Technische Durchführbarkeit der Blockierung von Filesharing-Diensten und Hindernisse bei der Beweisführung bei Urheberrechtsverletzungen

Gutachten

Prof. Dr. Dieter Homeister
Dipl.-Inform.
Fakultät für Informatik
Fachhochschule Heidelberg
11.10.2006

Kurzform

Filesharing- bzw. Peer-to-Peer-Programme werden leider auch zum Austausch urheberrechtlich geschützten Materials benutzt. Dennoch ist Filesharing nicht grundsätzlich rechtswidrig, denn es werden auch legale Inhalte transferiert. Für Eltern minderjähriger Kinder, die den elterlichen Internetzugang mitbenutzen, stellt sich neben Haftungsfragen das Problem, ob und mit welchen technischen Maßnahmen sie rechtswidrige Aktivitäten ihrer Kinder verhindern können. Das Sperren der sog. Ports, die Einrichtung von Firewalls oder das Anlegen von Benutzerkonten mit eingeschränkten Rechten scheinen auf den ersten Blick sinnvolle Maßnahmen zu sein. Leider existieren trickreice Programme, die solche Sperren leicht umgehen. Zudem haben die Kinder oft einen Wissensvorsprung vor den Eltern und können die Sperren überwinden. Über WLANs können Unbefugte in Heimnetzwerke eindringen und diese missbrauchen. Der Schutz dagegen ist ebenfalls unerwartet schwierig, wie später beschrieben. Dem Einbrechen in fremde WLANs wird sogar von den Herstellern Vorschub geleistet. Auch ist die Beweisführung der Rechteinhaber oft zweifelhaft, hier kann nicht gänzlich ausgeschlossen werden, dass Unbeteiligte zu Unrecht beschuldigt werden.

Legalität von Filesharing

Filesharing und andere Peer-to-Peer-Dienste (P2P) sind nicht grundsätzlich rechtswidrig. Es kommt in nennenswertem Umfang vor, dass freie Software (s. www.opensuse.org) oder Filme (s. orange.blender.org) auf diesem Wege legal verbreitet werden. Telefoniedienste wie Skype sind ein weiteres Beispiel für legale Peer-to-Peer-Dienste. Auch Medienkonzerne nutzen Peer-to-Peer-Technologien, z.B. EMI Music/Mashboxx.
Filesharing sollte also wie z.B. ein Messer zunächst wertfrei betrachtet werden; erst die Anwender können es positiv (Küchenmesser) oder negativ (Mord) handhaben. Dennoch werden keine Rufe nach Verboten aller Messer laut.
Ein generelles Verbot von Filesharing käme einer Kollektivstrafe nahe, da unschuldige legale Nutzer mitbestraft würden. Würde bei der Internetbenutzung jeder Dienst und jede Technologie gesperrt werden (z.B. e-mail, WWW, FTP), wenn damit möglicherweise gesetzeswidrige Handlungen begangen werden können, müsste in letzter Konsequenz nahezu das gesamte Internet eingestellt werden.

Filesharing ohne spezielle Filesharing-Programme

Der Webbrowser Opera bietet seit der Version 9 die Download-Möglichkeit mit dem BitTorrent-Protokoll an. Diese Möglichkeit nutzt der Autor dieses Gutachtens, um legale Dateien herunterzuladen, z.B. Linux-Distributionen (Sammlungen von freier Software für das Betriebssystem Linux) zu Unterrichtszwecken. Während des Herunterladens werden dabei diese Dateien automatisch auch zum Hochladen angeboten. Opera müsste erst umkonfiguriert werden, um dies zu verhindern. Dafür besteht bei legalen Inhalten keine Notwendigkeit.

Schwierigkeit bei Port-Sperrungen zur Blockierung von Filesharing-Diensten

Es ist also auch ohne Installation eines speziellen Filesharing-Programms möglich, Dateien hoch- oder herunterzuladen.
Das Sperren der beteiligten Ports (s. Anhang 2 zur Erklärung von Ports und IP-Adressen) führt wider Erwarten nicht dazu, den Filesharing-Dienst zu unterbinden.

Ein Versuch mit einem handelsüblichen Heim-Firewall-Router (SMC Barricade 7004) soll das veranschaulichen. Der Dateiaustausch von Opera über das BitTorrent-Protokoll funktioniert auch dann, wenn auf dem Firewall-Router nur die Ports 80, 22 und 443 freigegeben waren. Diese Ports ermöglichen die Internet-Nutzung für verschlüsselte und unverschlüsselte WWW-Seiten sowie die Einwahl von außen auf Rechner hinter der Firewall. Eine Netzwerkanalyse (mittels der Software Etherreal) zeigte, dass Opera/BitTorrent die Ports 58417 und 6881 verwendete, und das Firewall-Gerät dies nicht unterbinden konnte. Der Grund liegt bei den meisten Firewall-Routern gewöhnlich darin, das Firewall-Geräte ebenso wie viele Firewall-Programme nur eingehende Verbindungen über unbekannte Ports sperren. Verbindungen und Ports, die von innerhalb der Firewall initiiert werden, betrachten die Firewalls gewöhnlich als beabsichtigt und lassen sie dann in beiden Richtungen passieren (Bilder 1 und 2).

