Network Architectures
and Services
NET 2009-04-1

FI-Seminar SS 2009

Proceedings of the Seminar
Future Internet (FI)
Sommer Semester 2009

Munich, Germany, 16. - 17.04.2009

Editors

Georg Carle, Corinna Schmitt

Organisation

Lehrstuhl Netzarchitecturen und Netzdienste (I8),
Fakultät Informatik, Technische Universität München

Technische Universität München





Network Architectures
and Services
NET 2009-04-1

Seminar FI

SS 2009

Proceedings zum Seminar
Future Internet (FI) SS2009

Sommer Semester 2009
Vorträge im Zeitraum 16. – 17.04.2009

Editoren: Georg Carle, Corinna Schmitt

Seminar organisiert durch den
Lehrstuhl Netzarchitekturen und Netzdienste (I8),
Fakultät für Informatik,
Technische Universität München

Technische Universität München

Seminar FI SS 2009
Seminar “Future Internet”
Sommer Semester 2009
Chair for Network Architectures and Services (I8)
Technische Universität München

Editors:

Georg Carle
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching bei München, Germany
E-mail: carle@net.in.tum.de
Internet: http://www.net.in.tum.de/~carle/

Corinna Schmitt
Lehrstuhl Netzarchitekturen und Netzdienste (I8)
Technische Universität München
D-85748 Garching bei München, Germany
E-mail: schmitt@net.in.tum.de
Internet: http://www.net.in.tum.de/~schmitt/

Cataloging-in-Publication Data

Seminar FI SS 2009
Proceedings of Seminar “Future Internet”
Sommer Semester 2009
München, Germany, 16.04.2009 – 17.04.2009
Georg Carle, Corinna Schmitt (Eds.)
ISBN: 3-937201-06-8

ISSN: 1868-2634 (print)
ISSN: 1868-2642 (electronic)
Network Architectures und Services NET 2009-04-01
Series Editor: Georg Carle, Technische Universität München, Germany

Vorwort

Wir präsentieren Ihnen hiermit die Proceedings zum Seminar “Future Internet” (FI) aus dem Sommer Semester 2009 der Fakultät Informatik der Technischen Universität München.
In diesem Seminar wurden Vorträge zu allgemeinen und interessanten Themen im Forschungsbereich Futur Internet vorgestellt. Die folgenden Themenbereiche wurden von den Vortragenden abgedeckt:

- Verbesserte Performanz und Programmierbarkeit in Netzwerksystemen
- DoS-Angriffe
- Moderne Botnetze
- Schlauere Navigation durch Mobilfunk?
- BitTorrent
- Trusted Platform Module (TPM)
- Zero Platform Networking
- PSerPool: Standardisierte Serverreplikation
- Wuala

Wir hoffen, dass diese Beiträge, die einen aktuellen Querschnitt durch Forschungen im Bereich Internet darstellen, informativ sind und zu einer weiteren Beschäftigung mit den Themen anregen.
Falls Sie weiteres Interesse an unseren Arbeiten haben, besuchen Sie doch bitte unsere Homepage http://www.net.in.tum.de für weitere Informationen.

München, April 2009

Georg Carle

Corinna Schmitt

Preface

We are very pleased to present you the interesting program of our graduate-level seminar on “Future Internet” (FI) 2009.
In this seminar we deal with common and interesting topics in current research tasks in the field of future internet. Due to the fact that this seminar was hold in German, the contributions in the proceedings are also in German. The following topics are covered by this seminar:

- Optimized performance and programmability in networks
- DoS attacks
- Modern botnets
- Clever navigation by mobil communication?
- BitTorrent
- Trusted Platform Module (TPM)
- Zero Platform Networking
- PSerPool: Standardized Server replication
- Wuala

We hope that these contributions, which represent selected topics of Internet research, inspire for further interest into these topics.
If you are more interested in our work, please visit our homepage http://www.net.in.tum.de for more information, also including offers for thesis projects and job opportunities.

Munich, April 2009

Seminarorganisation

Lehrstuhl für Netzarchitekturen und Netzdienste (I8)

Lehrstuhlinhaber

Georg Carle,
Technische Universität München, Germany

Seminarleitung

Corinna Schmitt, *Technische Universität München*, Germany

Betreuer der Beiträge

Tobias Bandh, *Technische Universität München, Wiss. Mitarbeiter I8*
Benedikt Elser, *Technische Universität München, DFG Emmy Noether Research Group Member*
Marc Fouquet, *Technische Universität München, Wiss. Mitarbeiter I8*
Holger Kinkelin, *Technische Universität München, Wiss. Mitarbeiter I8*
Andreas Müller, *Technische Universität München, Wiss. Mitarbeiter I8*

Kontakt

{carle,schmitt,bandh,elser,fouquet,kinkelin,mueller}@net.in.tum.de

Seminar-Homepage

http://www.net.in.tum.de/de/lehre/ss09/seminare/

Inhaltsverzeichnis

Verbesserte Performanz und Programmierbarkeit in Netzwerksystemen

Florian Birnthaler
Betreuer: Benedikt Elser
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: birnthal@in.tum.de

***Kurzfassung*— Es wird eine Prozessorarchitektur für Netzwerkgeräte beschrieben, bei der die für Speicherbausteine zur Verfügung stehende Fläche auf der Platine beliebig an Hardware-Threads oder Caches zugewiesen werden kann. Dadurch kann sich ein solcher Netzwerkprozessor selbstständig an die Programme, die auf ihm laufen, anpassen und so eine optimale Performanz erzielen.**

***Schlüsselworte*— Memory Bottleneck, Prozessorarchitektur, Multithreading, Caching, Netzwerkverkehr**

I. Einleitung

Das so genannte 'Memory Bottleneck' ist ein signifikantes Problem für Paketverarbeitungsplatformen, da der Netzwerkverkehr über das Internet in den letzten Jahren stark zugenommen hat [3] und Anwendungen zur Paketverarbeitung – wie Virenscanner und Software zur Netzwerküberwachung und Intrusion Detection – immer umfangreicher werden.

Die typischen Ansätze, dieses Problem zu umgehen, sind Hardware-Multithreading, Caching, Algorithmen, die die Anzahl der Speicherzugriffe auf den Sekundärspeicher minimieren und kleinere 'Hardware-Hacks'. Außerdem wird für bestimmte Anwendungen oft maßgeschneiderte Hardware verwendet.

Eine Alternative hierzu ist eine Prozessorarchitektur, welche die Anzahl der verwendeten Threads und den Cache den jeweiligen Ansprüchen entsprechend anpassen kann. Diese Architektur, bei der, neben hoher Performanz, ein weiteres wichtiges Ziel eine einfache Programmierung ist, wird in dieser Arbeit näher vorgestellt.

Nach einer kurzen Erklärung zum Problem des 'Memory Bottleneck', wird dabei zunächst auf die Vorteile, die eine solche Architektur bieten würde, eingegangen. Danach wird der Aufbau eines solchen Prozessors vorgestellt, um daraufhin anhand des von ihm verwendeten Algorithmus deutlich zu machen, wie sich dieser an die Programme und den spezifischen Paketstrom anpassen kann. Schließlich wird die Performanz mit üblichen Netzwerkprozessoren, wie dem Intel IXP2800 [5], verglichen und ein Fazit gezogen.

II. Grundlagen – Das 'Memory Bottleneck'-Problem

Heutige Mikroprozessoren arbeiten mit sehr hohen Taktraten, die die des Hauptspeichers bei weitem übertreffen. [4, S.292f] Das bedeutet, dass der Prozessor auf Daten, die er vom Hauptspeicher anfordert, viele Taktzyklen lang warten müsste. Wenn, aufgrund der Struktur des gerade abzuarbeitenden Programms, während dieser Wartezeit keine anderen Befehle abgearbeitet werden können, wird sehr viel Rechenkraft verschwendet.

Um dieses 'Memory Bottleneck' zu umgehen, wird eine Speicherhierarchie eingeführt, die nahe am Prozessor einen kleinen, schnellen Speicher (Register und Level 2 Cache) vorsieht und weiter nach außen immer größeren, jedoch auch langsameren Speicher (Level 3 Cache, Hauptspeicher, Festplatte). Nun wird versucht, Daten, die in den nächsten Taktzyklen vom Prozessor benötigt werden, in einem Speicher möglichst nahe am Prozessor bereitzuhalten, damit dieser bei deren Anforderung nicht lange auf sie warten muss. Dies geschieht dadurch, dass Teilbereiche der größeren Speicher mit Hilfe einer Caching-Strategie (z.B. direct mapped cache [4, S.295ff] in die kleineren Speicher kopiert werden. Benötigt nun der Prozessor Daten, die in diesem Moment auch in einem prozessornahmen Cache gespeichert sind ('cache hit'), so muss er diese nicht, wie bei einem 'cache miss', vom langsamen, größeren Speicher anfordern und darauf warten, sondern kann gleich die Daten aus dem Cache verwenden. Dadurch entsteht ein enormer Performanzgewinn.

III. Warum ein sich anpassenden Prozessor?

Auf Netzwerkprozessoren werden viele verschiedene Programme ausgeführt, die jeweils unterschiedliche Eigenschaften bezüglich Parallelismus und Daten-Lokalität haben. Während es für manche Anwendungen vorteilhaft ist, möglichst viele Threads gleichzeitig arbeiten zu lassen, laufen andere Programme performanter, wenn stattdessen der Datencache größer ist. Es ergibt sich also für jede Anwendung ein optimales Verhältnis zwischen Größe des Datencaches und Anzahl der Threads, wobei wir davon ausgehen, dass auf dem Chip nur begrenzt Platz ist, um diese Elemente unterzubringen. Diese Verhältnisse schwanken zum Teil sehr stark. [1]

Programme, die man zur ersten Gruppe zählen kann sind beispielsweise *cast* (verwendet zur Verschlüsselung) und *md5* (bildet Prüfsummen über Pakete). Da diese Anwendungen das ganze Paket verarbeiten profitieren sie nicht von Caches. Die

Latenzzeit, die benötigt wird, um die Daten zum Prozessor zu bringen, kann also nicht verringert werden. Allerdings können mehrere Threads jeweils ein anderes Paket bearbeiten und so die Abarbeitung beschleunigen.

Zur zweiten Gruppe zählt man Programme wie *stream* oder *portscan*. Hier ist es oft wegen Locks auf bestimmte Datenbereiche nicht gut möglich, parallel zu arbeiten, da sich die Threads gegenseitig blockieren. Hier ist also ein größerer Cache besser.

Die meisten der getesteten Programme liegen zwischen diesen beiden Extremen, jedoch unterscheiden sich auch diese bei den Anforderungen für Cache oder Threads merkbar.

Das Problem ist, dass das Verhältnis von Threads und Cachespeicher bei den eingesetzten Netzwerkprozessoren unveränderbar ist, da diese Verteilung zur Entwicklungszeit fest bestimmt wurde. Für Entwickler von Programmen wie oben, die den Netzverkehr verarbeiten, entsteht dadurch das Problem, dass die Programme die Hardware nicht optimal ausnutzen. Deswegen stecken sie viel Zeit in die Anpassung des Programms an die Eigenschaften der Hardware durch Veränderung der verwendeten Algorithmen und kleinere 'Hardware-Hacks'. Dadurch können die Programme aber auch schwerer wart- und erweiterbar werden.

Wenn sich allerdings das Verhältnis von Threads zu Cachespeicher automatisch an das Programm anpassen kann, so muss man diesen zusätzlichen Aufwand nicht mehr betreiben. Anwendungen und Protokolle können dadurch viel schneller entwickelt werden. Vor allem aber wird die Programmierung von Software, die die Hardware optimal ausnutzt viel einfacher.

IV. Aufbau des Prozessors

Möglich ist ein solcher Prozessor, der Threads bzw. Cache mehr oder weniger Platz auf der Platine zuweisen kann, dadurch, dass Caching und Multithreading die gleiche Ressource verwenden, und zwar schnellen Speicher, der nahe am Prozessor liegt.

Während Caching diesen Speicher verwendet, um Daten bereitzustellen, die voraussichtlich als nächstes benötigt werden, speichert Multithreading in ihm Register, die für den jeweiligen Thread wichtig sind.

Bei dem hier besprochenen Prozessor kann dieser Speicher beliebig auf Threads und Caches verteilt werden. Damit dessen Performanz später besser verglichen werden kann, orientiert sich die Gesamtgröße des Chips an dem des Intel IXP2800. Auf dieser Fläche können neun Prozessorkerne verteilt werden, von denen jeder den in Abb. 1 gezeigten schematischen Aufbau hat.

Dort sieht man auf der linken Seite den groben Aufbau, der aus einem Multiport-Register-Cache (MRC) und einem primären Datencache, der sowohl Thread-spezifische Register als auch normale Cache-Daten aufnehmen kann, besteht. Dieser MRC ist notwendig, da eine möglichst geringe Latenzzeit, eine hohe Bandbreite und Kapazität nicht gleichzeitig erreichbar sind. Deswegen besteht der Speicher aus einem kleinen, schnellen Speicher mit hoher Bandbreite (dem MRC) und einem großen, langsameren Speicher mit geringerer Bandbreite.

Die Herausforderung beim Entwickeln des Prozessors besteht nun darin, zu erreichen, dass der schnellere Speicher nach außen hin so wirkt als hätte er die Kapazität des langsameren Speichers. Traditionelle Lösungsansätze für solche Speicherhierarchien sind Register Caching (beispielsweise mittels 'last recently used') und Double Buffering. Bei Double Buffering hat man zwei Datenpuffer: Auf einem arbeitet der gerade aktive Thread, während der andere mit den Daten des nächsten Threads gefüllt wird; bei Threadwechsel werden dann einfach beide Puffer gewechselt und dem Thread stehen sofort alle benötigten Daten zur Verfügung.

Um allerdings alle relevanten Register eines Threads im Voraus vollständig zu laden zu können, würde man einen MRC benötigen, der auf dem Chip zu viel Fläche einnehmen würde. Deswegen wird eine Kombination aus beiden Ansätzen zusammen mit einem System zur Vorhersage der als nächstes benötigten Daten verwendet, was im rechten Teil von Abb. 1 dargestellt ist:

Der Prozessor verwendet zwei Registerspeicher (in der Abbildung wird nur einer (S-file) gezeigt), um Daten bzw. Threadregister zu cachen. Während einer der beiden Speicher verwendet wird, wird der andere mit Daten für den nächsten Thread gefüllt. Welche Daten das sind wird vom 'Predictive Prefetcher' vorhergesagt. Dies geschieht mit Hilfe des Programmzählers des Threads, der als nächster dran ist. Denn zu diesem Programmzähler wird jedes mal wenn der Thread zurück in den Data Cache geschrieben wird ein Bitmuster mit abgespeichert, das bestimmt, welche Register bei der nächsten Aktivierung des Threads im Voraus geladen werden sollen. Dieses Bitmuster wird durch zusätzliche Kontrollbits (V für 'valid', R für 'read', M für 'modified') in den Registerspeichern bestimmt: Wenn für ein Register das R-bit gesetzt ist (d.h. es wurde vom Thread daraus gelesen), so wird es vorgeladen.

Die miss rate, die bei den Vorhersagen erzielt wird ist bei allen getesteten Anwendungen kleiner als 2 Prozent – meist sogar unter 1 Prozent. [1] Die Vorhersagen sind also sehr zuverlässig. Die Anzahl der richtig gecachten Register weicht meist nur um zwei bis drei (von insgesamt bis zu 18 Registern) von der einer optimalen Auswahl ab.

V. Algorithmus zur Anpassung des Prozessors

Der Algorithmus zur Anpassung des Cache-Thread-Verhältinsses läuft während dem Betrieb, benötigt aber dennoch so wenig Ressourcen, dass er das System nicht belastet. Deswegen wird er auch später bei den Performaztests vernachlässigt.

Der Algorithmus basiert auf drei Beobachtungen:

1) Von einer Festlegung des Thread-Cache-Verhältnisses zur Entwicklungszeit ist abzuraten, da man so nicht auf Änderungen im Traffic reagieren kann und Wartung und Erweiterungen von Systemen, die darauf beruhen, zu umständlich sind.

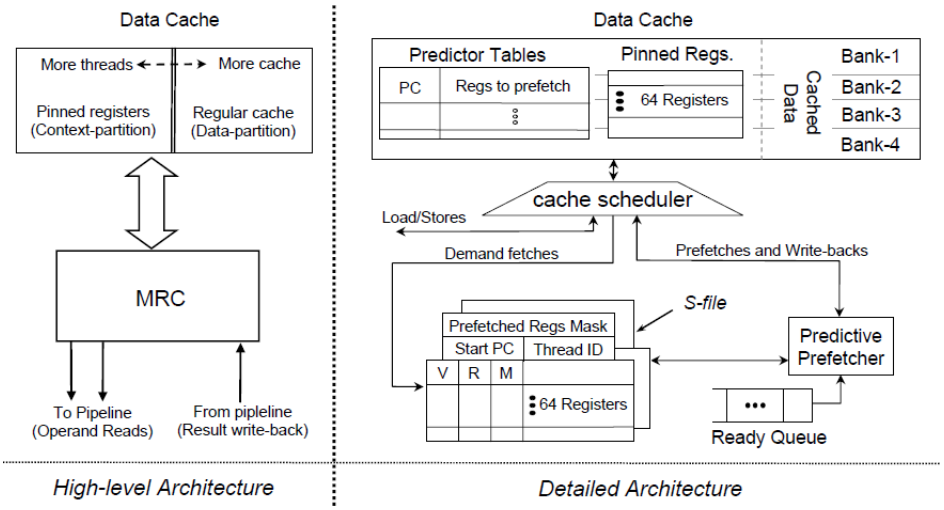

Abbildung 1. Architektur des Prozessors (Quelle: [1])

2) Wenn die Anzahl der Threads erhöht wird steigt die Performanz zunächst, fällt aber ab einem bestimmten Punkt wieder ab. Das liegt zum einen daran, dass die Cache-Kapazität im selben Maße kleiner wird, wie die Speicherzellen für die Threads zunehmen, und deswegen mehr cache misses auftreten. Ein weiterer Grund dafür ist, dass bei steigender Anzahl von Threads diese auch stärker um benötigte Seiten aus dem Speicher konkurrieren, was die Bandbreite des Caches negativ beeinflusst. Außerdem werden durch die höheren miss Raten auch öfter die Threads gewechselt. Denn bei jedem cache miss wird der Thread gewechselt, weil der gerade aktive Thread lange auf die angeforderten Daten warten muss. Dadurch gehen jedesmal zwei Taktzyklen (Dauer eines Threadwechsels) verloren.
3) Die Verringerung der Threadanzahl – und damit die Terminierung eines Threads – verringert die Performanz für eine kurze Zeit (einige hundert Pakete lang). Das liegt an folgendem: Wenn ein Thread entfernt wird, müssen alle Speicherzeilen, die von diesem Thread für die Vorhersage der als nächstes benötigten Speicherzellen verwendet werden, freigegeben werden. Allerdings können diese nicht ermittelt werden, weshalb diese 'Vorhersagespeicherzellen' von allen Threads freigegeben werden müssen. Deswegen funktionieren die nächsten Vorhersagen nicht und die anderen Threads müssen die Speicherzellen für ihre Vorhersagen nochmal neu reservieren. Dies führt zu einer Erhöhung der Bearbeitungszeit von Paketen um 6%, die aber nach den erwähnten wenigen hundert Paketen wieder verschwunden ist.

Aus diesen Beobachtungen folgt, dass die Anpassung zur Laufzeit stattfinden muss. Um die optimale Anzahl von Threads zu finden wird eine einfache lineare Suche verwendet und um seltener Threads terminieren zu müssen, wird bei der Suche nach unten immer um zwei Threads herunter geschaltet (anstatt wie bei der Suche nach oben um einen Thread).

Der Algorithmus ist in Abb. 2 im Pseudocode dargestellt.

Notation:
N_{max}: Max number of threads
N_c: Current number of threads
$\lambda[n]$: Throughput with n threads

```
1: for n = N_c + 1 to N_max do
2:    λ[n] = trialRun(n)
3:    if (λ[n] < λ[n − 1]) then
4:       break
5:    end if
6: end for
7: if ((n − 1) > N_c) then
8:    return (n − 1) /* increasing works */
9: end if
10: /* increasing does not work */
11: for n = N_c − 2 down-to 1 with step (-2) do
12:    λ[n] = trialRun(n)
13:    if (λ[n] < λ[n + 2]) then
14:       break
15:    end if
16: end for
17: /*also try the last thread skipped*/
18: λ[n + 1] = trialRun(n + 1)
19: return (λ[n + 1] > λ[n + 2])?(n + 1) : (n + 2);
```

Abbildung 2. Algorithmus zur Anpassung des Thread:Cache-Verhältnisses (Quelle: [1])

Hier wird nach und nach immer ein Thread zusätzlich hinzugefügt und der Durchsatz dieser Konfiguration mit der vorherigen verglichen. Dies geschieht so lange, bis keine Durchsatzsteigerung durch einen zusätzlichen Thread mehr erreicht werden kann.

Wenn allerdings nicht einmal das Hinzufügen eines einzigen Threads zur momentanen Anzahl von Threads einen Gewinn darstellt, dann werden nach und nach zwei Threads abgebaut, wodurch ein Performanzgewinn zu erwarten ist. Auch dies

wird so lange gemacht, bis es keine Durchsatzsteigerung mehr gibt.

Als Resultat bekommt man die ideale Anzahl an Threads.

Nachdem dieses erste Ergebnis gefunden ist, wird nicht mehr der ganze Algorithmus ausgeführt. Stattdessen werden nur noch kleine Anpassungen anhand von Veränderungen in der Balance, die während der Laufzeit verfolgt wird, gemacht. Diese Anpassungen finden alle 100.000 Pakete statt und erfordern vernachlässigbar geringe Prozessortätigkeit.

VI. Performanzvergleich mit anderen Prozessoren

Die Anwendungen, für die die Performanz des Prozessors getestet wird, implementieren folgende Funktionen:

- Integritätsprüfung ankommender Pakete
- Klassifizieren der Pakete
- Paketverarbeitung
- Scheduling von ausgehenden Paketen

Tabelle I zeigt eine Übersicht dieser Programme. Eine genaue Beschreibung zu diesen findet man in [2, S.19ff].

Da ein aussagekräftiger Performanztest wegen der enormen Anzahl an verschiedenen zu testenden Konfigurationen (mehrere Hunderttausend verschiedene Konfigurationen) nicht realisierbar ist, wurde der Simulator Simplescalar [6] verwendet, um repräsentative Programm-Traces zu erzeugen.

Für die Tests wurden Paket-Traces aus verschiedenen Stellen im Internet verwendet, die verschiedene Randbedingungen für die Programme bieten. So kommen beispielsweise die Pakete vom ANL-Trace (Verbindung zwischen Argonne National Lab und ihrem ISP) vom Rand eines Netzwerkes (des ANL-Netzwerkes) und haben damit ähnliche Header, was eine gute Voraussetzung für die Verwendung von Caches ist.

Auf der anderen Seite stammen die Pakete von MRA (aus [7]) aus einem Netzwerk-Zentrum und haben damit unterschiedlichste Header, wodurch hier mit Multithreading die besseren Ergebnisse erzielt werden können.

Im Vergleich zum IXP2800 [5], einem aktuellen und leistungsfähigen Netzwerkprozessor erreicht der vorgestellte Prozessor in 26% der Anwendungen einen um über 300% höheren Durchsatz. Verglichen mit einem Netzwerkprozessor, der zwar ein festes - jedoch optimales - Thread-Cache-Verhältnis hat, hat der sich anpassende Prozessor in über einem Drittel der Anwendungen einen um 60% höheren Durchsatz. Diese Resultate werden in Abb. 3 gezeigt. Ein grauer Balken gibt hier an, welcher Anteil der getesteten Programme einen bestimmten Durchsatz erreichen (beispielsweise erreichen 40% der Programme einen Durchsatz der 90-110% von dem des IXP2800 entspricht). Die obere Kurve gibt an, welcher Anteil der Programme *mindestens* einen bestimmten Durchsatz erreichen.

Zuletzt wird noch ein Vergleich (vergl. Abb. 4) zu einem hypothetischen Prozessor hergestellt, der nicht nur das Verhältnis von Cache zu Thread auf dem Chip verändern kann, sondern auch die Anzahl der Rechenkerne. Er würde also bei einer gegebenen Chipfläche ein für die Anwendung optimaler Prozessor sein. In 63% der Anwendungen ist der entwickelte

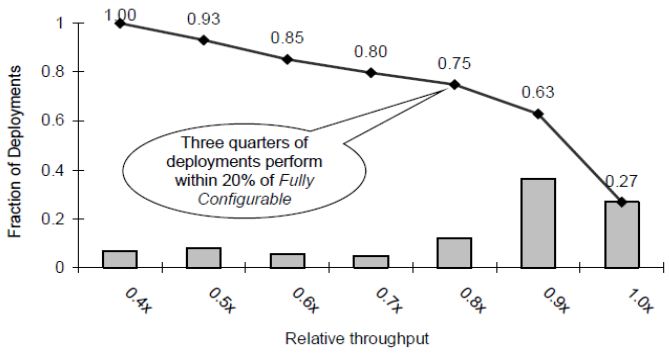

Abbildung 4. Performanz im Vergleich zum hypothetischen, voll konfigurierbaren Prozessor (Quelle: [1])

Prozessor nur um maximal 10% langsamer als der Hypothetische und in 80% der Anwendungen um maximal 30%. In 27% der Fälle erreicht er sogar den gleichen Durchsatz wie der hypothetische Prozessor.

VII. Fazit

Wie die Vergleiche mit den anderen Netzwerkprozessoren zeigen, stellt der hier vorgestellte Prozessor nicht nur eine Alternative zu ihnen dar. Vielmehr weist er in eine Richtung, in die sich zukünftige Netzwerkprozessoren orientieren sollten, um den stets steigenden Anforderung des Internetverkehrs gerecht zu werden. Dadurch, dass sich ein solcher Prozessor selbstständig an die Anforderungen der auf ihm laufenden Programme anpassen kann, muss bei der Programmierung dieser Programme nicht mehr auf die Eigenheiten der Zielhardware geachtet werden. Dadurch wird eine einfachere Programmierbarkeit erreicht.

Literatur

[1] J. Mudigonda, H. M. Vin, S. W. Keckler, *"Reconciling Performance and Programmability in Networking Systems"*

[2] J. Mudigonda, *"Addressing the Memory Bottleneck in Packet Processing Systems"*, PhD Thesis, University of Texas, 2005

[3] W. Riggert, *"Netzwerktechnologien"*, Carl Hanser Verlag, 2003

[4] A. Tanenbaum, *"Structured Computer Organization"*, Fifth Edition, Pearson, 2006

[5] *Intel IXP2800 Network Processor*, ftp://ftp5.chinaitlab.com/whitebook/Edge_Core_Applications.pdf

[6] *SimpleScalar*, http://www.simplescalar.com/

[7] *NLANR Network Traces*, http://pma.nlanr.net/Traces/

Functionality	Application	Source	Notes
Integrity verification	`checksum`	Free BSD	Protects headers in TCP, UDP and IP packets (RFC-1071)
	`md5`	R.S.A Inc.	MessageDigest 5 (MD5). Mostly used to protect the payload (RFC-1321)
Classification	`classify`	UT-Austin	Hashes the five-tuple: ⟨`srcIP,dstIP,srcPort,dstPort,protocol`⟩
Route Lookup (Longest prefix match)	`patricia`	Free BSD	Patricia tree. Can handle non-contiguous masks. Used in many end-systems
	`bitmap`	UT-Austin	Employs bitmaps to compress trie nodes. Used in many commercial routers
	`bsol`	UT-Austin	Binary search using hash tables. Has the best known avg. comp. complexity
	`ixp`	IXA SDK 3.0	Designed for IXP series of NPs. Two tries are searched simultaneously
Metering (Prioritize packets)	`srtcm`	IXA SDK 3.0	Enforces a *single* mean rate and a peak burst. (RFC-2697)
	`trtcm`	IXA SDK 3.0	Enforces *two* independent rates: mean and peak. (RFC-2698)
	`tswtcm`	UT-Austin	Enforces mean and peak rates over sliding windows. (RFC-2859)
Header processing	`stream`	Snort 2.0	TCP receive-side processing. Reassembles byte streams out of packets
	`portscan`	Snort 2.0	Detects portscan attack if too many ports are accessed too quickly
Payload processing	`cast`	SSLeay Lib	Encryption scheme. Used in Virtual Private Networks (VPNs) (RFC-2612)
	`vscan`	Snort 2.0	Pattern matcher. Scans packet payload for virus signatures
Scheduler	`drr`	UT-Austin	Deficit Round Robin. Found in many commercial routers

Tabelle I
GETESTETE PROGRAMME (QUELLE: [1])

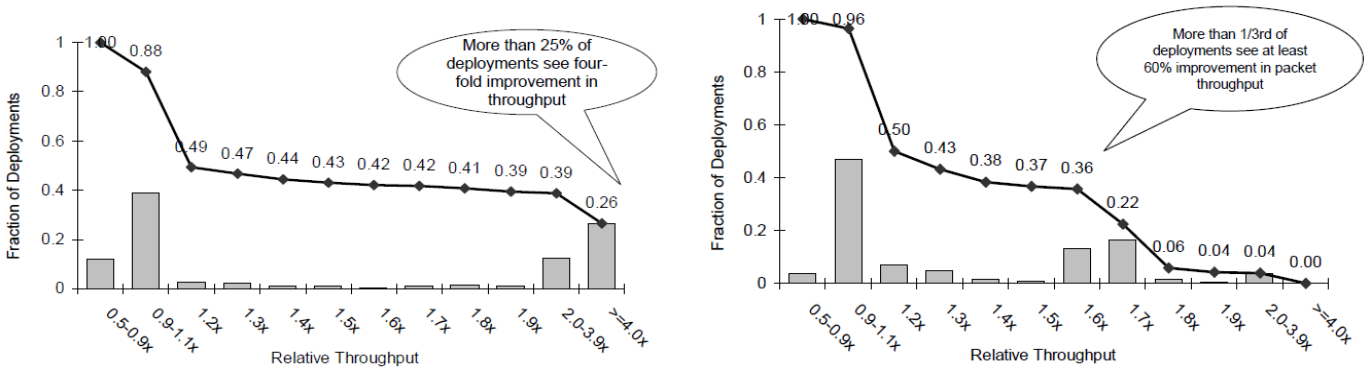

Abbildung 3. Performanz im Vergleich zum IXP2800 und einem optimalen, festen Prozessor (Quelle: [1])

Moderne Botnetze

Anton Hattendorf
Betreuer: Marc Fouquet
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
E-Mail: hattendo@in.tum.de

***Kurzfassung*— Botnetze sind ein Zusammenschluss von mehreren verteilen Rechner. Sie erfüllen Aufgaben, die ihnen übertragen werden. Dies sind unter anderen der Versand von SPAM und DDoS Angriffe. Um ihre Aufgaben zu koordinieren verfügen sie über eine Kommunikationsschnittstelle. Die ersten Botnetze haben IRC zur Kommunikation verwendet. Dies hatte den Nachteil, dass sie zentrale Infrastrukturen benötigen, welche leicht deaktiviert werden konnte. Deshalb werden die IRC-basierten Netze von anderen Strukturen abgelöst. Die folgende Ausarbeitung beschäftigt sich mit den Strukturen von modernen Fast-Flux und Peer-to-Peer Botnetzen. Weiterhin werden die Slapper und Storm Botnetze genauer betrachtet. Zu Abschluss wird erläutert, wie man gegen Botnetze vorgehen kann.**

***Schlüsselworte*— Botnetze, Fast-Flux, P2P, Slapper, Storm, Conficker**

I. Einleitung

A. Was sind Bots

Bots sind Teile eines verteilten Softwaresystemen. Die Software eines Bots wird auf mehreren physisch unabhängigen Rechnern ausgeführt. Die Bots kommunizieren über Netzwerke miteinander und koordinieren sich auf diese Art zu einem sogenannten Botnetz (botnet). Ein Bot verfügt über eine Steuerschnittstelle (siehe Abschnitt II-B) und wird in der Regel von einer übergeordneten Einheit, dem sogenannten Mutterschiff (mothership), gesteuert und überwacht. Die ersten Bots sind um 1993 im IRC entstanden [1]. Die Betreiber von Botnetzen werden auch „Bot Herder“ genannt.

Das Ziel eines Botnetzes ist nicht zwangsweise bösartig: gutartige Bots sind z.B. Webcrawler wie der Googlebot oder der NickServ im IRC. Diese gutartigen Bots verhalten sich kooperativ gegenüber den Web-Applikationen. Webcrawler durchsuchen das WWW und erstellen gemeinsam eine Datenbank, auf die die jeweiligen Suchmaschinen dann zugreifen. Dabei kooperieren sie mit den Webservern, indem sie z.B. den Robots Exclusion Standard [2] befolgen oder die genutzte Bandbreite der durchsuchten Seiten beschränken.

Bösartige Botnetze sind z.B. für SPAM-Mails, Phishing und DDoS Attacken verantwortlich. Sie werden für kriminelle oder zumindest fragwürdige Zwecke eingesetzt und verhalten sich in der Regel nicht kooperativ zu Dritten. Die Bots von bösartigen Botnetzen werden in der Regel auf anderen Rechnern ohne die Genehmigung des Betreibers installiert.

Die Systeme des Internets (Server, Firewalls, ...) reagieren auf die Handlungen von gutartige und bösartige Bots verschieden. So werden gutartige Bots häufig unterstützt, während bösartige Bots bekämpft werden. Dem Googlebot werden z.B. Inhalte zur Indizierung verfügbar gemacht, welche anderen Benutzern nicht unbedingt zugänglich sein sollten [3] (z.B. Bezahlinhalte). Im Gegensatz dazu werden SPAM-Bots dadurch, dass sie auf Blacklisten eingetragen werden, an der Erfüllung ihres Zwecks gehindert.

Beide Arten von Bots benutzen zwar teilweise die gleichen grundlegenden Technologien (verteilte Systeme, Steuerschnittstelle), werden aber von den Systemen des Internets verschieden behandelt. Deshalb wird sich diese Ausarbeitung im Folgenden nur mit bösartigen Bots beschäftigen und der Begriff Bot wird synonym für bösartiger Bot verwendet.

B. Abgrenzung zu Viren, Würmern, etc.

Die Gemeinsamkeit von Viren, Würmern und Bots ist, dass es sich um unerwünschte Software handelt. Eine Schadroutinen ist bei allen nicht notwendig, aber häufig anzutreffen. Häufig integrieren sie sich auch in das Betriebssystem, um sicherzustellen, dass sie auch nach einem Neustart des Systems wieder aktiv werden.

Viren: Sie infizieren fremde Programme, indem sie ihren Code in den Code der Programme integrieren. Bei compilierten Programmen oder Bibliotheken läuft dies auf der Ebene von Maschinencode ab. Sie können sich aber auch in interpretierten Skripten, Makros von Office-Programmen oder Boot-Sektoren von Datenträgern einnisten.

Ein Virus verbreitet sich durch die Weitergabe seines Codes in den infizierten Dateien. Sobald der Virus auf einem System ausgeführt wurde, infiziert er andere Programme.

Seit Beginn dieses Jahrtausends sind Viren durch die zunehmenden Vernetzung über das Internet in den Hintergrund getreten und durch Würmer abgelöst worden [4].

Würmer: Würmer verbreiten sich, indem sie aktiv andere Rechner über das Netzwerk infizieren. Dies kann auf verschiedene Arten ablaufen: E-Mails, Sicherheitslücken aller Art oder Dateifreigaben. Zusätzlich können sich Würmer noch wie Viren durch das Infizieren von Programmen verbreiten.

Bekannte Würmer sind z.B. der „ILOVEYOU“, welcher sich im Jahr 2000 mit Hilfe von E-Mails rasant ausbreite, oder der „Code Red“, der eine Sicherheitslücke im Microsoft IIS nutzte.

Die Verbreitung von Würmern ist nicht auf das Internet beschränkt. So verbreitete sich der SymbOS.Commwarrior.A über Bluetooth und MMS auf Handys und PDAs [5].

Im Gegensatz zu Bots verfügen Würmer über keine Steuerschnittstelle (siehe II-B) und können deshalb nicht überwacht oder ferngesteuert werden. Allerdings verwenden viele Bots Wurm-Techniken um sich zu verbreiten, weshalb der Unterschied zwischen Bot und Wurm fließend ist und nicht fest definiert ist. Viele Anti-Viren-Softwarehersteller verwenden auch für Bots den Begriff Wurm.

II. Bot Techniken

A. Infektion

Die meisten Bots verbreiten sich, indem sie andere Rechner über das Internet infizieren. Dabei gibt es verschiedene Möglichkeiten. Diese Infektionsmethoden werden auch von Würmern verwendet. Typische Methoden sind:

Software-Programmierfehler: Der Bot verwendet ein Exploit für eine Sicherheitslücke in einem fremden System um auf diesem gestartet zu werden. Dieser Exploit nutzt z.B. einen Buffer-Overflow aus. Dies betrifft sowohl Server- als auch Client-Systeme.

Auf Server-Systemen wird i.d.R. der Daemon, welcher einen Dienst zu Verfügung stellt, angegriffen. Dies sind meistens Webserver, da deren Dienste für das gesamte Internet angeboten werden. Neben der eigentlichen Server Software können auch Webapplikationen angegriffen werden.

Clients werden meistens durch Webseiten mit schadhaften Inhalten angegriffen. Diese Angriffe basieren häufig auf ActiveX Controls oder JavaScript. Allerdings ist es auch möglich, einen Client direkt anzugreifen, z.B. über eine Windows-Freigabe.

Software-Designfehler: Im Unterschied zu Programmierfehlern handelt es sich hier um ein Feature, welches bewusst in ein Programm eingebaut wurde und für bösartige Zwecke missbraucht werden kann. Dazu gehört z.B. automatisches Öffnen von Dateianhängen in alten Mail-Clients.

Schwache/keine Passwörter: Auf einem Server ist ein privilegierter Dienst nur mit einem schwachen oder gar ohne Passwort abgesichert ist. Ein Bot kann z.B. durch eine Wörterbuch-Attacke das Passwort in Erfahrung bringen. Über diesen Zugang kann sich der Bot auf dem Server installieren.

Social Engineering: Hierbei überzeugt der Bot den Benutzer davon, ihn auszuführen. Dies geschieht in der Regel, indem das Interesse des Benutzers geweckt wird. Typische Formen dieses Angriffes sind Mailattachments, die eine wichtige Nachricht enthalten sollen, oder Programme, die für den Benutzer interessant sind und heruntergeladen werden können. Diese Angriffe sind, wenn sie gezielt ausgeführt werden, auch mit technischen Maßnahmen nur schwer abzuwehren. Sie greifen den Benutzer auf sozialer Ebene an, indem sie Vertrauen zu ihm aufbauen und dieses missbrauchen [6].

Zu dieser Methode gehören sowohl das Verschicken des Bots mit SPAM-Mails als auch die Integration des Bots in Anwendungen wie Spielen oder Tools (Trojaner).

B. Steuerschnittstelle

Diese Steuerschnitstelle, auch C&C (Command & Control) genannt, unterscheidet Bots von Würmern und Viren. Der Betreiber eines Botnetzes steuert über sie die Aktionen des Botnetzes.

Typische Kommandos sind:

- Rechner infizieren (vergl. II-A)
- Update herunterladen (vergl. II-C)
- Schadroutinen ausführen (vergl.II-D)

C. Modularer Aufbau

Viele Bots sind modular aufgebaut. Das heißt, dass sie über einen Update-Mechanismus mit neuen Code versorgt werden können und dadurch ihre Funktionalität erweitern können. Die Art dieser Updates ist dabei sehr verschieden. Es kann sich um ausführbare Programme, Shared Libraries oder Skripte handeln.

D. Schadroutinen/Anwendungen

Die Struktur eines Botnetzes macht dieses für viele illegale Anwendungen interessant. Viele dieser Anwendungen können auch auf den Schwarzmarkt gebucht werden.

1) SPAM: Durch den Versand von SPAM kann das Botnetz zwei Ziele erreichen:

- eigene Verbreitung durch die Infektion anderer Rechner
- kommerzielle Ziele durch den Versand von Werbemails

Die Bots bekommen vom Netzwerk ein Template für eine SPAM Mail übermittelt und versenden auf dessen Basis SPAM Mails.

Die E-Mail-Adressen, an die der Bot den SPAM versenden soll, werden entweder vom Botnetz übertragen oder der Bot durchsucht den infizierten Rechner und/oder das Internet nach E-Mail Adressen. Außerden kann er E-Mail-Adressen zufällig generieren, indem z.B. Kombinationen aus häufigen Namen und verschiedenen Domains ausprobiert werden.

Dadurch, dass die Bots den SPAM versenden ist es schwieriger, diesen zu filtern. Die verschiedenen Absender-IP-Adressen machen das Blacklisting der versendenden Hosts sehr aufwendig.

Ein Bot verbreitet sich über den Versand von SPAM, indem er entweder seinen eigenen Code im Anhang der Mail plaziert, oder die Mail einen Link zu einem Webserver enthält, auf welchem der Bot liegt. Letztere wurde z.B. vom Storm verwendet [7]. Die Ziele des SPAM Versands sind neben der eigenen Verbreitung des Bots häufig auch kommerzieller Natur. Kommerzielle Ziele sind der Versand von Werbemails für Drogen oder Medikamente (häufig Viagra etc.). Weiterhin werden mit ihnen Opfer für Finanzbetrug gesucht (z.B. Nigeria Mail oder Phishing).

2) DDoS: Bei einem Distributed Denail-of-Service(DDoS) Angriff werden von mehreren Bots Pakete zu einem Ziel-Host gesendet, so dass die gesamte Bandbreite des Ziels blockiert ist. Dadurch steht keine Bandbreite mehr für andere Dienste zu Verfügung, und das Ziel ist nicht zu erreichen. Auch wenn die einzelnen Bots häufig nur über einen sehr schmalen

Uplink (128 kBit/s bei T-DSL 1000) verfügen, so können sie, wenn sie ihre Attacke zeitlich koordinieren, mit ihrer gesamten Bandbreite durchaus beachtlichen Schaden anrichten.

Die genaue Durchführung der Attacke kann sehr vielfältig sein. Das Spektrum geht von einfachen Flooding bis zu SYN-Flood-Attacken.

Ziel dieser Angriffe ist es, konkurierende Botnetze oder Betreiber anderer Internet Dienste durch das Blockieren ihrer System zu schädigen oder Lösegeld dafür zu erpressen, dass eine solche Attacke nicht durchgeführt wird. Letztes ist dem Wettanbieter mybet.com während der Fußball Europameisterschaft 2004 passiert. Die Erpresser forderten 15000 US-Dollar, und legten, weil mybet.com nicht zahlen wollte, deren Website für 16 Stunden still [8].

3) Phishing: Ein Bot kann auf verschiedene Arten an Phishing-Angriffen beteiligt sein: Einerseits können die Bots zum Versenden der Mails verwendet werden, welche die Benutzer auf Phishing-Seiten locken. Auf den anderen Seite kann das Botnetz auch die eigentliche Phishing Seite zu Verfügung stellen und die Daten sammeln. Dafür muss der Bot aber von Internet aus erreichbar sein. Dies geschieht meistens auf der Basis von Fast-Flux Netzwerken (vergl. III-C).

Mit solchen Angriffen bekommt der Bot Herder Zugriffsdaten für Konten und kann diese „abräumen“.

4) Sammeln vertraulicher Daten: Der Bot durchsucht den infizierten Rechner nach vertraulichen Daten und übermittelt diese an das Botnetz. Bei diesen kann es sich um Adresslisten, Schlüsseln für Programme, gespeicherte Passwörter, Seriennummern etc. handeln.

5) Proxy: Der Bot fungiert als Proxy zu beliebigen anderen Zielrechnern und ermöglicht damit dritten den Zugriff auf die Zielrechner ohne das diese mit ihrer IP für den Zielrechner erkennbar sind.

6) Datenspeicher: Daten können in dem Netz gespeichert werden, und von verschiedenen Benutzer heruntergeladen werden. Dabei kann es sich sowohl um von den Bots gesammelte Daten handeln als auch um Daten die der Betreiber seinen Kunden zu Verfügung stellt. Typische Daten sind Raubkopien von Programmen und pornografisches Bildmaterial.

E. Eigenschutz

Moderne Würmer und Bots vertuschen ihre eigene Existenz. Die ersten Schädlinge haben dies versucht, indem sie ihre Dateien als versteckt markieren. Dies reicht inzwischen aber nicht aus, um Virenscanner zu überlisten.

Effektiver ist die Verwendung eines Root-Kits. Da wird so in das System so eingegriffen, dass z.B. bestimmte Dateien nicht im Verzeichnis aufgelistet werden oder bestimmte Prozesse nicht angezeigt werden. Dadurch das ein Root-Kit auch seine eigene Existenz verschleiern kann, sind solche Schädlinge oft nur durch eine Analyse des Betriebssystem durch ein spezielles von CD gebootetes System zu finden. Ein Root-Kit kommt z.B. bei einigen Versionen des Storm Bots oder dem Srizbi Bot zum Einsatz [7], [9].

Andere moderne Eigenschutz Varianten sind die Manipulation von Virenscannern, so das diese zwar scheinbar funktionieren, aber den Schädling übersehen oder die Verwendung von Hardware-Virtualisierungsfunktionen.

