Anonym im Internet - Seite 7

Originalartikel

Backup

<html> <h2 id=„privoxy-proxy“>Privoxy Proxy</h2> <h3 id=„vorstellung“>Vorstellung</h3> Privoxy (Tor Onion Service) ist der Nachfolger des in den 90er Jahren bekannten Junkbuster Filter-Proxy. Anders als z. B. der Squid oder Polipo Proxy ist Privoxy kein Proxy mit HTTP Caching, der in einem Cache bereits abgerufene Inhalte zwischenspeichert. Privoxy verfügt dafür wie die erwähnten Proxys ebenfalls über Merkmale und Funktionen, die Absicherung und Anonymisierung von Internetanwendungen unterstützen, die das unverschlüsselte HTTP- und das verschlüsselte HTTPS-Protokoll verwenden oder Schnittstellen zur Verwendung von HTTP/HTTPS Proxys aufweisen: <ul readability=„16“><li class=„level1“ readability=„14“> Weiterleitung des Datenverkehrs von Internetanwendungen an HTTP/HTTPS und SOCKS Proxys und damit an Anon-Netze und -Anwendungen, die SOCKS- oder HTTP-Schnittstellen aufweisen. Zusätzlich mit optionaler Anweisung an diese Anwendungen, den Datenverkehr an einen nachgeschalteten HTTP-Proxy weiterzuleiten. <li class=„level1“ readability=„4“> Steuerung und Manipulation der ausgesendeten Anfrage-Kopfzeilen der Internetanwendungen (Clients) und der empfangenen Antwort-Kopfzeilen der Server. <li class=„level1“ readability=„9“> Filterung und Blockierung des Empfangs und Versands von Daten und Inhalten aus unverschlüsselten HTTP-Datenströmen. Da Privoxy kein HTTPS/SSL Proxy mit MITM Funktionen ist, trifft das nicht auf verschlüsselte HTTPS-Datenströme zu. <li class=„level1“ readability=„8“> Regelwerk aus Aktionen und Filtern mit Unterstützung von Wildcards und Perl-kompatiblen Regulären Ausdrücken (PCRE), mit dem obige Funktionen allgemein oder spezifisch ausgeführt bzw. angewendet werden können.   Wie für <a href=„https://de.wikipedia.org/wiki/Proxy%20(Rechnernetz)“ class=„interwiki iw_wpde“

title=„https://de.wikipedia.org/wiki/Proxy (Rechnernetz)“>Proxys typisch, wird Privoxy dafür als lokal installierter Proxy-Server zwischen die Internetanwendungen und die SOCKS- oder HTTP-Schnittstelle des verwendeten Anon-Netzes geschaltet. In der Beschreibung des Privoxy Pakets wird Privoxy so charakterisiert: „HTTP-Proxy zur Verbesserung der Privatsphäre. Privoxy ist ein Web-Proxy mit fortschrittlichen Filterfähigkeiten zum Schutz der Privatsphäre, Filterung von Webinhalten, Cookieverwaltung, Zugriffskontrolle und Entfernung von Werbung, Bannern, Pop-ups und anderem unbeliebten Internet-Müll. Privoxy ist sehr flexibel konfigurierbar und kann an individuelle Bedürfnisse und den eigenen Geschmack angepasst werden. Privoxy kann sowohl in eigenständigen Systemen als auch in Mehrbenutzernetzwerken angewendet werden.“
  <h2 id=„installation-und-start“>Installation und Start</h2> <h3 id=„per-apt-repository“>per APT-Repository</h3> <pre class=„code console“> sudo aptitude install privoxy </pre> <h3 id=„kompilierung“>Kompilierung</h3> Wer Privoxy selbst kompiliert kann folgende Konfigurationsoptionen übergeben, um die Funktionen zu erweitern und Privoxy abzusichern: <table class=„inline c4“ readability=„7“><col class=„c2“/><col class=„c3“/><tr class=„row0“ readability=„1“><td class=„col0“>–disable-toggle</td> <td class=„col1“>Privoxys Filter- und Blockierfunktionen können nicht deaktiviert werden.</td> </tr><tr class=„row1“ readability=„2“><td class=„col0“>–disable-force</td> <td class=„col1 leftalign“>Privoxys Filter- und Blockierfunktionen können nicht für einzelne Seiten durch den Benutzer umgangen werden.</td> </tr><tr class=„row2“ readability=„1“><td class=„col0“>–disable-editor</td> <td class=„col1“>Die Konfiguration der Filter und Aktionen per Browser im Webeditor ist deaktiviert.</td> </tr><tr class=„row3“ readability=„2“><td class=„col0“>–disable-graceful-termination</td> <td class=„col1“>Privoxy kann nicht per Weboberfläche beendet werden.</td> </tr><tr class=„row4“ readability=„5“><td class=„col0“>–enable-compression</td> <td class=„col1“>gepufferte Inhalte werden vor Auslieferung an Clients komprimiert.
Die Kompression kann mit der enable-compression Option ein- oder ausgeschaltet und der Kompressionsgrad mit der compression-level Option bestimmt werden.</td> </tr><tr class=„row5“ readability=„3“><td class=„col0“>–enable-extended-host-patterns</td> <td class=„col1“>PCRE auch für die Domain/Hostnamen Muster.
Mit dem tools/url-pattern-translator.pl Perlskript lassen sich einmalig Wildcard Muster in Aktionsdateien in PCRE Muster umwandeln.</td> </tr><tr class=„row6“ readability=„2“><td class=„col0“>–enable-external-filters</td> <td class=„col1“>Inhalte können mit externen Skripten und Anwendungen gefiltert werden.</td> </tr></table> Mit den drei ersten Konfigurationsoptionen entfallen später die toggle, enforce-blocks und enable-edit-actions Optionen in der Privoxy Konfigurationsdatei. Nach dem Download des Quellcodearchivs, der GnuPG Signaturdatei und des GnuPG Schlüssels: <pre class=„code console“> cd /downloadverzeichnis gpg –import fk-id.asc gpg –verify privoxy-version-stable-src.tar.gz.asc tar -xzf privoxy-version-stable-src.tar.gz sudo apt-get build-dep -y privoxy sudo adduser –quiet –system –home /etc/privoxy –no-create-home –ingroup nogroup –disabled-password privoxy sudo mkdir /etc/privoxy /var/log/privoxy sudo chown privoxy:adm /var/log/privoxy sudo chmod 750

