/cae/servlet/contentblob/476508/publicationFile/30643/web20_pdf.pdf
(Stand:26.08.2010).
- [Eck10] Peter Eckersley, Electronic Frontier Foundation: A Primer on Information Theory and Privacy <https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy>
(Stand:17.07.2011).
- [Eck11] Peter Eckersley, Electronic Frontier Foundation: How Unique Is Your Web Browser? <https://panopticklick.eff.org/browser-uniqueness.pdf> (Stand:03.05.2011).
- [Egg06] Tobias Eggendorfer: Bridge Filter http://www.linux-magazine.com/w3/issue/72/Squid_Bridge_HTTP_Proxy.pdf (Stand:07.06.2011).

- [Eik11] Ronald Eikenberg: Heise Security - WebGL als Sicherheitsproblem
<http://www.heise.de/security/meldung/WebGL-als-Sicherheitsproblem-1240168.html> (Stand:14.06.2011).
- [EPE09] Europäische Parlament, Europarat: Richtlinie 2009/136/EG Des Europäische Parlaments und des Rates <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:337:0011:0036:De:PDF> (Stand:13.07.2011).
- [Fai10] Till Faida: Untersuchung: Wie viel Geld verlieren Webmaster durch Adblocker? Hier neue Zahlen & Fakten <http://t3n.de/news/untersuchung-viel-geld-verlieren-webmaster-adblocker-279287/> (Stand:07.06.2011).
- [FeS02] Edward W. Felten, Michael A. Schneider: Timing Attacks on Web Privacy
<http://www.cs.princeton.edu/sip/pub/webtiming.pdf> (Stand:12.03.2011).
- [Fet10] Ian Fette: Gears - Improving Your Web Browser
<http://gearsblog.blogspot.com/2010/02/hello-html5.html> (Stand:05.04.2011).
- [Fra08] Thomas Frank: Session variables without cookies
<http://www.thomasfrank.se/sessionvars.html> (Stand:15.07.2011).
- [GFJ07] Hugo Gamboa, A.L.N. Fred, A. K. Jain: Webbiometrics: User Verification Via Web Interaction <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.4481&rep=rep1&type=pdf> (Stand:27.05.2011).
- [GrJ10] Jeremiah Grossman: I know who your name, where you work, and live (Safari v4 & v5) <http://jeremiahgrossman.blogspot.com/2010/07/i-know-who-your-name-where-you-work-and.html> (Stand:15.07.2011).
- [Gro10] Jeremiah Grossman: Stealing AutoComplete form data in Internet Explorer 6 & 7
<http://jeremiahgrossman.blogspot.com/2010/07/stealing-autocomplete-form-data-in.html> (Stand:04.05.2011).
- [Han11] Robert "RSnake" Hansen: CSS History Hack Without JavaScript
<http://ha.ckers.org/weird/CSS-history.cgi> (Stand:24.03.2011).
- [JJJ06] Ari Jules, Markus Jakobsson, Tom N. Jagatic: Cache Cookies for Browser Authentication <http://www.ravenwhite.com/files/cachecookies.pdf> (Stand:05.04.2011).
- [Kam10] Samy Kamar: evercookie -- never forget <http://samy.pl/evercookie/> (Stand:05.04.2011).
- [KNN08] Vinod Krishnan, Pradeep Kumar Narayanashetty, Mukesh Nathan: Who predicts Better? – Results from an Online Study Comparing Humans and an Online Recommender System http://www-users.cs.umn.edu/~vinod/index_files/vinod-recsys.pdf (Stand:01.07.2011).
- [Küh09] Ulrich Kühn: Geolokalisierung mit anonymisierten IP-Adressen
<http://www.hamburg.de/contentblob/2104970/data/geo-lokalisierung-dud.pdf> (Stand:10.09.2010).
- [McK08] Katherine McKinley, iSEC: Cleaning Up After Cookies
https://www.isecpartners.com/files/iSEC_Cleaning_Up_After_Cookies.pdf (Stand:02.09.2010).
- [Mic10] Microsoft: IE9 and Privacy: Introducing Tracking Protection
<http://blogs.msdn.com/b/ie/archive/2010/12/07/ie9-and-privacy-introducing-tracking-protection-v8.aspx> (Stand:11.06.2011).
- [Mic11] Microsoft: userData Behavior <http://msdn.microsoft.com/en-us/library/ms531424%28v=vs.85%29.aspx> (Stand:05.04.2011).
- [Moz06] Mozillazine: Dom.storage.enabled <http://kb.mozillazine.org/Dom.storage.enabled> (Stand:31.05.2011).

