Heinrich-Heine-Universität Düsseldorf
Philosophische Fakultät
Informationswissenschaft
Proseminar Information Retrieval – Violeta Trkulja, M.A.

Funktionsweise und Konsequenzen der Erfassung von Benutzerdaten mit Hilfe von Cookies in Internetsuchmaschinen am Beispiel von Google

Vorgelegt von
Daniel Ritter
Matrikelnummer: 905396
daniel@daniel-ritter.de

Einführung

Internetsuchmaschinen wie Google beantworten täglich Millionen von Suchanfragen ihrer Benutzer. Da die Suchvorgänge dieser Benutzer personenbezogen mit Hilfe von Cookies zentral gespeichert werden können, ergibt sich eine hohe datenschutzrechtliche Relevanz. Google besitzt die technischen Möglichkeiten jede Aktion eines Users auf den Seiten der Google-Familie dauerhaft zu protokollieren. Durch diese Datensammlung ist es möglich, Benutzerprofile zu erstellen und beispielsweise das „Google-Verhalten“ bestimmter Interessengruppen genau zu beobachten. So wäre es zum Beispiel möglich, aus einer Datenbank, welche aus Google-User-Suchen besteht, herauszufiltern, welcher User sich für bestimmte Themen interessiert. Da Google zur Zeit einen so riesigen Marktanteil besitzt, dass kaum ein Internetbenutzter Google *nicht* nutzt, dürften Bestrebungen staatlicher Stellen gross sein, Zugriff auf diesen gigantischen Interessenpool von Millionen von Internetnutzern zu erhalten. Mithilfe staatlicher Stellen ist sogar eine Rückverfolgung der Suchanfragen auf einen spezifischen Internet-Client und somit meistens auf ein bestimmtes Individuum möglich.

In dieser Arbeit werde ich zunächst die technischen Hintergründe erläutern, welche dieses so genannte „User-Tracking“ ermöglichen; Danach werde ich versuchen, einige datenschutzrelevante Kritikpunkte anzubringen, welche sich direkt aus der Umsetzung der technischen Möglichkeiten durch Google ergeben. Zum Schluss werde ich einige Hinweise geben, wie man mit aktuellen Browsern der Überwachung entgehen kann.

1. Technischer Teil

1.1 Was sind Cookies ?

Die Möglichkeit Cookies serverseitig zu setzten und auszulesen sind seit 1997 offizieller Bestandteil des Hypertext Transfer Protocols (HTTP) und in RFC 2109 beschrieben. Cookie-Funktionen werden von allen aktuellen Mainstream-Webbrowsern unterstützt. Cookies sollen so genannte „Sessions“, also Benutzersitzungen, ermöglichen. Mittels eines Cookies ist es einem Client möglich sich dem Server gegenüber als ein bestimmter Benutzer auszuweisen. So kann man für Nutzer zum Beispiel komfortable automatische Anmeldemechanismen implementieren oder Voreinstellungen von Benutzern für Webseiten lokal beim jeweiligen Benutzer abspeichern. Dies geschieht durch Ablegen oder Ändern einer Textdatei auf dem Rechner des Clients. Die gesamte