/var/log/privoxy cd privoxy-version-stable autoheader autoconf ./configure –sbindir=/usr/local/bin –disable-toggle –disable-force –disable-editor –enable-compression -enable-extended-host-patterns –disable-graceful-termination –enable-external-filters make sudo make install USER=privoxy GROUP=nogroup </pre> Nach der Installation wird Privoxy als Daemon über das /etc/init.d/privoxy Init-Skript automatisch mit dem privoxy Systembenutzer und der nogroup Systemgruppe gestartet. Standardmäßig lauscht der Privoxy Daemon an 127.0.0.1 und Port 8118 auf Verbindungsanfragen der Internetanwendungen. Neben Privoxy wird das privoxy-log-parser Perl-Skript installiert, mit dem sich die Ausgabe der Privoxy Logdatei verändern lässt.   Vom Privoxy Daemon werden folgende Verzeichnisse und Dateien genutzt: <table class=„inline c4“ readability=„9“><col class=„c5“/><col class=„c6“/><tr class=„row0“><td class=„col0“>/etc/init.d/privoxy</td> <td class=„col1“>Privoxy Init-Skript</td> </tr><tr class=„row1“ readability=„1“><td class=„col0“>/etc/privoxy/</td> <td class=„col1“>Konfigurations- und Heimatverzeichnis des Privoxy Daemon</td> </tr><tr class=„row2“ readability=„2“><td class=„col0“>/etc/privoxy/templates/</td> <td class=„col1“>enthält Vorlagen für den webbasierten Editor zur Privoxy Konfiguration</td> </tr><tr class=„row3“ readability=„1“><td class=„col0“>/etc/privoxy/config</td> <td class=„col1“>Privoxy Konfigurationsdatei</td> </tr><tr class=„row4“ readability=„1“><td class=„col0“>/etc/privoxy/default.action</td> <td class=„col1“>eingebaute Aktionen</td> </tr><tr class=„row5“ readability=„1“><td class=„col0“>/etc/privoxy/default.filter</td> <td class=„col1“>eingebaute Filter</td> </tr><tr class=„row6“ readability=„3“><td class=„col0“>/etc/privoxy/match-all.action</td> <td class=„col1“>Standard Aktionen, die durch die eingebauten und benutzerdefinierten Aktionen überschrieben werden</td> </tr><tr class=„row7“ readability=„2“><td class=„col0“>/etc/privoxy/user.action</td> <td class=„col1“>benutzerdefinierte Aktionen</td> </tr><tr class=„row8“ readability=„2“><td class=„col0“>/etc/privoxy/user.filter</td> <td class=„col1“>benutzerdefinierte Filter</td> </tr><tr class=„row9“ readability=„2“><td class=„col0“>/etc/privoxy/trust</td> <td class=„col1“>optional zu verwendende Whitelist aus Ziel- und Referrer-Adressen. Nur Anfragen von Referrer- oder Anfragen zu Ziel-Adressen sind erlaubt.</td> </tr><tr class=„row10“ readability=„2“><td class=„col0“>/var/log/privoxy/</td> <td class=„col1“>Verzeichnis, in dem die errorfile und die logfile Logdateien gespeichert werden</td> </tr><tr class=„row11“ readability=„2“><td class=„col0“>/var/run/privoxy.pid</td> <td class=„col1“>PID-Datei des Privoxy Daemon</td> </tr></table> <h3 id=„dokumentation“>Dokumentation</h3> Um alle Möglichkeiten auszuschöpfen, ist ein genaues Studium des Benutzerhandbuchs nötig. Privoxy enthält bereits ab Installation einen umfassenden Satz an Filtern und Aktionen, der viele Bereiche abdeckt. Von Fabian Keil gibt es die Anleitung zum werbefreien und spurenarmen Surfen mit Privoxy, die ebenfalls auf die Filterkonfiguration eingeht. Die Privoxy eigene Dokumentation kann man online im Web einsehen oder lokal über die Startseite des Privoxy Webeditors aufrufen. <h3 id=„verwendung-und-einbindung“>Verwendung und Einbindung</h3> <h4 id=„systemweit“>systemweit</h4> Für Internetanwendungen, die immer Privoxy verwenden sollen und Proxy Umgebungsvariablen auswerten, kann man die Verwendung von Privoxy durch das Setzen der Umgebungsvariablen automatisieren. Dazu setzt man systemweit oder in die .profile Datei des Benutzers: <pre class=„code“> export http_proxy=„http://127.0.0.1:8118/“ export https_proxy=„http://127.0.0.1:8118/“ export ftp_proxy=„http://127.0.0.1:8118/“ </pre> Da

Privoxy kein FTP-Proxy ist, werden mit Setzen und Auswerten der ftp_proxy Variable alle Verbindungen zu FTP-Servern blockiert. Mit der accept-intercepted-requests Option und entsprechenden iptables Regeln kann der ausgehende HTTP-Datenverkehr auch generell auf Privoxy umgeleitet werden. <h4 id=„anwendungsbezogen“>anwendungsbezogen</h4> Für einzelne Internetanwendungen wird die Verwendung von Privoxy aktiviert, indem 127.0.0.1:8118 als Adresse und Port den entsprechdenden Optionen bzw. Einstellungsfeldern der Internetanwendung übergeben wird. Für Firefox beispielsweise in den Verbindungs-Einstellungen und für andere Anwendungen per Option oder Skript: <pre class=„code console“> curl –proto https,http -x 127.0.0.1:8118 gpg –keyserver-options http-proxy=http://127.0.0.1:8118/ wget -e http_proxy=http://127.0.0.1:8118/ -e https_proxy=http://127.0.0.1:8118/ youtube-dl –proxy http://127.0.0.1:8118/ </pre> <pre class=„code bash“> #!/bin/sh

export http_proxy=„http://127.0.0.1:8118/“ export https_proxy=„http://127.0.0.1:8118/“ export ftp_proxy=„http://127.0.0.1:8118/“

/usr/bin/calibre

exit 0 </pre> <h2 id=„konfiguration“>Konfiguration</h2> <h3 id=„privoxy-webeditor“>Privoxy Webeditor</h3> Die Konfiguration von Privoxy kann über eine Oberfläche im Browser durchgeführt werden, die man mit der Adresse http://config.privoxy.org/ oder http://p.p/ aufruft: Startseite der Privoxy Weboberfläche. Über den ersten Menüpunkt View & change the current configuration (Anzeige und Änderung der aktuellen Konfiguration) gelangt man in die Konfiguration von Privoxy: Über das Anklicken der Edit Buttons gelangt man in eine Unterseite, die eine Übersicht aller Aktionsgruppen mit den zugeordneten Aktionen enthält. Eigene Änderungen sollten nach Möglichkeit in der user.action Datei durchgeführt werden. Da zum Konfigurieren und Anlegen von Aktionen und Filtern das Studium des Benutzerhandbuchs nötig ist, dem manuellen Editieren der Konfigurationsdateien der Vorzug gegeben wird und zur Absicherung von Privoxy selbst alle Webeditor-Funktionen mit Schreibberechtigungen deaktiviert sind, wird nicht näher auf den Webeditor eingegangen. <h3 id=„privoxy-individuell-und-manuell-konfigurieren“>Privoxy individuell und manuell konfigurieren</h3> Wenn Privoxy nur mit den Optionen und Funktionen betrieben werden soll, die für die Nutzung von Tor (oder anderen Proxys) relevant sind und die eine Anonymisierung unterstützen, kann man an Stelle der vorgegebenen Filter- und Aktionsdateien, die man über die obige Weboberfläche einstellt, eine eigene Konfiguration verwenden. Das kann u. a. sinnvoll sein, wenn man die Filterung unerwünschter Inhalte, Verwendung aktiver Inhalte usw. lieber über die Konfiguration der bevorzugten Internetanwendung regelt und nicht über einen Filterproxy. Die default.action und default.filter Dateien können trotzdem als „Repositories“ dienen, aus denen man Aktionen und Filter für die eigenen Dateien übernimmt. <h3