Hardware-Virtualisierung erlabt es, einen PC in mehrere virtuelle, voneinander unabhängige Maschinen zu unterteilen. Dadurch können Betriebssystem und Virenscanner in einer anderen Umgebungen als der Bot laufen und können die Existenz des Bots nicht feststellen [10]. Es ist allerdings noch kein Bot bekannt, welcher Hardware-Virtualisierung nutzt.

III. Botnetze

A. Einführung

Da die meisten Botnetze in der illegalen Aktivitäten nachgehen, sind zentrale Strukturen meist nicht besonders effektiv. Sobald der Server abgeschaltet wird werden die Bots nutzlos. Die Struktur des Netzwerk ist auch essentiell für die Effektivität der Steuerschnittstelle.

B. IRC

Die ersten Botnetze waren IRC [11] basiert. Bei diesen Botnetzen agierte ein IRC Server als zentrale Instanz und alle Bots haben sich sich mit einem Kanal des IRC Servers verbunden.

Zum Steuern des Netzes verbindet sich auch der Betreiber mit dem IRC Server und sendet seine Kommandos an den IRC Kanal. Diese werden von den Bots empfangen und ausgeführt [1], [12].

Der Vorteil dieser Botnetze ist, dass keine spezielle Server Software notwendig ist und entsprechenden Client Bibliotheken vorhanden sind. Weiterhin können neben den, durch den Botnetz Betreiber aufgesetzten, IRC Server auch Kanäle in öffentlichen IRC Netzen für die Kommunikation der Bots verwendet werden [13].

Nachteilig für IRC basierte Bots ist, dass falls der IRC Server bzw. der IRC Kanal abgeschaltet wird, es nicht mehr möglich ist, das Botnetz zu kontrollieren. Deshalb sind die IRC Server, die für solche Zwecke eingesetzt werden, meistens in Fernost platziert.

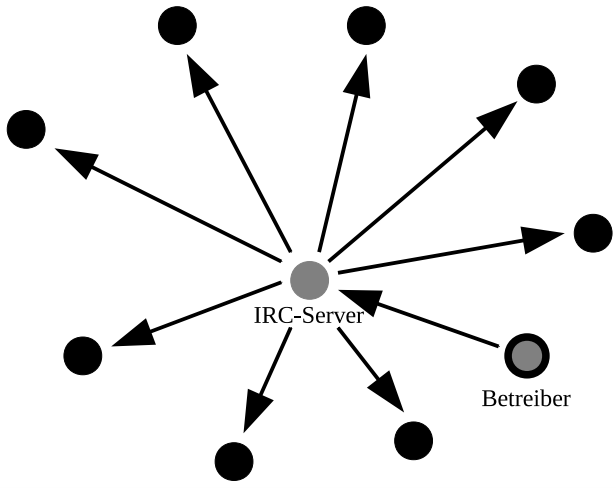

Abbildung 1. Struktur eines IRC basierten Botnetzes

C. Fast-Flux

1) Grundlagen: Fast-Flux Netzwerke nutzen das DNS [14], [15] um ihre Dienste zu Verfügung zu stellen.

Mit Hilfe eines Fast-Flux Netzwerkes soll erreicht werden, dass ein DNS Eintrag (z.B. www.fastflux.com) auf auf viele IP Adressen verweist (A Records). Die IP Adressen werden in einem Round Robin Verfahren mit hoher Frequenz getauscht und mit einer kurzen Lebensdauer (TTL) versehen. Bei dem Auswechseln der A Records berücksichtigt das Fast-Flux Netzwerk Verfügbarkeit und die Bandbreite der Hosts [16].

Ein Benutzer, der einen Host beim DNS abfragt, bekommt bei jeder Anfrage andere Ergebnisse. Diese Eigenschaft des DNS wird von großen Websites zum Lastausgleich benutzt. In Fast-Flux Botnetzen wird dieses nun gesteigert zu einem Lastausgleich zwischen Tausenden von Bots.

Die DNS Einträge verweisen auf Bots des Botnetzes. Diese Bots agieren häufig nur als Proxy für das Mutterschiff, welches dadurch im Hintergrund versteckt bleiben kann und von außen nicht gefunden werden kann. Das Mutterschiff ist das steuernde Element in dem Fast-Flux Netzwerk und übernimmt quasi die Funktion des C&C von IRC basierten Botnetzen.

Die Bots der ersten Reihe sind dabei entsorgbar, d.h. ein Verlust einzelner Bots beeinflusst das Botnetz nicht. Bei ihnen handelt es sich meisten um gehackte Maschinen, welche keinerlei Bezug zu den Betreibern des Botnetzes haben.

Ein Bot, der auf einen Rechner aktiviert wird, fragt beim DNS eine Domain ab. Er bekommt vom DNS IP Adressen von verschiedenen Proxys und baut eine Verbindung zu einem auf. Über diese Verbindung registriert er sich bei dem Netzwerk. Der Proxy leitet diese Information weiter an das Mutterschiff und schickt Kommandos zurück. Die Kommunikation mit dem Mutterschiff läuft oft über HTTP [17] da dieses von vielen Firewalls und Proxys anstandslos weitergeleitet wird.

Fast-Flux Netze könne durch deaktivieren der Domain stillgelegt werden. Dies geschah im November 2008 mit dem Srizbi Botnetz [18]. Die Zentral verwalteten Domain-Namen sind eine kritische Ressource des Botnetzes. Die meisten Bots verfügen deshalb über Ersatz-Domains, die sie in einem solchen Fall verwenden können. Weitere Details zu diesem Vorgehen sind in IV-B am Beispiel von Srizbi beschrieben.

Man unterscheidet zwischen Single-Flux und Double-Flux Netzwerken. Der Unterschied zwischen beiden ist, dass bei Double-Flux auch der DNS-Server, rotiert wird.

2) Single-Flux: Bei Single-Flux Netzwerken werden in den Name-Server für die Fast-Flux Domain regelmäßig neue Bots als A Record eingetragen. Das Eintragen erfolgt meistens auf Basis der gerade verfügbaren Bandbreiten der aktiven Bots. Die genauen Kriterien hängen von der Implementierung des Botnetzes ab und werden im Hintergrund von den Steuerroutinen getroffen.

Für die Domain wird ein sogenannter „Bullet Proof Domain Name“ benötigt. Dabei handelt es sich eine Domain, die bei einen Registrar registriert ist welcher diese nicht aufgrund von Beschwerden einfach abschaltet. Solche Domains sind bereits für 100 US Dollar pro Jahr zu bekommen und können anonym registriert werden [19].

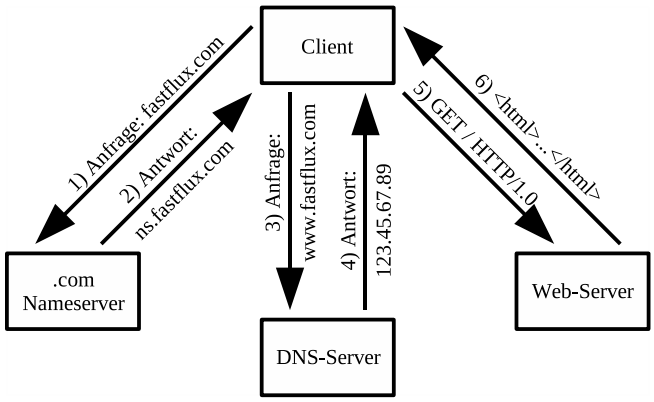

Abbildung 2. Normale Abfrage einer Website

Die Name-Server für Single-Flux Domains müssen besonders geschützt werden, da sie eine kritische Ressource sind. Deshalb befinden sie sich meistens in den Ländern wo die Strafverfolgung für „digitale Kriminalität“ nicht so ausgereift ist [19]. Dadurch wird das Abschalten solcher Domains deutlich erschwert.

Der Typische Kommunikationsverlauf für die Abfrage eine Single-Flux Domain ist in Abbildung 3 zu sehen.

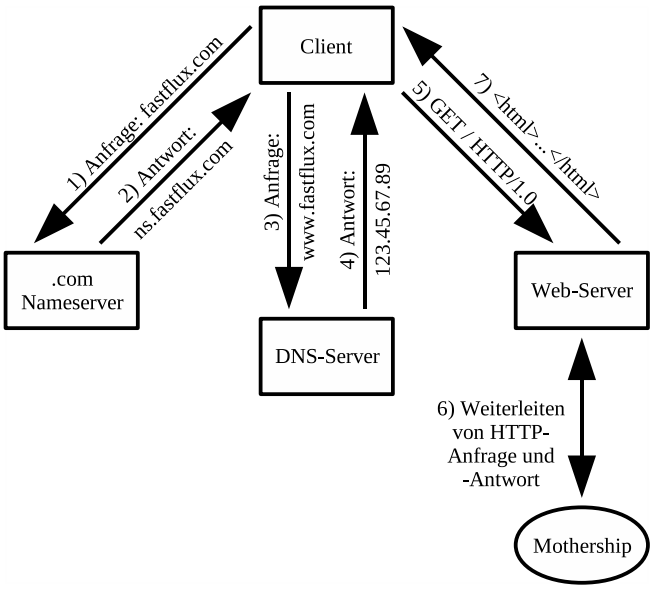

Abbildung 3. Abfrage einer Single-Flux Website

3) Double-Flux: Double-Flux Netzwerke wirken dem Problem entgegen, dass Name-Server für Single-Flux Domains eine kritische Ressource sind. Dies wird erreicht, indem auch die Nameserver rotieren. Diese Nameservern beziehen ihre Informationen dann auch vom im Hintergrund agierenden Mutterschiff.

Die Frequenz, mit der die Nameserver rotieren, ist deutlich geringer als Frequenz der Hosts. Die Frequenz liegt im Stundentakt. Es wird dennoch ein kooperativer Registrar gebraucht,

welcher eine automatisierte Schnittstelle für die Aktualisierung der NS Einträge bereitstellt und sich nicht daran stört, dass dieses im Stundentakt passiert [16], [19].

In Abbildung 4 ist ein Beispiel für die Kommunikation mit einem Double-Flux Netzwerk zu sehen.

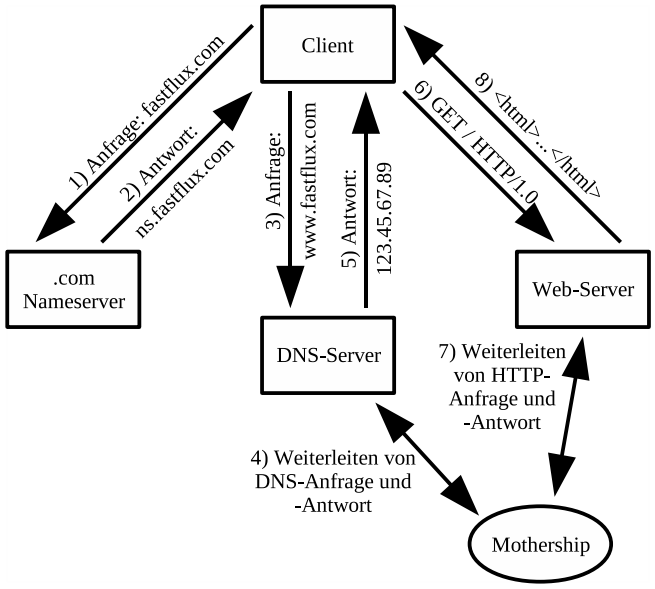

Abbildung 4. Abfrage einer Double-Flux Website

D. P2P

1) Einführung: Peer-to-Peer Netze sind Netze, welche von zentralen Strukturen unabhängig sind. In ihnen gibt es keine klassische Unterscheidung zwischen Client und Server und in der Regel können die Ressourcen von allen teilnehmenden Knoten gleichberechtigt genutzt werden. Die Teilnehmer in einen P2P Netz organisieren sich selbst, d.h. sie bauen selbst ihre Struktur auf und kümmern sich auch um deren Aufrechterhaltung. Nachrichten werden innerhalb des Netzes von Knoten zu Knoten weitergegeben. Typischerweise haben alle Knoten eine gleichartige Implementierung [20].

Für Botnetze ist die Unabhängigkeit von zentralen Strukturen von besonderen Vorteil. Netze mit zentralen Strukturen sind durch das Abschalten der zentralen Instanzen in viele kleine Inseln zufallen und die Bots können nicht mehr genutzt werden. Dadurch sind Botnetze mit zentralen Strukturen Providern und Regierungen „ausgeliefert“ und bieten auch Ansatzpunkte für die Strafverfolgung.

Ein P2P Netz ist in dieser Hinsicht deutlich robuster: es verkraftet auch den Ausfall einer größeren Menge an Knoten. Da alle Knoten gleichberechtigt sind ist auch nicht ersichtlich, hinter welchen sich das C&C verbirgt. Der Betreiber des Botnetzes bindet sich wie ein normaler Knoten in das Botnetz ein und sendet dann seine Kommandos ab. Aus Sicht des Botnetzes sieht der Betreiber wie ein normaler Knoten aus.

2) Bootstrapping: Der Vorteil, dass P2P Botnetze ohne zentrale Instanzen auskommen, bringt aber auch einen Nachteil mit: das Bootstrapping. Ein neuer infizierter Rechner, der sich

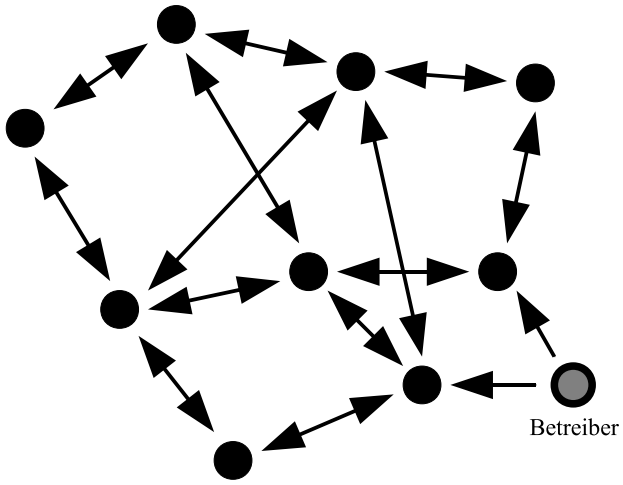

Abbildung 5. Struktur eines P2P Netzes

in das Botnetz einbinden möchte, muss erstmal Kontakt zu mindestens einen Knoten des Botnetzes aufnehmen. Von diesem kann der Knoten dann Kontaktadressen weiterer Knoten erhalten.

mitgelieferte Nachbarn: Bei der Infektion eines Rechners kann dem Schädling eine Liste mit Knoten mitgegeben werden, welche für die erste Kontaktaufnahme verwendet werden soll. Dieses Verfahren wird allerdings problematisch, wenn ein Knoten längere Zeit offline ist und alle bekannten Knoten zwischenzeitlich neue IP-Adressen bekommen haben. Dem kann durch eine hohe Anzahl von benachbarten Knoten entgegengewirkt werden. Das Slapper Botnetz verwendete ein solches Verfahren [21].

zentrale Bootstrapping Instanz: Für das initiale Bootstrapping eines Botnetzes kann auch eine zentrale Instanz verwendet werden. Diese ist dann der Rendezvous Punkt für das Botnetz.

nutzen anderer P2P Netze: Phatbot verwendet das Gnutella Netzwerk zum Bootstrapping: er verbindet sich mit dem Gnutella Netzwerk, verwendet aber einen andere Port. Dadurch ist er von anderen Gnutella Clients unterscheidbar und kann andere Knoten finden um sein eigenes Netzwerk zu erreichen [22].

IV. Beispiele für Botnetze

A. Agobot

Der Agobot wurde im Jahr 2003 von dem Deutschen Axel „Ago“ Gembe entwickelt und in Umlauf gebracht [23]. Er ist in C++ geschrieben und sehr Modular aufgebaut. Der Quelltext des Wurms wurde im April 2004 veröffentlicht [24].

Agobot ist auch unter den Namen Gaobot bekannt. Weiterhin gibt es viele Varianten wie den Phatbot, Forbot oder dem XtrmBot. Insgesamt gibt es mehr als tausend Varianten des Bots. Dies ist vor allem bedingt durch die Veröffentlichung des Quelltextes und dem modularen Aufbau [1].

Agobot infiziert ein System, indem er sich in die System Verzeichnisse kopiert und in der Registry Einträge anlegt,

die sicherstellen, dass er bei Systemstart ausgeführt wird. Er infiziert andere Systeme indem er versucht, sich auf ihre Windows-Freigaben zu kopieren oder Exploits ausnutzt [25].

Die meisten Varianten des Agobot haben eine IRC basierte Kommunikation. Er besitzt allerdings auch Funktionen für eine P2P Kommunikation. Diese war aber nicht besonders effektiv und wurde so gut wie nicht verwendet [1]. Die Agobot Variante Phatbot verwendet auch eine P2P Kommunikation, welche allerdings nicht auf der P2P Kommunikation des Agobot basiert, sondern eine WASTE Implementierung verwendet. Diese skaliert aber nicht auf große Netzwerke [22].

Nach der Infektion verbindet sich der Agobot mit mehreren einprogrammierten IRC Servern. Weiterhin ist er in der Lage, den Betreiber des Botnetzes zu identifizieren.

Die von Agobot unterstützten Kommandos sind sehr vielfältig: Es gibt Befehle um die Steuerschnittstelle der Bots zu beeinflussen. So kann der Bot angewiesen werden, einen anderen IRC Server zu verwenden oder den Kanal zu wechseln. Weiterhin gibt es Befehle um Dateien herunterzuladen und auszuführen, DoS Angriffe verschiedenster Art durchzuführen, verschiedene Proxys einzurichten und Prozesse auf dem Rechner zu beenden [13]. Der Befehlsumfang kann durch weitere Module erweitert werden.

B. Srizbi

Srizbi ist ein Schädling, der bei seinem Ausbruch im Juni 2007 als einer der innovativsten galt: Er arbeitet komplett im Kernel Mode. Weiterhin versteckt er sich mit Hilfe eines Root-Kits und versendet Spam. Das Srizbi Botnetz war im Frühjahr 2008 für 60 Milliarde SPAM Mails am Tag verantwortlich [26].

Nach der Infektion eines System verursacht Srizbi keinerlei Aktivität im User Mode. Er Installiert sich als Treiber und manipuliert die Netzwerk Treiber so dass er direkt über sie Pakete versenden kann. Dadurch kann er sogar lokale Firewalls und Sniffer umgehen und seine eigenen Pakete vor dem System verstecken [9].

Das Srizbi Netzwerk konnte im November 2008 durch die Abschaltung des Providers McColo vorübergehend stillgelegt werden. Dies hatte zu Folge, dass das SPAM aufkommen um bis zu 90 % reduziert werden konnte [18], [27].

Dadurch, dass die Bots den Kontakt zum Mutterschiff verloren hatten, war es ihnen nicht mehr möglich, neue Kommandos abzurufen. Allerdings verfügt der Srizbi Bot über eine Notfallkommunikation. Er berechnet alle 72 Stunden vier alterative Domains, welche er für einen Verbindungsaufbau ausprobiert.

Der Sicherheitsdienstleister FireEye hat diesen Algorithmus entschlüsselt und für eine gewisse Zeit diese Domains registriert. Dadurch konnte die erneute Kontaktaufnahme der Bots mit dem Mutterschiff verhindert werden. Die Registrierung der Domains verursachte allerdings Kosten in der Höhe von 4000 US Dollar pro Woche weshalb das Vorhaben aufgegeben werden musste. Kurz darauf wurde eine der Domains wieder von dem Botnetz Betreiber registriert und das SPAM Aufkommen stieg wieder an [28], [29].

FireEye hätte die Bots sogar anweisen könne, sich selbst zu deaktivieren. Dies war ihnen aus rechtliche Gründen allerdings nicht möglich. Weitere Erläuterungen zu den rechtlichen Rahmenbedingung sind in Abschnitt V-C erläutert.

C. Slapper

1) Übersicht: Der Slapper Bot wurde am 13. September 2002 in Rumänien das erste mal gesichtet. Er basiert auf dem Apache Scalper Bot, welcher ebenfalls 2002 in Umlauf kam. Slapper hat aber verbesserte Netzwerkeigenschaften und einen anderen Angriffscode.

Slapper dringt in den Apache Webserver auf IA32 Linux Systeme über einen OpenSSL Exploit ein. Es waren mindesten sechs verschiedene Linux Distributionen und neun Unterversionen des Apache Webserver betroffen.

Als Schadroutine verfügte er über die Möglichkeit, an DDoS Attacken teilzunehmen [30].

2) Architektur: Der folgende Absatz beschreibt die Architektur des Slapper Botnetzes und basiert auf einen Artikel von Iván Arce and Elias Levy [21].

Das Slapper Botnetz ist ein unstrukturiertes P2P Netz. Seine Hauptaufgabe ist, den Ursprung einer Nachricht durch mehrfaches Weiterleiten zu verschleiern. Das Protokoll basiert auf UDP und verwendet den Port 2002. Es verfügt noch über eine zusätzliche Sicherungsschicht die den Verlust von Nachrichten abfängt. Die Knoten werden in dem Netzwerk über ihre IP Adressen identifiziert.

Jede Nachricht im Slapper Netz wird durch eine zufällig gewählte Sequenznummer identifiziert. Ein Knoten verwaltet eine Liste mit den letzten 128 Nachrichten, die er empfangen hat. Wenn er eine Nachricht empfängt, deren Sequenznummer bereits in der Liste ist, wird diese verworfen. Weiterhin haben die Nachrichten eine ID, welche die Zuordnung einer Antwort zu der ursprünglich Nachricht ermöglicht.

Eine Nachricht, die durch das Netzwerk an einen bestimmten Knoten gesendet werden soll, enthält neben der IP Adresse des Ziel-Knotens auch einen Zähler, welcher die Anzahl der noch durchzuführenden Hops angibt. Wenn dieser Zähler Null ist, wird die Nachricht an die entsprechende IP Adresse weitergeleitet. Andernfalls wird die Nachricht mit um eins erniedrigten Zäher an zwei andere Knoten weitergeleitet. Die Nachricht wird also bei jeder Weiterleitung dupliziert. Der Maximalwert für den Zähler ist 16, üblicherweise wird aber fünf verwendet. Doppelt bei Empfänger ankommende Nachrichten werden verworfen.

Es ist auch möglich, ein Broadcast an das gesamte Netz zu senden. Diese Nachricht enthält keine Empfänger. Ein Broadcast wird allerdings nicht direkt an das gesamte Netz gesendet, sondern immer nur an jeweils zwei weitere Knoten. Wenn ein Knoten die Broadcast Nachricht bereits erhalten hat, wird sie nicht nochmal verteilt. Es ist nicht garantiert, dass alle Knoten eine Broadcast Nachricht erhalten.

Das Netz sogt dafür, dass sich möglichst alle Knoten kennen. Dies geschieht indem regelmäßig Nachrichten in das Netz gesendet werden und Knotenlisten ausgetauscht werden. Es gibt kein Entfernen von Knoten aus diesen Listen.

D. Storm

1) Übersicht: Das Storm Botnetz existiert seit ca. Ende 2006. Es sind seitdem viele Varianten des Bots erscheinen. Er verbreitete sich Anfangs über E-Mail Anhänge von Spam-Mails. Später beinhalteten die Mails nur noch Links, und der Benutzer wurde auf Webseiten gelockt und wo er den Bot herunterladen musste oder das Opfer von Browser Exploits wurde.

Viele Varianten des Bots verwenden Root-Kits um ihre Existenz zu verschleiern. Es gibt auch Varianten, die virtuelle Maschinen erkennen und dann entweder inaktiv bleiben oder diese zum Absturz bringen. Dies soll die Analyse des Bots erschweren.

Das Storm Botnetz kann sowohl zu Spam Versand als auch für DDoS Angriffe verwendet werden [7].

2) Architektur: Im folgenden wird die vom Storm Botnetz verwendete mehrstufige hybride Architektur beschrieben. Die Beschreibung basiert auf den Ergebnissen der Diplomarbeit von Frédéric Dahl [7].

An oberster Stelle steht Overnet, ein P2P Netz. Overnet wurde nicht nur von Storm sondern auch vom eDonkey2000 Client und seinen Nachfolgern verwendet. Overnet implementiert Kademila, eine verteilte Hashtabelle.

In Kademila ist jedem Knoten und jedem Datensatz ein zufällig generierter 160 Bit Bezeichner zugeordnet. Die Länge des gemeinsamen Präfix zweier Bezeichner gibt den Grad ihrer Ähnlichkeit an. Daten werden im Netz bei den Knoten gespeichert, denen sie am ähnlichsten sind.

Ein Knoten verwaltet für jede Ähnlichkeitsklasse eine Listen mit Knoten aus dieser Klasse. Wenn ein Knoten Kontakt zu anderen Knoten hat, wird dieser in die entsprechende Liste aufgenommen, sofern die Liste nicht voll ist. Jeder Knoten kennt aus jeder Ähnlichkeitsklasse gleichviele Knoten. Unter der Annahme, das die zufällig generierten Bezeichner sich gleichmäßig über dem Namensraum verteilen, kennt jeder Knoten viele der Konten, die ihm sehr ähnlich sind, und wenige derjenigen, die eine geringere Ähnlichkeit haben.

Bei der Suche nach einen Datensatz fragt der suchende Knoten zuerst die Knoten aus der Liste, die der Ähnlichkeitsklasse des suchenden Knoten mit den Bezeichner des gesuchten Datensatz entsprechen. Diese Knoten liefern eine Liste mit den ihnen bekannten Knoten, die dem gesuchten Bezeichner am ähnlichsten sind. Der suchende Knoten nimmt eine Teilmenge der ähnlichsten Knoten und fragt diese wiederum an. Das wird solange wiederholt, bis die Anfragen keine neuen ähnlicheren Knoten mehr liefern. Die ähnlichsten Knoten werden dann nach dem gesuchten Datensatz gefragt. Das Speichern von Daten läuft äquivalent.

Der Storm Bot berechnet jeden Tag aus dem aktuellen Datum 32 Bezeichner. Eine Suche in Netzwerk nach einem dieser Bezeichner bring eine Listen von Gateways. Gateways sind Knoten, die ohne Einschränkungen von anderen Knoten erreicht werden können und der mittleren Ebene des Storm Netzes angehören. Der Knoten baut eine Verbindung mit dem Gateways auf und fragt diesen nach Zielen von DDoS Attacke und Templates und Adressen für Spam-Mails.

Während des Verbindungsaufbaus mit dem Overnet erfährt ein Knoten, ob er ohne Einschränkungen erreichbar und damit ein Gateway ist. Ein Gateway veröffentlicht sich so in Overnet, dass er als ein solches erkennbar ist. Kurz danach wird er von der den Kontrollknoten aus der dritten Ebene des Storm Netzes kontaktiert.

Die Gateways bauen ein Fast-Flux Netzwerk (vergl. III-C) auf. Dieses wird für die Kommunikation der Gateways mit den Kontrollknoten, als auch um die in den Spam-Mails verlinkten Webseiten bereit zu stellen, verwendet.

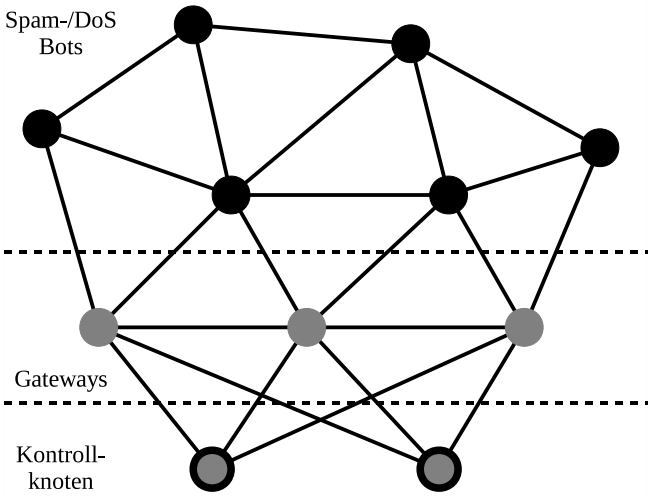

Abbildung 6. Struktur des Storm Botnetz

Seit Mitte Oktober 2007 verwendet das Botnetz eine einfache Verschlüsselung. Diese besteht aus einem 40 Bit Schlüssel, welcher per XOR mit dem Datenstrom verknüpft wird. Dadurch wurde das Storm Botnetz auch von dem restlichen, für Fileshareing benutzen, Overnet abgetrennt.

3) Sicherheitsanalyse: Das Storm Botnetz ist nicht besonders sicher aufgebaut. So wird nur eine sehr einfache Verschlüsselung mit eine kurzen sich wiederholenden Schlüssel eingesetzt. Weiterhin wird der Ursprung von Befehlen nicht überprüft.

Im Rahmen des 25C3 wurde demonstriert, wie man in das Storm Botnetz falsche C&C Server einschleust und Bots übernimmt oder deaktiviert [31], [32].

E. Conficker

1) Übersicht: Conficker, auch bekannt unter dem Namen Downadup verbreitet sich seit Mitte November 2008 über einen Windows RPC Exploit. Microsoft stellt zwar bereits seit dem 23. Oktober 2008 einen Patch für diesen zu Verfügung, dennoch konnte Conficker eine sehr große Verbreitung erlangen. Zusätzlich verbreitet er sich noch über lokale Dateifreigaben und entfernbare Medien. Der Ursprung von Conficker wird in der Ukraine vermutet [33].

Besonderes Medieninteresse erlangte Conficker am 13. Februar 2009, als bekannt wurde, dass er das Netzwerk der Deutschen Bundeswehr infiziert hatte. Zuvor waren bereits die

Netzwerke des britischen und französischen Militärs betroffen [34].

2) Architektur: Conficker berechnet täglich 250 Rendezvous-Domains und versucht von diesen ein Update herunterzuladen. Die Updates müssen mit einer Signatur versehen sein. Der öffentlichen Schlüssel für diese Signatur ist in den Code von Conficker eingebettet. Die Updates sind ausführbarer Code, welcher, sobald sein Ursprung verifiziert ist, gestartet wird.

Aktuell sind drei Varianten von Conficker in Verbreitung, welche sich aber teilweise ähneln. Alle verwenden ähnliche Algorithmen, um Domains zu generieren, aber teilweise mit verschiedene Initialisierungsparametern. Die neueren Versionen versuchen zusätzlich noch Sicherheitsmechanismen wie Virenscanner zu deaktivieren und können auch remote über eine Named Pipe der Windows IPC mit signierten Updates versorgt werden [33].

Conficker beinhaltet außer der Verbreitungsroutinen keinerlei Schadroutienen. Solche können allerdings jederzeit über ein Update nachgerüstet werden.

Aktuell ist nicht bekannt, dass der Update-Mechanismus von Conficker schon mal verwendet wurde. Außerdem verfügt er über keine C&C-Schnittstelle. Deshalb müsste man ihn zur Zeit eher als Wurm als als Bot bezeichnen. Die C&C-Schnittstelle könnte aber jederzeit über ein Update nachgerüstet werden [35].

V. Gegenmassnahmen

A. Systeme absichern

Die zuverlässigste Methode, dass Eindringen von Bots zu verhindern ist das System entsprechend abzusichern. Dazu gehören der Einsatz einer Firewall und eines Virenscanners. Insbesonders auf den Microsoft Betriebssystemen ist dieses notwendig. Durch ihre hohe Verbreitung, insbesonderes in Bereichen, in denen die Benutzer keine grundlegenden Kenntnissen über diese Problematik haben (z.B. der durchschnittliche Home-PC Anwender), bieten sie eine große Angriffsfläche für Bots.

Die Firewall verhindert, dass vom Internet aus auf das System zugegriffen werden kann. Auf den meisten Home- und Büro-Rechnern ist es nicht notwendig, dass der Rechner über das Internet erreicht werden kann. Dadurch können Lücken in den Systemdiensten nicht zum Einbruch genutzt werden. Bekannte Sicherheitslücken sollten möglichst schnell durch die Installation der entsprechenden Updates geschlossen werden.

Wichtig beim Einsatz eines Virenscanners ist, dass dieser regelmäßig aktualisiert wird. Ein Virenscanner arbeitet nur reaktiv. Er kann einen neuen Bot erst erkennen, wenn dieser dem Hersteller des Virenscanner aufgefallen ist, und das Signaturupdate dem Virenscanner verfügbar gemacht wurde. Agobot und Storm haben allerdings gezeigt, dass Virenscanner an ihre Grenzen kommen, wenn ein Schädling schnell in immer leicht verscheidenden Varianten auftaucht.

B. Abwehr von DDoS Angriffen

Präventiv kann man gegen DDoS Angriffe vor allem die eigene Infrastruktur auf des Eintreffen eines solchen Angriffs vorbereiten. Dazu gehören z.B.

- Verfügbar machen von TCP-SYN-Cookies [36]
- Einsatz von Paketfiltern
- Bereitstellen von Proxies und Reserve Servern
- Einsatz von Load Balancing

Während eines Angriffes können diese Maßnahmen dann gezielt aktiviert werden. Zusätzlich lohnt es sich, mit den Providern Kontakt aufzunehmen, damit die Pakete möglichst früh gefiltert werden [37].

C. Netz übernehmen und abschalten

Da viele Botnetze keine oder nur eine oberflächliche Authentifizierung verwenden, währe es theoretische möglich, die Bots mit einen gefälschten Befehl abzuschalten. Zusätzlich könnte z.B. auch noch Code nachgeladen werden, welcher die Bot-Software entfernt und die Sicherheitslöcher über die die Schädlinge eingedrungen sind schließt. Das konkrete Vorgehen ist von der Art des Botnetzes abhängig:

IRC: Solange keine asymmetrischen Verschlüsselungsverfahren zum Einsatz kommen enthält der Code des Bots alle Informationen um eine Verbindung zu dem IRC Kanal aufzubauen und Anweisungen an die Bots zu senden.

Domain gestützt: In diesem Fall muss die Domain übernommen werden. Dies kann z.B. durch einen Eingriff der Providers/Registras auf Anweisung einer höheren Instanz geschehen. Häufig haben die Bots auch einen Mechanismus, über welchen sie Domain-Namen berechnen und so versuchen Kontakt zu dem Netz herzustellen (vergl. Srizbi IV-B bzw. ConfickerIV-E). Da reicht es aus, einige solcher Domains zu reservieren und entsprechend auszustatten.

P2P: Hier muss mit einem manipulierten Client ein entsprechendes Kommando an das Netzwerk abgesetzt werden.

Konkret währen solche Eingriffe z.B. beim Storm [31] oder Srizbi [28], [29] möglich gewesen.

Ein solcher Eingriff ist allerdings rechtlich bedenklich: Das Entfernen des Bots könnte als eine unerlaubte Datenveränderung gemäß § 303a StGB [38] sein und währe dann unter Androhung einer Freiheitsstrafe von bis zu zwei Jahren verboten. Weiterhin könnte das Entfernen/Deaktivieren des Bots (aufgrund der hohen Verknüpfung mit den Betriebssystems) Schäden auf dem Rechner verursachen, wodurch Schadenersatzforderungen möglich wären.

VI. Zusammenfassung und Ausblick

Moderne Botnetze haben nicht mehr viel mit den alten IRC-basierten Netzen gemeinsam. Im allgemeinen zeichnet sich ein Trend zu Peer-to-Peer-basierten Botnetzen ab. Dies liegt darin begründet, dass diese Strukturen, sofern eine gute Implementierung verwendet wird, sehr robust sind. Weiterhin gibt es inzwischen auch einige freie P2P-Implementierungen, die für einen solchen Einsatz verwendet werden können.

Aber auch zentrale Instanzen kommen noch zum Einsatz. Wenn diese mit einen guten Fallback-Mechanismus ausgestattet sind, können sie sich sogar vom Abschalten ihres zentralen Servers erholen. Dies war im letzten Herbst besonders gut am Beispiel von Srizbi zu sehen.

Botnetze sind in Vergleich zu Viren eine neuere Erscheinung. Gegen sie können größtenteils ähnliche Mittel eingesetzt werden wie gegen Viren. Ein Problem ist allerdings, dass es oft viele Varianten des Bots gibt, wodurch die Bekämpfung erschwert wird. Gerade Bots mit öffentlichem Quelltext wie der Agobot haben dieses Problem verdeutlicht. Dadurch, dass Virenscanner nur reaktiv arbeiten, ist es ihnen nicht möglich, Bots mit absoluter Sicherheit aufzuspüren.

Viele der aktuellen Bots haben allerdings noch große Schwachstellen im Design. So zeigt das Beispiel Storm, dass ein Botnetz extrem angreifbar ist, wenn der Ursprung eines Kommandos nicht überprüft wird. Durch den Einsatz von verschlüsselter Kommunikation, wechselnden Ports und Anpassung des Kommunikationsverhaltens kann sich ein Botnetz noch besser tarnen. Es ist nur eine Frage der Zeit, bis auch solche Techniken kombiniert in einen Bot zum Einsatz kommen.

Eine effektive und langfristige Lösung des Problems ist aber nicht in Sicht. Jede Technik, die Botnetze verhindern soll, sorgt auf der Gegenseite für eine Anpassung, um diese zu umgehen.

LITERATUR

[1] E. Levy and I. Arce, "A Short Visit to the Bot Zoo," *IEEE Security & Privacy*, vol. ???, no. ???, pp. 76–79, 2005, abgerufen am 02.09.2008. [Online]. Available: ???

[2] I. Peacock, "Showing Robots the Door, What is Robots Exclusion Protocol?" *Ariadne*, vol. Issue 15, May 1998, abgerufen am 02.01.2009. [Online]. Available: http://www.ariadne.ac.uk/issue15/robots/

[3] Wikipedia, "Googlebot — Wikipedia, Die freie Enzyklopädie," 2008, stand 2. Januar 2009. [Online]. Available: http://de.wikipedia.org/w/index.php?title=Googlebot&oldid=51297108

[4] ——, "Computervirus — Wikipedia, Die freie Enzyklopädie," 2008, stand 2. Januar 2009. [Online]. Available: http://de.wikipedia.org/w/index.php?title=Computervirus&oldid=54744124

[5] Symantec, "SymbOS.Commwarrior.A," abgerufen am 02.01.2009. [Online]. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2005-030721-2716-99

[6] K. D. Mitnick and W. Simon, *Die Kunst der Täuschung: Risikofaktor Mensch*. Mitp-Verlag, März 2006.

[7] F. Dahl, "Der Storm-Worm," Diplomarbeit, März 2008. [Online]. Available: http://pi1.informatik.uni-mannheim.de/filepool/theses/diplomarbeit-2008-dahl.pdf

[8] P. Brauch, "Geld oder Netz!" *c't*, no. 14/2004, p. 48, 2004. [Online]. Available: http://www.heise.de/ct/04/14/048/

[9] K. Hayashi, "Spam from the Kernel: Full-Kernel Malware Installed by MPack," Juli 2007, abgerufen am 11.01.2009. [Online]. Available: https://forums.symantec.com/t5/Malicious-Code/Spam-from-the-Kernel-Full-Kernel-Malware-Installed-by-MPack/ba-p/305311#A139

[10] D. Bachfeld, "Rootkit verschiebt Windows in virtuelle Maschine," *heise security*, Oktober 2006, abgerufen am 26.03.2009. [Online]. Available: http://www.heise.de/security/news/meldung/print/79676

[11] J. Oikarinen and D. Reed, "Internet Relay Chat Protocol," RFC 1459 (Experimental), May 1993, updated by RFCs 2810, 2811, 2812, 2813. [Online]. Available: http://www.ietf.org/rfc/rfc1459.txt

[12] V. Kamluk, "Botnetze - Geschäfte mit Zombies," *Kaspersky Lab*, Mai 2008, abgerufen am 04.01.2009. [Online]. Available: http://www.kaspersky.com/de/downloads/pdf/vkamluk_botnetsbusiness_0508_de_pdf.pdf

[13] M. Romano, S. Rosignoli, and E. Giannini, "Robot Wars - How Botnets Work," *WindowSecurity.com*, Okt./Nov. 2005, abgerufen am 06.01.2009. [Online]. Available: http://www.windowsecurity.com/articles/Robot-Wars-How-Botnets-Work.html?printversion

[14] P. Mockapetris, "Domain names - concepts and facilities," RFC 1034 (Standard), Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592. [Online]. Available: http://www.ietf.org/rfc/rfc1034.txt

[15] ——, "Domain names - implementation and specification," RFC 1035 (Standard), Nov. 1987, updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 2137, 2845, 3425, 3658, 4035, 4033. [Online]. Available: http://www.ietf.org/rfc/rfc1035.txt

[16] P. Bächer, T. Holz, M. Kötter, and G. Wicherski, "The Honeynet Project - Know Your Enemy: Fast-Flux Service Networks," Juli 2007, abgerufen am 02.09.2008. [Online]. Available: http://www.honeynet.org/

[17] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616 (Draft Standard), Jun. 1999, updated by RFC 2817. [Online]. Available: http://www.ietf.org/rfc/rfc2616.txt

[18] B. Ungerer, "US-Provider ziehen Spam-Schleuder den Stecker," *iX News*, November 2008, abgerufen am 06.01.2009. [Online]. Available: http://www.heise.de/ix/news/meldung/print/118804

[19] J. Schmidt, "Hydra der Moderne," *c't*, no. 18/2007, p. 76, 2007, abgerufen am 02.09.2008. [Online]. Available: http://www.heise.de/security/artikel/print/94211

[20] T. Fuhrmann, "Internet Protokolle II," SS2008.

[21] I. Arce and E. Levy, "An Analysis of the Slapper Worm," *IEEE Security & Privacy*, vol. January/February, pp. 82–87, 2003.

[22] J. Stewart, "Phatbot Trojan Analysis," *SecureWorks*, März 2004, abgerufen am 21.03.2009. [Online]. Available: http://www.secureworks.com/research/threats/phatbot/

[23] C. Bryan-Low, "How Legal Codes Can Hinder Hacker Cases," *The Wall Street Journal Online*, Januar 2007, abgerufen am 04.01.2009. [Online]. Available: http://online.wsj.com/public/article_print/SB116900488955878543-yrMHYlacFyxijV14BxFZfXeU1_8_20070216.html

[24] P. Brauch, "Superwurm mit öffentlichem Quelltext," *heise online*, April 2004, abgerufen am 04.01.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/46634

[25] ca, "Virus Detail - Win32/Agobot Family," abgerufen am 05.01.2009. [Online]. Available: http://www.ca.com/us/securityadvisor/virusinfo/virus.aspx?id=37776

[26] K. J. Higgins, "Srizbi Botnet Sending Over 60 Billion Spams a Day," *DarkReading*, Mai 2008, abgerufen am 11.01.2009. [Online]. Available: http://www.darkreading.com/shared/printableArticle.jhtml?articleID=211201479

[27] B. Krebs, "Host of Internet Spam Groups Is Cut Off," *washingtonpost.com*, November 2008, abgerufen am 11.01.2009. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2008/11/12/AR2008111200658_pf.html

[28] D. Bachfeld, "Botnetz wiederauferstanden," *heise online*, Dezember 2008, abgerufen am 06.01.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/119703

[29] B. Krebs, "Srizbi Botnet Re-Emerges Despite Security Firm's Efforts," *washingtonpost.com*, November 2008, abgerufen am 11.01.2009. [Online]. Available: http://voices.washingtonpost.com/securityfix/2008/11/srizbi_botnet_re-emerges_despi.html

[30] Symantec Corporation, "Linux.Slapper.Worm," abgerufen am 30.04.2009. [Online]. Available: http://www.symantec.com/security_response/print_writeup.jsp?docid=2002-091311-5851-99

[31] J. Schmidt, "Sturmwurm-Botnetz sperrangelweit offen," *heise online*, Januar 2009, abgerufen am 21.03.2009. [Online]. Available: http://www.heise.de/newsticker/meldung/print/121310

[32] C. Klaß, "25C3: Storm-Botnet gekapert," *golem.de*, Dezember 2008, abgerufen am 27.03.2009. [Online]. Available: http://www.golem.de/print.php?a=64333

[33] P. Porras, H. Saidi, and V. Yegnewaran, "An analysis of conficker's logic and rendezvous points," SRI International, Tech. Rep., Februar 2009, abgerufen am 09.03.2009. [Online]. Available: http://mtc.sri.com/Conficker/

[34] S. Klettke, "Bundeswehr kämpft gegen Viren-Befall," *Spiegel Online*, Februar 2009, abgerufen am 21.03.2009. [Online]. Available: http://www.spiegel.de/netzwelt/web/0,1518,607567,00.html

[35] K. J. Higgins, “Widespread Conficker/Downadup Worm Hard To Kill,” *DarkReading*, Januar 2009, abgerufen am 22.03.2009. [Online]. Available: http://www.darkreading.com:80/shared/printableArticle.jhtml?articleID=212901489

[36] J. Schmidt, “Dämme gegen die SYN-Flut,” *heise security*, Dezember 2003, abgerufen am xx.03.2009. [Online]. Available: http://www.heise.de/security/artikel/print/43066

[37] B. für Sicherheit in der Informationstechnik, “Empfehlungen zum Schutz vor verteilten Denial of Service-Angriffen im Internet,” Juni 2000, version 1.1a vom 20.06.2000; Abgerufen am 24.03.2009. [Online]. Available: http://www.bsi.de/fachthem/sinet/gefahr/ddos.htm

[38] “Strafgesetzbuch – §303a Datenveränderung.” [Online]. Available: http://bundesrecht.juris.de/stgb/__303a.html

Schlauere Navigation durch Mobilfunk?