Kommunikation zwischen Webbrowser und Webserver basiert auf dem Hypertext Transfer Protokol, einer „Befehlssprache“ welche die Kommunikation zwischen Client und Server nach bestimmten Regeln festlegt. So gibt es für den Client zum Beispiel Befehle, ein Dokument von einem Server anzufordern. Als Antwort darauf gibt es Statusmeldungen und weitere Befehle des Servers, um dem Client mitzuteilen, ob dieses Dokument ausgeliefert werden kann. Das wohl bekannteste Beispiel eines HTTP Response Codes ist „404 – File not found“. Ein Server kann mit bestimmten Befehlen im HTTP-Response Header, einer Antwort auf eine Anfrage (einen HTTP-Request) des Clients diesen Auffordern einen Cookie anzunehmen und abzuspeichern. Im lokal abgelegten Cookie sollten nach RFC 2109 mindestens 4096 Byte Text abgelegt werden können. Dieser vom Server frei wählbare Text kann aus Klartextvariablen wie zum Beispiel „userlogedin=1“ aber auch aus kryptischen Benutzerkennnummern bestehen die nur für den Server einen Sinn ergeben. Cookies können zeitlich limitiert gespeichert werden, also beispielsweise nur für die aktuelle Sitzung, bis der User seinen Browser schliesst. Jedoch ist auch jedes zukünftige Datum nach RFC gültig für die (expire-time), das Abflaufdatum eines Cookies. Aktuelle Versionen des Microsoft Internet Explorers erlauben das Setzen von Cookies, welche bis ins Jahr 2038 Gültigkeit besitzten, für informationstechnische Verhältnisse also „bis in alle Ewigkeit“. Von einem Server gesetzte Cookies können von diesem Server jederzeit wieder ausgelesen werden, wenn der Benutzer die Webseite erneut besucht, welche den Cookie gesetzt hat. Der Server erhält also die Informationen, welche er auf dem Rechner des Clients abgelegt hat auf Wunsch wieder zurück, auch wenn der Cookie bereits vor einigen Wochen oder sogar vor einigen Jahren gesetzt wurde und der Cookie auf dem Clientrechner nicht gelöscht wurde.

Zusammenfassend kann man sagen, dass die Cookie-“Technologie“ es Serverbetreibern ermöglicht, Textdateien mit beliebigem Inhalt auf Clientrechnern abzulegen und den Inhalt dieser Dateien bei späteren Besuchen des Clients auf dem Webserver wieder auszulesen oder zu ändern.

Syntax des Cookie-Headers nach RFC 2109:

```
cookie          =       "Cookie:" cookie-version
                        1*((";" | ",") cookie-value)
cookie-value    =       NAME "=" VALUE [";" path] [";" domain]
cookie-version  =       "$Version" "=" value
NAME            =       attr
VALUE           =       value
path            =       "$Path" "=" value
domain          =       "$Domain" "=" value
```

1.2 Wie kommt der Cookie auf die Clientrechner ?

Google überprüft grundsätzlich beim Anfordern jeder Seite der Google-Familie ob der Client bereits einen Cookie von Google besitzt. Falls kein Cookie existiert, wird grundsätzlich von Google versucht, diesen Cookie zu setzten. Dies kann leicht mit manchen Webbrowsern überprüft werden.

Mit Hilfe eines HTTP-Protokoll-Sniffers kann die Kommunikation zwischen Browser und Client leicht protokolliert werden. Für das nachfolgende Protokoll habe ich „Live HTTP Headers“, ein kostenloses Plugin für den Mozilla-Webbrowser, benutzt. Nachfolgend führe ich das HTTP-„Gespräch“ zwischen Mozilla und Google, das schlussendlich ein Suchergebnis in meinem Browser hervorbrachte und in dem Google versucht seinen Cookie auf meinem Rechner abzulegen auf:

Hier beginnt der von meinem Browser gesandte Request Header, also der Befehlssatz der von meinem Browser an Google in Folge einer normalen „Google-Suche“ nach dem Suchwort „Cookie“ gesandt wird:

GET /search?q=cookie&ie=UTF-8&oe=UTF-8&hl=de&btnG=Google+Suche&meta= HTTP/1.1

Der GET Befehl fordert ein Dokument an. In diesem Fall das Indexdokument im Unterverzeichnis / search mit verschiedenen Suchparametern, wie zum Beispiel dem Parameter „q“ mit dem Wert „Cookie“, meinem Suchwort und weiteren Parametern, die über die Schaltflächen beim Google-Suchfeld konfiguriert werden können.

Host: www.google.de

Gemäss HTTP/1.1 Spezifikation werden Host und gewünschtes Dokument getrennt voneinander übermittelt, um Virtuelle Hosts, also mehrere Domains auf einer einzigen IP-Adresse zu unterstützen.