id=„hauptkonfiguration“>Hauptkonfiguration</h3> Die Hauptkonfiguration wird durch das Editieren der /etc/privoxy/config Datei durchgeführt. <h4 id=„systempfade“>Systempfade</h4> Die folgenden Optionen am Anfang der /etc/privoxy/config Datei bleiben unangetastet: <pre class=„code“> user-manual /usr/share/doc/privoxy/user-manual confdir /etc/privoxy templdir /etc/privoxy/templates logdir /var/log/privoxy </pre> Privoxy wird zukünftig nur die user.action und user.filter Datei verwenden: <pre class=„code“> # actionsfile match-all.action # actionsfile default.action actionsfile user.action # filterfile default.filter filterfile user.filter </pre> Mit den debug Optionen können der Umfang und die Inhalte der Logdatei-Ausgaben gesteuert werden. Um bestimmte debug Optionen zu aktivieren, muss das # Kommetarzeichen vor den entsprechenden debug Zeilen entfernt werden. Hier ein paar Beispiele: <table class=„inline c4“ readability=„4“><col class=„c5“/><col class=„c6“/><thead><tr class=„row0“><th class=„col0“>Option</th> <th class=„col1“>Erklärung</th> </tr></thead><tr class=„row1“ readability=„2“><td class=„col0“>debug 1</td> <td class=„col1“>Ziele, zu denen Privoxy Verbindungen durchlässt/weiterleitet</td> </tr><tr class=„row2“ readability=„1“><td class=„col0“>debug 8</td> <td class=„col1“>Auswertung der Kopfzeilen</td> </tr><tr class=„row3“ readability=„1“><td class=„col0“>debug 64</td> <td class=„col1“>Anwendung von Regulären Ausdrücken in Filtern</td> </tr><tr class=„row4“ readability=„1“><td class=„col0“>debug 128</td> <td class=„col1“>Umleitungen auf andere Websites und Webseiten</td> </tr><tr class=„row5“ readability=„2“><td class=„col0“>debug 1024</td> <td class=„col1“>Ziele, zu denen Privoxy Verbindungen blockiert mit Nennung der Gründe</td> </tr><tr class=„row6“ readability=„1“><td class=„col0“>debug 4096</td> <td class=„col1“>Start- und Warn-Meldungen</td> </tr><tr class=„row7“><td class=„col0“>debug 8192</td> <td class=„col1“>Fehler-Meldungen</td> </tr><tr class=„row8“ readability=„1“><td class=„col0“>debug 65536</td> <td class=„col1“>Anwendung aller Aktionen auf eine Anfrage</td> </tr></table>   Mit dem privoxy-log-parser Skript kann man die Anzeige der aktuellen Ausgaben in die Privoxy Logdatei oder die Ausgabe einer archivierten Logdatei verändern. Zum Beispiel kann man sich mit dem folgenden Kommando die aktuellen Ausgaben in die Logdatei anzeigen lassen: <pre class=„code console“> sudo tail -f -n 20 /var/log/privoxy/logfile | sudo privoxy-log-parser –shorten-thread-ids –show-ineffective-filters </pre> <h4 id=„zugangskontrolle-und-sicherheit“>Zugangskontrolle und Sicherheit</h4> <table class=„inline c4“ readability=„14“><col class=„c2“/><col class=„c7“/><col class=„c8“/><thead><tr class=„row0“><th class=„col0“>Option</th> <th class=„col1“>Wert</th> <th class=„col2“>Erklärung</th> </tr></thead><tr class=„row1“ readability=„6“><td class=„col0“>listen-address</td> <td class=„col1“>IP-Adresse:8118
127.0.0.1:8118
Hostname:8118</td> <td class=„col2“>Adresse(n) und Port(s), an denen Privoxy auf eingehende Verbindungsanfragen lauscht. Kann mehrmalig verwendet werden. Bei Angabe von Hostnamen sollte ihre korrekte lokale Namensauflösung sichergestellt sein.</td> </tr><tr class=„row2“ readability=„3“><td class=„col0“>toggle</td> <td class=„col1“>1</td> <td class=„col2“>Blockier- und Filtermodus aktiviert. Mit 0 wird Privoxy zum transparenten, inhaltsneutralen Web-Proxy.</td> </tr><tr class=„row3“ readability=„2“><td class=„col0“>enable-remote-toggle</td> <td class=„col1“>0</td> <td class=„col2“>webbasiertes Abschalten des Blockier- und Filtermodus deaktiviert.</td> </tr><tr class=„row4“ readability=„2“><td class=„col0“>enable-remote-http-toggle</td> <td class=„col1“>0</td> <td class=„col2“>Abschalten des Filtermodus per X-Filter: No Kopfzeile deaktiviert.</td> </tr><tr class=„row5“ readability=„1“><td class=„col0“>enable-edit-actions</td> <td class=„col1“>0</td> <td class=„col2“>Verwendung des Privoxy-Webeditors zur Konfiguration der

user.action deaktiviert.</td> </tr><tr class=„row6“ readability=„1“><td class=„col0“>enforce-blocks</td> <td class=„col1“>1</td> <td class=„col2“>Umgehung von Blockierungen durch Benutzer deaktiviert.</td> </tr><tr class=„row7“ readability=„5“><td class=„col0“>permit-access</td> <td class=„col1“>IP-Adresse
127.0.0.1
Hostname
Netzwerkadresse/CDIR</td> <td class=„col2“>Nur von den angegebenen IP-Adressen bzw. IP-Adressen der angegebenen Netzwerkadressen oder Hostnamen werden Verbindungsanfragen entgegengenommen und vom Rest verweigert. Kann mehrmalig verwendet werden. Bei Angabe von Hostnamen sollte ihre korrekte lokale Namensauflösung sichergestellt sein.</td> </tr><tr class=„row8“ readability=„2“><td class=„col0“>buffer-limit</td> <td class=„col1“>n</td> <td class=„col2“>jeder Privoxy Thread verwendet max. n KB RAM zur Pufferung der Inhalte zwecks Filtern des Inhalts (Standard: 4096 KB/4 MB).</td> </tr><tr class=„row9“ readability=„4“><td class=„col0“>accept-intercepted-requests</td> <td class=„col1“>0|1</td> <td class=„col2“>1, wenn HTTP/HTTPS Datenverkehr generell für alle Internetanwendungen per iptables auf Privoxy umgeleitet wird.</td> </tr><tr class=„row10“ readability=„2“><td class=„col0“>allow-cgi-request-crunching</td> <td class=„col1“>1</td> <td class=„col2“>Anfragen zu Privoxys CGI-Seiten werden blockiert oder umgeleitet.</td> </tr><tr class=„row11“ readability=„2“><td class=„col0“>max-client-connections</td> <td class=„col1“>n</td> <td class=„col2“>Privoxy bedient maximal n gleichzeitige Verbindungen der Internetanwendungen (Standard: 128).</td> </tr></table> <h4 id=„verbindungen“>Verbindungen</h4>   <table class=„inline c4“ readability=„14“><col class=„c2“/><col class=„c7“/><col class=„c8“/><thead><tr class=„row0“><th class=„col0“>Option</th> <th class=„col1“>Wert</th> <th class=„col2“>Erklärung</th> </tr></thead><tr class=„row1“ readability=„16“><td class=„col0“>client-header-order Host User-Agent Accept Accept-Language Accept-Encoding DNT Referer Cookie Connection Pragma Cache-Control Content-Type Content-Length</td> <td class=„col1“>Liste</td> <td class=„col2“>die mit Leerzeichen getrennten aufgeführten Kopfzeilen werden zuerst in der angegebenen Reihenfolge sortiert und dann an die Server ausgeliefert. Die Kopfzeilen, die eine Anwendung zusätzlich verwendet, aber nicht in der Liste aufgeführt sind, werden danach angeführt. Um z. B. alle Webanwendungen authentisch als Firefox auftreten zu lassen, müssen anwendungsspezifische Kopfzeilen zuerst mit einem client-header-filter Filter gelöscht und alle möglichen Kopfzeilen, die Firefox unterstützt, aufgeführt werden. Die richtige Reihenfolge erschwert das Browser-Fingerprinting, eine falsche oder mit Lücken behaftete Liste befördert das Browser-Fingerprinting.</td> </tr><tr class=„row2“ readability=„4“><td class=„col0“>keep-alive-timeout</td> <td class=„col1“>n</td> <td class=„col2“>Verbindungen der Internetanwendungen zu Privoxy und von Privoxy zu den Servern – falls vom Server unterstützt – werden n Sekunden offen gehalten und für weitere Verbindungsanfragen genutzt (Standard: 5 Sekunden).</td> </tr><tr class=„row3“ readability=„1“><td class=„col0“>tolerate-pipelining</td> <td class=„col1“>1</td> <td class=„col2“>die gebündelte Übertragung von Anfragen per HTTP/1.1 Pipelining werden von Privoxy bedient.</td> </tr><tr class=„row4“ readability=„6“><td class=„col0“>default-server-timeout</td> <td class=„col1“>n</td> <td class=„col2“>angenommener Schwellenwert der Zeitüberschreitung für Server in n Sekunden für offen gehaltene Verbindungen, wenn die Server selbst keinen Schwellenwert übermitteln. Nur in Verbindung mit deaktivierter connecton-sharing Option und einem Wert < keep-alive-timeout sinnvoll (Standard: deaktiviert).</td> </tr><tr class=„row5“ readability=„3“><td class=„col0“>connection-sharing</td> <td class=„col1“>n</td> <td class=„col2“>mit n = 1 kann eine offen gehaltene Verbindung zu einem Server von mehreren Internetanwendungen parallel für ihre

