

www.fabiankeil.de/privoxy-anleitung/

Anleitung zum werbefreien und spurenarmen Surfen mit Privoxy

1. [Einleitung](#)
2. [Funktionsprinzip eines Proxys](#)
3. [Verfügbarkeit](#)
4. [Grundkonfiguration ...](#)
 - [... von Privoxy](#)
 - [... des Browsers](#)
5. [Optionale Anpassungen](#)
 - [Vorstellung der Konfigurationsdateien](#)
 - [Konfiguration mit Privoxys Webinterface](#)
 - [Grundprinzip der Aktionsdateien default.action und user.action](#)
 - [Einfaches Beispiel: Unterdrückung von Iframe-Werbung](#)
 - [Komplexeres Beispiel: Unterdrückung von Googles Sponsored Links](#)
 - [Privoxy als Schutz der Privatsphäre](#)
 - [Privoxy und Tor](#)
 - [Cookies](#)
 - [JavaScript-Nervereien unterdrücken](#)
 - [Referrer fälschen oder \(bedingt\) löschen](#)
 - [Browser und Betriebssystem verbergen](#)
 - [Risiken bei HTTPS-Verbindungen](#)
6. [Privoxy-Probleme](#)
7. [Weitere erwähnenswerte Privoxy-Funktionen](#)

Einleitung

Privoxy ist ein Proxy-Server, der die Privatsphäre des Benutzers beim Surfen im Web erhöht. Der Name steht für "Privacy Enhancing Proxy", Privoxy ist jedoch mehr als *nur* ein Schutz der Privatsphäre. Der Proxy-Server ermöglicht es, alle aufgerufenen Seiten frei nach Geschmack zu filtern und umzuschreiben. Privoxy kann nicht nur lästige Spionage-Cookies und nicht weniger lästige Banner unterdrücken, sondern auch neumodischere Werbung auf Textebene vernichten.

Funktionsprinzip eines Proxys

Ein Proxy ist ein Programm, das zwischen Client und Server sitzt, Anfragen vom Client annimmt und an den Server weitergibt. Der Einsatz von Proxys (im Englischen Proxies) kann mehrere Gründe haben, die verbreitetsten sind Erhöhung der Sicherheit oder Effizienz.

Zur Steigerung der Sicherheit wird dem Client durch Einsatz einer Firewall verboten, außerhalb seines lokalen Netzwerks zu kommunizieren. Stattdessen muss er einen Proxy benutzen, der als einziger von der Firewall nicht blockiert wird und der die Anfragen vor der Bearbeitung auf Zulässigkeit überprüfen und bei Bedarf in einem Logfile festhalten kann. Soll die Effizienz gesteigert werden, speichert der Proxy einmal abgerufene Seiten eine Zeit lang (caching). Bei der Anforderung einer bereits im Speicher (Cache) vorhandenen Seite wird diese direkt geliefert, das Netzwerk somit entlastet.

Im Fall von Privoxy ist der Browser der Client. Privoxy kann, wenn die Ansprüche nicht zu hoch sind, als Sicherheits-Proxy eingesetzt werden, es gibt jedoch andere Proxys, die genau dafür geschaffen sind und mehr Möglichkeiten bei der Konfiguration bieten.

Privoxy ist auf die Erhöhung der Privatsphäre, die Unterdrückung von Werbung und die Veränderung von angeforderten Seiten spezialisiert, kann aber vor oder hinter einen anderen Proxy geschaltet werden, zum Beispiel vor [Squid](#) oder [Polipo](#), um sowohl zu filtern, als auch zu cachen.

Versucht der Benutzer eine Webseite aufzurufen, teilt der Browser Privoxy die Zieladresse mit, Privoxy überprüft ob die Adresse abgerufen werden darf, holt wenn die Überprüfung positiv ausgefallen ist die Webseite vom Server, überarbeitet die Seite eventuell und gibt sie an den Browser zur Darstellung weiter.

Enthält die Webseite Grafiken, schickt der Browser für jede Grafik eine neue Anfrage. Privoxy kann dadurch jede Grafik einzeln bearbeiten und Navigationselemente erlauben, aber Werbung abblocken. Privoxy filtert bereits die Webseite selbst, wenn eine Werbegrafik erkannt wird, wird die Adresse umgeschrieben und zeigt auf eine von Privoxy bereit gehaltene Ersatzgrafik. Der Surfer kann so erkennen, dass Werbung vorhanden war, wird aber nicht durch Animationen oder grelle Farben vom Seiteninhalt abgelenkt.

Verfügbarkeit

Privoxy läuft auf einer Vielzahl von Betriebssystemen, die (nicht vollständige) Liste umfasst:

1. Amiga OS,
2. BeOS,
3. FreeBSD, NetBSD und OpenBSD
4. GNU/Linux,
5. Mac OS X,
6. OS/2
7. Solaris
8. und Windows (ab 95)

Fertige Privoxy-Pakete können bei SourceForge.net runter geladen werden, Privoxy steht unter GPL.

Die Installation erfolgt nach den Gepflogenheiten des Betriebssystems: Bei Windows durch Doppelklick der Exe-Datei, bei den Unix-Derivaten durch Verwendung eines Paketmanagers oder durch Kompilation des Quellcodes.

Ein Hinweis für FreeBSD-Benutzer: Privoxy ist Teil der Ports-Collection und kann durch `make install clean` in `"/usr/ports/www/privoxy"` installiert werden, ohne dass der Benutzer sich um Abhängigkeiten kümmern muss.

Auch für die meisten GNU/Linux-Distributionen gibt es angepasste Privoxy-Pakete, deren Benutzung die Installation erleichtern kann.

Momentan aktuell ist Privoxy 3.0.6, der Umstieg von 3.0.3 bringt eine Reihe neuer Funktionen mit und ist allen Privoxy-Nutzern empfohlen.

Grundkonfiguration ...

... von Privoxy

Unter Windows ist Privoxy direkt nach der Installation einsatzbereit und startet mit jedem Booten. Unter anderen Betriebssystemen ist eventuell noch ein bisschen Arbeit angesagt, die sich jedoch auf wenige Zeilen beschränkt. Dort wird Privoxy von der Kommandozeile gestartet und die Konfigurationsdatei "config" als Argument angegeben. Mit der Option `--user benutzername.gruppenname` können zusätzlich der Benutzer und die Gruppe angepasst werden um die Sicherheit zu erhöhen. Wenn Gruppen- und Nutzer-Name identisch sind, reicht die Angabe des Nutzernamens aus.

Unter FreeBSD wird – wenn man wie beschrieben die Ports-Collection benutzt – Privoxy nach `"/usr/local/sbin/"` installiert, die Konfigurationsdateien nach `"/usr/local/etc/privoxy/"` und die Log-Dateien nach `"/var/log/privoxy/"`.

Der manuelle Privoxy-Aufruf lautet folglich: `/usr/local/sbin/privoxy --user privoxy /usr/local/etc/privoxy/config`. Der Benutzer "privoxy" und die gleichnamige Gruppe wurden vom Port bei der Installation angelegt.

Um Privoxy beim Rechnerstart automatisch zu starten muss unter FreeBSD `"/etc/rc.conf"` nur noch um die Einträge:

```
privoxy_enable="YES"
```

ergänzt werden, der Rest läuft über das rcNG-System.

Unter anderen Betriebssystemen müssen die Pfade entsprechend geändert werden, manchmal liegen die Konfigurationsdateien in `"/etc/privoxy/"` oder auch im Verzeichnis der Privoxy-Binärdatei. Viele Pakete bringen ebenfalls ein fertiges Start-Skript mit, die Benutzung sollte dokumentiert sein.