Matthias Kienzler
Betreuer: Tobias Bandh
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
E-Mail: kienzler@in.tum.de

***Kurzfassung*— Vorhandene Navigationssysteme haben den Nachteil, dass die Daten auf denen die Navigation basiert nicht aktuell sind und Staus nur berücksichtigen, falls sie TMC empfangen können. Dies wird mit Hilfe anderer Quelle versucht zu umgehen, sodass man auf den Straßen unterwegs sein kann, ohne lange im Stau zu stehen. Im folgenden wird anhand verschiedener Projekte untersucht, inwiefern Mobilfunk hilfreich sein kann um Daten über die reale Situation auf den Straßen zu gewinnen. Mit diesen Daten kann dann das Navigationssystem frühzeitig eine Ausweichroute berechnet um erst gar nicht in einen Stau hineinzugeraten. Neben einer Einführung in die Technologie der FFloating Car Data"werden sowohl großflächige Projekte von TomTom oder der University of Berkeley als auch City-Projekte, die nur einen bestimmten Bereich abdecken, vorgestellt. Alle Projekte beinhalten gute Ansätze, um in Zukunft besser durch dichten Verkehr zu gelangen und Stau von vornherein zu umfahren. Allerdings basieren die Systeme auf unterschiedlichen Technologien und auch die Funktionsweisen unterscheiden sich stark.**

***Schlüsselworte*— FCD, FPD, Stau, Route, Verkehr, Mobiltelefon, GPS**

I. Einleitung

Jeder kennt das Problem des Staus, sobald man in den Urlaub fährt oder es eilig hat um rechtzeitig bei einem Termin zu sein. Vermeiden lassen sich Staus aber leider nicht. Allerdings kann man sich die Frage stellen, wie man einen Stau schon frühzeitig antizipieren kann, um gar nicht in diesen hinein zu geraten. Damit es überhaupt zu einem Stau kommt braucht es nicht viel. Es reicht ein abruptes Abbremsen, eine Lücke im Verkehr oder ein Spurwechsel bei dichtem Verkehr, das heißt circa 30 Fahrzeuge pro Minute pro Spur. Sofort fängt es an zu stocken. Und wenn man einmal in einem Stau drin ist, ist die Aussicht schnell wieder herauszukommen sehr gering. Forscher haben herausgefunden, dass sich Stau mit einer Geschwindigkeit von nur 15 km pro Stunde ausbreitet. Man kann also nur hoffen, dass sich dieser schnell wieder auflöst [1]. Das beste bekannte Mittel um dem Ärger zu entgehen sind bislang die Verkehrsmeldungen im Radio, die bei frühzeitigem Hören durchaus sehr hilfreich sein können. Allerdings liegen die Probleme bei diesem Dienst darin, dass die Daten nicht automatisch verarbeitet werden, diese zu ungenau sind und natürlich auch zum richtigen Zeitpunkt das Radio eingeschaltet sein muss, denn die Kapazität dieses Kanals ist beschränkt. Wäre dies nicht so, würden nur noch Verkehrsnachrichten im Radio laufen. Eine andere Möglichkeit hat man nicht, wenn man schon unterwegs ist. Vielen ist auch bekannt, dass man sich im Internet Staus anzeigen lassen und dann auch gleich nach Alternativen schauen kann, aber das muss vor Abfahrt stattfinden und kann sich geändert haben bis man an besagter Stelle ist. Deshalb gibt es verschiedene Ansatzpunkte mit denen man den Staus auf den Grund geht und die gewonnenen Informationen direkt und ohne Zeitverzögerung an die Autofahrer übermittelt. Einige der neueren Techniken werde ich versuchen zu erläutern.

II. Floating Car Data

FFloating Car Data"(FCD) bezeichnet eine Technik bei der Autos als mobile Sensoren dienen um den Verkehrsfluss zu ermitteln.

A. Allgemeines

Damit die Autos als Sensoren genutzt werden können wird auf die GPS-Technologie zurückgegriffen, durch welche die aktuelle Position und die Geschwindigkeit des Fahrzeugs berechnet wird. Als Sensoren dienen GPS-fähige Handys. Diese senden die Daten dann über das Mobilfunknetz an eine Verkehrszentrale. GPS eine Abkürzung für "Global Positioning System", also ein weltweites System um Positionsbestimmungen durchzuführen. Es wird versucht die Positionen und die Positionsveränderungen von Autos über diese GPS-Signale zu erfassen. Dazu erst einmal ein kurzer Einblick, wie die GPS-Technologie überhaupt funktioniert. Es gibt Satelliten im Weltall die ständig ihren Ort und die Uhrzeit ausstrahlen. Ein GPS-Empfänger berechnet aus diesen Daten dann zunächst seine exakte momentane Position. Werden die kontinuierlichen Messung nun in Relation zueinander gesetzt kann die Geschwindigkeit des Empfängers ermittelt werden [2]. Ëxtended Floating Car Data"(XFCD) ist eine Erweiterung von FCD, die weiter Daten außer der Position aus dem Auto heraus an die Zentrale übermittelt. Wenn also nicht nur Ort und Geschwindigkeit über GPS bestimmt und übertragen werden, sondern zusätzlich auch noch Daten von ABS, ASR, ESP, Regensensoren [3]. Dadurch kann in der Zentrale ermittelt werden, wie die äußeren Begebenheiten im Moment sind. Springen zum Beispiel ABS und ESP an, dann ist die Straße vereist oder, falls durch den Regensensor die Scheibenwischer starten, so wird deren Geschwindigkeit ermittelt und die Informationen weitergegeben. Einer der größten Vorteile der

FCD-Technik sind, im Vergleich zu den bisher eingesetzten Systemen der Verkehrserfassung, die geringen Kosten um an Informationen zu kommen. Bislang musste der Verkehr über Sensoren, die in den Straßenbelag eingebaut waren gemessen oder über Radarsysteme und Kameras am Straßenrand erfasst werden. Allerdings verursachen diese Systeme hohe Kosten für Anschaffung, Installation und Wartung, sodass nur Autobahnen damit bestückt sind. Durch FCDs lassen sich Straßeninformationen sehr viel billiger und genauer gewinnen. Außerdem ist die Anzahl der abgedeckten Routen nicht auf Autobahnen beschränkt, sondern ausgedehnt auf alle befahrbaren Straßen [4], [5]. Die Fragen sind nun warum Handynutzer zustimmen sollten ihre GPS-Daten "herzugeben", welche Vorteile sie davon hätten und ob sie dadurch überwachbar sind. Der Vorteil ist natürlich, dass durch viele FCDs das Bild der Straßenlage genauer wird. Und von dieser Genauigkeit profitiert jeder Autofahrer, der in irgendeiner Weise auf eine Technik zurückgreift, bei der eine Route mit Hilfe von FCDs berechnet wird. Die Überwachung des Handybenutzers wäre theoretisch möglich. Allerdings garantieren die Betreiber, dass sich nicht nachzuvollziehen lässt, von welchem Handy das Signal ausgeht und auch, dass die Route nicht gespeichert wird. Ebenfalls wird zugesichert, dass keine persönlichen Daten weitergegeben werden, sondern nur die Lokalität des Signals. Dadurch wäre die Anonymität gewahrt. Zustimmen sollten Handynutzer einfach deswegen, weil keine Nachteile entstehen, da die Privatsphäre garantiert ist [4].

B. Architektur und Funktionsweise

Das Funktionsprinzip dieser Technik ist ziemlich einfach. Um von der oben beschriebenen GPS-Technik zur FCD-Technik zu kommen braucht es nur noch eine Funktion im Endgerät, dass dieses die berechneten Daten auch wieder freigibt und an eine Zentrale übermittelt. Dazu ist ein Programm nötig, das in einem bestimmen Zeitintervall, die durch GPS genau bestimmte Position, über das Mobilfunknetz aussendet. Nach einer hinreichenden Anonymisierung und der Konsolidierung mit den anderen FCDs wird dort ein virtuelles Bild der Straße gezeichnet. In diese komplexe Berechnung der Straßensituation fließen auch noch andere Daten ein, wie zum Beispiel durch Sensoren direkt an Straßen gewonnen Informationen oder Beobachtungen von Verkehrsmelder.

Am Ende des Prozesses steht ein sehr genaues und auch sehr schnell erstelltes Realbild von sehr vielen Straßen, das dann wieder zurück an die Autofahrer gereicht werden kann oder aber zu anderen Zwecken weiterverarbeitet wird. Die Übertragung an die Navigationsgeräte im Auto erfolgt wiederum über das Mobilfunknetz. Damit diese Übertragung funktioniert muss das Gerät natürlich mit einer SIM-Karte ausgestattet sein. Falls das Gerät mit einem Radioempfänger ausgestattet ist, so gibt es einen alternativer Weg der Datenübermittlung, nämlich über TMC. Der große Vorteil an der Übertragung über das Mobilfunknetz ist, dass das Gerät jederzeit Informationen über eine spezifische Route anfragen kann. Die Übermittlung über TMC ist hingegen regional begrenzt und wird nur periodisch ausgestrahlt. Abbildung 1 zeigt die verschiedenen Informationsquellen und den, aus der Übertragung an den Autofahrer, entstandenen Nutzen - dieser kann frühzeitig die staubelastete Straße verlassen [3], [5]–[7].

Abbildung 1. Konsolidierung der Informationen

C. Abgrenzung zu Floating Phone Data

Neben den FCDs gibt es noch die sogenannten FFloating Phone Data"(FPD). Der Unterschied ist, dass FCDs über Ortsbestimmung via GPS funktionieren und FPDs werden gewonnen indem die Bewegungen von Mobiltelefonen nachvollzogen werden. Jedes Handy mit aktivem Gespräch ist an mindestens einem Mobilfunkmasten angemeldet. Nun lässt sich das Gerät lokalisieren, da der Mobilfunkanbieter die Informationen über die Position der Masten speichert, über welchen das Gespräch geht. Diese Masten haben ein bestimmtes Gebiet das sie abdecken (Abbildung 2, links). Meldet sich nun ein Handy an einem Mast ab, weil es aus dem Abdeckungsbereich herausgeht, so meldet es sich automatisch und gleichzeitig am nächsten Mast an, zu dem das Gebiet gehört, in dem sich das Handy gerade befindet. Diese An- und Abmeldevorgänge werden als Zellwechsel bezeichnet, weil das Handy von einer Zelle in die nächste wechselt. Ist das Handy inaktiv, so muss der Tracing Mode aktiviert sein um eine Lokalisierung durchzuführen.

Gibt es nun einen Kooperation eines Mobilfunkanbieters mit einem Anbieter für Stauservice, so kann sehr leicht durch die, von den Zellwechseln gespeicherten, Daten die Route des Mobiltelefons nachvollzogen werden. Auch die ungefähre Geschwindigkeit kann errechnet werden indem die Zeit zwischen den Zellwechseln berechnet wird. Nun hat man also die Positionsveränderung eines Handys und die Geschwindigkeit mit der diese Veränderung geschieht. Diese Informationen können nun mit einem Straßennetz abgeglichen werden und so kann die Route dieses einen Handys verfolgt werden. Alle auf diese Weise gewonnen FPDs werden nun konsolidiert und diese

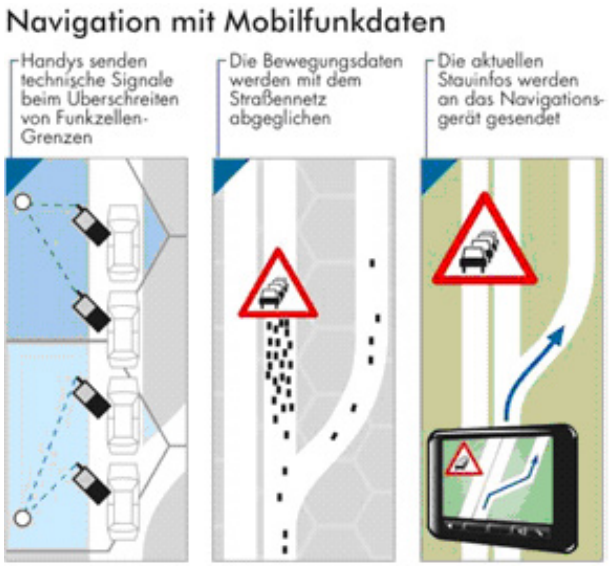

Abbildung 2. Floating Car Data

Daten dann mit dem Straßennetz abgeglichen. So erhält man also eine Vielzahl von verschiedenen Daten über den gleichen Streckenabschnitt und kann daraus nun ein Modell erstellen, wie die verkehrstechnische Lage auf diesem Streckenabschnitt gerade ist (Abbildung 2, mitte). Diese Informationen werden nun wieder weitergereicht und nachfolgende Autofahrer profitieren - wie in Abbildung 2, rechts gezeigt - davon, indem sie die Straße frühzeitig verlassen [8], [9].

III. Projekte

Im folgenden Teil werden Projekte vorgestellt, die zum Ziel haben Daten über die aktuelle Straßenlage zu gewinnen. Bei den vorgestellten Projekten geschieht die Datengewinnung jeweils unterschiedlich.

A. Mobile Millenium Project

Die University of California, Berkeley, und der Handyhersteller Nokia haben gemeinsam dieses Projekt initiiert. Ziel war es die zu diesem Zeitpunkt theoretisch existierenden Erkenntnisse über FCDs als Hilfsmittel zur Navigation praktisch nachzuvollziehen. Vorrangiges Ziel war es ein effizientes lauffähiges System zu entwerfen, zu testen und dann auch unter realen Bedingungen einzusetzen. Aber auch die Abwägung zwischen Abschätzungsgenauigkeit des Verkehrs, Schutz der Persönlichkeitsrechte und Intimsphäre der Probanden und der entstehenden Kosten spielte eine große Rolle. Für das Projekt ist eine Java-Software geschrieben worden, die alle Daten automatisch an einen Server der Universität übermittelt. Allerdings nur, falls der Nutzer zustimmt. Die Daten werden per GPS ermittelt und beinhalten die Position bis auf wenige Meter und die Geschwindigkeit bis auf circa 3 Meilen pro Stunde genau. Diese Software wurde auf Nokia, N95 Handys geladen, die dann in 100 Testautos platziert wurden. Die Testpersonen sind Studenten, deren Aufgabe es ist in einem bestimmten Testgebiet, mit dem präparierten Auto während des Versuchs zu fahren. Die gesendeten Daten werden sofort anonymisiert. Das Senden und Verwahren der Daten ist gesichert durch eine sehr gute Verschlüsselung. Treffen die Daten am Server ein werden sie unverzüglich mit allen anderen Daten zusammengefügt, um so das Bild der aktuellen Straßensituation zu erhalten. Diese Daten werden dann zunächst im Internet veröffentlicht, sodass sich jeder eine Vorstellung darüber verschaffen kann, wie es auf dem Testabschnitt momentan aussieht. Außerdem werden die Daten statistisch ausgewertet und über das Mobilfunknetz an Empfänger versendet [5], [10].

B. TomTom-Projekte

Der Navigationsgeräte-Spezialist TomTom ist seit langer Zeit auf der Suche nach besseren Navigationsmöglichkeiten. Dazu gehört neben einer Verbesserung der Routenberechnung auch eine kluge Stauumfahrung. Die Routenberechnung ist von daher ein Problem, da oft nur die großen Straßen genutzt und die kleinen außen vor gelassen werden. Oder aber auf den berechneten Routen Behinderungen durch Baustellen oder ähnliches auftreten. Um dies zu verbessern wurde zuerst die Map-Share-Funktion ins Leben gerufen. Die nächste Neuerung war IQ-Routes, das ebenfalls einer besseren Routenberechnung dient und vor allem die tatsächlich benötigte Zeit einberechnet. Die Stauumfahrung ist ein weitaus größeres Problem, das mit dem neusten Projekt von TomTom - HD-Traffic - gelöst werden soll. In dieses Projekt fließen ebenfalls die aus Map-Share und IQ-Route gewonnenen Daten und Statistiken ein.

1) Map-Share: Diese Funktion erlaubt es dem Benutzer eine berechnete Route zu verbessern und diese Verbesserungen dann via Internet an TomTom zu übermitteln. Ist beispielsweise eine Straße gesperrt, die laut Navigation genutzt werden sollte, so kann dies abgespeichert werden, sodass später berechnete Routen nicht mehr über diese Straße führen. Abgespeichert werden können auch Routen von denen ein ortskundiger Fahrer denkt, dass diese geschickter zu fahren seien als die angegebenen. Wird das Gerät dann an einen Computer angeschlossen, so werden die gesicherten Daten an TomTom geschickt und Alternativrouten von anderen Usern empfangen und auf dem Navigationsgerät abgespeichert [11], [12].

2) IQ-Routes: Dieser Dienst bietet ebenfalls eine bessere und zeitlich genauere Berechnung einer optimalen Route an. Das Navigationsgerät speichert den kompletten Verlauf der Fahrt. Also tatsächlich gefahrene Strecke, Tempo und Dauer. All diese gespeicherten Daten werden bei einer Computeranbindung via Internet an TomTom übertragen und in einer Datenbank gespeichert, in der auch die Daten der anderen IQ-Routes-Nutzer gespeichert sind. So hat TomTom Zugriff auf einen sehr großen Erfahrungsschatz der auf tatsächlich fahrbaren Geschwin-digkeiten, statt auf theoretisch möglichen Höchstgeschwindig-keiten beruht. Auch Straßenbegebenheiten und Umwelteinflüsse werden registriert und berücksichtigt.

Abbildung 3. Standardgeschwindigkeitsnetz (London)

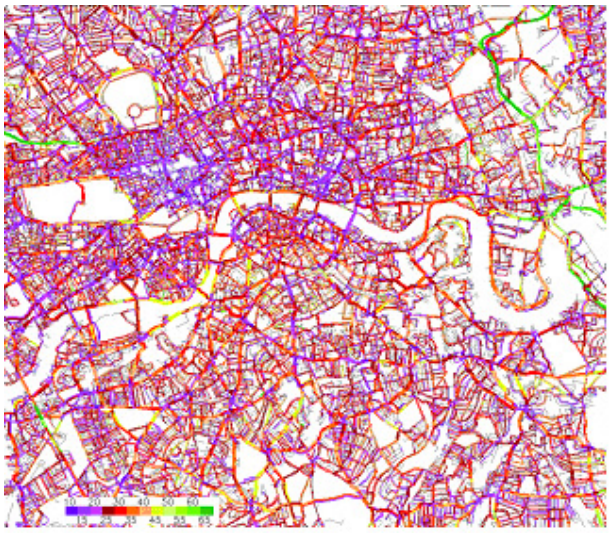

Abbildung 4. IQ-Routes-Geschwindigkeitsnetz (London)

Für die meisten Straßen wurden Geschwindigkeitsdurchschnittswerte bestimmt, die auf den theoretischen Höchst-geschwindigkeiten beruhen (Abbildung 3). Auf der Basis dieses Geschwindigkeitsnetzes erfolgt bei einem herkömmlichen Navigationsgerät nun für jede Route die Berechnung. Hierbei bedeutet grün, dass eine schnelle Fahrt möglich ist, also keine Behinderungen auf den Straßen sind. Dass dies nicht der Realität entspricht ist eindeutig in Abbildung 4 zu sehen. Hier wurde dank IQ-Routes das Geschwindigkeitsnetz angepasst. Es wurden wieder die Durchschnittsgeschwindigkeiten aus den erlangten Daten errechnet und über das gleiche Straßennetz gelegt. Außerdem wird eine statistische Erhebung durchgeführt zu welchen Tageszeiten sich die Durchschnitts-geschwindigkeit wie verhält. Die Farbe rot bedeutet, dass auf diesen Straßen nur langsam gefahren werden kann. Nun sieht man deutlich, dass so gut wie alle Straßen rot markiert wurden. Es kann also nicht von normalen Geschwindigkeiten und flüssigem Verkehr ausgegangen werden, sondern es ist mit Verstopfungen und zähem Verkehr zu rechnen. Das beeinflusst die Routenberechnung natürlich elementar, denn wie in Abbildung 4 zu erkennen ist, sind kaum noch Straßen grün gefärbt. In Abbildung 3 hingegen wird der von den Kilometern her kürzeste Weg ausgewählt, sodass die Route nicht über eine rote Straße führt.

Durch ortskundige Fahrer steigt auch die Anzahl der verschiedenen Alternativrouten, die bei der Berechnung einbezogen werden. Denn diese wissen durch eigene Erfahrungen, wann es besser ist eine bestimmte Strecke oder einen bestimmten Knotenpunkt zu meiden und fahren dann automatisch über eine Ausweichroute. Desweiteren wird auch Tag und Uhrzeit abgespeichert, sodass es möglich ist an verschiedenen Wochentage zu verschiedenen Uhrzeiten eine andere Strecke zu berechnen, die exakt zu diesem Zeitpunkt die beste ist. Aufgrund all dieser realen Informationen ist eine sehr viel bessere und zeitminimierende Routenführung möglich. Bei jeder Internetanbindung des Geräts werden nun nicht nur die gespeicherten Daten übertragen sondern auch neue Informationen von anderen Usern auf das eigene Navigationsgerät geladen, sodass die Berechnung einer Route immer auf höchstem Niveau stattfinden kann [13]. Etwas deutlicher wird das ganze, wenn man sich einen Fahrer vorstellt, der innerhalb von London möglichst schnell an ein bestimmtes Ziel kommen möchte. Herkömmliche Navigationsgeräte würden den Weg wie in Abbildung 5 berechnen. Es ist der kürzeste Weg von den Kilometern her und auch die Zeit ist geringer als auf anderen Routen, da von Geschwindigkeiten ausgegangen wird, die gefahren werden könnten bei freier Fahrt. Allerdings verlängert sich die reale Fahrzeit automatisch durch Verkehr, Ampel, Fußgänger, et cetera. Diese reale Fahrzeit wird in Abbildung 6 dargestellt.

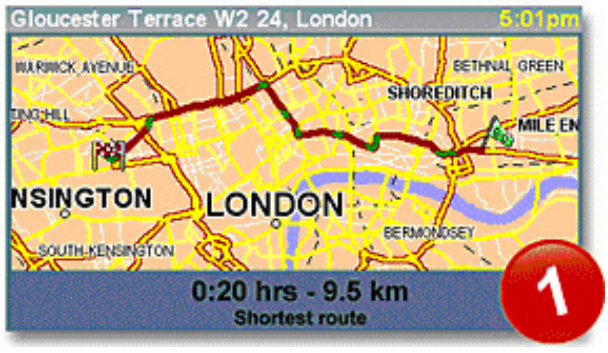

Abbildung 5. Standardberechnung - Kürzester Weg [14]

Dank IQ-Routes ist es nun möglich nicht nur Schätzdaten in die Routenberechnung einfließen zu lassen sondern auch die realen Erfahrungen von vielen anderen Nutzern die nach der Internetanbindung auf dem Gerät vorhanden sind. Dadurch wird eine komplett andere Route berechnet, die nicht zum Ziel

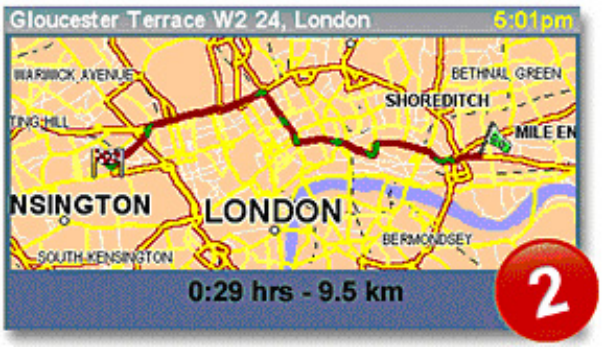

Abbildung 6. Standardberechnung - Tatsächlich benötigte Zeit [14]

hat einfach nur den kürzesten und schnellsten Weg nach dem Standardverfahren zu bestimmen, sondern es erfolgt auch eine Abschätzung, welche Route dank der neuen Informationen am geschicktesten für den Fahrer ist. Diese neue, laut Navigationsgerät, bessere Route ist in Abbildung 7 gezeigt. Sie ist kilometermäßig etwas länger, dafür aber schneller zielführend [11], [13], [15].

Abbildung 7. IQ-Routes-Berechnung - Schnellste Route [14]

3) HD-Traffic: Die neuste Innovation von TomTom ist HD-Traffic. Dies bezeichnet einen Service, mit dem es möglich ist, durch die Verfolgung von Mobiltelefonen, Informationen über den Verkehrsfluss zu bekommen. Dahinter steckt das Prinzip der FPDs (siehe 2.3). Um an die Mobilfunkdaten zu kommen arbeitet TomTom mit Vodafone zusammen an diesem Projekt [6]. Vodafone anonymisiert die Daten und übergibt dann nur die Daten bezüglich der Bewegung der Handys an eine TomTom-Zentrale. Es wird also sehr auf Datenschutz geachtet, sodass es nicht möglich ist die Daten zurückzuverfolgen. Diese Daten werden nun kombiniert mit klassischen Verkehrsinformationen von Landesmeldestellen, den FPDs von vielen Tausend FFahrzeugen aus dem Bereich von Flottenmanagement-Systemen“ [6] und den Daten aus IQ-Routes. Zusätzlich fließen in die Verkehrsberechnung FCDs ein, die von Speditionen geliefert werden. Denn diese müssen immer informiert sein, wo der Lastwagen im Moment ist. Alle diese Daten zusammen fließen in die Vorhersage der aktuellen Straßensituation ein und können dann an die Autofahrer weitergegeben werden [6], [11], [16]. Um die verteilungsfertigen Daten an die Navigationsgeräte zu senden gibt es nun verschiedene Techniken. Zum einen besteht die Möglichkeit der Übermittlung durch TMC. Das ist eine Technologie, die den Radiokanal benutzt um Verkehrsinformationen zu senden. Dazu müsste das Navigationsgerät mit einem Radioempfänger ausgerüstet sein. Bei dieser Technik besteht das Problem, dass die Nachrichten in manchen Fällen nicht zeitnah oder unvollständig übertragen werden, denn es muss immer eine Verbindung zu einem Sendemast bestehen. Ist der nächste Mast zu weit entfernt oder fährt man durch Tunnel, so wird das Signal unterbrochen und der Dienst fällt aus. Auch auf die Qualität der Nachricht hat TomTom keinen Einfluss. Darum hat sich eine andere Technik durchgesetzt. TomTom stattet die Navigationsgeräte nun mit SIM-Karten aus, über welche diese ständig mit neuen Informationen über das Mobilfunknetz versorgt werden können. So erlangt man Unabhängigkeit vom Radionetzwerk und kann Informationen mit größerer Geschwindigkeit und Zuverlässigkeit übermitteln. Da die Übertragung aus einer konzerneigenen Zentrale erfolgt ist auch die Qualität immer auf dem gewünschten Standard. Die Gefahren bei dieser Technik ist, dass sich eventuelle keine Verbindung herstellen lässt und dadurch dann keine Daten auf das Navigationsgerät übertragen werden können [7].

C. Innerstädtische Projekte

Neben dem großangelegten Projekten der University of Berkeley oder den bereits realisierten Projekten von TomTom, die darauf abzielen große Fläche abzudecken, gibt es mehrere kleinere Projekte, die in der Erforschung und Vorhersage künftiger Stauentwicklung und der Stauumfahrung innerhalb von großen Städten ihr Ziel haben. Diese lokal begrenzten Projekte sind allerdings noch nicht ausgereift genug um in großem Umfang kommerziell eingesetzt zu werden.

1) Nagel-Schreckenberg-Simulation: Vorweg ist zu sagen, dass es hier bislang noch keine Verwendung des Mobilfunks gibt. Allerdings bietet die Simulation gute Möglichkeiten den Verkehr zu bestimmen. Außerdem kann man die Unterschiede der Techniken die auf Mobilfunk basieren und solcher ohne Mobilfunk erkennen. Der Ansatz beruht nicht darauf auf Verkehr zu reagieren und dann unterwegs auf einen Ausweichroute auszuweichen, sondern schon von vornherein zu wissen wie sich der Verkehr entwickeln wird und gleich die beste Route zu wählen. Das Simulationsgebiet bezieht sich auf Nordrhein-Westfalen. Durch die Computersimulationen ist es inzwischen möglich die Verkehrslage bis zu einer Stunde im Voraus zu ermitteln und die Ergebnisse dann im Internet zu veröffentlichen [1]. Um diese Prognosen anzustellen ist ein Computersystem entwickelt worden mit dem der Verkehr simuliert wird. Darin sind die betrachteten Straßen dargestellt durch Zellen, die genauso lang sind wie ein Auto samt Abstand zum Vordermann. Eine Zelle ist nun entweder leer, das heißt auf diesem Streckenabschnitt ist kein Verkehr, oder es befindet sich genau ein Auto in der Zelle. Dies bedeutet dann einfach, dass sich ein Auto auf dem Streckenabschnitt befindet Es kann nicht sein, dass sich 2 Autos in einer Zelle befinden oder

dass ein Auto auf der Grenze zur nächsten Zelle steht. Die Autos können sich nun mit verschiedenen Geschwindigkeiten bewegen. Um eine Zelle vorzurücken wird die Geschwindigkeit 1 angenommen, was bedeutet, dass sich das Auto mit 27km/h bewegt. Die Höchstgeschwindigkeit beträgt 5, was so viel bedeutet wie 135 km/h. Hier wird das Auto dann also um 5 Zellen vorwärts bewegt . Nun wird also jedem Auto in der Simulation, bevor diese startet, eine Geschwindigkeit und eine Route vorgegeben. Erst danach starten die Autos sich, wie vorgegeben, zu bewegen. In der Simulation wechseln die Autofahrer jetzt die Spuren bei dichtem Verkehr oder trödeln bei freier Fahrt, immer auf der Grundlage des ßchlechtestmöglichen Verkehrs“. Das Problem ist, dass es inzwischen zwar sehr gut geht den Verkehr zu prognostizieren, wenn es keine Zwischenfälle gibt, allerdings ist das System anfällig bei Unfällen oder sonstigen unvorhersehbaren Ereignissen, denn bis diese verarbeitet sind dauert es sehr lange und somit wird die ganze Vorhersage ungenau [17], [18].

2) Taxi-FCD: Das Berliner Institut für Verkehrsforschung (IVF) hat tausende neue FCDs erschlossen, mit dem Ziel den Verkehr in den Metropolen besser zu regulieren und regeln. Dazu hat das Institut eine Kooperation mit den Taxi-Unternehmen geschlossen, um an die GPS- und Funkdaten zur Zentrale der Taxis zu gelangen. Mit diesen Bewegungsdaten der Taxis lässt sich jetzt die Straßensituation modellieren, indem sie mit Erfahrungswerten über den Tage und den Zeitpunkt zusammengebracht und dann mit dem Straßennetz der Stadt abgeglichen werden. Schließlich erfolgt eine Übermittlung der Daten über das Mobilfunknetz an die Empfänger. Durch die Speicherung des Verkehrsflusses zu bestimmten Uhrzeit an den Tagen lässt sich für jeden Tag der Woche und für jeden Uhrzeit dieses Tages eine individuelle Berechnung aufstellen, welche Route gut zu fahren ist und welche besser nicht gewählt werden sollte. Die Aktualisierung der momentanen Verkehrslage geschieht mehrmals pro Minute, sodass die Informationen immer hochaktuell sind. Taxis eignen sich deshalb sehr gut, da sie sowieso mit einem GPS-Empfänger ausgestattet sind. Über diesen hat die Taxi-Zentrale immer den Überblick welches Taxi momentan wo ist und wie schnell es fährt. Dies kann sehr nützlich sein um zu ermitteln, welches Taxi den kürzesten Weg zu einem neu eintreffenden Kundenauftrag hat. Ein weiterer Vorteil von Taxis ist, dass sie zuverlässig in Bewegung sind und auch meistens nur innerhalb der Stadt. Dadurch ist die Genauigkeit größer, als wenn unklar ist wo und ob überhaupt ein Auto sich in Bewegung befindet. Außerdem sind die von den Fahrern gewählten Routen oft identisch, sodass sich durch viele Daten über den gleichen Streckenabschnitt sehr gut abschätzen lässt wie der Verkehr dort ist. Die Route ist meistens eine große Straße, weil Taxis kaum über kleinere fahren. Dadurch sind dann automatisch alle kritischen Stellen in einem Stadtverkehr abgedeckt. Das große Ziel dieser Technik ist es die Durchschnitts-geschwindigkeit in großen Städten zu erhöhen. Momentan kann diese bei circa 15km/h festgesetzt werden. Realistisch sind, so die Forscher, bis zu 19 km/h. Erklären lässt sich die Erhöhung durch das frühzeitige Erkennen von Hindernissen, die dann umfahren werden können. Aber auch andere Verkehrsteilnehmer als die Autofahrer können von dieser Technologie profitieren. Zum Beispiel kann so die Route der Müllabfuhr dem Verkehr angepasst werden, oder auch die Stadtreinigung kann besser planen zu welchem Zeitpunkt sie an einer Stelle der Stadt arbeitet [**?**], [1], [19].

IV. Zusammenfassung

Alle hier vorgestellten Projekte bieten sehr gute Möglichkeiten das Problem des Staus zu vermindern. Allerdings lassen sich doch Unterscheidungen machen in der Qualität der Informationen, der Verarbeitungs- und Verbreitungszeit und den Distributionswegen. Die Qualität von Informationen ist natürlich sehr stark abhängig von der Zeit die vergeht zwischen der Registrierung eines Ereignisses und der Weitergabe an den Autofahrer. Ein weiterer wichtiger Aspekt ist der Weg auf dem die Informationen letztlich zum Nutzer kommen und in welcher Weise er dann beteiligt ist an der Endverarbeitung, um die beste Strecke herauszufinden. Es lässt sich also feststellen, dass es durchaus Chancen gibt die Navigation durch Mobilfunk zu verbessern. Denn die Ansätze, die auf FCD (Mobile Millennium Project und Taxi-FCD) oder FPD (TomTom HD-Traffic) basieren haben mehr Vorteile als solche, auf Basis von Simulationen oder den Erfahrungen der Autofahrer. Die bei den FCD-/FPD-Projekten gewonnen Informationen sind hochaktuell und schnell soweit verarbeitet, dass ein klares Bild der Straße entsteht. Der größte Vorteil, im Vergleich zu den Methoden die das Mobilfunknetz nicht nutzen, ist aber sicherlich die Verteilung über dieses direkt an die Navigationsgeräte. So hat der Autofahrer immer die neusten Daten auf dem Display und das beste für ihn daran ist, dass er sich nicht selber um die optimale Route kümmern muss, sondern das Navigationsgerät automatisch neu eintreffende Daten einberechnet. Kommt also die Meldung, dass eine eigentlich vorgesehen Route ungünstig ist, so wird automatisch eine andere berechnet und der Fahrer merkt dies nicht einmal unbedingt. Durch diese Technik ist der Erfolg eindeutig am größten und zusätzlich dazu auch noch an besten für den Fahrer, denn er kann ßtur“dem Navigationsgerät nachfahren und trotzdem sehr schnell am Ziel sein. Das Problem bei HD-Traffic ist aber, dass dieser Service erst seit kurzem angeboten wird und auch erst zwei Modelle von TomTom unterstützt werden. Die Durchdringung ist also sehr gering. Um die vollständigen Vorteile nutzen zu können müssten vermutlich sehr viel mehr dieser Geräte im Straßenverkehr zu finden sein. Vieles deutet daraufhin, dass der Service zur Zeit hauptsächlich über Statistiken läuft und weniger über tatsächliche Informationen in Echtzeit. Dies lässt sich aber aufgrund der dürftigen Quellenlage nicht mit Sicherheit sagen. Der Zeitaspekt ist bei der Nagel-Schreckenberg-Simulation eindeutig schlechter als bei den anderen. Denn hier braucht es länger, bis alle Informationen verarbeitet wurde. Was bei diesem Ansatz aber der noch gravierendere Unterschied ist, ist die Tatsache, dass sich der Autofahrer selbst um die beste Route kümmern muss. Er muss am besten noch vor Reiseantritt im Internet recherchieren, was laut Simulation die beste

Route wäre. Aber klar ist auch, dass sich auf der ausgewählte Route bis zum Zeitpunkt zu dem der Autofahrer tatsächlich die Straße fahren will wieder ein Stau ergeben kann, der in diesem System noch nicht erfasst war. Es ist festzustellen, dass dieses Projekt nicht auf die Vorteile des Mobilfunknetzes zurückgreift, obwohl die Nutzung des Mobilfunks eventuelle weitere Vorteile für das Projekt bringen würde. Es werden keine Daten daraus verarbeitet und auch eine Übertragung über das Mobilfunknetz findet nicht statt. Ebenfalls ohne die Nutzung des Mobilfunks sind die Projekte Map-Share und IQ-Routes zu betrachten, da die Kommunikation nur über das Internet abläuft. Hätten die Geräte, die diese Services unterstützen, eine Möglichkeit, die gewonnenen Daten direkt zu übermitteln und zu empfangen, dann wäre die Verbesserung der Routenführung aktueller und dadurch noch größer. Es muss aber auch angemerkt werden, dass die Daten die aus beiden Projekten gewonnen werden statistisch erfasst und bei der HD-Traffic-Technik verwendet werden. Merklich erleichtert werden kann die Navigation also nur durch den Einsatz von Mobilfunk, auch wenn diese Techniken auf Daten zurückgreifen, die nicht mit Hilfe dessen gewonnen wurden. Allerdings wurde noch keine endgültige und alle zufriedenstellende Lösung erarbeitet. Deshalb sind viele Projekte auch auf kleine Bezirke beschränkt und noch nicht in kommerziellem Einsatz

V. Ausblick

Letztlich lässt sich also festhalten, dass es schon eine sehr ausgefeilte Technik gibt, aber diese noch nicht flächendeckend genug eingesetzt wird. Die größte Abdeckfläche hat das TomTom HD-Traffic Projekt, das zur Zeit in den Niederlanden und in Deutschland aktiviert ist. Vereinbarungen über eine Zusammenarbeit gibt es auch schon mit Mobilfunkanbietern in Frankreich und der Schweiz. Das mittelfristige Ziel ist also das System in ganz Europa verfügbar zu machen, und falls dies gut gelingt ist eine weiter Expansion nicht ausgeschlossen. Es ist anzunehmen, dass auch TomTom alles daran setzten wird das Verfahren immer weiter zu verbessern, um seine führende Stellung in Sachen Stauumfahrung zu behalten oder sogar auszubauen. Auch bei Projekten die bislang nichts mit Mobilfunk zu tun haben, wie dem Nagel-Schreckenberg-Modell sind Verbesserungen denkbar, durch welche sich die Vorhersage noch genauer berechnen ließe. Zum Beispiel könnten die statistischen Erhebungen der bei anderen Projekten gewonnen FCDs in die Simulation einfließen. So hätte man den Vorteil, dass das Verkehrsaufkommen zu bestimmten Tageszeiten bekannt ist. Ein weitere großer Vorteil wäre natürlich die direkte Übermittlung der Ergebnisse an die Empfänger. Dazu wäre das Mobilfunknetz die geeignetste Möglichkeit zur Datenübertragung, dieser Kanal fehlt allerdings bislang. Zwar wird sich das große Problem des Staus an sich nicht lösen lassen über solche Projekte, aber es kann sich doch erheblich reduzieren, wenn die Autos nicht alle im Stau stehen, sondern viele Autos auf Ausweichrouten unterwegs sind.

Zusätzliche Literatur: [5], [20]–[25]

Literatur

[1] J. Wegner and M. Efler, "Verkehr: Den stau vorhersagen wie das wetter." [Online]. Available: http://www.focus.de/auto/unterwegs/verkehr-den-stau-vorhersagen-wie-das-wetter_aid_207940.html

[2] "Global positioning system." [Online]. Available: http://www.itwissen.info/definition/lexikon/global-positioning-system-GPS-GPS-System.html

[3] "Floating car data." [Online]. Available: http://www.itwissen.info/definition/lexikon/floating-car-data-FCD.html

[4] U. Fastenrath, "Floating car data on a larger scale." [Online]. Available: http://www.ddg.de/pdf-dat/ddgfcd.pdf

[5] S. Yang, "Joint nokia research project captures traffic data using gps-enabled cell phones." [Online]. Available: http://berkeley.edu/news/media/releases/2008/02/08_gps.shtml

[6] A. Strobel, "Großer stauangriff." [Online]. Available: http://www.connect.de/themen_spezial/HD-Traffic-Grosser-Stauangriff_3771410.html

[7] TomTom, "All about traffic tomtom's visions." [Online]. Available: www.tomtom.com/lib/img/hdt/doc/Whitepaper.doc

[8] T. Kuhn, "Wie handys zu staumeldern werden." [Online]. Available: http://www.wiwo.de/technik/wie-handys-zu-staumeldern-werden-383418

[9] J. Rähm, "Navteq will handydaten zur stauvermeidung nutzen." [Online]. Available: http://www.teltarif.de/arch/2009/kw04/s32689.html

[10] Heise-Online, "Gps-handys zur verkehrsvorhersage." [Online]. Available: http://www.heise.de/newsticker/GPS-Handys-zur-Verkehrsvorhersage--/meldung/119376

[11] A. News, "Tomtom bringt bessere iq-routes-version und hd-traffic." [Online]. Available: http://www.auto-news.de/navigationssysteme/anzeige_Neue-TomTom-Highlights-Bessere-IQ-Routes-Version-und-HD-Traffic_id_22409

[12] TomTom, "Map share technology." [Online]. Available: http://www.tomtom.com/page/mapshare

[13] A. Strobel, "Tomtom iq routes." [Online]. Available: http://www.connect.de/themen_spezial/Intelligente-Navigation_3770730.html

[14] [Online]. Available: http://www.tomtom.com/page/iq-routes

[15] TomTom, "Iq routes." [Online]. Available: http://www.tomtom.com/page/iq-routes

[16] ——, "Tomtom hd traffic." [Online]. Available: http://www.tomtom.com/services/service.php?id=2

[17] "Nagel-schreckenberg-modell." [Online]. Available: http://www.ptt.uni-duisburg.de/fileadmin/docs/paper/1992/origca.pdf

[18] "Verkehrsflusssimulation mit dem nagel-schreckenberg-modell." [Online]. Available: http://duepublico.uni-essen.de/servlets/DerivateServlet/Derivate-190/nagel_schreckenberg_modell_1.htm

[19] S. Lorkowski, P. Mieth, and R.-P. Schäfer, "New its-applications based on floating cat data." [Online]. Available: http://www.ectri.org/YRS05/Presentations/Session-6bis/LORKOWSKI-presentation-YRS2005.pdf

[20] L. 3sat, "ffloating car data"gegen den verkehrsinfarkt in berlin." [Online]. Available: http://www.3sat.de/nano/cstuecke/37788/index.html

[21] R. Anderson, "Social impacts of computing: Codes of professional ethics," *Social Science Computing Review*, vol. 2, pp. 453–469, 1992.

[22] A. S. P. template, "http://www.acm.org/sigs/pubs/proceed/template.html," ACM SIG PROCEEDINGS.

[23] S. Conger. and K. Loch, "Ethics and computer use."

[24] W. Mackay, "Ethics, lies and videotape..." in *CHI '95 (Denver CO)*. ACM Press, 1995, pp. 138–145.

[25] M. Schwartz and T. F. on Bias-Free Language, "Guidelines for bias-free writing," 1995.

Trusted Computing

Lukas Rupprecht
Betreuer: Holger Kinkelin
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: rupprech@in.tum.de

***Kurzfassung*— 1999 wurde die Trusted Computing Platform Alliance (heute: Trusted Computing Group) gegründet, ein Zusammenschluss großer Unternehmen, deren Ziel es ist, den Ansatz des Trusted Computing voranzutreiben und im Anwenderbereich zu etablieren. Trusted Computing ist ein Sicherheitsmechanismus, der durch eine im System verankerte Vertrauenswurzel, dem Trusted Platform Module, erreichen soll, dass das System eindeutig identifizierbar und attestierbar wird. Das bedeutet, dass es keine Möglichkeiten mehr geben soll, die Identität einer Plattform zu fälschen und deren aktuellen Systemzustand (laufende und installierte Software, Konfigurationen) vertrauenswürdig festzustellen. Durch die in den Spezifikationen der Trusted Computing Group vorgeschlagenen Mechanismen zur Umsetzung dieser Eigenschaften eröffnen sich jedoch für die Hersteller solcher Module und der zugehörigen Software eine Reihe an Kontrollmöglichkeiten über die Plattformen, was zu großer Kritik an dem Gesamtkonzept geführt hat. Diese Arbeit ist ein Einstiegspunkt in die Thematik und erklärt die grundlegenden Mechanismen und Eigenschaften einer Trusted Platform sowie den Aufbau und die Funktionsweise der notwendigen Komponenten. Auch die Pro-Kontra-Frage wird beidseitig beleuchtet.**
***Schlüsselworte*— Trusted Computing, Trusted Computing Group, Trusted Platform Module, Trusted Software Stack**

I. Einleitung

In der heutigen Zeit, in der fast jeder Rechner, vom Heim PC bis zum Handy, in irgend einer Art und Weise mit dem Internet verbunden ist, wurde Computersicherheit zu einem Thema, welches für jeden Anwender, sei es Großunternehmer oder Privatmann, eine große Rolle spielt. Immer neue Techniken und Lücken werden entdeckt, die es ermöglichen, in ein System einzudringen und dort Schaden zu verursachen. Viren, Würmer und Hacker kennt fast jeder, aber auch Begriffe wie Social Engineering gewinnen immer mehr an Bedeutung. Zum Schutz vor all diesen Gefahren wurden Systeme entwickelt, welche von einem kleinen Freeware Virenscanner bis hin zum komplexen Firewallsystem aus Soft- und Hardwarekomponenten reichen und versuchen, die eigenen Daten zu sichern. Ein Ansatz, um dieses Ziel zu erreichen, ist das *Trusted Computing*. Nach der *Trusted Computing Group* (TCG) bedeutet Trust hier: „Trust is the expectation that a device will behave in a particular manner for a specific purpose." ([12], S. 15), also die Erwartung oder das Vertrauen, dass die Plattform für einen bestimmten Zweck ein bestimmtes Verhalten aufweist. Das ist allerdings nicht die einzige Definition. Seit 20 Jahren wird dieser Begriff nun schon verwendet und je nach Auslegung anders definiert. So definiert die NSA Trust beispielsweise als: „A Trusted System or component is one whose failure can break the security" ([12], S. 14), also eher im Sinne von „für die Sicherheit verwantwortlich". Das Ziel des Trusted Computing (nach TCG) ist es, Rechensysteme so vertrauenswürdig zu machen, dass es unmöglich wird, deren Identität zu fälschen um sich somit gegenüber Kommunikationspartnern eindeutig zu identifizieren. Ein Mechanismus zum sicheren, lokalen Speichern von sensiblen Daten und die Möglichkeit, Plattforminformationen über Softwarekomponenten zu erhalten sind ebenfalls Teil dieser Initiative. So ein Eingriff in die Architektur kann natürlich auch Einschränkungen für den Benutzer mit sich bringen und bietet die Möglichkeit des Missbrauchs, da die Kontrollmöglichkeiten über die Plattform steigen. Dies führte zu einer beträchtlichen Bewegung gegen das Trusted Computing.
Diese Arbeit erläutert die Grundzüge des Trusted Computing, welche durch die Trusted Computing Group spezifiziert werden und geht dabei vor allem auf die dort verwendeten Techniken ein. Kapitel II stellt die Trusted Computing Group und deren Organisation vor. In Kapitel III werden die zum Verständnis notwendigen Grundlagen bereitgestellt. Kapitel IV beschäftigt sich mit dem Trusted Platform Module und Kapitel V führt den Trusted Software Stack ein. Kapitel VI beschreibt die Remote Attestation als Beispielanwendunge einer Trusted Platform und Kapitel VII beleuchtet die Argumente der Gegner und der Befürworter. In Kapitel VIII wird ein kleiner Ausblick auf zukünftige Entwicklungen sowie ein Fazit gegeben.