Verbindungen zum gleichen Server verwendet werden (Standard: deaktiviert).</td> </tr><tr class=„row6“ readability=„2“><td class=„col0“>socket-timeout</td> <td class=„col1“>n</td> <td class=„col2“>Schwellenwert der Zeitüberschreitung für Verbindungen über SOCKS-Proxys (wie z. B. Tor) in n Sekunden (Standard: 300 Sekunden).</td> </tr></table> Der Wert für keep-alive-timeout und default-server-timeout richtet sich z. B. danach, ob im Firefox network.http.keep-alive aktiviert (Standard: true) ist und wie hoch der Schwellenwert für network.http.keep-alive.timeout ist. Liegt der Schwellenwert z. B. bei 30 Sekunden, kann default-server-timeout den gleichen Wert erhalten und keep-alive-timeout > 30. Wenn man davon ausgeht, dass Datenströme verschiedener Internetanwendungen, Protokolle, Zieladressen usw. möglichst auf viele unterschiedliche Tor Verbindungsketten verteilt werden sollten (siehe multiple SOCKSPorts und Isolierungsmarker in der Tor Konfiguration), um mögliche „Identitäts-Korrelationen“ zu vermeiden, sollte connection-sharing deaktiviert bleiben. Für socket-timeout richtet sich der Wert z. B. für Tor nach der CircuitStreamTimeout Option und – unter der Annahme, dass für eine Weiterleitungsanfrage von Privoxy zu Tor erst eine Tor-Kette geschaltet werden müsste – der CircuitBuildTimeout bzw. LearnCircuitBuildTimeout Option in der Tor Konfiguration. Standardmäßig verwendet Tor überhaupt keine festen Schwellenwerte mehr, sondern interne, selbstregulierende Funktionen zur dynamischen Schwellenwertermittlung, sofern keine statischen Wert vom Benutzer vorgegeben werden. Da für beide Optionen die statischen Werte bei 60 Sekunden lagen und meistens maximale Schwellenwerte von 60 - 75 Sekunden von Tor ermittelt werden, kann man den Wert z. B. auf 120 Sekunden setzen. <h3 id=„weiterleitungen“>Weiterleitungen</h3> Mit Regeln zur Weiterleitung in der /etc/privoxy/config Konfigurationsdatei werden Verbindungen der Internetanwendungen von Privoxy an einen lokalen HTTP-Proxy, SOCKS-Proxy, SOCKS-Proxy + entfernten HTTP-Proxy weitergeleitet, die dann die Verbindung über Anon-Dienste oder -Netzwerke zum Server herstellen oder direkt zu Servern umgeleitet. Für Internetanwendungen, Anon-Dienste und -Netzwerke mit HTTP oder SOCKS-Proxy Schnittstelle können damit Verbindungen der Internetanwendungen anonymisiert werden. Andererseits kann man für Zieladressen, zu denen bewußt keine anonymisierte Verbindungen aufgenommen werden sollen, Ausnahme-Regeln definieren. Auf die direkte Weiterleitung von Verbindungen über einfache, offen gehaltene HTTP- und SOCKS-Proxys, die man über diverse Proxy-Listen im Web findet, wird nicht eingegangen, da Verschlüsselung, Anonymisierung und Unbeobachtbarkeit nicht gewährleistet sind. Solche Maßnahmen sind Anonymity by Obscurity. Sie werden nur als Ausnahme herangezogen, wenn Ausgang-Netzknoten verwendeter Anon-Dienste oder -Netzwerke von Betreibern der Zieladressen blockiert werden. <h4 id=„ziel-muster“>Ziel-Muster</h4> Für die Definition der Zieladressen, die von Weiterleitungsregeln erfasst werden sollen, können generelle oder spezifische Muster zur Erfassung von Domainnamen und/oder Pfaden herangezogen werden: <table class=„inline“ readability=„5“><thead><tr class=„row0“><th class=„col0“>Muster</th> <th class=„col1“>erfasst:</th> </tr></thead><tr class=„row1“ readability=„1“><td class=„col0“>/</td> <td class=„col1“>generell alle Zieladressen

(Domainnamen)</td> </tr><tr class=„row2“ readability=„2“><td class=„col0“>:Port/</td> <td class=„col1“>generell alle Zieladressen (Domainnamen), deren Server auf bestimmten Portnummern lauschen</td> </tr><tr class=„row3“ readability=„2“><td class=„col0“>subdomain.domain.tld[:Port]</td> <td class=„col1“>bestimmte Domainnamen – optional: deren Server auf bestimmten Portnummern lauschen</td> </tr><tr class=„row4“ readability=„2“><td class=„col0“>subdomain.domain.tld/pfad</td> <td class=„col1“>bestimmte Domainnamen mit bestimmten Pfadangaben</td> </tr><tr class=„row5“ readability=„3“><td class=„col0“>subdomain.domain.tld:Port/pfad</td> <td class=„col1“>bestimmte Domainnamen mit bestimmten Pfadangaben, deren Server auf bestimmten Portnummern lauschen</td> </tr><tr class=„row6“ readability=„1“><td class=„col0“>/pfad</td> <td class=„col1“>bestimmte Pfade unabhängig von Domainnamen</td> </tr></table> <h4 id=„domainnamen“>Domainnamen</h4> Domainnamen können komplett oder anteilig angegben werden, wobei Anteile und Namensbestandteile durch die Wildcards * für 0 oder beliebig viele Zeichen, ? für ein beliebiges Zeichen oder [Zeichenklasse(n)] für ein beliebiges Zeichen aus der angegebenen Zeichenklasse ersetzt werden können. Wird Privoxy mit der –enable-extended-host-patterns Konfigurationsoption selbst kompiliert, können auch Perl-kompatible Reguläre Ausdrücke (PCRE) verwendet werden. <table class=„inline c4“ readability=„5“><col class=„c7“/><col class=„c9“/><col class=„c10“/><thead><tr class=„row0“><th class=„col0“ colspan=„3“>Beispiele Domainnamen</th> </tr><tr class=„row1“><th class=„col0“>Muster</th> <th class=„col1“>Beispiel</th> <th class=„col2“>erfasst:</th> </tr></thead><tr class=„row2“ readability=„2“><td class=„col0“>subdomain.domain.tld</td> <td class=„col1“>www.google.com</td> <td class=„col2“>genau www.google.com</td> </tr><tr class=„row3“ readability=„1“><td class=„col0“>.domain.tld</td> <td class=„col1“>.google.com</td> <td class=„col2“>u. a. www|maps|news.google.com</td> </tr><tr class=„row4“ readability=„1“><td class=„col0“>.domain.</td> <td class=„col1“>.google.</td> <td class=„col2“>u. a. www|maps|news.google.com|de|us</td> </tr><tr class=„row5“ readability=„1“><td class=„col0“>.tld</td> <td class=„col1“>.onion</td> <td class=„col2“>alle Tor Hostnamen der Tor-Dienste</td> </tr><tr class=„row6“ readability=„3“><td class=„col0“>*</td> <td class=„col1“>www.paypal*.com:443</td> <td class=„col2“>u. a. www.paypal.com, www.paypalobjects.com und www.paypal-deutschland.com mit verschlüsselter TLS Verbindung über Port 443</td> </tr><tr class=„row7“ readability=„1“><td class=„col0“>?</td> <td class=„col1“>www.wdr?.de</td> <td class=„col2“>u. a. www.wdr2.de bis www.wdr5.de</td> </tr><tr class=„row8“ readability=„1“><td class=„col0“>[Zeichenklasse(n)]</td> <td class=„col1“>www.wdr[2-5].de</td> <td class=„col2“>genau www.wdr2.de bis www.wdr5.de</td> </tr><tr class=„row9“ readability=„1“><td class=„col0“>/</td> <td class=„col1“/> <td class=„col2“>alle Domain-/Hostnamen</td> </tr></table> <h4 id=„pfade“>Pfade</h4> In Pfadangaben können und müssen Perl-kompatible <a

