Network Architectures
and Services
NET 2011-05-2

FI & IITM
WS 10/11

Proceedings of the Seminars
Future Internet (FI) and
Innovative Internet Technologies and Mobile Communications (IITM)

Winter Semester 2010/2011

Munich, Germany, 04.10.2010 – 13.02.2011

Editors

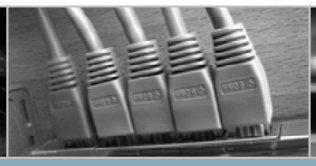
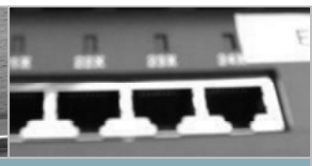
Georg Carle, Corinna Schmitt

Organisation

Chair for Network Architectures and Services
Department of Computer Science, Technische Universität München

Technische Universität München



Network Architectures
and Services
NET 2011-05-2

FI & IITM
WS 10/11

Proceedings zu den Seminaren
Future Internet (FI) und
Innovative Internettechnologien und
Mobilkommunikation (IITM)

Wintersemester 2010/2011

München, 04.10.2010-13.02.2011

Editoren: Georg Carle, Corinna Schmitt

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

Technische Universität München

Proceedings of the Seminars
Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM)
Winter Semester 2010/2011

Editors:

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

Cataloging-in-Publication Data

Seminar FI & IITM WS 2010/2011
Proceedings zu den Seminaren „Future Internet“ (FI) und „Innovative Internettechnologien und Mobilkommunikation“ (IITM)
München, Germany, 04.10.2010-13.02.2011
Georg Carle, Corinna Schmitt
ISBN: 3-937201-19-X

ISSN: 1868-2634 (print)
ISSN: 1868-2642 (electronic)
DOI: 10.2313/NET-2011-05-2
Lehrstuhl Netzarchitekturen und Netzdienste (I8) NET 2011-05-2
Series Editor: Georg Carle, Technische Universität München, Germany

Vorwort

Wir präsentieren Ihnen hiermit die Proceedings zu den Seminaren „Future Internet“ (FI) und “Innovative Internettechnologien und Mobilkommunikation” (IITM), die im Wintersemester 2010/2011 an der Fakultät Informatik der Technischen Universität München stattfanden.
Im Seminar FI wurden Beiträge zu unterschiedlichen Fragestellungen aus den Gebieten Internettechnologien und Mobilkommunikation vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- Mobilität in 3G und 4 G Netzen
- Flexibilität in Future Networks durch Service Oriented Architectures (SOAs) und selbstbeschreibende Mikroprotokolle
- Overlay Convergence Architecture for Legacy Applications (OCALA)
- Integration von Sensornetzen ins Internet
- Sicherheit, Datenschutz und Cloud Computing

Im Seminar IITM wurden Vorträge zu verschiedenen Themen im Forschungsbereich Sensorknoten vorgestellt. Die folgenden Themenbereiche wurden abgedeckt:

- MGRP – Passiv-aggressive Messungen
- Verständnis und Verwendung von „Balanced Security Scorecards“
- Sicherheit bei Cloud-Computing
- Clickjacking – Angriffe auf Webseiten
- Abhängigkeitsanalyse durch Verkehrsmessungen
- Stormbot – Ein Botnetz im Detail
- Digitale Signalmodulationsschemen

Wir hoffen, dass Sie den Beiträgen dieser Seminare wertvolle Anregungen entnehmen können. Falls Sie weiteres Interesse an unseren Arbeiten habe, so finden Sie weitere Informationen auf unserer Homepage http://www.net.in.tum.de.

München, Mai 2011

Georg Carle

Corinna Schmitt

Preface

We are very pleased to present you the interesting program of our main seminars on “Future Internet” (FI) and “Innovative Internet Technologies and Mobil Communication” (IITM) which took place in the winter semester 2010/2011.

In the seminar FI we deal with issues of Future Internet. The seminar language was German, and the majority of the seminar papers are also in German. The following topics are covered by this seminar:

- Mobility in 3G and 4G Networks
- Flexibility in Future Networks using Service Oriented Architectures (SOAs) and self-characterizing Micro-Protocols
- Overlay Convergence Architecture for Legacy Applications (OCALA)
- Integration of Sensor Networks into the Internet
- Security, Privacy and Cloud Computing

In the seminar IITM talks to different topics in innovate internet technologies and mobile communications were presented. The seminar language was German, and also the seminar papers. The following topics are covered by this seminar:

- MGRP - passive aggressive measurements
- Understanding and using balanced security scorecards
- Security in Cloud Computing
- Clickjacking – Attacks on websites
- Dependency analysis via network measurements
- Stormbot – A botnet in detail
- Digital Signal Modulation Schemes

We hope that you appreciate the contributions of these seminars. If you are interested in further information about our work, please visit our homepage http://www.net.in.tum.de.

Munich, May 2011

Seminarveranstalter

Lehrstuhlinhaber

Georg Carle, Technische Universität München, Germany

Seminarleitung

Corinna Schmitt, Technische Universität München, Germany

Betreuer

Tobias Bandh, *Technische Universität München, Wiss. Mitarbeiter I8*
Lothar Braun, *Technische Universität München, Wiss. Mitarbeiter I8*
Stephan Günther, *Technische Universität München, Wiss. Mitarbeiter I8*
Dirk Haage, *Technische Universität München, Wiss. Mitarbeiter I8*
Ralph Holz, *Technische Universität München, Wiss. Mitarbeiter I8*
Heiko Niedermayer, Technische Universität München, Wiss. Mitarbeiter I8
Johann Schlamp, *Technische Universität München, Wiss. Mitarbeiter I8*
Corinna Schmitt, *Technische Universität München, Wiss. Mitarbeiterin I8*
Matthias Wachs, *Technische Universität München, DFG Emmy Noether Research Group Member*

Kontakt:

{carle,schmitt,bandh,braun,guenther,haage,holz,niedermayer,schlamp,wachs}@net.in.tum.de

Seminarhomepage

http://www.net.in.tum.de/de/lehre/ws1011/seminare/

Inhaltsverzeichnis

Seminar Future Internet

Session 1: Mobilkommunikation

Session 2: Internet-Technologien

Seminar Innovative Internettechnologien und Mobilkommunikation

Session 1: Meßverfahren

Session 2: Sicherheit

Mobilität in 3G und 4G Netzen

Bernhard Adam
Betreuer: Tobias Bandh
Seminar Future Internet WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: adam@in.tum.de

KURZFASSUNG

Diese Arbeit beschäftigt sich mit den Abläufen beim 3G und 4G Handover. Es werden die Mittel vorgestellt, mit denen ein Mobilfunknetz die Mobilität eines Gerätes bereitstellt, also eine kontinuierliche logische Verbindung trotz Zellwechsel aufrecht hält. Außerdem werden sowohl für 3G als auch für 4G Netze die verschiedenen Handover-Methoden, sowohl bei ruhender als auch bei aktiver Verbindung aufgezeigt. Am Ende werden die Technologien kurz verglichen.

Schlüsselworte

Mobility Management, 3G, 4G, Handover, UMTS, LTE

1. EINLEITUNG

Die sehr schnelle Entwicklung in der Mobilfunkbranche lässt sich schwer detailliert verfolgen. Laufend erscheinen neue Technologien und Produkte mit immer mehr Funktionalität. Doch die Grundlagen der Netzstruktur verändern sich nicht so schnell. So ist eine der wichtigsten und angenehmsten Funktionen eines Mobilfunknetzes die Mobilität, die einem das Netz bietet. Selbst in einem Hochgeschwindigkeitszug kann man ohne Unterbrechung während der Fahrt telefonieren, die aktive Verbindung reisst nicht ab. Wie dies möglich ist wird hier nun an dem aktuellen Beispiel 3G und 4G Netzen gezeigt.

2. 3G NETZE

Mit der dritten Generation (3G) von Mobilfunknetzen verbindet man meistens direkt das Universal Mobile Telecommunications System, kurz UMTS. Als Weiterentwicklung des 2G Netzes ersetzt es GSM und die dazu gehörenden Standards. Um zu verstehen, wie einem User die Mobilität in einem 3G Netz gewährleistet wird, betrachte man zunächst den groben Aufbau eines 3G Netzes (Vgl. Abb. 1). Das mobile Gerät, das als Endpunkt der Verbindung dient, wird User Equipment (UE) genannt. Dieser Terminal muss die für die Mobilität nötigen Aufgaben, wie z.B. die Verbindungsübergabe, beherrschen. Das UE verbindet sich über die Sende- und Empfangsantennen zunächst mit einem Node-B. Ein Node-B kann hierbei mehrere Antennen verwenden, deren Sendebereich jeweils eine sog. Zelle darstellen. Die Hauptaufgabe eines Node-Bs ist die Leistungsregelung der Datenübertragungen. Die nächste Station ist der Radio Network Controller (RNC). Dieser ist mit mehreren Node-Bs verbunden und spielt eine zentrale Rolle für das Mobilitätsmanagement. Er ist, wie in den nachfolgenden Kapiteln beschrieben, für viele Entscheidungsprozesse bezüglich der Verbindung

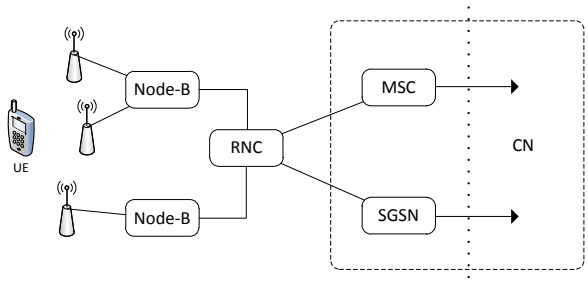


Abbildung 1: Grober Aufbau eines 3G Netzes

nötig. Der RNC ist dann mit dem Kernnetz, auch Core Network (CN) genannt, verbunden, welches die Zusammenfassung der restlichen Komponenten eines 3G Netzes darstellt. Es übernimmt die Verarbeitung und Interpretation des Datenverkehrs. Die ersten Komponenten des CN, die am Mobilitätsmanagement teilhaben, sind das sog. Mobile Switching Centre (MSC), das für die Verbindungen per Leitungsvermittlung zuständig ist, sowie der Serving General Packet Radio Service Support Node (SGSN), der für die Paketverbindungen zuständig ist. Das CN beinhaltet noch viele weitere Komponenten, die aber für das Mobilitätsmanagement keine direkte Rolle spielen und somit im CN zusammengefasst bleiben. Die einzelnen Komponenten nutzen für ihren Austausch untereinander jeweils verschiedene Interfaces, die eine für den Zweck optimal angepasste Verbindung bieten.

2.1 Handover in 3G Netzen

Bewegt sich ein eingeschaltetes UE durch ein 3G Netz, so soll sichergestellt werden, dass die Verbindung zum Netz so gut wie möglich aufrecht erhalten wird. Da der Bereich eines Netzes natürlich nicht mit einer einzigen Zelle abzudecken ist, passiert es zwangsläufig, dass sich der Terminal im Laufe seiner Bewegung aus der Reichweite einer Zelle heraus, in Reichweite einer anderen bewegt. Damit hierbei die Verbindung nicht abreißt, wie z.B. bei einem WLAN, das die für einen solchen Wechsel erforderliche Protokollstruktur nicht mitbringt, wird ein sog. Handover durchgeführt. Hierbei wird eine Verbindung über den Zellenwechsel hinaus aufrecht erhalten. Wichtig ist dabei, in welchem Zustand sich das System befindet.

doi: 10.2313/NET-2011-05-2_02

2.2 Handover im Leerbetrieb

Ist keine aktive Verbindung zwischen dem UE und dem Netz aufgebaut, so befindet sich der Terminal im sog. Idle State. Dies bedeutet, dass weder über das MSC noch über das SGSN Daten verschickt oder empfangen werden, jedoch das UE mit dem Netz verbunden ist. Außerdem kann sich das Gerät in einem Packet Data Protocoll (PDP) Kontext befinden. Dies bedeutet, dass dem mobilen Gerät eine globale IP-Adresse zugewiesen wurde, es also Teil des Netzes ist, aber kein aktiver Datentransfer besteht. Die einzige Aktivität des UEs ist in diesem Zustand, den Paging Channel (PCH) abzuhören, um z.B. einen eingehenden Anruf zu empfangen. Über den PCH werden sog. Paging-Messages vom Netz verschickt, die den Terminal anhand seiner eindeutigen Identifikation (International Mobile Subscriber Identity IMSI oder Temporary Mobile Subscriber Identity TMSI) über eingehende Daten informieren. Erhält ein Gerät eine Paging-Message, die seine IMSI oder TMSI enthält, baut es eine aktive Verbindung mit dem Netz auf. Da das ständige Abhören des PCH viel Energie des UEs verbraucht, teilt man alle Teilnehmer einem lokalen Bereich anhand ihrer IMSI in Gruppen (Paging-Groups) auf. Paging Messages für diese Gruppen werden dann nur in definierten Zeitintervallen gesendet, was dem UE die Möglichkeit gibt, in der restlichen Zeit den Empfänger auszuschalten und somit Strom zu sparen. Einziger Nachteil hier ist die verlängerte Dauer des Paging-Vorgangs im Gegensatz zur Dauer bei permanentem Abhören des PCH.

Befindet sich der Terminal nun im Idle-Zustand, so ist er allein für einen nötigen Zellenwechsel verantwortlich und muss eine neue Zelle wählen („Cell Reselection“). Das Netz hat nämlich keine Informationen über die Signalqualität beim UE, sondern nur Informationen darüber, in welchem Bereich sich das Gerät befindet. So ein Bereich kann aus bis zu mehreren Dutzend Zellen bestehen. Bewegt sich nun das UE in eine neue Zelle, die nicht Teil des aktuellen Bereiches ist, also in einen neuen Bereich, so muss eine aktive Verbindung aufgebaut werden, um die neuen Orts- und Routinginformationen auszutauschen. Die Verbindung kann hierbei über den Cell-DCH Modus, bei dem dem UE ein eigener Kanal zugewiesen wird, oder den Cell-FACH Modus, bei dem die Daten über ein geteiltes Medium verschickt werden, also mehrere Geräte den selben Kanal benutzen, erfolgen. Anschließend werden über die Verbindung die nötigen Daten zwischen UE und MSC bzw. SGSN ausgetauscht. Bei einer Paketverbindung über SGSN müssen außerdem noch Routing Informationen erneuert werden, da die Zellen in sog. Routing Areas eingeteilt sind, um die Weiterleitung der Pakete zu vereinfachen. Bezüglich des PGP-Kontextes ändert sich für das UE nichts, es hat immer noch dieselbe IP-Adresse, nur der Weg der Pakete hat sich verändert. Ist all dies geschehen, so wechselt der Terminal wieder in den Idle-State und der Vorgang ist abgeschlossen[2]. Viel kritischer als der eben beschriebene Handover im Idle-State ist natürlich ein Zellenwechsel bei einer aktiven Datenverbindung, also im Cell-DCH Status.

2.3 Handover bei aktiver Verbindung

Besteht eine aktive logische Verbindung, so ist es wichtig, diese auch bei einem Wechsel der Zelle so weit wie möglich aufrecht zu erhalten und gegebenfalls nötige Unterbrechungen der physikalischen Verbindung so kurz wie möglich zu gestalten. Im Cell-DCH State überwacht der RNC ständig die Verbindung zum UE in Bezug auf Signalstärke und Qualität. So sendet das Gerät mit den Daten auch immer Informationen über die Signalsituation bei sich mit. Der dem UE in diesem Zustand zugewiesene Kanal wird sowohl für Paket-, als auch für Leitungsverkehr genutzt. Bewegt sich der Terminal und der Wechsel zu einer anderen Zelle ist nötig, leitet der RNC diesen ein. Zu unterscheiden ist bei dem Handover im Cell-DCH Status vor allem, ob nur von einer Zelle in eine andere, die beide mit dem selben RNC verbunden sind, gewechselt wird oder ob zwischen zwei RNCs oder sogar zwei MSCs gewechselt wird. Wird nur die Zelle oder der Node-B gewechselt, so ist oft ein weicher Übergang (Soft Handover) möglich. Muss der RNC oder MSC gewechselt werden, so lässt sich dies meist nur durch einen harten Übergang (Hard Handover) realisieren.

2.3.1 Hard Handover

Hat der RNC einen Zellenwechsel entschieden, so müssen zunächst die benötigten Resourcen für die neue physikalische Verbindung reserviert werden. Dieser Vorgang ist mit dem eines neuen Verbindungsaufbaus zu vergleichen. Ist alles für die neue Verbindung bereit, so erhält das UE über die aktuelle Verbindung den Befehl zum Wechsel in die neue Zelle. Dieser Befehl enthält alle nötigen Informationen zum Tausch der Verbindung, wie die Frequenz, den Kanal und auch die verwendeten Scrambling Codes (diese dienen einer Kodierung des Datenstroms, um die verscheidenen Netzteilnehmer zu unterscheiden). Auf dieses Signal hin unterbricht der Terminal die alte physikalische Verbindung und baut sofort eine Neue auf. Da das Netz auf diese Verbindung vorbereitet ist, dauert dieser Übergang nur ca. 100ms[2]. Anschließend wird die Übertragung über die neue Verbindung fortgesetzt. Es ändert sich also nur die physikalische Verbindung über die die Daten fließen, nicht aber die logische Verbindung zum Kernnetz. Der harte Übergang ist also einem neuen Verbindungsaufbau sehr ähnlich, die alte Verbindung wird getrennt und eine neue, vorbereitete wird hergestellt. Dies ist vergleichbar mit einem Verbindungswechsel in alten 2G Netzen. Ein Novum in 3G sind die beschriebenen weichen Übergänge, bei denen die Verbindung nicht unterbrochen wird.

2.3.2 Soft Handover

Auch hier entscheidet der RNC auf Grundlage der Signalqualität der aktuellen und benachbarten Zelle, dass ein Zellenwechsel nötig ist und leitet den Soft Handover ein. Dabei wird die physikalische Verbindung des UE zum Netz nicht nur über eine Zelle, sondern über mehrere gleichzeitig hergestellt. Theoretisch werden dazu bis zu sechs Zellen in einem sog. „Active Set“ der Verbindung zusammengefasst, in der Praxis sind es meistens zwischen zwei und drei[2]. Das heißt, die Daten werden über alle Zellen gleichzeitig gesendet und empfangen. Geht die Verbingung zu einer Sende-/Empfangsstation verloren, so wird diese einfach aus dem Active Set entfernt. Da alle Zellen dieselben Daten vom UE empfangen, entscheidet der RNC anhand der Signalqualität bei jedem übertragenen Segment neu, welchen Datenstrom er an das CN weiterleitet, da das CN den Soft Handover selbst nicht unterstützt. Es besteht also immer nur eine logische Verbindung zwischen UE und CN. Das Management der redundanten physikalischen Verbindungen übernimmt der RNC. Die einzelnen Zellen des Active Set senden alle mit verschiedenen Scramblingcodes, was es dem UE ermög-

doi: 10.2313/NET-2011-05-2_02

licht, die Übertragungen der einzelnen Zellen zu unterscheiden. Dies erhöht natürlich den Dekodierungsaufwand beim Gerät, da die Daten mehrfach empfangen und verarbeitet werden müssen. Sind bei einem Soft Handover zwei oder mehr RNCs involviert, wird der Vorgang ein wenig komplexer. Der RNC, über den die Verbingung ursprünglich aufgebaut wurde, wird hier als Serving RNC (S-RNC) bezeichnet. Bewegt sich das UE in eine Zelle, die mit einem anderen RNC, dem sog. Drift-RNC (D-RNC), verbunden ist, so ist der Handover nur möglich, falls der S-RNC und der D-RNC verbunden sind. Um die Zelle des D-RNC in das active set aufzunehmen, muss sich der S-RNC über das sog. „I_{ur}“ Interface mit dem D-RNC verbinden. Die logische Verbindung zum CN besteht weiterhin über den S-RNC, der alle Daten dann nicht nur an seine Zellen, sondern auch an den D-RNC weiterleitet. Umgekehrt schickt der D-RNC alle vom UE erhaltenen Daten direkt an den S-RNC weiter. Hier entscheidet nun wieder der S-RNC anhand der Signalqualität, welchen Datenstrom er an das CN weitergibt. Dies funktioniert aber nur solange sich noch Zellen des S-RNC im Active Set befinden. Muss der S-RNC gewechselt werden, so ist ein Hard Handover nötig. Eine Variante des Soft Handover ist der sog. Softer Handover. Dieser findet statt, wenn verschiedene Zellen, die zu demselben Node-B gehören, in ein active set aufgenommen werden. Hier fällt schon der Node-B die Entscheidung über die Weiterleitung der Daten, was den RNC entlastet.

Ein weicher Übergang hat viele Vorteile gegenüber dem harten. So wird zum einen die physikalische Verbindung nie komplett unterbrochen, weshalb er eine komplett unterbrechungsfreie Übertragung ermöglicht. Desweiteren erhöht sich die Verbindungsqualität, da mehrere Zellen gleichzeitig zur Übertragung bereitstehen. Dies ist vor allem von Vorteil, wenn sich der Terminal in Bereichen bewegt, in denen häufig Sendeschatten auftreten, wie z.B in Städten. So führt ein plötzlicher starker Abfall der Signalqualität zu einer Antenne nicht gleich zu einem entsprechenden Abfall der Gesamt-Verbindungsqualität, da andere Zellen des Active Set diesen Qualitätsverlust eventuell ausgleichen können. Außerdem lässt sich der Energiebedarf des UE minimieren. Wenn sich der User in einen Bereich mit schlechtem Empfang bewegt, z.B. im Sendeschatten eines Gebäudes, so müsste das UE die Sendeleistung erhöhen. Erhält jedoch eine zweite Zelle des Active Sets noch ein gutes Signal, so ist eine Leistungserhöhung nicht notwendig. Zuletzt ist es einleuchtend, dass die Wahrscheinlichkeit für einen kompletten Verbindungsverlust geringer ist, wenn das UE mit mehreren Zellen in Verbindung steht.

Neben den hier vorgestellten Handover-Typen gibt es noch weitere Arten von Übergaben, die den Übergang von 3G in 2G Netze regeln. Diese sind ähnlich zu den vorgestellten Methoden, sollen jedoch hier nur erwähnt bleiben.