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.6) Gecko/20040113

Hier weist sich mein Mozilla Browser als auf Windows laufender Mozilla 1.6 (englisch) aus.

Accept: application/x-shockwave-flash,text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1

Hier teilt mein Browser dem Google-Server mit, welche MIME-Typen er zu deuten weiss.

Accept-Language: en-us,en;q=0.5

Hier teilt der Browser die preferierte Sprache mit, falls Dokumente vom Server mehrsprachig ausgeliefert werden können.

Accept-Encoding: gzip,deflate

Hier teilt der Browser mit, dass er auch gepackte Dateien empfangen kann. Dies wird benutzt, um Internettraffic geringer zu halten. Falls Server und Browser es unterstützen, können Dokumente komprimiert übermittelt werden. Sie werden dann auf der Clientseite nach Empfang wieder ausgepackt und danach erst angezeigt.

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Hier teilt mein Browser dem Server mit, welches Character-Set, also welche Zeichenkodierung er versteht. Manche Server können Dokumente in verschiedenen Kodierungen, zum Beispiel mit den deutschen Umlauten ausliefern. Deswegen kann diese Information für den Server interessant sein.

Keep-Alive: 300

Mozilla teilt dem Browser mit, dass die Verbindung 300 Sekunden clientseitig gehalten wird und dass Antworten über dasselbe Socket zurückgesandt werden können, dass also keine neue Verbindung geöffnet werden muss.

Connection: keep-alive

Mozilla teilt dem Server mit dass es sich bei der Verbindung um eine, wie oben bereits beschriebene keep-alive Verbindung handelt. Dieser Parameter ist sinnlos, da nach der HTTP-Spezifikation alle Verbindungen automatisch keep-alive Verbindungen sind, wenn dies nicht explizit verneint wird.

Referer: http://www.google.de/

Hier sendet mein Browser den HTTP-Referer an Google, also die URL der Seite, welche als letztes geladen wurde, bevor die aktuelle Anfrage gestellt wurde. Dieser Wert kann serverseitig auch ausgelesen und gespeichert werden und erlaubt zum Beispiel herauszufinden, von wo aus Benutzer die eigene Webseite besuchen.

Hier beginnt der vom Google Server gesandte „Response-Header", also die Antwort von Google auf meine soeben abgesetzte Suche:

HTTP/1.x 200 OK

Google teilt mit, dass es HTTP 1.0 sowie HTTP 1.1 versteht. Danach folgt der Response Code 200 und die für Menschen besser verständliche Klartextmeldung „OK". Google hat also den Request meines Browsers verstanden und das Dokument existiert. Google teilt meinem Browser mit, dass alles in Ordnung ist und dass das von mir gewünschte Dokument nun ausgeliefert wird. Bei einem nicht existierenden Dokument hätte Google hier ein „HTTP/1.x 404 File not found" gesandt.

Cache-Control: private

Google sendet diesen Befehl um zu verhindern, dass das Dokument auf dem Client-Rechner „gecached", also zwischengespeichert wird. Ob dieser Wunsch vom Browser akzeptiert wird, liegt einzig und allein in der Implementierung dieser Funktion im Browser.

Content-Type: text/html

Hier teilt Google meinem Browser mit, in welchem Datenformat die Antwort gesandt werden wird, also in Klartext, HTML. Hätte ich ein GIF-Bild angefordert wäre hier beispielsweise image/gif erschienen.

Set-Cookie:
PREF=ID=15a41e05bcce1796:LD=de:TM=1076506595:LM=1076506595:S=uBqn1YSlg0Jn-fBE; expires=Sun, 17-Jan-2038 19:14:07 GMT; path=/; domain=.google.de
Hier fordert Google meinen Browser auf den Cookie anzunehmen und abzuspeichern. Da dies der für uns interessanteste Teil der HTTP-Konversation ist, werde ich den Inhalt dieses Befehles im übernächsten Kapitel, so weit wie möglich, entschlüsseln.