- [Moz11] Mozilla Developer Network: DOM Storage
<https://developer.mozilla.org/en/dom/storage> (Stand:04.05.2011).
- [Net97] Netscape Communications: HTTP State Management Mechanism
<http://www.ietf.org/rfc/rfc2109.txt> (Stand:23.02.2011).
- [Ohm09] Paul Ohm: Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1450006&http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1450006 (Stand:15.07.2011).
- [OSF11] Open Security Foundation: DATALOSS db - Open Security Foundation
<http://datalossdb.org/> (Stand:23.06.2011).
- [Rap11] RapLeaf: RapLeaf <https://www.rapleaf.com/people/rapleaf101> (Stand:24.06.2011).
- [ScE10] Seth Schoen, Electronic Frontier Foundation: What Information is "Personally Identifiable"? <http://www.eff.org/deeplinks/2009/09/what-information-personally-identifiable> (Stand:31.08.2010).
- [Sch10] Seth Schoen, Electronic Frontier Foundation: New Cookie Technologies: Harder to See and Remove, Widely Used to Track You <http://www.eff.org/deeplinks/2009/09/new-cookie-technologies-harder-see-and-remove-wide> (Stand:26.08.2010).
- [SCM09] Ashkan Soltani, Shannon Canty, Quentin Mayo: Flash Cookies and Privacy
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1446862 (Stand:24.03.2011).
- [Smi10] Eric Smith: iPhone Applications & Privacy Issues: An Analysis of Application Transmission of iPhone Unique Device Identifiers (UDIDs) <http://www.pskl.us/wp/wp-content/uploads/2010/09/iPhone-Applications-Privacy-Issues.pdf> (Stand:28.04.2011).
- [Sta11] StatOwl: Web Browser Plugin Market Share
http://www.statowl.com/plugin_overview.php (Stand:24.03.2011).
- [UNO48] UNO: Allgemeine Erklärung der Menschenrechte
<http://www.un.org/Depts/german/grunddok/ar217a3.html> (Stand:22.09.2010).
- [W3C10] W3C: Webstorage <http://www.w3.org/TR/webstorage/> (Stand:31.08.2010).
- [WaB90] Samuel D. Warren, Louis D. Brandeis: The Right To Privacy
http://www.estig.ipbeja.pt/~ac_direito/privacy.pdf (Stand:22.09.2010).
- [Web07] WebKit: WebKit Does HTML5 Client-side Database Storage
<http://www.webkit.org/blog/126/webkit-does-html5-client-side-database-storage/> (Stand:15.07.2011).
- [Web11] WebKit : The WebKit Open Source Project <http://www.webkit.org/> (Stand:05.04.2011).
- [WSJ11] The Wall Street Journal: What They Know - Mobile <http://blogs.wsj.com/wtk-mobile/> (Stand:23.03.2011).
- [YLW09] Jun Yan, Ning Liu, Gang Wang: How much can Behavioral Targeting Help Online Advertising? <http://www2009.org/proceedings/pdf/p261.pdf> (Stand:29.03.2011).