3. 4G NETZE

Als vierte Generation oder auch 4G Netze, bezeichnet man sog. Long Term Evolution (LTE) und System Architecture Evolution (SAE) Netze, zusammengefasst unter dem Begriff Evolved Packet System (EPS). Diese sind die Weiterentwicklung der 3G Architektur und sollen auch UMTS ersetzten. Der Aufbau eines EPS Netzes hat viele Gemeinsamkeiten mit dem eines 3G Netzes, jedoch hat vor allem die Tatsache, dass man in EPS keine Leitungsvermittlungsverbindungen mehr verwendet, viele Änderungen mit sich gebracht. Um die Abläufe zur Mobilität in 4G Netzen zu verstehen, wird zunächst der Aufbau der beteiligten Netzkomponenten vorgestellt (Vgl. Abb. 2). Der Unterschied zu einem 3G Netz

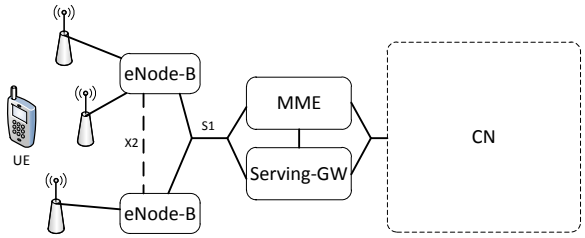


Abbildung 2: Grober Aufbau eines 4G Netzes

beginnt bei den Node-Bs. Die Funktionalität, die in einem 3G Netz der RNC übernimmt, wird hier teilweise von dem Enhanced Node-B (eNode-B) übernommen. Neu kommt hier hinzu, dass die eNode-Bs über das sog. X2-Interface verbunden sein können. Dies ist, wie man später sehen kann, auch bei der Mobilität von UEs im Netz von Vorteil. Ein eNode-B übernimmt die Aufgabe des Datenflusskontrolle und stellt die Quality of Service sicher. Außerdem führen sie die Handover aus. Die Verbindung zwischen dem Radio-Netzteil und dem CN bilden zwei neue Komponenten. Dies sind die Mobility Management Entity (MME) und der Serving Gateway (Serving-GW). Beide zusammen übernehmen die Rolle des SGSN in einem 3G Netzwerk. Da es nur noch Paketverkehr gibt, existiert kein MSC. Die Verbindung zwischen eNode-B und MME/Serving-GW geschieht über das S1-Interface. Die MME kümmert sich um die Verwaltung von Sessions, d.h. Handover zwischen eNode-Bs oder in andere Netze, z.B. 3G. Außerdem ist sie für die Überwachung der Position des UEs zuständig, wenn keine aktive Verbindung besteht. Der Serving-GW hat die Funktion, IP Pakete zwischen dem Internet und den mobilen Geräten weiterzuleiten.

3.1 Handover in 4G Netzen

Natürlich soll auch in einem 4G Netz sichergestellt sein, dass eine Verbindung nicht abreißt, wenn sich ein UE durch verschiedene Zellen des Netzes bewegt. Zu unterscheiden ist wieder, ob der Terminal gerade eine aktive Verbindung aufgebaut hat oder nicht.

3.2 Handover im Leerbetrieb

Besteht keine aktive Verbindung zum Netz, d.h. das UE befindet sich im sog. RRC_IDLE (Radio Resource Control Idle) Status, so wird zum Wechsel der Zelle eine Cell Selection durchgeführt. Dies wird vom UE alleine entschieden und geschieht auf Basis der Signalqualität der aktuellen und benachbarten Zelle. Wenn ein UE das Broadcast Signal einer neuen Zelle empfängt, entscheidet es selbständig, ob diese neue Zelle besser geeignet ist als die aktuelle. Dies passiert, neu bei 4G, auch nach bestimmten Kriterien. Ob eine neue Zelle gewählt wird, hängt nicht nur von der Stärke des Signals ab, sondern auch von einem Prioritätswert, der der Zelle gegeben werden kann. Somit lassen sich bessere Entscheidungen treffen, z.B. wird bei einem Handy das Netz des

doi: 10.2313/NET-2011-05-2_02

favorisierten Betreibers eher gewählt als ein anderes mögliches. Wenn eine neue Zelle gewählt wird, muss sich das UE dort wieder anmelden, dieser Vorgang kommt einer Neuanmeldung gleich. Wird die Zelle gewechselt, wird auch ein Positions-Update an die MME übermittelt um die Tracking Area zu aktualisieren. Die MME teilt alle Zellen, ähnlich wie bei 3G Netzen, in sog. Tracking Areas ein. Die Größe dieser Bereiche ist unterschiedlich. So muss bei einer größeren Tracking Area ein Positions-Update, also die Information, in welcher Area sich ein UE befindet, nicht so häufig durchgeführt werden. Eine kleinere Tracking Area verringert wiederum den Paging-Aufwand, um Geräte im Idle-Zustand über eingehende Daten zu informieren. Der Zellenwechsel im Leerbetrieb ist aber, wie bei 3G, unkomplizierter als bei aktiver Verbindung.

3.3 Handover bei aktiver Verbindung

Auch in 4G Netzen ist es ein wichtiges Ziel, eine aktive Verbindung möglichst verlustfrei aufrecht zu erhalten, selbst wenn sich ein User mit großer Geschwindigkeit durch das Netz bewegt. Bei einer aktiven Übertragung befindet sich das UE im sog. RRC_CONNECTED Status. In diesem Zustand entscheidet das Netz, ob und wann ein Handover durchgeführt wird und nicht das UE. Die Entscheidung des Netztes basiert dabei auf den Signalmessdaten, die das Netz vom UE erhält. Im Unterschied zu UMTS gibt es in EPS nur einen Hard Handover. Jedoch kann man diesen aufteilen in einen Handover über das S1 Interface und den über das X2 Interface.

3.3.1 S1 Handover

Diese Art des Handovers ist fast identisch mit dem 3G Hard Handover. Hierbei werden nach der Entscheidung zum Zellenwechsel alle Vorbereitungen getroffen, um eine neue physikalische Verbindung herzustellen. Sind die erforderlichen Resourcen für die Verbindung reserviert, sendet die MME das Kommando zum Handover. Neu in 4G ist jetzt das Senden von Statusinformationen vom aktuellen eNode-B zur MME, um einige Verbindungsparameter der Paketverbindung an den neuen eNode-B weiterleiten zu können. Auf den Handover Befehl hin unterbricht das UE die alte physikalische Verbindung und baut die neue, vorbereitete, auf. Der Handover wird nun vom UE an den neuen eNode-B bestätigt, der wiederum die entsprechende MME informiert. Der Handover ist abgeschlossen und der Datenverkehr kann über die neue Verbindung weiterfließen, die Resourcen der alten physikalischen Verbindung zwischen eNode-B und MME werden freigegeben. Die logische Verbindung zum Kernnetz bleibt hierbei während des ganzen Vorgangs intakt. Dieser Handover über S1 wird jedoch nur verwendet, wenn keine X2 Verbindung besteht, oder der eNode-B explizit für diese Art konfiguriert ist. In allen anderen Fällen wird ein X2-Handover durchgeführt.

3.3.2 X2 Handover

Der X2-Handover besteht aus denselben grundlegenden Schritten wie der S1-Handover. Der wichtigste Unterschied besteht darin, dass der Handover direkt zwischen den zwei beteiligten eNode-Bs durchgeführt wird. Der alte eNode-B entscheidet sich auf Grund der Signaldaten vom UE für den Handover. Jetzt baut der eNode-B über das X2 Interface eine Verbindung zum Ziel-eNode-B auf. Ist diese Verbindung hergestellt, kann der andere eNode-B den Handover-Befehl zum Ziel-eNode-B an das UE geben. Das UE ist nun mit dem neuen eNode-B verbunden. Das CN weiß jedoch noch nichts von dem Handover. Deswegen sendet der alte eNode-B alle vom CN kommenden Daten über das X2 an den neuen Knoten (Vgl. Abb. 3). Die Verbindung zum CN wird erst zum Schluss erneuert, dieser Vorgang heißt Late path switching. Durch die X2 Verbindung ist ein verlustloser Handover möglich. Der alte, oder Quell-eNode-B, sendet alle IP-Daten-Pakete, die noch nicht vom UE bestätigt wurden an den Ziel-eNode-B. Dieser leitet diese dann mit höchster Priorität an das UE weiter. Da es sein kann, dass das UE einige IP-Pakete empfangen hat, jedoch die Bestätigungen hierfür nicht mehr beim Quell-eNode-B angekommen sind, muss das UE in der Lage sein doppelte Pakete zu erkennen. Der Ziel-eNode-B würde sie nämlich erneut an das UE senden, da sie vom Quell-eNode-B als unbestätigt weitergeleitet werden. In der anderen Richtung muss das UE alle Pakete, die vor dem Wechsel nicht mehr vom alten Knoten bestätigt wurden, erneut übertragen.

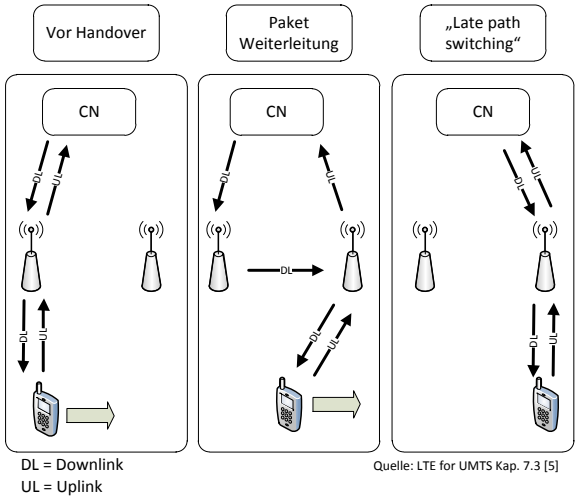


Abbildung 3: X2 Handover bei 4G

Neben den hier vorgestellten Handover Methoden existieren noch weitere, die dem Übergang in verschiedenartige Netze, z.B. 3G oder 2G, dienen. Diese sind ähnlich mit den genannten Methoden, unterscheiden sich jedoch natürlich in gewissen Punkten, da die Kernnetzkomponenten getauscht werden müssen. Eine genauere Beschreibung ist hier nicht vorgesehen.

4. VERGLEICH VON 3G UND 4G HANDOVER

Nicht nur der Aufbau der 3G und 4G Netze ist unterschiedlich, sondern auch die Art, wie sie die Mobilität innerhalb des Netzes sicherstellen ist anders. Beim Zellentausch ohne aktive Verbindung besteht der größte Unterschied in der Cell Reselection Prozedur, bei der in 4G nun auch die Priorisierung der Zelle beachtet wird. Dieses System wurde vor allem eingeführt, um der Entwicklung standzuhalten, dass verscheidene sog. Radio Access Technologies (RATs) im selben Netz vorhanden sind. Ein weiterer Unterscheid hier ist,

doi: 10.2313/NET-2011-05-2_02

dass bei 3G Netzen das Positionsupdate bei der Zellenwahl jeweils in Bezug auf die Positionsbereiche, in die die Zellen aufgeteilt werden, als auch in Bezug auf die in Kapitel 2.2 erwähnten Routing Areas, durchgeführt werden muss. Bei 4G reicht ein Update der Tracking Area.

Beim Handover mit aktiver Verbindung ist auffällig, dass bei 4G auf die in 3G gängigen Soft Handovers ganz verzichtet wird. Dieser ist jedoch bei einem 4G Netz wegen dem verbesserten Hard Handover über X2 und vor allem dadurch nicht nötig, da man sich auf Paketvermittlungsverbindungen beschränkt. Der verlustfreie Hard Handover in 4G steht dem Soft Handover in 3G in puncto Datenverlust in nichts nach. Da jedoch beim Soft Handover mehrere Zellen gleichzeitig die Verbindung unterhalten, lassen sich schnelle Schwankungen in der Signalqualität, wie z.B. durch Bewegung in Gebäuden, reduzieren. Dies geschieht jedoch unter großem Mehraufwand.

5. ZUSAMMENFASSUNG

In dieser Arbeit wurde gezeigt, wie Handover in UMTS und EPS Netzen funktionieren. Bei UMTS wurden die Funktionsweisen des Hard und Soft Handovers vorgestellt, sowie die Zellenauswahl ohne aktive Verbindung. Ebenso wurden die entsprechenden Handover-Methoden bei aktiver und passiver Verbindung eines EPS Netzes erklärt. Man sieht wie sich die Art der Verbindungsübergabe an die Übertragungsart, nämlich die Paketvermittlung, angepasst hat.

6. LITERATUR

[1] J. Schiller: *Mobilkommunikation*, 2., überarbeitete Auflage, ADDISON-WESLEY, 2003.

[2] M. Sauter: *Communication Systems for the Mobile Information Society*, WILEY, 2006.

[3] E.Dahlman, S. Parkvall, J. Sköld, P. Beming: *3G Evolution - HSPA and LTE for Mobile Broadband*, second edition, Academic Press, 2008.

[4] M. Sauter: *Beyond 3G - Bringing Networks, Terminals And The Web Together*, WILEY, 2009.

[5] H. Holma, A. Toskala: *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access*, WILEY, 2009.

[6] S. Sesia: *LTE - The UMTS Long Term Evolution*, WILEY, 2009.

[7] 3GPP Technical Specification 36.300 *E-UTRAN Overall Description; Stage 2*, v.10.0.0.

doi: 10.2313/NET-2011-05-2_02

Integration von Sensornetzen ins Internet

Ingmar Kessler
Betreuerin: Corinna Schmitt
Seminar Future Internet WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: kesslein@in.tum.de

KURZFASSUNG

Menschen und Maschinen benötigen Informationen, um Entscheidungen zu treffen, weshalb die Computerisierung und Automatisierung des Sammelns von Informationen durch Sensornetze großes Potential bietet. Sensornetze sind WLAN-Netzwerke aus mit Sensoren bestückten, günstigen, batteriebetrieben Mikro-Computern. Mit Hilfe von mehr und besseren Informationen können bessere Entscheidungen getroffen werden, sei es in Umweltfragen, im Krankenhaus oder zu Hause im Intelligenten Haus. Die gesammelten Informationen werden umso wertvoller, wenn man sie nicht nur an einem lokalen Rechner einsehen kann, sondern global über das Internet auf sie zugreifen kann. In Bezug auf Sensornetze ergeben sich durch die Anbindung an das Internet einige spezifische Herausforderungen in Sachen Effizienz, Zuverlässigkeit und Sicherheit. Im Folgenden sollen einige Ansätze und Techniken vorgestellt werden, die zeigen dass die genannten Herausforderungen überwindbar sind.

Schlüsselworte

Drahtloses Sensorsystem, Internet, 6LoWPAN, Sicherheit

1. EINLEITUNG

Die von Sensornetzen gelieferten Informationen lassen sich für vielerlei Zwecke nutzen. So können in einem Intelligenten Haus zum Beispiel Temperatur- und Lichteinstrahlungsdaten genutzt werden, um die Energieeffizienz des Hauses zu steigern, Sensoren können als Alarmanlage fungieren und ältere Menschen können von der Automatisierung des Hauses und dem Sammeln von gesundheitsrelevanten Informationen profitieren. Es macht Sinn, solche Sensornetze nicht nur in einem lokalen Netzwerk zu betreiben, sondern mit dem Internet zu verbinden. Zum Beispiel können ältere Menschen länger alleine, selbst bestimmt und sicher in ihrem eigenen Zuhause wohnen, wenn ihre Ärzte, Krankenhäuser und Verwandten global Zugriff auf diese häuslichen und gesundheitsrelevanten Informationen haben.

Der globale Zugriff auf Sensornetze und ihre Informationen stellt eine neue Herausforderung in Sachen Technik, Standardisierung, Benutzerfreundlichkeit und Zuverlässigkeit dar. Aber das wohl essentiellste Problem stellen die Fragen der Datensicherheit, Zugriffskontrolle und des Schutzes vor böswilligen Angriffen dar, denn umso wichtiger die Daten sind, die von Sensornetzen gesammelt werden, desto wichtiger ist es, sie vor unbefugtem Zugriff zu schützen.

In den folgenden Kapiteln soll gezeigt werden, wie diese Ziele trotz der technischen Einschränkungen von Sensornetzen erreicht werden können. Im nächsten Kapitel werden zuerst die grundlegenden Eigenschaften von Sensornetzen betrachtet und im Kapitel 3 werden mögliche Arten der Internetanbindungen vorgestellt. Im 4. Kapitel wird es um eine IPv6-Implementation für den Gebrauch innerhalb von Sensornetzen gehen und das 5. Kapitel wird auf einige Sicherheitsaspekte in Bezug auf Sensornetze und ihre Internetanbindung eingehen. Kapitel 6 soll einen Ausblick auf eigenständige Sensoren liefern, die direkt an das Mobilfunknetz angeschlossen werden. Anschließend folgt eine Zusammenfassung der Ergebnisse.

2. EIGENSCHAFTEN VON SENSORNETZEN

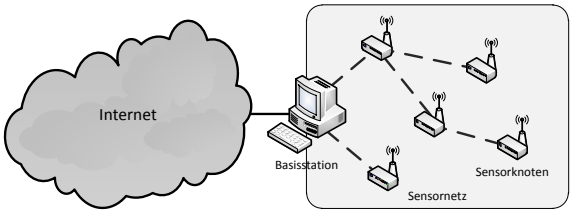


Abbildung 1: Aufbau eines Sensornetzes [1][3]

Sensornetze bestehen aus zwei Arten von Geräten: Einer Vielzahl von Sensorknoten, die Umweltdaten messen, und einer oder mehreren Basisstationen, an welche die gesammelten Daten übermittelt werden und die gegebenenfalls die zentrale Konfiguration übernimmt.

Sensorknoten sind kleine, leistungsschwache und günstige Geräte, die im Wesentlichen nur aus einem frei programmierbaren Rechenbaustein, einer WLAN-Antenne, einer Batterie und mindestens einem Sensor bestehen und von ihren Ressourcen her hauptsächlich der Klasse II aus Tabelle 1 zugeordnet werden können [1]. Ihr Zweck, als Datenquellen im Sensornetz, besteht allein darin die Sensordaten aufzunehmen und zu speichern, bis sie erfolgreich an die Basisstation übertragen wurden. Dabei können die Sensorknoten dauerhaft und in Echtzeit ihre Umwelt messen und so kontinuierliche Messreihen erstellen oder auch nur das Einhalten von bestimmten Schwellenwerten überprüfen und gegebenenfalls Alarm schlagen. Ebenso können sie nur bei Bedarf Messungen einholen oder sogar über ihre eigentliche Messfunktion hinausgehen und mit Aktuatoren statt Sensoren aktiv in ihre Umwelt eingreifen. Für die Datenübertragung müssen die Sensorknoten dabei nicht direkt mit der Ba-

doi: 10.2313/NET-2011-05-2_02

Tabelle 1: Klassifizierung von Sensorknoten [1]

Klasse	Taktrate	RAM	ROM	Energie
I	4 Mhz	1 kB	4-16 kB	1.5 mA
II	4-8 Mhz	4-10 kB	48-128 kB	2-8 mA
III	13-180 Mhz	256-512 kB	4-32 MB	∼40 mA

sisstation verbunden sein, sondern sie können mittels eines WLAN-Netzwerkes (meistens IEEE 802.15.4) über mehrere andere Knoten hinweg eine Verbindung mit der Basisstation aufbauen, was ihre Reichweite erhöht bzw. den Energiebedarf ihrer WLAN-Antennen senkt. Ein großer Vorteil von Sensorknoten ist, dass sie weder auf ein Daten- noch auf ein Stromkabel angewiesen sind und deshalb leicht überall platziert werden können, einschließlich entlegenen, unzugänglichen oder gefährlichen Orten. Die freie Platzierung im Raum bedeutet natürlich auch, dass sie über keine dauerhafte Stromquelle verfügen und dementsprechend auf Batterien angewiesen sind. Aufgrund der Arbeitskosten, die mit dem Batteriewechsel verbunden sind [8], und den giftigen bzw. schwierig zu entsorgenden Altbatterien wird auch an kleinen Stromquellen geforscht [2], die den Batteriewechsel überflüssig machen sollen. Momentan jedoch ist es notwendig nach einigen Tagen bis einigen Jahren [1] die Batterien der Sensorknoten auszutauschen, was natürlich entgegen des Ziels der automatischen Messung durch Sensornetze ist. Um diese Energieeffizienz, die ein ständiges Forschungsgebiet ist, überhaupt erst zu erreichen, sind einige Einschränkungen an die Rechenleistung notwendig. Außerdem werden aufgrund des hohen Strombedarfs einer WLAN-Antenne beim Senden, Empfangen und selbst nur beim passiven Lauschen die Antennen oft die meiste Zeit ausgeschaltet und nur sporadisch aktiviert um die gesammelten Daten zu senden bzw. weiter zu senden [3].
Basisstationen, die der Klasse III aus Tabelle 1 entsprechen, werden üblicherweise am Stromnetz betrieben und fungieren als Datensenke für das Sensornetz und als Router, die bei der Installation oder beim Ausfall eines Sensorknotens das Netzwerk automatisch konfigurieren. Diese automatische Konfiguration, welche die Sensorknoten auch teilweise selbstständig durchführen können, ist ein weiterer Reiz von Sensornetzen. Durch sie wird die Benutzerfreundlichkeit und die Redundanz des Netzes beim Ausfall einzelner Knoten gesteigert und damit auch die mit einer Sensornetz-Installation verbunden Personalkosten gesenkt. Die Basisstation kann wiederum an das Internet angeschlossen werden oder der Nutzer kann an ihr direkt die Messdaten ablesen.

3. ARTEN DER INTERNETANBINDUNG

Die Integration von Sensornetzen in das Internet kann Sinn machen, um mehr Menschen Zugriff auf die Daten zu gewähren oder um an Orten zu messen, an denen sich keine Menschen befinden. So werden Sensornetze in Island zur Gletschermessung und Klimaforschung verwendet [10] und in Nationalparks können sie bei Waldbränden die Reaktionszeiten der Löschkräfte senken [4]. Aufgrund der Beschränkungen von Sensornetzen und des Fehlens von Standards bietet die Internetanbindung von Sensornetzen einige spezielle Herausforderungen und Lösungen.

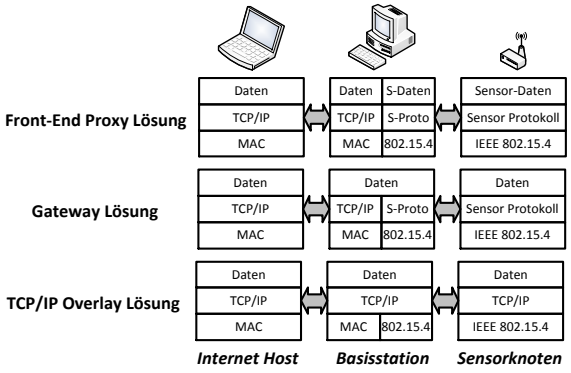


Abbildung 2: Integrationsstrategien von Sensornetzen ins Internet [1]

3.1 Front-End Proxy Lösung

Bei der Front-End Proxy Lösung gibt es keine direkten Verbindungen zwischen dem Internet und den Sensorknoten, sondern die Basisstation arbeitet als Interface und interpretiert die Kontrollbefehle und schickt sie dann an die Sensorknoten weiter. In der Gegenrichtung arbeitet die Basisstation zudem als Cache-Server und speichert die Sensorinformationen, die sie zum Beispiel als Web-Server dem Internet zur Verfügung stellt. Der große Vorteil daran ist, dass das Sensornetz damit nicht Teil des Internets ist und eigene Protokolle und Algorithmen verwenden kann. Zudem wird durch die strikte Trennung zwischen Internet und Sensornetz das Sicherheitsproblem auf das eines gewöhnlichen Web-Servers reduziert [1].