II. Die Trusted Computing Group

Trusted Computing ist ein Begriff, der schon seit längerer Zeit existiert und, wie so viele andere Innovationen im Kommunikations- und Securitybereich, durch das Militär entstanden ist. Hört man heutzutage diesen Begriff verbindet man mit ihm allerdings eher die Trusted Computing Group[1] und deren Arbeit. Die Trusted Computing Group ist ein Zusammenschluss von Unternehmen, welche den Ansatz des Trusted Computing im zivilen Bereich und für den Privatanwender weiterentwickeln und standardisieren will. Sie entstand 2003 aus der 1999 gegründeten *Trusted Computing Platform Alliance*, die ein Zusammenschluss der Firmen Microsoft, IBM,

[1] http://www.trustedcomputinggroup.org

Hewlett Packard und Compaq war. [6] Heute gehören ihr ca. 170 Unternehmen weltweit an, darunter z.B. Intel, AMD, Infineon, Motorola oder Nokia. Die TCG bezeichnet sich selbst als eine „not-for-profit organization" mit dem Ziel, Standards zu entwickeln und zu veröffentlichen, um Trusted Computing zu einem wesentlichen Bestandteil moderner Rechensysteme zu machen. Sie stellt Spezifikationen für die Hauptbestandteile eines *Trusted Computing System* (TCS) bereit, erarbeitet grundlegende Konzepte, welche frei einsehbar und somit auch frei umsetzbar sind und fördert dadurch deren Einsatz und deren Verwendung. Der TCG angehörige Unternehmen haben es sich zum Ziel gemacht, ihre entworfenen Standards auch konkret zu implementieren und so gab und gibt es schon einige Systeme, die die geforderten Funktionalitäten unterstützen. Das von Microsoft entwickelte Palladium bzw. NGSCB ist ein Beispiel für die Umsetzung eines *Trusted Operating System* und Firmen wie Lenovo oder auch Infineon liefern bereits die für Trusted Platfformen notwendige Hardware (das sog. *Trusted Platform Module* (TPM)). Trusted Computing ist jedoch nicht allein die TCG und deren Standards sind keine Dogmen, die bei der Entwicklung eines Trusted Computing System erfüllt werden müssen. Allerdings sind deren Spezifikationen vorreitend und maßgebend für die Umsetzung solcher Systeme und deswegen wird in dieser Arbeit auch auf Trusted Computing in der Form der TCG eingegangen. Im Folgenden wird betrachtet, was konkret ein Trusted Computing System ist und welche Eigenschaften und Anforderungen es vorweisen und erfüllen muss.

III. Grundlagen und Konzepte

Ein TCS basiert auf vielen grundlegenden Techniken und Ansätzen aus der Kryptologie, um ein System vertrauenswürdig machen zu können. Die Grundlagen, die in einem TCS umgesetzt werden und die zum Verständnis notwendig sind, werden nun kurz erläutert.

A. Public Key Infrastrukturen

Ein Konzept, welches in einem TCS eingesetzt wird, sind Public Key Infrastrukturen oder PKI's. Diese werden heutzutage häufig eingesetzt, wenn es um verschlüsselte, sichere Nachrichtenübertragung geht. Eine PKI setzt ein Public Key Kryptosystem voraus. Ein solches System verwendet einen privaten Schlüssel zum Entschlüsseln von Nachrichten und einen zugehörigen öffentlichen Schlüssel zum Verschlüsseln derselben. RSA beispielsweise ist ein populäres Verfahren dieser Gattung. Kurz zusammengefasst funktionieren PKI's folgendermaßen:
Es existiert ein privater Schlüssel (Private Key), welcher mit Hilfe eines Algorithmus (z.B. RSA) erzeugt wurde. Aus diesem lässt sich nun ein öffentlicher Schlüssel berechnen. Dieser Public Key ist frei verfügbar und jeder, welcher mit dem Besitzer des privaten Schlüssels sicher kommunizieren möchte, kann den Public Key verwenden um Nachrichten zu verschlüsseln.
Das Wichtigste an solche Verfahren ist, dass es bei ausreichenden Schlüssellängen nicht möglich ist, aus dem öffentlichen Schlüssel den privaten Schlüssel zu ermitteln und dass chiffrierte Nachrichten nur mit dem privaten Schlüssel entschlüsselt werden können.

B. Vertrauenskette und Vertrauenswurzel

Ein weiterer Ansatz ist die Vertrauenskette (Chain of Trust) mit der Vertrauenswurzel (Root of Trust) als Ausgangspunkt. Die Idee dahinter lässt sich am besten anhand der oben beschriebenen PKI's erläutern: Ein Problem, welches sich bei PKI's ergibt, ist die Bereitstellung des öffentlichen Schlüssels. Für einen Nutzer dieses Schlüssels muss gewährleistet werden, dass der Schlüssel, den er verwenden will, wirklich auch der ist, welcher von dem gewünschten Kommunikationspartner bereitgestellt wurde. Um dies sicherzustellen und die Integrität und Gültigkeit der Informationen zu validieren, werden *Zertifikate* eingesetzt. [3] Ein solches Zertifikat verpackt die Informationen über den Besitzer und dessen öffentlichen Schlüssel und stellt diese, wiederum verschlüsselt, zur Verfügung. Um nun an die enthaltenen Daten zu gelangen, wird wieder ein Schlüssel benötigt. Dieser wird von *Certification Authorities* (CA's), welche die Zertifikate erstellen, bereitgestellt. Zwischen Sender und Empfänger können nun beliebig viele Zertifikate existieren, die die Authentizität des darunter liegenden sicherstellen. Der Sender, der ja den Public Key benötigt, muss sich durch diese Hierarchie hangeln, bis eine CA erreicht ist, der er vertraut. So entsteht also eine Kette aus Zertifikaten. Das oberste Glied in dieser Kette nennt man Root CA. Hieran erkennt man nun sehr gut, nicht nur wegen der begrifflichen Parallelen, dass Konzept der Vertrauenskette. Ein Zertifikat wird verwendet, um die Vertrauenswürdigkeit eines darunter liegenden Zertifikates zu gewährleisten. Die Root CA, welche die Vertrauenswurzel bildet, hat keine darüber liegende Instanz mehr, welche deren Vertrauenswürdigkeit bestätigt, und somit muss man dieser von sich aus vertrauen. Beim Trusted Computing geht es nun darum, nicht die Vertrauenswürdigkeit eines Empfängers, sondern die einer Plattform zu gewährleisten. Hierzu muss also eine Vertrauenswurzel im System verankert werden, von welcher ausgehend sich das restliche System überprüfen lässt.

C. Plattform Attestation und Authentication

Der dritte Punkt beschäftigt sich mit der Bewertung (Attestation) und Identifizierung (Authentication) einer Plattform. [14] Genau genommen sind dies zwei verschiedene Kriterien, da sie jedoch ähnlich umgesetzt werden, sind sie hier in einem Unterpunkt zusammengefasst.
Unter *Attestation* versteht man die Bewertung eines Systems. Bewertet wird die Vertrauenswürdigkeit nach Kriterien wie ausgeführter und installierter Software sowie verschiedenen Konfigurationsdateien. Hierfür muss der Zustand eines Systems festgehalten und protokolliert werden. Diesen Vorgang bezeichnet man als *Integritätsmessung* (Integrity Measurement). Das bedeutet, dass über ausgewählte Systemkomponenten und Programme ein SHA-1 Hashwert berechnet und gespeichert wird. Um eine verlässliche Messung zu liefern, muss diese von der Vertrauenswurzel des Systems ausgehen,

da nur so sichergestellt werden kann, dass weitere Messungen nicht verfälscht wurden.
Authentication bezeichnet den Vorgang der Identitätsbestimmung. Die Plattform muss sich gegenüber einem Dritten authentifizieren um zu beweisen, dass sie die ist, für die sie gehalten wird. Auch dieser Vorgang erfordert eine Vertrauenswurzel, von der ausgehende die Identifizierung stattfinden kann.
Die Vertrauenswurzel soll es unmöglich machen, dass falsche Informationen in der Vertrauenskette weitergereicht werden und so Möglichkeiten bieten könnten, Identitäten zu fälschen oder Systemzustände vorzutäuschen, die so gar nicht vorhanden sind. Ein Ziel des Trusted Computing besteht also darin, ein System eindeutig identifizierbar zu machen. Es soll nicht möglich sein, dass das System sich als etwas ausgibt, was es nicht ist. Im folgenden werden die einzelnen Komponenten einer Trusted Plattform erläutert

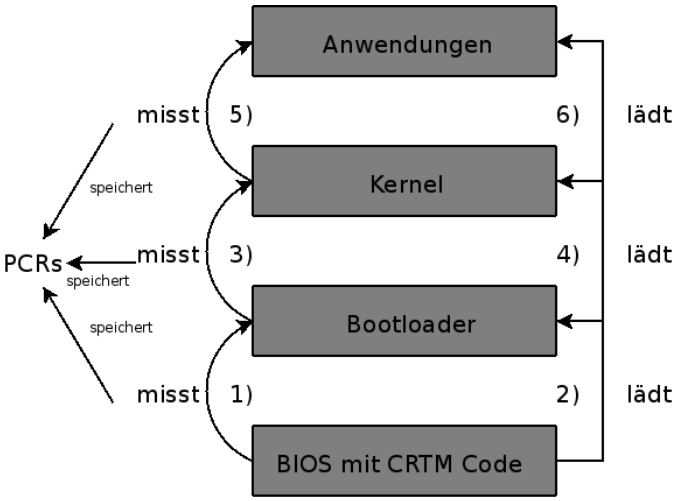

Abbildung 1. Vertrauenskette eines Trusted Bootvorgang

IV. Das Trusted Platform Module

Um die oben erwähnten Konzepte für eine Plattform umzusetzen wird spezielle Hardware verwendet, das *Trusted Platform Module* (TPM). [2] [13] Es ähnelt einer SmartCard, nur dass es nicht an einen Benutzer sondern an ein System gebunden ist. Das TPM ist ein Mikrocontroller und bildet die Vertrauenswurzel des Systems. In ihm werden alle notwendigen Funktionen bereitgestellt, die die Plattform sicher und vertrauenswürdig machen sollen. Im Folgenden werden die einzelnen Bestandteile eines TPM vorgestellt.

A. Die Cryptoengine

Ein Trusted Platform Modul besitzt integrierte Funktionen, um kryptographische Berechnungen auszuführen. Dazu zählen unter anderem ein RSA Schlüssel Generator, ein „echter" Zufallszahlengenerator oder integrierte Berechnungen von Hashfunktionen wie z.B. SHA-1. Durch die Kapselung dieser Funktionen im TPM wird erreicht, dass keine sensiblen Informationen das Modul verlassen müssen.

B. Die Core Root of Trust for Measurement

Die *Core Root of Trust for Measurement* (CRTM) wird zur Integritätsmessung in einem System verwendet. Die daraus resultierenden Messwerte (die Hashwerte) werden im TPM (in den sog. *Platform Configuration Registers* (PCR's)) abgelegt. Die CRTM befindet sich im BIOS und wird als erste Komponente beim Bootvorgang geladen. Die Grundidee ist hierbei, dass ausführbarer Code und Konfigurationsdateien gemessen werden bevor sie geladen werden. Dadurch wird ein Trusted Boot Vorgang realisiert (s. Abb. 1). Dieser beginnt mit dem Laden des CRTM und der Messung der ersten Komponente (normalerweise dem BIOS). Von diesem Punkt aufsteigend wird eine Vertrauenskette gebildet, bei der die einzelnen Softwarekomponenten von ihren darunter liegenden Komponenten gemessen und dann ausgeführt werden. Ein Trusted Bootloader (z.B. Trusted Grub[2]) beginnt damit, den Code, der zum Starten des Betriebssystems notwendig ist, Stück für Stück zu laden. Vor dem Laden eines Teils wird dieser vermessen, danach geladen und ausgeführt, dann der nächste Teil gemessen usw. bis der komplette Kernel geladen und das Betriebssystem ausführbar ist. Die Werte werden im TPM protokolliert. Da die Messungen bei der CRTM starten und diese sich im TPM befindet, also eine vertrauenswürdige Wurzel darstellt, kann eine lückenlose Vertrauenskette aufgebaut werden, in der keine Manipulationen oder Fälschungen möglich sein sollen.

[2]http://trousers.sourceforge.net/grub.html

C. Der Protected Storage

Das TPM stellt einen geschützten Speicherbereich zur Verfügung, in dem geheime Daten wie Schlüssel, Passwörter und auch die Hashwerte der CRTM abgelegt werden können. Um die Daten verschlüsselt speichern zu können wird eine Schlüssel-Hierarchie benötigt, welche den *Storage Root Key* als Wurzel hat. Diese Konzept wird in Abschnitt IV-D.3 genauer betrachtet.

D. Schlüssel und Zertifikate

Neben der Hard- und Firmware eines TPM existieren einige essentielle Daten, die die Eindeutigkeit des Moduls und somit der Plattform garantieren, und die zur Umsetzung der Anforderungen benötigt werden. Die folgenden Auflistung stellt diese vor. [4]
1) Der Endorsement Key: Der *Endorsement Key* (EK) ist die Vertrauenswurzel des Trusted Platform Moduls. Er ist ein RSA-Schlüssel mit einem privaten und einem öffentlichen Teil und immer nur genau einer Plattform zugeordnet. Er wird bei der Aktivierung des TPM erzeugt und der private Teil verlässt dieses auch nie. Er kann auch nicht auf ein anderes System übertragen werden. Mit Hilfe des EK wird die spezifikationsgerechte Funktionsweise des zugehörigen TPM garantiert. Neuere Spezifikationen (v. 1.2) erlauben zwar die Erstellung anderer EK's, dies führt allerdings zum Vertrauensverlust, da die durch den EK garantierten Eigenschaften nun nicht mehr gegeben sein müssen.

2) Attestation Identity Keys: Attestation Identity Keys (AIK's) sind Schlüsselpaare, die nach Bedarf im TPM erzeugt werden und zur Plattformauthentifizierung und -attestierung benutzt werden. Sie sind notwendig, damit Daten nicht mit dem Endorsement Key signiert werden müssen. Dieses Konzept löst ansatzweise ein durch die Eindeutigkeit des EK bedingtes Privacy Problem. Würden nämlich Dokumente mit diesem signiert, wäre die Signatur und somit das Dokument genau einer Plattform zuzuordnen und die Anonymität des Benutzers ginge damit verloren. Eine Beschreibung der genauen Erzeugung und Funktion der AIK's wird in Abschnitt VI gegeben.

3) Storage Root Key: Der *Storage Root Key* (SRK) wird ähnlich wie der Endorsement Key im TPM angelegt und verlässt dieses niemals. Er ist zuständig für die Verwaltung und den Zugriff auf den geschützten Speicherbereich (s. IV-C) des TPM und stellt Schlüssel zur Ver- und Entschlüsselung dort abgelegter Daten bereit. Der private Teil des SRK ist das oberste Element der TPM Key Hierarchie. Wenn ein neuer Schlüssel zum geschützten Ablegen von Daten benötigt wird, kann dieser im TPM erzeugt werden. Um den Schlüssel selbst zu sichern wird dieser nun mit dem in der Hierarchie über ihm liegenden Schlüssel verschlüsselt. Somit ist sicher gestellt, dass die Daten nur mit Hilfe des TPM wieder entschlüsselt werden können (Eine solche Hierarchie befindet sich in Abb. 2). So verschlüsselte Daten werden entsprechend *Key-Blobs* (Blob = binary large object) oder *Data-Blobs* genannt. Allgemein bezeichnet man diese als *TPM protected objects*.
Eine Besonderheit des TPM ist auch, dass Daten logisch an die Plattform gebunden werden können (*Sealing*). Hierbei kann bei der Erstellung eines TPM protected objects der aktuelle Systemzustand, welcher bei der CRTM Messung festgehalten wurde, mit in die Verschlüsselung einbezogen werden. Folglich können solche Objekte nur wieder entschlüsselt werden, wenn der Systemzustand exakt dem Zustand während der Erstellung entspricht. Diesen Mechanismus bezeichnet man als *Sealed Storage*.

4) Zertifikate: Ein TPM muss zusätzlich zu den oben genannten Schlüsseln noch die drei, in Zertifikatform vorliegenden, Informationen enthalten:

- Endorsement Credential
- Platform Credential
- Conformance Credential

Im Endorsement Credential wird der öffentliche Teil des Endorsement Key bereitgestellt. Es bestätigt die Authentizität der Plattform. Das Platform Credential stellt sicher, dass die erforderlichen Plattformkomponenten (nach Spezifikation) vorhanden und validiert sind und im Conformance Credential wird garantiert, dass das System wie erwartet funktioniert.
Nachdem nun die Hardwarekomponenten eines Trusted Computing System abgehandelt sind, geht es im nächsten Teil darum, wie ein Betriebssystem bzw. Software die bereitgestellten Funktionen nutzen kann.

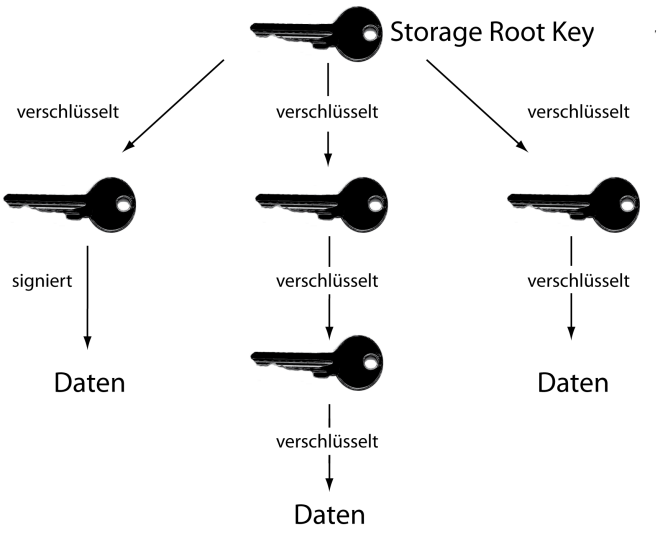

Abbildung 2. TPM Key Hierarchie

V. Softwareunterstützung und Trusted Software Stack

Die bis jetzt vorgestellten Funktionen eines TPM waren alle passiv. Es wurde nur gemessen, protokolliert und Möglichkeiten wie geschützter Speicher zur Verfügung gestellt. Um ein TPM auch aktiv nutzen zu können, benötigt man neben Hard- und Firmware auch noch eine Softwarekomponente. Diese bezeichnet man als den *Trusted Software Stack* (TSS). [20] Über ihn können das Betriebssystem und laufende Anwendungen mit dem TPM kommunizieren und dessen Dienste in Anspruch nehmen. Der TSS besteht aus mehreren Komponenten:

- Die unterste Komponente ist der TPM Treiber. Anwendungen können nur über diesen mit dem TPM kommunizieren und es darf keine Möglichkeit geben, ihn zu umgehen. Er bildet somit die einzige Schnittstelle zum TPM.
- Auf den Treiber aufsetzend folgt die TDDL (TCG Device Driver Library). Diese stellt eine homogene Schnittstelle für alle TPM's zur Verfügung und bildet den Übergang zwischen User und Kernel Mode.
- Die TSS Core Services (TCS) machen Anwendungen alle Grundfunktionen des TPM zugänglich. Sie kommunizieren mit dem TPM über das TDDLI (TCG Device Driver Library Interface)
- Die TSS Service Providers (TSP) sind für die Nutzung der kompletten Möglichkeiten eines TPM zuständig. Die Kommunikation mit dem TPM erfolgt über das Trusted Software Stack Core Services Interface. Die Service Providers stellen für die eigentlichen Anwendungen das TSPI (TSS Service Providers Interface) bereit, über das diese dann auf das TPM zugreifen können.

In Abb. 3 ist der Aufbau noch einmal grafisch dargestellt. Es existieren bis jetzt schon einige Implementierungen eines TSS. Eine kommerzielle Implementierung wird z.B. von

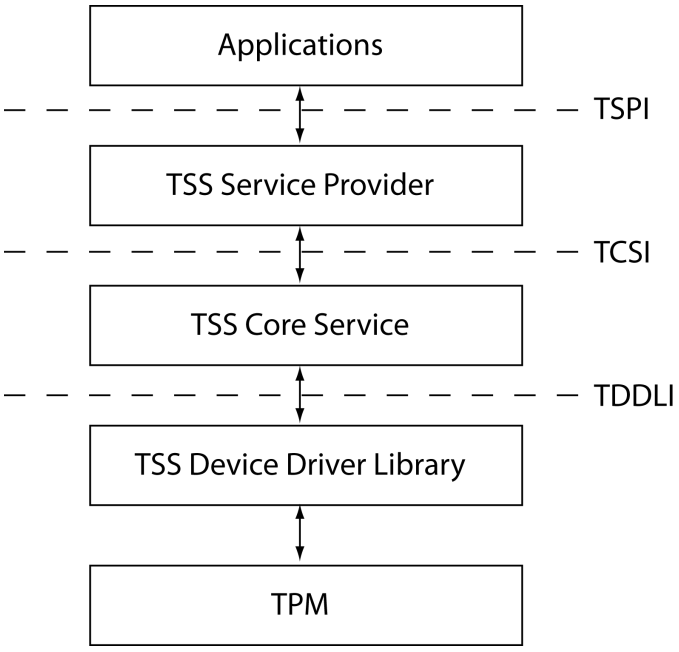

Abbildung 3. Aufbau des TSS

STMicroelectronics angeboten[3]. TrouSerS[4] oder das an der TU Graz entwickelte Trusted Java[5] sind Open Source Varianten eines TSS. Mit Hilfe der Softwareunterstützung kann nun ein gängiges Anwendungsbeispiel von Trusted Computing Systemen betrachtet werden.

VI. Remote Attestation als Beispielanwendung einer Trusted Plattform

In vielen Fällen ist es wichtig zu wissen, wie genau der Kommunikationspartner aussieht. Man möchte z.B. wissen, ob das gegenüberliegende System aktuelle Software mit aktuellen Patches enthält, oder ob noch veraltete, unsichere Versionen verwendet werden. Oder man möchte sichergehen, dass keine kompromittierte Software auf dem System vorhanden ist. Und natürlich möchte man auch sicher sein, dass der Kommunikationspartner keine gefälschte Identität verwendet und jemand ganz anderes ist. Mit herkömmlichen Mitteln ist dies nur bedingt möglich, da man zwar Zertifikate zur Clientauthentifizierung verwenden kann, allerdings über den Zustand des Clientsystems wenig bis gar keine Informationen besitzt. Trusted Computing bietet hierfür einen Lösungsansatz, die *Remote Attestation*. Wie bereits in Abschnitt III-C erläutert, wird hier ein Mechanismus bereitgestellt, der es ermöglicht, den genauen Zustand eines Kommunikationspartners festzustellen und zwar mit der Sicherheit, keine gefälschten Informationen zu erhalten. Die genaue Funktionsweise soll in diesem Abschnitt am Beispiel der von IBM implementierten *IMA*[6] (Integrity Measurement Architecture) unter Linux erklärt werden [15], welche die TPM-internen Messmechanismen auf das laufende Betriebssystem erweitert und somit das System dynamisch vermessen und protokollieren kann. Remote Attestation ist auch ohne Softwareunterstützung möglich, allerdings reichen die bereitgestellten Funktionen des TPM dann nur bis zur Vermessung des Kernels. Im Folgenden wird der Kommunikationspartner, der Informationen eines Systems anfordert als *Verifier* (Prüfer) bezeichnet, das System, welches Informationen über sich liefern soll, wird *Attesting Party* (zu beglaubigender Teilnehmer) genannt.

A. Das Attestation Identity Key Konzept

Das erste Problem welches Auftritt, wenn sensible Daten verschickt werden, ist deren Integrität und deren Authentizität zu gewährleisten. Diese Eigenschaften werden normalerweise mit digitalen Signaturen und Zertifikaten bestätigt. Auch bei der Remote Attestation werden verschickte Messwerte so validiert. Allerdings, wie bereits in IV-D.2 erwähnt, kommt es zu Privacy Problemen, wenn die Signaturen mit dem Endorsement Key erstellt werden und als Authentifikation das Endorsement Credential bereitgestellt wird, da der EK eindeutig ist und somit die Anonymität verletzt werden würde. Um diese Probleme zu lösen wurden die Attestation Identity Keys (AIKs) eingeführt. [2] [4] Mit ihnen können vertrauenswürdige Signaturen erstellt werden und jeder AIK erhält bei seiner Erstellung ein Zertifikat um seine Authentizität zu garantieren. Das Anlegen eines neuen AIK ist in Abb. 4 dargestellt. Hierbei erstellt das TPM zuerst ein neues Schlüsselpaar (den AIK) und schickt den öffentlichen Teil zusammen mit dem Endorsement Credential, dem Platform Credential und dem Conformance Credential an einen vertrauenswürdige Privacy CA. Die Daten werden mit dem privaten Teil signiert, um die Verbindung von privatem und öffentlichem Schlüssel zu gewährleisten. Die Privacy CA validiert nun die erhaltenen Daten und die Signatur. Ist die Prüfung erfolgreich, also sind Endorsement Credential, usw. gültige Zertifikate, erstellt sie für den neu angelegten AIK ein *Identity Credential* und schickt dieses, verschlüsselt mit dem öffentlichen Teil des Endorsement Key, zurück an das TPM. Somit ist der AIK jetzt vertrauenswürdig und kann zum signieren von Daten verwendet werden. Da ein AIK jedoch nicht mehr eindeutig ist, ist er auch nicht mehr genau einer Plattform zuordenbar. Kritiker bemängeln jedoch, dass dadurch nur eine weitere Instanz zwischengeschaltet wurde, dass eigentliche Problem aber bestehen bleibt, da die Privacy CA immer noch eine eindeutige Zuordnung durchführen kann. [1]

B. Die Integrity Measurement Architecture für Linux

Nachdem es jetzt möglich ist, Informationen zwischen Verifier und Attesting Party auszutauschen, ohne die Anonymität der Attesting Party zu verletzen, kann nun die eigentliche Remote Attestation stattfinden. Deren Ziel ist es, die laufende Software der Attesting Party vertrauenswürdig festzustellen, um so eine Aussage über die Sicherheit und die Vertrauenswürdigkeit von dieser machen zu können. Hierzu werden ein TPM und die entsprechende Softwareunterstützung, die IMA, benötigt,

[3] http://www.st.com/stonline/products/literature/bd/10928.htm
[4] http://trousers.sourceforge.net/
[5] http://trustedjava.sourceforge.net/
[6] http://domino.research.ibm.com/comm/research_projects.nsf/pages/ssd_ima.index.html

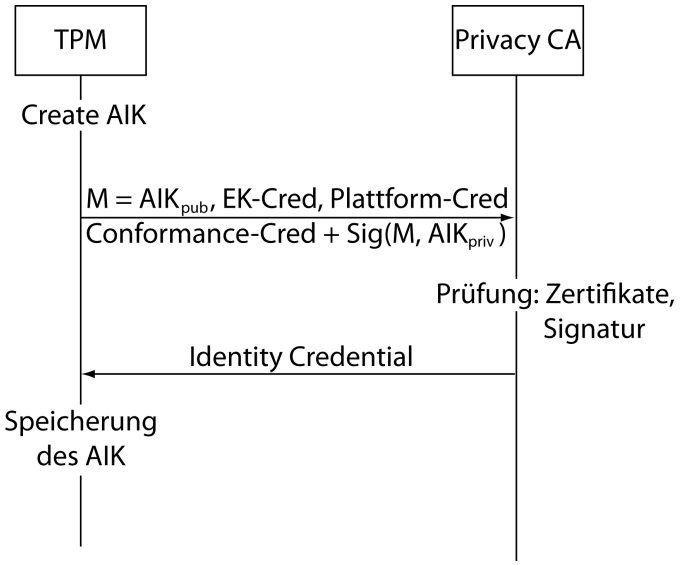

Abbildung 4. Erstellung eines neuen AIK

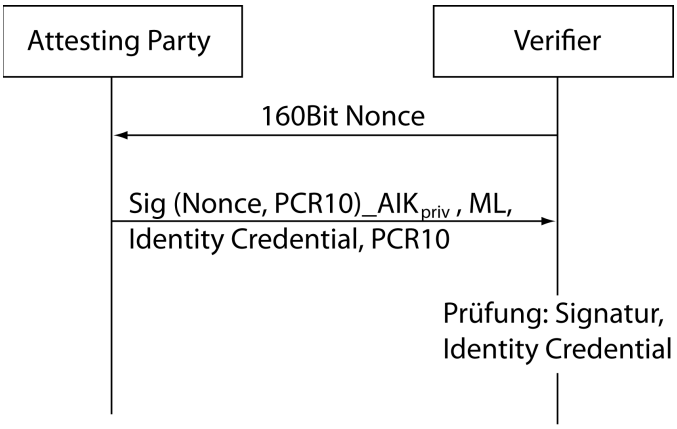

Abbildung 6. Das Integrity Challenge Protocol

um auch die laufenden Anwendungen mit in die Messungen einzubeziehen. [15]
Die IMA besteht aus drei Komponenten:

- Measurement Mechanism (MM)
- Integrity Challenge Mechanism (ICM)
- Integrity Validation Mechanism (IVM)

Die drei Bestandteile und deren Funktionen werden im Folgenden erklärt.
1) Der Measurement Mechanism: Dieser ist zuständig für die Messungen im System. Die Idee ist hierbei, dass Bootvorgang und Kernel, wie in Abschnitt IV-B beschrieben, vermessen werden und der MM die Messungen im Laufenden System übernimmt. Dadurch, dass die IMA teilweise im Kernel vorhanden ist und mit dem Betriebssystem geladen wird, ist diese auch vertrauenswürdig. Der MM bildet nun immer, bevor ein Programm ausgeführt werden soll, einen SHA-1 Wert über dieses (Fingerprint) und speichert das Ergebnis in einer *Measurement Liste* (ML). Es können auch bestimmte sensible Konfigurationsdaten und anderer ausführbarer Code gemessen werden, was eine vollständige Protokollierung ermöglicht. Zusätzlich wird bei jedem Schreibvorgang in die Measurement Liste, das PCR10 *extended*. Das bedeutet, dass der eben gemessene Wert an das Register angehängt wird , davon ein SHA-1 berechnet wird und dieses Ergebnis als neuer Wert in PCR10 geschrieben wird. Dadurch werden die Messwerte vor Fälschung geschützt.

```
var newPCR10 = PCR10.concat(measureValue);
newPCR10 = SHA-1(newPCR10);
PCR10 = newPCR10;
```

Abbildung 5. Die Funktion „extend"

2) Der Integrity Challenge Mechanism: Die oben erzeugten Informationen können nun bei Bedarf durch den Verifier abgefragt werden, mit der Sicherheit, dass diese auch wirklich vertrauensvoll sind. Hierzu existiert der ICM, welcher das Integrity Challenge Protocol implementiert. Dies wird benötigt um die Daten auch sicher auszutauschen und sie z.B. gegen Replay Attacken oder Fälschung während der Übertragung zu immunisieren. Die Funktionsweise ist in Abb. 6 dargestellt. Zuerst wird eine 160 Bit Nonce an die Attesting Party geschickt (normalerweise eine Zufallszahl, da diese nicht vorhersagbar sein darf). Die Attesting Party signiert nun mit einem AIK PCR10 und Nonce und schickt diese Signatur zusammen mit Measurement Liste, zugehörigem Identity Credential und PCR10 zurück an den Verifier. Dieser Vorgang wird als *quote* bezeichnet. Über das zugehörige Identity Credential wird der AIK validiert und anschließend die Signatur geprüft. Zusätzlich wird aus der Measurement Liste der PCR10 Wert wie in Abb. 5 berechnet und mit dem erhaltenen PCR10 Wert verglichen. Stimmen diese und die Werte der Nonce überein, war die Prüfung erfolgreich und die Measurement Listen sind vertrauenswürdig und können zur Bewertung der Attesting Party herangezogen werden. Der Datentransfer sollte natürlich über eine sichere Verbindung, z.B. SSL, erfolgen.
3) Der Integrity Validation Mechanism: Nachdem nun die Informationen vertrauenswürdig und unverfälscht beim Verifier angekommen sind, müssen sie geprüft werden, um den eigentlichen Zustand der Attesting Party festzustellen. Hierzu existiert auf dem Verifier System eine Datenbank mit verschiedenen Fingerprints, die mit den erhaltenen Measurement Listeneinträgen verglichen werden. Es existiert eine Policy, die den Umgang mit unbekannten oder nicht vertrauenswürdigen Fingerprints regelt. Wird eine Übereinstimmung mit einem nicht vertrauenswürdigen Eintrag festgestellt, wird die Attesting Party meist als nicht vertrauenswürdig eingestuft. Die Datenbank kann und muss immer wieder aktualisiert werden, um beispielsweise neue Software mit aufzunehmen oder die Fingerprints gepachter Versionen zu aktualisieren. Es besteht auch die Möglichkeit, Fingerprints, die von vertrauenswürdigen Dritten als nicht schädlich eingestuft und entsprechend zertifiziert wurden, mit in die Datenbank zu integrieren.

Remote Attestation ist nur eine Möglichkeit, die Funktionen einer Trusted Plattform zu nutzen. Es gibt noch viele weiter Anwendungen für die Trusted Computing eingesetzt werden könnte. Allerdings kommt unweigerlich die Frage auf, ob durch dieses Konzept nicht ein zu hohes Maß an Kontrolle ermöglicht wird, welches Software- oder Dienstanbieter missbrauchen könnten, um ihre Kunden zu überwachen. Einen kleinen Einblick in diese Frage und in die Argumente der Gegner und der Befürworter, soll das nächste Kapitel geben.

VII. Pro und Kontra Trusted Computing

„A trusted computer is a computer that can break my security" [16], so endet Ross Andersons FAQ über Trusted Computing. Mit seinem Paper, in dem er als erster Kritik an Trusted Computing übt, hat er eine große Widerstandswelle ausgelöst. Er weckte die Befürchtung, dass Trusted Computing dazu entwickelt wurde, um die Kontrolle über Trusted Plattformen zu gewinnen. Seiner Meinung nach sei die Hauptmotivation bei der Entwicklung der TCG-Spezifikationen das Digital Rights Management gewesen und das Ziel, Softwarepiraterie zu bekämpfen. Über eine Trusted Plattform ist es möglich, festzustellen, ob eine gültige Lizenz für eine Software oder für Dateien (z.B. Musik oder Filme) vorliegt. Ist dies nicht der Fall, kann die Ausführung dieser Inhalte unterbunden oder diese sogar gelöscht werden. Das sog. *Traitor Tracing* soll Raubkopien über Wasserzeichen erkennen und entfernen und das dafür verantwortliche System auf eine Blacklist setzen. Ross betont auch die Schwierigkeiten, die bei Trusted Computing Systemen entstehen können, wenn man alternative Software nutzen möchte. Um umzusteigen und alte Dateien mit der neuen Software bearbeiten und nutzen zu können ist immer eine Zustimmung der ursprünglichen Dateibesitzer notwendig, was den Wechselaufwand stark erhöht. So, laut Ross, kann z.B. Microsoft seine Marktstellung noch mehr stärken und Preise dirigieren. Seine größte Sorge sind die Zensurmöglichkeiten, die Trusted Computing mit sich bringt. Mit den oben erwähnten Blacklists könne nicht nur Piratensoftware gebannt, sondern auch politisches Material kontrolliert und verboten werden, was eine erhebliche Einschränkung der Freiheit wäre. Andere, wie Arbaugh [1], versuchen, Trusted Computing im Gesamtzusammenhang zu betrachten und auch die positiven Aspekte und Möglichkeiten für die Computersicherheit zu berücksichtigen. Arbaugh beispielsweise sieht das Problem eher im Privacy Bereich, da eine Trusted Plattform eindeutig ist und sie somit immer genau dem Besitzer zugeordnet werden kann. Auch dass Konzept der Attestation Identity Keys ist (wie bereits in Abbschnitt VI-A erwähnt) in in seinen Augen noch keine Lösung, da trotz allem die Zertifizierungsstelle, die für die Zertifizierung der AIK's zuständig ist, die einzelnen Schlüssel immer noch der Plattform und somit dem Besitzer zuordnen kann, da sie Informationen wie das EK Credential erhält.

Auf die viele Kritik antworteten die Entwickler, die an Trusted Computing beteiligt waren mit einem Rebuttal [17], in dem sie Ross's Argumente und die anderer Kritiker widerlegten und ihre Technologie verteidigten. Sie argumentierten, dass Begriffe wie Trusted Computing und DRM synonym verwendet worden sind, jedoch eine klare Trennung zwischen diesen Technologien besteht und diese auch sonst keine direkte Verbindung haben. Außerdem wurden Spekulationen über Trusted Computing gemacht, welche als Tatsachen dargestellt wurden, in Wirklichkeit aber so gar nicht in den Spezifikationen vorhanden sind. Sie kritisierten auch, dass die Papers voll von Fehlern in Bezug auf das technische Verständnis der Spezifikationen seien und dadurch viele Missverständnisse und falsche Annahmen aufkamen. Trusted Computing hat als primäres Ziel, dem Benutzer sicheres Schlüssel- und Datenmanagement zur Verfügung zu stellen und will in keinster Weise Einfluss auf seine Rechte zu nehmen. Das, was Systeme wie DRM oder das von Microsoft entwickelte NGSCB (Next-Generation Secure Computing Base) daraus machen, hat nichts mit dem Konzept des Trusted Computing an sich zu tun.

So entstand ein Hin- und Her zwischen Gegnern und Befürwortern. Die Quellen [1], [16], [17] und [18] bieten einen guten Anfang, um tiefer in die Diskussion einzusteigen.

VIII. Ausblick und Fazit

Nach einer Statistik von IDC[7] sind bis heute schon über 50 Millionen Systeme mit der notwendigen Technologie ausgestattet um als Trusted Plattform agieren zu können. Jedoch sind die TPMs standardmäßig abgeschalet und müssen vom Benutzer der Plattform erst explizit aktiviert werden um verwendet werden zu können. Ob dies jedoch auch getan wird ist eine andere Sache da viele Nutzer noch nicht überzeugt davon sind (s. Abschnitt VII) und auch erst wenige Anwendungen (neben der Remote Attestation) für Trusted Plattformen existieren. Einen großen Schritt zur Nutzung hat Microsoft mit dem *BitLocker* Konzept getan. [7] Dieses ist in das Windows Vista Betriebssystem integriert und bietet sichere Festplattenverschlüsselung mit Hife eines TPM an.

Trotz der hardwareunterstützten Sicherheitsmechanismen von Trusted Computing gibt es immer noch Möglichkeiten, diese zu umgehen. In [11] wird erfolgreich demonstriert, dass durchaus auch die Hardwarekomponenten eines TPM Schwachstellen haben können und somit angreifbar sind. Um sich im (v.a. Privat-)Anwenderbereich etablieren zu können, müssen Entwicklungen noch viel gegen die möglichen Verletzungen der Privatsphäre und die damit verbundenen Ängste, die mit Trusted Computing zusammenhängen, tun. Ein Ansatz ist z.B. die Virtualisierung, die heutzutage eine sehr große Rolle spielt. Ein rein virtuelles TPM wird in [19] vorgestellt. Virtuelle Maschinen besitzen dadurch die Möglichkeit, Trusted Computing in Anspruch zu nehmen. Vielleicht kann auf diese Weise ein Nutzer wieder die komplette Kontrolle über seine Plattform erlangen und trotzdem die Vorteile einer Trusted Plattform nutzen.

Vor allem aufgrund der hitzigen Diskussion um Trusted Computing ist ersichtlich, dass noch viele Fragen und Probleme

[7] http://www.itseccity.de/?url=/content/dailynews/090305_dailynews_text.html zuletzt besucht am 05.03.2009

offen sind. Wie bei so vielen Innovationen kommt es auch beim Trusted Computing darauf an, dass es richtig verwendet wird, dann kann es durchaus große Fortschritte im Bereich Sicherheit mit sich bringen. Der Einsatz in Unternehmensinfrastrukturen ist sicherlich sinnvoll, weil so eine sehr gute und zuverlässige Prüfung der einzelnen Rechner im und außerhalb des Netzes ermöglicht wird; dass diese immer aktuell sind und keine Schadsoftware darauf vorhanden ist. Sobald aber dadurch der Benutzer überwacht wird und somit eine Einschränkung seiner Freiheit erfährt, werden die positiven Aspekte schnell zu Negativen. Deshalb muss auf die richtige Verwendung der verfügbaren Technologien geachtet werden.

LITERATUR

[1] Willian A. Arbaugh. The tcpa; what's wrong; what's right and what to do about it. Technical report, University of Maryland, 2002.

[2] Sundeep Bajikar. Trusted platform module (tpm) based security on notebook pcs - white paper. Technical report, Mobile Platforms Group, Intel Corporation, 2002.

[3] Nicholas Bohm Brian Gladman, Carl Ellison. Digital signatures, certificates and electronic commerce, 1999.

[4] Claudia Eckert. *IT-Sicherheit, Konzepte-Verfahren-Protokolle*. Oldenbourg, 5. edition, 2007.

[5] Trusted Computing Group. Backgrounder, more secure computing, 2006. http://www.trustedcomputinggroup.org.

[6] http://de.wikipedia.org. Trustd computing platform alliance. zuletzt besucht am 22.04.2009.

[7] Jan Trukenmüller Jan-Peter Stotz Sven Türpe Jan Steffan, Andreas Poller. *BitLocker Drive Encryption im mobilen und stationären Unternehmenseinsatz*. Fraunhofer-Institut für Sichere Informationstechnologie.

[8] William J. Caelli Jason F. Reid. Drm, trusted computing and operating system architecture. Technical report, Information Security Research Center, Queensland University of Technology, 2005.

[9] Ed Dawson Eiji Okamoto Jason Reid, Juan M. Gonzalez Nieto. Privacy and trusted computing. Technical report, Information Security Research Center, Queensland University of Technology.

[10] Steve Johnson. Trusted boot loader. Technical report, Chair Security WG, Panasonic, 2006.

[11] Bernhard Kauer. Oslo: Improving the security of trusted computing. Technical report, Technische Universität Dresden.

[12] Thomas Müller. *Trusted Computing Systeme*. Springer Verlag Berlin Heidelberg, 2008.

[13] Siani Pearson. Trusted computing platforms, the next security solution. Technical report, HP Laboratories Bristol, 2002.

[14] James P. Ward Reiner Sailer, Leendert Van Doorn. The role of tpm in enterprise security. Technical report, Thomas J. Watson Research Center, 2004.

[15] Trent Jaeger Leendert van Doorn Reiner Sailer, Xiaolan Zhang. Design and implementation of a tcg-based integrity measurement architecture. Technical report, IBM T.J. Watson Research Center, 2004.

[16] Anderson Ross. 'trusted computing' frequently asked questions, 2003. http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html zuletzt besucht am 05.03.2009.

[17] David Safford. Clarifying missinformation on tcpa. Technical report, IBM Research, 2002.

[18] Seth Schoen. Trusted computing: Promise and risk.

[19] Kenneth A. Goldman Ronald Perez Reiner Sailer Leendert van Doorn Stefan Berger, Ramón Cáceres. vtpm: Virtualizing the trusted platform module. Technical report, IBM T.J. Watson Research Center, 2006.

[20] Trusted Computing Group. *TCG Software Stack (TSS)*, 2007. Version 1.2.