A Anhang

Tabelle 12 - Webseitenuntersuchung

Webseite	Cookies	Einschränkungen ohne Cookies	Einschränkungen ohne JavaScript
google.de	HTTP, SessionStorage	4	5
facebook.com	HTTP	1	4
google.com	HTTP, SessionStorage	4	5
youtube.com	HTTP, Local-Storage, IE- UserData, Flash-Cookie	4	5
ebay.de	HTTP	2	5
wikipedia.org	keine	5	5
amazon.de	HTTP	2	4
spiegel.de	Flash-Cookies, HTTP-Cookies im Onlineshop	4	3
yahoo.com	HTTP, Flash-Cookies	4	4
bild.de	keine, HTTP-Cookies im Onlineshop	4	4
web.de	keine	2	5
twitter.com	HTTP	1	1
gmx.net	HTTP	5	4
blogspot.com	HTTP	4	4
xing.com	HTTP	1	1
uimserve.net	HTTP	5	5
t-online.de	keine	3	3
leo.org	HTTP	5	5
mpnrs.com	HTTP	5	5
live.com	HTTP	1	2
chip.de	HTTP	5	5
immobilienscout24.de	HTTP	5	3
gutefrage.net	HTTP	5	5
wordpress.com	HTTP, Flash-Cookie	4	4
heise.de	keine	4	5
studivz.net	HTTP	1	3
paypal.com	HTTP	1	1
wer-kennt-wen.de	HTTP	4	5
youporn.com	HTTP	1	1
1und1.de	HTTP	1	1
sueddeutsche.de	HTTP	4	3
meinvz.net	HTTP	1	3
googleusercontent.com	HTTP	1	5
kicker.de	HTTP, Flash-Cookie	4	5
flickr.com	HTTP, Flash-Cookie	3	5
kino.to	HTTP, Flash-Cookie	5	1
welt.de	HTTP	5	3
rtl.de	HTTP, Flash-Cookie	5	3
xhamster.com	HTTP	5	1
apple.com	HTTP	5	4

Webseite	Cookies	Einschränkungen ohne Cookies	Einschränkungen ohne JavaScript
bing.com	HTTP	4	4
microsoft.com	HTTP	5	5
msn.com	HTTP, SessionStorage	5	5
mobile.de	HTTP	4	5
postbank.de	keine	5	3
bahn.de	HTTP	5	2
stern.de	HTTP	5	3
zanox-affiliate.de	HTTP	4	5
livejasmin.com	HTTP	5	1
focus.de	HTTP	5	3

Die in Tabelle 12 aufgeführten Webseiten wurden im Rahmen dieser Arbeit auf ihre Funktionalität ohne Cookies, und ohne JavaScript hin untersucht. Jede einzelne Webseite wurde dazu einmal mit deaktivierten Cookies und einmal mit deaktiviertem JavaScript im Firefox geladen.

Die Einschränkungen der jeweiligen Webseite wurden mit Nummern von 1 – 5 bewertet und sind in der Tabelle aufgeführt. Eine 1 steht dabei für starke Einschränkungen und die 5 für keine Einschränkungen. Bei der Bewertung wurde berücksichtigt, dass die wesentlichen Bestandteile funktionieren sollen. So wurde ein nicht funktionierender Login bei einer Suchmaschine weniger stark abgewertet als bei einem sozialen Netzwerk.

Zusätzlich wurde anhand einer Analyse des JavaScripts, des FlashSpeichers und des HTTP-Cookie-Speichers des Browsers auf den Startseiten untersucht, ob folgende Cookie verwendet werden:

- HTTP-Cookie,
- IE-UserData,
- Session-Storage,
- Local-Storage,
- Global-Storage,
- Database Cookie und
- Flash-Cookie.

Durch die Untersuchung ist nicht auszuschließen, dass die jeweiligen Seiten weitere Cookiearten einsetzen oder dass diese auf den Unterseiten andere Cookies benutzen.

Die Untersuchung wurde am 23.3.2011 durchgeführt.

Abschließende Erklärung

Ich erkläre hiermit, dass die vorliegende Arbeit von mir selbst und ohne fremde Hilfe verfasst wurde. Alle benutzten Quellen sind im Literaturverzeichnis angegeben. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Brandenburg an der Havel, 04.08.2011