3.2 Gateway Lösung

Bei der Gateway Lösung kann das Sensornetz weiterhin eigenständige und anwendungsoptimierte Protokolle verwenden, aber eine Daten-Verbindung zwischen Internet-Hosts und Sensorknoten wird ermöglicht. Dies geschieht, indem die Basisstation auf der Anwendungsebene als Gateway fungiert und mittels einer Übersetzungstabelle Nachrichten und Adressen der darunterliegenden Schichten zum Beispiel von TCP/IP in ein spezialisiertes Protokoll übersetzt [1].

3.3 TCP/IP Overlay Lösung

Bei der TCP/IP Overlay Lösung verwenden die Sensorknoten TCP/IP, weshalb die Basisstation fast nur noch als gewöhnlicher Router fungieren muss und die Sensorknoten Teil des Internets werden und somit direkt mit Internet-Hosts Pakete austauschen können [1]. Da die Sensorknoten nur über geringe Ressourcen verfügen, wird das Sensornetz zwar IP-kompatibel betrieben, aber es werden sparsamere Protokolle wie 6LoWPAN verwendet, weshalb es notwendig ist, dass die Basisstation Standard-IPv4- bzw. Standard-IPv6-Pakete übersetzt. Ein weiterer Vorteil der TCP/IP Overlay Lösung, neben der direkten Kommunikation zwischen beliebigen Internet-Hosts und Sensorknoten, ist die Verwendung eines standardisierten und optimierten Protokolls wie IPv6, aus dem sich hoffentlich eine standardisierte und kompatible Art der Internetanbindung von Sensornetzen entwickelt.

doi: 10.2313/NET-2011-05-2_02

4. IP-BASIERENDE SENSORNETZE

Es macht für Sensornetze offensichtlich Sinn ein IP-Protokoll für die Kommunikation zwischen Knoten zu verwenden, wenn das Sensornetz mittels einer TCP/IP Overlay Lösung Teil des standardisierten und allumspannenden Internets werden soll. Aber auch Sensornetz-intern kann es gute Gründe geben, ein standardisiertes Protokoll wie IPv6 zu verwenden. So können die Entwicklungskosten gesenkt oder die Kompatibilität zwischen den Modellen verschiedener Generationen bzw. Hersteller verbessert werden. Zusätzlich handelt es sich bei IPv6 um ein ausgereiftes und sauber entworfenes Protokoll, dessen Leistungsfähigkeit sich auch mit anwendungsspezifischen Protokollen im Bereich der Sensornetze messen kann [3].

4.1 Gründe für IPv6

Hersteller und Anwender von Sensornetzen haben lange Zeit IPv4 und IPv6 zu Gunsten von anwendungsspezifischen Protokollen gemieden [3], was sich auf die geringen Ressourcen von Sensorknoten und den Wunsch nach anwendungsspezifischen Optimierungen zurückführen lässt. Auch spielte die große Anzahl an Knoten und der damit verbundene, nicht vertretbare Aufwand bei einer manuellen Konfiguration eine wichtige Rolle. Der Vorteil von IPv6 gegenüber IPv4 ist natürlich der größere Adressraum, der es theoretisch erlaubt, an jeden Sensorknoten eine eigene IP zu vergeben. Zudem wurden Protokolle wie ARP und DHCP Teil von IPv6 und erlauben zusammen mit Autoconf und ICMPv6 eine sparsame, automatische Konfiguration des Sensornetzes [3]. IPv6 bietet eine generalisierte, funktionsreiche Struktur um viele Anforderungen eines Sensornetzes zu erfüllen und wo Sensornetz-spezifische Lösungen notwendig sind, bietet es auch die notwendige Erweiterbarkeit beispielsweise über ein flexibles Header-Format.

4.2 IPv6 für Sensornetze

Obwohl IPv6 grundlegend für Sensornetze geeignet ist, ist es dennoch notwendig IPv6 für Sensornetze anzupassen, um ihr volles Potential auszuschöpfen. Deshalb soll hier die IPv6-Implementation namens 6LoWPAN vorgestellt werden, das von einer herstellerunabhängigen IETF Working Group entwickelt wird. Mit einem ROM-Bedarf von 24 kB und einem RAM-Bedarf von 4 kB [3] ist es auch auf Geräten mit geringen Ressourcen lauffähig ist und bietet trotzdem einen Grad an Zuverlässigkeit, Energieeffizienz und Allgemeinheit, der sich durchaus mit Sensornetz-spezifischen Protokollen messen kann [3]. Die Verwendung einer angepassten IPv6-Implementation ist nicht allein wegen den Header- und MTU-Größen notwendig, wird dort aber wohl am deutlichsten: Sensorknoten verwenden den WLAN-Standard 802.15.4, der nur Pakete mit maximal 127 Bytes unterstützt, und bei einem MAC-Header von maximal 25 Bytes, einer 128-Bit AES-Verschlüsselung, die 21 Bytes benötigt, einem IPv6-Header mit standardmäßig 40 Bytes und einem UDP-Header mit 8 Bytes bleiben nur 33 Bytes für die eigentlichen Daten übrig [5]. Nebenbei wird von IPv6-fähigen Geräten erwartet, dass sie eine Paketlänge von mindestens 1280 Bytes unterstützen, weshalb (IPv6-)Pakete, die länger als (81 bzw.) 127 Bytes sind, im Sensornetz ankommen können und auf der Sicherungsschicht fragmentiert werden müssen [5]. Das Header-Kompressionsverfahren von 6LoWPAN verwendet keinen expliziten Kontext, d.h. die Verbindungspartner müssen vor der Datenübertragung nicht erst die Art der Kompression aushandeln. Stattdessen ist die Kompression kontextfrei bzw. sie verwendet einen impliziten Kontext, indem angenommen wird, dass alle Knoten im Sensornetz ausschließlich 6LoWPAN verwenden. Damit kann der Overhead eines UDP/IP-Pakets von 48 Bytes auf 6 bis 25 Bytes reduziert werden. Dies geschieht, indem im IPv6-Header nur häufig verwendete Werte erlaubt werden, wie zum Beispiel dass Version "6" und Traffic Class und Flow Label "0" sein müssen und der nächste Header nur UDP, TCP oder ICMPv6 sein darf [3]. Zudem werden redundante Daten wie die Länge des Paketinhalts aus dem IPv6-Header entfernt und aus dem MAC-Header errechnet. Es werden außerdem, wenn möglich, kürzere IP-Adressen verwendet, die nur lokal gültig sind.

Neben der Header-Kompression wird das Router Advertisement im IPv6-Standard so modifiziert, dass Konfigurationsinformationen von Routern, d.h. Basisstationen, nicht nur an direkte Nachbarn sondern über mehrere Knoten hinweg verbreitet werden. Um einen hohen Netzwerk-Overhead zu vermeiden, werden in einem Trickle-Verfahren bei Änderungen bzw. neuen Informationen Router-Advertisement-Pakete häufiger (weiter-)verschickt und seltener wenn die Netzwerk-Konfiguration lange unverändert bleibt. DHCPv6 erlaubt eine automatische Adresskonfiguration für alle Sensorknoten, indem die Knoten ihre Anfragen an den zentralen Router verschicken und über mehrere anderen Knoten hinweg ihre Antwort erhalten.

Um eine möglichst hohe Energieeffizienz und Netz-Robustheit zu erreichen, werden Nachrichtenübertragungen auf einer Knoten-zu-Knoten-Basis behandelt. Dies bedeutet, dass die Vermittlungsschicht bei jedem Paketverlust neu evaluieren kann, ob wieder an den gleichen Knoten weitergesendet werden soll oder ob eine neue Route gewählt werden soll. Zudem wird ein Streaming-Prinzip verwendet, bei dem ein Sender auf dem ersten Datagramm kennzeichnen kann, ob sofort weitere Pakete folgen, weshalb nicht erneut die Verbindungskonditionen ausgehandelt werden müssen. Dadurch wird eine höhere Übertragungsrate erreicht, die es den Knoten erlaubt, mehr Zeit im Schlafmodus zu verbringen und so Energie zu sparen. Eine Stau-Kontrolle wird wiederum durch ein 'additive-increase and multiplicative-decrease'-Verfahren und indem Pakete nie fallen gelassen werden erreicht, wodurch bei einer vollen Warteschlange ein Rückstau über alle Knoten hinweg entsteht [3].

Das Routing in einem Sensornetz stellt eine besondere Herausforderung dar, da die Verbindungsqualität zwischen den Knoten und selbst ihre Positionen im Raum sehr variabel sein können. Zudem verfügen die Knoten nicht über genug Rechenleistung für komplexe Vermessungsverfahren und Extra-Traffic zum Auskundschaften des Netzwerkes ist zu vermeiden. Da man sich den damit verbundenen Energieverbrauch nicht leisten kann, werden die optimalen Routen oft nur mit eingeschränkten Informationen geschätzt. Die Basisstation unterhält eine Route zu jedem Sensorknoten, während die Knoten selbst nur die direkten Nachbarn kennen und eine Default-Route für die Basisstation und alle anderen Ziele unterhalten. Die Knoten und die Basisstation speichern dabei zu jedem Zeitpunkt nicht nur eine Route, sondern testen mehrere Alternativ-Routen gleichzeitig. Dabei senden sie einen Großteil ihrer Pakete über die Default-Route, die am billigsten ist und bei der die Zuversicht über die Kostenabschätzung am höchsten ist. Die restlichen Pakete werden über die Alternativ-

doi: 10.2313/NET-2011-05-2_02

Routen versendet, um ihre Zuversicht zu erhöhen und wenn möglich eine bessere Default-Route zu finden. Da das Sensor-Netzwerk nicht statisch ist, werden mit den Paketen die erwartete Anzahl der Hops und die benötigten Übertragungsversuche mit versandt. Eine Abweichung in der Anzahl der Hops kann auf eine Schleife hindeuten, während eine Abweichung in der Anzahl der benötigten Übertragungsversuche bedeuten kann, dass es sich um eine (in)effiziente Route handelt.
Da dem Sensornetz ein kompatibles IPv6-Protokoll zur Verfügung gestellt wird, können bis auf die Header-Komprimierung standardmäßige Transportprotokolle wie TCP und UDP verwendet werden.

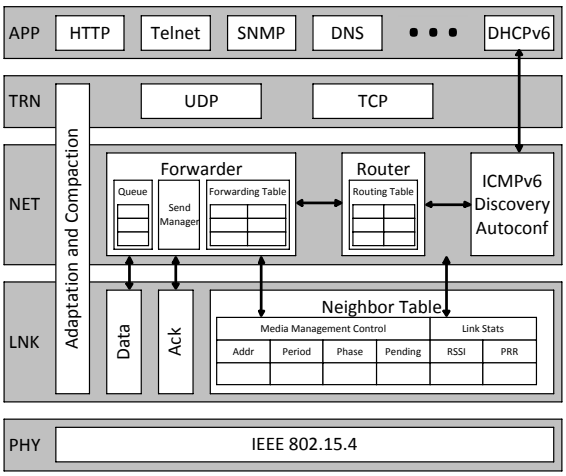


Abbildung 3: IP-Stack der Sensorknoten [3]

5. SICHERHEITSASPEKTE

Wenn Sensornetze mit oder ohne Internetanbindung in Zukunft eine tragende Rolle in militärischen bzw. zivilen Anwendungen spielen sollen, ist es notwendig sie vor störenden Umwelteinflüssen und Angriffen zu schützen. So sind zum Beispiel Sensornetze, die bei der Koordination von Rettungskräften helfen sollen, nutzlos oder sogar gefährlich, wenn sie keine Erdbeben oder Stürme überstehen können. Gleiches gilt für sicherheitskritische Sensornetze, die zur Kontrolle von Fahrzeugen, Produktionsanlagen, der Infrastruktur oder als Sicherheitssysteme für wichtige Gebäude verwendet werden und von Angreifern leicht ausgeschaltet werden können. Es ist schwierig, wenn auch nicht unmöglich, Sensornetze zu schützen, bei denen Angreifer physikalischen Zugriff zu einem oder mehreren Sensorknoten besitzen. Aber auch gegenüber Angreifern, die keinen physikalischen Zugriff besitzen, ist es wichtig einige Sicherheitsmaßnahmen zu treffen.
Für den sicheren Betrieb eines Sensornetzes sind innerhalb und außerhalb des Netzes vier wesentliche Punkte wichtig: Verschlüsselung gegen Abhörversuche, Authentifizierung gegen Angreifer, die sich als legitime Nutzer, Sensorknoten oder Basisstationen ausgeben, Autorisierung gegen unberechtigten oder für eingeschränkten Zugriff und Protokollierung um die Verantwortlichkeit von Nutzern zu garantieren. Selbst mit den geringen Ressourcen eines Sensorknotens ist eine 128-Bit AES-Verschlüsselung im WLAN bzw. eine SSL-Verschlüsselung zwischen einem Internet-Host und dem Sensornetz möglich [1], auch wenn dadurch natürlich der Rechen-, Speicher- und Energieverbrauch in einem gewissen Rahmen erhöht wird. Innerhalb des Sensornetzes kann die Authentifizierung und Autorisierung durch Public Key Kryptographie bzw. Zertifikate zumindest in kleinen Netzen sichergestellt werden. Dabei stößt man aber schon an die Grenzen der Leistungsfähigkeit von Klasse II Sensorknoten [1] und zudem wird die automatische Konfiguration durch den Austausch von Zertifikaten erschwert.
Die sichere Kommunikation zwischen Sensorknoten und dem Internet ist dagegen schwieriger herzustellen, da einerseits leichter größere Angriffe gestartet werden können, weshalb stärkere Verfahren notwendig sind und andererseits sehr viel mehr Authentifizierungs-, Autorisierungs- und Protokollierungsdaten gespeichert werden müssen, was die Speichergrenzen von typischen Sensorknoten sprengt. Aus diesem Grund kann es Sinn machen, die Leistungsfähigkeit und den zur Verfügung stehenden Speicher von Sensorknoten zu erhöhen, was auch anderen Funktionen und anderen Sicherheitsaspekten zu Gute kommen würde. Dadurch würden aber allerdings einige Vorteile von Sensorknoten wie ihr geringer Preis und ihre lange Batterielaufzeit, die sie durch ihren geringen Ressourcenbedarf erlangen, wieder verschwinden. Eine andere Alternative wäre zum Beispiel die Schlüssel dezentral unter den Knoten im Sensornetz zu verteilen [1], was allerdings kompliziert und wenig robust gegenüber dem Ausfall anderer Knoten ist und dem Gedanken von Sensorknoten als eigenständigen Internet-Hosts wiederum entgegen läuft.
Da sparsame Sensorknoten in jedem Fall den Betrieb einer leistungsfähigeren Basisstation notwendig machen, sei es auch nur zur Umwandlung von IPv4- oder IPv6-Paketen in 6LoWPAN-Pakete, macht es Sinn die schon gegebene Abhängigkeit weiter auszunutzen. So können aufwendige Aufgaben auf die Basisstation ausgelagert werden [1] und die Basisstation kann gegebenenfalls auch Funktionen weiter auf einen zentralen Web-Server auslagern. Die Gefahr einer Single-Point-of-Failure in der Basisstation wird damit zwar nicht nur aus technischer sondern auch aus sicherheitsrelevanter Sicht erhöht, aber andererseits kann die Basisstation mit ihrer Leistung eine stärkere, zentrale Sicherheit implementieren und unter den gegebenen Umständen erscheint die Lösung als guter Kompromiss. Außerdem ergeben sich weitere Vorteile, wie zum Beispiel dass die Basisstation historische Messdaten der Sensorknoten speichern kann, welche von den Knoten aufgrund des Speichermangels entweder gelöscht, zusammengefasst oder anderweitig komprimiert werden müssten [1]. Zusätzlich kann die Basisstation als Cache-Server agieren und so bei häufigen Anfragen schnelle Antworten liefern und den Sensorknoten so Energie sparen oder auch beim Ausfall eines Sensorknotens zumindest historische Daten und Fehlerdiagnoseinformationen liefern.
Diese Art der zentralen Sicherung eines Sensornetzes ist bei der Front-End Proxy Lösung besonders einfach, da die Basisstation auf standardmäßige und lange Zeit im Internet erprobte Verfahren für die sichere Kommunikation zwischen zwei gleichwertigen Internet-Hosts setzen kann und das Internet und das Sensornetz streng getrennt werden [1]. Aber auch bei der Gateway und der TCP/IP Overlay Lösung sind Hybrid-Verfahren denkbar, welche die Basisstation als 'Gatekeeper' einsetzen.

doi: 10.2313/NET-2011-05-2_02

5.1 DoS-Angriffe

Denial of Service (DoS) bedeutet, dass das Sensornetz seine zugewiesene Funktion nicht mehr erfüllen kann, was zum Beispiel durch eine zu große Anzahl gleichzeitiger, legitimer Nutzer oder durch den Ausfall von zu vielen Knoten durch Umwelteinflüsse passieren kann. Hier sollen aber bewusste Angriffe betrachtet werden, die darauf abzielen die Funktion eines Sensornetzes zu mindern oder komplett zu stören.

DoS-Angriffe können auf mehreren Schichten stattfinden, wobei ein Angriff auf physikalischer Ebene aus einem simplen Radiostörsender bestehen kann [6]. Die Existenz eines Störsenders ist leicht durch hohe und unablässige Aktivität auf der entsprechenden Frequenz zu erkennen. Die Knoten innerhalb des Störgebietes können nur versuchen, auf anderen, hoffentlich ungestörten Frequenzen oder auf mehreren Frequenzen gleichzeitig zu senden, was für Sensorknoten mit ihren geringen Ressourcen schwierig ist. Der Störsender kann auch nur kurze Störimpulse senden, was zu Übertragungsfehlern und somit zum erneuten Versenden von Paketen führt, wodurch die Batterien der Sensorknoten schnell geleert werden. Durch die kurzen Störimpulse spart der Störsender auch Strom und kann damit länger senden bzw. kleiner gebaut werden. Auch hier können die Sensorknoten die Störung nicht beseitigen, sondern nur mittels bestimmter Verfahren mindern [6]. Falls das Störgebiet nicht das gesamte Sensornetz umfasst, können Knoten außerhalb des Gebietes versuchen um das Gebiet herumzusenden und so den Ausfall des Sensornetzes möglichst gering zu halten. Da die Sensorknoten also nur eingeschränkt gegen das Störsignal vorgehen können, muss die endgültige Beseitigung des Störsignales erfolgen, indem man den Störsender findet und ausschaltet.

Auf der Transportschicht kann ein Sensorknoten oder eine Basisstation gestört werden, indem ihre Ressourcen durch einen Angreifer aufgebraucht werden, der unzählige Verbindungen aufbaut, sie aber nicht nutzt. Falls der Angreifer und das Opfer über ähnliche Leistung verfügen, kann das Opfer von potentiellen Kommunikationspartnern verlangen, ein Puzzle zu lösen, das leicht zu erstellen, aber schwer zu lösen ist, bevor eigene Ressourcen für den Verbindungsaufbau verwendet werden [6]. Legitime Kommunikationspartner verfügen über ausreichend Ressourcen um einige Puzzle zu lösen, was den Overhead zwar ein wenig erhöht, aber Angreifer, die versuchen hunderte bis hunderttausende Verbindungen aufzubauen, stoßen bald an ihre eigenen Leistungsgrenzen. Die Stärke des Puzzles kann auch mit dem Ausmaßdes Angriffs wachsen und so einerseits unnötigen Overhead vermeiden und andererseits stärkeren Angriffen widerstehen.

5.2 Firewalls

Firewalls können Computern, besonders solchen mit Internetanschluss, helfen sich vor ungewollter Kommunikation und Einbruchsversuchen zu schützen, weshalb Hardware- und Software-Firewalls Teil jedes Computers mit Internetanschluss sein sollten. Basisstationen von Sensornetzen bieten aufgrund ihrer ausreichenden Ressourcen eine Plattform, auf der Firewalls ein Sensornetz effektiv schützen können. Der Einsatz von Firewalls, die oft eine beträchtliche Menge an Rechenzeit benötigen, auf Sensorknoten scheint dagegen aufgrund der geringen Rechenleistung und Stromversorgung der Knoten als unrealistisch. Dennoch würden sie eine weitere Mauer zum Schutz der Sensorknoten bieten, falls die Firewall auf der Basisstation versagen sollte oder falls der Angriff nicht über das Internet sondern direkt über das WLAN erfolgen sollte. Diese Art des Angriffs ist damit zwar nicht spezifisch für Sensornetze mit Internetanbindung, aber es wird sie dennoch genauso betreffen und in unkontrollierbaren ländlichen und urbanen Gebieten werden wohl insbesondere Sensornetze mit Internetanschluss eingesetzt werden.

Die geringen Ressourcen von Sensorknoten sind zwar ein Hindernis für den Einsatz von Firewalls, aber die Entwicklung der AEGIS-Firewall [9] hat gezeigt, dass man die Vorteile von Firewalls auch auf beschränkten Geräten nutzen kann. Die AEGIS-Firewall ist in der Lage den ein- und ausgehenden Traffic eines Sensorknotens mit Hilfe einer kontextfreien Regelbasis zu filtern, welche die Sender- und Empfängeradresse, die Richtung und das sendende und empfangende Modul umfasst. Sie ist auf Sensorknoten lauffähig, da die entworfenen Regeln nicht in einer Datei auf dem Sensorknoten gespeichert werden und bei jedem Filtervorgang aufwendig geladen und ausgewertet werden müssen. Stattdessen erfolgt zuerst auf einem PC ein aufwendiger Optimierungsvorgang, bei dem die Regeln zuerst auf Konfliktfreiheit hin untersucht werden und dann als Ganzes zusammen mit der Firewall in speziell optimierten Bytecode kompiliert werden, der, nur als Beispiel, die Verschwendung von Rechenzeit durch redundante Regeln vermeidet. Eine erste Version der Firewall wurde schon erfolgreich in einem Sensornetz getestet, wobei sie nur 2.9 kB ROM-Speicher [9] benötigte. Sie verfügt zwar noch nicht über die Funktionen moderner Firewalls und IP-Spoofing stellt noch eine Schwachstelle dar, aber sie zeigt, dass Sensorknoten aufgrund ihrer geringen Ressourcen nicht kategorisch auf moderne Sicherheitstechniken verzichten müssen.

6. GPRS-SENSOREN

Mit Hilfe von Basisstationen können Sensorknoten viele Aufgaben und Funktionen eines Sensors bzw. Sensornetzes auslagern, wodurch sie mit geringen Ressourcen auskommen. Dadurch erreichen sie eine geringe Größe, geringe Kosten, eine lange Laufzeit der Batterie und zusammen mit der Basisstation ein bestimmtes (automatisches) Konfigurationsschema für das Sensornetz. Durch die Teilnahme eines Sensorknotens an einem solchen Sensornetz geht er jedoch auch eine geographische und funktionale Abhängigkeit zu einer Basisstation ein, die zu einem Single-Point-of-Failure wird. Zudem ist er für die Übertragung seiner Daten auch an das korrekte Funktionieren anderer räumlich naher Sensorknoten angewiesen. Auch aus Sicht der Konfiguration kann es Sinn machen, viele unabhängige, möglicherweise identische Sensoren, die direkt mit dem Internet verbunden sind, statt einem komplexen Sensornetz zu betreiben.