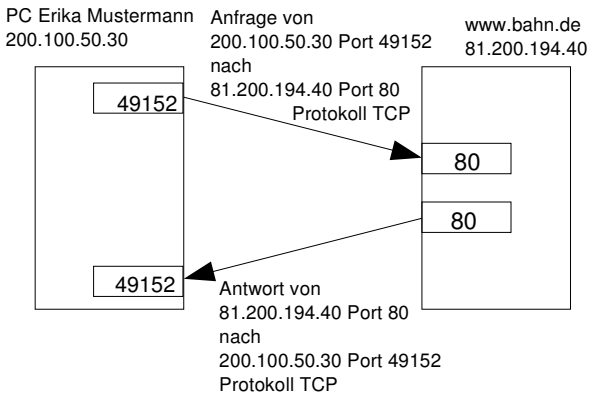

Bild 1: Vereinfachtes Beispiel des Aufrufs der Web-Seite der Deutschan Bahn

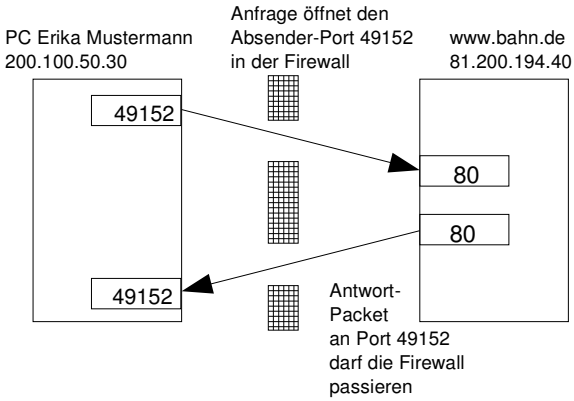

Bild 2: Szenario von Bild 1 mit zusätzlichem Firewall-Router. Ausgehende Datenpackete dürfen hier immer die Firewall passieren.

Kurioserweise lassen sich daher durch Firewall-Router zwar die Downloads (also eingehende Verbindungen) in einigen Fällen blockieren, nicht aber die urheberrechtlich problematischeren Uploads (ausgehende Verbindungen, die von den Firewall-Routern nicht beschränkt werden).
Selbst wenn alle Ports geschlossen werden sind noch alle Uploads und einige Downloads möglich. Das Abblocken ausgehender Verbindungen ist auch nicht Sinn eines Heim-Firewall-Routers, es sollen vorwiegend Angriffe von außen abgeblockt werden.

Die entsprechenden Ports zu schließen ist daher in vielen Fällen für den versierten Heimanwender zwar möglich, aber wenig nutzbringend. Doch selbst wenn es möglich wäre, alle für Filesharing bekannten Ports in beide Richtungen zu schließen, wird damit auch noch kein vollständiger Schutz erreicht.

Verhalten von Filesharing-Programmen bei gesperrten Ports

Viele Filesharing-Programme umgehen Firewalls mit verschiedenen Methoden.
Einige Filesharing-Programme nutzen z.B. wahlweise den Port 80, welcher eigentlich nur für die Kommunikation mit Webservern vorgesehen ist.
Weitere Filesharing-Programme nutzen zufällig ausgewählte Ports.
Vorschläge, alle Ports, über die Filesharing betrieben wird, zu sperren, sind daher nicht praktikabel.
Die Forderung würde nach Sperrung aller relevanten Ports würde daher implizieren, alle Ports zu sperren und damit den Internetzugang komplett unbenutzbar zu machen.
Das Schließen der Ports - insbesondere von Port 80 - würde nebenbei die Benutzung des WWW unmöglich machen, wenn ein- und ausgehende Verbindungen über die Ports gesperrt würden.

Umgehen von Firewalls durch Tunneling

Durch Tunneling-Techniken kann nahezu jede heutige Firewall überwunden werden. Dabei werden die zu versteckenden Datenpakete in andere verpackt, die der Firewall harmlos erscheinen - vergleichbar mit LKWs, die auf Eisenbahnwagen befördert werden. VPNs (Virtual Private Networks) arbeiten nach einem ähnlichen Verfahren, um eine sichere Unternehmenskommunikation über das ungesicherte Internet abzuwickeln.

Zumutbarkeit von Maßnahmen gegen Filesharing durch EDV-Laien

Die Ausführungen mögen etwas kompliziert erscheinen, entsprechen aber der heutigen Realität im Computer-Netzwerkbereich.

Aus den oben genannten Gründen ist heute technisch unmöglich, sämtliche Filesharing-Aktivitäten zu blockieren, es ist lediglich möglich, einige Protokolle zu sperren und Filesharing zu erschweren. Selbst für Computer-Spezialisten ist dies - wenn überhaupt - nur mit hohem Aufwand möglich. Selbst Systemadministratoren in Firmen, Internet-Providern oder Hochschulen gelingt es nicht, alle Filesharing-Aktivitäten komplett zu unterbinden. Sobald eine Lücke geschlossen ist, tauchen geänderte Programme auf und umgehen die Blockierungen auf andere Weise.