Denial of Service

Carl Denis
Betreuer: Marc Fouquet
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: denis@in.tum.de

***Kurzfassung*— Thema dieser Arbeit ist die Analyse verschiedener Denial of Service (DoS) Techniken, den Motiven, ihrer Ausführung und die Häufigkeit mit der sie Auftreten. Es wird ein Überblick gegeben, der einen Einstieg in die Analyse vereinfachen soll, um mögliche Prognosen über die zukünftige Entwicklung dieser Angriffsart aufzustellen.**

***Schlüsselworte*— Denial of Service, ICMP/TCP/SYN-Flood, Cyberattack**

I. Einleitung

Wohlbekannt sind Denial of Service (DoS) Angriffe aus den Medien. Immer wieder werden sie für Schlagzeilen bezüglich des "Cyberkrieges" verwendet und breiten bei Laien, die trotzdem täglich mit Computern zu tun haben, eine gewisse Panik aus. Schutzlos ist man der Willkür von Hackern ausgesetzt. Was nun passiert, wie und warum Einzelne, vielleicht auch Jugendliche in der Schule ganze Regierungsnetze lahmlegen können, und wie häufig es wirklich geschieht, werden wir im Folgenden abhandeln.

A. Definition

Denial of Service bedeutet wörtlich übersetzt "Dienstverweigerung" und beschreibt eine jegliche Art und Weise, einen Dienst über ein Netzwerk unerreichbar zu machen. Dem inbegriffen sind auch die weniger beachteten physikalischen Angriffe, die ein lokaler Angreifer zum Beispiel durch abzwicken eines Netzwerkkabels erreichen könnte. Untersucht werden in dieser Arbeit jedoch lediglich entfernte Attacken, die einen direkten/lokalen Zugriff auf den Host oder das anvisierte Netzwerk ausschließen.

B. Beispiele

- Im Mai 2007 wurde Estland von DDoS Angriffen heimgesucht, teilweise ist Estland digital vom Rest der Welt abgeschnitten gewesen [1], [2].
- Im März 09 wurde die Videostreamingleitung von ESL auf der CeBit Hannover lahmgelegt.

II. Motivation

Das Motiv, einen Dienst unerreichbar zu machen kann auf sehr verschiedene Gründe zurückzuführen sein. Diese lassen sich aber in verschiedene Kategorien einteilen. Die Motivation, globalpolitisch oder im kleinen, ist proportional zur Wichtigkeit und Resistenz des Ziels. Damit ein Angriff auf ein prominentes Ziel, wie zum Beispiel ein Regierungsserver, überhaupt als Angriff gewertet werden kann, müssen ganz anderen Mittel in Bewegung gesetzt werden als um einen heimischen Webserver außer Gefecht zu setzen.

- Cyberwarfare: Der digitale Krieg ist auch heute nicht nur mehr in Filmen präsent, was sich unter anderem durch die aktiven Überlegungen über Restrukturierung der Sicherheitsvorkehrungen der US-Regierung zeigt [3]. Simultan mit dem Georgienkrieg gestartete Cyberattacken [4], [5], sowie der DDoS [1] auf die Estländische Regierung 2007 bekräftigen, dass diese Methoden mit der immer größeren weltweiten Vernetzungen verschiedener Systeme mit Breitbandanbindungen zu immer durchschlagkräftigeren Waffen mutieren.
- Organisierte Kriminalität: Durch Erpresserbriefe werden Firmen aufgefordert Zahlungen zu tätigen um im Gegenzug ihre Internetpräsenz ungehindert betreiben zu können. Besonders konzentriert tauchten diese nahe für Betreiber wichtige Terminen auf, wie zum Beispiel für Online-Wettbüros zur Fußball Europameisterschaft 2004 [6].
  Botnets[2] scheinen auch immer mehr untergliedert zu werden um evntl. Teile davon zu vermieten [7], [8], demnach ist es auch denkbar, dass für Marketingzwecke Demonstrationen und anschließend für Kunden breit angelegte Angriffe durchgeführt werden [9].
- Die kleine Rache: als neuer Volkssport in der elektronischen "Sportwelt" scheint das DoS aufgetaucht zu sein. Tutorials wie man einen verhassten Gegner aus einem Onlinespiel nimmt, indem man zum Beispiel seine Internetleitung an der die X-Box hängt überlädt sind frei zugänglich [10], [11].

III. DoS Angriffe durchführen

Es wird hier keine Anleitung gegeben um Systeme in die Knie zu zwingen, lediglich ein Überblick über Methoden gegeben die anderweitig schon frei verfügbar und ausführlich dokumentiert sind.

DoS Angriffe kann man prinzipiell in 3 Unterklassifizierungen einordnen, die Einfachen, welche ein einzelner Computer

[1] Distributed Denial of Service, siehe III-C.

[2] Netzwerk von kompromitierten "Zombiekomputern" welche für einen Kriminellen Zweck missbraucht werden.

zur Ausführung ausreicht, diese die andere Netze als ungewollte Reflektoren benutzen und schlussendlich Distributed-DoS.

A. Die elementarsten Techniken

1) Fehlerhafte Implementierung: Bekannt wurde diese Art von Angriffen durch den sogenannten "Ping of Death" der hier [12] beschrieben ist. Es geht darum, ein unzulässiges IP-Paket nach dem RFC-791 [13] zu produzieren, welches die maximale Paketgröße von 65535 Bytes beim Wiederzusammenfügen eines fragmentierten Pakets überschreitet, und beim Client einen Bufferoverflow erzeugt. Damit wird erreicht, dass bei einem anfälligen Betriebssystem zufällige Bits im Speicher überschrieben werden können, was als Folge nichts, ein Einfrieren oder ein Neustart des Systems haben kann.

Der "Ping of Death" ist nur ein Beispiel von verschiedenen "Nuke" [3] Techniken, welche bei fehlerhafter Implementation genutzt werden können. Meistens handelt es sich aber um Speicherprobleme, welche sobald sie erkannt sind, durch einen Patch effektiv bekämpft und dauerhaft abgestellt werden können.

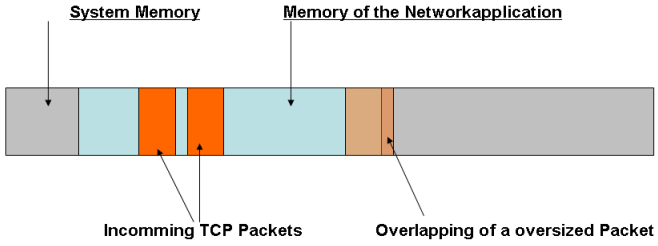

Abbildung 1. Ping of Death

2) Dauerhafter Hardwareschaden: Besonders interessant für einen Angreifer ist es auch dauerhaften Schaden anzurichten, welcher sich nicht nach dem Ende des Angriffs von alleine behebt. Im schlimmsten Fall ist sogar ein Austauschen der Hardware nötig. Permanent Denial of Service (PDoS) ist wohl am besten gegen an ein Netz angeschlosse, eingebettete Systeme, wie zum Beispiel Drucker oder Router, durchzuführen. Durch Sicherheitslücken im Fernwartungssystem, sei es durch Programmierfehler oder duch administrative Versäumnisse wie fehlende Patches oder nicht geänderte Standardpasswörter, kann sich ein Angreifer Zugang zu den Geräten verschaffen und evtl. ein fehlerhaftes Firmwareimage hochladen, welches beim nächsten Neustart dann gebootet wird.
Das Gerät ist dadurch unbrauchbar geworden. Wenn dies nun auf einem Router passiert sind alle dahinterliegende Systeme mit einem Schlag nicht mehr zu erreichen. Dieser Fehler ist einfach nicht mehr zu beheben und mit relativ geringem Aufwand zu bewerkstelligen, da nur ein einmaliger Vorgang nötig ist im Gegensatz zu anderen DoS Methoden, welche durchgehende Aktionen des Angreifers erfordern.
Vorgeführt wurde diese Art von Angriff von Rich Smith von HP Systems Security Labs auf der EUSecWest Sicherheitskonferenz [14], [15].

3) Überflutung des Opfers: Der "flood" ist die wohl meist eingesetzte Art des DoS. Es geht darum möglichst viele Pakete[4] an das Opfer zu schicken und damit zu bezwecken, dass dem Opfer irgendeine Ressource ausgeht, sei es Speicher, Bandbreite oder CPU-Leistung. Wenn das Opfer einmal mit dem illegitimen Verkehr überlastet ist, kann es berechtigte Anfragen nicht mehr bearbeiten.
Um nicht von einer wachsamen Firewall sofort ausgesperrt zu werden verwendet man zusätzlich IP-spoofing (Fälschung) indem man die versandten Pakete mit einer anderen Herkunfts-IP-Adresse versieht und somit bei dem Empfänger vortäuscht, dass das Paket von einer anderen Maschine stammt. Wegen der Struktur des Internets ist es für dem Empfänger nicht möglich die korrekte Herkunft des Pakets zu überprüfen.

- SYN flood ist ein Angriff auf der Netzwerkschicht 4 und nutzt die Statusallokation welche in TCP für jede Verbindung gebraucht wird, um den Arbeitsspeicher langsam aufzubrauchen. Das Opfer wird von SYN Paketen (initiieren den Aufbau einer Kommunikation) überflutet und sendet falls es möglich ist, zum Beispiel bei einem Webserver, ein SYN-ACK Paket und begibt sich in den Status "Wartend" bis entweder wieder ein ACK eintrifft oder ein Timeout ausläuft. Wenn man jetzt das Opfer dazu bringen kann schneller Verbindugen zu öffnen als diese wieder ablaufen, kann man erreichen dass der Arbeitsspeicher nicht mehr ausreicht um neue Verbindungen zu alloziieren und es beginnt eine Diensteverweigerung.
  Um einem solchen Angriff zumindest teilweise entgegenzuwirken, gibt es mehrere Ansätze. Einer davon sind die SYN-Cookies welche als Antwort auf ein SYN an den vermeintlichen Absender geschickt werden. Ist dieser der Reale, empfängt er dieses Cookie und kann es gekoppelt mit einem SYN-Paket erneut an den Server schicken, welcher erst zu diesem Zeitpunkt die Verbindung alloziiert [16].
  Diese Methode wird oft erst bei höherer Last auf einem Server zugeschaltet, um im normalen Verlauf keinen zusätzlichen Roundtrip zum Verbindungsaufbau zu benötigen.

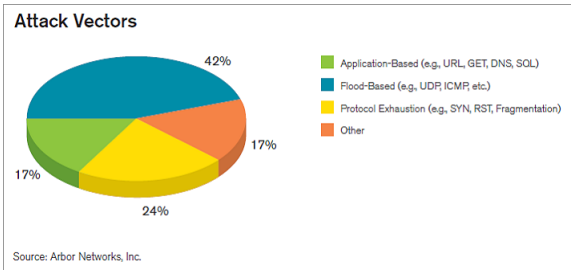

Abbildung 2. Angriffsvektoren

[3]Steht für Denial of Service.

[4]Der Pakettyp ist bei einem breit angelegten flood nicht ausschlaggebend, sei es TCP/UDP oder ICMP

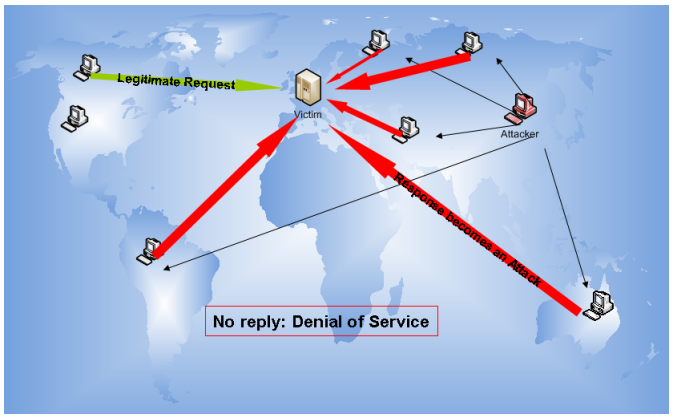

Abbildung 3. Amplifikation durch Reflektion

- CPU flood bezeichnet einen Angriff der darauf aus ist, die Rechenlast eines Knoten soweit zu erhöhen, dass er nichts mehr Sinnvolles leisten kann. Beliebte Ziele sind kryptographische Endgeräte, welche zum nachprüfen von Signaturen und Verschlüsselung erheblichen Rechenaufwand haben und somit ist diese Ressource besonders schnell aufgebraucht. Interessant kann auch je nach Bauart eines Routers dessen Überlastung sein. Durch komplizierte Fragmentierung oder geschickt manipulierte Pakete, welche dann nicht von den in Hardware implementierten Bausteinen bearbeitet werden können, kann Last auf dem begrenzten Prozessor eines Routers erzeugt werden. Zusätzlich kann bei vielen auch ein Cacheüberlauf hervorgerufen werden, da die Netzwerkinterfaces auf einen größeren Datendurchsatz ausgelegt sind, kann der Cache dieser CPU nicht ausreichen. Egal welcher Fall eintritt, der Router ist außer Gefecht gesetzt und wie vorher schon erwähnt, die hinter ihm liegenden Systeme ebenfalls.
- Clients welche nur eine sehr magere Anbindung haben, wie zum Beispiel Einwahlleitungen , kann man schon mit einem beliebigen flood mit “irgendeinem” Paket vom Netz abtrennen, weil die Leitung einfach überlastet (vorrausgesetzt man verfügt selber über genügend Bandbreite) wird und legitimer Traffic das Endgerät nicht mehr erreicht. In der Praxis ist diese Methode ohne Angriffs-Amplifikation wohl nur schwer anwendbar, weil Breitbandverbindungen immer verbreiteter werden, welche weniger anfällig sind.

B. Angriffs-Amplifikation für größere Ziele

Wenn die Leitung des Opfers nun aber größer ausgelegt ist als die Eigene, ist es natürlich wesentlich schwieriger einen effektiven Angriff durchzuführen. In diesem Fall ist es besonders nützlich, wenn man in den Weiten des Internets andere Geräte dazu überreden kann, an dem Angriff teilzunehmen.

1) Reflektion - Smurf Attack: Bei dieser Angriffsart werden wenn möglich, ein oder mehrere Subnetze dazu verwendet als Spiegel zu fungieren. Man sendet über eine Broadcastadresse[5] ein Paket an ganze Netze, welche der Amplifikation dienen, und fälscht dabei die Absenderadresse, welche nunmehr die des Opfers sein soll. In diesem Fall werden alle erreichbaren Clients in diesem Netz, welche den genutzten Dienst verwenden, eine Antwort an das Opfer schicken. Beim Schlumpf-Angriff (SmurfAttack) wird ein ICMP echo request (ping) über Broadcast an ein Netz versandt. Nebeneffekt ist die Anonymisierung des Angreifers, weil das Opfer nur die Adressen der Schlümpfe warnehmen kann.
Viele Netze sind heute dagegen imunisiert als Schlumpf für einen Angreifer aus einem externen Netz zu fungieren, da Router am Rande eines Netzes heute Broadcasts von Außen verbieten.

2) DNS-Amplifikation: Hier werden öffentlich zugängliche, rekursive[6] und antwortende DNS Server dazu missbraucht mit ihrer großen Bandbreite die Leitung des Opfers auszulasten. Dies ist möglich weil eine kleine Anfrage von wenigen Bytes eine sehr große Antwort des DNS Servers erzeugen kann. Ist diese Anfrage nun mit der gefälschten Absenderadresse des Opfers versehen, wird dieses die ganzen Antworten erhalten, was erheblich den Datendurchsatz von legitimen Paketen zum Endystem erschwert. Auch hier ist der Angreifer anonymisiert. 21% der Internetprovider haben angegeben, ihre rekursiven DNS-Server nicht vor Clients außerhalb ihres eigenen Netzes abzuschirmen [17].

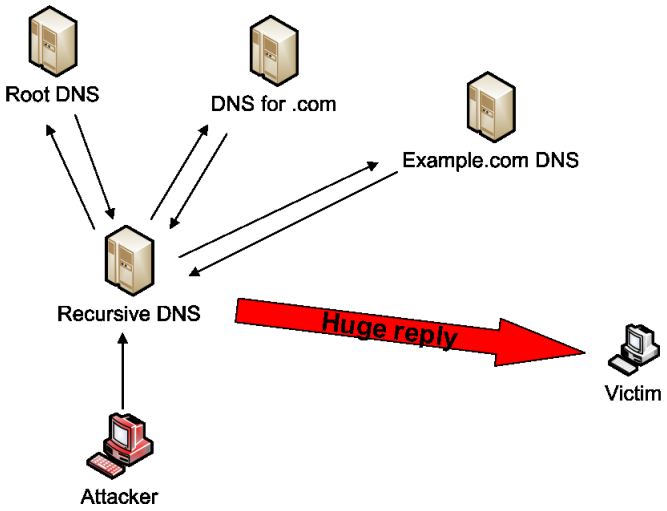

Abbildung 4. Amplifikation eines DoS durch öffentliche rekursive DNS

Wie auch bei Smurf wären durchgehend verbreitete Filter auf Providerebene, welche die Injektion von gefälschten Paketen direkt am Eingang abblocken eine wirksame Maßnahme gegen diese Angriffe [18].

[5]Höchste Adresse in einem Subnetz, oft der Art: x.y.z.255
[6]Rekursive DNS führen die Anfrage selber durch, anstatt den Client nur auf einen anderen Server hinzuweisen

C. Distributed Denial of Service

Verteidigungsmechanismen gegen vorher angesprochene DoS Techniken basieren immer darauf den bösartigen Verkehr vom legitimen zu unterscheiden und diesen frühzeitig (nahe an der Quelle, damit so wenig wie möglich Last entsteht) im Netz rauszufiltern. Besonders kompliziert wird es bei dem sogenannten Distributed Denial of Service, wenn der Angriff nicht mehr nur von einem Angreifer und einer Leitung ausgeht, sondern von einer Vielzahl an Rechnern, die sehr breit durch die ganze Welt verteilt sein können. Die von Botnetzen aufgebauten Zombiearmeen können nämlich von ihrer Art her legitimen Traffic erzeugen. Es liegt eben in der Natur eines Webservers auf Seitenanfragen zu antworten. Wenn das jetzt zehntausende Clients gleichzeitig tun, ist es nicht nachvollziehbar ob es sich dabei um einen Angriff oder einen sogenannten "flash" handelt. Wenn eine kleinere Webseite spontan an Anziehungskraft gewinnt, weil sie zum Beispiel von einem vielgelesen Portal wie Slashdot oder Heise.de verlinkt wurde, kann diese von den anstürmenden Lesern überlastet werden.

Eine in der Forschung in Erwägung gezogene Möglichkeit legitimen Traffic von illegitimen zu unterscheiden, scheint eine Methode zu sein, welche den Client auffordert eine höhere Bandbreite zu benutzen, wobei davon ausgegangen wird, dass ein Angreifer diese sowieso schon ausschöpft und seine Übertragungsrate nicht mehr erhöhen kann. Diese Anfragen würden dann ignoriert [19]. Ein konkretes Umsetzungsbeispiel scheint es noch nicht zu geben.

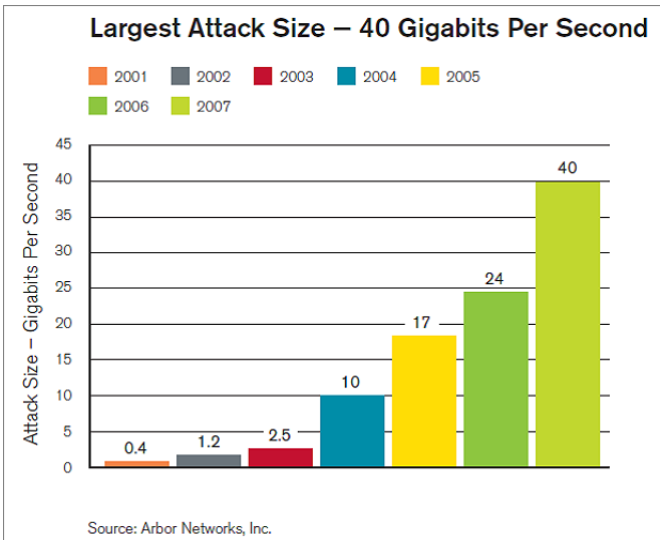

Abbildung 5. Entwicklung der stärksten Angriffe zwischen 2001 und 2007

Wenn nun die Leitung als solche überlastet ist, bringt auch solch ein Ansatz nichts mehr. Das passierte Turtle Entertainment auf der CeBit 2009 in Hannover bei ihrer Videoübertragung vom Messegelände. Wie ein persönliches Telephoninterview mit einem der beteiligten Administratoren vor Ort ergab, konnten sie wohl zuererst eine Verlangsamung ihrer 100Mbit Leitung vom Messegelände feststellen, sowie ein ansteigender Trafficload auf ihrem Monitoring. Kurz darauf brachen die Firewalls zusammen und der Messestand war Offline. Nachdem sie sich mit ihrem Provider verständigt hatten, stellt sich herraus, dass es sich um einen DDoS von ungefähr 60 bis 70 Clients aus China handelte. Diese erzeugten Spitzenlasten von 120 bis 130Mbits Traffic mit einem UPD Flooding auf Port 21 mit über 10.000 Paketen pro Sekunde. Als Gegenmaßnahme half, sich vom Provider einen neuen IP-Block geben zu lassen, was den Angriff anschließend ins Leere laufen ließ. Die Downtime betrug ungefähr 30 Minuten. Eine wirklich effiziente Verteidigung gibt es keine, ein erneuter Angriff auf den neuen IP-Block hätte sofort die selbe Auswirkung gehabt.

IV. Häufigkeit und Intensität von DoS heute

Die Präsenz solcher Angriffe im Internet steht außer Zweifel aber um das Gefahrenpotenzial genauer einschätzen zu können bräuchte man ein Monitoring aller DoS Angriffe die stattfinden. Leider scheint dies aber unmöglich und man muss auf andere Methoden zurückgreifen, um Näherungswerte zu erlangen. Backscatter-Analysis [20] scheint ein Weg zu sein, wenigstens einen Teil der DoS Techniken in ihrer Frequenz und Intensität zu erforschen.

Um ein Opfer effektiver anzugreifen und dem Angreifer bessere Anonymität zu gewährleisten kann die Quell-IP-Adresse in dem für den Angriff verwendeten Paket modifiziert worden sein. Demnach werden die Antworten des Opfers, solange dieses den Dienst nicht vollständig verweigert bei zufällig gewählten (den gespooften[7]) IP-Adressen landen. Wie groß der Anteil der Angriffe ist, welche IP-Spoofing verwenden ist leider nicht so einfach zu bestimmen.

In diesem Experiment [20] wurde auf 1/256 aller Adresses des IPv4 Addressraums nach Backscatterpaketen[8] gelauscht um so eine Idee zu bekommen wieviele Pakete dieser Art im Netz herumschwirren. Davon ausgehend können dann Hochrechnungen gemacht werden. Mögliche Informationen welche man extrahieren kann, sind das Ausmaß des Angriffs, wer ihm zum Opfer fällt (Source-IP des Backscatterpakets) und was für ein Angriffstyp verwendet wird.

Nach [20] kann diese Art von DoS Angriffen mit 2000-3000 pro Woche beziffert werden, mit einer Intensität von über 100.000 Paketen pro Sekunde, was eine immense Durchschlagkraft in sich birgt. Vorwiegend werden diese Angriffe über TCP (zu 95%) durchgeführt; an zweiter Stelle steht ICMP.

Ferner ist es wichtig zu beachten, dass dies nur ein Teil der tatsächlich verübten Angriffe darstellen kann, da diese Methode es leider nicht ermöglicht DoS-Angriffe, welche kein Backscatter erzeugen, zu erfassen.

Ein weiterer Ansatz zur Ermittlung der Gefahren welche überhaupt im Internet kursieren ist eine seit 2005 alljährliche Umfrage von Arbor Networks Inc [17]. Im Jahre 2008 wurden

[7] gefälschten
[8] Antwortpakete die wegen IP-Spoofing bei einer Zieladresse ankommen

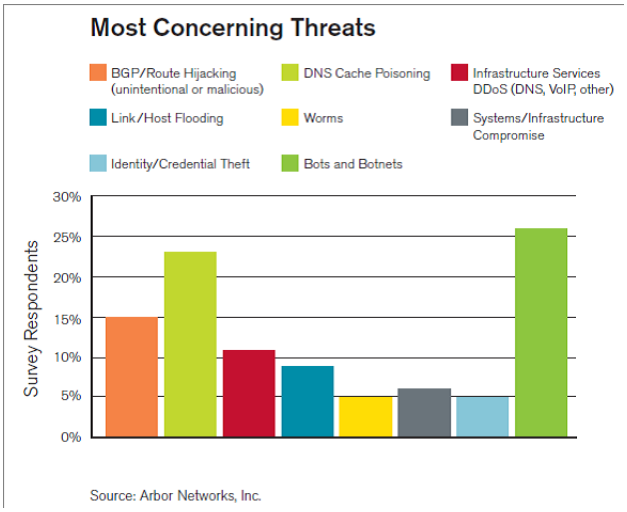

Abbildung 6. Die größten Bedrohungen

66 verschiedene ISP[9] und sonstige bedeutende Internetdienstleister zu verschiedenen Angriffen aus den letzten 12 Monaten und Gefahren aus dem Internet befragt. Wie in Abbildung 4 sichtbar wird, können neben den in dieser Arbeit angesprochenen Gefahren: "Bots and Botnets" (26%), "Infrastructure Service DDoS" (11%) und "Link/Host Flooding" (9%), alle der als kritisch eingeschätzten Bedrohungen indirekt zu einem DoS führen. Die Intensität der DoS Angriffe wird immer gewaltiger und 2008 wurde zum ersten mal die Schranke der 40 Gigabits pro Sekunde erreicht und damit der Rekord des bisher stärksten Angriffs gebrochen [17].

V. Zusammenfassung und Ausblick

In dieser Seminararbeit wurde dargestellt, was es für verschiedene Arten an Denial-of-Service Angriffen gibt und auf welchen Prinzipien sie basieren. Obwohl über die letzten zwei Jahre DoS, durch erscheinen neuer Gefahren, bei den Betreibern etwas an Wichtigkeit verloren hat [17], [21], ist die Bedrohung nicht zu unterschätzen. Es werden auch weiterhin Recherchen in Maßnahmen zur Abschwäschung dieser Angriffe benötigt werden.

[9]Internet Service provider

Literatur

[1] F. Rötzer, "Estland beschuldigt Russland des Cyberterrorismus," *Telepolis*, May 2007.

[2] B. Tittelbach, "Angriff auf Estland," in *IAIK - Kritische Infrastrukturen*, October 2008.

[3] J. A. Lewis, "Securing cyberspace for the 44th presidency," in *A Report of the CSIS Commission on Cybersecurity for the 44th Presidency*, Washington DC., december 2008.

[4] J. Swaine, "Georgia: Russia 'conducting cyber war'," *Telegraph.co.uk*, August 2008.

[5] K. Coleman, "Cyber war 2.0 – russia v. georgia," *DefenceTech*, August 2008.

[6] P. Brauch, "Geld oder Netz!" *C'T 14/04*, 2004.

[7] J. Steward, "Storm Worm DDoS Attack," February 2007, http://www.secureworks.com/research/threats/storm-worm/.

[8] ——, "The changing Storm," October 2007, http://www.secureworks.com/research/blog/index.php/2007/10/15/the-changing-storm/.

[9] K. Poulsen, "FBI busts alleged DDoS Mafia," *Security Focus*, 2004.

[10] H. Gieselmann, "Schlechte Verlierer: Xbox-Spieler setzen Gegner per DDOS außer Gefecht," February 2009, http://www.heise.de/newsticker/Schlechte-Verlierer-Xbox-Spieler-setzen-Gegner-per-DDOS-ausser-Gefecht–/meldung/133316.

[11] HostBooter4free, "XR Bio Zombie 1.6 Halo 3/More Host Booter," http://www.youtube.com/watch?v=iCbSrbg8nA8, Videoanleitung für DDoS Angriffe.

[12] M. Kenney, "Ping-of-death," *available at http://insecure.org/sploits/ping-o-death.html*, 1996.

[13] "Internet protocol, darpa internet program protocol specification," September 1981, RFC-791.

[14] R. Smith, "Phlashdance, discovering permanent denial of service attacks against embedded systems," in *EUSecWest Security conference 2008*, May 2008.

[15] K. J. Higgins, "Permanent denial-of-service attack sabotages hardware," *darkreading.com*, May 2008.

[16] T. Aura, P. Nikander, and J. Leiwo, "DOS-resistant authentication with client puzzles," in *Lecture Notes in Computer Science*. Springer-Verlag, 2000, pp. 170–177.

[17] D. McPherson, D. C. Labovitz, and M. Hollyman, "Worldwide infrastructure security report," *ARBOR Networks*, October 2008, volume IV.

[18] D. S. Paul Ferguson, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," January 1998.

[19] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, "Ddos defense by offense," in *ACM SIGCOMM 2006*, Pisa, Italy, September 2006.

[20] D. Moore, C. Shannon, D. Brown, G. M. Voelker, and S. Savage, "Inferring internet Denial-of-Service activity," in *Inproceedings of the USENIX Security Symposium*, 2001.

[21] D. McPherson, D. C. Labovitz, and M. Hollyman, "Worldwide infrastructure security report," *ARBOR Networks*, September 2007, volume III.

Zero Configuration Networking

Daniel Siegel
Betreuer: Andreas Müller
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: siegel@in.tum.de

***Kurzfassung*— Eine Lampe wird an das Stromnetz angeschlossen, eingeschaltet und sie funktioniert. Es soll genauso einfach sein, ein Netzwerk ohne manuelle Konfiguration zu erstellen und zu benutzen. Das war das Ziel, als die Zeroconf Working Group ihre Arbeit in diesem Gebiet 1999 begann. Automatische Allokation von IP-Adressen ohne DHCP Server (IPv4 Link-Local Addressing), Übersetzung von Namen und IP-Adressen ohne einen DNS Server (Multicast DNS) und das Finden von Services im lokalen Netzwerk ohne einen Directory Server (DNS Service Discovery) war und ist die Grundlage von Zeroconf.**

In dieser Arbeit wird Zero Configuration Networking vorgestellt und auf den Aufbau, die Funktionsweise, bestehende Implementierungen und die Sicherheit von Zeroconf eingegangen. Es wird gezeigt, dass konfigurationslose Netzwerke kein Zukunftsszenario mehr darstellen und bereits jetzt im vollen Umfang nutzbar sind.

***Schlüsselworte*— Zero Configuration Networking, Zeroconf, IPv4LL, mDNS, DNS-SD, lokale Netzwerke, Zeroconf Working Group, Bonjour, Avahi**

I. Einleitung

Netzwerke einzurichten und zu konfigurieren galt schon immer als eine nicht sehr einfach zu lösende Aufgabe. In den meisten Fällen benötigt es gut ausgebildete Netzwerkadministratoren, die diese Aufgabe übernehmen. Hierbei wählen die Administratoren traditionellerweise zwischen statischer und dynamischer Adressierung: Statische Adressierung findet man vor allem bei fortwährend verbundenen Computern oder Netzen, wie z.B. bei Servern. Dynamische Adressierung hingegen wird vor allem bei begrenzter Anzahl von IP-Adressen, großer Teilnehmeranzahl oder schnell wechselnden Konfigurationen von Teilnehmern benutzt.

Bei allen oben genannten Möglichkeiten benötigt es einen Netzwerkadministrator, der entweder die statischen Adressen einrichtet oder Dienste, die dynamische Addressierung bewerkstelligen, verwaltet. Neue Teilnehmer zu diesen bereits bestehenden Netzwerken hinzuzufügen, bedeutet so einen Mehraufwand für den Administrator. Große Netzwerke einzurichten und zu verwalten ist zwar möglich, benötigt aber Manpower und Arbeitsaufwand.

Wenn nun aber nur wenige Teilnehmer schnell ein lokales Netzwerk aufbauen möchten, so ist in den meisten Fällen weder die Zeit, das Wissen oder ein Netzwerkadministrator zur Stelle, der dies bewerkstelligt. Wie ist es nun aber möglich, ein Ad-Hoc Netzwerk zu erstellen, das sich automatisch konfiguriert und nicht auf einen Netzwerkadministrator angewiesen ist? Genau diese Frage versucht Zero Configuration Networking zu lösen.

Als Analogie kann man sich folgendes Beispiel vorstellen: Man kauft sich eine Lampe bei einem Fachgeschäft. Zu Hause angekommen, steckt man nun die Lampe in eine Steckdose. Die Arbeit ist somit abgeschlossen. Wenn man nun aber nur schnell eine Datei über ein lokales Netzwerk auf einen Computer übertragen möchte, so scheitert es beispielsweise bereits an der korrekten Subnetzmaske. Eine Lampe besitzt so eine Subnetzmaske nicht. Zero Configuration Networking legt großen Wert auf die Einfachheit, auch wenn dies nicht das einzige Ziel ist [1].

Die Ausarbeitung ist wie folgt gegliedert: Zuallererst geht die Arbeit kurz auf die geschichtliche Entwicklung ein und zeigt ein Beispiel von Zeroconf auf. Kapitel II beschreibt die Anwendungsgebiete von Zeroconf, Kapitel III geht folgend auf die Funktionsweise ein. Einige Implementierungen werden in Kapitel IV vorgestellt, die Sicherheit von Zeroconf wird dann in Kapitel V beschrieben. Kapitel VI schließt die Arbeit mit einem Fazit und einem Ausblick ab.

A. Geschichte

Zur Zeit, als das Internet Protokoll (IP) entwickelt wurde, wurden von mehreren Unternehmen proprietäre Systeme erstellt, die ähnliche Ziele wie Zero Configuration Networking hatten. Dazu gehören AppleTalk von Apple, IPX von Novell und NetBIOS/SMB3 von Microsoft, die bereits automatische Adressierung, das Finden von Services im lokalen Netzwerk und teilweise Übersetzung von Namen mitbrachten und somit die Kommunikation und Benutzung von Services im lokalen Netzwerk ermöglichten. Somit war es bereits damals möglich, Drucker über das Netzwerk anzusprechen, wie in [2] beschrieben.

Diese Systeme wurden aber eingestellt bzw. kaum mehr benutzt, da IP eine robustere, bessere und offen dokumentierte Basis darstellt. So kann man Zeroconf als den Nachfolger von AppleTalk ansehen, jedoch auf der Basis von IP und vollständig funktionstüchtig [3].

B. Zero Configuration Networking an einem Beispiel

Zero Configuration Networking lässt sich am einfachsten an einem Beispiel erklären. Angenommen bei einer Präsentation

möchte der Computer des Vortragenden kein Bild anzeigen. Um nicht viel Zeit zu verlieren, sollen nun die Vortragsfolien auf einen anderen Präsentationslaptop übertragen werden. Um dies zu erreichen ist nur ein Ethernet-Kabel bzw. eine Wireless-Karte auf jedem der Computer nötig. Die Konfigurationen werden automatisch vorgenommen und sobald dieser Prozess abgeschlossen ist, können die Folien nun übertragen werden ohne weitere Einstellungen vorzunehmen. Dies funktioniert in dem Fall, dass bereits beide Computer mit einem anderen Netzwerk verbunden sind, z.B. Universität oder Internet, aber auch in dem Fall, wo kein solches Netzwerk verfügbar ist, ja nicht einmal ein Router oder Forwarder.

Die oben genannte automatische Konfiguration umfasst nun nicht nur die automatische Zuweisung einer IP-Adresse, sondern auch die Auffindung und Benutzung von Diensten, wie in den nächsten Kapiteln beschrieben wird.

II. Anwendungsgebiete

Zeroconf kann in sehr vielen Anwendungsgebieten benutzt werden, teilweise als Ersatz, teilweise als Erweiterung eines bereits bestehenden Netzwerkes. Grob kann man die Anwendungsgebiete in drei Szenarien einteilen [4]:

1) Ad-Hoc Netzwerke: Schnelle und kurzlebige Netzwerke, die meistens das Ziel haben, nur wenige bestimmte Aktionen, wie z.B. Dateitransfer, Präsentationen o.ä., auszuführen. Dabei können die Teilnehmer sehr schnell wechseln, sowie auch das Netzwerk wieder aufgelöst werden kann. Des weiteren sind hier meistens nur wenige Personen beteiligt.

2) Private Heimnetzwerke: Da immer mehr elektronische Geräte mit Netzwerktechnologien verfügbar sind und so sich in Wohnungen immer mehr solche Geräte befinden, kann eine Konfiguration eines Netzwerkes sehr viel Wissen und Zeit beanspruchen. Um dies zu umgehen, setzen bereits jetzt sehr viele Hersteller auf Zeroconf, um ein einfaches Einbinden der Geräte in das Netzwerk bereitzustellen.

Als Beispiele seien hier Netzwerkdrucker, Audiogeräte, Kameras u.v.m. genannt.

3) Große Netzwerke: Da ein herkömmliches Netzwerk oft sehr viel Aufwand mit sich bringt, verwenden Administratoren zunehmend Zeroconf, um den Aufwand zu minimieren. Besonders bei externen Mitarbeiter oder Gästen einer Firma, zahlen sich automatische Konfigurationen aus, da diese Personen oft nicht lange im Unternehmen verweilen.

III. Zero Configuration Networking

Zero Configuration Networking, in Kurzform auch Zeroconf genannt, wird am besten durch die Definition der Zero Configuration Working Group beschrieben [5]:

> The goal of the Zero Configuration Networking (ZEROCONF) Working Group is to enable networking in the absence of configuration and administration. Zero configuration networking is required for environments where administration is impractical or impossible, such as in the home or small office, embedded systems 'plugged together' as in an automobile, or to allow impromptu networks as between the devices of strangers on a train.

Frei übersetzt bedeutet dies folgendes: Das Ziel von Zero Configuration Networking ist, Benutzern die Möglichkeit zu eröffnen, ein Netzwerk ohne Konfiguration und ohne Verwaltungsaufwand zu erstellen. Sei es nun, weil ein die Verwaltung und Erstellung eines Netzwerkes unmöglich ist, aber auch wenn es nicht praktikabel ist. Sozusagen soll Zero Configuration Networking die Erstellung von Ad-Hoc Netzen und die Verwendung derselben erleichtern, aber nicht nur darauf beschränkt werden.

Die Zeroconf Working Group hat dabei folgende drei Problembereiche identifiziert und bereits gelöst [6], [7]:

- Automatische Allokation von IP-Adressen ohne DHCP Server (IPv4 Link-Local Addressing)
- Übersetzung von Namen und IP-Adressen ohne einen DNS Server (Multicast DNS)
- Finden von Services im lokalen Netzwerk ohne einen Directory Server (DNS Service Discovery)

Des weiteren soll Zeroconf keine Störungen in bereits existierenden Netzwerken hervorrufen und für den Benutzer transparent erscheinen. Deshalb werden von der Zero Configuration Networking Working Group folgende Punkte gefordert [1], [5], [6]:

- Zeroconf darf keine Auswirkungen auf bereits bestehende Netzwerke haben, so muss beispielsweise die Adressauflösung auch bei einem bereits bestehenden DHCP Server funktionieren und das Netzwerk darf keinen Schaden davon ziehen.
- Zeroconf soll die Auswirkungen auf Anwendungen minimal halten und versuchen, so transparent wie nur möglich zu sein. So sollen bestehende Anwendungen immer noch korrekt funktionieren.
- Der Sicherheitsstandard der Zeroconf Protokolle darf nicht kleiner sein, als andere aktuelle ähnliche bzw. zugehörige IETF Protokolle, die dem IETF Standard angehören.

A. Automatische Allokation von IP-Adressen ohne DHCP Server

Der erste Problembereich von Zeroconf ist die automatische Zuweisung von IP-Adressen. Um in einem IP-basierenden Netzwerk Pakete verschicken zu können, benötigt jeder Teilnehmer eine eigene, im Netzwerk einmalige, IP-Adresse. In zentralen Netzwerken geschieht diese Zuweisung meist durch einen DHCP-Server oder durch bereits vorher für jeden Teilnehmer festgelegte Adressen.

Zeroconf vergibt die IP-Adressen jedoch zufällig und ohne Eingreifen eines Administrators. Das verfolgte Ziel hierbei ist zu bewerkstelligen, dass jeder Teilnehmer eines Netzwerkes über die jeweils anderen Teilnehmer im selben Netzwerk Bescheid weiß. Dies erfordert IP-Adressen mit besonderen Eigenschaften, die IPv4 Link-Local Adressen genannt werden [8]. Dazu gehören alle IP-Adressen im Bereich von 169.254/16 (169.254.xxx.xxx). Die ersten und letzten 256 Adressen (169.254.0.0 bis 169.254.1.0 und 169.254.254.255 bis 169.254.255.255) sind jedoch für zukünftigen Gebrauch

reserviert und dürfen nicht verwendet werden [8]. Des weiteren dürfen diese Adressen in einem Netzwerk nur einmalig vorkommen. Die Voraussetzungen für eine automatische Adressvergabe beinhaltet nun auch, dass es einem Teilnehmer möglich ist [2]

- selbstständig seine Netzwerkschnittstellen mit einmaligen Adressen zu konfigurieren
- festzustellen welche Subnetzmaske zu benutzen ist
- festzustellen, ob eine Adresse doppelt benutzt wird
- Kollisionen zu bewältigen

Die Adressvergabe wird nun mit einem auf dem Address Resolution Protocol (ARP) aufbauenden Verfahren umgesetzt. Der Teilnehmer wählt sich pseudo-zufällig aus dem oben genannten Adressraum eine IP-Adresse aus. Dabei werden rechnerspezifische Informationen, wie z.B. die MAC-Adresse der Netzwerkschnittstelle, berücksichtigt, um soweit möglich immer die gleiche IP-Adresse zu erhalten. Dies erhöht die Stabilität von Zeroconf [8].

Wenn nun aber identische Informationen benutzt werden, wie z.B. die Systemzeit zweier gleichzeitig eingeschalteter Systeme, können leicht Konflikte entstehen. Nachdem die IP-Adresse generiert wurde, muss nun also überprüft werden, ob diese nicht schon in Verwendung ist. Natürlich darf bis zu dem Punkt, wo feststeht, dass dieser Teilnehmer der einzige Besitzer dieser IP-Adresse ist, die gewählte IP-Adresse nicht veröffentlicht werden, beispielsweise durch IP- oder ARP-Pakete [9].

Des weiteren darf diese Überprüfung nur nach der Adressgenerierung durchgeführt werden und nicht wiederholt werden, um Netzwerkressourcen nicht zu verschwenden. Die Überprüfung geschieht mit Hilfe von ARP Probes. Eine ARP Probe ist ein ARP Paket in welchem die gewählte IP-Adresse als Empfänger und 0.0.0.0 als Absender eingetragen ist. Auch die Ziel-MAC-Adresse wird dabei ignoriert und mit Nullen aufgefüllt [9].

Tabelle I

IPV4 LINK-LOCAL KONSTANTEN, ENTNOMMEN AUS [8]

PROBE_WAIT	1 second	initial random delay
PROBE_NUM	3	number of probe packets
PROBE_MIN	1 second	minimum delay till repeated probe
PROBE_MAX	2 seconds	maximum delay till repeated probe
ANNOUNCE_WAIT	2 seconds	delay before announcing
ANNOUNCE_NUM	2	number of announcement packets
ANNOUNCE_INTERVAL	2 seconds	time between announcement packets
MAX_CONFLICTS	10	max conflicts before rate limiting
RATE_LIMIT_INTERVAL	60 seconds	delay between successive attempts
DEFEND_INTERVAL	10 seconds	minimum interval between defensive ARPs

Nach dem Versenden dieser ARP Probe, wartet der Teilnehmer eine zufällige Zeit zwischen 0 und PROBE_WAIT (diese und folgende Konstanten sind in Tabelle I aufgelistet) Sekunden. Nach der Wartezeit versendet der Teilnehmer PROBE_NUM ARP Probes, wobei der Zeitabstand zwischen den einzelnen Probes zwischen PROBE_MIN und PROBE_MAX betragen muss. Nach der letzten ARP Probe wartet der Teilnehmer noch ANNOUNCE_WAIT Sekunden. Hat der Teilnehmer nun zwischen dem Anfang der Überprüfung und dem Ende der Wartezeit von ANNOUNCE_WAIT eine ARP Probe empfangen, die die gewünschte IP-Adresse des Teilnehmers im Empfänger Feld des ARP Pakets enthält und dabei die MAC Adresse nicht die der Netzwerkschnittstelle des Teilnehmers entspricht, so tritt ein Konflikt auf. In diesem Fall muss nun eine neue IP-Adresse generiert werden und die Prozedur beginnt wieder von vorne. Wenn mehrere Teilnehmer die gleiche Adresse überprüfen oder ein Teilnehmer bereits diese Adresse besitzt, das, wie bereits oben erwähnt, passieren kann, so tritt dieses Szenario auf [8].

Hierbei kann nun eine Endlosschleife eintreten und das Netzwerk wird mit ARP Paketen überlastet. Um dies zu verhindern, muss jeder Teilnehmer nach MAX_CONFLICTS Konflikten seine Geschwindigkeit, mit der er seine IP-Adressen generiert und überprüft, drosseln. Reduziert wird auf eine Überprüfung pro RATE_LIMIT_INTERVAL [9].