Beispielsweise bei mobilen und räumlich stark getrennten Sensoren kann es deshalb Sinn machen, sie eigenständig zu verwenden. Wie in den vorangegangen Kapiteln, insbesondere bei der TCP/IP Overlay Lösung, gezeigt wurde, reichen die beschränkten Ressourcen von typischen Sensorknoten aus, um sie als eigenständige Internet-Hosts zu betreiben. Außerdem sind auch geringe Erweiterungen der Ressourcen für weitere Funktionalitäten insbesondere in Anbetracht des ständigen technischen Fortschritts in der Computerbranche realistisch. Da die Sensoren und Rechenbausteine der Sensorknoten damit also weitgehend identisch zu denen in typischen Sensornetzen sein können, ist noch die Kommunikation

doi: 10.2313/NET-2011-05-2_02

mit dem Internet zu betrachten. Eine Internetanbindung über WLAN scheint aufgrund der großen Entfernungen bzw. des Mangels an Wireless Hotspots unrealistisch, weshalb die Verwendung des Mobilfunknetzes, das insbesondere in urbanen Umgebungen praktisch überall verfügbar ist, als passende Alternative erscheint. In der Tat wird GPRS schon als Alternative für die Festnetzanbindung von Sensornetzen an das Internet verwendet [4] und es gibt schon erfolgreiche Implementationen von eigenständigen Sensoren, die per GPRS mit einem zentralen Server im Internet kommunizieren und mit Hardware arbeiten, die ähnlich zu derer von gewöhnlichen Sensorknoten ist [7].
Eine offene Frage bleibt jedoch, ob sich die Kommunikation solcher Sensoren mit Hilfe eines GPRS-Modems energieeffizient genug gestalten lässt, um mit der von Sensorknoten erreichten Batterielaufzeit mithalten zu können. Dies ist besonders relevant, da es einige Anwendungsgebiete von Sensornetzen gibt, für die eine Batterielaufzeit von bis zu mehreren Jahren dringend erforderlich ist. Es ist außerdem klar, dass für die Mobilfunknetze eine geringe Anzahl von GPRS-Sensoren unerheblich ist, aber es ist fraglich, ob sie mehrere Millionen wenn nicht gar mehr verkraften können. Auch stellt sich die Frage, ob die Mobilfunkbetreiber willig sind passende Entgeltungspläne für Geräte mit einem geringen Datenaufkommen wie GPRS-Sensoren zu erstellen. Insbesondere für sicherheitskritische Sensoren, die zum Beispiel die Infrastruktur einer Stadt überwachen, ist es außerdem fraglich, ob Mobilfunknetze, die bei Großereignissen oder Katastrophen oft überlastet sind, eine ausreichende Zuverlässigkeit bieten können.
Bedingt durch diese Hindernisse macht es Sinn sich aufgrund einer Anforderungsanalyse für ein Sensornetz oder GPRS-Sensoren zu entscheiden. Natürlich können beispielsweise durch technische Entwicklungen, die eine höhere Batteriekapazität bzw. sparsamere GPRS-Modems ermöglichen, oder durch die Integration von Energiequellen wie zum Beispiel Solarzellen in die GPRS-Sensoren die möglichen Laufzeitprobleme gelöst werden. Die Belastung der Mobilfunknetze könnte auch gesenkt werden, indem die Sensoren auf eine GPRS-Verbindung verzichten und sich falls möglich in einem ad-hoc WLAN-Sensornetz zusammenschließen.
Die grundsätzliche Eignung von GPRS-Sensoren für bestimmte Anwendungen wurde schon gezeigt [7] und so können die Nützlichkeit und die Anwendungsgebiete von GPRS-Sensoren durch technische Entwicklungen und weitere Forschungen nur erweitert werden.

7. ZUSAMMENFASSUNG

Das große Interesse an Sensornetzen in der Forschung und in der technischen Entwicklungsgemeinde zeugen von der wichtigen Rolle, die Sensornetze im Alltag, in der Wirtschaft und in der Wissenschaft spielen werden. Aber schon heute gibt es weitreichende Produkte und Standards für Sensornetze im Heim- und Büroeinsatz für das Energiemanagement, im medizinischen Einsatz, für die Sicherung von ziviler Infrastruktur wie Brücken, für die Prozesssteuerung in der Fertigungstechnik, in der Automobilindustrie und in der Rohstoffförderung [8].
Ein Teil der wachsenden Bedeutung von Sensornetzen ist ihre Anbindung an das Internet und wie hier verdeutlicht wurde, ist die Umsetzbarkeit schon durch praktische Implementationen gezeigt worden, die auch schon teilweise im Produkteinsatz sind [8]. Es wurden ebenso durchaus relevante Implementierungs- und vor allem Sicherheitsaspekte angesprochen, die zwar schon Teillösungen besitzen, aber immer noch einigen Forschungs- und Entwicklungsbedarf besitzen. Aufgrund des anhaltend großen Volumens an Forschung im Bereich von Sensornetzen und des Marktwertes, ist zu erwarten, dass es keine größeren Hindernisse bei der Verbreitung von Sensornetzen mit Internetanbindung geben wird und dass alle Probleme mittelfristig zufriedenstellend gelöst werden können.

8. LITERATUR

[1] Rodrigo Roman, Javier Lopez, (2009) "Integrating wireless sensor networks and the internet: a security analysis", Internet Research, Vol. 19 Iss: 2, pp.246 - 259

[2] Robert Bogue, (2009) "Energy harvesting and wireless sensors: a review of recent developments", Sensor Review, Vol. 29 Iss: 3, pp.194 - 199

[3] Hui, J. W. and Culler, D. E. 2008. IP is dead, long live IP for wireless sensor networks. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (Raleigh, NC, USA, November 05 - 07, 2008). SenSys '08. ACM, New York, NY, 15-28. DOI= http://doi.acm.org/10.1145/1460412.1460415

[4] Santos, Jorge and Santos, Rodrigo and Orozco, Javier, (2009) "On the Feasibility of Early Detection of Environmental Events through Wireless Sensor Networks and the Use of 802.15.4 and GPRS", GeoSensor Networks, Vol. 5659, pp.122 - 130, Springer Berlin / Heidelberg

[5] Montenegro, et al., "IPv6 over IEEE 802.15.4", RFC 4944, IETF, September 2007

[6] A. D. Wood and J. Stankovic, (2002) "Denial of Service in Sensor Networks", IEEE Computer, Vol. 35 Iss: 10, pp.54 - 62

[7] A. R. Al-Ali, Imran Zualkernan, Fadi Aloul, (2010) "A Mobile GPRS-Sensors Array for Air Pollution Monitoring", IEEE Sensors Journal, Vol. 10 Iss: 10, pp.1666 - 1671

[8] Robert Bogue, (2010) "Wireless sensors: a review of technologies, products and applications", Sensor Review, Vol. 30 Iss: 4

[9] Hossain, Mohammad and Raghunathan, Vijay, (2010) "AEGIS: A Lightweight Firewall for Wireless Sensor Networks", Distributed Computing in Sensor Systems, Vol. 6131, pp.258-272

[10] Martinez, Kirk and Hart, Jane and Ong, Royan, (2009) "Deploying a Wireless Sensor Network in Iceland", GeoSensor Networks, Vol. 5659, pp.131 - 137

doi: 10.2313/NET-2011-05-2_02

Flexibilität in Future Networks durch Service Oriented Architectures (SOAs) und selbstbeschreibende Mikroprotokolle

Yannick Scherer
Betreuer: Heiko Niedermayer
Seminar Future Internet WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
scherery@in.tum.de

KURZFASSUNG
Die Verwendung eines standardisierten, festen Protokollstacks in Netzwerken geht mit vielen Vorteilen einher, darunter die Interoperabilität zwischen Systemen, sowie ein hoher Grad an Robustheit und Vorhersehbarkeit. Doch neue Technologien wie *Voice over IP (VoIP)* oder *Video on Demand (VoD)* stellen auch neue Ansprüche an die zugrundeliegende Architektur, wodurch die Anpassung alter und die Einführung neuer Protokolle notwendig wird. Es soll hier gezeigt werden, wie sich durch Abkehr vom verbreiteten Schichtenmodell hin zu Service Oriented Architectures, sowie durch einen Mechanismus, der an individuelle Bedürfnisse angepasste Protokollstacks erstellt, die Akzeptanz und Verwendung neuer Protokolle erhöht und vereinfacht, die Evolvierbarkeit und Flexibilität in zukünftigen Netzen also gesteigert werden kann.

Schlüsselworte
SOA, SOCS, Mikroprotokolle, Future Networks, Dynamic Protocol Configuration

1. EINLEITUNG

Die Gründe, dass aus einem überschaubaren Forschungsnetz in Nordamerika das weltumspannende "Netz der Netze", das Internet, wurde, sind vielfältig, doch ist es der TCP/IP-Protokollstack, dem mit seiner Omnipräsenz der spürbarste Anteil am Erfolg zugestanden werden muss: Als gemeinsame performante Basis verschiedenster Anwendungen auf unterschiedlichsten Systemen ermöglicht er es, die weltweite Heterogenität auf einen gemeinsamen Nenner zu bringen, großen vernetzten Systemen also überhaupt erst einen Sinn zu geben.

Doch trotz des ungeheuren Erfolges kommt man v.a. in letzter Zeit nicht umhin, die Schwächen dieser starren Architektur, ja die Schwächen von Architektur im Allgemeinen zu erkennen: Neue Anforderungen, wie sie z.B. durch Streamingservices wie VoIP und VoD, aber auch durch die wachsende Nutzung des Internets mit Mobilgeräten entstehen, machen Anpassungen an bestehenden Kernkomponenten einzelner Netze unabdingbar. Diese sind jedoch viel zu komplex und starr ausgelegt, als dass die Adaption neuer Innovationen ohne großen Aufwand vonstatten gehen könnte. Das beste Beispiel hierfür ist IPv6, denn obwohl der Abschluss der Spezifikation mittlerweile Jahre zurückliegt, ist die Akzeptanz und Verbreitung im Vergleich mit dem Voränger IPv4 immer noch sehr gering.

Was kann man also tun, um neuen Gegebenheiten in Gegenwart eines festen (und sei es auch nur eines "moralischen") Rahmens gerecht zu werden? Die Antwort pragmatisch denkender Entwickler liegt nahe [1]: Man sprengt den Rahmen. Statt mit IPv6 begegnen wir der schwindenden Zahl freier IP-Adressen mit Network Address Translation (NAT), ganz egal, ob dadurch das *End-to-End*-Prinzip des Internets verloren geht oder nicht; dem Risiko, das durch die Möglichkeit von *Denial of Service (DoS)* Attacken entsteht, schieben wir einen Riegel vor, indem wir (ebenfalls End-to-End-unfreundliche) Firewalls verwenden; und neue Funktionalität lässt sich nicht auf Netzwerk- (IP) oder Transport-Ebene (TCP) einführen - zumindest nicht, wenn man noch irgendetwas mit dem Internet zu tun haben will - sondern nur auf höheren Schichten. Dies produziert möglicherweise unnötigen Overhead, doch v.a. wird es fast zwingend erforderlich, den einzelnen Schichten die Kommunikation untereinander zu erlauben, da nur so ein an die Art der Daten, die den unteren Ebenen ja prinzipiell nicht bekannt ist, angepasster, optimierter Fluss möglich wird. Und das wiederum ist in einer Schichtenarchitektur wie dem OSI-Modell schlichtweg nicht erlaubt.

Das Internet ist nicht das Netz, das es selbst sein wollte. Es besteht aus einem immer noch verlässlichen Rumpf, der jedoch von vielen, vielen Workarounds und "Hacks" immer mehr zersetzt wird [2]. Die Komplexität steigt und mit ihr die Fehleranfälligkeit; die Auflösung der strikten Trennung zwischen den einzelnen Schichten (so z.B. zu sehen bei HTTP-Firewalls oder der Verwendung von Portnummern zu einer Vielzahl an Zwecken [4]) führt zu einer starken Kopplung verschiedener Netzwerkebenen, was darin resultiert, dass Anpassungen erschwert, ja fast schon gefährlich und unvorhersehbar werden.

Es gibt zwei Möglichkeiten, mit dieser Situation umzugehen: Man gesteht sich entweder ein, dass das Internet in seiner aktuellen Form zwar Schwächen, jedoch auch weiterhin das Recht hat, zu existieren, man also eine Alternative *neben* dem bestehenden Netz aufbauen sollte. Oder man will sich nicht auf das Risiko einlassen, das durch etwaige Altlasten entsteht, und favorisiert einen kompletten Neuanfang.

doi: 10.2313/NET-2011-05-2_03

In dieser Arbeit wird letztere Sicht vertreten: Zunächst sollen in Kapitel 2 die Anforderungen, die an ein zukünftiges World Wide Web zu stellen sind (und warum die aktuelle Architektur nicht beibehalten werden kann), sowie deren Verbindung mit Service Oriented Architectures (Kapitel 3.1) und Mikroprotokollen (Kapitel 3.2) beschrieben werden. Anschließend, in Kapitel 4, wird ein Mechanismus dargestellt, der die automatische Verwendung eines *optimierten* Stacks selbstbeschreibender Mikroprotokolle ermöglicht, bevor letztlich auf die offenen Probleme, die damit einhergehen, Bezug genommen wird.

2. ANFORDERUNGEN AN FUTURE NETWORKS

Im Folgenden soll aufgezeigt werden, welche Eigenschaften ein zukünftiges Internet haben und welchen Anforderungen es genügen sollte.

2.1 Evolvierbarkeit

Das Internet der Zukunft muss sich weiterentwickeln können. Es muss möglich sein, Protokolle ohne großen Aufwand hinzuzufügen, zu entfernen oder anzupassen, d.h. die Abhängigkeiten zwischen Anwendungen und den zugrundeliegenden Protokollen müssen auf ein Minimum reduziert werden.

Aktuell besteht das World Wide Web aus Schichten, von denen jede eine große Anzahl an Aufgaben übernimmt. Die einzige Möglichkeit, auch nur winzigste Änderungen vorzunehmen, ist der Austausch der (als Black-Box gehaltenen) kompletten Schichten [4]. Da die Schnittstellen zu den angrenzenden Ebenen hier beibehalten werden müssen, lassen sich Modifikationen transparent verwenden. Dennoch können wir nicht einfach einen frischen, dem bisherigen Modell treu bleibenden Protokollstack entwickeln und einführen, da wir schlicht und einfach nicht wissen, welche Herausforderungen uns in Zukunft bevorstehen! [2]

Ein zukünftiges Netz sollte die Sache deshalb feiner angehen und ermöglichen, klein gehaltene Funktionalitätseinheiten (wie z.B. die in Kapitel 3.2 beschriebenen Mikroprotokolle) auf dynamische Art und Weise einzubinden und zu verwenden. Eine Schlüsseleigenschaft ist die lose Kopplung zwischen den einzelnen Einheiten selbst, wobei sich hier als erfolgsversprechender Ansatz *Service Oriented Architectures (SOAs)* in den Vordergrund drängen. Diese werden im nächsten Kapitel genau beschrieben.

2.2 Anwendungsorientierung

Nicht alle Anwendungen haben die gleichen Bedürfnisse: ein Instant Messenger hat andere Sicherheitsansprüche als ein Online-Banking-Client, ein Videostream legt nicht so sehr wert auf eine verlässliche Datenübertragung wie ein User, der eine Datei von einem Server bezieht. Und während diese Bedürfnisse früher einen eher diskreten Charakter hatten (z.B. “Verlässliche Verbindung oder nicht?”), muss heute in Abstufungen gedacht werden (z.B. “Wie effektiv soll die Fehlerkorrektur sein?”). [5]

Ein zukünftiges Internet sollte die Fähigkeit haben, alle oder zumindest einen Großteil der Ebenen und Teilaufgaben (z.B. Forwarding) an die Bedürfnisse der Anwendungen anzupassen, die über das Netz kommunizieren. Natürlich ist ein anwendungsorientierter Datenaustausch auch mit den bereits bestehenden Technologien möglich, allerdings hauptsächlich in höheren Schichten, was einen möglichen Daten-Overhead begünstigt und zu den bereits in der Einleitung angesprochenen Problemen bezüglich der effizienten, datenorientierten Kommunikation führt.

Als Ansatz, ein System möglichst anwendungsorientiert zu gestalten, lassen sich Mikroprotokolle nennen: Komplexe Aufgaben werden in ihre kleinsten Teile zerlegt, sodass aus einer geringen Menge an “Bauteilen” eine große Zahl komplexer Kompositionen mit unterschiedlichsten Eigenschaften erzeugt werden kann. (Kapitel 3.2)

2.3 Effizienz

Ein letzter Blick sei hier auf die Effizienz gelegt, denn was nützt das schönste Konzept, wenn am Ende das Netzwerk von einer Datenflut heimgesucht wird oder sich einzelne Knoten zu Tode rechnen? Vor allem ein minimaler Protokoll-Overhead ist nötig, um den Datenfluss in den Netzen der Zukunft zu optimieren.

Betrachtet man z.B. den IPv4-Header, sieht man Felder, die die Fragmentierung steuern und selbst dann präsent sind, wenn durch andere Maßnahmen wie PMTU (oder noch einfacher: dem DF-Flag im Header selbst) bereits sichergestellt ist, dass das Paket nicht zerlegt werden muss. Oder aber das *Header Checksum*-Feld, das von vielen Routern aus Performancegründen gar nicht erst ausgewertet wird.

Wir werden in Kapitel 3.2 sehen, dass Mikroprotokolle durch ihren modularen Aufbau und ihre namensgebende Kompaktheit, sowie durch die im vorhergehenden Abschnitt besprochene Anwendungsorientierung sehr gute Performance und geringen Overhead erzielen können.

3. GRUNDLAGEN

In diesem Kapitel geht es um die Bauteile, die, wenn zusammen verwendet, den Anforderungen in Kapitel 2 gerecht werden und damit die Basis für ein Future Internet legen können.

3.1 Service Oriented Architectures (SOAs)

Service Oriented Architectures ermöglichen es, Funktionalität dynamisch zu verwalten und lose gekoppelt zur Verfügung zu stellen. Die grundlegende Einheit sind hierbei sog. *Services* - Dienste, die sich v.a. durch die folgenden Eigenschaften auszeichnen [4] [6]:

Eigenständigkeit Ein Service ist in sich abgeschlossen, d.h. er stellt seine Funktionalität in einer Art und Weise zur Verfügung, die es ermöglicht, ihn unabhängig von anderen Diensten zu nutzen. Das bedeutet aber nicht, dass Services nicht miteinander kommunizieren dürfen, wie wir weiter unten sehen werden.

Wohldefinierte Schnittstelle Die Verwendung eines Dienstes muss nur durch Kenntnis seiner Schnittstelle, also ohne Wissen über interne Implementierungsdetails, möglich sein. So könnte ein Service, der die Integrität von Daten mithilfe einer Prüfsumme sicherstellen soll, auf verschiedene Arten realisiert werden, z.B. mithilfe von CRC oder einer Hashfunktion wie SHA-1.

doi: 10.2313/NET-2011-05-2_03

"Blindheit" Um eine möglichst lose Kopplung der einzelnen Services zu gewährleisten, müssen Annahmen über die internen Gegebenheiten anderer Dienste außen vor bleiben. So sollte ein Service A, der den Wert eines Prüfsummenservices B vorgesetzt bekommt, nicht davon ausgehen, dass die Berechnung dieser Summe z.B. mit CRC geschehen ist. Das ermöglicht den einfachen Austausch von B durch einen anderen Service.

Wiederverwendbarkeit Eine sich aus den vorhergehenden Punkten ergebende Eigenschaft von Services ist die Wiederverwendbarkeit der Funktionalität in unterschiedlichsten Umgebungen. Noch wichtiger: Low-Level Services lassen sich zu neuen, komplexeren Diensten zusammensetzen, was z.B. in Service Oriented Communication Systems (SOCS, siehe Kapitel 3.3) genutzt wird.

Services sind also modulare Einheiten, die *on demand* instanziiert werden können.

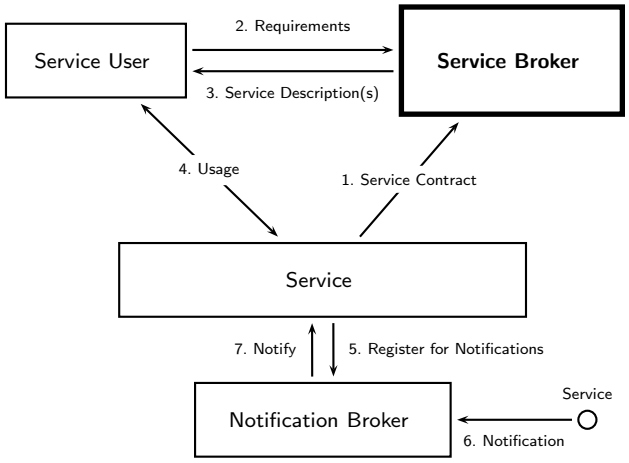


Abb. 1: Datenfluss in SOAs (nach [3], [4])

Der beste Service ist jedoch nutzlos, wenn er mangels Bekanntheit von niemandem in Anspruch genommen werden kann. An dieser Stelle kommt der *Service Broker* ins Spiel, der - wie ein Telefonbuch - eine Liste von ansprechbaren Diensten verwaltet, in die sich jeder Service erst einmal eintragen muss. Der dabei übertragene *Service Contract* beinhaltet u.a. die Spezifikation der Schnittstelle, sowie die Eigenschaften eines Dienstes, anhand derer der Broker in der Lage ist, eine passende Liste von Services zu einer gegebenen Menge an Anforderungen zu ermitteln. (siehe Abbildung 1, Schritte 1-3)

Als Beispiel seien Services gegeben, die Bücher in einer Bibliothek für eine bestimmte Zeit reservieren und dafür eine Bearbeitungsgebühr verlangen: Während manche User einen möglichst langen Reservierungszeitraum bevorzugen, ist für andere vielleicht wichtiger, dass die Gebühren minimal sind. Dies kann in den Anforderungen festgelegt werden, der Broker trifft dann die jeweils beste Entscheidung.

Es ist offensichtlich, dass für die Kommunikation zwischen Service und Broker bzw. zwischen Broker und User ein einheitliches Format verwendet werden muss, das von beiden Seiten verstanden wird und die notwendigen Informationen übermitteln kann. Im Bereich der WebServices (die einige SOA-Prinzipien widerspiegeln [6]) wird z.B. oftmals WSDL [8] für Schnittstellenspezifikationen genutzt; auch UDDI-Verzeichnisse [7] arbeiten hauptsächlich mit XML-Daten (SOAP-Schnittstelle). Auf die Frage, inwieweit eine formale Spezifikation eines Services überhaupt möglich ist bzw. an welchen Stellen es Probleme geben kann, wird aber im letzten Teil dieser Arbeit nochmals eingegangen.