... des Browsers

Bevor ein Browser Privoxy als Proxy-Server einsetzt, muss er entsprechend konfiguriert werden. Bei der Vielzahl verschiedener Browser ist es nicht möglich hier jeden einzelnen zu berücksichtigen, die Einstellungen ähneln sich jedoch stark. Ein paar Beispiele wie man die Konfigurationsmasken bei unterschiedlichen grafischen Browsern erreichen kann:

- Microsoft Internet Explorer: "Extras/Verbindungen/LAN-Einstellungen/Erweitert"
- Mozilla Firefox: "Edit/Preferences/Connection Settings/Manual Proxy Configuration"
- Konqueror: "Settings/Configure Konqueror/Proxy/Use Proxy/Manually Specify Settings/Setup"

In der Konfigurationsmaske ist bei HTTP-Proxy und HTTPS-Proxy "127.0.0.1" als Adresse und "8118" als Port einzutragen. Als letzte Hürde gilt herauszufinden ob das Protokoll mit angegeben werden soll. Bei Internet Explorer und Mozilla Firefox ist dies nicht der Fall, Konqueror verlangt jedoch "http://127.0.0.1" sowie "https://127.0.0.1", ohne das es aus der Eingabemaske hervor geht.

Kommandozeilenbrowser wie Lynx haben naturgemäß keine grafische Eingabemaske. Sie halten sich an interne Konfigurationsdateien, oder werten die Variable `http_proxy` aus. Diese wird mit `export http_proxy="http://127.0.0.1:8118/"` gesetzt. Am besten in einer Startdatei die beim Booten aufgerufen wird, alternativ vor jedem Browseraufruf.

Erster Test

Ob die Einstellungen erfolgreich waren, lässt sich durch Aufrufen der Seite <http://config.privoxy.org/> klären. Im Erfolgsfall meldet sich dort Privoxy mit "This is Privoxy 3.0.3 on localhost (127.0.0.1), port 8118, enabled" (Werte können abweichen).

Wird stattdessen nach <http://www.privoxy.org/config/> weitergeleitet, so ist die Konfiguration fehlgeschlagen, Privoxy ist nicht gestartet, oder der Browser hat die alte Seite zwischengespeichert. Ein Leeren des Caches oder ein neuer Konfigurationsversuch bringen Klarheit.

Nach der Grundkonfiguration filtert Privoxy bereits mit einer beachtlichen Trefferquote. Viele Privoxy-Nutzer begnügen sich mit der Installation und nehmen keine Anpassungen an der Konfiguration vor. Dagegen ist nichts einzuwenden, auch wenn Privoxy so nicht ausgereizt wird.

Optionale Anpassungen

Um die Trefferquote weiter zu erhöhen, oder um Privoxy an eigene Bedürfnisse abseits der Werbeunterdrückung anzupassen, ist die Änderung verschiedener Konfigurationsdateien erforderlich. Sie lassen sich in zwei Kategorien aufteilen: "Aktions-" und "Filterdateien".

Vorstellung der Konfigurationsdateien

Aktionsdateien

Aktionsdateien haben die Dateierdung .action und legen fest, welche Handlung (filtern, blockieren ...) mit welchen Seiten vollzogen werden soll. Im Normalfall werden drei verschiedene Aktionsdateien verwendet:

- *default.action* enthält die Basiskonfiguration und sorgt für Privoxys recht gute Filterleistung, ohne das der Benutzer dafür einen Finger rühren muss.
- *standard.action* ist für Privoxys internen Gebrauch bestimmt und legt die Aggressivitätsstufen fest, die als Grundeinstellungen benutzt werden können. Die Auswahl der Aggressivitätsstufe selbst erfolgt über Privoxys Webinterface, *standard.action* definiert nur die Stufen selbst und muss vom Nutzer nicht verändert werden.
- Als letzte Aktionsdatei kommt *user.action* ins Spiel, dort werden die vom Benutzer gewünschten Änderungen vorgenommen. *user.action* kann zwar mit jedem Texteditor angepasst werden, um Fehler zu vermeiden ist jedoch auch hier die Benutzung des Webinterfaces ratsam.

Um die Übersicht zu erhöhen können beliebige weitere *.action-Dateien angelegt werden, nötig ist das aber nicht.

Filter-Dateien: *default.filter* und *user.filter*

Die Datei *default.filter* enthält die offiziellen Privoxy-Filter, die von den Privoxy-Entwicklern bereitgestellt werden. Ein Filter ist eine Methode um den Inhalt der aufgerufenen Seite zu verändern, ab Privoxy 3.0.5 beta können auch die Header mit Privoxy-Filtern verändert werden.

Bevor ein Filter in einer der Aktionsdateien verwendet werden kann, muss er in einer Filter-Datei definiert sein.

In Privoxy können Filter-Dateien noch nicht über das Privoxy-Interface verändert werden, jede Änderung erfolgt von Hand mit Hilfe eines Texteditors.

Seit Privoxy 3.0.5 beta werden mehrere Filter-Dateien unterstützt und es ist ratsam, für Eigenkreationen eine getrennte Filter-Datei zu nutzen. Üblicherweise wird sie *user.filter* genannt, in Privoxys Konfigurations-Datei wird dazu die Zeile: "filterfile *user.filter*" auskommentiert. Falls die Datei noch nicht existiert, muss sie vom Benutzer angelegt werden.

Konfiguration mit Privoxys Webinterface

Privoxys Webinterface wurde bereits beim Test der Privoxy-Installation angesprochen, der eigentliche Zweck liegt jedoch in der Konfiguration.

Um den Inhalt der Aktionsdateien anzuzeigen, ruft man im Browser die Adresse *p.p* auf und wählt im erscheinenden Menü die Option "View & change the current configuration" aus. *p.p* ist der kürzeste Weg zum Privoxy-Interface, die Langform ist die weiter oben benutzte Adresse <http://config.privoxy.org/>.

Wie dem Namen zu entnehmen, erlaubt "View & change the current configuration" sowohl das Ändern der Konfiguration, als auch die Kontrolle der Einstellungen. Bevor man hier irgend etwas verändert, sollte man sich vergewissern, dass die Einstellungen korrekt sowie vollständig sind und Privoxy die drei oben beschriebenen Aktionsdateien und eventuell ihren Pfad auflistet.

Falls eine der Aktionsdateien fehlt, muss die Privoxy-Standard-Konfigurationsdatei entsprechend angepasst werden. Die Datei ist gut dokumentiert, Änderungen beschränken sich meist auf das Auskommentieren einer Zeile, also dem Entfernen eines Sharp-Zeichens (#).

Privoxy überprüft seine Konfigurationsdateien bei jeder Anfrage auf Änderungen, es ist daher nicht nötig, Privoxy neu zu starten oder ein SIGHUP zu schicken.

Zeigt Privoxy die Nutzung aller drei Aktionsdateien an, kann mit der Konfiguration begonnen werden. Dies geschieht, wenig überraschend, durch Benutzung des Edit-Buttons. Der View-Button zeigt die Datei in Textform an, erlaubt aber keine Änderungen. Da *standard.action* nicht verändert werden sollte, fehlt dort der Edit-Button – die Anzeige ist aber möglich.

Als erstes ist in *default.action* die Standard-Aggressivität einzustellen. Sie gilt für alle Seiten, kann aber durch Ausnahmeregeln umgangen werden. Es kann eine von drei Stufen gewählt werden: "Cautious" (Vorsichtig), "Medium" (Mittel) und "Adventuresome" (Privoxy 3.0.3) oder "Advanced" (Privoxy 3.0.5 beta).