href=„https://de.wikipedia.org/wiki/Regul%C3%A4rer%20Ausdruck“ class=„interwiki iw_wpde“ title=„https://de.wikipedia.org/wiki/Regulärer Ausdruck“>Reguläre Ausdrücke (PCRE) verwendet werden. Da das Thema Reguläre Ausdrücke komplex und umfangreich ist, wird auf Beispiele verzichtet. In den Wikipedia Artikeln und über die darin enthaltenen Links erhält man ausreichend Informationen, um die Verwendung Regulärer Ausdrücke zu erlernen und zu testen. Zusätzlich sind in den mitgelieferten Aktions- und Filterdateien von Privoxy genügend Beispiele enthalten und im Privoxy Handbuch wird in den Kapiteln The Path Pattern und Regular Expressions in Reguläre Ausdrücke eingeführt. Für eine erste Abgrenzung der obigen Wildcards von Regulären Ausdrücken: <table class=„inline c4“ readability=„9“><col class=„c7“/><col class=„c9“/><col class=„c10“/><thead><tr class=„row0“><th class=„col0“>Zeichen</th> <th class=„col1“>Wildcard (in Domainnamen)</th> <th class=„col2“>RegEx (in Pfaden)</th> </tr></thead><tr class=„row1“ readability=„2“><td class=„col0“>*</td> <td class=„col1“>steht für 0 oder beliebig viele Zeichen</td> <td class=„col2“>das * vorangestellte RegEx-Muster – RegEx* – trifft nullmal oder beliebig oft zu</td> </tr><tr class=„row2“ readability=„2“><td class=„col0“>?</td> <td class=„col1“>steht für ein beliebiges Zeichen</td> <td class=„col2“>das ? vorangestellte RegEx-Muster – RegEx? – trifft null- oder einmal zu</td> </tr><tr class=„row3“ readability=„1“><td class=„col0“>.</td> <td class=„col1“>keine Bedeutung</td> <td class=„col2“>ein beliebiges Zeichen</td> </tr><tr class=„row4“ readability=„5“><td class=„col0“>[Z-Klasse(n)]</td> <td class=„col1“>ein Zeichen aus den angegebenen Zeichenklassen</td> <td class=„col2“>ähnlich, als RegEx-Muster mit optional angefügten *, ?, + Quantoren – z. B. [Z-Klasse(n)]+</td> </tr><tr class=„row5“ readability=„1“><td class=„col0“>+</td> <td class=„col1“>keine Bedeutung</td> <td class=„col2“>das + vorangestellte RegEx-Muster – RegEx+ – trifft ein- oder mehrmalig zu</td> </tr><tr class=„row6“ readability=„8“><td class=„col0“>(Ausdruck)
(Ausdruck1|AusdruckN)</td> <td class=„col1“>keine Bedeutung</td> <td class=„col2“>Zusammenfassung/Gruppierung mehrer RegEx-Muster bzw. RegEx-Teilmuster – (RegEx1RegexN) bzw. (RegEx-Teil1|RegEx-TeilN), wobei „|“ für „oder“ steht, der man die *, ?, + Quantoren anfügen kann</td> </tr></table> Will man die *, ?, ., +, [,], (,) u. a. Zeichen in den Pfadangaben nicht als Metazeichen Regulärer Ausdrücke, d. h. mit ihrer Bedeutung für Reguläre Ausdrücke verwenden, sondern direkt in ihrer Eigenschaft der jeweiligen Zeichenklasse, müssen sie mit einem vorangestellten „\“ maskiert werden – \? hebt z. B. die Bedeutung von ? als RegEx-Quantor auf und ? wird als Fragezeichen erfasst. <h4 id=„regel-syntax“>Regel-Syntax</h4> <h5 id=„reihenfolge“>Reihenfolge</h5> Für die Anordnung der Weiterleitungsregeln gilt: „der letzte Treffer zählt“. Bei einer Verbindungsanfrage einer Internetanwendung geht Privoxy alle Weiterleitungsregeln durch. Die letzte Weiterleitungsregel, deren Zieladressen-Muster auf die Zieladresse in der Verbindungsanfrage passt, wird dann auf die Verbindung angewendet. Deshalb sollte man sich angewöhnen, Weiterleitungsregeln wie folgt anzuordnen: <table class=„inline“ readability=„5“><thead><tr class=„row0“><th class=„col0“>Position</th> <th class=„col1“>Weiterleitungsregeln</th> </tr></thead><tr class=„row1“ readability=„2“><th

class=„col0“>1</th> <td class=„col1“>Generelle Regel für das Anon-Netzwerk, das hauptsächlich für alle Verbindungen verwendet wird</td> </tr><tr class=„row2“ readability=„2“><th class=„col0“>2</th> <td class=„col1“>Regeln für Verbindungen zu Zieladressen, die über andere Anon-Netzwerke weitergeleitet werden</td> </tr><tr class=„row3“ readability=„3“><th class=„col0“>3</th> <td class=„col1“>Regeln für Verbindungen zu Zieladressen, die über Anon-Netzwerke mit zusätzlicher Weiterleitung an einen HTTP-Proxy weitergeleitet werden</td> </tr><tr class=„row4“ readability=„4“><th class=„col0“>4</th> <td class=„col1“>Regeln für Verbindungen zu Zieladressen, die über kein Anon-Netzwerk weitergeleitet, sondern – als Ausnahmen – den Servern direkt zugestellt werden</td> </tr></table> <h5 id=„lokale-socks-proxy-weiterleitung“>Lokale SOCKS-Proxy Weiterleitung</h5> Mit der Regel werden Verbindungen zur Zieladresse an den lokalen SOCKS-Proxy der Anonymisierungs-Anwendung weitergeleitet, der sie über das Anon-Netzwerk zum Server der Zieladresse weiterleitet. Alle Anon-Dienste und -Netzwerke, deren lokale Anwendungen als SOCKS-Proxy arbeiten, werden über die Regel bedient. <dl class=„code“ readability=„0“><dt>Regel Syntax</dt> <dd readability=„1“> <pre class=„code h“> forward-socks5 Zieladresse SOCKS-Proxy:Port . </pre></dd> </dl><dl class=„code“ readability=„0“><dt>Regel Beispiele</dt> <dd readability=„1“> <pre class=„code h“> forward-socks5 / localhost:9050 . forward-socks5 / localhost:4001 . </pre></dd> </dl> Die erste Regel leitet alle Verbindungen an den lokalen Tor Onion Proxy, der an der lokalen Schnittstelle auf Port 9050 lauscht, für das Tor Anon-Netzwerk weiter. Mit forward-socks5t verwendet Privoxy Tor-spezifische Erweiterungen des SOCKS-Protokolls wie z. B. OptimisticData. Die zweite Regel leitet alle Verbindungen an den lokalen JonDo Proxy für den JonDonym Anon-Dienst weiter, wenn man die kostenpflichtigen JonDonym Mix-Kaskaden nutzt, deren Mix-Netzwerkknoten SOCKS unterstützen. <h5 id=„lokale-http-proxy-weiterleitung“>Lokale HTTP-Proxy Weiterleitung</h5> Mit der Regel werden Verbindungen zur Zieladresse an den lokalen HTTP-Proxy der Anonymisierungs-Anwendung weitergeleitet, der sie über das Anon-Netzwerk zum Server der Zieladresse weiterleitet. Alle Anon-Dienste und -Netzwerke, deren lokale Anwendungen als HTTP-Proxy arbeiten, werden über die Regel bedient. <dl class=„code“ readability=„0“><dt>Regel Syntax</dt> <dd readability=„1“> <pre class=„code h“> forward Zieladresse HTTP-Proxy[:Port] </pre></dd> </dl><dl class=„code“ readability=„0“><dt>Regel Beispiele</dt> <dd readability=„1“> <pre class=„code h“> forward .i2p localhost:4444 forward / localhost:4001 forward .domain.tld localhost:4001 </pre></dd> </dl> Die erste Regel dient der Weiterleitung der Verbindungen durch I2P Tunnels zu Eepsites des I2P Anon-Netzwerks über die lokale HTTP-Proxy Schnittstelle der I2P Anwendung. Die zweite Regel leitet alle Verbindungen an den lokalen JonDo