Die Kommunikation zwischen Services geschieht (der losen Kopplung halber) indirekt, d.h. über eine Vermittlungsstation: Hat ein Service Daten, die er veröffentlichen/anderen zur Verfügung stellen will, so speichert er diese an einem von allen berechtigten Services zugänglichen Ort und benachrichtigt den sog. *Notification Broker*. Dieser informiert wiederum alle weiteren Services, die sich für die entsprechende Art von Benachrichtigungen (*Notifications*) registriert haben [4]. (siehe Abbildung 1, Schritte 5-7) So kann es z.B. einen Monitoring-Service geben, der alle Fehler-Notifications abfängt und in Log-Dateien niederschreibt.

SOAs ermöglichen es also, Funktionalität flexibel zur Verfügung zu stellen - eine Eigenschaft, die auch in den Netzen der Zukunft vorhanden sein soll.

3.2 Mikroprotokolle

Will man ein Haus bauen, so muss man sich über eine Reihe von Dingen Gedanken machen: Wo positioniert man beispielsweise die Fenster? Oder welche Art von Dach passt am besten? Natürlich kann man das alles umgehen und auf bewährte, schnell verfügbare Lösungen, ja vielleicht gleich auf ein ganzes Fertighaus zurückgreifen - doch selbst wenn bisher 80% der Nutzer zufrieden mit der Isolierung waren, heißt das nicht, dass sie auch für ein Domizil in Nordsibirien geeignet ist. Und ganz ehrlich: wenn alle Häuser gleich aussehen, sieht doch keines mehr gut aus.

Auf den heutigen Systemen ist TCP/IP so ein (wenn auch gut funktionierendes) Fertighaus - ja sogar eines, um das man nicht umhin kommt. Will man mit dem Rest der Welt (sprich: Anwendungen auf anderen Rechnern) in Kontakt bleiben, so *muss* man IP und evtl. TCP bzw. UDP "sprechen", egal ob man alle deren Features überhaupt braucht oder nicht.

Mikroprotokolle sind demgegenüber mehr wie die fertigen Fenster, Dächer und Türen, die man sich aussuchen kann, um dem eigenen Haus eine gewisse Individualität zu verpassen. Jedes Mikroprotokoll übernimmt eine kompakte, klar definierte Aufgabe [2], was es ermöglicht, sie zu neuen, größeren Protokollen zusammenzusetzen. So könnte es z.B. ein Mikroprotokoll zur Fehlererkennung (beispielsweise über eine Prüfsumme) geben, das folgendermaßen aussieht:

Daten	Prüfsumme

Dazu noch eines, das mithilfe von Sequenznummern und Acknowledgements die richtige Reihenfolge und die verlässliche Ankunft der Daten sicherstellt:

Sequenznummer	Acknowledgement
Daten	

doi: 10.2313/NET-2011-05-2_03

Und zuletzt noch ein simpler Forwardingmechanismus anhand von Quell- und Zieladressen (nicht notwendigerweise IP-Format!):

Quelle	Ziel
Daten	

Kombiniert man diese Mikroprotokolle nun, erhält man ein neues, komplexeres Protokoll, das eine gewisse Ähnlichkeit zum bestehenden TCP/IP hat, allerdings z.B. auf Fluss- und Staukontrolle verzichtet und dementsprechend in Netzen, die diese Mechanismen nicht benötigen, den Overhead reduziert:

Quelle	Ziel
Sequenznummer	Acknowledgement
Daten	
Prüfsumme	

Mikroprotokolle ermöglichen es also, an die Bedürfnisse von Anwendungen und Netzen angepasste Protokolle durch Komposition kleinerer Teile zu erstellen. Wichtig ist jedoch, dass sich sowohl Sender als auch Empfänger darüber einig sind, in welcher Reihenfolge die einzelnen Teile zu bearbeiten sind. Am Beispiel unseres "Makroprotokolls" ist z.B. nicht direkt ersichtlich, ob sich die Prüfsumme auf das gesamte Paket oder möglicherweise nur die Daten bezieht.

An dieser Stelle liegt die Entscheidung darüber, welche "Bausteine" zu verwenden, aber immer noch bei der Anwendung selbst, d.h. die Akzeptanz neuer Protokolle ist nach wie vor gering, da ihre Verwendung nicht automatisch geschieht. Sehen wir Mikroprotokolle aber als Services im Sinne einer SOA, so können wir die letztliche Auswahl der einzelnen Funktionen in eine externe Einheit, nämlich den Broker, verlagern, und stehen mit einem Mal einer Fusion aus Flexibilität und Anwendungsorientierung gegenüber: *SOCS*.

3.3 Service Oriented Communication Systems (SOCS)

Die Funktionsweise von *Service Oriented Communication Systems* - also Systemen, die die Kommunikation zwischen zwei Endpunkten mithilfe von Services praktizieren - dürfte an dieser Stelle bereits erahnt werden: Eine Anwendung, die eine Verbindung aufbauen will, schickt eine Liste mit Anforderungen an den (lokalen) Service Broker, der sie mit den Servicespezifikationen in seiner Datenbank vergleicht. Anschließend ermittelt er den besten Treffer und stellt ihn der Anwendung zur Verfügung.

Hierbei gibt es zwei Möglichkeiten: entweder der Broker verwaltet eine Liste fertig verwendbarer Protokollstacks (*Communication Services*), oder er erstellt aus kleineren Services (z.B. Mikroprotokollen) einen neuen, individuell angepassten Communication Service. Die erste Möglichkeit ist mit weniger Rechenaufwand (dafür mehr Konfiguration) verbunden, schränkt aber auch die Anpassbarkeit der Kommunikation ein, während die zweite Möglichkeit einen effizienten Auswahlalgorithmus erfordert, demgegenüber aber sehr nahe an die Anforderungen herankommen kann. Natürlich lassen sich einmal erstellte Kombinationen auch zwischenspeichern und bei Bedarf anpassen, was einer Hybridlösung der beiden Ansätze entspricht.

Wichtig ist: Ein Service Broker ist bei seiner Auswahl nicht nur auf die formalen Spezifikationen beschränkt, die ihm übergeben werden! Er kann - und dies ist v.a. im Netzwerkumfeld entscheidend - auch auf externe Daten zurückgreifen [3], z.B. Rechnerauslastung, Firewallkonfigurationen, Netzwerkcharakteristika, etc... Am Ende steht dann ein optimierter Protokollstack, den die Anwendung über eine einheitliche Schnittstelle verwenden kann. (siehe Fig. 2)

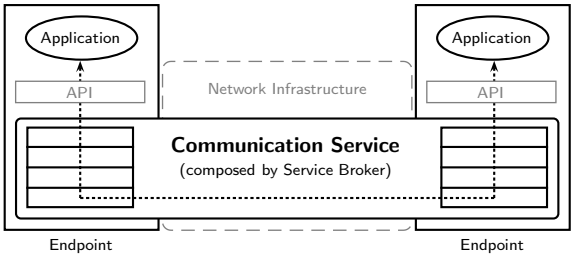


Abb. 2: Aufbau von SOCS (nach [3])

Wir betrachten in dieser Arbeit den Ansatz, Mikroprotokolle als Basis zur Komposition von SOCS zu verwenden. Die letzte offene Frage diesbezüglich ist nun: *Wie* kann ein Service Broker die beste Kombination, also den passendsten Protokollstack, bestimmen?

4. SELBSTBESCHREIBENDE PROTOKOLLE

Bisher haben wir uns mit der allgemeinen Funktionsweise einzelner Teile auseinandergesetzt: SOAs zur losen Kopplung, Mikroprotokolle zur Anwendungsorientierung und Effizienz, sowie SOCS als Fusion der beiden Ansätze. In diesem Kapitel werden wir uns nun jedoch detailliert mit einem Auswahlverfahren zur Komposition von Communication Services befassen, das anhand von Mechanismen und Effekten (siehe nächster Abschnitt) ein ideales Ergebnis erzielen soll.

4.1 Implementierung, Mechanismus & Effekt

Betrachten wir an dieser Stelle einmal das Standardwerk eines Programmieranfängers: Hello World. Es gibt viele verschiedene Arten, die bekannte Ausgabe zu erzeugen, z.B. in Java:

```
System.out.println("Hello World");
```

Genausogut wäre es möglich, den String zuerst in einer Variable zu speichern, bevor er ausgegeben wird:

```
String hello = "Hello World";
System.out.println(hello);
```

Beide Programme sind konkrete *Implementierungen* eines abstrakten *Mechanismus*, d.h. einer implementierungsunabhängigen Beschreibung von Funktionalität [2]. In unserem Fall lautet der Mechanismus: "Schreibe 'Hello World' auf die Konsole!".

Verschiedene Implementierungen können demnach derselben Spezifikation genügen. Dies gilt für einfache "Hello-World"-Programme genauso wie für Mikroprotokolle oder gar TCP: Es gibt viele Möglichkeiten, z.B. die Staukontrolle (einen

doi: 10.2313/NET-2011-05-2_03

Mechanismus!) zu implementieren, und alle sind zulässig, solange sie nur der formalen, schriftlich festgehaltenen Spezifikation von TCP entsprechen.

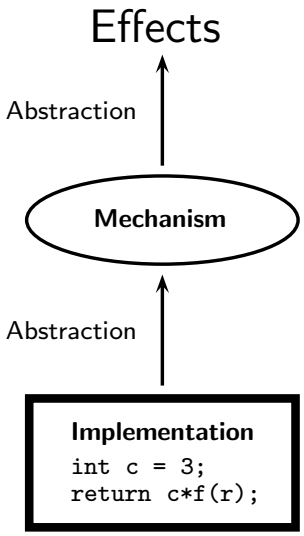


Abb. 3: Von der Implementierung zum Effekt [2]

Mechanismen lassen sich (siehe Abb, 3) anhand ihrer Ergebnisse beschreiben, den sog. *Effekten* [2]: So bewirkt z.B. die Verwendung des Mechanismus "CRC-Prüfsumme" den Effekt "Korrektheit der Daten", während etwa "Sequenznummern" die "Richtige Reihenfolge" begünstigen.

Neben den normalen Effekten existieren noch sog. *Meta-Effekte*, also solche, die sich durch Komposition mehrerer anderer ergeben. "Verlässlichkeit" setzt sich beispielsweise aus "Korrektheit der Daten", "Richtige Reihenfolge" und "Vollständigkeit" zusammen. (siehe Abb. 4). Natürlich kann ein Mechanismus auch mehrere Effekte mit sich bringen, genauso wie es möglich ist, dass ein Effekt erst durch Verwendung mehrerer Mechanismen entsteht. Doch dies ist in [2] detailliert beschrieben, weshalb hier nicht genauer darauf eingegangen werden soll.

4.2 Mechanismen, Services & Mikroprotokolle

Da Effekte eine solide Grundlage zur Beschreibung von Mechanismen darstellen, eignen sie sich sehr gut, um etwa in einem SOCS Service Broker zur Auswahl eines passenden Protokollstacks herangezogen zu werden. Hierbei werden Mechanismen (in unserem Fall also Mikroprotokolle) in Services gekapselt, die sich beim Broker unter Angabe mindestens der folgenden Daten registrieren müssen:

Liste von Effekten Jedes Mikroprotokoll hat gewisse Effekte - und der Broker muss wissen, um welche es sich dabei handelt, da ansonsten eine Auswahl passender "Bausteine" schwer wird. Hierbei sollte man unterscheiden zwischen *qualitativen* Effekten, die eine Bewertung des Protokolls zulassen [3] (z.B. im Bereich des *Quality of Service*), oder aber *inhärenten* Effekten, die einen absoluten Charakter (vorhanden oder nicht vorhanden) haben. In [2] werden nur letztere dargestellt, doch in meinen Augen darf man, v.a. in Anbetracht der Anforderungen, die durch VoIP et al erwachsen, nicht auf qualitative Betrachtungen verzichten.

Anforderungen Manche Mikroprotokolle funktionieren nur in Verbindung mit anderen richtig, z.B. macht es keinen wirklichen Sinn, Sequenznummern zur Sicherstellung der Reihenfolge zu verwenden, wenn nicht auch die Prüfsumme des Protokollheaders vorhanden ist, um zufälligen Änderungen in der Nummerierung von Paketen vorzubeugen. Dies muss der Service Broker wissen, um einen konsistenten Protokollstack erstellen zu können.

Kosten Jedes Mikroprotokoll ist mit einer gewissen Menge an Rechenaufwand verbunden, welcher natürlich (neben den o.g. qualitativen Aspekten) ebenfalls in die Berechnungen des Brokers einfließen sollte.

Hat ein Service Broker all diese Daten, kann er loslegen.

4.3 Aufbau

Die Komposition eines passenden Mikroprotokollstacks geschieht auf Basis einer bestehenden Netzwerkarchitektur, die sich aus den vorhandenen Mechanismen $\mathbb{M}$ (z.B. Prüfsumme, Sequenznummern, Acknowledgements, Staukontrollfenster, ...), den existierenden Effekten und Meta-Effekten $\mathbb{E}$ (z.B. Vertraulichkeit, Forwarding, Vollständigkeit, ...), sowie denjenigen Mechanismen $\mathbb{M}_f$ zusammensetzt, die in jedem Fall verwendet werden, da sie auf niedrigeren Netzwerkebenen implementiert sind (z.B. die Frame Check Sequence in Ethernet-Paketen). [2]

Der Service Broker hat, wie oben festgestellt, zu jedem Mechanismus eine Liste von Effekten gespeichert, was wir hier als Funktion $P : \mathbb{M} \rightarrow 2^{\mathbb{E}}$ darstellen wollen. Gleichzeitig existiert eine Liste von benötigten Mechanismen, die mithilfe von $R : \mathbb{M} \rightarrow 2^{\mathbb{M}}$ realisiert wird.

4.4 Konsistenz

Eine Auswahl von Mechanismen sei genau dann *konsistent*, wenn sie alle Mechanismen enthält, die benötigt werden:

$$consistent(M \subseteq \mathbb{M}) = \left[\bigcup_{m \in M} R(m) \right] \subseteq M$$

Dies stellt sicher, dass keine unvollständigen Protokollstacks als Communication Services verwendet werden.

4.5 Auswahl von Mechanismen

Erhält der Service Broker eine Reihe von erwünschten Effekten $\mathbb{D}$, so berechnet er eine passende Menge von Mechanismen, also einen passenden Protokollstack, als konsistente Mechanismusliste mit minimalen Kosten, d.h. [2]:

$$S(\mathbb{D}) = \arg\min_{M \subseteq \mathbb{A}} C(M)$$

wobei C die Gesamtkosten einer Liste von Mechanismen berechnet und $\mathbb{A}$ alle konsistenten Mengen passender Mechanismen darstellt:

$$\mathbb{A} = \{M \subseteq \mathbb{M} | \mathbb{M}_f \subseteq M \wedge consistent(M) \wedge D \subseteq P^*(M)\}$$

(P^* ist die Menge aller Effekte, die durch eine Liste von Mechanismen erzielt werden.)

doi: 10.2313/NET-2011-05-2_03

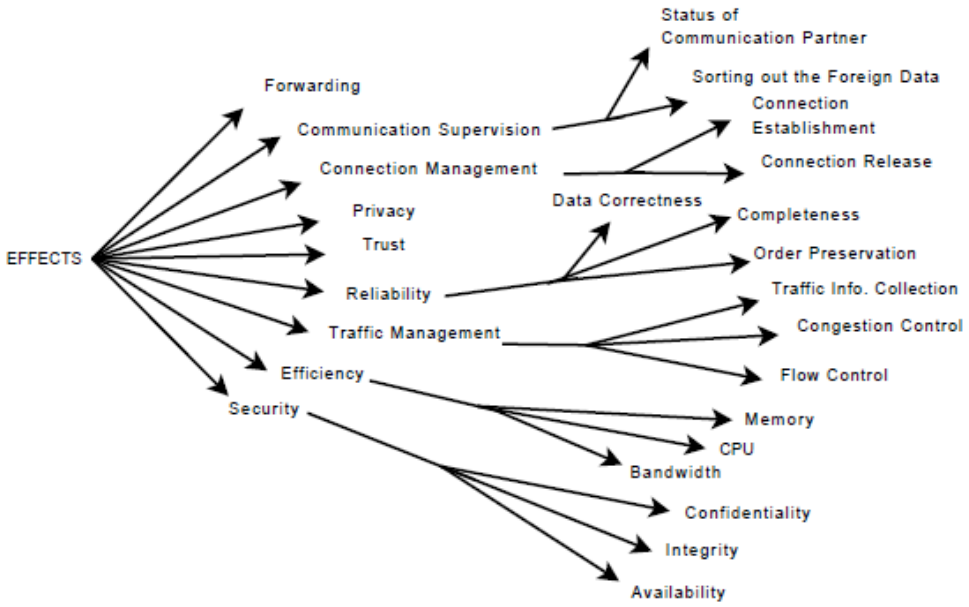


Abb. 4: Beispielhafte Effekthierarchie [2]

Ein sehr komplexer Punkt ist hierbei die Kostenfunktion C, da in verschiedenen Verbindungen von Mechanismen einzelne Gewichtungen ebenfalls unterschiedlich sein können (z.B. wenn Werte im Protokollheader als Zwischenergebnisse wiederverwendet werden). Zusätzlich sollte die Qualität von Mechanismen - relativ zu den entsprechenden Anforderungen - mit in die Berechnung einfließen, was keine simple Aufgabe ist.

Alles in allem ist der Algorithmus, der $S(\mathbb{D})$ berechnet, die wohl kritischste Komponente des hier beschriebenen Ansatzes, denn arbeitet dieser nicht effizient, kommt es zu Verzögerungen beim Verbindungsaufbau, die nicht akzeptabel sind. In [2] werden mehrere Implementierungsmöglichkeiten angeschnitten, darunter eine naive Herangehensweise (alle konsistenten Lösungen - und das sind sehr viele - finden und bewerten); eine, die mit zuvor schon berechneten Elementen arbeitet und sie nur noch an die Gegebenheiten anpasst (wie so eine Anpassung aussehen kann, ist in [3] beschrieben); und letztlich eine, die auf parallel ausgeführte Heuristiken baut.

Wie die Auswahl am Ende geschieht, ist offen und kann von Netz zu Netz unterschiedlich sein. Wir sehen aber, dass die Beschreibung von Mikroprotokollen anhand ihrer Effekte ein vernünftiges Mittel zur Optimierung von SOCS darstellt.

4.6 Ein Beispiel

Das war nun alles sehr theoretisch und formal, weshalb der Sachverhalt an einem Beispiel verdeutlicht werden soll. Gehen wir davon aus, dass der Service Broker die folgenden, beispielhaft definierten Mechanismen in seiner Datenbank gespeichert hat (wobei die Funktionalität niedrigerer Schichten an dieser Stelle ausnahmsweise ignoriert wird):

Header-Prüfsumme	
benötigt	-
erzielt	Integrität
kostet	2

Sequenznummern	
benötigt	Header-Prüfsumme
erzielt	Reihenfolge
kostet	1

Acknowledgements	
benötigt	Header-Prüfsumme
erzielt	Vollständigkeit
kostet	1

Prüfsumme	
benötigt	-
erzielt	Korrektheit
kostet	2

Sliding Window	
benötigt	Sequenznummern, Acknowledgements
erzielt	Flusskontrolle, Effizienz (Bandbreite)
kostet	3

Stop-and-Wait	
benötigt	Acknowledgements
erzielt	Flusskontrolle
kostet	1

doi: 10.2313/NET-2011-05-2_03

(Hier haben wir übrigens ein Beispiel, in dem Services miteinander kommunizieren müssen, um eine vernünftige Funktionalität herzustellen, denn wie können die Flusskontrollmechanismen arbeiten, wenn sie keine Daten vom Acknowledgement-Service bekommen?)

Wenn der Service Broker einen Communication Service zusammenstellen soll, der Flusskontrolle und Korrektheit garantiert, wird er (bei naiver Herangehensweise) zuersteinmal die folgenden Kombinationen ermitteln:

- {Sliding Window, Prüfsumme}
- {Stop-and-Wait, Prüfsumme}

Anschließend werden die Mengen so lange mit benötigten Mechanismen angereichert, bis sie konsistent sind:

- {Sliding Window, Prüfsumme, Sequenznummern, Acknowledgements, Header-Prüfsumme}
- {Stop-and-Wait, Prüfsumme, Acknowledgements, Header-Prüfsumme}

Die Kostenfunktion wird (ohne Rücksicht auf irgendwelche Abhängigkeiten) die beiden Werte 9 und 6 berechnen, weshalb nur die Stop-and-Wait-Variante in den Protokollstack einfließt, den der Nutzer verwenden soll. Wäre in den Anforderungen noch die effiziente Ausnutzung der Bandbreite aufgetaucht, hätte das Sliding Window den Vorzug erhalten.

Hier sehen wir auch einen Grund dafür, qualitative Effekte zuzulassen, v.a. im Bereich der Effizienz und Übertragungsqualität: Was passiert etwa, wenn ein Nutzer angeben will, dass die Effizienz der Bandbreite nicht zu 100%, sondern nur zu 40% in die Entscheidungen des Brokers einfließen sollen (weil z.B. in Firmennetzwerken ab einer gewissen Uhrzeit sowieso weniger los ist)? Wenn dann nicht bekannt ist, *wie* effizient ein gegebener Mechanismus ist - und wie sich das auf die Kosten auswirkt - hat man ein Problem. (Es sei hier auf [3] verwiesen, wo sich ein Beispielverfahren zur Auswahl anhand qualitativer Eigenschaften findet.)

5. BEWERTUNG UND OFFENE PROBLEME

5.1 Heterogenität

Durch den ständigen Wandel der Funktionalität einzelner Knoten, wie sie durch SOAs ermöglicht wird, entsteht zwangsläufig Heterogenität, der entsprechend begegnet werden muss. Nicht alle Netzelemente können mit allen Mechanismen umgehen [4] und das simple Aushandeln der verwendeten Protokolle durch die Endpunkte garantiert nicht, dass ein Paket auch wirklich den gesamten Weg übersteht.

Es gibt verschiedene Ansätze, dieses Problem zur Laufzeit zu lösen [4]:

Funktionalität ignorieren So ist z.B. Flusskontrolle wünschenswert, aber nicht notwendig, um Daten zu übertragen, und kann dementsprechend von einzelnen Knoten ignoriert werden.

Funktionalität auflisten Existieren nur wenige Mechanismen, die betrachtet werden sollen, so können diese zwischen den Knoten ausgetauscht werden, um entsprechend interoperable Lösungen zu wählen. Bei einer größeren Menge an Protokollen wird das aber nicht mehr effizient möglich sein.