Server: GWS/2.1
Hier weist sich die Webserversoftware aus. Wir haben es vermutlich mit dem „Google Web Server Version 2.1“ zu tun.

Transfer-Encoding: chunked
Da Mozilla dies unterstützt, wird das Dokument „chunked“ übertragen, also wird bereits mit der Übertragung des dynamisch generierten Suchergebnisses begonnen, bevor feststeht, wie gross dieses tatsächlich sein wird. Dies ist rescourcenschonend für Serverbetreiber, da die auszuliefernden Dokumente nicht komplett in den Speicher geladen werden müssen.

Content-Encoding: gzip
Google überträgt das Dokument komprimiert, da Mozilla in seinem Request Header mitgeteilt hat, dass komprimierte Dokumente akzeptiert werden.

Date: Wed, 11 Feb 2004 13:36:35 GMT
Das Datum der Antwort.

Nun folgt die gzip-komprimierte HTML Antwort auf meine Suchanfrage, die der Browser auspacken, rendern und mir, dem Endbenutzer in Form eines Google-Suchergebnisses darstellen wird.

1.3 Wie kommt der gesetzte Cookie zurück zu Google ?
Gemäss der HTTP-Spezifikation überprüfen Browser bei jedem Request an einen Server, ob der Server bei einer vorhergehenden Response einen Cookie gesetzt hat und falls dies der Fall ist übermitteln sie diesen in Ihrem Request Header. Nachfolgend der (relevate Teil) eines Request Headers zu Google bei einem bereits verhandenen Google-Cookie :

Cookie:*PREF=ID=15a41e05bcce1796:LD=de:TM=1076506595:LM=1076506595:S=uBqn1YSlg0Jn-fBE;*

Theoretisch würde also diese Information bis ins Jahr 2038 bei jedem Zugriff auf eine Google-Seite an Google versandt werden.

1.4 Analyse des Google-Cookies

Nachdem der Prozess des Erhaltens und Versendens von Cookies nun geklärt ist, möchte ich die Informationen, die in dem oben identifizierten Google-Cookie abgelegt wurden, so weit wie möglich aufschlüsseln. Dazu teile ich die Informationen im Befehl den Cookie zu setzten zunächst in eine etwas lesbarere Ordnung auf:

Set-Cookie:
PREF=
ID=15a41e05bcce1796:
LD=de:
TM=1076506595:
LM=1076506595:
S=uBqn1YSlg0Jn-fBE;

expires=Sun, 17-Jan-2038 19:14:07 GMT;
path=/;
domain=.google.de

Set-Cookie:

Der HTTP-Befehl einen Cookie zu setzten, hat mit dem eigentlichen Cookie nichts zu tun.

PREF=

Der Name des Cookies heisst „PREF“. Server können mehrere verschiedene Cookies ablegen, deshalb müssen Cookies benannt werden.

LD=de:

Sprachvoreinstellung (deutsch). Wird der Cookie von google.de gesetzt, wird diese Einstellung vorgenommen. Besucht man mit diesem Cookie google.com, wird man automatisch zu google.de weitergeleitet.

expires=Sun, 17-Jan-2038 19:14:07 GMT;

Die Gültigkeitsdauer dieses Cookies.

path=/;

Dies ist der Pfad auf google.de in dem der Cookie vom Server ausgelesen werden kann. Er steht auf „/“ root. Also gilt der Cookie für jedes Dokument auf google.de.

domain=.google.de

Dies ist die Domain für die der Cookie gilt und von welcher er ausgelesen werden kann.

ID=15a41e05bcce1796:

Hier findet sich nun der Kern des Problems, die eindeutige Benutzerkennnummer, die Google bei den Benutzern mit Hilfe des Cookies ablegt.