Je aggressiver die Grundeinstellung, desto mehr Werbung und Spionage-Blödsinn wird unterdrückt, es häufen sich jedoch auch die unbeabsichtigten Effekte und es kann passieren, dass manche Seiten nicht mehr nutzbar sind. Für den Anfang ist es sicher nicht verkehrt, erste Erfahrungen mit "Cautious" zu sammeln und die Stufe zu erhöhen, wenn man mit der Privoxy-Bedienung etwas besser vertraut ist.

Die Konfiguration von *default.action* ist damit abgeschlossen, weitere Schritte sollten in *user.action* vorgenommen werden, wozu ein wenig Theorie nötig ist.

Grundprinzip der Aktionsdateien *default.action* und *user.action*

Aktionsdateien sind in Sektionen (sections) aufgeteilt, die sich wiederum aus Aktionen (actions) und URL-Mustern (URL patterns) zusammensetzen. Aktionen wurden bereits angesprochen, sie legen fest, was Privoxy mit einer Seite machen soll. URL-Muster geben dagegen an, auf *welche* Seite die Aktionen angewandt werden sollen.

Sie heißen URL-Muster und nicht einfach nur URLs, weil sie durch reguläre Ausdrücke angegeben werden können.

Reguläre Ausdrücke erlauben es, mehrere URLs mit einem Muster abzudecken. Es wäre müßig jede Werbe-URL einzeln zu behandeln, daher überprüft Privoxy einfach ob eine URL Zeichenfolgen wie "Werbung" oder "banner" enthält. Am Anfang muss man sich noch nicht mit regulären Ausdrücken befassen.

Sektionen dienen dazu, Aktionen und Ziele zu bündeln. Es können sowohl mehrere Aktionen als auch mehrere URLs auf einmal festgelegt werden. Das ist praktisch, wenn die gleiche Aktionsfolge für mehrere Seiten gelten soll. "Abteilung" wäre eine elegantere Übersetzung, "Sektion" erfordert vom Leser jedoch weniger Denkaufwand und wird in dieser Anleitung daher bevorzugt.

Verschiedene Sektionen dürfen sich widersprechen, denn Privoxy wertet alle Sektionen nacheinander aus. Bei Widersprüchen gelten die Werte der letzten Sektion. Das ist kein Bug sondern ein Feature. Dadurch ist es möglich, erst eine Reihe von Aktionen für eine große Zahl von Zielen vorzunehmen und später Ausnahmen festzulegen.

Das Prinzip ähnelt Paket-Filter-Regeln, es ist jedoch zu beachten, dass viele Paket-Filter bereits die erste passende Regel befolgen und die Bearbeitung abbrechen. Trifft keine spezielle Regel zu, wird die Standard-Regel angewandt, die am Ende der Regelliste steht. Privoxy geht umgekehrt vor: die Standard-Regel wird zuerst gelesen, anschließend wird sie von den folgenden Regeln abgeändert.

Einfaches Beispiel: Unterdrückung von Iframe-Werbung

Iframes sind Unterseiten, die in eine andere Seite eingebettet werden. Die Unterseite ist ein eigenes Dokument.

Iframes werden hauptsächlich zur Einblendung von Werbung in Textform benutzt, können aber auch Banner, Videos oder Flash-Animationen enthalten. Vom Aussehen her sind sie leicht mit *normaler* Textwerbung – also Werbung die im gleichen Dokument wie der Nutzinhalt steckt – zu verwechseln.

Ein Beispiel für *normale* Textwerbung sind [Googles Sponsored Links](#).

Für den Privoxy-Benutzer hat Iframe-Werbung einen großen Vorteil: da der Browser sie von Privoxy gesondert anfordert, kann Privoxy einfach die Lieferung verweigern.

Normale Textwerbung kann nicht einfach verweigert werden, ein Block bezieht sich immer auf das ganze Dokument, der Benutzer würde auch den eigentlichen Inhalt nicht zu sehen bekommen. Textwerbung kann man durch einen anderen Text oder durch Leerzeichen ersetzen, muss dafür aber einen eigenen Filter schreiben. Dazu später mehr.

Blockieren kann Privoxy bereits von selbst, es ist nur noch das Ziel anzugeben.

Als Beispiel dient heise.de, dort verwendet man Iframe-Werbung, die vom Server "contentsearch.de.espotting.com" abgerufen wird. Dies lässt sich im Browser feststellen: bei Mozilla Firefox rechtsklickt man dazu in den Bereich des Iframes und wählt "This Frame/View Page Info" aus, um sich die Adresse anzeigen zu lassen.

Bevor man zum Blockieren nun eine neue Sektion anlegt, sollte man erstmal überprüfen ob eventuell schon eine vorhanden ist. *user.action* enthält bereits eine Sektion, der nur eine einzige Block-Aktion zugeordnet ist. Wenn Privoxy frisch installiert ist, ist schon ein Beispiel angegeben "www.example.com/nasty-ads/sponsor.gif", dies kann über den nebenstehenden Edit-Button durch den Espotting-Server ersetzt werden.

Espotting ist eine reine Werbe-Firma, daher ist es unwahrscheinlich, dass man jemals von deren Servern irgend etwas empfangen wollen wird. Statt nur "contentsearch.de.espotting.com" zu blockieren, kann ein *noch* allgemeineres Muster gewählt werden.

Richtig gelesen, auch "contentsearch.de.espotting.com" ist bereits ein – wenn auch sehr einfach zu verstehendes – Muster. Es umfasst jeden URL, der die Zeichenfolge "contentsearch.de.espotting.com" enthält.

Je kleiner man das Muster macht, desto allgemeiner wird es. Zur Erinnerung: das kürzeste Zeichen ist ".". Es umfasst einfach alles. Einfach alles zu blocken ist keine gute Idee, ein Kompromiss zwischen hoher Trefferquote und wenig Fehlern ist in diesem Fall ".espotting.com/". Damit berücksichtigt man alle Subdomains (alles vor dem ersten Punkt) mit allen Unterseiten (alles hinter dem Slash).

Sobald der Browser nun versucht, eine Esportting.com-Seite aufzurufen, blockiert Privoxy den Aufruf und liefert stattdessen eine Nachricht darüber. Der Benutzer kann nun über einen eingeblendeten Spezial-Link die Seite trotzdem abrufen, sich von Privoxy die für die Seite zuständigen Aktionen anzeigen lassen, oder einfach nur die Werbeunterdrückung befriedigt zur Kenntnis nehmen.

Komplexeres Beispiel: Unterdrückung von Googles Sponsored Links (veraltet)

Im Folgenden geht es um die Erstellung und Anpassung eines Privoxy-Filters. Der beschriebene Filter ist veraltet und dient nur als Beispiel. Leser denen es nur um das Endergebnis, die Werbeunterdrückung bei Google, geht, können dazu einfach den Filter "google" aktivieren, der seit Privoxy 3.0.6 mitinstalliert wird, aber standardmäßig deaktiviert ist.

Nun zu einem etwas komplexeren Beispiel, der Entfernung von Googles Sponsored Links. Da die Sponsored Links zusammen mit den Suchergebnissen in einer Seite geliefert werden, kann man sie nicht einfach blockieren (ich wiederhole mich). Sie müssen mit einem eigenen Filter behandelt werden.