Funktionalität delegieren Ist ein Knoten bekannt, der mit einem Mechanismus umgehen kann, so kann man die Daten an diesen zur Verarbeitung weiterleiten. Dies ist jedoch nicht immer praktikabel und erhöht außerdem die Netzauslastung.

Generell sind Lösungen, die zur Laufzeit angewandt werden, nicht unbedingt ideal, da meist zusätzlicher Traffic erzeugt wird, der den Vorteil des geringen Overheads von Mikroprotokollstacks zunichte macht. Machbarer und auch logisch sinnvoller ist die Klärung der Kompetenzen einzelner Knoten bereits im Vorfeld, wobei zwei Prinzipien zu nennen sind:

Capability Domains Ein Knoten kann sich in einer Capability Domain registrieren, wenn er eine bestimmte Funktionalität (bzw. einige wenige Funktionalitäten) besitzt. So wären mögliche Domains z.B. definiert durch die Fähigkeit, Stau- oder Flusskontrolle zu verwenden, IP-Adressen zu verstehen, etc... Die entsprechenden Informationen könnten ähnlich zu Routing-Protokollen verteilt werden. [4]

Overlays Ein anderer Ansatz ist es, das Gesamtnetz in kleinere (virtuelle) Teile, sog. Overlays (vgl. SpovNet [9]), zu zerlegen, in denen alle enthaltenen Knoten eine bestimmte Menge an Funktionalität besitzen. Dementsprechend darf sich auch niemand in einem Overlay registrieren, der nicht alle Vorraussetzungen bzgl. der Funktionalität erfüllt. [4]

Der Unterschied ist subtil: Während bei Capability Domains alle Rechner im selben Netz liegen und somit beim Aufbau der Kommunikation zwischen zwei *bestimmten* Knoten immer noch geklärt werden muss, welche Funktionen - sprich: Services - auf beiden Seiten vorhanden sind (anhand des Vergleichs der Capability Domains), stellen Overlays Netze dar, in denen *alle* Knoten eine bestimmte Reihe von Fähigkeiten besitzen, aus denen ohne weitere Abstimmung frei gewählt werden kann. Demgegenüber erschwert das abgrenzende Prinzip der Overlays aber die Kommunikation im Gesamtnetz, während Capability Domains eine Anpassung an unterschiedliche Gegebenheiten ermöglichen. Welche der beiden Varianten die bessere ist, lässt sich zu diesem Zeitpunkt allerdings noch nicht sagen.

Eine letzte Möglichkeit wäre die Festlegung eines Standard-(Transport-)Protokollstacks, der als *Fallback*-Mechanismus agiert, wenn keine vernünftige Lösung gefunden wird. Dies birgt aber die Gefahr, dass der Einfachheit halber die Kommunikation zwischen zwei Endpunkten über Gateways geleitet wird, die den Fallback-Mechanismus verwenden. Protokollspezifische Eigenschaften hätten dann nämlich keinen Wert mehr.

5.2 Spezifikation von Services & Optimierung

Es ist schwierig, ein einheitliches Format festzulegen, das von Services verwendet wird, um ihre Beschreibungen an den Service Broker zu übermitteln: Wir können nicht wissen, welche Anforderungen zukünftige Dienste an dieses Format haben, genausowenig wie wir sicher sein können, welche Da-

doi: 10.2313/NET-2011-05-2_03

ten ein “neuartiger” Broker bräuchte, um ideale Ergebnisse zu liefern.

In [3] wird deshalb vorgeschlagen, eine Liste von *Properties*, also Name-Wert-Paaren zu übermitteln, die beliebig erweiterbar ist. Dies erhöht die Flexibilität bei der Definition von Services, doch löst noch lange nicht alle Probleme, denen wir hier begegnen.

Zwischen Services existieren unterschiedlich geartete Abhängigkeiten: Die *Header-Prüfsumme* macht z.B. keinen Sinn, wenn sie sich nicht auf den gesamten Header bezieht; und ein Acknowledgements-Mechanismus erzielt alleine vllt. die Effekte “Vollständigkeit” und “Richtige Reihenfolge” (wenn nämlich als Stop-and-Wait-Mechanismus implementiert), kann die Reihenfolge der Daten aber nicht mehr sicherstellen, wenn zusammen mit dem Sliding-Window-Ansatz verwendet. Auch die Präsentation von Rechenkosten einzelner Mechanismen ist sehr komplex (siehe Kostenfunktion in Kapitel 4.5), da manche Mikroprotokolle z.B. “billiger” werden, wenn sie auf Zwischenergebnisse anderer Services zurückgreifen können.

Die Beibehaltung der Flexibilität der Service-Spezifikation bei gleichzeitiger Darstellbarkeit komplexer Beziehungen ist somit ein Punkt, der kritisch für die Optimierung zukünftiger Netze ist.

5.3 Verlässlichkeit

Man kann in keiner Architektur garantieren, dass alle Dienste zu jedem Zeitpunkt verfügbar sind. Während dies allerdings bei “traditionellen” Systemen zum Totalausfall führen könnte, bietet die Dynamik, die mit SOAs einhergeht, ein sehr starkes Mittel zur Anpassung an solche Vorfälle. [6]

5.4 Performance

Die Performance des hier beschriebenen Ansatzes hängt stark davon ab, wie gut und schnell der Service Broker Entscheidungen trifft, sowie von den verfügbaren Mikroprotokollen. Dies sind jedoch Fragen geeigneter Optimierung und Auswahl, welche von Netz zu Netz unterschiedlich sein wird.

Es lässt sich hier jedoch getrost festhalten, dass sich nicht alle Anwendungsgebiete für den Einsatz von Mikroprotokollstacks eigenen: So scheinen z.B. im Bereich der Hochleistungsrechner stark spezialisierte Protokolle besser geeignet als “ungefähr optimale” Kompositionen.

5.5 Implementierungs- & Dokumentationsaufwand

Die Verwendung von SOAs erfordert v.a. eine Anpassung der Denkweise einzelner Programmierer, da die Beziehungen zwischen Services nicht mehr statisch, sondern extrem wandelbar sind. So wird es in erster Linie unbedingt notwendig, die Dokumentation solcher Beziehungen so ausführlich und detailliert wie möglich zu gestalten, um (z.B. in großen Firmen) nicht aus Angst vor Änderungen letztlich wieder an ein starres System gebunden zu sein.

Was einzelne Services angeht, wird sowohl Implementierung und Dokumentation einfacher: Jeder Dienst hat eine bestimmte Aufgabe, ist also nicht übermäßig komplex, und stellt durch seinen selbstbeschreibenden Charakter schon alle Informationen zu Schnittstelle & Co. zur Verfügung. (Dokumentation muss “sowieso” betrieben werden, kann also nicht mehr als lästige Zusatzaufgabe gesehen werden.)

Der Broker und der Auswahlalgorithmus für einzelne Services sind die wohl kritischsten Teile der Implementierung und erfordern sehr hohen Aufwand. Zukünftige Verbesserungen werden vermutlich v.a. hier ansetzen, was ebenfalls einen hohen Dokumentationsgrad mit sich bringt.

Alles in allem wird der Aufwand bei Implementierung und Dokumentation durch SOAs nicht geringer werden, doch die sich daraus ergebenden Vorteile, besonders im Bereich der Netzwerkorganisation, werden sich irgendwann bezahlt machen.

6. ZUSAMMENFASSUNG

Die Verwendung von Service Oriented Communication Systems auf Basis selbstbeschreibender Mikroprotokolle ermöglicht es, ein Netz aufzubauen, das für zukünftige Anforderungen bestens gewappnet ist. Dennoch existieren noch einige offene Fragen - allen voran das Problem der Heterogenität - deren Klärung entscheidend für den Erfolg und die Akzeptanz solcher Systeme ist. Der nächste logische Schritt ist deshalb der Aufbau einfacher Prototypen (so z.B. in [2] und [3] angekündigt), die dann die Praktikabilität und Konkurrenzfähigkeit dieses in meinen Augen erfolgsversprechenden Ansatzes untermauern.

7. LITERATUR

[1] T. Roscoe: *The End of Internet Architecture*, in *Proceedings of the fifth workshop on hot topics in networks (HotNets-V)*, S.55-60, Irvine, USA, November 2006

[2] D. Schwerdel, Z.Dimitrova, A. Siddiqui, B. Reuther, P. Müller: *Composition of Self Descriptive Protocols for Future Network Architectures*, Karlsruhe, Germany, August 2009

[3] B. Reuther, D. Henrici: *A model for service-oriented communication systems (Journal Version)*, in *Journal of Systems Architecture*, June 2008

[4] B. Reuther, J, Götze: *An Approach for an Evolvable Future Internet Architecture*, in *1st Workshop, New Trends in Service & Networking Architectures*, November 2008

[5] N. Van Wembeke, E. Expósito, M. Diaz: *Transport Layer QoS Protocols: The Micro-Protocol approach*, OpenNet Workshop 1, March 2007

[6] M.H. Valipour, B. Amirzafari, Kh. N. Maleki, M. Valadbeigi, N. Daneshpour: *Concepts of Service Orientation in Software Engineering: A Brief Survey*, in *MASAUM Journal of Reviews and Surveys, Vol. 1, No. 3*, S.244-250, November 2009

[7] Universal Description Discovery and Integration (UDDI), online: `http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm`

[8] WebService Description Language (WSDL), online: `http://www.w3.org/TR/wsdl20/`

[9] Spontaneous Virtual Networks, online: `http://www.spovnet.de`

doi: 10.2313/NET-2011-05-2_03

Overlay Convergence Architecture for Legacy Applications (OCALA)

Dieter Panin
Betreuer: Heiko Niedermayer
Seminar Future Internet WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: panind@in.tum.de

KURZFASSUNG

Die Umstellung von der IP-basierten Architektur des Internets auf die eines Overlaynetzwerks war für die User stets mit Einstellungen am Betriebssystem, sowie an den auszuführenden Anwendungen verbunden. Um aber dem Nutzer die Vorteile der Overlays näher zu bringen, muss es möglich sein, ein Overlay bequem und ohne nötige Voreinstellungen laufen zu lassen. Dieses Ziel setzten sich die Entwickler von OCALA[1][2], einer Netzarchitektur, die eine automatische Abstimmung zwischen dem Overlay und IP, sowie mehreren Overlays untereinander ermöglicht. OCALA fügt dazu eine neue Schicht in das ISO/OSI-Modell ein, die Overlaykonvergenzschicht OC. Diese befindet sich unter der Transportschicht und besteht aus zwei Subschichten, der overlayunabhängigen OC-I, die eine Schnittstelle nach oben für die Anwendung bereitstellt, und der overlayabhängigen OC-D, die mit der darunter liegenden Schicht Daten austauscht.

SCHLÜSSELWORTE

Network Architecture, Netzarchitektur: Ein Konzept, das den Aufbau von und die Interaktion in Netzwerken beschreibt.

Overlaynetzwerk (kurz: Overlay): Eine bestimmte Form von Netzarchitektur, die auf einem vorhandenen Netzwerk, meist dem Internet, basiert. Overlays sind logische Netzwerke und verfügen meist über einen eigenen Adressraum mit eigener Adressierung. Ein bekanntes Beispiel für Overlays sind Peer-to-Peer-Netzwerke. In dieser Arbeit sollen aber nur die Overlays RON[3][4] und i3[5] näher betrachtet werden. (Das Internet selbst ist übrigens ein Overlay des Telefonnetzwerks.)

Legacy Application: Im Rahmen dieser Arbeit sind damit alle bekannten Netzwerkanwendungen gemeint, die über die IP-Schicht kommunizieren. Allgemein spricht man von einer „vererbten Anwendung“ bei Software, die bereits vor einer entscheidenden Neuerung im Gesamtkonzept ausgeführt wurde.

1. EINLEITUNG

Ein Hauptgebiet der aktuellen Netzwerkforschung ist die Aufhebung der funktionalen Grenzen, die das Internet hat. Ein in dieser Richtung oft eingeschlagener Ansatz ist die Entwicklung von alternativen Netzarchitekturen und als Spezialform von diesen, die der Overlaynetzwerke. Diese bieten Möglichkeiten wie ein effizienteres Routing der Datenpakete, Anonymität, die Adressierung von Hosts hinter einem NAT, die immer wichtiger werdende Mobility und viele weitere. So entstand über die letzten zwei Jahrzehnte eine Vielzahl an Vorschlägen und Realisierungen dieser Architekturen, die jedoch zum Großteil nicht in der Praxis getestet werden konnten. Ein Grund dafür ist die Schwierigkeit der Endnutzer, sich mit den nötigen Einstellungen auseinanderzusetzen, die zur letztendlichen Ausführung und Evaluierung der Overlays nötig wären. Somit ist eine Architektur erforderlich, die dem Nutzer für all seine Anwendungen von Browser bis Remote Desktop diese Arbeit abnimmt und diese mit dem Overlay kommunizieren lässt, als ob ihnen die sonst erwartete IP-Adressierung und DNS Name Resolution zugrunde liegen würden. Denn Overlays können sich anderer Schnittstellen bedienen, die diese Funktionen erfüllen. Hierzu gab es Lösungen, die sich auf ein bestimmtes Overlay bezogen, jedoch noch keine, die für alle Overlays funktioniert.

Diese Aufgabe soll OCALA, die Overlay Convergence Architecture for Legacy Applications lösen und damit einen einfachen Zugriff auf die nützlichen Funktionen, die Overlays bieten, ermöglichen. Im Rahmen der Implementierung sollen dabei vier Anforderungen erfüllt werden: (1) Anwendungen, die auf einem Host laufen, sollen gleichzeitig auf verschiedene Overlays zugreifen können, (2) die Verbindung mehrerer unterschiedlicher Overlays, sodass Nutzer in verschiedenen Overlays miteinander kommunizieren können, (3) die Möglichkeit, zwei Hosts sogar dann durch ein Overlay kommunizieren zu lassen, wenn einer von ihnen nur IP versteht und (4) Erweiterbarkeit, also die Möglichkeit, ein neues Overlay mit minimalem Aufwand in OCALA zu integrieren.

Diese Anforderungen unterscheiden OCALA von anderen Systemen wie Oasis[6], einem Projekt auf dem Gebiet der Overlays, das Anwendungen ihren Traffic über Overlays routen lässt. Einige andere Konzepte konzentrieren sich hingegen auf die Implementierung von Verknüpfungen von Overlays, also die mögliche Interaktion von Hosts in zwei verschiedenen Overlays untereinander. OCALAs Aufgabe soll es hingegen vor allem sein, die Benutzer einfach auf Overlays zugreifen zu lassen und ihnen so die damit verbundenen Vorteile näher zu bringen.

doi: 10.2313/NET-2011-05-2_04

Das Konzept von OCALA sieht dazu vor, die Architektur als eine Proxy zu implementieren, die unter der Transportschicht eine Overlaykonvergenzschicht einfügt. (Ihre genaue Funktionalität wird in Abschnitt 2.2 behandelt.) Die Implementierung als Proxy bietet dabei den Vorteil, dass für OCALA selbst wiederum auch keine Änderungen, weder an Anwendungen noch am Betriebssystem, nötig sind. Ferner sollen in dieser Arbeit die Möglichkeiten und Grenzen von OCALA aufgezeigt und erläutert, sowie eine genauere Darstellung der eigentlichen Funktionsweise gegeben werden.

2. FUNKTIONALITÄT ALLGEMEIN

Da OCALA unter der Transportschicht arbeitet, bietet es keine Unterstützung für Overlays der Transport- und Anwendungsschicht. Das Hauptaugenmerk liegt auf Overlays, die einen End-to-End-Pakettransfer nach dem Muster von IP unterstützen. Diese Overlays haben die Möglichkeiten, die Leistung des Internets zu erhöhen, neue Funktionen wie *Mobility*[7] zu implementieren oder, wie i3, ihre architekturspezifischen Vorteile zu bieten.

Wie schon angedeutet, ist die Adressierung in einem Overlay vollkommen den Entwicklern überlassen. Die einzige Beschränkung ist, dass jeder Endhost über einen eindeutigen Identifier (*ID*) in seinem Overlay erreichbar ist. Dazu kann, wie im Fall von RON, die IP-Adresse des Hosts verwendet werden. Andere Overlays bedienen sich jedoch eigener IDs, wie Hashes von Public Keys. Zusätzlich können zur Vereinfachung der Adressierung von Hand für Menschen besser lesbare Namen den IDs zugeordnet werden.

2.1 Die vier Ziele

OCALAs Aufbau und Funktionsweise lassen sich am besten mit den vier Zielen beschreiben, die sich die Entwickler gesetzt haben. Diese vier Ziele sind:

- **Transparenz**: Die Anwendungen sollen fehlerfrei funktionieren, auch wenn sie über einem Overlay an Stelle von IP laufen.
- **Überbrückung verschiedener Overlays**: Hosts in verschiedenen Overlays sollen Daten austauschen können und Hosts, die kein Overlay verwenden, sollen aus einem Overlay heraus erreichbar sein.
- **Funktionalität der Overlays offenlegen**: Die Nutzer sollen frei wählen können, welche Overlays und mit ihnen verbundenen Funktionalitäten sie für ihre Anwendungen einsetzen wollen.
- **Gemeinsame Funktionen bereitstellen**: Wichtige Funktionen, wie die auf dem Gebiet der Security, sollen nicht von Overlays übernommen, sondern in OCALA implementiert werden.

2.2 Overlaykonvergenzschicht

Wie bereits erwähnt, implementiert OCALA eine neue Schicht in das ISO/OSI-Modell. Diese befindet sich aus der Sicht der Anwendung an Stelle der Netzwerkschicht. Dies ist nötig, da das Overlay die sonst vorhandene Netzwerkschicht ersetzt. Diese Overlaykonvergenzschicht (siehe Abbildung 1), kurz OC, besteht aus zwei Subschichten, der overlayunabhängigen OC-I und der overlayabhängigen OC-D.

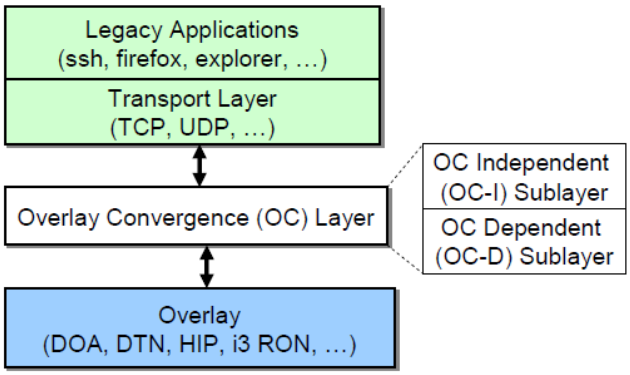


Abb 1. Lage und Aufbau der OC im Protokollschichtenstack[1]

Die OC-I simuliert für die Anwendung die IP-Schicht, nimmt von ihr also Daten entgegen. Außerdem ordnet sie die Daten den einzelnen Anwendungen und den von ihnen verwendeten Overlays zu und implementiert ein paar wichtige Funktionen, wie Authentifizierung und Verschlüsselung.

Die OC-D ihrerseits besteht aus einzelnen overlayspezifischen Modulen, die für das jeweilige Overlay eine Verbindung aufbauen und Pakete verschicken und empfangen können. In diesem Zusammenhang kann IP als ein Default Overlay gesehen werden, an das die Daten gehen, wenn kein eigentliches Overlay verwendet wird.

Abbildung 2 zeigt die OC in Aktion, ein Host A führt 3 Programme aus, mit Firefox greift er über das Internet auf *cnn.com* zu, über i3 chattet er anonym in IRC und lässt seine ssh-Daten mit RON effizienter routen. Dabei sieht man, dass Hosts B und C auch OCALA ausführen, da auch bei ihnen eine OC vorhanden ist. Dies ist nicht zwingend nötig, da OCALA Hosts auch über verschiedene Overlays hinweg kommunizieren lässt.

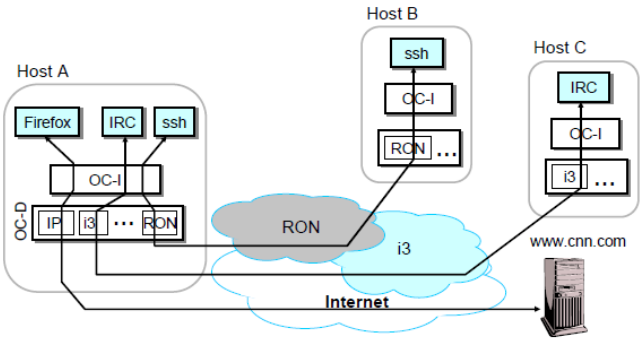


Abb. 2: Drei Verbindungen über drei verschiedene OC-D-Module[1]

Wie ein Host A über i3 einen Host C erreichen kann, der RON nutzt, zeigt Abbildung 3. Dazu wird ein Host B dazwischengeschaltet, dessen OC den Traffic aus i3 nach RON leitet. Man sieht, dass dazu aus dem i3-Modul der OC-D die Daten an die OC-I gehen. Der ganze Übergang von einem Overlay in ein

doi: 10.2313/NET-2011-05-2_04

anderes ist also nur ein zusätzlicher Hop auf der Route zwischen A und C.

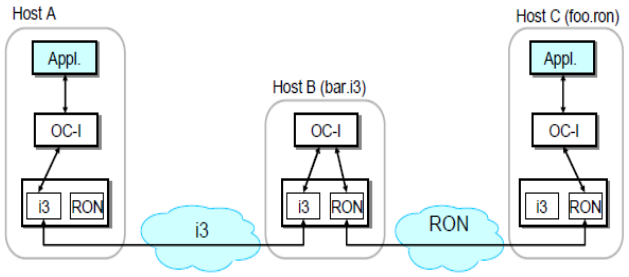


Abb. 3: Übergang aus einem Overlay in ein anderes[1]

Der Kommunikationskanal zwischen zwei Endhosts auf der OC-I wird dabei Pfad und der auf der OC-D Tunnel genannt. Im Beispiel aus Abbildung 3 ist der Pfad A,B,C und besteht aus den beiden Tunneln A,B und B,C.

Damit lässt sich eine Parallele zur Funktionalität der zweiten und dritten Schicht des ISO/OSI-Modells ziehen, denn so wie die niedriger liegende Sicherungsschicht (*data link layer*) die Verbindung zwischen zwei Knoten in einer Link Layer Domain implementiert, implementiert die OC-D-Subschicht die Kommunikation zwischen zwei Knoten in einem Overlay. Und wie die höhere Vermittlungsschicht (*network layer*) eine Kommunikation zwischen verschiedenen Link Layer Domains ermöglicht, ermöglicht die OC-I die Kommunikation zwischen verschiedenen Overlays.

3. GENAUE FUNKTIONSWEISE

Die Funktionsweise von OCALA lässt sich am besten mit der Realisierung der in Abschnitt 2.1 erwähnten Ziele beschreiben.