Wenn bereits Computerfachleute hier Mühe haben, stellt sich neben den begrenzten Erfolgsaussichten die Frage, wieweit es dann einem technischen Laien zuzumuten ist, zu versuchen, eine Firewall so zu konfigurieren, dass der Datenaustausch durch Filesharing-Programme in jedem Fall unterbunden wird.
Hinzu kommt die Tatsache, dass Kinder und Jugendliche meist in bezug auf Computerkenntnisse (oder z.B. schon beim Programmieren eines Videorecorders) einen deutlichen Wissensvorsprung gegenüber ihren Eltern besitzen.
Es kann also von Eltern nicht realistisch erwartet werden, alle Filesharing-Aktivitäten ihrer Kinder zu unterbinden, es sei denn durch komplette Abschaltung jeglicher Internet-Verbindungen.
Selbst der Versuch einer solche Kontrolle würde einen nicht zumutbaren Aufwand für Eltern darstellen, und dies würde die Möglichkeiten der meisten Eltern übersteigen.

Im Anhang 3 ist exemplarisch die Anleitung zur Installation einer Firewall-Erweiterung

beschrieben, die Filesharing-Programme der Kazaa-Familie (FastTrack, WinMX, OpenNAP) abblockt. Damit sind aber z.B. BitTorrent, eMule etc. immer noch offen. Die Einrichtung überfordert PC-Benutzer ohne tiefe Netzwerk- und Linux- und Englisch-Kenntnisse.

Zumutbarkeit bei Firewall-Router-Geräten

Sog. Firewall-Router sind kleine Rechner (oft nur so groß wie eine VHS-Videokassette), die zwischen den/die Heimrechner und die Internetverbindung eingeschleift werden. Damit kann ein komplettes Heimnetzwerk geschützt werden. Die Konfiguration ist mit Grundkenntnissen über Computernetzwerke meist einfach, ein Laie ohne Netzwerkkenntnisse kann aber dadurch überfordert werden. Meist bieten diese Geräte zwar die Möglichkeit, eingehende Verbindungen zu beschränken, nicht aber ausgehende.
Aus technischer Sicht wäre es eine bessere Lösung, das Heimnetzwerk durch einen dedizierten Firewall-PC, z.B. unter dem Betriebssystem Linux, zu schützen. Diese PC übernähme dann ausschliesslich die Firewall-Funktion. Dann können auch ausgehende Verbindungen kontrolliert werden, z.B. mit der Software iptables und der im Anhang 3 beschriebenen Methode. Neben den zusätzlichen Kosten für diesen PC scheitert dies daran, dass ein Normalbenutzer von der Einrichtung dieses spezialisierten PCs restlos überfordert wäre. Zudem können auf diese Art zwar einige, aber nicht alle Filesharing-Datenpakete blockiert werden.

Zumutbarkeit bei Personal-Firewall-Programmen

Sog. Personal Firewalls sind dagegen Programme, die auf einem einzelnen Rechner laufen und nur genau diesen schützen. Befinden sich mehrere Rechner hinter einem Internet-Zugang, müssen diese separat geschützt werden.
Die Firewall-Software vom Microsoft kann nur eingehende Verbindungen filtern, nicht aber abgehende. Erst für das für 2007 angekündigte Windows Vista ist eine Filterfunktion für abgehende Verbindungen angekündigt.
Programme von Dritthersteller wie Zonealarm können meist auch ausgehende Verbindungen gezielt blockieren. Die Konfiguration von Zonealarm ist nicht trivial und hat schon viele Computerbenutzer zur Verzweiflung gebracht.
Für das Betriebssystem Linux steht mit netfilter/iptables eine sehr mächtige, auch professionell genutzte Firewall-Software zur Verfügung. Die Konfiguration ist jedoch nicht trivial und erfordert tiefe Netzwerk-Kenntnisse (vergl. Anhang 3).

Verhältnismäßigkeit der Beauftragung eines Fachmanns

Die Beauftragung eines Fachmanns (Informatiker etc.) durch die Eltern, um die technisch möglichen Maßnahmen zur Verhinderung von Urheberrechtsverletzungen umzusetzen ist aus folgenden Gründen problematisch.
1. Bei dem Arbeitsaufwand würden Kosten von mehreren hundert EUR entstehen. Das würde die Anschaffungskosten vieler PCs übersteigen.
2. Wegen neuer oder geänderter Filesharing-Programme müsste der Fachmann mehrmals im Jahr die Schutzmaßnahmen anpassen (laufende Kosten).
3. Die Maßnahmen wären aus den o.g. Gründen nur begrenzt wirksam, eine komplette Abschottung kann nicht erreicht werden. Nach heutigem technischen Stand sind keine wirksamen Maßnahmen bekannt, alle Filesharing-Programme zu blockieren.
4. Technisch versierte Kinder könnten praktisch jede Maßnahme umgehen, z.B. durch Neuinstallation des Betriebssystems oder ein Netzwerkkabel am Firewall-Router vorbei.
5. Der Aufwand würde bei Weitem den Aufwand übersteigen, den z.B. eine Anbringung eines Schildes "Es ist verboten, mit diesem und andern PCs Urheberrechte zu verletzen!" am Rechner der Kinder anzubringen (vergl. BGH GRUR 1984, S54/55 - Kopierläden).