Um einen Filter zu erstellen, muss man wissen, was man überhaupt filtern möchte. Der Quelltext der Seite hilft weiter. Bei Google.com sind die Sponsored Links wie folgt aufgebaut:

```
<td height=25 align=center><font color=#6f6f6f size=-1>Sponsored Links</font></td></tr><tr
height=7><td></td></tr><tr><td nowrap><font size=-1><a id=aw1 href=/url?q=http://example.org/
onMouseOver="return ss('go to example.org')"
onMouseOut="cs()"><b>Werbesite</b></a><br>Werbetext durch ein<br>unterbrochen<br><font
color=green>www.example.org</font><br><br></font></td></tr><tr height=7><td></td></tr><tr><td
height=25 align=center><font size=-1><a
href=http://www.google.com/url?q=http%3a%2f%2fadwords.google.com%2fmehrparameter class=fl>See
your message here...</a></font>
```

Um dieses Ungetüm zu entfernen muss man es verallgemeinern, wie im folgenden Filter geschehen:

```
#####
# filter google adwords #
#####
FILTER: google_adwords Remove top and rightside text ads

s@Sponsored Links@@Ug
s@See your message here...@U
s@<td( class=ch)? id=(taw|tpa|spa)\d.*</td>@<td></td>@Ug
s@<a id=aw\d.*</font></td>@</font></td>@Ug
```

Dieser Filter stammt von Jan Willamowius. Bevor man auch nur darüber nachdenkt, einen eigenen Filter zu erstellen, lohnt es zu schauen, ob man nicht irgendwo kopieren kann.

Vor dem Einsatz eines fremden Filter sollte er dennoch verstanden worden sein. Daher hier eine kurze Erklärung des Aufbaus:

Die ersten drei Zeilen sind ein Kommentar, der den Filter von anderen abtrennt. Sie dienen nur der Übersichtlichkeit in der Filterdatei, auf den Filter selbst haben sie keinen Einfluss.

"FILTER:" ist für Privoxy der Beginn eines neuen Filters (Wer hätte das gedacht?). Die Großbuchstaben sind zwingend, "Filter:" wird nicht akzeptiert. Das erste dem Schlüsselwort "FILTER:" folgende Wort gibt den Filternamen an. Dieser Filter heißt "google_adwords". Im Filternamen sind keine Leerzeichen erlaubt, daher die Verwendung des Unterstrichs.

Alles, was nach dem Filternamen in der gleichen Zeile kommt, wird als Beschreibung des Filters interpretiert. Hier: "Remove top and rightside text ads". Die Filterbeschreibung wird später von Privoxy im Webinterface angezeigt und hilft dem Benutzer beim Finden eines passenden Filters. Die Angabe ist freiwillig aber empfehlenswert.

Die Leerzeile ist – wie der Kommentar – optional.

Die Zeile "s@Sponsored Links@@Ug" enthält die erste Filteraktion. Das erste "s" gibt die Filterart an: Suchen und ersetzen. Die @-Zeichen dienen als Trenner zwischen dem zu suchenden Text und dem Ersatz. Da die letzten beiden @-Symbole ohne Zwischenraum sind, wird kein Ersatz eingefügt. Anders ausgedrückt: der Suchtext wird in der Seite gelöscht.

Das "U" ist eine Privoxy-interne Syntax-Erweiterung und steht für "Ungreedy" (nicht gefräßig/gierig) ist hier ohne Nutzen; schaden tut es allerdings auch nicht (Erklärung dazu kommt später). Das "g" steht für "global" und gibt an, dass das Suchen und Ersetzen so oft wie möglich durchgeführt werden soll. Ohne das "g" würde die Aktion maximal einmal durchgeführt werden.

"s@Sponsored Links@@Ug" bedeutet somit: alle Vorkommnisse von "Sponsored Links" ersatzlos streichen.

Die Bedeutung von "s@See your message here...@@U" ist analog: "See your message here..." streichen wenn vorhanden. Da keine g-Option angegeben ist, wird die Funktion nur einmal vorgenommen. Mehr ist nicht nötig.

"s@<td (class=ch)? id=(taw|tpa|spa)d.*</td>@<td></td>@Ug" ist schon etwas umfangreicher. Neuerungen mit Sonderfunktionen sind: runde Klammern, ein Fragezeichen, Pipe-Symbole (Hochstriche), die Zeichenfolge "d", Punkte und ein Asteriskus (Sternchen).

Die runden Klammern kennzeichnen die Zeichenfolge "class=c" als zusammengehörig. Das Fragezeichen gehört dazu, es ist ein Quantifizierer und bedeutet: die vorherige Zeichengruppe soll ein- oder keinmal vorkommen. Ohne die runden Klammern würde das Fragezeichen nur für das letzte "h" gelten.

Die nächsten runden Klammern kommen ohne Fragezeichen daher, der Inhalt soll genau einmal vorhanden sein – nicht mehr und nicht weniger. Die Pipe-Symbole stehen für eine Auswahl, "(taw|tpa|spa)" deckt daher die Zeichenfolgen "taw", "tpa" oder "spa" ab. Eine davon muss vorhanden sein, oder das ganze Muster passt nicht.

"\d" ist eine Zeichenklasse und steht für eine beliebige Ziffer.

Der Punkt steht für ein beliebiges Zeichen (Ausnahmen wie Zeilenumbruch nicht berücksichtigt). Der dem Punkt folgende Asteriskus gibt – wie das Fragezeichen – die Anzahl des vorherigen Zeichens oder der Zeichengruppe an. Es steht standardmäßig für: so oft wie möglich. Es ist ein gieriges (greedy) Zeichen und kriegt nie genug. Der Appetit wird dem Sternchen durch das "U" abgewöhnt. Die Bedeutung ist daher: so oft wie nötig, bis das nächste Teilmuster passt.

Die restlichen Zeichen zwischen den @-Trennzeichen stehen nur für sich selbst und müssen jeweils einmal vorkommen.

Zusammenfassung: "s@<td (class=ch)? id=(taw|tpa|spa)d.*</td>@<td></td>@Ug" sagt Privoxy: Ersetze alle Zeichenfolgen "<td", denen eine oder keine Zeichengruppe "class=ch" folgt, die vor der Zeichenfolge "id=", wiederum gefolgt von entweder "taw", "tpa" oder "spa" steht, auf die eine beliebige Ziffer folgt, deren Position vor einer möglichst kleinen Anzahl von beliebigen Zeichen ist, die wiederum vor den Zeichen "</td>" stehen, mit "<td></td>". Vereinfacht: Eine bestimmte Tabellenzelle wird samt Inhalt entfernt, eine leere Ersatz-Zelle eingefügt.

Manche Leute bezeichnen die Syntax von Perls regulären Ausdrücken auch als "lesegeschützt" – warum auch immer.

Der Ausdruck "s@<a id=aw\d.*</td>@</td>@Ug" enthält keine Neuerungen. Jede Zeichenkette die sich aus "<a id=aw", einer Ziffer sowie einer möglichst kleinen Anzahl beliebiger Zeichen, denen "</td>" folgt, zusammensetzt, wird durch "</td>" ersetzt.

Nachdem man den Filter verstanden hat, steht seinem Einsatz nichts mehr im Weg. Er wird in "default.filter" (oder wenn möglich "user.filter") rein kopiert und ist anschließend als Aktion in den Aktionsdateien benutzbar.

Da der Filter nur für Google gelten soll, legt man über Privoxy's Webinterface in `user.action` eine neue Sektion an. Dafür könnte man den Button "Insert new section at top" benutzen, es ist jedoch guter Stil, derartige Sektionen ans Ende zu packen. So wird verhindert, dass später kommende Sektionen die Aktion überschreiben.

Beim Google-Beispiel ist zwar unwahrscheinlich, dass eine spätere Sektion explizit festlegt, dass der Adwords-Filter nicht angewandt werden soll. Doch mit der Regel: "allgemeine Sektionen nach vorne, spezielle Sektionen nach hinten", macht man nichts verkehrt. Befolgen der Regel hilft zudem beim späteren Wiederfinden der Sektionen.