Wurde nun jedoch kein entsprechendes ARP Paket empfangen und somit kein anderer Teilnehmer mit der gewählten IP-Adresse gefunden, so kann der Teilnehmer diese IP-Adresse für sich beanspruchen. Jetzt muss dieser noch den anderen Teilnehmern mit Hilfe von ARP Announcements bekannt geben, dass er diese Adresse für sich beansprucht. Ein ARP Announcement ist wiederum ein ARP Paket, in welchem die jeweilige beanspruchte IP-Adresse in das Empfänger- und Absenderfeld eingetragen wird. Er sendet nun ANNOUNCE_NUM ARP Announcements mit einem Abstand von ANNOUNCE_INTERVAL Sekunden. Jetzt kann und muss jeder Teilnehmer seinen Cache auffrischen und die neue Adresse eintragen. Ansonsten wäre es möglich, dass ein anderer Teilnehmer eine veraltete Adresse gespeichert hat [8].

Empfängt nun der Teilnehmer ein ARP Paket, und somit eine ARP Probe auf seine IP-Adresse, entsteht wieder ein Konflikt. Der Teilnehmer hat nun zwei Möglichkeiten: Entweder er verteidigt seine IP-Adresse oder er wählt eine neue. Sofern der Teilnehmer noch offene Verbindungen hat, beispielsweise einen Dateitransfer, wird er die Adresse meistens verteidigen. Die Verteidigung erfolgt durch das Verschicken eines ARP Announcements. Hat der Teilnehmer jedoch zuvor schon einen Konflikt feststellen können, so muss eine neue Adresse gewählt werden, um eine Endlosschleife zu vermeiden. Diese kann entstehen, wenn beide Teilnehmer versuchen ihre Adresse zu verteidigen [8].

Abbildung 1 zeigt ein beispielhaftes Zeroconf Netzwerk. Jede Netzwerkschnittstelle jedes Teilnehmers kann eine eigene, eindeutige IP-Adresse haben, obwohl sich im Netzwerk Wireless- und Kabelgebundene Schnittstellen befinden. Teilnehmer 1 und Teilnehmer 2 benutzen eine Wireless Verbindung und die Adressen A und B sind verschieden und eindeutig. Teilnehmer 2 und Teilnehmer 3 sind über ein Kabel verbunden und somit sind Adressen C und D eindeutig. Teilnehmer 2 wird nun aber auf keinen Fall bei der

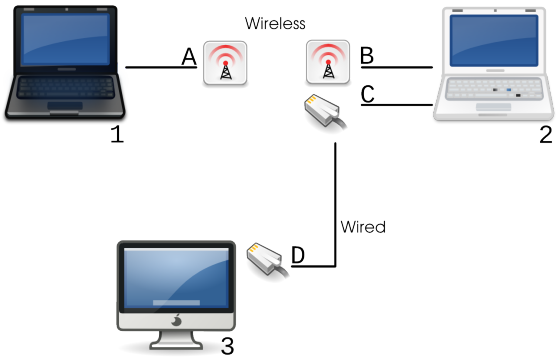

Abbildung 1. Automatische IP-Konfiguration. In diesem Schaubild besitzt jede Netzwerkschnittstelle A-D eine einmalige IP-Adresse für jede Wireless oder kabelgebundene Verbindung. [1]

kabelgebundenen Verbindung die Adresse A von Teilnehmer 1 verwenden. Allgemein wird ein Teilnehmer keine Adresse wählen, die mit irgendeinem anderen Teilnehmer auf irgendeiner Netzwerkschnittstelle kollidieren. Dies aus dem einfachen Grund, da Missverständnisse für diejenigen Teilnehmer auftreten könnten, die über mehrere Netzwerkschnittstellen mit dem Netzwerk verbunden sind.

Dass das Verfahren trotzdem gut skaliert, zeigen einige Experimente, wie z.B., dass die Wahrscheinlichkeit eine noch nicht benutzte Adresse in einem Netzwerk von 1300 Teilnehmer nach dem zweiten Versuch zu wählen, auf bis zu 99.96% anwachsen kann [10].

Bisher wurde nur IPv4 betrachtet, IPv6 bringt im Gegenteil zu IPv4 die automatische Adressierung von Haus aus schon mit. Dies gehört aber nicht explizit zu Zeroconf, deshalb wird nur kurz darauf eingegangen. Bei IPv6 befinden sich die Link-Local Adressen im Bereich fe80::/64. Die Adressen werden aus der MAC-Adresse der Netzwerkschnittstelle und dem fe80-Prefix erstellt. Des weiteren fällt die Konfliktüberprüfung weg, da jede MAC Adresse bereits eindeutig ist. Ein weiterer Punkt ist das Faktum, dass Link-Local IPv6-Adressen nicht geroutet werden dürfen, da sie ja schon eindeutig sind. Die automatische Adressierung wird in [11] genauer beschrieben.

B. Übersetzung von Namen und IP-Adressen ohne einen DNS Server

Da bekanntermaßen Namen besser gemerkt werden können als Zahlen, wäre es sehr unhandlich, andere Teilnehmer über deren IP-Adresse anzusprechen. Deshalb benutzt man das Domain Name System (DNS) um Namen auf IP-Adressen umzuwandeln und umgekehrt.

[1]Diese und folgende selbst erstellten Abbildungen benutzen Werke aus dem Tango Project (`http://tango.freedesktop.org/Tango_Desktop_Project`): *gnome-icon-theme-extras* unterliegt der GPL. *tango-icon-theme* ist unter Public Domain veröffentlicht.

In den meisten Netzwerken befinden sich aus diesem Grund auch DNS Server, die diese Tätigkeit ausüben. Aber gerade an Orten, wie z.B. bei Konferenzen oder Flughäfen ist es sehr unwahrscheinlich ein bereits konfiguriertes Netzwerk mit DNS Server vorzufinden. Das Fehlen solcher DNS Server führt natürlich zu dem Problem, dass eine Umwandlung von Namen zu IP-Adressen und umgekehrt nicht mehr möglich ist.

Hierfür gibt es zwei sehr ähnliche Lösungen:

- Link-Local Multicast Name Resolution (LLMNR) von Microsoft, das jedoch kaum bis keine Verwendung findet und erst kürzlich als RFC beschrieben wurde [12].
- Multicast DNS (mDNS) von Apple, das offen beschrieben wurde und Bestandteil von Zeroconf ist.

Aus diesen naheliegenden Gründen wird nur auf mDNS eingegangen. Multicast DNS schlägt, wie der Name bereits sagt eine geringfügige Änderung von DNS vor, nämlich Multicast. Dies bedeutet einfach eine andere Verfahrensweise wenn ein Teilnehmer eine DNS Abfrage schicken möchte, die gleich beschrieben werden. Multicast bedeutet hierbei, dass eine Nachricht von einem Teilnehmer eines Netzwerkes gesendet wird und von einer Gruppe von Teilnehmern des Netzwerkes empfangen wird. So kann jeder Teilnehmer, der Interesse an dieser Nachricht zeigt, diese empfangen. Um dies im Netzwerk zu realisieren, müssen solche Nachrichten an die IPv4-Adresse 224.0.0.251 (IPv6: ff02::fb) geschickt werden [13].

Grundsätzlich gibt es bei Zeroconf eine neue Top Level Domain (TLD), nämlich `.local`. Alle Domains mit dieser Endung sind frei verfügbar und können auch nicht, wie andere Domains, erworben werden. So kann sich jeder Teilnehmer eine Domain aus der `.local`-Domäne auswählen. Einzig zu beachten dabei ist, dass kein bereits vergebener Name benutzt werden soll. Dass dies zu Konflikten führen kann, ist offensichtlich. Von der Zeroconf Working Group wurde dieser Fall jedoch absichtlich nicht beschrieben, da es zum einen eher unwahrscheinlich scheint, dass zwei Teilnehmer den gleichen Namen auswählen. Zum anderen, auch weitaus wichtigeren Punkt, kann es für solche Mehrfachvergaben auch sinnvolle Anwendungen geben. Als Beispiel sei hier Load Balancing genannt [13].

Jeder Teilnehmer wählt nun also einen Fully Qualified Domain Name (FQDN), z.B. *macbook.local*, beim Betreten des Netzwerkes und kündigt diesen im Netzwerk an. Dadurch hat der Teilnehmer Besitzansprüche auf diese Domain. DNS Anfragen werden nun in das Netzwerk über Multicast geschickt und der Teilnehmer, der für die angefragte Domain zuständig ist, antwortet wiederum über Multicast. So kann nun jeder Teilnehmer im Netzwerk seinen Cache aktualisieren [10].

Auch Reverse-DNS Anfragen, die im Gegensatz zu DNS zu einer IP den Namen des Besitzers der IP erfahren möchten werden über oben genannte Multicast Adresse versandt [9].

Nun können solche DNS Anfragen eine hohe Netzwerklast einfordern und so wurden von der Zeroconf Working Group folgende Maßnahmen vorgeschlagen, welche zu einer Reduzierung des Traffics über das Netzwerk führen sollen. Dadurch sollen doppelte oder mehrfache DNS Anfragen verhindert werden [13].

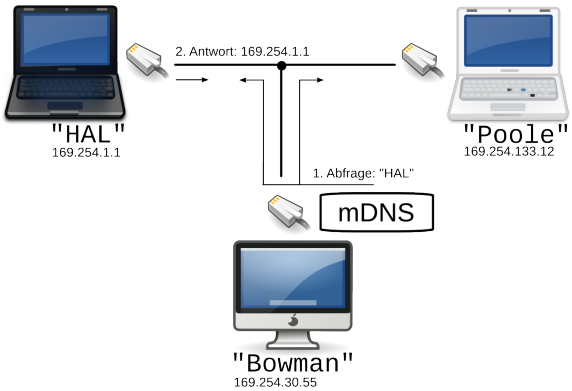

Abbildung 2. Multicast DNS. Der Teilnehmer „Bowman“ möchte die IP-Adresse von „HAL“ erfahren. Dazu schickt er eine Multicast DNS Anfrage an die bekannte Multicast Adresse, die dann jeder Teilnehmer im Netzwerk empfängt. Wenn „HAL“ im Netzwerk existiert, so antwortet dieser, was im Schaubild zu sehen ist. Gleichzeitig aktualisiert „Poole“ seinen Cache mit der Antwort von „HAL“. Aus Gründen der Übersichtlichkeit wurde der Suffix .local nicht an die Namen angefügt, jedoch endet jeder FQDN mit .local.

1) Known Answer Suppression: Wenn ein Teilnehmer eine Multicast DNS Anfrage verschicken möchte, zu der er schon einige Antworten in seinem Cache hat, so fügt er diese Antworten an seine Anfrage an.

Der Besitzer muss nun nicht auf diese Anfrage antworten, wenn die korrekte Antwort bereits in der DNS Abfrage enthalten ist und die Time To Live (TTL) dieser Antwort noch größer ist, als die Hälfte der üblichen TTL. Ist die TTL jedoch kleiner, so muss dieser die Anfrage beantworten, um die TTL der Caches zu aktualisieren.

Da der Besitzer eine Antwort verschicken muss, falls die TTL zu klein ist, ist dem Teilnehmer, der eine Abfrage verschickt, nicht erlaubt solche Einträge mitzuschicken, deren TTL schon kleiner als die Hälfte der üblichen TTL ist.

Des weiteren dürfen die restlichen Teilnehmer die Antworten dieser Multicast DNS Abfragen nicht in ihren Cache aufnehmen, da diese nicht vom Besitzer der Domain stammen und so möglicherweise schon veraltet oder falsch sein könnten.

2) Multi-Packet Known Answer Suppression: Falls nun ein Teilnehmer zu einer Abfrage mehr Antworten besitzt, als in einer Multicast DNS Abfrage Platz finden, so muss dieser die Abfrage senden und so viele Antworten in das Paket geben, wie Platz verfügbar ist. Danach setzt dieser das Truncated (TC) Bit und schickt eine nächste DNS Abfrage jedoch ohne Interesse an einer Domain, aber mit den restlichen Antworten, die der Teilnehmer im Cache gespeichert hat. Dies macht er solange, bist alle Antworten verschickt wurden. Beim letzten Paket wird natürlich das TC Bit nicht mehr gesetzt.

Der Besitzer wartet nun eine zufällige Zeit von 400 bis 500ms zwischen den Paketen, um dann diese Abfrage zu beantworten. Ist nun eine der Antworten in den Paketen dieselbe, die der Besitzer geben würde, so löscht er diese aus der Abfrage und beantwortet diese nicht. Falsche oder fehlende Antworten werden aber trotzdem mit der richtigen Antwort beantwortet.

3) Duplicate Question Suppression: Wenn ein Teilnehmer eine Abfrage verschicken möchte und ein anderer Teilnehmer in diesem Moment die gleiche Abfrage bzw. eine ähnliche, die jedoch auf die gleiche Antwort hinausläuft, so soll dieser Teilnehmer die Abfrage des anderen Teilnehmers als seine eigene betrachten und so redundanten Netzwerkverkehr vermeiden. Er wartet also auf die Antwort, die über Multicast DNS ja sowieso alle Teilnehmer erreicht.

4) Duplicate Answer Suppression: Wenn ein Besitzer einer Domain gerade eine Antwort vorbereitet und eine Abfrage eines Teilnehmers erfolgt, die die selben Antworten und eine TTL, die jedoch mindestens gleich groß ist, wie die TTL in der Antwort des Besitzers beinhaltet, so versendet der Besitzer diese Antwort nicht und nimmt die Antwort als gegeben an.

C. Finden von Services im lokalen Netzwerk ohne einen Directory Server

Um einen Service benutzen zu können, muss ein Benutzer zuallererst wissen, wo sich der Service befindet und welches Protokoll dieser benutzt. Nun gibt es aber keine Directory Server in unserem Netzwerk, weshalb eine andere Lösung verwendet werden muss.

Die Zeroconf Working Group schlägt DNS Service Discovery (DNS-SD) als Lösungsansatz vor. DNS-SD setzt auf mDNS auf, kann jedoch auch auf „normalen“ DNS Servern operieren [14].

DNS-SD erweitert DNS, sodass es möglich ist, auch Services abfragbar zu machen. DNS bietet sich gerade deshalb an, weil bereits mit mDNS und DNS zwei Lösungen für lokale Netzwerke und solche mit DNS Servern existieren.

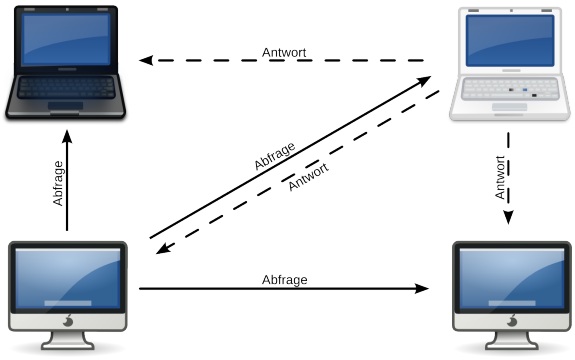

Abbildung 3. Der Teilnehmer (in der Abbildung links unten) stellt eine Anfrage für ein bestimmten Service, z.B. _http._tcp.local, die über Multicast an alle Teilnehmer versandt wird. Derjenige, der nun in seinen Resource Records diesen Service gespeichert hat, schickt eine Antwort, wiederum über Multicast, zurück, wo der Service zu finden ist.

Die Anforderungen sind hierbei, dass es zum einen möglich sein muss, Services eines bestimmten Typs in einer bestimmten Domain zu suchen. Diese Information soll nun dazu verwendet werden, die IP-Adresse und den Port des Services herauszufinden. Des weiteren müssen die Services auffindbar bleiben, auch wenn sich die IP-Adresse oder der Port verändern.

DNS-SD benutzt nun DNS SRV Resource Records (RR) [15], um Services zu finden. DNS SRV Resource Records propagieren per DNS Services, die unter der aktuellen Domain verfügbar sind. Dazu haben DNS SRV RR ein spezielles Format, das in Tabelle II dargestellt ist [15].

Tabelle II
AUFBAU SRV RESOURCE RECORDS [15]

```
_Service._Proto.Name TTL Class SRV Priority Weight Port
```

Service	Der symbolische Name des Services.
Proto	Das Protokoll des Services.
Name	Die Domain für diesen RR.
TTL	Gibt an, wie lange der RR im Cache gehalten werden darf.
Class	Die DNS Klasse, immer auf „IN“ für Internet.
Priority	Die Priorität dieses Services, falls mehrere identische Services angeboten werden. Je kleiner diese Zahl, desto höhere Priorität hat der Service.
Weight	Das Gewicht des Services, falls mehrere Services die gleiche Priorität haben. Je höher das Gewicht, desto wichtiger ist der Service.
Port	TCP oder UDP Port.
Target	Der Canonical Name (CNAME) des Rechners, der diesen Dienst anbietet.

Ein SRV RR für einen Webserver auf `www.example.com`, der auf Port 80 läuft mit Priorität 10 und Gewicht 0 ist beispielhaft in Tabelle III dargestellt.

Tabelle III
BEISPIELHAFTE DARSTELLUNG EINES SRV RESOURCE RECORDS

```
_http._tcp.example.com 3600 IN SRV 10 0 80 ↩
                                        www.example.com.
```

Es ist auch möglich, mehrere SRV RR für denselben Service einzutragen um beispielsweise Load Balancing durchzuführen. So kann man einfach eine Anfrage auf `_http._tcp.example.com` stellen und bekommt alle Webserver in dieser Domäne.

Wie bereits erwähnt wird jeder Service eindeutig durch `Service.Protocol.Domain` identifiziert. Hervorzuheben ist hierbei, dass man für jeden Service mehrere Instanzen haben kann, z.B. `laser_printer._ipp._tcp.example.com`. Dabei handelt es sich um einen UTF-8 kodierten Text, der im DNS Paket mitgeschickt werden kann. Aufgrund dessen und dem Punkt, dass bei der Entwicklung dieser Protokolle auf keine Kompatibilität Wert gelegt werden musste, entstehen keine Limitierungen bei der Namensvergebung [3]. Es kann also ein beliebiger Text, wie z.B. „Drucker im 2. Stock, pro Blatt 0,3€ in die Kasse!“ als Bezeichnung der Instanz dienen. Außerdem lautet die Domain in einem Ad-Hoc Netzwerk `.local`, d.h. `_service._prot.local` liefert alle Services von „service“ über das Protokoll „prot“ im lokalen Netzwerk zurück.

Weiters kann bei der Abfrage noch ein TXT Record abgefragt werden, der zusätzliche Informationen für den Service bereitstellt. Informationen werden darin in Key/Value Paaren abgelegt und der Eintrag kann bis zu einer Größe von 65535 Bytes befüllt werden. Die darin enthaltenen Informationen sind optional und natürlich für jeden Service unterschiedlich. Trotzdem muss ein TXT Record zu einem SRV RR vorhanden sein, der mindestens ein Byte beinhaltet, z.B. das Nullzeichen [14]. Beispielsweise würde man sich bei einem Druckservice wünschen, die Papiergröße, und die Telefonnummer des Supports zu kennen, bevor der Druckauftrag begonnen hat.

Neben DNS-SD gibt es noch mehrere andere Dienste, wie SLP (Service Location Protocol) oder SSDP (Simple Service Discovery Protocol), die jedoch nicht von der Zeroconf Working Group akzeptiert wurden und wenig Verwendung finden [16] beschreibt das SLP Protokoll und [17] SSDP.

IV. IMPLEMENTIERUNGEN

Da die Zeroconf Protokolle offen spezifiziert sind und jedem freien Zugang ermöglichen, gibt es zahlreiche Implementierungen. Es wird nun nur auf die beiden wichtigsten kurz eingegangen und so sind diese beiden in Tabelle IV dargestellt.

V. SICHERHEIT

Auto-Konfigurationsprotokolle, wie die Allokation von IP-Adressen, die Übersetzung von Namen in IP-Adressen und das Finden von Services im lokalen Netzwerk sind nicht sicher, da keine Sicherheitsmechanismen eingebaut wurden. [2] Aus diesem Grund konnten die Protokolle einfach gehalten werden, sind aber auch gegen Angriffe verwundbar.

Auch wenn Automatisierbarkeit eigentlich Zeroconf ausmacht und eines der Ziele davon ist, kann Sicherheit nicht automatisiert werden [2]. Wie bereits vorher erwähnt, dürfen aber Zeroconf Protokolle nicht weniger sicher sein, als vergleichbare IETF-Standards. Trotzdem kann Zeroconf einen weiteren Zugang zu einem Netzwerk bedeuten und soll besonders dann, wenn weitere Netzwerke daran grenzen abgesichert oder nicht benutzt werden.

Die Zeroconf Working Group ist sich diesem Problem bewusst und hat bereits mehrere Mechanismen für die Absicherung von Zeroconf besprochen, jedoch wurde nach [2] noch keine genaue Spezifikation vorgeschlagen.

Nun ist aber auch die Nichtbenutzung von Zeroconf kein Gewinn für die Sicherheit, denn Angreifer wissen bereits, wie man Teilnehmer und Services eines Netzwerkes auffinden kann, auch wenn kein Zeroconf benutzt wird [7]. So kann man sagen, dass Zeroconf nur dem Benutzer hilft, einfacher ein Netzwerk zu erstellen und zu benutzen.

VI. ZUSAMMENFASSUNG UND AUSBLICK

Die drei großen Bereiche von Zeroconf wurden vorgestellt, die Automatische Allokation von IP-Adressen (IPv4 Link-Local Addressing), die Übersetzung von Namen und IP-Adressen (Multicast DNS) und das Finden von Services im lokalen Netzwerk (DNS Service Discovery).

Tabelle IV
ZWEI DER WICHTIGSTEN ZEROCONF IMPLEMENTATIONEN

Bonjour

Autor:	Apple Inc.
Lizenz:	Apple Inc. - Proprietäre Freeware. Teile unter der Apache License, Version 2.0.
OS:	Mac OS X, Microsoft Windows
Webseite:	`http://developer.apple.com/bonjour`
Beschreibung:	Bonjour [18] (ehemals auch Rendezvous genannt) ist eine Zeroconf Implementierung, die von Apple Inc. entwickelt wird. Beteiligt ist dabei auch Stuart Cheshire, der einer der leitenden Köpfe der Zeroconf Working Group ist. Bonjour ist im Mac OS X Betriebssystem seit Version 10.2 enthalten und kann des weiteren auch auf Microsoft Windows Systemen installiert werden. Teile von Bonjour, wie der mDNSResponder sind auch unter *BSD und GNU/Linux verfügbar und lauffähig.

Avahi

Autor:	Lennart Poettering und Trent Lloyd
Lizenz:	LGPL
OS:	GNU/Linux, Solaris, Mac OS X, *BSD
Webseite:	`http://www.avahi.org`
Beschreibung:	Avahi ist eine freie Implementierung von Zeroconf, die unter der LGPL veröffentlicht ist. Seit 2004 wird Avahi entwickelt (vormals FlexMDNS) und ist momentan eine der besten Implementierungen von Zeroconf [3]. Es unterstützt IPv4LL, mDNS and DNS-SD und ist der de-facto Standard unter den meisten Linux und *BSD Distributionen [19].

Zeroconf eignet sich besonders für Ad-Hoc Netzwerke, Heimnetzwerke und kleine Firmennetzwerke. Besonders auch dann, wenn die beteiligten Personen wenig Fachwissen über dieses Gebiet mitbringen.

Aber auch aus dem Grund, da Zeroconf mit einem bereits bestehenden Netzwerk keine Probleme hat, kann sich der Administrator Arbeit sparen, wenn etwa einige neue Drucker und Dateiserver in das Netzwerk eingebunden werden sollen. Hier muss er etwa diese Geräte nur dem Netzwerk hinzufügen. Das Bereitstellen der Services geschieht dann automatisch. Zu beachten ist hier einzig, dass sich das Verkehrsaufkommen erhöhen wird.

Da Zeroconf keine wesentliche Sicherheitsmechanismen mitbringt, ist die Wahrscheinlichkeit sehr hoch, dass nicht befugte Benutzer diesem Netzwerk beitreten. Hier benötigt man zusätzliche Sicherheitsmechanismen, und sollte ohne diese keine kritischen Aufgaben erledigen.

Zeroconf ist nun bereits Standard für Auto-Konfigurationsnetzwerke, Apple besitzt eine sehr gute Integrierung von Zeroconf in seinen Produkten, GNU/Linux und restliche Unix-Systeme haben auch eine sehr gute Implementierung von Zeroconf, nämlich Avahi, und eine teilweise so gute Implementierung in bestimmten Applikationen. Windows hat nur Apple's Bonjour zur Verfügung und unterstützt Zeroconf nur teilweise von Haus aus. So bleibt abzuwarten, ob sich die Windows-Welt diesem Trend anschließt.

Ein weiterer Punkt ist, ob Sicherheitstechnologien in Zeroconf in naher Zukunft noch aufgenommen werden. Dies scheint aber wegen der Auflösung der Zeroconf Working Group [5] unwahrscheinlich.

ACKNOWLEDGMENT

Vielen Dank an Lennart Poettering, dem Entwickler von Avahi, für die Beantwortung zahlreicher Fragen.

LITERATUR

[1] Zeroconf Working Group, "Official website of the Zeroconf Working Group," http://www.zeroconf.org, March 2009.
[2] E. Guttman, "Autoconfiguration for IP Networking: Enabling Local Communication," *IEEE Internet Computing*, vol. 5, no. 3, pp. 81–86, May/June 2001.
[3] S. Cheshire, "Stuart Cheshire speaks about Zeroconf at Google," Google TechTalks, http://video.google.com/videoplay?docid=-7398680103951126462, November 2005.
[4] T. Kollbach and C. Beier, "Zero Configuration Networking," http://www2.informatik.hu-berlin.de/~beier/txts/2007-Rechnerkommunikation%20--%20Zeroconf%20Networking.pdf, July 2007.
[5] Zeroconf Working Group, "Description of the Zeroconf Working Group," http://www.zeroconf.org/zeroconf-charter.html, March 2009.
[6] A. Williams, "Requirements for Automatic Configuration of IP Hosts," IETF Internet-Draft, March 2003.
[7] IT-University of Gothenburg, "Zero Configuration Networking," Design of Complex Systems, http://pop.blandband.se/cv-files/ZeroConfArticle.pdf, 2005.
[8] S. Cheshire, B. Aboba, and E. Guttman, "Dynamic Configuration of IPv4 Link-Local Addresses," RFC 3927 (Proposed Standard), May 2005.
[9] G. Stamboliev, "Zeroconf," http://wwwspies.informatik.tu-muenchen.de/lehre/seminare/SS05/hauptsem/Ausarbeitung03.pdf, 2005.
[10] T. Niemueller, "Zero Configuration Networking," *Lecture Notes in Informatics (LNI) - Seminars*, vol. S-3, pp. 143–146, 2006.
[11] S. Thomson, T. Narten, and T. Jinmei, "IPv6 Stateless Address Autoconfiguration," RFC 4862 (Proposed Standard), September 2007.
[12] B. Aboba, D. Thaler, and L. Esibov, "Link-Local Multicast Name Resolution (LLMNR)," RFC 4795 (Proposed Standard), January 2007.
[13] S. Cheshire and M. Krochmal, "Multicast DNS," IETF Internet-Draft, September 2008.
[14] ——, "DNS-Based Service Discovery," IETF Internet-Draft, September 2008.
[15] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," RFC 2782 (Proposed Standard), February 2000.
[16] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," RFC 2608 (Proposed Standard), June 1999.
[17] Y. Y. Goland, T. Cai, P. Leach, Y. Gu, and S. Albright, "Simple Service Discovery Protocol/1.0," IETF Internet-Draft, October 1999.
[18] Apple Inc., "Bonjour project site," http://developer.apple.com/networking/bonjour, March 2009.
[19] Avahi Project, "Avahi project site," http://www.avahi.org, March 2009.
[20] S. Cheshire, "IPv4 Address Conflict Detection," RFC 5227 (Proposed Standard), July 2008.

Reliable Server Pooling (RSerPool)

Konrad Windszus
Betreuer: Nils Kammenhuber
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: windszus@in.tum.de

***Zusammenfassung*— Zur Erfüllung von Anforderungen wie Redundanz und Verfügbarkeit wurde eine neue Protokollsuite mit Namen Reliable Server Pooling standardisiert, die die Verwaltung und Zuteilung von Servern aus einem Serverpool spezifiziert. Diese Ausarbeitung soll einen Überblick über die Protokollsuite und deren Anwendungsmöglichkeiten bieten.**

***Schlüsselworte*— RSerPool, Reliable Server Pooling, ASAP, ENRP, SCTP, Round-Robin, Least-Used**

I. Einleitung

Schon seit 2002 existiert der RFC3237 [1], der die Anforderungen an eine neue Protokollsuite mit dem Namen Reliable Server Pooling beschreibt. Unter Mitarbeit von Siemens, Nokia, Motorola und Cisco entstand dieses Anforderungsdokument. Orientiert hat man sich dabei am Signaling System No. 7 (SS7), einer Protokollsuite aus der Telefonwelt [2]. Eine ähnliche Protokollsuite sollte nun auch für IP-basierte Dienste entstehen und hohe Anforderungen in Bezug auf Verfügbarkeit und Redundanz, aber auch in Bezug auf Echtzeit (Reaktion bei Serverausfall), Skalierbarkeit, Erweiterbarkeit und Einfachheit erfüllen. Insbesondere die letzte Eigenschaft sollte es möglich machen, bestimmte Entitäten der Reliable-Server-Pooling-Architektur wie z.B. die Clients auch auf kleinen Geräten (wie z.B. mobilen oder integrierten Computern) zu betreiben.

Reliable Server Pooling (abgekürzt RSerPool) wurde von der RSerPool-Arbeitsgruppe innerhalb der IETF standardisiert und im Rahmen von mehreren RFCs [3]–[7] schließlich 2008 vollständig spezifiziert.

II. Überblick

RSerPool ist eine Protokollsuite mit zwei Applikationsprotokollen ASAP und ENRP, die den Aufbau eines Serverpools und die Kommunikation mit diesem Serverpool, insbesondere die Auswahl eines Servers, beschreiben. Ein Serverpool wird gebildet von mehreren Servern, die denselben Service anbieten. Die Einsatzmöglichkeiten sind vielfältig und beeinhalten zum Beispiel Load-Balancing von Webservern oder VoIP mit SIP-Proxys. Der Serverpool kann flexibel erweitert bzw. verkleinert werden, erkennt Ausfälle von Servern automatisch und kann auch selbstständig darauf reagieren. RSerPool bietet also eine Alternative zu klassischen Hard-/Software-Load-Balancern. Methoden zum Abgleich von Inhalten sind nicht in dieser Protokollsuite enthalten und werden weiterhin applikationsspezifisch geregelt, genau wie die Kommunikation zwischen Client und Anwendungsserver.

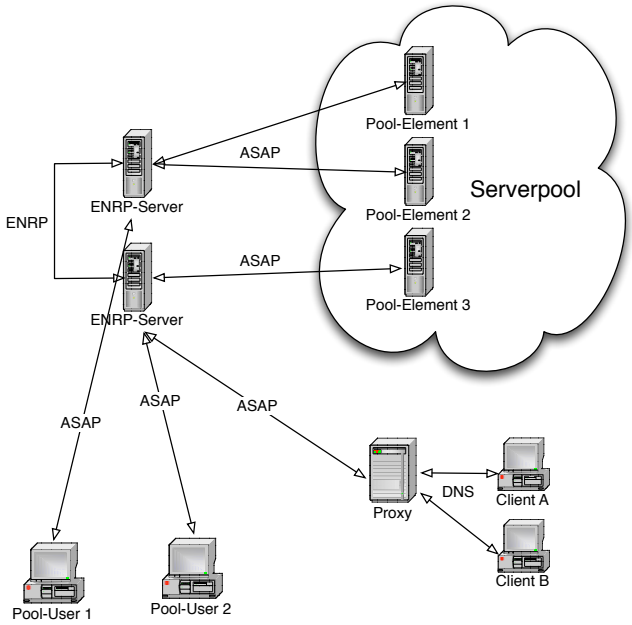

Abbildung 1. Aufbau von RSerPool

Der grundsätzliche Aufbau für RSerPool sieht folgendermaßen aus: Die einzelnen Server eines Serverpools werden als Pool-Elemente (PE) bezeichnet. Jedes PE registriert sich bei einem der ENRP-Server, der die Verwaltung des Serverpools und die Zuweisung von einzelnen PEs an die Clients vornimmt. Die Clients werden in diesem Szenario als Pool-User (PU) bezeichnet. Sowohl die Kommunikation von PE mit ENRP-Server als auch von PU mit ENRP-Server läuft über ASAP. Um Ausfallsicherheit zu gewährleisten, sind auch die ENRP-Server redundant ausgelegt. Sie gleichen sich jeweils mit ENRP-Nachrichten ab.

Nachdem die PEs in einem Serverpool registriert wurden und diesem Serverpool ein sogenanntes Pool-Handle zugewiesen wurde„ können die PUs beim ENRP-Server die Adresse eines PEs erfragen. RSerPool übernimmt dabei die Rolle des klassischen DNS und löst Anfragen auf ein Pool-Handle auf

und teilt eine bzw. mehrere IP-Adressen von dazugehörigen PEs mit.

Da RSerPool auch immer einen Teil der Logik auf den Client auslagert, sind auch Proxys denkbar, die z.B. das klassische Protokoll DNS anbieten, damit auch Clients RSerPool nutzen können, die ASAP noch nicht unterstützen.

III. Nachrichtenformat

A. Überblick

ASAP- und ENRP-Nachrichten sind sehr ähnlich aufgebaut. Alle Felder werden in Network-Byte-Order (d.h. Big-Endian) übertragen. Eine Nachricht folgt folgendem Schema, das an die SCTP-Paketstruktur angelehnt ist [5].

Tabelle I
Nachrichtenaufbau ASAP und ENRP

Feld	Länge
Typ	8 Bit
Flag	8 Bit
Länge	16 Bit
Parameter	variabel

Der Nachrichtentyp wird von einer 8-Bit-Konstante vorgegeben. Das Längenfeld umfasst die komplette Nachricht (inkl. Längenfeld selber). Die mitgegebenen Parameter sind in der Anzahl und im konkreten Typ abhängig vom Nachrichtentyp. Alle Parametertypen haben aber denselben Aufbau.

B. Parameterformat

Tabelle II
Parameteraufbau ASAP und ENRP

Feld	Länge
Typ	16 Bit
Länge	16 Bit
Wert	variabel
Füllbytes	variabel

Alle Parameter bestehen aus einem 16-Bit-Parametertyp, der die Art des übertragenen Parameters angibt. Danach folgt die Länge als 16 Bit unsigned integer. Grundsätzlich umfasst die Länge auch die Größe des Parametertyps und des Längenfeldes, berücksichtigt allerdings nicht die evtl. vorhandenen Füllbytes. Alle Parameter müssen in der Länge ein Vielfaches von 4 Byte sein. Ist das nicht der Fall, wird der Parameter mit (maximal 3) Füllbytes aufgefüllt.

IV. ASAP

A. Überblick

Das Aggregate Server Access Protocol (ASAP) ist für die Kommunikation zwischen PU und ENRP-Server zuständig, sowie für die Kommunikation zwischen PE und ENRP-Server [4]. Für Letzteres muss als Transportprotokoll SCTP genutzt werden, für Ersteres kann optional auch TCP genutzt werden.

Zunächst muss ein ENRP-Server mittels ASAP_SERVER_ANNOUNCE seine ID und seine Transport-Parameter bekanntmachen. Dies geschieht üblicherweise über einen Multicast auf einem festgelegten Kanal.

Zum Registrieren eines PE wird eine ASAP_REGISTRATION-Nachricht verwendet. Dazu wird vom PE einer der bekannten ENRP-Server ausgewählt und diesem als Parameter ein eindeutiges Pool-Handle mitgeteilt. Außerdem werden als Parameter mehrere Informationen über das PE mitgegeben. Der wichtigste ist ein 32-Bit-PE-Identifier, der bei allen weiteren Nachrichten jeweils das PE identifiziert. Außerdem kann angegeben werden, wie lange die Registrierung gültig ist und welcher Algorithmus zur Auswahl eines PEs verwendet werden soll.

Ein PE kann sich explizit vom Pool abmelden. Dazu nutzt er die ASAP_DEREGISTRATION-Nachricht.

Die PUs nutzen die ASAP_HANDLE_RESOLUTION-Nachricht um zu erfahren, welche PEs zu einem Server-Pool mit einem bestimmten Handle gehören. Als Antwort erhalten sie eine Liste mit PE-Parametern, die unter anderem auch die IP-Adresse des PEs umfasst.

B. Fehlerbehandlung

Um einen Ausfall von Pool-Elementen zu erkennen, gibt es die Möglichkeit, dass entweder ein Client einen ENRP-Server explizit mit einer ASAP_ENDPOINT_UNREACHABLE-Nachricht über einen Ausfall informiert, oder dass der ENRP-Server im Zuge seiner regelmäßigen ASAP_ENDPOINT_KEEP_ALIVE-Nachrichten keine Antwort mehr vom PE erhält.

C. Session-Management

ASAP unterstützt erweitertes Session-Management, das optional auch für die Kommunikation zwischen PU und PE genutzt werden kann. Das Session-Management wird hierbei von einer erweiterten API angeboten, im Folgendem RSerPool-Service genannt. Der RSerPool-Service ist in der Lage beim Ausfall eines Servers die Verbindung auf einen anderen Server umzuleiten. Dieses Fail-Over geschieht für den Programmierer und auch Anwender transparent. Die Verbindung zwischen PU und PE muss in dem Fall über den RSerPool-Service aufgebaut werden, und umfasst neben der Datenverbindung mit dem applikationsspezifischen Protokoll für die Anwendungsdaten auch eine separate ASAP-Kontrollverbindung. Über diese zweite Verbindung wird dem PU vom PE beim Beginn einer neuen Session ein Cookie mitgeteilt, mit dessen Hilfe die Session auf anderen PEs wiederhergestellt werden kann. Dies geschieht über die ASAP_COOKIE-Nachricht. Damit das funktioniert, müssen entweder im Cookie selbst alle Sessioninformationen oder ein eindeutiger Identifier enthalten sein, mit dem jedes PE auf die Session zugreifen kann. Die Sessioninformationen selbst müssen in diesem Fall über andere Wege zwischen den PEs eines Pools ausgetauscht werden.

Außerdem werden über diesen Kanal auch ASAP_BUSINESS_CARD-Nachrichten ausgetauscht. Diese dienen dazu, das Pool-Handle zur aktuellen Verbindung zu ermitteln. Nur wenn dieses Pool-Handle bekannt ist, kann beim Ausfall auf ein anderes PE desselben Serverpools gewechselt werden. Damit ein Ausfall von der Anwendung möglichst nicht bemerkt wird, wird auch die Datenverbindung

vom RSerPool-Service gemanagt, damit auch diese beim Serverausfall transparent auf ein anderes PE umgeschaltet werden kann.

V. ENRP

A. Überblick

Das Endpoint HaNdlespace Redundancy Protocol (ENRP) wird von den ENRP-Servern genutzt, um gegenseitig die Daten abzugleichen und so die notwendige Redundanz zu schaffen. Dazu werden entweder andere Peer-ENRP-Server direkt adressiert oder Nachrichten werden über einen vorher festgelegten Multicast-Kanal ausgetauscht. Als Transportprotokoll bei ENRP ist zwingend SCTP vorgeschrieben [8]. Jeder ENRP-Server generiert zu Beginn eine 32-Bit-Server-ID, die ihn eindeutig identifiziert. Normalerweise wird dafür einfach eine Zufallszahl erzeugt. Diese ID ist unveränderlich solange der Server in Betrieb ist.

Danach versucht der ENRP-Server zunächst seinen Mentor-Server zu kontaktieren. Dessen Adresse ist fest vorgegeben. Wenn mehrere Adressen bekannt sind, wird ein beliebiger Server zum Mentor-Server, der über eine der Adressen erreichbar ist. Sobald der ENRP-Server seinen Mentor-Server gefunden hat, kann er von diesem im Rahmen der Initialisierung mittels einer ENRP_LIST_REQUEST eine komplette Liste aller anderen ENRP-Server anfordern.

Um nun außerdem die Informationen über die bekannten Serverpools zu bekommen, werden ENRP_HANDLE_TABLE_REQUEST-Nachrichten genutzt. Als Antwort auf diese Nachrichten werden alle Pool-Element-Parameter nach Pool-Handles geordnet zurückgegeben.

Jedes Pool-Element kommuniziert immer nur mit einem einzigen ENRP-Server. Dieser Server wird Home-ENRP-Server in Bezug auf dieses Pool-Element genannt. Updates einzelner Pool-Elemente in einem Server-Pool werden vom jeweiligen Home-ENRP-Server an alle anderen ENRP-Server mittels einer ENRP_HANDLE_UPDATE-Nachricht verbreitet.

B. Fehlerbehandlung

Jeder Server verwaltet intern eine Liste aller bekannten Peer-Server. Sobald eine ENRP-Nachricht von einem bis dato unbekannten Peer-Server kommt, wird dieser mittels einer ENRP_PRESENCE-Nachricht aufgefordert, sich bei dem anfragenden Server zu identifizieren. Außerdem senden alle Server in regelmäßigem Abstand ebenfalls eine ENRP_PRESENCE-Nachricht an alle ihre Peers. Wenn ein Server diese Nachricht auch nach einer expliziten Aufforderung nicht mehr sendet, werden die Aufgaben des Servers von demjenigen Peer übernommen, der zuerst bemerkt hat, dass der Server nicht mehr antwortet. Damit nicht mehrere Server gleichzeitig versuchen, die Aufgaben vom toten Peer zu übernehmen, wird zunächst eine ENRP_INIT_TAKEOVER-Nachricht verschickt, die von allen anderen ENRP-Servern bestätigt werden muss. Erst dann beginnt die Übernahme des nicht mehr antwortenden Servers. Dazu wird jedes Pool-Element, benachrichtigt, das den übernommenen Server als Home-ENRP-Server genutzt hat. Dies geschieht über ASAP_ENDPOINT_KEEP_ALIVE-Nachrichten. Alle anderen ENRP-Server streichen den übernommenen ENRP-Server von ihrer internen Liste.

VI. Regeln zur Serverauswahl

Um die Last zwischen den Pool-Elementen möglichst optimal zu verteilen, existieren unterschiedliche Vorgaben zur Serverauswahl, die im RFC5356 spezifiziert sind [7]. Allen Vorgaben ist gemein, dass die Serverauswahl aufgesplittet wird zwischen PU und ENRP-Server. Der ENRP-Server gibt zunächst eine Liste von in Frage kommenden Servern zurück, die er mithilfe einer Regel ausgewählt hat. Der PU wählt dann aus dieser Liste wiederum einen einzigen Server aus. Diese doppelte Auswahl hat den Vorteil, das bei Ausfall eines Servers nicht erneut der ENRP-Server kontaktiert werden muss. In einem solchen Fall kann vom Client aus seiner Serverliste einfach ein anderer Server ausgewählt werden [9].

Grundsätzlich wird zwischen nicht-adaptiven und adaptiven Techniken unterschieden.

A. Nicht-adaptive Verfahren

Zu den nicht-adaptiven Verfahren gehört Round-Robin, das die einzelnen PEs reihum an die PUs verteilt. Haben alle PEs bereits einen PU erhalten, geht es wieder beim ersten Server los. Eine Verbesserung dieses Verfahrens stellt die Gewichtung der einzelnen PEs aufgrund ihrer unterschiedlichen Kapazität dar [10]. Diese Gewichtung ist statisch, und muss nur einmal bei der Registrierung den ENRP-Servern mitgeteilt werden. Bei beiden Verfahren wird vom ENRP-Server jeweils eine vollständige Liste von PEs an den PU übermittelt. Dieser ermittelt dann mithilfe von Round-Robin einen Eintrag. Damit nicht jeder PU denselben Server auswählt, ist bei jeder Anfrage jeweils das nächste PE am Anfang der Liste. Diese Verfahren erfordern deshalb einen Status, in dem gespeichert wird, welches PE zuletzt ausgewählt wurde bzw. am Anfang der Liste stand. Standen alle PEs einmal am Anfang der Liste, wird wieder mit dem ersten PE begonnen. Die Liste enthält PEs mit höherem Gewicht entsprechend mehrmals, so dass diese häufiger ausgewählt werden, als diejenigen mit niedrigerem Gewicht. Die mehrfachen Servereinträge werden möglichst mit äquidistantem Abstand zueinander in der Liste eingefügt. Die relative Häufigkeit der Einträge pro PE entspricht dabei deren relativem Gewicht.

Ein zustandsloses Verfahren stellt das Zufallsverfahren (RAND) dar, das ebenso in einer gewichteten Ausprägung spezifiziert ist. Es funktioniert ähnlich wie Round-Robin, allerdings erfolgt jede Auswahl unabhängig von der vorherigen. Deshalb muss weder der ENRP-Server noch der PU einen Status vorhalten. Der ENRP-Server gibt eine Liste von PEs zurück, in denen kein PE doppelt vorkommt. Die Auswahl dieser Liste erfolgt bei jeder Anfrage neu und die Wahrscheinlichkeit, dass ein PE in der Liste landet, muss gleich dessen relativem Gewicht sein. Aus dieser Liste wiederum wählt der PU zufällig ein Element aus.

Als letztes nicht-adaptives Verfahren ist ein Prioritätenverfahren spezifiziert, bei dem vom ENRP-Server immer eine

Liste von PEs Reihenfolge ihrer Priorität zurückgegeben wird. Die Priorität der PEs wird dabei bei deren Registrierung einmal festgelegt. Die PUs wählen immer den ersten Eintrag, d.h. den Eintrag mit der höchsten Priorität aus. Dieses Verfahren kann genutzt werden, damit standardmäßig immer ein PE angesprochen wird. Erst im Fehlerfall wird das PE mit der nächstniedrigeren Priorität genutzt.