Begrenzter Nutzen des Einrichtens von Benutzerkonten durch die Eltern

Das Einrichten verschiedener Benutzerkonten wäre den Eltern zuzumuten. Leider würde diese Maßnahme oder das Sperren bestimmter WWW-Seiten auf Windows-Rechnern nicht dazu führen, dass Urheberrechtsverletzungen unterbunden werden und dennoch ein Zugang zum Internet möglich bleibt. Sobald ein einfacher WWW-Zugang existiert, kann - notfalls über kleine Tricks und Umwege (z.B. http://www.google.com/translate?langpair=en|en&u=www.ZENSIERT.de) gesperrte Seiten aufgerufen werden oder Software oder Musikstücke heruntergeladen werden.
Mit guten Windows-Kenntnissen wäre es immerhin möglich, den Kindern als Benutzern die Installation von Programmen zu verbieten, viele Computerlaien überfordert dies bereits. Diese Maßnahme verpufft jedoch wirkungslos bei Programmen, die von Web-Browsern ohne Installationsprivilegien ausgeführt werden können, z.B. Java- oder Javaskript-Programme.
Technisch versierte Kinder können - oft ohne das Wissen der weniger versierten Eltern - das Betriebssystem komplett neu aufspielen oder neben der "offiziellen" Betriebssystem-Installation ein weiteres Betriebssystem installieren, welches nicht mehr der Kontrolle der Eltern unterliegt. Viele 10- bis 14-Jährige haben schon einen erstaunlich hohen Wissensstand - aus Sicht der Informatik übrigens durchaus erfreulich.

Störerhaftung und Schwierigkeit der Blockierung von WLAN-Trittbrettfahrern

Bei einer WLAN-Anbindung ans Internet kann ein Laie bei vielen WLAN-Geräten nicht zuverlässig verhindern, dass Unbefugte den Internetanschluss mitbenutzen.
Viele WLAN-Geräte werden mit einer Standardeinstellung ausgeliefert, die ohne jegliche Verschlüsselung arbeitet. Ein technischer Laie läuft Gefahr, dies zu übersehen. Doch selbst bei der immer noch weit verbreiteten WEP-Verschlüsselung und der neueren WPA-Verschlüsselung ist keine Sicherheit gegen illegale Mitbenutzung gegeben, da beide Verfahren mit schnellen Rechnern innerhalb von Minuten überwunden werden können. Es kann davon ausgegangen werden, dass dies fast allen technischen Laien nicht bekannt ist. Bei vielen WLAN-Geräten hat der Anschlussinhaber überhaupt keine Möglichkeit, Unbefugten mit hinreichend krimineller Energie die Mitbenutzung der Internetverbindung zu verwehren.
Einige ältere WLAN-Geräte erlauben überhaupt keine Verschlüsselung.
Der Autor dieses Gutachtens benutzt daher generell kein unverschlüsseltes WLAN oder WLAN mit WEP- oder WPA-Verschlüsselung.
Es wäre zu prüfen, inwieweit Hersteller und Vertreiber von WLAN-Geräten fahrlässig handeln, wenn sie ihre Geräte mit der Standardeinstellung ausliefern, die keinerlei Verschlüsselung vorsieht. Die Konfiguration einer verschlüsselten WLAN-Anbindung ist meist aufwändig. Als Grund für die unsichere Standardeinstellung kann daher gemutmaßt werden, dass die Hersteller ihre Verkaufszahlen gefährdet sehen, wenn die Kunden erst komplizierte Vorbereitungsarbeiten vornehmen müssten, bevor sie die Geräte benutzen könnten. Die Hersteller werben auch oft mit der einfachen Inbetriebnahme der Geräte. Nur wenige vorbildliche Hersteller bieten Geräte an (z.B. die AVM FritzBox), die nicht ohne WEP-Verschlüsselung betrieben werden können.
Hier wäre zu prüfen, ob eine Störerhaftung eher gegen WLAN-Hersteller/Vertreiber als gegen private Internet-Nutzer gerechtfertigt ist, insbesondere wenn mit kausalem Zusammenhang argumentiert wird.
Ebenso müsste eine eventuelle Mithaftung der Hersteller der verwendeten Software erwogen werden, z.B. Microsoft. Diese sind kausal an der Rechtsverletzung beteiligt, denn ohne diese Hersteller wäre die Rechtsverletzung nicht möglich.

Bewertung der Zahl angebotener Dateien

Die Zahl der angebotenen Dateien bei Filesharing-Programmen ist aus drei Gründen nicht alleine aussagefähig:

1. Obwohl es unstrittig ist, dass viele Dateien angeboten werden, für die die Anbieter nicht das Urheberrecht besitzen, kann es sich bei jeder einzelnen angebotenen Datei auch um legale Inhalte handeln.
2. Einige Filesharing-Programme zeigen eine Datei bereits als "shared" an, wenn beim Download nur eine leere Datei angelegt wurde. Ein Hochladen ist in diesem Falle technisch unmöglich.
3. Nahezu alle Filesharing-Programme zeigen eine Datei bereits als "shared" an, wenn sich nur Fragmente der Datei auf der Platte des Benutzers befinden. Dann kann auch nur ein Teil eines urheberrechtlich geschützten Werkes zum Hochladen zur Verfügung stehen.