Um die neue Sektion am Ende von "user.action" anzulegen, scrollt man bis zum Ende der Seite und benutzt den "Insert new section below"-Button. Mit dem Edit-Button wird nun die gewünschte Aktion ausgewählt. Er führt zu einer Liste, die alle verfügbaren Aktionen anzeigt.

Neben jeder Aktion gibt es drei Radio-Buttons, die für "Enable", "Disable" und "No Change" stehen. Als Anfangswert ist "No Change" aktiviert, was bedeutet, dass die Sektion die betreffende Aktion nicht beeinflusst. Es gelten die Einstellungen, die von vorherigen Sektionen vorgenommen wurden. "Disable" deaktiviert eine Aktion und überschreibt damit jede vorhergehende Sektion, die die gleiche Aktion beeinflusst. "Enable" aktiviert eine Aktion, auch sie überschreibt jede vorhergehende Sektion.

Der Filter "google_adwords" ist folglich auf "Enable" zu setzen. Es ist die einzige Aktion, die in der Google-Sektion nötig ist. Der "Submit"-Button führt zum vorherigen Menü zurück und speichert die vorgenommene Einstellung.

Der Sektion fehlt nun noch das URL-Muster. Eine Möglichkeit wäre ".google.", es könnte allerdings schon zu grob sein. Alle gewünschten Google-Domains einzutragen ist ein brauchbarer Kompromiss. Zum Beispiel ".google.de/", ".google.com/" und ".google.net/".

Die Idee, das Ganze durch eine Bündelung als ".google.(net|de|com)/" auszudrücken, ist nachzuvollziehen, funktioniert allerdings nicht. Privoxy erlaubt keine Perl-Syntax in der Domain, sondern erwartet dort Shell-Syntax, die weniger rechenintensiv, aber deutlich leistungsschwächer ist. Der Pfad wiederum wird mit Perl-Syntax beschrieben. Am Anfang kann das verwirren – später auch.

Filteranpassung

Wer den Adwords-Filter nun auf www.google.de ausprobiert, wird feststellen, dass die Werbelinks zwar unterdrückt werden, jedoch "Anzeigen" und "Sehen Sie Ihre Anzeige hier..." stehen bleiben. Da der Filter für die englische Google-Version erstellt wurde, sollte das niemanden überraschen. Er muss noch angepasst werden.

Der Filter soll entweder "Sponsored Links" oder "Anzeigen" löschen, und entweder "See your message here..." oder "Sehen Sie Ihre Anzeige hier...". Das schreit nach einer Gruppierung mit Alternativen. Die beiden regulären Ausdrücke sind nach:

```
s@(Sponsored Links|Anzeigen)@@Ug
s@(See your message here|Sehen Sie Ihre Anzeige hier)...@U
```

zu ändern. Auch die deutsche Google-Seite wird nun gesäubert.

Der Filter hat jedoch eine weitere Schwäche: er ist zu allgemein, jedes Wort "Anzeigen" wird gelöscht. Viele wird es nicht stören, wer jedoch nach "Anzeigen" sucht, hat ein Problem.

Wenn der Verdacht aufkommt, ein Filter würde zu viel löschen, kann man es leicht überprüfen, indem man ihn etwas anpasst. Statt nur zu löschen, lässt man ihn ein seltenes Wort einfügen. Ändert man "s@(Sponsored Links|Anzeigen)@@Ug" nach "s@(Sponsored Links|Anzeigen)@Von Privoxy gefressen@Ug" und führt die Suche nach "Anzeigen" erneut durch, so sieht man sofort, dass Privoxy zu viel ersetzt. Der Filter muss empfindlicher werden.

Zu erst kann man das "g" entfernen, der Filter wird dadurch effizienter und einmal unterdrücken reicht. Die gesteigerte Effizienz wird man nicht wahrnehmen, wohl aber, dass der Filter nur noch einmal tätig wird. Momentan leider an der falschen Stelle, nämlich dort wo das Wort "Anzeigen" zuerst vorkommt.

Damit der Filter nur noch an der gewünschten Stelle eingreift, muss die Zeichenfolge nach der gesucht wird vergrößert werden. Schaut man sich die Stelle in Googles Quelltext an, sieht man, dass hinter "Werbung" ein schließender Font-Tag, ein schließender TD-Tag und ein schließender TR-Tag stehen. Davor steht ein öffnender Font-Tag. Als Spezialisierung sollte es ausreichen. Neuer Filter: "s@size=-1>(Sponsored Links|Anzeigen)</td></tr>@size=-1>Von Privoxy gefressen</td></tr>@U"

Ein Test zeigt, dass "Von Privoxy gefressen" an der gewünschten Stelle auftaucht, die *Debug*-Zeichenfolge kann nun wieder gelöscht werden.

Werden HTML-Elemente im Suchmuster benutzt, so ist es wichtig, sie auch im Ersatztext anzugeben. Es kann sonst zu unerwünschten Effekten kommen. Im oberen Filter fängt das Suchmuster mit dem Ende eines Font-Tags an, würde man es nicht im Ersatztext ebenfalls angeben, hätte der Browser ein Problem beim Rendern der Seite. Auch die drei schließenden Tags am Ende des Suchmusters würde er vermissen.

Wer keine Lust hat, sich mit der Funktion verschiedener Tags auseinander zu setzen, sollte sicherstellen, keine zu entfernen.

Der Filter arbeitet jetzt korrekt, wenn man nach "Anzeigen" sucht, macht aber noch Probleme wenn nach "Sehen Sie Ihre Anzeige hier" gesucht wird. Passiert wahrscheinlich nicht so oft, als Übung dennoch ein Lösungsvorschlag: "s@class=fl>(See your message here|Sehen Sie Ihre Anzeige hier)...@class=fl>Von Privoxy gefressen@U". Nach einem kurzen Test scheint auch das zu funktionieren, der Ersatztext kann bis auf die Tags gekürzt werden.

Nach der Veränderung sieht der Adwords-Filter nun so aus:

```
#####
# Adwords-Filter #
# Basiert auf der Arbeit von Jan Willamowius #
# http://www.willamowius.de/privoxy.html #
#####
FILTER: google_adwords Entfernt Googles Adword-Werbung. Hinterlässt blauen Balken zur Kontrolle
s@size=-1>(Sponsored Links|Anzeigen)</font></td></tr>@size=-1></font></td></tr>@
s@class=fl>(See your message here|Sehen Sie Ihre Anzeige hier)...</a></font>@class=fl></a></font>@
s<td( class=ch)? id=(taw|tpa|spa)\d.*</td>@<td></td>@Ug
s<a id=aw\d.*</font></td>@</font></td>@Ug
```

Erläuterung: Der Kommentar wurde angepasst, um den Urheber zu würdigen. Die Filterbeschreibung wurde geändert, der blaue Balken – der vom Filter nicht erfasst wird – gilt jetzt als Feature. Da er kaum auffällt lohnt es nicht, bei seiner Beseitigung Zeit zu verschwenden. Als Anzeiger, wenn für eine Suche Adwords gekauft wurden, hat er eine neue Bestimmung bekommen.

Die ersten beiden Filterzeilen enthalten keinen gierigen Asteriskus, die "U"s wurden daher entfernt. Da die erste Filterzeilen nur einmal zum Einsatz kommen soll, wurde auch der Globalisierer gestrichen.

Die dritte Filterzeile bleibt unverändert.

Die vierte Filterzeile enthält einen Asteriskus, das "U" bleibt, das "g" wurde jedoch entfernt.