Proxy des JonDonym Anon-Dienstes weiter, wenn man die kostenlosen JonDonym Mix-Kaskaden nutzt, für deren Mix-Netzwerkknoten SOCKS nicht verfügbar ist, während die dritte Regel nur Verbindungen zu ausgewählten Domainnamen über die kostenlosen JonDonym Mix-Kaskaden weiterleitet. <h5 id=„lokale-socks-proxy-http-proxy-weiterleitung“>Lokale SOCKS-Proxy -> HTTP-Proxy Weiterleitung</h5> Mit der Regel werden die Verbindungen zur Zieladresse zuerst an den SOCKS-Proxy der Anonymisierungs-Anwendung weitergeleitet, der sie über das Anon-Netzwerk zum Ausgang-Netzknoten des Anon-Netzwerks weiterleitet. Vom Ausgang-Netzknoten wird die Verbindung über den angegebenen HTTP-Proxy an den Server der Zieladresse weitergeleitet. Die Regel eignet sich dazu, anonymisierte Verbindungen zu Servern herzustellen, die Verbindungen von Ausgang-Netzknoten der Anon-Dienste und -Netzwerke ablehnen bzw. blockieren oder um einen weiteren Dienst zusätzlich anonymisiert zu nutzen, der über HTTP-Proxys angesprochen wird. <dl class=„code“ readability=„0“><dt>Regel Syntax</dt> <dd readability=„1“> <pre class=„code h“> forward-socks5[t] Zieladresse SOCKS-Proxy:Port HTTP-Proxy[:Port] </pre></dd> </dl><dl class=„code“ readability=„1“><dt>Regel Beispiele</dt> <dd readability=„3“> <pre class=„code h“> forward-socks5 europa.eu localhost:9050 IP-Adresse:Port forward-socks5 .bit localhost:9050 178.32.31.43:8888 forward-socks5 translate.google.com localhost:9050 googlesharing.riseup.net:80 forward-socks5 encrypted.google.com localhost:9050 gs.netsend.nl:81 </pre></dd> </dl> Für die erste Regel sucht man sich einen offenen HTTPS-Proxy bei einem Proxy-Listen Anbieter wie z. B. Hide My Ass! und setzt dessen IP-Adresse und Port ein, um die Blockierung von Verbindungen zur Europäischen Kommission zu umgehen, die per Tor anonymisiert wurden. Die zweite Regel anonymisiert Verbindungen zum Proxy des Dot-Bit Projekts per Tor, um sich zu Dot-Bit Websites (*.bit) zu verbinden. Die beiden letzten Regeln anonymisieren die Verbindungen zu Googlesharing Proxys (z. B. von Riseup), die man für Google Dienste benutzen kann, womit sich nebenbei der Cookie- und Captcha-Zwang reduzieren lässt. <h5 id=„direkte-weiterleitung“>Direkte Weiterleitung</h5> Mit der Regel werden die Verbindungen zur Zieladresse direkt zum Server der Zieladresse weitergeleitet. Die Regel eignet sich für die Definition von Ausnahmen für Verbindungen zu Zieladressen, die nicht anonymsiert bzw. über ein Anon-Netzwerk geleitet werden sollen. <dl class=„code“ readability=„0“><dt>Regel Syntax</dt> <dd readability=„1“> <pre class=„code h“> forward Zieladresse . </pre></dd> </dl><dl class=„code“ readability=„2“><dt>Regel Beispiele</dt> <dd readability=„4“> <pre class=„code h“> forward 192.168.*.*/ . forward meine.lokaledomain.local/ . forward 127.*.*.*/ . forward localhost/ . forward .meine-bank.tld:443 . forward .domain.tld:443/login/ . forward downloads.sourceforge.net/project/(.*/)+.*&use_mirror=[a-z0-9]+$. forward .dl.sourceforge.net/project/(.*/)+[a-z\-\._0-9]+\.(asc|bz2|deb|gz|sig|zip)$. </pre></dd> </dl> <h3 id=„aliase-aktionen-und-filter“>Aliase, Aktionen und Filter</h3> <div class=„level3“

readability=„41“> Aktionen und Aliase werden in der /etc/privoxy/user.action Datei definiert. Aktionen wenden eingebaute Funktionen und definierte Filter an, die der Blockierung und Manipulation dienen: <ul readability=„5“><li class=„level1“> von URLs <li class=„level1“ readability=„2“> von unverschlüsselt übertragenen Webseiteninhalten <li class=„level1“ readability=„4“> von markierten Kopfzeilen, die von Internetanwendungen ausgesendet werden <li class=„level1“ readability=„4“> von markierten Kopfzeilen, die Server zu Internetanwendungen übertragen.   Aktionen zielen wie die Weiterleitungsregeln auf alle oder spezifische Domainnamen und Pfade, alle oder spezifische Internetanwendungen und Server-Kopfzeilen. Für die Definition der Domainnamen und Pfade kann man in gleicher Weise wie in den Weiterleitungsregeln Wildcards bzw. Reguläre Ausdrücke einsetzen. Mehrere Aktionen und Filter können miteinander kombiniert und als Alias oder in „Aktionscontainern“ gebündelt werden, um sie allgemein oder auf bestimmte Ziele und markierte Kopfzeilen anzuwenden. Für die Anordnung der Aktionscontainer gilt: für jede Anfrage werden alle Aktionscontainer bzw. die darin enthaltenen Aktionen und Filter von oben nach unten durchgegangen und auf Übereinstimmung mit den Zielmustern und Markierungen überprüft. Alle Aktionen und Filter, die auf verschiedene Aktionscontainer verteilt sind und wo die Übereinstimmung mit den Zielmustern und Markierungen gilt, werden summiert, wobei zuletzt definierte Aktionen und Filter vom gleichen Typ/Zweck vorangehende Aktionen und Filter überschreiben. Deshalb gilt für ihre Anordnung das gleiche Schema wie für Weiterleitungsregeln. Bei ihrem Einsatz muss man überlegen, für welche Internetanwendungen die Aktionen und Filter – neben der Verwendung von Privoxy an sich – in Frage kommen. Außerdem muss man abwägen, ob und welchem Umfang Filtermaßnahmen, Blockierungen und Manipulationen zentral über Privoxy durchgeührt werden oder über die Interetanwendung und ihre Plugins, Erweiterungen usw. selbst. Zum Beispiel erübrigen sich für den Firefox Webbrowser viele Aktionen und Filter, wenn man entsprechende Erweiterungen einsetzt. <h4 id=„aliase-und-aktionen“>Aliase und Aktionen</h4> Ein Alias ist ein Sammelname für die Zusammenstellung mehrerer Aktionen und Filter. Ist ein Alias definiert, kann er später an anderer Stelle der user.action Datei als {Anker} verwendet werden, um die Eingabe von Kombinationen aus Aktionen und Filtern abzukürzen. Wenn man Aliase nutzen will, müssen sie an erster Stelle in der user.action Datei definiert werden. Der (erste) Abschnitt in der user.action Datei, in der die Aliase stehen, wird mit alias eingeleitet. <dl class=„code“ readability=„1“><dt>Alias Syntax</dt> <dd readability=„2“> <pre class=„code h“> alias aliasname1 = -/+aktionsname1 -/+filter2 usw. aliasname2 = -/+aktionsname4 -/+filter5 usw.