Öffentlichmachung von unvollständigen nach UrhG geschützten Werken

Das Herunterladen einer von einer bestimmten Person angebotenen Datei durch einen Zeugen ist ein schwaches Indiz, da gängige Filesharing-Programme verschiedene Fragmente der Datei von vielen verschiedenen Nutzern herunterladen.
Dabei ist bei Dateien mit vielen Quellen sogar mit hoher Wahrscheinlichkeit davon auszugehen, dass keines der Fragmente von der anvisierten Person stammt.
Hinzu kommt das sog. "Traffic Shaping" einiger Internet-Zugangsanbieter und einiger Betreiber von Weitverkehrsnetzen, die gezielt der Durchsatz unerwünschte Datenpakete von Filesharing-Netzwerken verlangsamen oder ganz blockieren. In der Folge kann es sein, dass Nutzer von Filesharing-Programmen zwar Dateien anbieten, aber eine tatsächliche Übertragung nicht oder nur extrem langsam möglich wäre.
Dies alles könnte auch als ein erfolgloser Versuch einer Öffentlichmachung gesehen werden.
Vielen Nutzern ist auch nicht bewusst, dass bereits während des Downloads von Dateien eine Freigabe in Teilen zum Upload erfolgt.

Es ist also wichtig zu unterscheiden, wer einerseits die Datei potentiell anbietet, und von wem andererseits der tatsächliche Datentransfer stammt, und welche Fragmente der Datei woher kommen.
Bei Downloads mit vielen Filesharing-Programmen ist es nur unter Schwierigkeiten möglich, diese Quelle der Datenpakete eindeutig zu ermitteln.
Manche Filesharing-Programme verbergen diese Information, so dass die exakte Quelle dann nur mittels einer Aufzeichnung von Netztwerk-Analyse-Werkzeugen möglich ist.

Fragwürdigkeit von Beweisen

Nebenbei verwundert auch, dass gelegentlich Beweise für einen durchgeführten Download zugelassen wurden, welche die Zeugen selbst widerrechtlich mittels einer Urheberrechtsverletzung beschafft haben.
Zur Beweiskraft von Papierausdrucken s. Anhang 1.

Anhang 1: Zweifel an der Beweiskraft von Papierausdrucken

Aus Sicht der Informatiker ist es erstaunlich, dass die Echtheit von Papierausdrucken von Computerdateien nicht mit großer Skepsis betrachtet wird, z.B. Ausdrucke von e-mails oder Logdateien.
Textuelle Ausdrucke können vor dem Druckvorgang auf Papier beliebig manipuliert werden, hierzu genügt ein einfaches Textbearbeitungsprogramm (Editor). Große Vorkenntnisse sind dazu nicht erforderlich. Werden die Originaldaten im Computer gelöscht, ist ein Nachweis nahezu unmöglich.
Das zu große Vertrauen in die Papierform könnte darauf beruhen, dass eine unauffällige Manipulation eines Papierausdrucks schwierig ist. Genaugenommen wird aber hier im Falle einer

Manipulation gar nicht die Papierform manipuliert (etwa mit Rasierklinge und Retuschierstift), sondern es wird die Information im Computer manipuliert, noch bevor sie auf Papier gedruckt wird. Bei graphischen Ausgaben wie Bildschirmkopien (Screenshots) ist eine Manipulation vor dem Ausdrucken zwar etwas komplizierter, aber dennoch gut machbar (Bild 3). Allerdings ist hier bei unsauberer Durchführung u.U. ein Nachweis der Fälschung möglich.
Eine weitere Manipulationsmöglichkeit ist das Verstellen der Uhr im aufzeichnenden PC, was z.B. dazu führen kann, dass mittlerweile die (dynamisch vergebene) IP-Adresse an eine andere Person vergeben wurde, wodurch möglicherweise eine falsche Person beschuldigt würde.