Noch ein Hinweis zu den drei Punkten in der zweiten Filterzeile: Sie stehen nicht einfach nur für drei Punkte, sondern für drei beliebige Zeichen. Will man die Sonderfunktion ausschließen, muss man ihnen einen Backslash vorstellen. Der Filter ist auch so speziell genug, es wurde hier darauf verzichtet, da es auf Kosten der Lesbarkeit geht. Der Bedeutung von Sonderzeichen muss man sich bewusst sein, sie sind eine häufige Fehlerquelle.

Es wurde erst an der Oberfläche von Privoxy Filtermöglichkeiten gekratzt, für weniger anspruchsvolle Filter und um das Prinzip zu zeigen, sollte es jedoch ausreichen. Wer *kompliziertere* Filter bauen möchte, besorgt sich am besten ein Buch über perlkompatible reguläre Ausdrücke.

[Privoxy-Filter-Test](#) beschleunigt die Erstellung und den Test von Privoxy-Filtern deutlich, Installation und Einrichtung sind allerdings nicht trivial und die (unvollständige) Dokumentation nur in Englisch verfügbar.

Filtern für Faule: Werbeunterdrückung mit CSS

Das Ersetzen von Seiteninhalten ist nicht die einzige Möglichkeit das Erscheinungsbild einer Website zu verändern: mit Hilfe von CSS kann man sich die Arbeit deutlich erleichtern. Statt jedes unerwünschte Seiten-Element einzeln zu bearbeiten werden dabei einfach ein paar neue CSS-Anweisungen am Kopf der Seite eingefügt. Den Rest macht der Browser.

Dazu ein Beispiel aus der [Privoxys default.filter-Datei](#):

```
#####
#
# yahoo: CSS-based block for Yahoo text ads. Also removes a width limitation.
#
#####
FILTER: yahoo CSS-based block for Yahoo text ads. Also removes a width limitation.
s@</head>@<style type="text/css">\n
/* Style sheet inserted by Privoxy's yahoo filter. */\n
\#symadbn, \#ymadbn, \#yschsec, \#yschanswr, .yschftad,\
.yschspn, .yschspns {display: none !important;}\n
\#yschpri {width: 100% }\n</style>\n$0@
```

Der Filter kommt mit einer einzigen Regel aus, Privoxy soll der Zeichenkette "</head>" ein paar CSS-Anweisungen voranstellen. Die wiederum veranlassen den Browser dazu, Teile der Website verändert oder auch gar nicht anzuzeigen. [Eine ausführlichere Beschreibung folgt...](#)

Privoxy als *Schutz* der Privatsphäre

Auch wenn Privoxy keine vollständige Anonymität schafft, hilft der Proxy doch beim Vermeiden unnötiger Spuren. Viele Spionage-Server, die ihre Rechenzeit mit der Erstellung von Nutzungsprofilen verschwenden, werden bereits mit der Standard-Konfiguration geblockt.

Achtung: Privoxy hält den Browser (oder andere Programme) nicht davon ab, direkte Verbindungen aufzubauen. Im Normalfall respektieren Browser die Proxy-Einstellungen, eine Reihe von Browser-Plugins sind jedoch schlampig programmiert und ignorieren die Proxy-Einstellungen für eigene Abfragen. In solchen Fällen *sieht* Privoxy die Abfragen nicht und kann sie auch nicht filtern oder an Tor weiterleiten.

Privoxy und Tor

Die IP-Nummer des Privoxy-Benutzers bekommt der Server einer abgerufenen Webseite weiterhin zu sehen, ohne Hilfe eines Gerichts sind einer IP-Nummer im Normalfall jedoch nur Informationen über den Provider, nicht aber über den Surfer zu entlocken.

Wer seine Anonymität weiter steigern möchte, kann zur Einwahl ins Internet ein unregistriertes Prepaid-Mobil-Telefon beziehungsweise ein offenes W-LAN nutzen. Deutlich bequemer ist es, zusätzlich zu Privoxy das anonymisierende Proxy-Netzwerk Tor zu verwenden.

Nach der Tor-Installation muss dazu in Privoxys Konfigurationsdatei die Zeile:

```
forward-socks4 / 127.0.0.1:9050 .
```

ergänzt (Privoxy 3.0.3) oder auskommentiert (ab Privoxy 3.0.5) werden. Damit erreicht man Anonymität gegenüber dem Serverbetreiber, doch Privoxys DNS-Abfragen (die Wandlung von Domain-Namen in die zugehörigen IP-Nummer) sind weiterhin unverschlüsselt und für einen lokalen Lauscher mithörbar. Er erhält damit die Information mit welchen Servern Kontakt aufgenommen wird, weiß aber nicht, welche Seiten und welche Inhalte übermittelt werden.

Privoxy kann die DNS-Abfragen auch dem Tor-Netzwerk überlassen, in die Konfigurationsdatei schreibt man dazu:

```
forward-socks4a / 127.0.0.1:9050 .
```

Die Tor-Nutzung hat nicht nur Vorteile, sie bringt einen spürbaren Geschwindigkeitsverlust mit sich und beim Verbindungsaufbau über Tor kommt es gelegentlich zu Fehlern, in den meisten Fällen reicht es aber aus, die Anfrage im Browser einfach noch mal zu wiederholen.

Keinem Programm sollte vertraut werden, ohne das Prinzip dahinter verstanden zu haben. Vor der ersten Tor-Nutzung empfiehlt sich daher der Besuch der offiziellen Website des Tor-Projekts. *Kai Ravens Sicher und anonym im Internet mit Proxys* ist eine gute Ergänzung zu dieser Anleitung, besonders der Tor-Abschnitt ist deutlich ausführlicher.

Cookies

Cookies sind kurze Texte, die ein Server an den Browser schicken und wieder abrufen kann. Besucht ein Surfer eine Website erneut, kann er mit dem Cookie erkannt werden. Hat er beim letzten Besuch Adresse, Name oder Telefonnummer hinterlassen, können ihm auch diese Daten erneut zugeordnet werden.

Für Onlineshops ist das gelegentlich praktisch, der Benutzer kann sich das Einloggen sparen. Die Kosten sind jedoch die Aufgabe der (begrenzten) Anonymität, der Seitenbetreiber kann ein Nutzungsprofil über mehrere Sessions hinweg anlegen, selbst wenn der Surfer nur stöbert.

Die berechnete Cookie-Nutzung mit Vorteil für den Surfer ist im Verhältnis zur unberechtigten Nutzung verschwindend gering. Meist dient sie nur Spionage Zwecken, wer einen gewissen Wert auf Privatsphäre legt, sollte Cookies daher standardmäßig vernichten und nur in Ausnahmefällen zulassen.

Viele Onlineshop-Ersteller sind leider zu dumm, Session-IDs zu benutzen. Ohne Cookies gibt es dort für den Surfer keine Ware und für den Betreiber kein Geld. Wer unbedingt mit dummen Menschen Geschäfte machen möchte, muss Cookies für diese speziellen Server erlauben. Verzicht auf optionalen Session-IDs ist keine "vorübergehenden Störung der Geistestätigkeit", sondern eine andauernde. Mit BGB § 105 Nichtigkeit der Willenserklärung sind daher keine Probleme zu erwarten.

Kurze Unterbrechung der Ideologie und zurück zur Technik.

Cookies lassen sich von Privoxy komplett vernichten, oder in Session-Cookies ändern. Session-Cookies verfallen, wenn der Browser geschlossen wird. Wer Session-Cookies als Standard-Wert wählt, kann man auch bei den oben genannten defekten Shops bestellen, ohne jedesmal die Einstellungen zu ändern.

Sitzt der Surfer nicht hinter einem Proxy, sind ihm seine Seitenaufrufe eh über die IP-Nummer temporär zuordbar, durch Aktivierung von Session-Cookies wird dann keine weiter Beschneidung der Privatsphäre vorgenommen. Viele Leute sitzen allerdings hinter einem Proxy ohne es zu wissen, denn Firmen wie AOL verwenden transparente Proxys, Proxys die unabhängig von den Browsereinstellungen wirksam sind.