TM=1076506595:

Hier findet sich das aktuelle Datum und die Uhrzeit zu welcher der Cookie auf dem Clientrechner abgelegt wurde. Die Zeit ist verschlüsselt in einem UNIX-Timestamp, einer in der Informationstechnologie gängigen Methode Zeit-Daten abzuspeichern. Der Wert gibt die Anzahl der Sekunden seit dem 1.1.1970 um 0:00 Uhr an. Daten in dieser Form abzulegen ist ein in der Programmierung gängiger Stadard, da es möglich ist mit solchen Daten viel leichter zu rechnen als mit mit echten „menschlich lesbaren“ Datumsangaben. So gut wie alle gängigen Programmiersprachen bieten Möglichkeiten an diesen Timestamp wieder in eine menschlich lesbare Form zurückzuführen. Hier ein Beispiel in der Programmiersprache PHP:

```
<?
echo date("l dS of F Y h:i:s A",1076506595);
?>
```

Ergebnis dieses Codes:
Wednesday 11th of February 2004 02:36:35 PM

also genau der Zeitpunkt, an dem von mir der Request an Google gesandt wurde.

LM=1076506595:
Auch dies ist ein UNIX-Timestamp, er zeigt an wann das letzte Mal die Preferences, also die Voreinstellungen bei Google geändert worden sind. Er gleicht in diesem Fall dem ersten Timestamp, da ich bis jetzt keine Voreinstellungen geändert habe.

S=uBqn1YSlg0Jn-fBE;
Diese Information im Cookie ist ohne das Wissen über Google-Interna nicht zu entschlüsseln. Sie stellt jedoch möglicherweise eine Checksumme da, die es Google ermöglicht zu überprüfen ob die obigen Informationen tatsächlich von Google in den Cookie geschrieben worden sind oder ob der Cookie vom Benutzer manipuliert wurde.

2. Datenschutzrelevante Schlussfolgerungen aus der technischen Analyse

2.1. Einleitung

Nach dieser sehr technischen Analyse der Funktionalität von Cookies möchte ich nun auf die datenschutzrelevanten Probleme zu Sprechen kommen, die solch eine eindeutige Identifikationsnummer mit sich bringen kann. Da Google nicht preisgibt, was genau mit den Benutzerinformationen geschieht und auch nicht angibt, welche Informationen über die Benutzer abgelegt werden, bleibt mir leider nur die Möglichkeit Szenarien zu erstellen was mit den

gesammelten Daten geschehen *könnte*. Da ich mich lieber an Fakten als an Mutmassungen halte, werde ich den nun folgenden Teil meiner Arbeit jedoch zunächst mit einer Liste der Daten beginnen die Google mit seinen technischen Möglichkeiten über den User sammeln kann. Ob dies wirklich geschieht ist für mich leider auch nicht verifizierbar.

2.2 Welche Daten kann Google personenbezogen abspeichern, beziehungsweise an welche weiteren Daten können Google und theoretische Partizipanten mit den erfassten Daten gelangen ?

- Land, eventuell Region. Bei jedem Request wird die eigene IP-Adresse übertragen, da sonst gar keine Antwort des Servers an den Client möglich wäre. Die IP-Adressen lassen sich immer bestimmten Providern, unter Umständen sogar einem bestimmten Provider in einer Stadt oder Region zuteilen. Mit Hilfe staatlicher Stellen ist sogar die Identifikation eines bestimmtem Internet-PC's möglich durch Verbindung der IP mit den gespeicherten Verbindungsdaten des Providers des entsprechenden Benutzers.

- Die Häufigkeit und genaue Datierung der Besuche bei Google.

- Komplette Suchhistorie des Benutzers in allen Google-Diensten. Daraus resultierend ein sehr engmaschiges Persönlichkeits- , Interessen- und Konsumprofil bei aktiven Google-Benutzern.

- Benutzer Browser.

- Benutztes Betriebssystem

- Browsereinstellungen.