3.1 Transparenz

Um einen fehlerfreien Einsatz der Anwendungen zu ermöglichen, hat man an ihrem eigentlichen Funktionsschema angeknüpft. Dieses ist meist so, dass eine DNS-Anfrage gestellt wird und dann IP-Pakete mit der zurück erhaltenen IP-Adresse ausgetauscht werden. Allgemein haben Anwendungen zwei Möglichkeiten der Adressierung, entweder die beschriebene, wobei der DNS-Anfrage ein Name übergeben wird oder der direkte Weg über die Angabe einer IP-Adresse.

Ausgehend von diesem Konzept bietet auch OCALA zwei Möglichkeiten der Adressierung von Overlays an.

Die erste Möglichkeit ist die, den Benutzer Regeln festlegen zu lassen, wie Pakete anhand ihrer IP-Header zu versenden sind. So kann man Pakete für *64.236.24.4 Port 80* über RON und Pakete an *207.188.7.x* über i3 leiten lassen. Der Vorteil dieser Möglichkeit liegt dabei darin, dass es für jede Anwendung funktioniert, da jede (hier relevante) Anwendung IP-Pakete austauscht.

Die zweite Möglichkeit ist die, das zu verwendende Overlay im DNS-Namen anzugeben. Dazu gibt der Benutzer einen Namen der Form *bsp.ov* an, wobei *bsp* der (eindeutige) Name eines Hosts in dem Overlay *ov* ist. Erhält die OC eine Anfrage dieser Form, baut sie eine Verbindung zu *ov* auf und leitet den Traffic der sendenden Anwendung an *bsp.ov* weiter. Diese Möglichkeit bietet gleich mehrere Vorteile. Erstens können so Hosts adressiert werden, die keine eigene IP besitzen, was zum Beispiel bei Hosts hinter NATs der Fall ist. Zweitens sind die Namen für Menschen lesbarer und damit praktischer im Gebrauch als IPs. Drittens braucht der User die IP nicht zu kennen, denn oftmals kann er sie (aus verschiedenen Gründen) auch nicht kennen.

3.1.1 End-to-End Pfadaufbau

Ein Verbindungsaufbau kann durch zwei Umstände ausgelöst werden, erstens: eine DNS-Anfrage für einen zuvor unbekannten Overlayhost wird erhalten oder zweitens: den Empfang eines ersten Datenpakets aus einem Overlay. Am Ende des Verbindungsaufbaus soll ein End-to-End-Pfad zwischen den OC-I-Schichten der beiden Kommunikationspartner bestehen. Der Einfachheit halber wird im folgenden Beispiel nur ein Tunnel dafür verwendet.

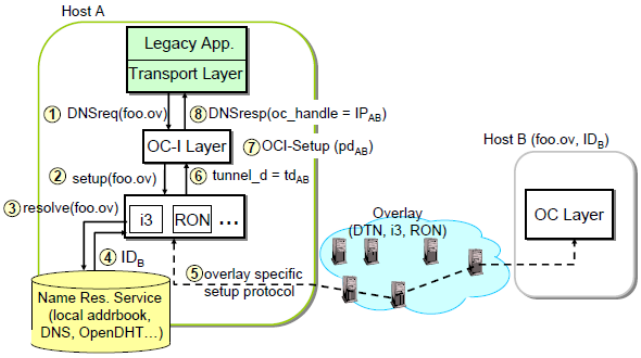


Abb. 4: Der Pfadaufbau in seinen Einzelschritten[1]

Abbildung 4 zeigt, wie eine Anwendung auf Host A mit einer Anwendung auf B, mit dem Namen *foo.ov* kommunizieren will. Zunächst wird eine DNS-Anfrage für *foo.ov* gestellt (1), welche die OC-I abfängt. Dabei erstellt die OC-I nebenbei einen *path descriptor* (*pd*), der den Pfad später eindeutig identifizieren soll. (Ein 128-Bit-Feld zur Kollisionsvermeidung.) Dann schickt die OC-I den Befehl, an das *ov*-Modul der OC-D, einen Tunnel zu foo.ov aufzubauen (2). Die OC-D ihrerseits benutzt den individuellen Name Resolution Service des Moduls um *foo.ov* zu finden (3). (Zur Erinnerung: diese Resolution Services können DNS oder ein overlayinterner Dienst sein, wie das Namenshashing in i3.) Der Name Res. Service liefert die zu *foo* gehörende ID an die OC-D zurück (4). Die OC-D hat nun alle Informationen, um eine Verbindung zu B aufzubauen. Sie tut dies mit einem für das Overlay definierten Protokoll (5). Ist der für den Datenaustausch benötigte Status erreicht, wird ein Pointer auf diesen Status, der *tunnel descriptor* (*td*) an die OC-I übergeben (6). Nach dem Empfang des Tunnel Descriptors führt die OC-I von A zusammen mit der OC-I von B einen OCI-Setup durch (7). Ist dies abgeschlossen, so übergibt die OC-I der Anwendung einen *oc_handle* in Form einer DNS-Response. Das *oc_handle* ist eine local-scope-IP, die der Verbindung mit foo.ov zugewiesen wird. Diese Adresse wird die Anwendung nun für den Paketaustausch mit foo.ov verwenden. Der vergebene IP-Adressenbereich beschränkt sich dabei auf *1.x.x.x*. Tunnel descriptors werden zwischen der OC-I und OC-D eines Hosts ausgetauscht, path descriptors zwischen den OC-Is verschiedener Hosts. Die Trennung von Tunnel und Path Deskriptoren hat den Vorteil, dass verschiedene paths denselben tunnel descriptor nutzen können. Z.B. Pfade (A,B,C) und (A,B,D) nutzen Tunnel (A,B).

doi: 10.2313/NET-2011-05-2_04

3.2 Überbrückung von Overlays

Es gibt einige Erreichbarkeitsprobleme, die Overlays betreffen, denn diese verfügen über sehr unterschiedliche Funktionalitäten. So können Hosts hinter einem NAT zwar über i3, jedoch nicht über RON erreicht werden, da i3 jedem Host eine ID zuordet während Hosts in RON über ihre IP-Adresse identifiziert werden. Außerdem sollen auch Hosts aus einem Overlay heraus erreicht werden können, die nur das Internet nutzen.

OCALA löst diese Probleme mit dem Konzept der *remote resolution*. Diese funktioniert so, dass wenn ein Host im Overlay ov1 eine Anfrage für *bsp.ov2* sendet, diese Anfrage an ein Gateway geleitet wird, das *ov2* ausführt. Dort geht sie aus dem *ov1*-Modul der OC-D an die OC-I und von dort wieder an die OC-D, jedoch an das zuständige *ov2*-Modul, wo sie aufgelöst wird.

Dass ein Host, der OCALA ausführt, zwar ein Overlay nutzen kann, aber auch die Möglichkeit hat, seine Anwendungen weiterhin direkt über IP kommunizieren zu lassen, ist bereits bekannt. Dazu wird der Traffic lediglich über das IP-Modul der OC-D geleitet. Was aber, wenn er nur aus einem Overlay heraus auf einen Internethost zugreifen möchte? Oder, was vielleicht noch wichtiger ist, was haben die Leute von OCALA, die nur über das Internet kommunizieren wollen? Diese Fragen löst OCALA mit dem Konzept der *legacy gateways*.

3.2.1 Legacy gateways

Legays gateways haben Ähnlichkeit mit Overlay gateways (vgl. Abbildung 3), wobei einer der beiden Tunnel über IP verläuft. Die Funktionalität des Overlays gilt für diesen Tunnel natürlich nicht. Es lassen sich zwei Arten von Legacy Gateways unterscheiden.

Das legacy server gateway ermöglicht die Verbindung eines Overlay clients mit einem *legacy server*. Er bekommt die Daten vom OC-D-Modul des Overlays, das der Host verwendet, übergibt sie der OC-I-Subschicht und leitet sie von dort an das sogenannte *LegacyServerIP*-Modul. Das legacy gateway verhält sich in diesem Fall wie ein NAT des *overlay clients*, es weist der Verbindung einen seiner lokalen Ports zu und ändert entsprechend die Adressen im IP-Header. (siehe Abbildung 5).

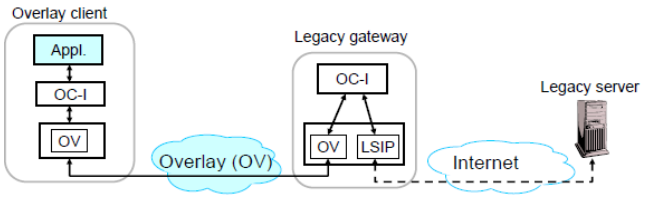


Abb. 5: Legacy server gateway in Aktion[1]

Das legacy client gateway lässt Hosts, die sich weder in einem Overlay befinden, noch über eine OC-Schicht verfügen, eine Verbindung zu einem Overlay herstellen. Der Traffic vom *legacy client* wird mit einem *LegacyClientIP*-Modul empfangen, zur OC-I geleitet und von dort aus zum gewünschten Overlaymodul der OC-D transferiert. Sobald die Verbindung aufgebaut wurde, schickt das legacy gateway eine DNS-Response mit einer IP-Adresse an den Client. Immer, wenn der Client nun IP-Pakete mit dieser IP schickt, werden sie an den *overlay server* übermittelt (siehe Abbildung 6). Ein bestehendes Problem dieser Methode ist aber, dass die im DNS-Response enthaltene IP-Adresse routbar sein muss. Dies beschränkt die Anzahl der möglichen Verbindungen mit dem legacy gateway.

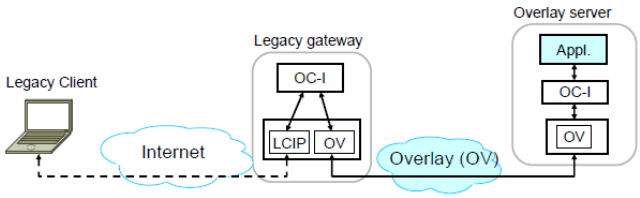


Abb. 6: Legacy client gateway in Aktion[1]

3.3 Overlayfunktionalität offenlegen

Jedes Overlay bietet seine eigenen, individuellen Möglichkeiten zur Verbesserung der Datenübertragung. In RON kann man z.B. selbst die Routingmetriken für Pakete angeben und mit i3 kann man sogenannte Middleboxes in einen Pfad einfügen. Der Nutzer von OCALA soll dabei nach eigenem Ermessen für jeden Tunnel entscheiden können, welche Optionen für ihn sinnvoll sind und auf welche er verzichten möchte.

Der erste Ansatz, dies möglich zu machen, war, in die DNS-Namen eine Syntax der Form *bsp.delay50ms.overqos* einzufügen. Dieser Befehl bewirkt eine Verbindung mit weniger als 50 Millisekunden Delay an den Host mit dem Namen *bsp* im *OverQoS*-Overlay[8]. Diese Idee hat sich aber in der Praxis als unhandlich herausgestellt und wurde, trotz ihrer bestehenden Implementierung in OCALA, nicht verwendet. Stattdessen wurde eine GUI erstellt, in der der Benutzer vor dem Verbindungsaufbau die overlayspezifischen Optionen festlegen kann. Diese GUI schreibt entsprechende Einträge in dafür bestimmte Konfigurationsdateien im XML-Format. Der User kann aber auch direkt auf diese zugreifen und Änderungen an ihnen vornehmen.

3.4 Gemeinsame Funktionen bereitstellen

Um die Arbeit der Overlayentwickler zu erleichtern, implementiert OCALA auf der OC-I-Subschicht generische Funktionen, die für alle Overlays nützlich sind. Dazu zählen derzeit Securityfunktionen und Datenkomprimierung.

Die Securityfunktionen bieten Verschlüsselung von Daten und Authentifizierung von Verbindungen. Beide beruhen auf der Annahme des Vorhandenseins einer zentralen Stelle für Zertifizierung und Namenszuweisung im Overlay. Das Protokoll, das OCALA dafür verwendet, ist dem von SSL ähnlich.

4. OCALAS PROBLEME UND MÖGLICHKEITEN

Die Ausführung von OCALA bringt neben vielen Vorteilen auch ein paar Probleme mit sich. Im Folgenden sollen diese beschrieben und anschließend mit den einzelnen Möglichkeiten, die OCALA bietet, aufgewogen werden.

4.1 Probleme

Zunächst werden Overlays der Transport- und Anwendungsschicht auch in Zukunft nicht unter OCALA laufen, da diese sich über der OC einfügen. Diese Overlays können aber dem Nutzer einige nützliche Funktionen bieten.

Intuitiv drängt sich auch die Frage nach dem mit OCALA verbundenen Rechenaufwand auf. Wie die Tests der Entwickler

doi: 10.2313/NET-2011-05-2_04

zeigten, sind die mit dem zusätzlichen Overhead verbundenen Kosten zwar bei lokalen Netzwerken gering, können aber bei Fernnetzen schnell ansteigen. Dies hängt aber auch mit der Ausprägung der Overlaynetzwerke zusammen.

Ferner unterstützt OCALA derzeit nur Unicast-Anwendungen. Anwendungen, die sich Multicast bedienen, laufen unter OCALA nicht. Dies ist jedoch eine Aufgabe, an der die Entwickler von OCALA noch arbeiten werden[1].

Wie schon bekannt, liefert die OC-I lokale IP-Adressen. Die damit verbundenen Probleme sind erstens, dass diese Adressen auf anderen Hosts nicht erreichbar wären. Zweitens würden Anwendungen, die ftp benutzen, welches IP-Adressen in Datenpaketen verschlüsselt, nicht mit OCALA vereinbar sein, da die OC-I die IP-Header vor der Übermittlung überschreibt. Obwohl dies in einem Overlay durch gemeinsame Verhandlung über die der Anwendung übergebenen IP-Adresse verhindert werden kann, werden beim Übergang eines legacy gateways stets die Adressen überschrieben.

4.2 Möglichkeiten

NAT Traversal: Da sich über i3 auch Maschinen hinter NATs erreichen lassen, ist es mit dem i3-Modul der OC-D möglich, auch einen legacy server hinter einem NAT aufzustellen. Außerdem kann man sich so bequem von überall aus mit dem Heimcomputer verbinden, indem man seinen i3-Namen adressiert.

Middleboxes: In i3 kann man angeben, ob man eine direkte Verbindung wünscht oder über eine Middlebox, die als zusätzlicher Hop auf dem Weg zum Empfänger eingefügt wird, kommunizieren will. Auf diesen Middleboxes lassen sich Anwendungen ausführen, die Funktionen wie *intrusion detection* für die Verbindung ermöglichen.

Abgesicherte Mobility: Mit HIP[9] lässt sich eine abgesicherte Verbindung zwischen zwei mobilen Hosts herstellen. Dieses Feature nutzt OCALA, um eine *ssh*-Verbindung selbst dann aufrecht zu erhalten, wenn einer der Hosts seine IP ändert.

Sicherheit und Flexibilität in VPNs: Die Entwickler von OCALA bieten einen Vorschlag zur Verbesserung der Sicherheit und Flexibilität in VPNs. Dazu wird intern ein legacy server gateway zwischengeschaltet, durch das alle Daten von externen Hosts (aus Overlays) an interne geleitet werden. Die Sicherheitsfunktionen der OC-I spielen dabei eine wichtige Rolle. Damit kann der Benutzer gleichzeitig auf mehrere Intranets zugreifen, selbst wenn sie alle denselben Adressraum verwenden. Ein weiterer Vorteil ist, dass der externe Host keine interne IP-Adresse zugewiesen bekommt. Das erschwert Scanangriffe auf das interne Netz.

Overlay Composition: Ohne OCALA kann der User lediglich ein Overlay verwenden, so ist es für ihn nicht möglich, die Funktionalitäten verschiedener Overlays miteinander zu kombinieren. Durch OCALA ist dies mit der Zwischenschaltung von Overlaygateways möglich. Der User kann also mit Hilfe von i3 ohne Verbindungsverlust zwischen verschiedenen drahtlosen Netzwerken wechseln und die Routingentscheidungen dabei von RON optimieren lassen.

5. ZUSAMMENFASSUNG

Die Overlays stellen mit Sicherheit eine Verbesserung im Hinblick auf die Funktionen des normalen Internets dar und können deshalb in Zukunft mehr an Bedeutung gewinnen. Dank OCALA können sie, wenn es so weit ist, auch den Endnutzern zugänglich gemacht werden. Ein aktueller Trend in der Entwicklung des Internets lässt sich aber leider nur schwer erkennen. Seit Jahren wird von der Umstellung von IPv4 auf IPv6 gesprochen, die auch weiterhin auszubleiben scheint. Mit Hilfe von Overlays könnte aber genauso gut der Adressraum von IPv4 erweitert werden. Dies ist für mich ein Zeichen dafür, dass das Potential, welches Overlays mit sich bringen, weitgehend unterschätzt wird.

6. LITERATUR

[1] Dilip Joseph, Jayanthkumar Kannan, Ayumu Kubota, Karthik Lakshminarayanan, Ion Stoica, Klaus Wehrle: "*OCALA: An Architecture for Supporting Legacy Applications over Overlays*", San Jose, CA, USA, Mai, 2006.

[2] OCALA, online: http://ocala.cs.berkeley.edu/

[3] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. *Resilient Overlay Networks*. In *Proc. of SOSP*, 2001.

[4] Resilient Overlay Networks, online: http://nms.lcs.mit.edu/ron/

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. *Internet Indirection Infrastructure*. In *SIGCOMM*, 2002.

[6] H. V. Madhyastha, A. Venkataramani, A. Krishnamurthy, and T. Anderson. Oasis: *An Overlay-aware Network Stack*. *SIGOPS Operating Systems Review*, 40(1), 2006.

[7] P. Yalagandula, A. Garg, M. Dahlin, L. Alvisi, and H. Vin. *Transparent Mobility with Minimal Infrastructure*. Technical Report TR-01-30, UT Austin, June 2001.

[8] dL. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz. *OverQoS: An Overlay-based Architecture for Enhancing Internet QoS*. In *Proc. of NSDI*, 2004.

[9] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. *Host Identity Protocol*, 2003. http://www.hip4inter.net/documentation/drafts/draft-moskowitz-hip-08.html

doi: 10.2313/NET-2011-05-2_04

Security, Privacy and Cloud Computing

Jose Tomas Robles Hahn
Supervisor: Ralph Holz
Future Internet Seminar - Winter Term 2010/2011
Chair for Network Architectures and Services
Faculty of Computer Science, Technische Universität München
Email: jtrh@mytum.de

ABSTRACT
Cloud computing usage is growing every day as users discover its many advantages. However, the move to the cloud exposes users to new security and privacy threats. We give details on cloud-specific vulnerabilities and caveats and show some technical ways to address them. We also note that users cannot rely on technology alone to completely solve all their problems: how to protect against government surveillance remains an open question.

Keywords
Cloud computing, Security, Privacy, Surveillance, Encryption

1. INTRODUCTION
In the 1980s, the emergence of the personal computer (PC) revolutionized the relationship between society and computers. Before the PC, computers were the domain of governments, large enterprises and academic institutions. The massification of the personal computer, together with the Internet, enabled a new world of possibilities: electronic commerce, 24 hour access to information located anywhere on Earth, and much more.

Today, we are again experiencing an exciting paradigm shift in the IT industry. Cloud computing solutions are being massively adopted by governments, businesses and consumers. The market research company IDC forecasts a threefold growth of customer spending on cloud computing services, reaching US$ 42 billion by 2012 [1].

Cloud computing brings many benefits to its users. Businesses and government agencies can enjoy substantial cost-savings in IT infrastructure and increased flexibility to react to changes in requirements of computing power. Consumers are provided with ubiquitous access to their data and redundant storage, which protects them against inconveniences such as having to carry their computer everywhere, or data loss caused by a failing, non-backed up disk drive [2].

Unfortunately, certain groups are exploiting cloud computing to the detriment of its legitimate users. We are talking about hackers and governments: These two groups both wish to covertly access users' data, but for different reasons. Computer criminals want to access data such as credit card numbers, bank login credentials, financial records and other confidential information in order to gain profit, while governments want access to that data in the name of fighting crime and terrorism. Cloud computing puts users' data at a higher risk of being accessed by unauthorized individuals, since their information is now stored in datacenters which they do not control. This also allows governments to invade users' privacy by getting their data directly from the cloud provider, without informing the affected user [2].

In this article we will discuss the security and privacy risks that users face by moving their data to the cloud and show how we can use technology to solve them. We will base our discussion on Christopher Soghoian's article *Caught in the Cloud: Privacy, Encryption, and Government Back Doors in the Web 2.0 Era* [2].

This article is organized as follows: First, in chapter 2 we will talk about the definition and characteristics of cloud computing. We will also talk about the benefits that cloud computing brings to the table. Then, in chapter 3 we will explore the security challenges cloud computing faces, the technologies that exist to solve those challenges, and some actual security attacks. Finally, we discuss cloud computing privacy issues in chapter 4.

2. CLOUD COMPUTING
2.1 Defining cloud computing
In the last decades, the computing paradigm has evolved substantially. Voas and Zhang identified six distinct phases [3]. The first phase corresponds to the traditional terminal era, where mainframes shared by many users did all the hard computing work. In phase two came the now ubiquitous personal computer. In the next phase, local networks appeared. Phase four came to be with the interconnection of local networks, which ultimately resulted in the Internet. In the following phase came distributed computing. Finally, in phase six, cloud computing made its appearance, opening the door to access a potentially unlimited amount of computing resources, in a scalable and simple way.

But, what is cloud computing? How do we define it? Unfortunately, so many different products in the market are associated to cloud computing by their respective manufacturers, that it becomes difficult to specify what cloud computing actually is. Larry Ellison, co-founder and CEO of Oracle Corporation, when asked about cloud computing while at a conference, said *"I have no idea what anyone is talking about"*, *"We've redefined cloud computing to include everything that we already do"*, and *"I can't think of anything that isn't cloud computing with all of these announcements"* [4].

doi: 10.2313/NET-2011-05-2_05

A notable definition was created by the U.S. National Institute of Standards and Technology (NIST) [5]. According to their definition, cloud computing has five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. They also define three cloud service models (which can be viewed as layers, as shown in figure 1), which must be deployed on top of cloud infrastructure that has the five essential characteristics mentioned above:

- Cloud Software as a Service (SaaS): Provider offers users access to its application over a network. Usually, this is implemented as a Web application. Providers in this category include Facebook [6], Google Apps [7] and Google Mail [8].
- Cloud Platform as a Service (PaaS): Provider offers a platform where users can deploy their own applications. Some well-known providers in this category are Microsoft Windows Azure Platform [9] and Google App Engine [10].
- Cloud Infrastructure as a Service (IaaS): Provider offers computing resources such as processing power, storage and network capacity. Some providers in this category include the Amazon Elastic Compute Cloud (Amazon EC2) [11] and Rackspace Cloud Servers [12].