Cookies im Browser oder in Privoxy behandeln?

Anständige Browser ermöglichen die Cookie-Kontrolle von sich aus, Privoxy ermöglicht es jedoch, Einstellungen für mehrere Browser gleichzeitig vorzunehmen. Wird nur ein Browser benutzt, ist es egal, wo die Einstellungen vorgenommen werden. Ab zwei Browsern ist Privoxy jedoch die komfortablere Lösung.

Privoxy verfügt über folgende Aktionen die für Cookies relevant sind:

1. crunch-incoming-cookies
2. crunch-outgoing-cookies
3. session-cookies-only
4. send-vanilla-wafer
5. send-wafer
6. filter{content-cookies}

Mit "crunch-incoming-cookies" vernichtet man eingehende Cookies, bevor sie beim Browser ankommen. Mit "crunch-outgoing-cookies" vernichtet man vom Browser gesendete Cookies.

"session-cookies-only" löscht das *Halbbarkeitsdatum* des Cookies, der weiter oben genannte Session-Cookie entsteht. Wenn man in einer Sektion "session-cookies-only" aktiviert, sind gleichzeitig "crunch-incoming-cookies" und "crunch-outgoing-cookies" zu deaktivieren.

Um Missverständnissen vorzubeugen: die verschiedene Cookie-Einstellungen in verschiedenen Sektionen sind erlaubt. Wenn jedoch eine der beiden Optionen bereits in einer allgemein gehaltenen anderen Sektion aktiviert wurde, gilt sie auch für jede folgende Sektion, da ihre Einstellung dort auf "No Change" steht. "session-cookies-only" wäre dann wirkungslos, explizites Deaktivieren der beiden Session-Cruncher sorgt vor.

"send-vanilla-wafer" und "send-wafer" ermöglichen es, dem Cookie-Freund ins Logfile zu müllen – falls er eines anlegt. "send-vanilla-wafer" ist allgemein gehalten und bittet den Server darum, die Session nicht zu verfolgen; "send-wafer" erlaubt die Veränderung der Botschaft. Die Verwendung dieser beiden Aktionen ist **keine gute Idee**. Der Nutzen ist fraglich, und mit den Sondercookies kann man über mehrere Besuche hinweg verfolgt werden, da die Zahl der Privoxy-Benutzer zu klein sein dürfte, um im Rauschen unter zu gehen. Beide Optionen sind grundsätzlich aus – dabei sollte es bleiben.

"content-cookies" ist in der Filter-Region zu finden, da er im Seitentext (dem Content) arbeitet, anstatt auf HTTP-Ebene. Er entfernt Metatext- und JavaScript-Cookies (wenn sie erkannt werden).

Mit JavaScript ist es möglich, Privoxys JavaScript-Cookie-Filter zu umgehen. Über JavaScript können zudem einige Systemvariablen gelesen werden, aus denen eine den Surfer identifizierende Prüfsumme generiert werden kann.

Zum Schutz der Privatsphäre (und der Sicherheit allgemein) ist es sinnvoll, JavaScript nur für vertrauenswürdige Websites zu aktivieren und nur, wenn die Verbindung zwischen Website und Anwender verschlüsselt erfolgt.

JavaScript-Nervereien unterdrücken

Es gibt tausende verschiedene Methoden, HTML mit JavaScript zu erweitern. Die passenden regulären Ausdrücke sind daher komplex und fehleranfällig. Privoxys JavaScript-Filter sind in Version 3.0.3 etwas zu aggressiv und ersetzen auch schon mal anderen Text, der wie mögliches JavaScript aussieht. `open()` wird zum Beispiel generell nach `PrivoxyWindowOpen()` umgeschrieben. Wer regelmäßig C-, Java- oder Perl-Code im Browser betrachtet, wird sich dadurch gestört fühlen.

Die Situation hat sich in späteren Privoxy-Versionen nur geringfügig verbessert, Privoxys JavaScript-Filter basieren nach wie vor auf regulären Ausdrücken und wenn ein Seitenbetreiber es darauf anlegt, kann er sie problemlos umgehen, indem er den eigentlichen Code erst im Browser zusammensetzen lässt.

Mittlerweile verfügt nahezu jeder Browser über eigene JavaScript-Optionen, Privoxys eigene JavaScript-Filter kann man daher getrost ignorieren, in keinem Fall sollte man sich auf sie verlassen.

Referrer fälschen oder (bedingt) entfernen

Die meisten Browser schicken – wenn man einem Link folgt – die Adresse der verlinkenden Seite mit.

"Referrer" und "Referer" sind beides gebräuchliche Bezeichnungen für diese Adresse, die erste Schreibweise ist korrektes Englisch, die zweite stimmt mit der Schreibweise des HTTP-Headers überein.

Defekte Browser schicken den Referrer-Header grundsätzlich mit, selbst wenn der Benutzer einem Bookmark folgte, die neue Adresse direkt eingegeben hat, oder einen Link in einer E-Mail benutzte.

Der Referrer ist für den Server unwichtig, er kann jedoch Informationen über den Surfer enthalten. Beispielsweise seine E-Mail-Adresse – manche Webmail-Interfaces bauen sie im URL ein – oder eine noch gültige Session-ID, mit der die alte Session des Surfer fortgeführt werden kann (eher unwahrscheinlich, da die Session nur eine bestimmte Zeit lang gilt).

Wird der Referrer von einer Website ausgewertet, kann man generell davon ausgehen, dass es nicht zum Vorteil des Surfer geschieht. Im besten Fall hat der Surfer keinen Nachteil.

Privoxy ermöglicht es, den Referrer zu grundsätzlich zu unterdrücken – manche Seiten die ihn auswerten funktionieren dann aber nicht mehr. Die [Download-Seite von Doom9.org](#) ist so ein Fall, der Referrer wird dort benutzt, um Leute auszuschließen, die direkten Links von fremden Seiten folgen, die auf von Doom9.org gehostete Programme zeigen.

Als Surfer kann man den Versuch nachvollziehen: fremde Bandbreite als eigene auszugeben ist schlechter Stil. Der Referrer-Check schließt jedoch auch die berechtigten Surfer von der Nutzung aus, die keinen Referrer mitsenden.

Als Lösung bietet Privoxy gefälschte Referrer. Dabei wird das Wurzelverzeichnis des Servers als Ursprungsadresse angegeben. Die entsprechende Privoxy-Aktion heißt "hide-referrer", als Option wird im Webinterface "Fake as the root directory of the site" aktiviert.

Referrer-Fälschung erschwert Server-Betreibern das Nachspionieren, kann vom Server-Betreiber aber bemerkt werden. Wenn alle Anfragen von der Homepage zu kommen scheinen ist das verdächtig: auf den meisten Websites sind nicht alle Dokumente von der Homepage aus verlinkt.

Um auch dieses Problem zu vermeiden, kann man Privoxy um die [hide-referrer-Option conditional-block](#) erweitern. [Seit Privoxy 3.0.5 beta ist der Patch bereits integriert.](#)

Anschließend wird der Referrer-Header innerhalb des gleichen Hosts ungefiltert durchgelassen und nur dann entfernt, wenn die alte und neue Seite von einem unterschiedlichen Host geliefert werden. Für den Server-Betreiber sieht es so aus, als ob der Surfer eine gewöhnliche Browser-Konfiguration hätte und durch Direkteingabe der Adresse auf die Website gelangt wäre.