- Seiten, von denen aus man zu Google gekommen ist. (HTTP-Referer)

- Gesprochene Sprache(n).

2.3 Einige für den Datenschutz bedenkliche Szenarien die aus der oben geschilderten Problematik entstehen könnten

2.3.1 Das Google-Interne-Szenario

Google beschreibt in seiner Privacy-Policy selbst, dass Suchanfragen der Benutzer dazu genutzt werden, die individuellen Sucherfahrungen von Benutzern zu verbessern. Google selbst hat so einen sehr genauen Überblick darüber, welche Interessen Benutzer aus einem bestimmten Kulturkreis zur Zeit haben. Einen Teil dieser Erhebungen, die Google intern anstellt, werden übrigens von Google selbst unter „Google Zeitgeist“ veröffentlicht. Ein Indiz dafür, dass Google Benutzerdaten durchaus sammelt und auch auswertet. Google selbst wäre es möglich, Usern gezielte Suchergebnisse zu liefern und die Benutzer auf diese Art und Weise zu Seiten zu lenken, welche Google für den Benutzer als interessant oder Relevant erachtet. Google kann auch selbst alle Möglichkeiten nutzen, die ich im nachfolgenden Abschnitt „Das Konsumszenario“ beschreibe.

2.3.2 Das Konsumszenario

Die Googledatenspeicherung eröffnet ungeahnte Möglichkeiten für Werbetreibende und kommerzielle Dienste. So ist relativ eindeutig zu erkennen, für welche Produkte sich ein Benutzer interessiert. Häufen sich zum Beispiel die Anfragen nach „tuning“ , „bmw“ , „rückspiegel“ , „autohaus“ , etc. kann man mit ziemlicher Sicherheit davon ausgehen, einen Autonarr, der sich für BMW's interessiert als aktuellen Client zu haben. Die Information dass ebendiese Person X ein Autonarr ist, dürfte für Webseitenbetreiber aus der Automobil- und Tuningbranche äusserst interessant sein. Es ist technisch möglich, den Betreibern anderer Webseiten das Auslesen von Cookies zu erlauben. Hätte also eine andere Webseite die Cookie-Informationen eines Google-Benutzers und würde Google diesem Webseitenbetreiber einen Dienst bereit stellen über die Benutzernummer dieses Benutzers an seine Suchanfragen zu kommen, wäre dieser Dienst vielen Parteien sicher bares Geld wert. Ausserdem wäre es so möglich, dem Benutzer auf seine Bedürfnisse zurechtgeschneiderte Werbeeinblendungen auszuliefern. Diese Vorgehensweisen wären das elektronische Pendant zu den seit Jahrzehnten agierenden Adressenhändlern, bei denen Firmen Adressdaten von bestimmten Konsumgruppen käuflich erwerben können. Der neue Google-Dienst „Froogle“, eine Produktsuchmaschine passt erstaunlich gut in das soeben geschilderte Szenario. Es gibt keine mir bekannten Beweise, dass Google dieses tut, jedoch ist es in Zukunft nicht ausgeschlossen und technisch ohne Probleme möglich.