{aliasname1} Ziele oder TAG: </pre></dd> </dl><dl class=„code“ readability=„2“><dt>Alias Beispiel</dt> <dd readability=„4“> <pre class=„code h“> alias

allow-all = -block -filter -hide-referer -crunch-incoming-cookies -crunch-outgoing-cookies \ -add-header -hide-if-modified-since -overwrite-last-modified -crunch-if-none-match \ -server-header-filter +forward-override{forward .} -fast-redirects

{allow-all} wiki.kairaven.de localhost:port </pre></dd> </dl> Aktionen und in ihnen referenzierte Filter werden nach den definierten Aliase in der user.action Datei eingetragen. Aktivierte Aktionen und Filter werden mit +, nicht anzuwendende Aktionen und Filter mit - Präfix angegeben. In einem {Aktionscontainer} können folgende Aktionen und Filter als Bestandteile angegeben werden: <dl class=„code“ readability=„1“><dt>Aktionscontainer Syntax</dt> <dd readability=„3“> <pre class=„code h“> { \ -/+aktionsname \ -/+aktionsname{parameter} \ -/+filter{filtername} \ -/+client-header-filter{filtername} \ -/+client-header-tagger{name} \ -/+server-header-filter{filtername} \ -/+server-header-tagger{name} \ } Ziele oder TAG: </pre></dd> </dl> Wenn man nur eine oder wenige Aktionen und Filter für Ziele miteinander kombiniert, kann man für einen Aktionscontainer auch Einzeiler verwenden: <dl class=„code“ readability=„1“><dt>Einzeiler Syntax</dt> <dd readability=„2“> <pre class=„code h“> { -/+aktionsname{parameter} -/+filter{filtername} -/+-client-header-filter{filtername} } Ziele oder TAG: </pre></dd> </dl><div class=„table sectionedit10“> <table class=„inline c4“ readability=„28“><col class=„c11“/><col class=„c11“/><col class=„c5“/><thead><tr class=„row0“><th class=„col0“>Bestandteil</th> <th class=„col1“>Beispiel</th> <th class=„col2“>Erklärung</th> </tr></thead><tr class=„row1“ readability=„5“><td class=„col0“>aktionsname</td> <td class=„col1“>crunch-incoming-cookies</td> <td class=„col2“ readability=„5“>Eingebaute Aktion. Beispiel: löscht die Set-Cookie: Antwort-Kopfzeile der Server, sprich eingehende Cookies. </td> </tr><tr class=„row2“ readability=„8“><td class=„col0“>aktionsname{parameter}</td> <td class=„col1“>hide-referrer{forge}</td> <td class=„col2 leftalign“ readability=„5“>Eingebaute Aktion mit zusätzlichem Parameter, der die Aktion näher bestimmt. Beispiel: fälscht die Ursprungsadresse in der Referer: Anfrage-Kopfzeile immer auf die Stammadresse der aufgerufenen Website. </td> </tr><tr class=„row3“ readability=„14“><td class=„col0“>filter{filtername}</td> <td class=„col1“>filter{webbugs}</td> <td class=„col2“ readability=„9“>Angabe des Filters mit der filtername Bezeichnung für Webseiteninhalte, wobei filtername die Verknüpfung zum gleichnamigen Filter ist, der in der user.filter Datei definiert wurde. Beispiel: entfernt in Webseiten den HTML-Code verlinkter, 1×1 Pixel großer Bilddateien, die von fremden Websites geladen werden, um den Besucher zu verfolgen. </td> </tr><tr class=„row4“ readability=„15“><td class=„col0“>client-header-filter{filtername}</td> <td class=„col1“>client-header-filter{hide-tor-exit-notation}</td> <td class=„col2“ readability=„9“>Angabe des Filters mit der filtername Bezeichnung für Kopfzeilen von Internetanwendungen, wobei filtername die Verknüpfung zum gleichnamigen Filter ist, der in der user.filter Datei definiert wurde. Beispiel: entfernt aus der Host: und Referer: Kopfzeile den or-nickname.exit Anteil in URLs bei Verbindungen zu Domains, die mit der MapAddress Tor Funktion an einen bestimmten Tor Ausgang-Router gebunden werden, damit der Server korrekte Host: und Referer: Kopfzeilenwerte erhält. </td> </tr><tr class=„row5“ readability=„12“><td class=„col0“>client-header-tagger{name}</td> <td class=„col1“>client-header-tagger{user-agent}</td> <td class=„col2“ readability=„7“>Angabe der Markierung mit der name Bezeichnung für Kopfzeilen der Internetanwendungen, wobei name die Verknüpfung zur gleichnamigen Markierung ist, die in der user.filter Datei definiert wurde. Beispiel: markiert bzw. „erkennt“ in jeder Anfrage der Internetanwendungen den Inhalt der User-Agent: Kopfzeile und speichert sie für

die späteren Zuordnung zu einem Aktionscontainer per TAG:^User-Agent: UA-Name. </td> </tr><tr class=„row6“ readability=„8“><td class=„col0“>server-header-filter{filtername}</td> <td class=„col1“>server-header-filter{x-httpd-php-to-html}</td> <td class=„col2 leftalign“ readability=„4“>Angabe des Filters mit der filtername Bezeichnung für Kopfzeilen von Servern, wobei filtername die Verknüpfung zum gleichnamigen Filter ist, der in der user.filter Datei definiert wurde. Beispiel: ändert den Wert application/x-httpd-php der Content-Type: Kopfzeile in text/html. </td> </tr><tr class=„row7“ readability=„12“><td class=„col0“>server-header-tagger{name}</td> <td class=„col1“>server-header-tagger{content-type}</td> <td class=„col2“ readability=„7“>Angabe der Markierung mit der name Bezeichnung für Kopfzeilen der Server, wobei name die Verknüpfung zur gleichnamigen Markierung ist, die in der user.filter Datei definiert wurde. Beispiel: markiert bzw. „erkennt“ in jeder Antwort der Server den Inhalt der Content-Type: Kopfzeile und speichert sie für die späteren Zuordnung zu einem Aktionscontainer per TAG:^Content-Type: MIME Datentyp. </td> </tr></table> <h4 id=„filter“>Filter</h4> Filter werden in der /etc/privoxy/user.filter Datei definiert. Folgende Filtertypen können eingesetzt werden: <ul readability=„5“><li class=„level1“ readability=„2“> Filter für Inhalte der Webseiten (FILTER) <li class=„level1“ readability=„2“> Filter für Kopfzeilen der Internetanwendungen (CLIENT-HEADER-FILTER) <li class=„level1“ readability=„2“> Filter für Kopfzeilen der Servern (SERVER-HEADER-FILTER) <li class=„level1“ readability=„2“> Filter zur Markierung von Kopfzeilen der Internetanwendungen (CLIENT-HEADER-TAGGER) <li class=„level1“ readability=„2“> Filter zur Markierung von Kopfzeilen der Server (SERVER-HEADER-TAGGER) <dl class=„code“ readability=„0“><dt>Filter Syntax</dt> <dd readability=„1“> <pre class=„code h“> FILTER-TYP: name Beschreibung

filter-job1 filter-jobN </pre></dd> </dl> Nach Nennung des Filtertyps (z. B. CLIENT-HEADER-FILTER) folgt mit name die Bezeichnung des Filters (z. B. hide-tor-exit-notation). Über die gleiche Bezeichnung mit Nennung des Filtertyps (+filtertyp{name}) in der user.action Datei wird in Aktionscontainern der Filter in der user.action Datei verknüpft und aufgerufen. Zur Information folgt eine kleine, einzeilige Beschreibung des Filterzwecks. Darunter folgen ein- oder mehrmalige Filter-Jobs, die mit PCRE/PCRS Regulären Ausdrücken definiert werden. <dl class=„code“ readability=„1“><dt>Filter Beispiel</dt> <dd readability=„2“> <pre class=„code h“> CLIENT-HEADER-FILTER: hide-tor-exit-notation Removes the Tor exit node notation…

s@^((?:Referer|Host):\s*(?:https?:*)?[^/]*)\.[^\./]*?\.exit@$1@i </pre></dd> </dl> Das Filter-Beispiel ist in der mitgelieferten /etc/privoxy/default.filter Datei enthalten, aus der man in gleicher Weise Filter in die eigene user.filter Datei übernehmen kann, um sie in der eigenen user.action Datei aufzurufen. Zur Definition eigener Filter-Jobs sind die Informationen zu Regulären Ausdrücken und das Filter Files Kapitel im Privoxy Handbuch heranzuziehen. <h3 id=„praktische-beispiele“>Praktische Beispiele</h3> <h4 id=„useraction“>user.action</h4> <pre class=„code“>