B. Adaptive Verfahren

Die zweite Gruppe der Verfahren passt die Serverauswahl an bestimmte, sich ändernde Rahmenbedingungen an, oft an die Auslastung der einzelnen PEs. Diese melden in regelmäßigen Abständen ihre aktuelle Auslastung über Registrierungsnachrichten an die ENRP-Server, die diese Informationen bei der Auswahl berücksichtigen.

Die Least-Used-Policy ist ein solches Verfahren, bei der die ENRP-Server eine Liste von Servern mit geringer Last an den PU melden. Dieser wählt dann aus der Liste denjenigen Server mit der geringsten Last aus. Haben mehrere PEs die gleiche Auslastung gemeldet, wird ein PE nach Round-Robin ausgewählt.

Die Aktualisierung der Auslastung erfolgt allerdings sehr selten, so dass die Lastinformationen meistens nicht aktuell sind. Darum wurde ein Verfahren entwickelt, bei dem die Auswahl eines PEs automatisch die Auslastung des ausgewählten PEs um einen konstanten Wert hebt. Diese Least-Used-With-Degradation-Policy (LUD) funktioniert nur auf ENRP-Server-Seite. Die PUs wählen einfach den ersten Eintrag aus der Liste aus, ohne dass das Einfluss auf die Auslastungsinformationen hätte. Wenn beim ENRP-Server wieder eine Aktualisierung der Auslastung in Form einer Registrierungsnachricht eintrifft, wird der lokale Cache mit Auslastungsinformationen wieder überschrieben.

Bei diesem Verfahren wird allerdings nicht berücksichtig, wie stark die Last steigt, wenn ein Server ausgewählt werden würde. Deshalb wurde die Priority-Least-Used-Policy aufgenommen, bei der die Server sortiert werden nach der Reihenfolge ihrer Auslastung, die sie im Falle einer Anfrage hätten. Durch das Verfahren kann also die Last noch besser verteilt werden.

Daneben existiert noch das Randomized-Least-Used-Verfahren (RLU) das dem Zufallsverfahren gleicht, allerdings die Last mit in die Wahrscheinlichkeit der Auswahl einfließen lässt.

Eine Bewertung der Verfahren von Thomas Dreibholz hat ergeben, dass die adaptiven Verfahren wie erwartet, besser mit sich stark ändernden Bedingungen umgehen können [11]. Bei hoher Netzwerk-Latenz verschwinden diese Vorteile jedoch und die Ergebnisse nähern sich denen der nicht-adaptiven Verfahren an, da auch die adaptiven Verfahren nicht schnell genug auf Änderungen reagieren können . Ebenso wurde dort gezeigt, dass lokales Caching von PEs bei den PUs selten sinnvoll ist, da dann die ENRP-Server die Last nicht mehr gleichmäßig verteilen können.

VII. SICHERHEITSASPEKTE

ASAP und ENRP bieten für sich genommen keine Sicherheit in der Kommunikation. Darum wurden im RFC 5355 [6] Möglichkeiten beschrieben, wie bestimmte Angriffsszenarien zu unterbinden sind. Verpflichtend ist die Nutzung von Transport Layer Security (TLS) mit einer AES128-Verschlüsselung für alle Verbindungen. Sowohl PEs als auch ENRP-Server müssen sich gegenseitig mithilfe eines gemeinsamen geheimen Schlüssels (PSK) authentifizieren. PUs müssen ENRP-Server mithilfe von Zertifikaten authentifizieren. TLS muss auf Basis von SCTP unterstützt werden, wie es im RFC3436 spezifiziert ist [12].

Mit dieser Maßnahme kann verhindert werden, dass beliebige Rechner Registrierungs- und Deregistrierungsnachrichten verschicken. Auch ein Statusupdate erfolgt erst dann, wenn das PE authentifiziert wurde. Damit werden Angriffe von nicht-authentifizierten Rechnern wirkungslos. Auch ein neuer ENRP-Server kann nicht einfach eingeschleust werden, da sich auch die Peer-ENRPs zunächst gegenseitig authentifizieren. Man-in-the-Middle-Attacken zwischen ENRP-Server und Pool-User werden durch Zertifikate in Kombination mit Verschlüsselung verhindert.

PUs selber nutzen keine Zertifikate und dementsprechend muss bei der Kommunikation vom ENRP-Servern darauf geachtet werden, dass Meldungen von PUs nicht unbedingt vertrauenswürdig sind. Auch ein Flooding von ASAP_ENDPOINT_UNREACHABLE-Nachrichten wird verhindert, indem die ENRP-Server nur eine bestimmte Anzahl von diesen Nachrichten innerhalb einer Zeitspanne von einem PU zulassen. Jede dieser Nachrichten wird außerdem noch vom ENRP-Server verifiziert.

Im Gegensatz zum DNS wurde also bei RSerPool von Anfang an darauf geachtet, dass die Protokolle besonders sicher sind. Durch die Nutzung von TLS ist das besonders einfach, da die darüberliegende Schicht von ASAP/ENRP davon so gut wie nichts mitbekommt.

VIII. SCTP

SCTP ist ein Transportprotokoll auf Schicht 4 des OSI-Modells. Es wurde im RFC4960 [13] spezifiziert. Das Protokoll wurde entwickelt, um einen besonders zuverlässigen Transport von Telefonsignalen über IP zu ermöglichen. Deshalb wurde besonders viel Wert auf Verfügbarkeit gelegt. Aus diesem Grund ist das Protokoll für RSerPool als Transportprotokoll vorgeschrieben.

SCTP ähnelt stark TCP, verhindert allerdings bestimmte Angriffsszenarien wie z.B. SYN-Flooding durch einen 4-Wege-Handshake. Der Header von SCTP besteht aus Quell- und Zielport sowie einer CRC32-Checksumme. Außerdem enthält er ein Verifikationstag, mit dem der Absender sich authentifiziert. Dieses Tag wird initial beim Handshake ausgetauscht und wird danach in allen Paketen die zu der Verbindung gehören mitübertragen.

Die Inhalte werden in Datenblöcken (sogenannten Chunks) übertragen, die wiederum aus einem Header bestehen, der den Typ, einige Flags und die Länge des Chunks spezifiert. Hinter

Tabelle III
SCTP-PACKET-FORMAT

Feld	Länge
Quellport	16bit
Zielport	16bit
Verifikationstag	32bit
Checksumme	32bit
Chunktyp	8bit
Chunkflags	8bit
Chunklänge	16bit
Chunkinhalt	variabel
Chunk 2 - n	variabel

dem Header stehen die eigentlichen Daten. Chunks werden dabei nicht nur für die Datenübertragung genutzt, sondern auch für den Verbindungsauf- und abbau. Dafür sind bestimmte Chunk-Typen reserviert.

Durch die Unterteilung in Chunks ist es möglich, dass mehrere Applikationen sich eine Verbindung teilen (Multi-Streaming). Optional ist es möglich die Chunks zu nummerieren, so dass sie in der gleichen Reihenfolge beim Empfänger ankommen, wie sie losgeschickt wurden. Für andere Anwendungen (beispielsweise Echtzeit-Anwendungen) ist es allerdings wichtiger, dass die Pakete möglichst schnell ankommen. Darum kann SCTP auch als Alternative zu UDP genutzt werden.

Der Verbindungsaufbau ist im Gegensatz zu TCP vierstufig, so dass ein SYN-Flooding-Angriff unmöglich ist. Während des Handshakes wird kein Status auf dem Server gehalten, sondern nur in einem Cookie zwischen Client und Server ausgetauscht. Der Client initiiert die Verbindung mit einem INIT, auf dass der Server mit einem INIT-ACK antwortet. In dieser Nachricht wird bereits ein Cookie übermittelt, das der Client mithilfe der COOKIE-ECHO-Nachricht wiederum an den Server verschicken muss. Erst dann werden die Resourcen für die Verbindung vom Server reserviert und dieser antwortet mit einem COOKIE-ACK. Damit der Verbindungsaufbau den Datenfluss nicht zu sehr verzögert, ist es bereits in den letzten beiden Nachrichten möglich Daten mitzuschicken.

Zur Flusskontrolle nutzt SCTP wie TCP eine adaptive Fenstergröße und zur Fehlerkontrolle selektive Bestätigungen (Sel ACK). Fast Retransmission ist im Gegensatz zu TCP nicht mehr optional, sondern vorgeschrieben.

SCTP unterstützt das Multihoming, bei dem Endpunkte über mehrere IP-Adressen erreichbar sind (beispielsweise ein Server mit mehreren Ethernet-Schnittstellen, die unterschiedlich geroutet werden). Davon wird üblicherweise zwar nur eine genutzt, um die Auslastung der Netze gering zu halten, beim Ausfall dieser Verbindung kann aber auf die andere Schnittstelle gewechselt werden, ohne dass die Verbindung von der Anwendung neu aufgebaut werden muss. Im Gegensatz zu TCP sind bei SCTP auch One-To-Many-Verbindungen möglich, die im Zusammenhang mit ENRP verwendet werden, um gleichzeitig alle anderen ENRP-Server zu adressieren.

Fast alle Unix-Betriebssysteme unterstützen SCTP, allerdings ist das Protokoll weder in Windows Vista noch in Max OS X enthalten. Für dieses Betriebssystem gibt es aber Implementierungen von Fremdanbietern [14].

IX. ZUSAMMENFASSUNG UND AUSBLICK

RSerPool ist noch eine sehr neue Protokollsuite, die bis jetzt noch keine weite Verbreitung außerhalb des akademischen Umfelds gefunden hat. Insbesondere der obligatorische Einsatz von SCTP als Transportprotokoll stand einem Einsatz bis jetzt im Wege. RSerPool stellt aber häufig einen kostengünstigeren und flexibleren Ansatz als klassische Load-Balancer dar und wird deshalb gerade im Umfeld von Webservices seine Anwendung finden.

Kritisch zu hinterfragen ist jedoch die Zeitverzögerung der Adressauflösung durch RSerPool gegenüber dem DNS bzw. direkter IP-Adressierung. Zwar ist das System von der Anzahl der Nachrichten mit dem DNS zu vergleichen, allerdings ist es im Gegensatz zum klassischen DNS aufgrund der Lastverteilung häufig nicht sinnvoll, ASAP-Adressauflösungen zu cachen. Auch das 4-Way-Handshake der SCTP-Verbindung verzögert den Verbindungsaufbau, verglichen mit den DNS-Anfragen über UDP.

Mit rsplib steht schon eine leistungsfähige Implementierung unter der GPL bereit und auch Motorola hat bereits eine Closed-Source Implementierung entwickelt. Für das normale Web-Umfeld wird DNS in Kombination mit klassischen Load-Balancern wohl das Mittel der Wahl bleiben, da nicht mit einer baldigen Implementierung von ASAP in den Webbrowsern zu rechnen ist.

LITERATUR

[1] M. Tuexen, Q. Xie, R. Stewart, M. Shore, L. Ong, J. Loughney, and M. Stillman, "Requirements for Reliable Server Pooling," RFC 3237 (Informational), Jan. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3237.txt

[2] M. T. T. Dreibholz, "High Availability using Reliable Server Pooling," in *Linux Conference Australia (LCA'2003)*, Januar 2003. [Online]. Available: http://tdrwww.iem.uni-due.de/dreibholz/rserpool/rserpool-publications/RSerPool-Paper.pdf

[3] P. Lei, L. Ong, M. Tuexen, and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols," RFC 5351 (Informational), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5351.txt

[4] R. Stewart, Q. Xie, M. Stillman, and M. Tuexen, "Aggregate Server Access Protocol (ASAP)," RFC 5352 (Experimental), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5352.txt

[5] ——, "Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) Parameters," RFC 5354 (Experimental), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5354.txt

[6] M. Stillman, R. Gopal, E. Guttman, S. Sengodan, and M. Holdrege, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats," RFC 5355 (Informational), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5355.txt

[7] T. Dreibholz and M. Tuexen, "Reliable Server Pooling Policies," RFC 5356 (Experimental), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5356.txt

[8] Q. Xie, R. Stewart, M. Stillman, M. Tuexen, and A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)," RFC 5353 (Experimental), Sep. 2008. [Online]. Available: http://www.ietf.org/rfc/rfc5353.txt

[9] T. Dreibholz, "Das rsplib-Projekt - Hochverfügbarkeit mit Reliable Server Pooling," in *LinuxTag 2005, Karlsruhe*, Jun. 2005. [Online]. Available: http://tdrwww.iem.uni-due.de/dreibholz/rserpool/rserpool-publications/LinuxTag2005.pdf

[10] M. T. T. Dreibholz, E. P. Rathgeb, “Load Distribution Performance of the Reliable Server Pooling Framework,” in *IEEE International Conference on Networking (ICN 2005)*, April 2005. [Online]. Available: http://tdrwww.iem.uni-due.de/dreibholz/rserpool/rserpool-publications/ICN2005.pdf

[11] E. P. R. Thomas Dreibholz, “On the performance of reliable server pooling systems,” in *30th IEEE Local Computer Networks Conference*, November 2005. [Online]. Available: http://tdrwww.iem.uni-due.de/dreibholz/rserpool/rserpool-publications/LCN2005.pdf

[12] A. Jungmaier, E. Rescorla, and M. Tuexen, “Transport Layer Security over Stream Control Transmission Protocol,” RFC 3436 (Proposed Standard), Dec. 2002. [Online]. Available: http://www.ietf.org/rfc/rfc3436.txt

[13] R. Stewart, “Stream Control Transmission Protocol,” RFC 4960 (Proposed Standard), Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4960.txt

[14] R. Stewart, “SCTP Implementations.” [Online]. Available: http://www.sctp.org/implementations.html

Wuala

Seminar Future Internet SS2009

Florian Wohlfart
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: wohlfart@in.tum.de

***Zusammenfassung*— Wuala ist ein Cloud-Storage-Dienst, der auf einem Peer-to-Peer-Netzwerk als verteiltem Speicher basiert. Die Wuala-Entwickler haben es geschafft die aufwendige Technik ihres Peer-to-Peer-Netzwerks hinter einer einfach zu benutzenden Oberfläche zu verstecken, die wie ein Dateimanager aufgebaut ist. Die Benutzeroberfläche ist jedoch gerade für Vielnutzer zu wenig anpassbar und lässt sich nicht so schnell wie ein gewöhnlicher Dateimanager bedienen. Wuala wirbt damit dass die privaten Dateien in Wuala so sicher seien, dass sie angeblich nicht mal von Wuala selbst entschlüsselt werden könnten. In der Tat ist Wuala relativ sicher, solange man eigene Dateien hochlädt, die sonst niemand besitzt und ein sicheres Passwort wählt. Im Vergleich mit anderen Cloud-Storage-Diensten ist das Konzept von Wuala durchaus konkurrenzfähig und hebt sich durch einzigartige Features von der Masse ab.**

***Schlüsselworte*— Wuala, Peer-to-Peer, Cloud-Storage, Online-Backup, File-Sharing, Soziales Netzwerk**

I. Einleitung

A. Einführung in Wuala

Cloud-Storage-Dienste [1] sind eine relativ neue Entwicklung des Internets, welche es dem Benutzer ermöglichen seine Daten ständig zur Verfügung zu haben, sofern ein Internetzugang vorhanden ist. Wuala bietet hierbei den neuen, ungewöhnlichen Ansatz die Daten in einem Peer-to-Peer-Netzwerk zu speichern. Die Idee ist, dass Anwender auf ihrem Rechner Speicherplatz freigeben, auf dem Wuala Dateien von anderen Nutzern speichern darf. Im Gegenzug dazu bekommt der Nutzer Online-Speicherplatz im Wuala-Netzwerk. Man kann also lokalen Speicherplatz gegen Online-Speicherplatz eintauschen und so die Qualität seines Speichers verändern.
Dateien können in Wuala geheim gehalten werden, mit Freunden geteilt oder öffentlich zugänglich gemacht werden. Um das Geheimhalten von Daten im Peer-to-Peer-Netzwerk zu ermöglichen, werden die Daten verschlüsselt. Zur Verwaltung von privaten und öffentlichen Daten wurde eigens ein System zur Schlüsselverwaltung entwickelt. Auch wurde die Wuala-Software speziell auf den Umgang mit großen Multimedia-Dateien ausgerichtet.
Die Wuala-Software bekommt außerdem durch Features wie dem Bilden von Freundschaften unter Benutzern, erstellen von Gruppen, teilen von Dokumenten mit Freunden und dem Kommentieren von Dateien einen Community-Charakter. Darurch umfasst Wuala mehr als nur einen Online-Speicherplatz, es ausserdem ist ein soziales Netzwerk und kann als Ersatz für One-Click-Hoster dienen.
Das Konzept von Wuala baut darauf auf, dass viele Nutzer nur einen kleinen Teil ihrer zur Verfügung stehenden Ressourcen wie Speicherplatz und Bandbreite nutzen. Da die meisten Nutzer inzwischen eine Internet-Flatrate besitzen, verursacht Wuala bei ihnen keine zusätzlichen Kosten, bietet aber einen Mehrwert in Form von stets zugreifbarem Internet-Speicher.

B. Überblick

In dieser Arbeit soll nun zuerst die Funktionsweise von Wuala näher betrachtet werden, um die späteren Ausführungen zu verstehen. Hierbei werden einige effiziente und elegante Lösungen erklärt, wie etwa der Einsatz von Erasure Codes zur redundanten Speicherung der Dateien und das neu entwickelte System zum Schlüsselmanagement. Als nächstes wird die Benutzerfreundlichkeit der Benutzeroberfläche untersucht, wobei der Fokus auf effizientem Arbeiten und der multimediatauglichkeit von Wuala liegen. Darauf folgt der Kern dieser Arbeit, die Sicherheit von Wuala. Dazu wird untersucht in wieweit Wuala die IT-Schutzziele erfüllt. Im Anschluss folgt ein Vergleich mit ähnlichen Produken, sowie ein Fazit über die Software.

II. Funktionsweise

Da Wuala eine Reihe neuer, interessanter Ideen und Algorithmen beinhaltet wird hier auf die Technik und Funktionsweise von Wuala eingegangen, um die Software besser zu verstehen.

A. Redundante Speicherung

Da in einem Peer-to-Peer-Netzwerk nicht ständig alle Clients online sind, wird jede Datei in Wuala mehrfach redundant gespeichert, um eine hohe Verfügbarkeit garantieren zu können. Um diese Redundanz zu erzeugen gibt es zwei Möglichkeiten: die Vervielfältigung der Datei und die Vervielfältigung von Dateifragmenten durch Erasure Codes [2]. Da die Verfügbarkeit einer Datei beim Einsatz von Erasure Codes im Vergleich zur simplen Vervielfältigung der Datei um Größenordnungen höher ist [3], kommen in Wuala Erasure Codes zum Einsatz, die auf Basis von Dateifragmenten arbeiten.

1) Erasure Codes: Angenommen eine Datei besteht aus n Fragmenten. Dann werden mit einem Erasure Code aus den n Fragmenten weitere m redundante Fragmente errechnet. Nun kann aus einer Untermenge der insgesamt $n + m$ Fragmente

wieder die vollständige Datei wiederherstellt werden, auch wenn ein paar der Fragmente fehlen. Wuala verwendet als Erasure Code den Reed-Solomon Code [4], welcher in der Lage ist aus beliebigen n der insgesamt $n+m$ Fragmente die vollständige Datei zu rekonstruieren. Eine Datei wird immer in $n = 100$ Fragmente aufgeteilt [5], die Größe eines Fragments hängt also von der Dateigröße ab. Die Anzahl der redundanten Fragmente wird durch den Redundanz-Faktor festgelegt, der unter anderem von der durchschnittlichen Online-Zeit der Rechner, auf denen die Fragmente gespeichert werden, und der Beliebtheit der Datei abhängt. Der Redundanz-Faktor beträgt in typischen Fällen 4, das heisst es werden ungefähr $m = 400$ redundante Fragmente gebildet. Somit werden insgesant ungefähr $n + m = 500$ Fragmente einer Datei im Netzwerk gespeichet. [3].

2) Hochladen von Dateien: Beim Hochladen einer Datei ins Wuala-Netzwerk geschieht also folgendes: Zuerst wird die Datei verschlüsselt. Dann wird sie in n Fragmente aufgeteilt und es werden m redundante Fragmente hinzugefügt. Schliesslich werden alle $n+m$ Fragmente ins Wuala-Netzwerk geladen, wobei ein Rechner nur höchstens ein Fragment einer Datei speichert, um die Datei gut zu verteilen. Zusätzlich werden die ersten n Fragmente der Datei auf den Wuala-Server geladen, welcher die ständige Verfügbarkeit der Datei sicherstellen soll und als Backup-Server dient.

3) Herunterladen von Dateien: Soll eine Datei aus dem Wuala-Netzwerk heruntergeladen werden, so versucht der Wuala-Client n beliebige Fragmente der Datei gleichzeitig aus dem Wuala-Netzwerk zu laden. Fehlen noch Fragmente, da sie momentan im Netzwerk nicht verfügbar sind, werden diese vom Wuala-Server geladen. Durch die vielen parallelen Downloads bringt diese Variante einen spürbaren Geschwindigkeitsvorteil gegenüber einem normalen Download vom Server. Nach dem Download wird die Datei aus den Fragmenten wiederhergestellt und entschlüsselt. Jetzt kann die Datei gelesen werden.

4) Wartung von Dateien: Verlässt ein Rechner das Wuala-Netzwerk dauerhaft, so gehen alle auf ihm gespeicherten Fragmente verloren. Deshalb überprüft der Wuala-Client regelmässig, ob noch genug Fragmente jeder Datei im Netzwerk vorhanden sind. Fehlt ein Fragment dauerhaft, so wird es automatisch neu berechnet und hochgeladen. Dies wird Wartung der Datei genannt.

B. Schlüsselmanagement

Da kein vorhandenes System zur Schlüsselverwaltung und Zugangskontrolle den Anforderungen der Wuala-Entwickler entsprach, entwickelten sie eine eigene Datenstruktur zum Schlüsselmanagement namens ”Cryptree” [6]. Laut den Wuala-Entwicklern ist Cryptree die erste kryptografische Datenstruktur, welche die neuesten Forschungsergebnisse im Bereich der kryptografischen Schlüsselhierarchien mit denen der Zugangskontrolle in Dateisystemen verbindet.

1) Kryptografische Links: Um den Aufbau des Cryptrees zu verstehen muss man wissen, was kryptografische Links sind. Kryptografische Links stellen einen Zusammenhang zwischen

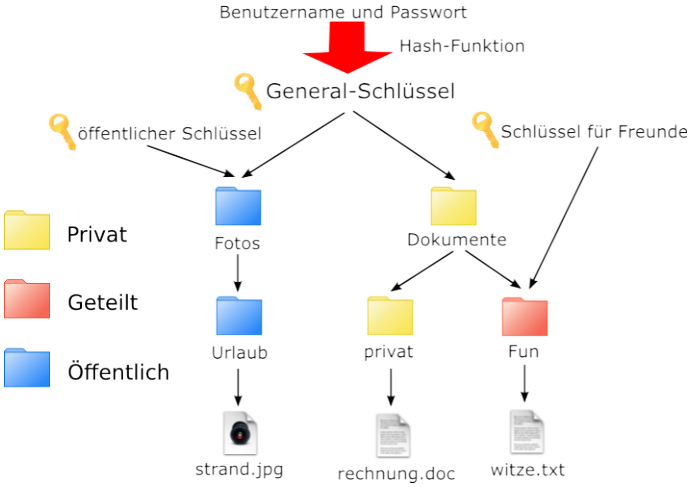

Abbildung 1. Beispielstruktur eines Cryptrees zum Lesezugriff. Jeder Ordner und jede Datei besitzt einen eigenen Schlüssel. Die Pfeile stehen für gerichtete kryptografische Links. In Richtung der Pfeile kann die Ordnersruktur entschlüsselt werden.

zwei verschiedenen Schlüsseln K_1 und K_2 her, der es erlaubt von K_1 auf K_2 zu schliessen, jedoch nicht von K_2 auf K_1. Man schreibt $K_1 \rightarrow K_2$. Dieser Link lässt sich einfach erzeugen indem man K_2 mit K_1 verschlüsselt, wobei man ein symmetrisches oder asymmetrisches Verschlüsselungsverfahren verwenden kann. Nun kann jeder Besitzer von K_1 diesen Link entschlüsseln und so K_2 erhalten. Verwendet man ein asymmetrisches Schlüsselpaar um den Link zu erstellen, so benutzt man den öffentlichen Schlüssel und den geheimen Schlüssel um den Link zu entschlüsseln. Das hat den Vorteil, dass bei einer Änderung von K_2 nur der öffentliche Teil von K_1 bekannt sein muss, um den Link neu zu erstellen. Asymmetrisch verschlüsselte Kryptografische Links werden jedoch Aufgrund der Länge eines asymmetrischen Schlüssels nur an wenigen Stellen benutzt.

2) Cryptree: Der Cryptree besteht aus zwei baumähnlichen Verkettungen aus kryptografischen Links, die eine Ordnerstruktur nachbilden. Ein Baum regelt den Lese- und der andere den Schreibzugriff. Jeder Ordner und jede Datei besitzen ihren eigenen Schlüssel. Diese Schlüssel werden wie in Abbildung 1 gezeigt durch Kryptografische Links verbunden. Die Anordnung der gerichteten kryptografischen Links macht es möglich, jeweils die darunterliegenden Schlüssel zu entschlüsseln, jedoch nicht die darüberliegenden. Es existieren zwar sogenannte Backlinks zum jeweils darüberliegenden Ordner, mit diesen ist es jedoch nur möglich den Namen des Ordners zu entschlüsseln, um den aktuellen Pfad herauszufinden. Damit der Nutzer auf seine eigenen Daten zugreifen kann benötigt er den Generalschlüssel, der die Wurzel der Baumstruktur bildet. Da Wuala aus Sicherheitsgründen diesen Generalschlüssel nicht speichert, wird dieser durch eine Hashfunktion aus dem Benutzernamen und dem Passwort des Benutzers gebildet. Somit kann Wuala nur mit dem richtigen Passwort den Generalschlüssel berechnen. Will man einen Ordner mit Freunden teilen oder veröffentlichen, so generiert Wuala einen neuen

Schlüssel und einen Kryptografischen Link von diesem auf den Schlüssel des Ordners. Diesen neuen Schlüssel schickt man nun seinen Freunden bzw. veröffentlicht ihn. Als Konsequenz davon ist es nicht möglich eine Datei als öffentlich oder privat zu deklarieren, sondern nur ganze Ordner. Weiterhin ist des nicht möglich innerhalb eines öffentlichen Ordners private oder nur mit Freunden geteilte Unterordner zu erstellen. Diese intuitive Handhabung vereinfacht jedoch die Verwaltung der Zugriffsregeln und verhindert eine Zersplitterung von zugreifbaren Dateien.

C. Routing

Anfragen werden im Wuala-Netzwerk mit Hilfe einer verteilten Hash-Tabelle geroutet, das heisst das Routing ist dezentralisiert und gut skalierbar. Wuala benutzt ein eigenes Routingverfahren, welches auf Kademlia [7] aufbaut und somit den Zielhost innerhalb von $O(logn)$ Hops findet. Es gibt im Wuala-Netzwerk drei Klassen von Hosts [3]. Hat man keinen Speicher auf seinem Rechner freigegeben, so ist dieser ein ”client node”, das heisst seine Funktion im Wuala-Netzwerk ist ausschliesslich der Datenkonsum. Hat man auf seinem Rechner Speicher freigegeben, so gilt dieser als ”storage node”. Storage nodes speichern Dateifragmente, die von anderen Nutzern abgerufen werden können. Hinzu kommt eine geringere Anzahl an ”super nodes”, die für das Routing verantwortlich sind. Jedem storage- und client node ist ein solcher super node zugewiesen, der dessen Pakete über das Wuala-Netzwerk routet. Super nodes besitzen eine Routing-Tabelle mit den Adressen von anderen super- und storage nodes, um Anfragen weiterleiten zu können. In der Routing-Tabelle befinden sich die benachbarten super nodes ebenso wie zufällige andere super nodes und die dem super node zugewiesenen client und storage nodes. Die Kombination von benachbarten und zufälligen Einträgen in der Routing-Tabelle hat sich in Tests als besonders effizient erwiesen [3].

D. Fairness

Ebenfalls eine Eigenentwicklung des Wuala-Teams ist ”Havelaar” [8], ein robustes System zur Bewertung des Nutzerverhaltens in Peer-to-Peer-Netzwerken. Verbraucht ein Benutzer sehr viele Ressourcen, trägt aber selbst nichts zum Netzwerk bei, so soll mit dem System seine seine verfügbare Bandbreite herabgesetzt werden. Man will jedoch die Bandbreite bei solchen Benutzern nicht künstlich begrenzen, sondern gibt stattdessen anderen Benutzern den Vortritt, wenn mehrere Benutzer gleichzeitig beim gleichen storage node nach einem Dateifragment fragen. Dieses Bewertungssystem soll also keine Nutzer bestrafen, sondern nur eine Belastung des Wuala-Netzwerkes durch Benutzer, die sehr viel Bandbreite beanspruchen, verhindern.

ver

III. Benutzerfreundlichkeit

Nach der Einführung in die Funktionsweise von Wuala soll nun die Benutzerfreundlichkeit und Praxistauglichkeit von Wuala untersucht werden. Da ich unter Linux arbeite beziehen sich alle Beobachtungen auf die Linux-Version von Wuala. Die Linux-Version könnte sich, vor allem weil es sich um eine Beta-Version handelt, von denen auf anderen Plattformen unterscheiden.

A. Der Java-Client

Die Wuala-Entwickler legen Wert auf eine intuitive Benutzeroberfläche und haben es geschafft die aufwendige Netzwerk-Technologie hinter einer einfachen Benutzeroberfläche zu verbergen, so dass der Nutzer fast nicht merkt, dass die Daten in einem Netzwerk gespeichert werden. Aufgrund der Ähnlichkeit mit einem Dateimanager findet man sich schnell zurecht und die meisten Funktionen sind selbsterklärend. Private, geteilte und öffentliche Ordner lassen sich durch verschiedene Farben gut unterscheiden. Zudem lassen sich Dateien per Drag&Drop herunter- beziehungsweise hochladen. Sehr gut gelungen ist auch die Integration von Multimedia-Inhalten. Große Ordner mit vielen Fotos werden schnell geladen und jedes Foto als Thumbnail präsentiert. Öffnet man eine Datei, so wird sie automatisch im passenden Programm geöffnet. Ausserdem gibt es eine Streaming-Funktion für große Dateien, wie zum Bespiel Videos. Ein nettes Feature ist eine Funktion zur automatischen Größenänderung von Fotos vor dem Upload.
Das Stöbern im öffentlichen Welt-Teil der Software, wo alle als öffentlich markierten Dateien gesammelt werden, ist nichts besonderes, da sich dort zur Zeit fast nur aus dem Internet kopierte Inhalte befinden. Sehr praktisch ist jedoch die Idee des Schweizer Fernsehens (SF) einen Teil ihrer Sendungen in Wuala anzubieten [9]. Für Vielnutzer ist die Benutzeroberfläche von Wuala jedoch zu wenig anpassbar. Die Bedienung ist nur über die Maus möglich, man kann nicht per Tastatur navigieren. Deshalb möchte man auf Dauer lieber mit seinem gewohnten Dateimanager arbeiten, was durch Wualas Integration ins Dateisystem möglich ist. Leider besitzt Wuala keine automatische Backup-Funktion, wodurch es nicht geeignet ist, um seine Daten zu sichern.

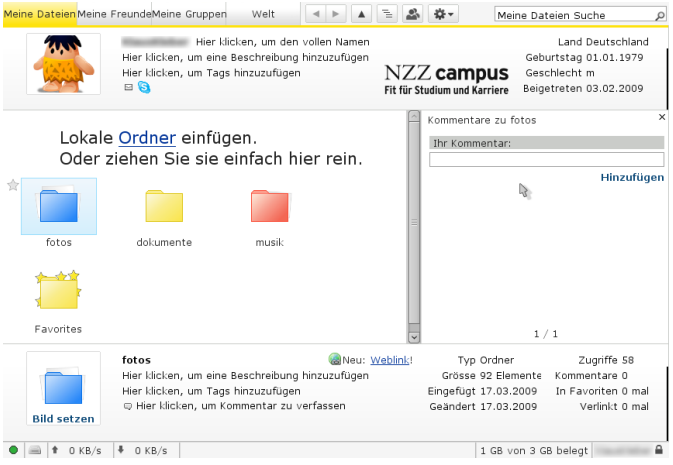

Abbildung 2. Bildschirmfoto des Wuala-Client

B. Dateisystemintegration

Die Dateisystemintegration erfolgt unter Linux durch einbinden eines NFS-Laufwerks. Somit hat man problemlosen Zugriff auf seine Wuala-Dateien und kann sie mit seinem eigenen Dateimanager verwalten. Die Integration ist jedoch nicht optimal gelöst. So kann man im Dateimanager nicht erkennen ob ein Ordner privat, geteilt oder öffentlich ist. Durch die Integration ins Dateisystem können auch andere Programme auf Dateien in Wuala zugreifen, wie zum Beispiel ein Skript zum Online-Backup wichtiger Dateien.

C. Zugriff über die Wuala-Webseite

Die Online-Schnittstelle zur Verwaltung der Dateien ist gut gelungen. Die Webseite ist schlicht gehalten und einfach zu benutzen. Sehr praktisch ist, dass hochgeladene Fotos automatisch als Bildergalerie angezeigt werden. Auch die Möglichkeit private Links zu verschicken, um jemand der keinen Wuala-Account hat den Online-Zugriff auf ausgewählte Dateien zu gewähren ist nützlich und ersetzt lästige One-Klick-Hoster. Ich musste jedoch feststellen, dass die Seite vor allem bei der Anzeige von Fotos relativ langsam lädt. Auch ist es per Design nicht möglich Dateien über ein Web-Formular hochzuladen, da die Dateien vor dem Upload verschlüsselt werden müssen.

D. Tauschen von Speicher

Benötigt man mehr Online-Speicher, so kann man entweder zusätzlicher Speicher bei Wuala kaufen, oder lokalen Speicher gegen Online-Speicher tauschen. Hierbei gilt die Formel:

$$\begin{aligned} Onlinespeicher \quad = \quad & Zur\ Verfügung\ gestellter \\ & Speicherplatz\ * \\ & prozentuale\ Onlinezeit \end{aligned}$$

Somit bekommt ein Nutzer, der 20GB auf seiner Festplatte freigibt und im Durchschnitt jeden Tag 6 Stunden online ist, 5GB Online-Speicherplatz. Gibt man mehr als 20GB auf seiner Festplatte frei, so müssen mindestens 10% des freigegebenen Speichers auch bereits von Wuala benutzt werden, damit der zusätzliche Online-Speicher angerechnet wird [10]. Somit muss man bevor man den zusätzlichen Online-Speicher nutzen kann zuerst warten bis Wuala genug fremde Fragmente auf dem Rechner gesammelt hat.

IV. Sicherheit

Der Schutz der Dateien in Wuala ist gerade wegen des Konzeptes private Dateien in einem Peer-to-Peer-Netzwerk zu speichern von großer Bedeutung. In der IT-Sicherheit wird der Schutz von Dateien in mehrere konkrete Schutzziele untergliedert [11]. Im Folgenden wird untersucht wie gut Wuala für seinen Einsatzzweck relevante Schutzziele erfüllt. Am Ende dieses Kapitels wird noch kurz auf den Aspekt des nicht öffentlichen Quellcodes eingegangen.

A. Authentizität

Der Benutzer authentifiziert sich gegenüber Wuala durch ein Passwort, wie bei fast allen Diensten im Internet. Diese Methode ist nicht gerade sicher, jedoch gibt es keine Alternative die sich ähnlich einfach implementieren und benutzen lässt.

B. Datenintegrität

Da die Daten im Wuala-Netzwerk auf viele fremde Rechner verteilt liegen und deshalb nicht vor Manipulationen geschützt werden können, müssen unberechtigte Änderungen der Daten erkannt werden. Die Integrität einer Datei wird deshalb durch eine separate Hash-Datei auf dem zentralen Wuala-Server überprüft. Wird eine neue Datei ins Wuala-Netzwerk geladen oder eine alte überschrieben, so wird automatisch ein Hash-Wert der Datei gebildet und auf den Wuala-Server geladen. Beim Abrufen der Datei wird nun der gespeicherte Hash-Wert vom Wuala-Server geladen und mit dem Hash-Wert der heruntergeladenen Datei verglichen. Zum Bilden des Hash-Wertes kommt SHA-256 zum Einsatz [12]. Die Kollisionsfreiheit des Algorithmus macht es praktisch unmöglich zwei verschiedene Dateien mit dem gleichen Hash-Wert zu finden. Somit kann eine unbemerke Änderung der Datei (nahezu) ausgeschlossen werden, solange der Hash-Wert vor Veränderungen geschützt ist. Dieser Hash-Wert wird im Klartext auf dem Wuala-Server gespeichert, kann jedoch nur überschrieben werden, wenn die Schreib-Operation mit dem passenden Signatur-Schlüssel signiert ist. Im Besitz dieses Schlüssels sind nur Benutzer mit Schreibrecht für diese Datei. Dadurch kann trotz der Speicherung der Daten in einem nicht verlässlichen Netzwerk deren Integrität garantiert werden.

C. Informationsvertraulichkeit

Um unberechtigte Lesezugriffe auf Dateien zu verhindern werden alle Dateien vor dem Upload ins Wuala-Netzwerk verschlüsselt. Dazu wird AES mit 128 bit Schlüssellänge benutzt. Der AES-Algorithmus mit einer Schlüssellänge von 128 bit gilt als sicher und ist im Moment auch mit speziellen Supercomputern nicht zu knacken [13]. Als Modus der AES-Blockchiffre wird inzwischen Cipher Block Chaining (CBC) [11] verwendet [14], welches eine höhere Sicherheit bietet als zuvor verwendete Electronic Code Book (ECB) [11]. Zusätzlich wird sichergestellt, dass die für den CBC-Modus benötigten Initialisierungsvektoren für alle Dateien, die mit dem gleichen Schlüssel verschlüsselt sind, verschieden sind. Die AES-Schlüssel werden mit dem schon vorgestellten Cryptree verwaltet. Durch den Aufbau von Cryptree ist der Besitzer des Schlüssels eines Ordners in der Lage dessen Dateien und alle Unterordner zu entschlüsseln, jedoch keine Nachbar- oder höher gelegene Ordner. Es können nur die Namen der höher gelegenen Ordner entschlüsselt werden um den globalen Pfad des Ordners zu ermitteln. Dies ermöglicht es die Schlüssel für einzelne Ordner und Dateien freizugeben, ohne die anderen Schlüssel zu gefährden. Gelangt ein Angreifer an den Schlüssel des Wurzel-Ordners eines Benutzers, so kann er alle Ordner und Dateien des Benutzers entschlüsseln. Damit der Benutzer selbst an den Schlüssel des Wurzel-Ordners kommt, wird dieser wie bereits erwähnt aus einem Hash-Wert über den Benutzernamen und das Passwort gebildet. Somit ist jeder Benutzer selbst für die Sicherheit seiner Daten verantwortlich, indem er ein sicheres Passwort wählt. Der AES-Schlüssel zur Verschlüsselung einer Datei wird aus einem dem SHA-256 Hash-Wert über der Datei selbst generiert [15].

Dadurch unterscheiden sich gleiche Dateien auch nach der Verschlüsselung nicht und eine weit verbreitete Datei, die viele Nutzer hochgeladen haben muss nur einmal gespeichert werden, was Speicherplatz spart. Der Nachteil daran ist, dass man über die Hashes zur Verifikation der Datei herausfinden kann wer alles diese Datei besitzt, auch wenn der Benutzer diese Datei in einem privaten Ordner gespeichert hat [16]. Zwar hat nur Wuala allein die Möglichkeit den Hashwerten Benutzer zuzuordnen, jedoch ist es Wuala dadurch möglich das Netzwerk zu zensieren: Dazu hat Wuala eine Sammlung an Dateien, die sie verbieten möchten. Nun werden diese Dateien verschlüsselt und ihre Hash-Werte gebildet und mit den Hash-Werten aus Wuala verglichen. Wuala kann diese Dateien laut den AGBs [17] ohne Angabe eines Grundes löschen. Weiterhin kann Wuala alle Benutzer ausfindig machen, die eine dieser Dateien besitzen. Speichert man in Wuala private, selbst erstellte Dateien die niemand sonst besitzt, so ist dies nicht möglich.

D. Verfügbarkeit

Um eine möglichst hohe Verfügbarkeit der Dateien im Wuala-Netzwerk zu gewährleisten wird jede Datei wie in Kapitel 2 beschrieben mit Hilfe von Erasure Codes redundant gespeichert, was eine hohe Verfügbarkeit gewährleistet. Zusätzlich wird jede Datei auf dem zentalen Wuala-Server gespeichert. Somit kann eine Datei auch abgerufen werden, wenn sie nicht oder nicht vollständig im Netzwerk verfügbar ist. Sollte der Wuala-Server einmal ausfallen, so könnte eine Datei im Großteil aller Fälle theoretisch immer noch aus dem Netzwerk geladen werden. Das macht jedoch keinen Sinn, da die Integriät der Datei ohne den Wuala-Server nicht geprüft werden kann. Deshalb ist die Verfügbarkeit von Dateien im Wuala-Netzwerk von der Verfügbarkeit des Wuala-Servers abhängig.

E. Verbindlichkeit anstatt Anonymität

Wuala ordnet jeder Datei und jedem Kommentar einen Besitzer in Form eines Wuala-Benutzers zu. Außerdem wird bei jedem Log-in eines Wuala-Nutzers seine IP-Adresse und die Zeit mitprotokolliert [18]. Dadurch kann es Strafverfolgern im Falle eines Gesetzesbruchs ermöglicht werden Rückschlüsse auf die reale Person, welche die Datei ins Wuala-Netzwerk geladen hat, zu ziehen. Anonymität ist im Wuala-Netzwerk von den Entwicklern nicht gewollt [19]. Da es keine Möglichkeit gibt Dateien anonym hochzuladen ist es möglich, das Verbreiten von illegalen Dateien und Urheberrechtsverletzungen zu verfolgen, was Wuala in den AGBs [17] ausdrücklich ankündigt wenn ein solcher Verstoß gemeldet wird. Dort kann ebenfalls nachgelesen werden, dass Wuala selbst jedoch nicht aktiv werden das Netzwerk nach illegalen Inhalten durchsuchen will. Im Oktober 2008 wurde in Wuala auf Druck der Filmindustrie eine Gruppe geschlossen, die hauptsächlich dem Tausch unheberrechtlich geschützter Dateien diente [20]. Da sich scheinbar Mitarbeiter der Filmindustrie in der Gruppe befanden und Wuala über die Missachtung der Uhrheberrechte informierten, wurde die Gruppe geschlossen um die Nutzer vor Abmahnungen zu schützen. Dieser Fall zeigt, dass Wuala sich so gut wie möglich aus rechtlichen Streitigkeiten heraus halten will und nicht darauf aus ist seine eigenen Nutzer zu verklagen.

F. Nicht-öffentlicher Quellcode

Obwohl Wuala auf einer ganzen Reihe von Open-Source-Projekten basiert [21], ist der Quellcode von Wuala nicht öffentlich zugänglich. Auf meine Anfrage hin wurde mir mitgeteilt, dass es auch nicht geplant sei den Quellcode zu veröffentlichen. Da man also nicht selbst überprüfen kann was das Programm macht bleibt nur den Wuala-Entwicklern zu vertrauen, dass Wuala keine Spyware, Trojaner oder sonstige Schadprogramme enthält. So könnte der Wuala-Client beispielsweise das Benutzerpasswort speichern und an einen Wuala-Server senden. Privatpersonen werden trotzdem wohl nicht allzu skeptisch sein und dem schweizer Startup ihre privaten Daten anvertrauen. Für Firmen - welche ausdrücklich zur Wuala-Zielgruppe gehören [22] - die ihre Geschäftsdaten in Wuala speichern möchten stellt dies jedoch eine grössere Hürde dar.