2.3.3 Das Überwachungsszenario

Meiner Meinung am bedenklichsten sind die umfassenden Überwachungsmöglichkeiten von Individuen, die Google durch seine hypothetische, aber doch wahrscheinliche Vorratsdatenspeicherung realisieren kann. So wäre es technisch möglich, durch intelligente Filter, welche die angesammelten Datenmengen durchforsten wie bei einer Rasterfahndung, Rückschlüsse auf Individuen zu ziehen, welche sich für ganz bestimmte Dinge Interessieren. Diese Daten wären für Strafverfolgungsbehörden und Geheimdienste von sehr grossem Wert. So ist seit einiger Zeit ein Ex-CIA-Mitarbeiter ein Angestellter von Google. Möglich wäre die gezielte Erfassung von Personen einer bestimmten politischen Gesinnung, bestimmter sexueller Präferenzen oder einer bestimmten Religion. Wie bereits weiter oben beschrieben ist mit Hilfe von Providerdaten welche die Provider in Deutschland und in den USA den Strafverfolgungsbehörden übergeben müssen ein Rückschluss auf einzelne Personen möglich, nicht nur auf einzelne *unbekannte* Personen. Interessant könnte Google in der Ermittlungsarbeit sein, falls zum Beispiel von Benutzern nach bestimmten (seltenen) Eigennamen verdächtiger Personen gesucht wird oder. in Fällen von Industriespionage, kryptische Produktbezeichnungen gesucht werden. Die Möglichkeiten einen solchen Datenpool auszuschöpfen sind endlos. Ich habe nur versucht einige plausible Beispiele zu nennen. Da die Cookies sich nicht auf die Web-Suche allein beschränken, sondern auch bei Google News, Google Groups und weiteren Diensten aktiv sind, sind sogar die Meinungen von Personen inklusive E-Mail-Adresse (bei Google Groups) und die tagesaktuellen Interessen der Benutzer (bei Google News) offengelegt. Auch diese Möglichkeiten sind nur Mutmassungen, technisch möglich wären sie jedoch ohne Probleme und neue Gesetzgebungen, besonders seit den WTC-Anschlägen 2001, zeigen einen deutlichen Trend auf, gerade solche Datenpools ausschöpfen zu wollen.

2.4 Möglichkeiten Google Cookies zu deaktivieren

2.4.1 Internet Explorer

Leider bietet der Internet Explorer keine Möglichkeit Cookies nur für bestimmte Seiten zu aktivieren oder zu deaktivieren. Cookies können entweder komplett ein- oder ausgeschaltet werden. Dies wird erreicht im Menü:
Extras / Internetoptionen / Datenschutz /Cookies Sperren

Software von Drittanbietern wie zum Beispiel „WebWasher“ erlaubt es auch Internet-Explorer-Nutzern Cookies auf einer „per-Seiten-Basis“ komfortabel zu managen.

2.4.2 Mozilla

Mozilla bietet sehr komfortable Möglichkeiten an, Cookies auch seitenbezogen zu erlauben oder zu verbieten.
Dies wird erreicht im Menü:
Edit / Preferences / Privacy & Security / Cookies

Hier ist es möglich die Option „Ask me before storing a cookie“ zu wählen. So erhält man die Möglichkeit zu wählen, ob ein Cookie auf der Festplatte abgelegt werden darf.

Schlusswort

Der einzige Komfort, den Google seinen Benutzern durch die Cookies bietet, ist das Abspeichern von Sucheinstellungen. Also zum Beispiel die Sprachen in denen die Suchergebnisse geliefert werden sollen oder ob sich nach einem Klick auf ein Ergebnis die Seite in einem neuen Fenster öffnen soll. Ich bin der Meinung, dass dieser Komfort nicht aufzuwiegen ist mit dem potentiellen Verlust persönlicher Freiheit der durch die Google-Cookies enstehen könnte. Deshalb möchte ich empfehlen, Google und anderen Servern, bei denen kein ersichtlicher Grund besteht Cookies zu erlauben, insbesondere wenn in den Cookies eindeutige ID-Nummern oder Benutzernamen gespeichert werden, das Ablegen von Cookies zu verbieten.

Quellennachweis

RFC 2616 HTTP 1.1
http://www.w3.org/Protocols/rfc2616/rfc2616.html
Juni 1999 (11.02.2004)

RFC 2109 Zusatz über Cookies zum RFC Protokoll
http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2109.html
Februar 1997 (11.02.2004)

Google's Privacy Policy
http://www.google.com/privacy.html
(11.02.2004)

Google Watch „Google's Cookie“
http://www.google-watch.org/cgi-bin/cookie.htm
(11.02.2004)

Google Zeigeist
http://www.google.com/press/zeitgeist.html
(11.02.2004)