Aliase alias allow-all = -block -filter -hide-referer -crunch-incoming-cookies -crunch-outgoing-cookies \ -add-header -hide-if-modified-since -overwrite-last-modified -crunch-if-none-match \ -server-header-filter +forward-override{forward .} -fast-redirects frame-noopt = -crunch-server-header{X-Frame-Options} -server-header-filter{x-frame-security} ## Generell { \ +client-header-tagger{user-agent} \ +change-x-forwarded-for{block} \ +hide-from-header{block} \ +filter{html-annoyances} \ +filter{webbugs} \ +filter{content-cookies} \ +filter{frameset-borders} \ +filter{no-ping} \ +filter{no-targets} \ +filter{kill-resource} \ +crunch-server-header{X-Frame-Options} \ +client-header-filter{hide-tor-exit-notation} \ +server-header-filter{x-httpd-php-to-html} \ +server-header-filter{cache-control} \ +server-header-filter{x-frame-security} \ +limit-connect{80,443,11371} \ +session-cookies-only \ +limit-cookie-lifetime{180} \ +hide-if-modified-since{-60} \ +overwrite-last-modified{randomize} \ +crunch-if-none-match \ } / ## Feedreader, Linkchecker, Download-Manager { \ +client-header-filter{crunch-headers} \ +hide-user-agent{Mozilla/5.0 (Windows NT 6.1; WOW64; rv:12.0) Gecko/20100101 Firefox/12.0} \ +add-header{Accept: text/html,application/xhtml+xml,application/xml;q=0.9,/*;q=0.8} \ +add-header{Accept-Language: en-us,en;q=0.5} \ +add-header{Accept-Encoding: gzip,deflate} \ +hide-referrer{forge} \ +crunch-server-header{X-Frame-Options} \ +crunch-incoming-cookies \ +crunch-outgoing-cookies \ +fast-redirects{check-decoded-url} \ +filter{shockwave-flash} \ +filter{iframes} \ +filter{kill-resource} \ +server-header-filter{x-frame-security} \ } TAG:^User-Agent: Akregator/ TAG:^User-Agent: LinkChecker/ TAG:^User-Agent: Wget/ TAG:^User-Agent: curl/ ## HTTP -> HTTPS (wie HTTPS-Everywhere) { +redirect{s@^http://@https://@} } subdomain.domain.tld ## BREACH / HTTP chunked encoding ## https://community.qualys.com/blogs/securitylabs/2013/08/07/defending-against-the-breach-attack { \ +prevent-compression \ } :443/ ## Ausnahmen {frame-noopt} subdomain.domain.tld {allow-all} p.p config.privoxy.org 192.168.1.1:Port kairaven.de ## Blockierung { +block{Blockiert} +handle-as-empty-document } .doubleclick.net/ www.facebook.com/(extern|plugins)/(login_status|like(box)?|activity|fan)\.php /(.*/)?plugins/wp-likes/ .googleadservices. .google-analytics./ .googlesyndication. </pre> <h4 id=„filteraction“>filter.action</h4> <pre class=„code“> FILTER: html-annoyances Abwehr von HTML Manipulationen. s/(<a\s+href[^>]+resizable=)(['„]?)(?:no|0)\2/$1$2yes$2/igU s/(<a\s+href[^>]+location=)(['“]?)(?:no|0)\2/$1$2yes$2/igU s/(<a\s+href[^>]+status=)(['„]?)(?:no|0)\2/$1$2yes1$2/igU s/(<a\s+href[^>]+scrolling=)(['“]?)(?:no|0)\2/$1$2auto$2/igU s/(<a\s+href[^>]+menubar=)(['„]?)(?:no|0)\2/$1$2yes$2/igU s-</?(blink|marquee).*>–sigU FILTER: webbugs Löscht WebBugs (1x1px Bilddateien). s@<img[^>]*\s(?:width|height)\s*=\s*['“]?[01](?=\D)[^>]*\s(?:width|height)\s*=\s*['„]?[01](?=\D)[^>]*?>@@siUg FILTER: no-targets Filtert target Attribut. s/\starget\s*=\s*(['“]?)_?(blank|new)\1?/ /ig FILTER: shockwave-flash Löscht eingebettete Flash Objekte. s|<object [^>]*macromedia.*</object>|<!– Squished Shockwave Object –>|sigU s|<embed [^>]*(application/x-shockwave-flash|\.swf).*>(.*</embed>)?|<!– Squished Shockwave Flash Embed –>|sigU FILTER: no-ping Löscht ping Attribute in <a> and <area> Tags. s@(<a(?:rea)?[^>]*?)\sping=(['„]?)([^“'>]+)\2([>\s]?)@\ <strong style=„color:white; background-color:red;“ title=„Privoxy removed ping target '$3'“>PING!\n$1$4@ig FILTER: frameset-borders Frames immer mit Rahmen und mit Größenveränderung s/(<frameset\s+[^>]*)framespacing=(['„]?)(no|0)\2/$1/igU s/(<frameset\s+[^>]*)frameborder=(['“]?)(no|0)\2/$1/igU s/(<frameset\s+[^>]*)border=(['„]?)(no|0)\2/$1/igU s/(<frame\s+[^>]*)noresize/$1/igU s/(<frame\s+[^>]*)frameborder=(['“]?)(no|0)\2/$1/igU s/(<frame\s+[^>]*)scrolling=(['„]?)(no|0)\2/$1/igU FILTER: iframes Removes all detected iframes. Should only be enabled for individual sites. s@<iframe.*</iframe>@<!– iframe removed by*

Privoxy's iframe filter –>@Uisg FILTER: kill-resource Killt resource: URIs s|src=„resource:\/\/|src=„http:\/\/|sigU s|href=„resource:\/\/|href=„http:\/\/|sigU s|(<script.*)resource:\/\/(?=.*</script>)|$1http:\/\/|sigU SERVER-HEADER-FILTER: x-httpd-php-to-html Ändert Content-Type: x-httpd-php Kopfzeile auf text/html. s@^(Content-Type:)\s*application/x-httpd-php@$1 text/html@i SERVER-HEADER-FILTER: cache-control Ändert oder löscht Cache-bezogene Server Kopfzeilen. s@^(Cache-Control:)\s.*@$1 private,max-age=3600@i s@^(?:Expires|Pragma|Telligent-Evolution|X-Varnish):.*@@i s@^(Strict-Transport-Security:)\s.*@$1 max-age=2592000 ; includeSubDomains@i SERVER-HEADER-FILTER: x-frame-security Ändert/Setzt X-Frame-Options Header immer SAMEORIGIN s@^HTTP.*@$0\r\nX-Frame-Options: SAMEORIGIN@ CLIENT-HEADER-FILTER: hide-tor-exit-notation Löscht or-nickname.exit aus Host und Referer Kopfzeilen. s@^((?:Referer|Host):\s*(?:https?:)?[^/]*)\.[^\./]*?\.exit@$1@i CLIENT-HEADER-FILTER: crunch-headers Löscht Kopfzeilen für Wechsel auf Firefox UA. s@^(?:Accept|Accept-Language|Accept-Encoding|Accept-Charset|Proxy-Connection):.*@@i CLIENT-HEADER-TAGGER: user-agent Markiert die User-Agent Kopfzeile. s@^User-Agent:.*@$0@i CLIENT-HEADER-FILTER: kill-optout-header Löscht Privacy-by-Obscurity Do-Not-Track Kopfzeilen. s@^(?:X-Behavioral-Ad-Opt-Out|X-Do-Not-Track|DNT):.*@@i </pre> Inhalt | Seite 6 | Seite 8 <h2 id=„verweise-auf-aktuelle-seite“>Verweise auf aktuelle Seite</h2> anonym, pseudonym, anonymisierung, anonymität, proxy, privoxy
 </html>