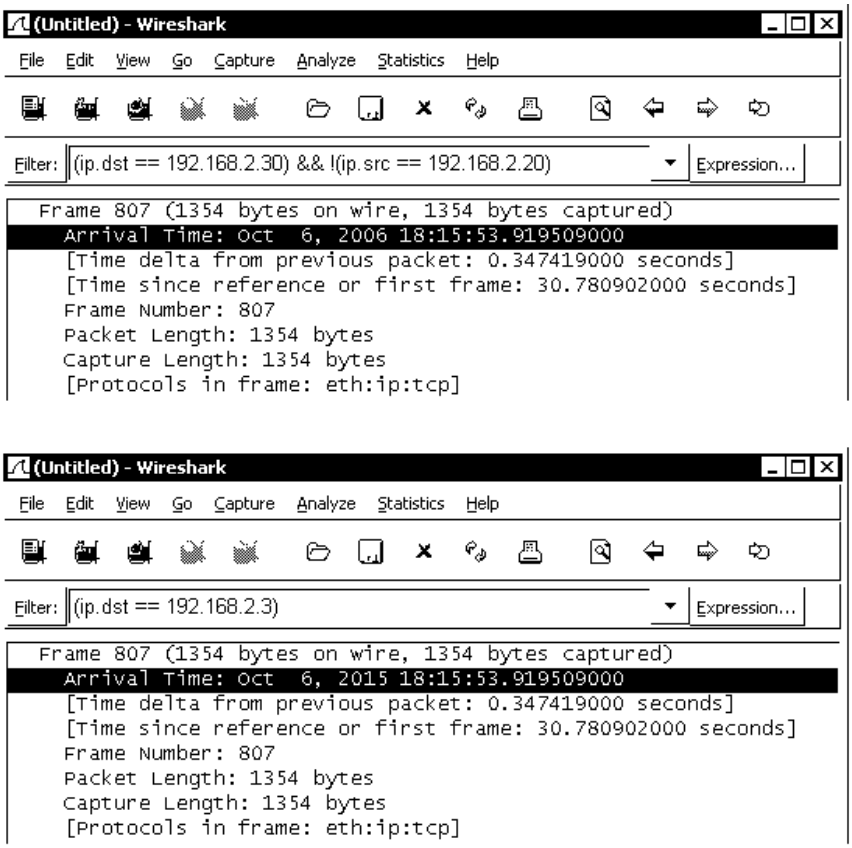

Bild 3: Demonstration einer Manipulation einer realen Bildschirmkopie (Screenshot) eines Netzwerkanalyseprogramms. Insbesondere wurde das Datum in der dunkel markierten Zeile von 2006 auf 2015 verändert. Verwendet wurde das Bildbearbeitungsprogramm GIMP unter Linux - Photoshop unter Windows wäre ebenso geeignet. Der Zeitaufwand betrug hier nur 6 Minuten.

Anhang 2: Erläuterung IP-Adresse und Port

Die IP-Adresse kann in etwa mit einer Postadresse (Staat, Ort, Str., Hausnr.) verglichen werden. Die IP-Adresse identifiziert jeden Rechner im Internet weltweit eindeutig, leider jedoch mit einigen Ausnahmen:
Sog. dynamische IP-Adressen werden z.B. von Internet-Providern verwendet, mit der Folge dass dem Kunden bei jeder Einwahl ins Internet eine andere IP-Adresse vergeben werden kann. Private IP-Adressen, z.B. 192.168.x.x werden in hausinternen Netzen verwendet, sind im externen Internet nicht sichtbar und dürfen mehrfach vergeben werden.
Bei Privatnutzern finden sich solche privaten IP-Adressen hinter sog. Routern, die es erlauben, an einem Internetanschluss mehrere Rechner zu betreiben. Von Außen erscheint dieser

Internetanschluss als ein einzelner Rechner, mit der IP-Adresse des Routers (NAT, Network Address Translation). Eine Zuordnung zum einzelnen Rechner hinter dem Router wird dadurch unmöglich. Sog. Proxy-Rechner können ebenfalls dahinter liegende Rechner verbergen.
In seltenen Fällen kann auch durch Fehlkonfiguration oder bewusste Manipulation ein Rechner die IP-Adresse und die Identität eines anderen annehmen.
Unter Umständen kann die IP-Adresse der eingehenden Pakete gefälscht sein. Diese Manipulation ist jedoch schwierig.

Portnummern (auch Kanalnummern oder Dienstnummern genannt) können in etwa verglichen werden mit einem Briefkasten innerhalb eines Hauses. Die Portnummer dient dazu, bei einer Kommunikation zwischen zwei Rechnern einen bestimmten Dienst bzw. ein bestimmtes Programm anzusprechen. Ohne Portnummern könnten nicht mehrere Programme eines Rechners gleichzeitig Außenverbindungen unterhalten, es käme zu Verwechslungen. Verschiedene Übertragungsmechanismen (Protokolle) können die Portnummern unabhängig voneinander vergeben. Per Konvention ist z.B. Port 80 für HTTP (also für Web-Server) vorgesehen. Technisch ist es jedoch ohne Weiteres möglich, dass auch andere Programme, z.B. Filesharing-Programme diesen Port verwenden. Portnummern von 49152 bis 65535 sind sog. dynamische oder private Ports. Sie werden, mit wechselnden Portnummern, z.B. für die Antwort eines Webservers verwendet. Würden Firewalls diese Ports einfach sperren, wäre die Benutzung des WWW unmöglich.