V. Vergleich mit serverbasierten Cloud-Storage-Diensten

Wuala sticht durch zwei Besonderheiten aus der Masse der Cloud-Storage-Services hervor: Erstens speichert es die Dateien als einziger Service in einem Peer-to-Peer-Netzwerk ab und nicht nur auf einem zentralen Server. Zweitens ist es der einzige Service, der Community-Features eines Sozialen Netzwerks wie das Bilden von Freundschaften und Gruppen, sowie eine Kommentarfunktion bietet [23].
Auch bei der Sicherheit geht Wuala andere Wege. Wuala verschlüsselt die Dateien nämlich schon vor dem Upload ins Netz. Dadurch kann angeblich nicht einmal Wuala selbst meine privaten Dateien entschlüsseln. Da der Quellcode von Wuala nicht öffentlich ist, kann das nicht überprüft werden. Andere Services, wie zum Beispiel Dropbox [24], verwenden eine SSL-Verschlüsselung zur Übertragung auf den Server wo sie dann vom Anbieter verschlüsselt werden. Dadurch ist der Anbieter in der Lage alle Dateien zu entschlüsseln.
Bei der Geschwindigkeit hat Wuala einen Vorteil gegenüber rein serverbasierten Diensten, da eine Datei parallel von mehreren Rechnern geladen wird, anstatt sequentiell von einem Server. Dieser Vorteil wird umso spürbarer, je mehr das Wuala-Netzerk wächst, da dann tendenziell weniger Dateien vom Server geladen werden müssen. Beim Upload gibt es im Normalfall keine spürbaren Unterschiede. Falls jedoch eine Datei ins Wuala-Netzwerk geladen werden soll, die bereits vorhanden ist, muss diese nicht nochmals hochgeladen werden, wodurch sich wieder ein Geschwindigkeitsvorteil für Wuala ergibt.
Für die meisten Cloud-Storage-Dienste muss ein Client-Programm installiert werden, welche meist für Windows und MacOS verfügbar sind. Linux wird nur ungefähr von der Hälfte der Cloud-Storage-Dienste unterstützt. Hervorzuheben ist hier Box.net [25] welches keine Client-Software benötigt

und komplett über den Browser bedient wird.
Die Dateisystem-Interation ist bei Wuala Wuala bietet zwar eine Integration ins Dateisystem, diese ist jedoch noch verbesserungswürdig. Andere Anbieter wie Dropbox [24] und ZumoDrive [26] haben ihre Dienste besser ins Dateisystem integriert. Hier zeigt ein kleines Symbol an jeder Datei an, ob die Datei aktuell mit der Online-Version synchronisiert ist. Zudem bieten beide Dienste eine History-Funktion, mit der alte Versionen einer Datei wiederhergestellt werden können.
Die Preise für zusätzlichen Speicher für Wuala liegen am unteren Ende der Preisspanne für Cloud-Storing-Dienste. Zusätzlich bietet Wuala als einzigster Anbieter die Option lokalen Speicherplatz gegen Online-Speicher zu tauschen, und so völlig kostenlos zusätzlichen Speicher zu erhalten.

Tabelle I
VERGLEICH AUSGEWÄHLTER CLOUD-STORAGE-DIENSTE

	Wuala	Dropbox	ZumoDrive	Box.net
Kostenloser Speicher	1GB	2GB	1GB	1GB
50 GB extra (pro Jahr)	60 Euro	75 Euro	109 Euro	-
Dateisystem-Integration	ja	ja	ja	nein
Betriebsysteme	W,M,L	W,M,L	W,M	alle
History-Funktion	nein	ja	ja	ja

Legende: W = Windows, M = MacOS, L = Linux, Stand: 29.04.2009

VI. ZUSAMMENFASSUNG UND AUSBLICK

Wuala ist für den Heimanwender eine gute Lösung seine Dateien online verfügbar zu machen. Es besitzt keine gravierenden Sicherheitslücken und bieten kostenlos jede Menge Speicher. Durch die Einzigartigkeit und Leistungsfähigkeit der Software hat Wuala gute Karten auf dem umkämpften Online-Speicher-Markt zu bestehen. Gespannt sein darf man außerdem welche neuen Entwicklungen die Fusion von Wuala mit dem Festplattenhersteller LaCie [27] bringt. Da LaCie bereits eine Festplatte mit Internetzugriff [28] im Programm hat, liegt es nahe dass LaCie eine Wuala-fähige Festplatte auf den Markt bringen könnte.

LITERATUR

[1] Wikipedia.com, "Cloud Storage," *http://en.wikipedia.org/w/index.php?title=Cloud_computing&oldid=286817477#Storage*, 29.04.2009
[2] Wikipedia.com, "Erasure code," *http://en.wikipedia.org/w/index.php? title=Erasure_code&oldid=258867740*, 29.04.2009
[3] D. Grolimund, "Wuala - a distributed file system," *http://www.youtube.com/watch?v=3xKZ4KGkQY8*, 29.04.2009
[4] Wikipedia.com, "Reed–Solomon error correction," *http://en.wikipedia.org/wiki/Reed_Solomon*, 29.04.2009
[5] GetSatisfaction.com, "Cleversafe - Wuala's Twin?," *http://getsatisfaction.com/wuala/topics/cleversafe_wualas_twin*, 29.04.2009
[6] D. Grolimund, L. Meissner, S. Schmid, R. Wattenhofer, "Cryptree: A Folder Tree Structure for Cryptographic File Systems," SRDS, 2006
[7] Petar Maymounkov, David Mazières, "Kademlia: A Peer-to-peer Information System Based on the XOR Metric," New York University, *http://pdos.csail.mit.edu/ petar/papers/maymounkov-kademlia-lncs.pdf*
[8] D. Grolimund, L. Meissner, S. Schmid, R. Wattenhofer, "Havelaar: A Robust and Efficient Reputation System for Active Peer-to-Peer Systems," NETECON, 2006
[9] Wuala, "Schweizer Fernsehen - Wuala, social online storage," *http://www.wuala.com/Schweizer%20Fernsehen*
[10] GetSatisfaction.com, "Send me data!," *http://getsatisfaction.com/wuala/topics/send_me_data*, 29.04.2009
[11] C. Eckert, "IT-Sicherheit," 5. Auflage, München: Oldenbourg-Verlag, 2008
[12] C. Percival, "Wuala update," *http://www.daemonology.net/blog/2007-10-26-wuala-update.html*, 29.04.2009
[13] National Institute of Standards and Technology, "AES Questions & Answers," *http://www.nist.gov/public_affairs/releases/aesq&a.htm*, 29.04.2009
[14] C. Percival, "Wuala's improved security," *http://www.daemonology.net/blog/2008-11-07-wuala-security.html*, 29.04.2009
[15] GetSatisfaction.com, "Instant upload?! Hows that work with encryption?," *http://getsatisfaction.com/wuala/topics/instant_upload_hows_that_work_with_encryption*, 29.04.2009
[16] GetSatisfaction.com, "Dateien löschen innerhalb des Wuala," *http://getsatisfaction.com/wuala/topics/dateien_l_schen_innerhalb_des_wuala*, 29.04.2009
[17] Wuala, "Allgemeine Geschäftsbedingungen," *http://www.wuala.com/de/about/terms*, 29.04.2009
[18] GetSatisfaction.com, "IP-Log," *http://getsatisfaction.com/wuala/topics/ip_log*, 29.04.2009
[19] GetSatisfaction.com, "Complete Privacy; feature request," *http://getsatisfaction.com/wuala/topics/complete_privacy_feature_request*, 29.04.2009
[20] gulli.com, "Filmindustrie lässt gulli-Usergroup löschen (update)," *http://www.gulli.com/news/wuala-filmindustrie-l-sst-2008-10-09*, 29.04.2009
[21] Wuala, "Quellcode von Drittanbietern," *http://www.wuala.com/de/about/thirdpartycode*, 29.04.2009
[22] Wuala, "Wer benutzt Wuala?," *http://www.wuala.com/de/learn/usecases*, 29.04.2009
[23] Neue Zürcher Zeitung, "Festplatte mit sozialer Ader," *http://www.nzz.ch/magazin/mobil/festplatte_mit_sozialer_ader_1.804251.html*, 29.04.2009
[24] Dropbox, *https://www.getdropbox.com*, 29.04.2009
[25] Box.net, *http://www.box.net*, 29.04.200
[26] ZumoDrive, *http://zumodrive.com*, 29.04.200
[27] Wuala, "Exciting news: Wuala merges with LaCie," *http://www.wuala.com/blog/2009/03/exciting-news-wuala-merges-with-lacie.html*, 29.04.2009
[28] LaCie, "LaCie Internet Space," *http://www.lacie.com/de/products/ product.htm?pid=11136*, 29.04.2009

BitTorrent

Simon Mittelberger
Betreuer: Benedikt Elser
Seminar Future Internet SS2009
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München
Email: simon.mittelberger@in.tum.de

***Kurzfassung*—Man stelle sich vor man besitze ein Musikstück, ein Video, oder eine beliebige andere Datei, welche relativ groß ist und sehr viele andere Benutzer interessiert. Soll diese Datei nun an viele Benutzer verteilt werden, wird ein Server und genügend Bandbreite benötigt, welche ein normaler Computeranwender nicht zur Verfügung hat. BitTorrent bietet eine Lösung zu diesem Problem.**
BitTorrent ist ein Peer-to-Peer System durch welches Dateien dezentral verteilt werden können. Die Last wird nicht von einem oder einer Gruppe von Servern getragen, sondern wird von jedem einzelnen Client mitgetragen. Ein so genannter Tracker wird benötigt um die Clients zu koordinieren. Er teilt einem Client mit, welcher andere Client Teile einer Datei besitzt. Der Client entscheidet selbständig von welchen Clients er lädt und an welche anderen Clients er verteilt.
Verschiedene Strategien regeln, die Reihenfolge in der die Teile der Dateien von anderen Clients geladen werden und an welche anderen Clients Teile der Dateien verteilt werden.

***Schlüsselworte*—BitTorrent, Torrent, Filesharing, Tit for Tat, Pareto Effizienz, Gefangenendilemma, Peer, Seeder, Leecher, Chunk, Choking-Algorithmus, Peer-to-Peer, P2P**

I. Einleitung

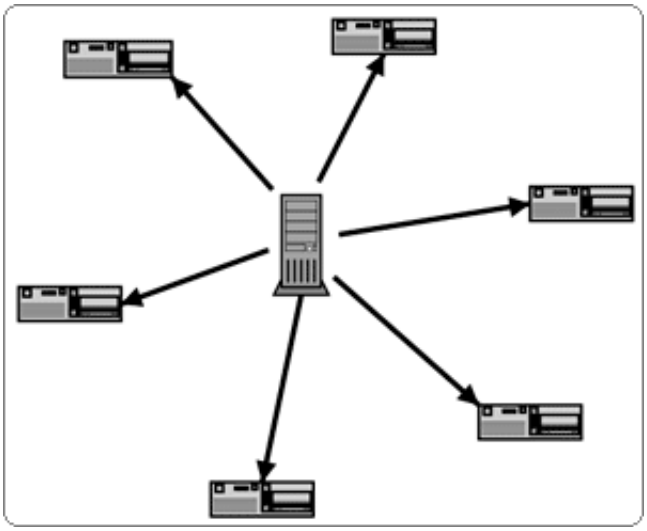

Abbildung 1. Client-Server Modell: Daten-Verteilung auf herkömmlichem Weg; ein zentraler Server verteilt die Daten an viele Clients. [1]

Um eine Datei auf herkömmlichem Weg unter verschiedenen Nutzern des Internets zu verteilen benötigt man einen bzw. mehrere Server, welche die Datei hosten und eine Schnittstelle zum Download für die Clients anbieten. Will man nun eine große Datei unter sehr vielen Benutzern verteilen, benötigt der Server ausreichend Leistung, was heutzutage kein Problem mehr darstellt, und genügend Bandbreite. Die Bandbreite des Servers ist in solchen Szenarien heute meist der Flaschenhals. Der Begriff BitTorrent steht für ein Protokoll zum Datenaustausch über das Internet, für ein Client-Programm, welches dieses Protokoll benutzt, sowie für ein Unternehmen, welches den BitTorrent Client und verschiedene andere Produkte rund um das Thema Datenaustausch über das Internet anbietet.
Das Protokoll BitTorrent wurde im April 2001 von Bram Cohen entwickelt, eine erste Implementation folgte im Juli 2001. Im Jahre 2004 wurde das Unternehmen BitTorrent Inc. von Bram Cohen und Ashwin Navin gegründet, welches auch die Weiterentwicklung des Protokolls vorantreibt.

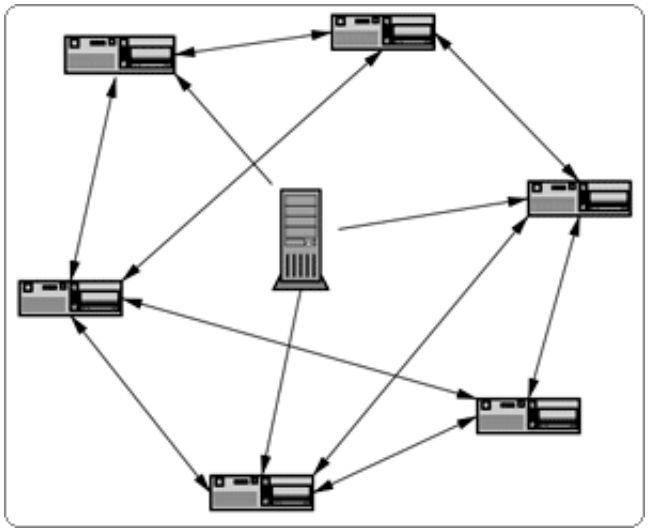

Abbildung 2. Torrent Netzwerk: Ein zentraler Server übernimmt Torrent-Verteilung und Tracking-Aufgabe. Die Clients laden die Datei vom Server und von anderen Clients, was eine Entlastung des Servers zur Folge hat. [1]

Das Protokoll ermöglicht einen Datenaustausch ohne zentralen Server. Jeder Client ist zugleich Uploader und Downloader. Er verteilt die Teile der Datei, welche er bereits empfangen hat an andere Clients weiter, wodurch die Last auf die Clients

aufgeteilt wird.
Über das Torrent Netzwerk lassen sich beliebige Daten übertragen. Man findet von der Linux Distribution und verschiedener Software über Ebooks und Musik bis hin zu Filmen, nahezu alle Arten von Dateien. Vor allem wegen Letzteren hat die Medienindustrie dem Torrent Netzwerk den Kampf angesagt. Torrent Verteiler, wie die schwedische Webseite ThePirateBay.org stehen ganz oben auf der Abschussliste von Warner Bros. Entertainment und Co.
Diese Arbeit beschäftigt sich mit dem Technischen Hintergrund des BitTorrent Systems. Kapitel II befasst sich mit der Client Seite des Systems, Kapitel III mit der System Seite. In Kapitel IV und V wird ein genauerer Einblick in verschiedene Strategien hinter dem BitTorrent System gegeben. Um das Lesen und das Verständnis dieser Arbeit zu erleichtern befindet sich auf der letzten Seite eine Begriffserklärung.

II. Client

Der Download einer Datei über ein BitTorrent Netzwerk gestaltet sich aus der Sicht eines Benutzers als sehr einfach. Der Benutzer lädt eine Torrent Datei über seinen Webbrowser herunter, oder speichert diese über einen anderen Weg auf seinem Computer. Auf den Inhalt der Torrent Datei wird in Kapitel II-B kurz eingegangen. Diese Torrent Datei wird mit dem Client-Programm geöffnet, der Speicherort der Datei wird ausgewählt und schon beginnt der Download. Im Fenster werden nun verschiedene Daten angezeigt, wie zum Beispiel die Download- und Uploadraten, der Fortschritt des Downloads und die Anzahl der Peers mit denen aktuell eine Verbindung unterhalten wird. Diese Einfachheit hat sicherlich wesentlich zur Popularität von BitTorrent beigetragen, vielleicht sogar mehr als seine Performance- und Lastverteilungseigenschaften [2].
2006 liefen über 70 Prozent des deutschen Internetverkehrs über P2P-Systeme. BitTorrent hat die bis dahin bekannteste Tauschbörse eDonkey überholt und verursacht mehr als die Hälfte des gesamten P2P-Verkehrs in Deutschland. Die am häufigsten getauschten Daten seien nach den Aussagen der ipoque GmbH aktuelle Kinofilme, Musik, Serien und Computerspiele. Allerdings ist auch bei Büchern und Hörbüchern ein Zuwachs zu verzeichnen. Derzeit weist BitTorrent, laut BitTorrent Inc., über 160 Millionen installierte Clients weltweit auf [3], [4].
Es gibt eine ganze Reihe von Client-Programmen, die das BitTorrent Protokoll implementieren. Der erste Client wurde von Bram Cohen im Oktober 2002 unter dem Namen BitTorrent herausgegeben. Mittlerweile wird der Client vom Unternehemn BitTorrent Inc. weiterentwickelt und ist seit Version 6.0 keine freie Software mehr. In vielen Clients wurde das BitTorrent Protokoll erst im Nachhinein implementiert, da sie nicht von Anfang an dafür vorgesehen waren. Folgende Liste stellt eine (sehr) kleine Auswahl aller Clients dar:

A. BitTorrent

B. Gnome BitTorrent

C. KTorrent

D. LimeWire

E. qBitTorrent

F. rTorrent

G. Torrent

[5]

III. Technischer Hintergrund

BitTorrent ist ein Peer to Peer System zum Dateiaustausch. Es besteht aus zwei Programmen, dem Client-Programm und dem Server-Programm, genannt Tracker.

A. Peer to Peer (P2P)

Peer to Peer Systeme klassifizieren sich dadurch, dass Teilnehmer, so genannte Peers, sowohl Dienste anbieten, als auch Dienste anderer Teilehmer nutzen. Einfache P2P Netze haben keinen zentralen Punkt, sondern die Peers stellen sich gegenseitig Betriebsmittel und Ressourcen zur Verfügung. Die Weiterentwicklung von P2P Netzen mit zentralen Komponenten nennt man Super Peer Netze. In einer solchen Konfiguration übernehmen so genannte Super Peers die Organisation des Netzes. Das BitTorrent System besitzt in der Regel eine zentrale Einheit, den Tracker, und fällt damit unter die Super Peer Netze. Es gibt aber auch die Möglichkeit des trackerlosen Betriebs, wie später beschrieben wird.

B. Die Veröffentlichung einer Datei über BitTorrent an einem Beispiel

Um zum Beispiel die Datei Ebook.pdf über das BitTorrent Netzwerk zu verteilen, wird eine Torrent Datei, zum Beispiel Ebook.torrent erstellt. Diese Torrent Datei ist sehr klein und enthält Informationen über Ebook.pdf, wie den Namen, die Länge, die Hashwerte der Chunks (Chunks werden später erleutert) und die URL des Trackers. Sie wird über das Internet verfügbar gemacht, zum Beispiel durch einen Webserver, FTPServer, ... Ein Client, der die vollständige Ebook.pdf Datei besitzt muss gestartet werden. Öffnet nun ein Client-Programm die Torrent Datei, kann es die veröffentlichte Datei laden.

C. BitTorrent System

Um eine bessere Performance zu erreichen verteilt das BitTorrent System nicht komplette Dateien, sondern kleinere Einheiten. Diese kleineren Einheiten sind typischerweise 256 KByte große Stücke der Dateien. Man nennt sie Chunks.

1) Der Client: Ein Client lädt Chunk für Chunk einer Datei herunter und setzt diese dann wieder zusammen. Zum Zeitpukt, an dem ein Client einen Chunk fertig geladen hat verifiziert er diesen mit der Prüfsumme aus der Torrent Datei. Nach der Prüfung teilt er dem Tracker mit, dass er im Besitz des Chunks ist. Durch diese Vorgehensweise können die empfangenen Chunks sofort an andere Clients weiterverteilt werden, ohne darauf zu warten, dass die komplette Datei fertig geladen ist. Für die Clients gibt es verschiedene Bezeichnungen:

Sworm	Eine Gruppe von Clients, welche an der selben Datei interessiert sind.
Seeder	Ein Client, der im Besitz aller Chunks einer Datei ist. Er lädt nicht mehr herunter, sondern verteilt nur mehr weiter.
Leecher	Zu diesem Begriff gibt es verschiedene Definitionen. Manche Quellen geben an es handle sich um einen Client, der nur herunterlädt und nicht weiterverteilt [6], andere geben an es sei dasselbe wie ein Peer [7].
Peer	Ein Client, der sowohl herunterlädt, als auch weiterverteilt.

Tabelle I
VERSCHIEDENE BEZEICHNUNGEN FÜR DIE CLIENTS.

2) Der Tracker: Der Tracker verwaltet eine Liste der Clients und ihrer Chunks. Ein Client fordert vom Tracker eine Liste mit Clients an, die einen bestimmten Chunk besitzen. Der Tracker gibt typischerweise eine Liste mit fünf zufälligen Clients, welche im Besitz dieses Chunks sind, zurück. Die Kommunikation mit dem Tracker erfolgt über ein einfaches Protokoll, welches auf HTTP aufbaut. Der Download des Chunks wird zwischen den Clients ausgehandelt und muss nicht immer stattfinden. In Kapitel V: Der Choking Algorithmus wird darauf genauer eingegangen.
Das BitTorrent-Netzwerk existiert nicht als ein gemeinsames Gesamtnetz, wie zum Beispiel eDonkey, sondern vielmehr baut jeder Tracker mit den sich beteiligenden Clients ein eigenes Netz auf. Ein Tracker- bzw. Torrent Anbieter kann sich somit leichter von illegalen Inhalten distanzieren. Ein Tracker kann auch mehrere Dateien verwalten.

D. Trackerloser Betrieb

Der Tracker muss ständig erreichbar sein. Ist er nicht mehr erreichbar können sich die Clients untereinander nicht finden und keine neuen Chunks mehr anfordern. Seit der, im November 2005 erschienen Version 4.2.0 unterstützt der BitTorrent Client den so genannten trackerlosen Betrieb. Die Trackerfunktion wird dabei von den Clients übernommen und ähnlich wie beim Kademilla-Netzwerk als verteilte Hashtabelle auf der Clientseite abgelegt.

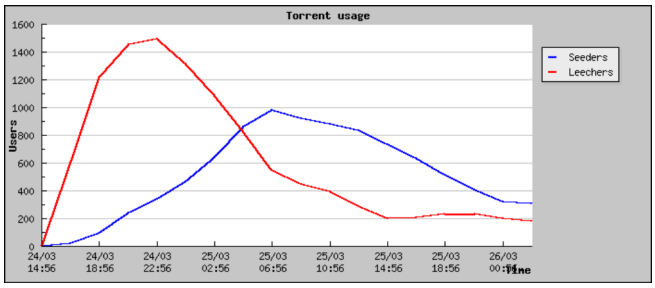

Abbildung 3. Der Graph zeigt die Entwicklung der Seeder und Leecher einer großen Datei (über 400 MByte) über die Zeit. Die Anzahl der Leecher steigt zu Beginn rapide an, erreicht ihren Höhepunkt und fällt dann exponentiell. Die Anzahl der Seeder steigt wesentlich langsamer, erreicht ihren Höhepunkt nach der Anzahl der Leecher und fällt dann ebenfalls exponentiell ab. [8]

E. Von welchem Peer laden?

Der Tracker gibt bei einer Anfrage eine Liste von Peers zurück, welche an der Verteilung der Datei(en) beteiligt sind. Der Client versucht von allen anderen Clients, die im Besitz von Chunks sind, welche ihn interesssieren, herunterzuladen. Die Clients, welche meist in ganz normalen Haushalten lokalisiert sind, können eine geringe Uploadrate besitzen. Bei populären Dateien, welche sehr viele interessierte Clients haben, kann die Bandbreite der Clients unter Umständen nicht ausreichen, um alle anfragenden Clients zu bedienen. Der Choking Algorithmus entscheidet für jeden anfragenden Client, ob seine Anfrage beantwortet (cooperate) oder ob seine Anfrage zurückgewiesen (choke) wird. Eine genauere Beschreibung wird in Kapitel V gegeben.

F. Unterbrechungsfreiheit

Um die Ausreizung der Übertragungsgeschwindigkeit zu erreichen ist es wichtig mehrere Anfragen parallel zu führen, weil eine Antwort des Clients unter Umständen etwas länger dauern kann. BitTorrent erreicht dies indem Chunks nochmal in 16 KByte große Teile, genannt Sub-Pieces, aufgeteilt werden. Der Client fordert nicht einen kompletten Chunk, sondern immer eine gewisse Anzahl an Sub-Pieces, meist fünf, parallel an. Sobald ein Sub-Piece fertig geladen wurde, wird sofort das Nächste angefordert. Dieses Verfahren erhöht die Kommunikation zwischen den Clients und hilft Fehler in der Übertragung zu vermeiden.

IV. WELCHER CHUNK?

Jeder Client ist selbst verantwortlich in welcher Reihenfolge er Chunks von welchem Client herunterlädt. Damit die Bandbreite so gut wie möglich ausgenutzt wird, hat BitTorrent eine Strategie, um die Reihenfolge der Chunks zu bestimmen. Diese Strategie besteht aus vier Teilen, welche zu unterschiedlichen Zeitpunkten eingesetzt werden. Der BitTorrent Client entscheidet dabei aber nicht nur zu seinem eigenen Vorteil, sondern handelt auch im Interesse des Netzes, also auch der Peers, welche von ihm downloaden.

A. Strict Priority

Sobald ein Sub-Piece eines Chunks angefordert wurde, werden auch alle übrigen Sub-Pieces dieses Chunks angefordert, bevor andere Sub-Pieces anderer Chunks angefordert werden. Dadurch werden unvollständige Chunks so schnell wie möglich komplettiert und die Wahrscheinlichkeit, dass die Quelle eines Chunks verschwindet, wird minimiert. Eine Quelle eines Chunks kann zum Beispiel verschwinden, wenn der Peer, der im Besitz dieses Chunks ist das Internet verlässt, oder, wenn ein Benutzer das Client-Programm beendet.

B. Rarest First

Zum Zeitpunkt an dem ein Chunk vollständig geladen wurde, steht der Client vor der Wahl, welchen Chunk er als nächstes anfordert. Rarest First besagt hier, dass der Client den Chunk wählt, der in der geringsten Anzahl auf den anderen Peers vorhanden ist. Diese Strategie sorgt dafür, dass der Client immer Chunks hat, welche von vielen seiner Peers im selben Netz angefordert werden. Somit kann er immer uploaden, wodurch seine Upload-Bandbreite gut ausgenutzt wird.
Ein Torrent Netz kann vor folgenden Problemen stehen:

1) zu Beginn des Downloads: Wenn zum Beispiel eine Datei über ein Torrent Netzwerk verfügbar gemacht werden soll und dafür zu Beginn nur ein Seed eingesetzt wird, hat man das Problem, dass alle interessierten Clients von diesem Seed laden möchten. Man hat also eine herkömmliche Client - Server Situation. Rarest First garantiert hier, dass jeder Interessierte einen anderen Chunk lädt und sich die Quellen für jeden Chunk somit rasch vervielfachen, was die Downloadgeschwindigkeit für zukünftig anfordernde Peers steigert.

2) Der Start-Seeder wird heruntergefahren: Der Seed, mit welchem der Upload gestartet wurde, kann, sei es aus Kostengründen, oder Wartungsgründen, vom Netz genommen werden. Nun ist es möglich, dass ein bestimmter Chunk nicht mehr verfügbar ist, weil alle anderen Peers, welche diesen Chunk ebenfalls besitzen, zufälligerweise offline sind. Der Torrent ist jetzt unbrauchbar, da ein vollständiger Download der Datei nicht mehr möglich ist. Diesem Problem, beugt Rarest First vor, da die unverbreitetsten Chunks so schnell als möglich vervielfältigt werden und so die Wahrscheinlichkeit einer solchen Situation zu begegnen, verringert wird.

C. Random First Piece

Rarest First liefert zum Zeitpuntk, an dem ein Peer mit dem Download einer Datei beginnt schlechtere Resultate. Der Peer hat zu dieser Zeit keine Chunks, welche für andere Peers von Interesse wären und kann demnach nicht uploaden. Würde er nun als erstes den Chunk runterladen, der am seltensten vorkommt würde der Download, im Verhältnis zum Download eines öfter vorkommenden Chunks, langsamer von statten gehen. Deshalb ist es wichtig für Peers, nach dem Downloadstart so schnell wie möglich einen vollständigen Chunk zu bekommen. Würde die Strategie in diesem Fall immer den schnellsten Peer wählen, würde dieser überlastet werden, da jeder neue Peer sich sofort auf ihn stürzen würde. Random First Piece, der Peer wählt zu Beginn einen zufälligen Chunk aus, hat hier gute Resultate gezeigt. Sobald der Client einen vollständigen Chunk hat wird mit der Strategie Rarest First fortgefahren.

D. Endgame Mode

Wenn ein Chunk von einem Peer mit einer langsamen Anbindung angefordert wird, ist das inmitten eines Downloads kein Problem, aber es kann die Beendigung eines Downloads verzögern, was unerwünscht ist. Um dieses Problem gegen Ende eines Downloads zu unterbinden startet der Client den Endgame Modus. Alle fehlenden Sub-Pieces werden von allen Peers im Torrent Netz angefordert. Sobald eines dieser Sub-Pieces erhalten wurde wird die Anfrage dieses Sub-Pieces mittels einer Cancel-Mitteilung an alle Peers zurückgezogen, um nicht zu viel Bandbreite zu verschwenden. Das Ende einer Datei wird somit immer schnell geladen. Zeitlich gesehen ist der Endgame Modus von nur sehr kurzer Dauer, wodurch dieses Verhalten die Bandbreite nicht zu sehr belastet.

V. Der Choking Anlgorithmus

Um die Downloadgeschwindigkeit zu steigern wird jeder Peer versuchen von so vielen Peers wie nur möglich herunterzuladen. Man kann sich leicht vorstellen, dass diese Strategie das Netzwerk, sowie auch die einzelnen Peers überlasten würde. Aus diesem Grund wurde der Choking-Algorithmus eingeführt. Angenommen Peer A möchte von Peer B herunterladen, dann kann B die Anfrage von A bedienen oder ignorieren. Ein Peer erlaubt immer nur einer gewissen Anzahl an anderen Peers, typischerweise vier, das Downloaden. Alle anderen werden abgelehnt. Dieses Annehmen und Ablehnen wird durch den Choking-Algorithmus geregelt. Er ist im eigentlichen Sinne nicht Teil des BitTorrent Protokolls, aber unbedingt nötig um die Effizienz des gesamten Systems zu gewährleisten. Der Algorithmus versucht durch eine Strategie namens Tit for Tat Pareto Effizienz herzustellen [8].

A. Pareto Effizienz

Pareto Effizienz, benannt nach dem Ökonom und Soziologen Vilfredo Pareto (1848-1923), spielt in Wirtschaftswissenschaften eine Rolle. Ein System, in dem zwei Parteien miteinander in Aktion treten ist pareto effizient, wenn es den zwei Parteien gelingt einen Austausch durchzuführen, der Beiden nützt. Der Austausch wäre hierbei das Up- und Downloaden von Chunks. In der Informatik ist die Suche nach Pareto Effizienz ein lokaler Optimierungs-Algorithmus, welcher nach einem Weg sucht, durch den zwei Parteien eine Vermehrung des gemeinsamen Gutes erreichen können. Diese Art von Algorithmen neigt dazu, zu einem globalen Optimum zu führen. Das bedeutet, dass die Handlungen eines Peers nicht nur ihm allein, sondern dem gesamten System zu einer Verbesserung verhelfen [9].

B. Entwicklung von Tit for Tat

Die Entwicklung von Tit for Tat ist mit dem Gefangenendilemma verknüpft, weshalb vorher kurz auf dieses eingegangen

wird. Das Gefangenendilemma ist ein Paradoxon, welches von zwei Individuen handelt, welche angeblich ein gemeinsames Verbrechen begangen haben. Sie werden getrennt voneinander eingesperrt und befragt. Beiden wird angeboten, durch Verrat des Anderen die eigene Haftstrafe zu mindern. Verrät Häftling A Häftling B, so bekommt Häftling A 1 Jahr und Häftling B 5 Jahre aufgebrummt und umgekehrt. Schweigen beide werden beide zu 2 Jahren verurteilt und verraten sie sich gegenseitig werden beide für 4 Jahre weggesperrt [10].

	A schweigt	A verrät
B schweigt	A: 2 Jahre, B: 2 Jahre	A: 1 Jahr, B: 5 Jahre
B verrät	A: 5 Jahre, B: 1 Jahr	a: 4 Jahre, B: 4 Jahre

Tabelle II
DIESE TABELLE ZEIGT DIE STRAFEN ZU DEN VERSCHIEDENEN ENTSCHEIDUNGSMÖGLICHKEITEN DER GEFANGENEN A UND B. [10]

Ist es nun besser seinen Mittäter zu verraten, oder zu schweigen?
Dieser Frage wollte Robert Axelrod auf den Grund gehen. Dazu schrieb er einen Wettbewerb um die beste Strategie für ein auf 200 Runden basierendes Gefangenendilemma aus. Eine Runde besteht dabei aus jeweils einer Aktion von Gefangenem A und Gefangenem B. Die Jahre Gefängnis, welche der jeweilige Verbrecher in einer Runde bekommt werden aufsummiert. Ziel ist es natürlich am Ende so wenig wie möglich Jahre zu bekommen. Die beste Strategie in diesem Wettbewerb war überraschenderweise auch die einfachste. Diese Strategie wurde von Anatol Rapoport eingebracht und hat den Namen ”Tit for Tat” [11].

C. Tit for Tat

Der Begriff „Tit for Tat (TFT)“, oder auf deutsch „wie du mir, so ich dir“, kommt von der Spieltheorie und bezeichnet eine Strategie in der ein Spieler mit seinem Zug auf einen vorangegangenen Zug seines Gegenspielers antwortet.
Unter Betrachtung eines beliebigen, zugbasierten Spiels in dem Spieler A gegen Spieler B antritt, kann man sich TFT folgendermaßen erklären (angenommen Spieler B wendet TFT an): Spieler A handelt entgegen den Zielen von Spieler B, so wird Spieler B in seinem nächsten Zug ebenfalls entgegen den Interessen von Spieler A handeln. Profitiert Spieler B hingegen aus dem Zug von Spieler A, so wird auch Spieler B in seinem nächsten Zug entgegenkommend handeln.
Bei TFT handelt es sich um eine freundliche Strategie, was bedeutet, dass der erste Zug immer kooperierend ist. Angenommen zwei TFT Spieler begegnen sich, so kooperieren diese während des gesamten Spiels.

Abbildung 4. Das Händeschütteln am Beginn einer Konversation kann ein initiales Kooperieren darstellen [12]

TFT hat nach Axelrod vier wichtige Elemente, welche jede wirkungsvolle Strategie im wiederholten Gefangenendilemma auszeichnen. [11]

Klarheit	TFT ist so klar und einfach wie nur möglich.
Nettigkeit	TFT kooperiert im ersten Zug und auch in folgenden, so lange kein Verrat vorliegt.
Provozierbarkeit	TFT bestraft jeden Verrat sofort.
Nachsichtigkeit	TFT ist bereit die Kooperation sofort wieder aufzunehmen.

Tabelle III
VIER EIGENSCHAFTEN WELCHE JEDE WIRKUNGSVOLLE STRATEGIE IM WIEDERHOLTEN GEFANGENENDILEMMA AUSZEICHNEN [11].

D. Funktionsweise vom Choking-Algorithmus

Eine Kooperation im Sinne von Tit for Tat stellt im BitTorrent System einen Upload eines Chunks zu einem anderen Peer dar. Nicht kooperieren würde eine Ablehnung dieses Uploads bedeuten. Diese Kooperation und nicht Kooperation wird anhand der Downloadrate bestimmt. Peer A lädt von einer gewissen Anzhal von Peers jeweils mit einer gewissen Geschwindigkeit. Eine bestimmte Zahl anderer Peers wiederum möchten von Peer A herunterladen. Peer A muss nun aus diesem Pool vier auswählen, mit welchen er kooperiert. Peer A kontrolliert nun zuerst, ob auch er von diesen Peers herunterlädt und berechnet die Downloadgeschwindigkeit. A wird mit diesen Peers kooperieren, von welchen er am schnellsten herunterlädt. Lädt er von keinem herunter, passiert die Auswahl zufällig. Die Bestimmung der Downloadrate gestaltet sich als nicht ganz so einfach, wie man zunächst annehmen würde, weil die

Übertragungsrate relativ starken Schwankungen unterliegen kann. Die Downloadrate eines Peers könnte zum Zeitpunkt, in dem er angenommen wurde kurzzeitig gut sein und mit der Zeit einbrechen. Um zu verhindern, dass die Verbindung mit diesem Peer zu lange aufrechterhalten bleibt, berechnet ein Peer seine angenommenen Peers alle 10 Sekunden neu. 10 Sekunden reichen aus um neuen TCP-Verbindungen die Chance zu geben ihre volle Übertragungsrate zur Geltung zu bringen. Die Downloadrate kann anhand bereits laufender Downloads bestimmt werden. An diesem Punkt kann man die Analogie zu Tit for Tat erkennen. Lädt Peer A von Peer B so darf auch Peer B von Peer A laden [8].

E. Snubbed?

Wenn sich nun A für die schnellsten vier entschieden hat, kann es vorkommen, dass einem fünften Peer B der Download verweigert wird, obwohl er zu A uploadet. Es kann vorkommen, dass an Peer B nur vier Anfragen ankommen, was bedeutet, dass diese alle ohne Beachtung der Kriterien angenommen werden. B wird A also nicht dafür bestrafen, dass A den Download verweigert. Um in dieser Situation dennoch eine Bestrafung einzuführen handelt das BitTorrent System folgendermaen. Ein Peer betrachtet sich als abgewiesen (snubbed) von einem anderen Peer, wenn nach über einer Minute immer noch keine Verbindung zustande kommt. In diesem Fall wird auch Peer B die Verbindung zu Peer A unterbrechen. Wenn nun dasselbe auf der Seite von A passiert, wäre die Verbindung zwischen diesen beiden Punkten des Netzes verloren, weil nach Tit for Tat keine Möglichkeit mehr besteht, wie die Verbindung wieder aufgenommen werden kann, was ein erheblicher Nachteil für das gesamte Netzwerk darstellen kann. [8].

F. Optimistic Unchoke

Aus oben beschriebenem Grund hat der Choking Algorithmus eine Erweiterung zu Tit for Tat. Jeder Peer wählt zusätzlich zu den vier Peers, welche er annimmt, einen zusätzlichen aus dem Pool anfragender Peers aus, welchen er annimmt, ohne die Downloadgeschwindigkeit zu ihm zu überprüfen. Es handelt sich hierbei um eine otimistische Annahme (optimistic unchoke). Diese optimistische Annahme wird alle 30 Sekunden getätigt und bleibt auch für diese 30 Sekunden erhalten. Ein weiterer Grund für die optimistische Annahme ist, dass der erste Teil des Choking-Algorithmus keine Möglichkeit bietet neue, schnellere Verbindungen zu finden. Auch dies soll hierdurch gelöst sein [8].

G. Upload Only

Sobald ein Peer eine Datei komplett heruntergeladen hat, und sein Client-Programm nicht durch den Benutzer beendet wird, bietet er die Chunks weiterhin zum Upload an. Da er nicht mehr runterlädt hat er zu diesem Zeitpunkt allerdings keine Downloadraten mehr als Eingaben für den Choking-Algorithmus. BitTorrent betrachtet dann nicht mehr die Download- sondern die Uploadraten. Dadurch wird die volle Uploadkapazität ausgenutzt [8].

VI. Zusammenfassung und Ausblick

In den ersten Jahren wurde BitTorrent von der Medienindustrie als Feind angesehen, Trackerseiten wurden ausgeschalten, ein regelrechter Krieg gegen P2P-Systeme wurde eröffnet. Dem ist heute nicht mehr so. Firmen Präsident Ashwin Navin erklärt, dass der bereits 2005 fertig entwickelte Delivery Network Accellerator (DNA) erst 2007 veröffentlicht wurde da das Unternehemen vorher in einem schlechten Licht stand. Erst als BitTorrent mit 50 Medien Unternehmen im BitTorrent Entertainment Network Frieden geschlossen hatte, sah sich das Unternehmen bereit DNA zu veröffentlichen. DNA ermöglicht es Downloads in Teilen über ein P2P System abzuwickeln und verteilt dadurch die Last auf die Nutzer. Es wird zum Beispiel bei Video-Streaming eingesetzt.
Ab Version 6.0 ist BitTorrent Closed Source und daran soll sich auch nichts mehr ändern. Dieser Schritt sei laut BitTorrent-Präsident Ashwin Navin nötig geworden, da es öfters zu Problemen mit Drittanbietern gekommen sei. Diese sollen BitTorrent teilweise um Spyware ergänzt und unter eigenem Namen vertrieben haben.
Als Schwäche von BitTorrent könnte man anführen, dass nur wenig gegen das Verschwinden von unpopulären und älteren Dateien unternommen wird. Dies ist laut BitTorrent-Entwickler Bram Cohen aber auch Teil der Philosophie hinter BitTorrent: die populärsten Inhalte sollen sich am schnellsten verbreiten. In Zukunft soll aber wohl der Hauptclient (Client, der der erste Seeder einer Datei ist) Inhalte länger behalten, bzw verstärkt Inhalte anbieten, die zu verschwinden drohen.
Mit über 160 Millionen installierten Clients bietet BitTorrent ein mächtiges Werkzeug um Inhalte zu verbreiten. Seine Einfachheit hat sicherlich stark an der Verbreitung mitgewirkt. Aber nicht nur die Einfachheit sondern auch Eigenschaften wie Lastverteilung und Performance lassen BitTorrent zu einem der effektivsten P2P Systeme werden.
[13], [14]

VII. Begriffserklärung

Tracker: Server-Programm, welches es den Clients ermöglicht sich untereinander zu finden.
Chunk: BitTorrent teilt Dateien in kleinere Einheiten, genannt Chunks ein.
Sworm: Menge der Clients, welche an der selben Datei interessiert sind.
Seeder: Client, welcher eine vollständige Datei besitzt.
Peer: Client, der eine Datei sowohl verteilt, als auch herunterlädt.
Leecher: Zu diesem Begriff gibt es verschiedene Definitionen. Manche Quellen geben an es handle sich um einen Client, der nur herunterlädt und nicht weiterverteilt [6], andere geben an es sei dasselbe wie ein Peer [7].
.torrent: Datei, welche Informationen über die Datei, sowie über den Tracker enthält
choke, Choking: Die Zurückweisung eines anfragenden Clients
Sub-Piece: Teil eines Chunks
Pareto-Effizienz: Zustand, in dem es nicht möglich ist, ein

Individuum besser zu stellen, ohne zugleich ein anderes Individuum schlechter zu stellen.
snubbed: Zustand, den ein Peer einem anderen zuweist, wenn er ihm länger als eine Minute den Download verweigert.

LITERATUR

[1] BitTorrent.org, "Bittorrent.org," http://www.bittorrent.org, 2009.
[2] B. Cohen, "The bittorrent protocol specification," http://bittorrent.org/beps/bep_0003.html, 2008.
[3] J. Ihlenfeld, "Bittorrent überholt edonkey," http://www.golem.de/0610/48522.html, 2006.
[4] BitTorrent.Inc., "Bittorrent inc." http://www.bittorrent.com/, 2009.
[5] Wikipedia.org, "List of bittorrent clients," http://en.wikipedia.org/wiki/List_of_BitTorrent_clients, 2009.
[6] ——, "Leechen," http://de.wikipedia.org/wiki/Leecher, 2009.
[7] ——, "Leech," http://en.wikipedia.org/wiki/Leech_(computing), 2009.
[8] B. Cohen, "Incentives build robustness in bittorrent," http://bittorrent.org/bittorrentecon.pdf, May 2003.
[9] Wikipedia.org, "Pareto efficiency," http://en.wikipedia.org/wiki/Pareto_efficiency, 2009.
[10] A. Arbia, "Gefangenendilemma," http://www.scienceblogs.de/zoonpolitikon/2008/04/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma.php, 2008.
[11] C. Meredith, "Tit for tat," http://www2.owen.vanderbilt.edu/mike.shor/Courses/GTheory/docs/Axelrod.html, 1998.
[12] Wikipedia.org, "Tit for tat," http://en.wikipedia.org/wiki/Tit_for_tat, 2009.
[13] J. Ihlenfeld, "Bittorrent setzt künftig auf closed source," http://www.golem.de/0708/54016.html, 2007.
[14] ——, "Bittorrent richtet sich neu aus," http://www.golem.de/0710/55486.html, 2007.

Individuum besser zu stellen, ohne zugleich ein anderes Individuum schlechter zu stellen.
snubbed: Zustand, den ein Peer einem anderen zuweist, wenn er ihm länger als eine Minute den Download verweigert.

LITERATUR

[1] BitTorrent.org, "Bittorrent.org," http://www.bittorrent.org, 2009.
[2] B. Cohen, "The bittorrent protocol specification," http://bittorrent.org/beps/bep_0003.html, 2008.
[3] J. Ihlenfeld, "Bittorrent überholt edonkey," http://www.golem.de/0610/48522.html, 2006.
[4] BitTorrent.Inc., "Bittorrent inc." http://www.bittorrent.com/, 2009.
[5] Wikipedia.org, "List of bittorrent clients," http://en.wikipedia.org/wiki/List_of_BitTorrent_clients, 2009.
[6] ——, "Leechen," http://de.wikipedia.org/wiki/Leecher, 2009.
[7] ——, "Leech," http://en.wikipedia.org/wiki/Leech_(computing), 2009.
[8] B. Cohen, "Incentives build robustness in bittorrent," http://bittorrent.org/bittorrentecon.pdf, May 2003.
[9] Wikipedia.org, "Pareto efficiency," http://en.wikipedia.org/wiki/Pareto_efficiency, 2009.
[10] A. Arbia, "Gefangenendilemma," http://www.scienceblogs.de/zoonpolitikon/2008/04/spieltheorie-einfach-erklart-i-einleitung-und-gefangenendilemma.php, 2008.
[11] C. Meredith, "Tit for tat," http://www2.owen.vanderbilt.edu/mike.shor/Courses/GTheory/docs/Axelrod.html, 1998.
[12] Wikipedia.org, "Tit for tat," http://en.wikipedia.org/wiki/Tit_for_tat, 2009.
[13] J. Ihlenfeld, "Bittorrent setzt künftig auf closed source," http://www.golem.de/0708/54016.html, 2007.
[14] ——, "Bittorrent richtet sich neu aus," http://www.golem.de/0710/55486.html, 2007.

ISBN 3-937201-06-8

ISSN 1868-2634 (print)
ISSN 1868-2642 (electronic)