Browser und Betriebssystem verbergen

In den Standard-Einstellungen senden Browser einige Informationen über die Umgebung in der sie laufen an den Webserver. Beispiel: "Mozilla/5.0 (X11; U; FreeBSD i386; en-US; rv:1.7) Gecko/20040707 Firefox/0.9.1". Diese Browser-Kennung heißt User-Agent, für den Server ist die Information unwichtig, anständige Websites funktionieren in jedem Browser.

Die Übermittlung des User-Agents ist nicht zwangsweise ein Eingriff in die Privatsphäre des Surfers, kann unter Umständen aber genutzt werden, um den Surfer wieder zu erkennen. Wer seinen Browser fertigkompiliert runterlädt und

unter Windows nutzt, geht im Rauschen unter. Wer jedoch selbst kompiliert und ein weniger weit verbreitetes Betriebssystem nutzt, fällt auf.

Viele Browser übermitteln nicht nur Version und Betriebssystem an den Server, sondern auch das Datum des Kompilierens. Manch User-Agent wird damit zum Unikat: das Verhalten des Surfers kann auch ohne Cookies nachverfolgt werden.

Da der überwiegende Teil der Surfer Cookies akzeptiert, wird sich kaum ein Server-Betreiber die Mühe machen. Doch besser ist es, sicher zu gehen. Privoxy kann dazu den User-Agent nach Belieben des Benutzers verändern.

Wer nicht an der Browserkennung erkannt werden, sich aber auch nicht als Windows-Nutzer ausgeben möchte, sollte den User-Agent regelmäßig ändern. Sinnvollerweise mit jedem IP-Wechsel.

Manuelles Ändern der Konfiguration ist nicht nur lästig, sie wird auch *gelegentlich* an der Bequemlichkeit des Privoxy-Benutzers scheitern. Ich benutze daher das [Perl-Skript `uagen.pl`](#), das den von Privoxy gesendeten User-Agent regelmäßig ändert und dabei Betriebssystem, Plattform, Datum sowie Lokalisierung variiert.

Risiken bei HTTPS-Verbindungen

Verschlüsselte Verbindungen kann Privoxy nur weiterleiten, Filter können auf sie nicht angewendet werden, da der Inhalt Privoxy nicht im Klartext passiert.

Um Privoxy zu umgehen könnte eine unverschlüsselte Webseite Bilder oder CSS-Dateien über eine verschlüsselte Verbindung einbinden, verschlüsselte Verbindungen sollten daher mit der Aktion `+limit-connect{ , }` generell gesperrt und nur für vertrauenswürdige Websites freigeschaltet werden.

Solange man bei einer Website nicht absichtlich persönlichen Daten hinterlässt, etwa um einen Online-Shop zu nutzen, gibt es keinen Grund HTTPS einzusetzen. Die Verbindung wurde bereits über Tor anonymisiert und HTTPS wäre dazu auch gar nicht in der Lage, da Start- und End-Punkt der Verbindung im Klartext gesendet werden.

Wenn man einer Website weit genug vertraut persönliche Daten zu hinterlassen, gibt es wenig wovon Privoxy noch schützen könnte: im Vergleich zu Kontodaten oder Passwörtern sind Cookies und Spionage-Pixel nicht der Rede wert.

Privoxy-Probleme

Früher oder später können beim Privoxy-Einsatz folgende Probleme auffallen:

Seitenaufbau springt anstatt zu fließen

Wird einer der Privoxy-Filter benutzt, so kommt es zu einer Verzögerung in der Darstellung. Die Bearbeitung der Filter selbst dauert nur Sekundenbruchteile, der Grund für die Verzögerung liegt in der Arbeitsweise. Bevor der Mustervergleich auf einer Seite vorgenommen werden kann, muss sie komplett geladen werden.

Die Darstellung ist sprunghaft: der Benutzer sieht erst mal nichts, dann plötzlich die ganze Seite (ohne eventuelle Bilder).

Ohne Privoxy fängt der Browser dagegen schon mit dem Rendern der Seite an, sobald er die ersten Pakete empfangen hat. In dem Moment, wo der Browser in der Seite Verweise auf Bilder findet, baut er parallele Verbindungen auf, um auch sie möglichst schnell darstellen zu können. Der Benutzer empfindet den Vorgang als flüssig und schnell.

Bei einer ausreichend schnellen Verbindung, einem zügig reagierenden Server und einer durchschnittlichen Seitengröße wird sich der Benutzer nicht gestört fühlen. Quälen sich jedoch mehrere hundert Kilobyte große Seiten über ein Modem, wird die Geduld überstrapaziert.

Für regelmäßig betrachtete Seitenmonster lohnt sich daher eine eigene Sektion, in der alle Filter deaktiviert werden. Zum Erkennen von Werbung kann Privoxy dann zwar nicht mehr die im Quelltext angegebene Grafikgröße benutzen, durch den Bilderpfad verraten sich dennoch einige Banner und mit einer entsprechenden Anpassung sollte sich auch der letzte Werbetupfer wieder ausradieren lassen.

Kein Webinterface-Zugriff auf die Konfigurationsdateien

Die Fehlermeldung:

Privoxy Configuration access denied

The feature you are trying to access has either been disabled by the Privoxy administrator, or you came here by following an unsafe external link.

ist meist ein Zeichen dafür, dass der Benutzer den Browser angewiesen hat, generell keinen Referer mehr zu setzen. Wie bereits im [Referer-Abschnitt](#) erläutert, ist das keine gute Idee, da es unnötig für Aufmerksamkeit sorgt und keine Vorteile gegenüber einem bedingten Block hat.

Privoxy lässt Änderungen nur über *sichere Links zu*, fremde Website könnten sonst über versteckt eingebundene Bilder die Privoxy-Konfiguration verändern. Falls Privoxys Webinterface nicht im Referer auftaucht – also auch wenn der Referer gar nicht gesetzt wurde – enden Zugriffe auf die Konfigurations-Dateien mit obiger Fehlermeldung.

Um das Problem zu lösen kann man die `Referer`-Filterung Privoxy überlassen und den `Referer`-Header im Browser wieder aktivieren.

Weitere erwähnenswerte Privoxy-Funktionen

Diese Anleitung ist nicht nur oberflächlich, viele Privoxy-Funktionen wurden komplett übergangen. Teilweise weil ich sie nicht benutze, teilweise weil sie nur für eine Minderheit der Benutzer interessant sind.

Das [englische Privoxy-Handbuch](#) und die [Privoxy-FAQ](#) sollte man als Privoxy-Benutzer kennen, an dieser Stelle daher als Appetit-Häppchen weitere Privoxy-Funktionen:

Access Control

Um andere Benutzer im Netzwerk ebenfalls auf Privoxy zugreifen zu lassen.

Anpassung der Templates

Um das Webinterface an den eigenen Geschmack anzupassen.

Trustfile

Ermöglicht den Aufbau einer sich selbst ergänzenden Whitelist.

Privoxy kann übrigens auch benutzt werden um [defekte Webshops zu reparieren](#) (veraltet), [Lemminge zu kennzeichnen](#), [getarnte Links sichtbar zu machen](#) und um auf mögliche [Sicherheits-Probleme mit Gmail](#) hinzuweisen.

Wer häufig von unsicheren Rechnern über unsichere Netze surft, sollte sich das auf [OpenBSD](#) basierende [Anonym.OS](#) und das auf [FreeBSD](#) basierende [FreeSBIE 2.0](#) (etwas aktueller) näher ansehen. Beide starten von CD und bringen Privoxy sowie Tor bereits mit.

Fabian Keil

www.fabiankeil.de/ fk@fabiankeil.de

\$Id: index.html,v 1.27 2007/05/03 12:01:36 fk Exp \$