Anhang 3: Beispiel für die Komplexität einer Firewall-Installation

Installationsanleitung einer Firewall-Erweiterung, die Datenpakete von Filesharing-Programmen der Kazaa-Familie (FastTrack, WinMX, OpenNAP) ausfiltert.
Quelle: http://www.lowth.com/p2pwall/ftwall/docs/INSTALL_v1.php
Mit freundlicher Genehmigung des Autors Chris Lowth

```
Installing ftwall
-----------------

By:   Chris Lowth <chris@lowth.com>
Date: 25 July 2003


The home site for this software is: http://p2pwall.sourceforge.net.

Documentation, FAQs and support forums can all be accessed by this web
site.

=============================================================================
STEP 1 - OBTAINING THE FTWALL SOURCE CODE
=============================================================================

Get the latest ftwall source file tarball from..

        http://p2pwall.sourceforge.net/ftwall

and download to your hard drive, then uncompress and "un-tar" ..

        gunzip ftwall-X.XX.tar.gz
        tar xvf ftall-X.XX.tar
        cd ftwall-X.XX
```

(replace X.XX with the ftwall version number)

```
==========================================================================
STEP 2 - OBTAINING OR BUILDING "LIBIPQ"
==========================================================================

Notes about the libipq library can be found at..
  http://p2pwall.sourceforge.net/ftwall/docs/libipq.php

"ftwall" uses the "libipq" library from iptables. It needs the files
"libipq.a" and "libipq.h". If there is an iptables-devel package for
your linux distribution, then it is likely that this is where the
required files can be found - so install it before trying to compile the
ftwall program. If your system already has these files installed, you
can skip this step.

The RedHat distributions (and, possibly others) dont include this peice
of the iptables software, so you will need to build it for yourself.
It's quite easy to do. Here is what needs to be done.

1. Obtain the iptables sources that match the version installed on your
   system. These can be downloaded from..

       ftp://ftp.netfilter.org/pub/iptables

   You will need the file iptables-VERSION.tar.bz2 (where "VERSION" is
   the version of iptables installed on your firewall).

2. Copy the downloaded file into the ftwall sources directory and
   "unzip" and "untar" (please note: for the build process to work, the
   iptables source directory MUST be placed under the ftwall source
   directory - this location is NOT optional).

       bunzip2 iptables-VERSION.tar.bz2
       tar xf iptables-VERSION.tar

3. "cd" into the iptables sources directory and build the software.

       cd iptables-VERSION
       make
       cd ..

That's it - now you can compile the ftwall software. The libipq library
does NOT need to be installed for the ftwall compilation to work - the
ftwall "make file" will find the files provided that they are located
under the ftwall source tree.


==========================================================================
STEP 3 - BUILDING FTWALL
==========================================================================

This is just a matter of running "make" in the ftwall source directory.


==========================================================================
STEP 4 - INSTALLING FTWALL
==========================================================================

If you are installing on a RedHat 7.x, 8 or 9 system, installation is
done by running the command..

       make redhat_install

For recent Mandrake systems, run..

       make mandrake_install


On other systems, the following manual procedure can be used ...


1. Copy the "ftwall" program to a directory of your choise. /usr/sbin
```

seems like a good option for this. Use the "-a" option of the "cp"
command to ensure that the ownership and modes are carried over
during the copy process.

```
cp -a ftwall /usr/sbin/ftwall
```

2. If your system uses the RedHat daemon startup logic of placing files
in the directory /etc/init.d then copy the "ftwall.redhat.init" file
to /etc/init.d/ftwall and edit it to change the configuration options.

```
cp -a ftwall.redhat.init /etc/init.d/ftwall
chkconfig --add ftwall
vi /etc/init.d/ftwall
```

Note: on some recent distributions, the /etc/init.d directory is
"sym-linked" to /etc/rc.d/init.d.

If your system does not use this mechanism, then visit the p2pwall
web site or "contributions" forum to see whether anyone has
contributed instructions or code suitable for your system. If not -
then cut your own and send news of your progress to the forum.

The "contributions" forum can be found at..

```
https://sourceforge.net/forum/forum.php?forum_id=294611
```

```
========================================================================
STEP 5 - CONFIGURING IPTABLES
========================================================================

A detailed discussion of these rules and why they are needed is included
in the documentation available on the project web site.
 at -- http://p2pwall.sourceforge.net/ftwall/docs/iptables_rules.php

*********************************************************************
*                                                                   *
*   WARNING #1 - PUT THE "QUEUE" RULES AT THE END OF THE CHAINS     *
*                                          ===                      *
*                                                                   *
*********************************************************************

The "QUEUE" target for iptables is a "terminating" target - in other
words; any packets passed to it STOP traversing the rules, but are
either accepted (in which case they are transmitted) or dropped (in
which case they vanish completely) and are not processed by further
rules in the chain. For this reason, you should probably place the rules
at the END of the chains.

*********************************************************************
*                                                                   *
*     WARNING #2 - ALLOW UDP PACKETS TO GET THROUGH TO FTWALL       *
*                                                                   *
*********************************************************************

Ftwall makes use of the contents of UDP packets to identify the
fast-track clients and peers. If these packets dont get through to the
software then it cannot identify the connections unless you have the
"string match" logic in place and the client attempts a download. Please
ensure that UDP packets can flow through your network and other FORWARD
chain rules - but DONT ignore warning #1. If ftwall is not blocking
fast-track traffic, try running it with logging of UDP packets turned
on, and make sure that it is dropping UDP packets withn a few seconds of
a fast-track client being started.

If you want to be more specific about the UDP packets you allow through,
then you can use the fact that the actual packets needed by ftwall are
contain the string "KaZaA". You can use the "string" match to set up
the relevant filters.


-- NB: ftwall's filter is focussed on "outbound" traffic --
```

These notes and the ftwall software add the feature of preventing Fast Track clients in the HOME network from establishing links to peers on the PUBLIC internet ("outbound" connections). It is assumed that you already have iptables rules in place to prevent TCP/IP connections FROM the public internet TO workstations on the HOME network ("inbound" connections).

-- A note about port 1214 --

You may be tempted to set up rules that block port 1214. This might seem to make sense if you know that this is the port that old Fast Track clients used to use. The new versions make use of (almost) any port that they can "break through" on. Since the ftwall logic works best when it gathers most information about addresses that refer to Fast Track peers, it is actually beneficial to allow the port 1214 packets to be queued to ftwall rather than being blocked directly by iptables. If you have "port 1214 block" rules in place - please remove them when deploying ftwall. If you wish to leave explicit rules to block port 1214 traffic from the public internet (inbound connections), then you may do so - this has no impact on ftwall.

-- Rule for queuing Forwarded UDP packets --

All UDP packets forwarded FROM hosts in the "Home" network TO the public internet must be passed to the ftwall program using the iptables "QUEUE" target. The following rule is one way that this can be done, assuming that the firewall interface connected to the home network is "eth0" and the home network is 192.168.0.0/24.

```
iptables -A FORWARD -p udp -i eth0 -s 192.168.0.0/24 -j QUEUE
```

It is *very* important that only OUTBOUND udp packets are passed to the queue. On no account should any of the UDP packets coming from the public internet back into the home network be passed to the QUEUE - if they are, the ftwall logic will break down and the software will block the wrong traffic.

If there are multiple "workstation" subnets in the home network, then you need to ensure that they are all covered by the rule(s).

If you have servers or other systems that can be guaranteed NOT to run Fast track software, then it would be an advantage to use IPtables rules to prevent traffic from those systems passing through ftwall. This boosts performance and reduces the risk of false possitive classifications.

-- Rule for queuing Forwarded TCP/IP SYN packets --

All TCP/IP SYN packets forwarded FROM hosts in the "Home" network TO hosts on the public internet must also be passed to ftwall via the "QUEUE" target. Assuming that the interface on the home network is "eth0", and the home subnet is 192.168.0.0/24, then the following command achieves this end..

```
iptables -A FORWARD -p tcp -i eth0 -s 192.168.0.0/24 --syn -j QUEUE
```

The same warnings given for the forwarded UDP packets (above) apply to TCP/IP SYNs. You must ensure that ONLY and ALL the relevant packets are passed. Further, you must ensure that they are passed to the queue from EXACTLY the same set of hosts as the UDP packets. The logic will not work if there is a mismatch between these two rules.

-- Rule for queuing responces to UDP probe packets --

The ftwall program periodically sends UDP packets to identified Fast track clients, and these are returned via the iptables INPUT chain. Assuming that the interface connected to the home network is "eth0", and the home subnet is 192.168.0.0/24 then the following rule can be used..

```
   iptables -A INPUT -p udp -i eth0 -s 192.168.0.0/24 -j QUEUE

The same warnings given for the forwarded UDP and TCP/IP SYN packets
(above) applies to these "INPUT" UDP packets. You must ensure that ONLY
and ALL the relevant packets are passed. Further, you must ensure that
they are passed to the queue from EXACTLY the same set of home network
hosts as the UDP and TCP/IP SYN packets. The logic will not work if
there is a mismatch between these two rules.


-- Rule for TCP/IP data packet match --

If you wish to configure the safety net of filtering file transfers on
the basis of the contents of HTTP-like headers employed (see the
documentation on the web site for information on why you may wish to do
this), then you need to pass data packets into ftwall as well. Since
passing ALL data packets is likely to have a negative impact on the
performance of your firewall, you may wish to pre-filter the packets
using the the experimental "string" module of iptables.

It is likely that your kernel does NOT include this module, and so you
will need to add it. This requires a kernel re-build.

A description of the pros and cons of using the string match, see..
  http://p2pwall.sourceforge.net/ftwall/docs/why_string_match.php

The process for adding this logic is described at..
  http://p2pwall.sourceforge.net/ftwall/docs/adding_string_match.php

Once the module is in place, the following rule is required to pass the
relevant packets to ftwall (adjust the interface and subnet
specifications to match your system/network)..

   iptables -A FORWARD -p tcp -i eth0 -s 192.168.0.0/24 \
       -m string --string X-Kazaa -j QUEUE

If you dont wish to install the module (and rebuild the kernel), but
wish to filter data packets anyway and are happy to live with the
performance hit, then you can simply remove the "--syn" flag from the
TCP/IP SYN packet rule (above). This will cause ALL TCP/IP packets to be
queued.


On systems that use the /etc/sysconfig/iptables file to hold iptables
configuration, the easiest way to configure these rules is to type the
iptables commands in by hand (as root) and then run..

   service iptables save
--------
```