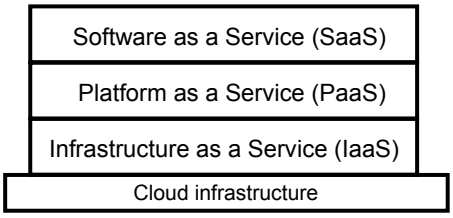


Figure 1: Cloud computing service models viewed as layers, deployed on top of cloud infrastructure that has NIST's five essential cloud computing characteristics. (Image source: Own work)

As in Soghoian's article [2], this paper will focus on Web applications executed in a Web browser, where the application's code is downloaded as needed from a remote server that also stores users' files. So keep in mind that when we talk about cloud computing applications, we will be referring to Web applications.

2.2 Cloud computing in practice

In the personal computing paradigm, users' can run locally installed applications like word processors, spreadsheets, personal financial management software, picture organizers, and so on. Thus, their data is always stored locally, i.e. in their own computer [2]. They maintain physical control over their data and therefore must assume the responsibility which that entails: For example, they have to take measures themselves to keep their data safe from hardware failures (e.g. making backups regularly), and they have to ensure that their data is available at any place they need it (e.g. copying it to a USB flash drive).

Today, users are increasingly moving their data to the cloud. Email, which requires Internet access to check for new messages, was not surprisingly, the first application to move [2]. Some time later, other applications became available online, such as office suites like Google Apps [7] and Microsoft Office Web Apps [13], and picture editors, like Adobe Photoshop Express [14].

We will now discuss the benefits of cloud computing for service providers, businesses and consumers.

- Cloud service providers benefit from cloud computing because it solves the software piracy problem, since part of the software code resides exclusively on the providers' servers. For the same reason, trade secrets such as in-house developed algorithms are also safe from reverse engineering [2].

  A purely economic advantage is vendor lock-in: as customers cannot easily take their data from one provider to a competitor (unless the provider itself provides them with data export functionality, and the competitor with import functionality), they are discouraged from changing cloud providers [15].

  In the case of paid services, another economic advantage is the subscription payment model, where customers make periodic payments to the provider, which in the long term can amount to much higher revenues than one-time purchases typical of classic software products [16].
- Business users benefit from not having to maintain a datacenter, redundant storage, having less IT personnel costs and turning IT infrastructure from a fixed cost into a variable cost. Another important advantage of cloud computing is the flexibility to acquire additional computing capacity only as required, saving money on investments made to support infrequently occurring peak loads.
- Consumers benefit from many cloud service providers offering their Web applications for "free" or cheaper than their desktop counterparts. Note that in this case, free often means subjecting oneself to targeted advertising and data mining [17]; and that cheaper does not necessarily mean long term savings [16] (after some time, the customer may have already paid the cost of a one-time software purchase). Another benefit is that the cloud provider handles backups and hardware failures. A useful feature in today's mobile world is having ubiquitous access to your data: anywhere in the world, just an Internet connection is needed, even a mobile phone is enough.

As Internet is not available everywhere yet, offline access to data stored in the cloud is an important feature that Web applications do not normally have. To solve this issue, Google created Gears [18], which allows Web applications like Google Mail to store a copy of users' mail on their computers. Google announced in 2010 [19] that Gears would be replaced in the future by HTML5 Web Storage [20].

As Web applications look more and more like their desktop counterparts, it is becoming increasingly difficult for users

doi: 10.2313/NET-2011-05-2_05

to realize whether they are using a desktop application or a Web application, and where their data is actually stored [2]. This confusion highlights the importance of security and privacy in cloud computing: If users are going to use Web applications without noticing it, then cloud computing should be made as secure and private as possible.

3. CLOUD COMPUTING SECURITY

Security is an important subject in today's networked world, and cloud computing, still in its infancy, is no exception. Two years ago, a survey by IDC [21] showed that security was the top challenge ascribed to the cloud computing model. Thus, security vulnerabilities can even affect the decision of whether to adopt a cloud computing solution at all.

In this chapter we will discuss the concept of cloud-specific vulnerabilities, followed by which issues are affecting cloud security together with security technologies to solve them, and finally, some concrete security attacks.

3.1 Cloud-specific vulnerabilities

Cloud Web applications depend on widely used technologies, such as DNS, TLS and Web browsers. This means that the vulnerabilities of those technologies could *also* be considered vulnerabilities of Web applications. But, are there any vulnerabilities *specific* to cloud computing? How can we classify a vulnerability as *cloud-specific*?

Grobauer, Walloschek and Stocker attempted to answer that question in [22]. They consider a vulnerability to be cloud-specific if at least one of the following conditions is met:

- The vulnerability is intrinsic to or prevalent in a core cloud technology, such as Web applications and services, virtualization and cryptography.
- The vulnerability's root cause is one of the five essential cloud characteristics identified by NIST (see also section 2.1 and [5]).
- The vulnerability is caused by cloud innovations making the implementation of security best practices difficult.
- The vulnerability is common in most modern cloud offerings.

Although the following security weaknesses and vulnerabilities also apply to non-cloud technologies, we conclude from the above conditions that they can be considered cloud-specific.

3.2 Security weaknesses in cloud computing

In this section we will talk about security weaknesses and caveats affecting cloud-related technologies.

3.2.1 Cloud providers fail to provide encryption to their users

Soghoian has strongly criticized cloud service providers for not providing encrypted access to their Web applications [2]. For example, webmail providers such as Yahoo! Mail [23], encrypt their login page with HTTPS, but then revert to plain HTTP. Although users' passwords remain protected, session cookies, together with the reading and writing of email messages, are transmitted *in the clear* (i.e. unencrypted). Another example is Facebook, which encrypts the transmission of login credentials, but not the login page itself. Therefore, Facebook users cannot easily verify that they are filling the form on the real Facebook website.

Soghoian argues that there is no economic incentive for cloud providers to provide encrypted access and encrypted storage by default [2]. One reason for this is the higher operating cost of encrypted access, which demands more processor time per client connection to sustain the same number of unencrypted connections, requiring additional hardware purchases in order to keep quality of service constant. If the cloud service is provided for "free", then there is even less incentive for cloud providers to provide encryption, as providing free service is not free for them. To pay for their costs, free cloud providers mine users' data so they can show highly targeted advertisements to users [17]. If that data were stored encrypted, it would not be possible to analyze it for advertising purposes.

Our last point concerns market demand: If users do not demand encryption from cloud providers, they will probably never offer it. A reason for this situation is lack of information: Cloud providers do not openly disclose to users the risks to which their data is subject [2]. From insider attacks [42] to government surveillance, there are enough reasons to desire encryption (for more information, see also chapter 4).

3.2.2 Man-in-the-middle attacks

This is an attack form in which the attacker redirects traffic between a client and a server through him, so that he can log and possibly also alter the communication. Both client and server believe they are talking directly to each other. This attack is normally implemented by tricking the client into connecting to the attacker instead of the desired server and then relaying the traffic to the real destination [24]. Man-in-the-middle (MITM) attacks can be perpetrated by forging DNS packets, DNS cache poisoning, or ARP spoofing, for example. DNSSEC and HTTPS/TLS are two technologies that can prevent MITM attacks (see sections 3.3.4 and 3.3.2, respectively).

3.2.3 Data encryption caveats

Before implementing a data encryption technology, some important questions need to be considered. Their possible answers illustrate the limits of data encryption.

- Where will the encryption key be stored? If the cloud provider is in possession of the key, the customer must trust the service provider not to use it for unauthorized purposes and to store it safely outside the reach of hackers. Furthermore, it is important to know that the cloud service provider may be forced by law enforcement to disclose the encryption key to them, sometimes without being allowed to inform the customer (using a so-called *gag order*) [25]. An example of a

doi: 10.2313/NET-2011-05-2_05

cloud service provider that stores customer data in encrypted form is Hushmail [26].

- Where will the encryption and decryption processes be performed? If the cloud service provider is storing the key, then it will also perform the encryption and decryption. Further discussion deserves the case where the customer is in sole possession of her key. On the one hand, if encryption and decryption procedures are executed exclusively on the customer's premises, then it is guaranteed that the cloud provider does not have access to the encrypted data even for a single instant. On the other hand, if the customer supplies her key to the cloud provider each time she needs to encrypt or decrypt data, so that the cloud provider performs the encryption or decryption and then deletes the customer's key from memory, then the customer's key is at risk.

  An example that shows the importance of this question is the case of a drug dealer that used Hushmail's secure email service [27]. Hushmail offers both a server-side and a client-side encryption mode. The more secure client-side encryption mode is done using an open source Java applet, while the less secure server-side encryption mode is performed through a webmail interface where the client supplies the passphrase to his key when needed, which is immediately deleted from memory after use. Law enforcement officials ordered Hushmail to record the customer's passphrase instead of deleting it from memory and then used it to decrypt all the customer's mail. Note that the open source Java applet would not have saved the drug dealer, because law enforcement can also order Hushmail to supply the customer with a backdoored applet. To be safe from that backdoor, a client would have to read and compile the applet's source himself, which is more work than just accepting Hushmail's compiled binary.

Irrespective of which choices are made regarding the above two questions, we also want to make you aware of the following caveat, which also applies to network encryption: encryption is not a magic solution; encrypted data can be stored indefinitely until enough computing power is available to decrypt it. What today is considered impossible may be feasible in, say, five or ten years.

3.2.4 User interface attacks

Web applications are accessed through a Web browser, so the browser's user interface becomes an important security factor.

One kind of user interface attack is that in which an attacker tries to fool the user into thinking that she is visiting a real website instead of a forgery. Techniques used here include fake HTTPS lock icons, which are only detected by attentive users [38], homographic attacks with international characters that look like certain national characters [38], and browser software vulnerabilities, which can trick the browser into showing incorrect information, like a fake URL in the address bar.

In section 3.4.2 we will look at a security attack that exploits the Web browser's user interface.

3.3 Security measures in cloud computing

We will now present some security technologies that are relevant to cloud applications.

3.3.1 Single site browsers

Users do not need to download or install Web applications — they just execute them in a Web browser. That same Web browser is also used to interact with sensitive websites such as banks or webmail. Browsers also store a history of all the websites a user visited, and often, website passwords. All that information, stored in a single place, is at risk of being stolen by hackers exploiting Web browser vulnerabilities [2].

Single site browsers, also known as site-specific browsers, seek to reduce that risk by creating a separate browser instance for a Web application. The most advanced single site browser technology is Mozilla Prism for Firefox [28], which allows users to create a dedicated shortcut on their desktop for a Web application, which will open a dedicated browser window. This dedicated browser instance maintains its own preferences and user data, which is safe against access by malicious websites and Web applications running in separate Prism sessions or Firefox windows.

However, single site browsers are not all about security. They also improve usability by hiding user interface elements such as the toolbar and the address bar [2]. This makes sense, since the back and forward buttons are document-oriented and therefore not suitable for an application. Even though these usability improvements lower the adoption barrier of Web applications, they come at a price. Since all user interface elements are hidden, users have no way to verify that they are connected through a secure connection [2]. It remains to be seen how Web browser vendors attempt to solve this problem.

3.3.2 Network encryption: HTTPS, TLS and PKI

Network encryption protects data as it travels from one computer to another. Once data arrives at its destination, it may either be stored encrypted (see section 3.3.3) or unencrypted.

The most popular network encryption technology used to secure communication between cloud clients and Web applications is the Hypertext Transfer Protocol Secure (HTTPS), which is a combination of HTTP with the Transport Layer Security (TLS) [29] protocol. TLS resulted from the standardization of the Secure Sockets Layer 3.0 protocol. It provides confidentiality, integrity and authentication between clients and servers.

The most common authentication method used on the Web today is the Web server providing the Web browser with a certificate, which contains its public key together with a digital signature that binds it with an identity. Since the Web server's certificate is provided through an insecure channel, the Web browser must have a way to verify that the certificate it received actually came from the Web server it is talking to. This is assured through a Public Key Infrastructure (PKI). In a PKI, an independent, trusted entity called a Certificate Authority (CA) is in charge of verifying that a certain public key is associated to a certain identity. This attestation is provided in form of a digital signature with

doi: 10.2313/NET-2011-05-2_05

the CA's private key. Web browsers verify this digital signature using the CA's own certificate, which is supposed to be obtained through a secure channel, but normally comes pre-installed together with the operating system or browser.

TLS certificates may be revoked for various reasons. A revocation means that the CA does not consider the certificate to be valid anymore. Possible reasons may include, for example, fraudulently obtained certificates or a legitimate certificate owner's private key being compromised. TLS implementations, such as Web browsers, need some way to verify that a certificate has not been revoked. One method to accomplish this task are Certificate Revocation Lists (CRLs) [30], which are issued periodically by CAs and contain a signed timestamped list of serial numbers of revoked certificates. A disadvantage of CRLs is that revocation reports will not be published until the next periodic update. For critical applications, such a delay may not be acceptable. The Online Certificate Status Protocol (OCSP) [31] is an alternative to CRLs that makes revocations available as soon as they are issued by the CA. All modern Web browsers try to check the certificates they receive with OCSP before accepting them.

3.3.3 Data storage encryption

Cloud service providers store users' data on servers outside the control of their customers. Customer data is at risk of being accessed by unauthorized individuals such as hackers, thieves, and even datacenter employees [42]. Encryption is the tool of choice to protect that data.

Some important aspects to consider before implementing data storage encryption were discussed in section 3.2.3.

3.3.4 DNS security: DNSSEC

When a user types a Web application's URL in her Web browser's address bar (e.g. `http://www.facebook.com/`), one of the first actions the Web browser takes is using DNS to find out which IP address (`69.63.190.18`) corresponds to the hostname in the URL (`www.facebook.com`).

DNS can be viewed as a tree-structured distributed hierarchical database of *zones* [32]. A zone is an independent administrative entity that contains resource records (RRs) describing many different types of information, like IP address-to-name mappings and delegations. Delegation RRs indicate that a zone (the *parent zone*) has assigned responsibility for a certain subset of it (the *child zone*) to a different name server.

Now let us examine the process of finding out the mapping of a name to an IP address: First, the Web browser (a *DNS client*) contacts a stub resolver provided by the operating system, which in turn contacts a local recursive or resolving name server (a *resolver*). The resolver then queries successively all necessary name servers in the hierarchy starting from the root zone and stopping either at the requested RR or at an error. Finally, the result is passed to the stub resolver, which then informs the browser about either the requested IP address or a look up error. Of course these steps were greatly simplified, for example, by ignoring DNS caches present on client computers and resolvers.

We now know why DNS security is important for cloud computing: the mapping of names to IP addresses. If a malicious attacker could somehow manage to modify the IP address that a client received when trying to access `www.facebook.com`, he could redirect to his own server all packets sent by the client and then forward them to Facebook's server, of course after recording or altering them. This is the man-in-the-middle attack we mentioned in section 3.2.2. We will now present a DNS extension designed to ensure that DNS answers cannot be altered in transit.

The DNS Security Extensions (DNSSEC) [33] extend DNS by adding authentication and integrity to DNS RRs through a hierarchy of cryptographic digital signatures [32]. Resolvers can verify the digital signatures attached to DNS RRs they receive by following a chain of trust of public keys and digital signatures that starts at the root zone and goes through zone delegations until finally reaching the name server that stores the RR to be authenticated (see figure 2).

To achieve its task, the DNSSEC specification defines four new RR types [34]:

- DNSKEY (DNS Public Key): This RR stores the public key corresponding to the private key used to sign the zone's resource record sets (RRsets). Resolvers use the public key to validate and authenticate those RRsets. For the zones illustrated in figure 2, there are DNSKEY RRs in the root, `se.` and `nic.se.` zones.

- DS (Delegation Signer): The DS RR is stored in a parent zone and points to the DNSKEY RR of the name server responsible for the child zone. That DNSKEY RR is stored only in the child zone's name server. Because the DS RR is signed, a chain of trust consisting of linked DS and DNSKEY RRs is formed. For the zones illustrated in figure 2, there are two DS RRs: one in the root zone (pointing to the DNSKEY RR of the `se.` zone) and one in `se.` (pointing to the DNSKEY RR of `nic.se.`).

- RRSIG (Resource Record Digital Signature): RRSIG RRs store digital signatures for RRsets, which can be of any RR type, including the other three DNSSEC RR types and mappings of names to IP addresses. In figure 2, there would be RRSIG RRs signing the DNSKEY and DS RRs of `.`, `se.` and `nic.se.`, and the RR mapping `www.nic.se.` to `212.247.7.218`.

- NSEC (Next Secure): This RR type is used to authenticate *not found* answers. NSEC RRs form a chain linking all RRs in a zone, so that if a resolver requests a non-existent RR, it receives as answer an NSEC RR containing information about the two chained RRs that come *before* and *after* the requested RR (see also *Canonical Form and Order of Resource Records* in [34]). Since no RRs can exist between any two RRs of an NSEC chain, a resolver can be sure that the requested RR does not exist. Unfortunately this NSEC chain also allows attackers to follow the entire chain and obtain a list of all RRs in a zone. To solve this privacy and security problem, hashed names are used in the new NSEC3 RR [35].

doi: 10.2313/NET-2011-05-2_05

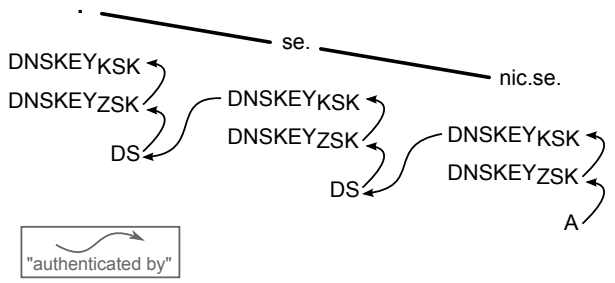


Figure 2: Following the DNSSEC chain of trust — An example: An *A* resource record (RR) mapping the name *www.nic.se.* to the IP address *212.247.7.218* is authenticated by the zone signing key (ZSK) of the *nic.se.* zone, which in turn is authenticated by the key signing key (KSK) of the same zone. The *DS* RR in zone *se.* authenticates the KSK of zone *nic.se.* and is authenticated by the ZSK of zone *se.*, which in turn is authenticated by the KSK of the same zone. In the root zone, there is a *DS* RR authenticating the KSK of the *se.* top-level domain (TLD). This *DS* RR is authenticated by the root zone's ZSK, which in turn is authenticated by the KSK of the same zone. Because the chain ends at the root zone, the authentication of the root zone's KSK must be performed by other means. (Image source: Own work)

We note that while DNSSEC provides end-to-end authentication (answers remain authenticated as they pass through intermediate DNS servers and are therefore protected against man-in-the-middle attacks), no hop-to-hop encryption is provided (an attacker is able to read requests and answers while they are in transit). DNSCurve, which encrypts packets but does not digitally sign them, complements DNSSEC by providing privacy for DNS traffic [36].

We end our discussion of DNSSEC with an interesting idea proposed in an RFC: Since DNSSEC can protect any RR type, we could distribute general-purpose certificates stored in signed CERT RRs, which could be used for applications such as secure email. This would provide an alternative to classic PKIs. More about this can be found in [37].

3.4 Security attacks

In this section we will present four security attacks that are of particular relevance to Web applications. They are concrete examples that show how the security weaknesses we discussed in section can be exploited in practice.

3.4.1 Signing TLS certificates with another site's certificate

TLS site certificates are supposed to be signed only by certificate authorities (CAs) or intermediate CAs designated by them. In this attack, Marlinspike discovered that it was possible to sign a TLS site certificate for *any* website of the attacker's choice, with just a site certificate legitimately obtained from an established certificate authority (CA), and have most browsers and other TLS implementations accept it as valid even though a site's certificate is not supposed to be able to sign other certificates [38]. This vulnerability was possible because even though TLS certificates can have their `BasicConstraints` field set to `CA:FALSE` (which means that the certificate cannot be used to sign other certificates), most CAs either did not bother to include that field at all or did not declare it as `critical` (fields set as critical must be obeyed by TLS implementations). Another reason was that most Web browsers and TLS implementations did not bother to check that field, even if it was present.

Marlinspike created a tool called *sslsniff* to exploit this vulnerability in an automated fashion. *Sslsniff*, supplied with a legitimately obtained site certificate for any domain, carries out a man-in-the-middle attack, intercepting HTTPS traffic and generating a certificate for the destination website signed with the supplied legitimate site certificate on the fly. Even though this vulnerability has been corrected in the meantime, *sslsniff* can still be useful as a general man-in-the-middle tool for TLS.

3.4.2 HTTPS via HTTP attack

The security researcher Moxie Marlinspike made the observation that most people arrive at HTTPS sites after being sent from an HTTP site [38]. Specifically, he named two ways: Clicking on links (or submitting a form) and through HTTP 302 redirects. For example, many online banking login pages are transmitted through HTTP. These pages normally either contain an IFRAME with an HTTPS login form or a form that posts the login credentials to an HTTPS URL. Facebook's login form [6] also features an insecure form that posts to an HTTPS URL.

Marlinspike proposed to attack not the HTTPS connection, as would be usual, but to attack the *HTTP* connection. For this purpose he created *sslstrip*, which, like *sslsniff*, is a man-in-the-middle attack tool. *Sslstrip* watches HTTP traffic looking for `https://...` links and changes them to `http://...` links pointing to a Web server controlled by the attacker, keeping track of what was changed. The attacker's Web server proxies the HTTP requests as HTTPS to the destination Web server and also rewrites all `https://...` links sent back to the client. Web browsers do not show any warning because only HTTP traffic is seen by them. For additional believability, *sslstrip* can watch out for favicon requests and send a lock icon to make the fake website look even more real.

3.4.3 Null prefix TLS attack [39] [40]

In a TLS certificate, the website hostname to which it belongs is specified in the Common Name (CN) field in the subject of the certificate.

Certificate Authorities (CAs), before signing a certificate, normally verify ownership of the domain name specified in the CN field, without caring about any subdomains and without verifying other subject information like Organization (legal name of the subject) or Country.

The CN field is represented as a Pascal string, which in its memory representation specifies first the length of the string, and then the string itself. This is different from C strings, which are just sequences of characters that are terminated by a single null character. The structure of Pascal strings

doi: 10.2313/NET-2011-05-2_05

has the effect that null characters are treated as any other character. Marlinspike observed this characteristic and realized that he could include null characters in the CN field, so that for example, he could generate a certificate signing request (CSR) for `www.facebook.com\0.attacker.org`. A certificate authority will ignore the null character in the CN and only verify the ownership of the `attacker.org` domain, because, as we said above, the CA does not care about subdomains. This verification procedure is usually just an email message to the registered contact of the domain `attacker.org`, which of course the attacker himself controls.

Of course this attack is not yet complete as we have not said anything about the role of the Web browser. Marlinspike noticed that most TLS implementations treat the CN field as a C string, using C string comparison and manipulation functions. This means that when the browser compares the CN specified in the certificate (`www.facebook.com\0.attacker.org`) with the current website hostname (`www.facebook.com`), the comparison functions stops at the null character and returns *equal*.

If the certificate authority were to revoke the certificate for `www.facebook.com\0.attacker.org`, we would need to use Marlinspike's OCSP attack, which we will explain in the following section.

3.4.4 OCSP attack

The Online Certificate Status Protocol (OCSP) [31] enables Web browsers and other TLS implementations to verify that a legitimately obtained certificate is still considered valid by the certificate authority.

When a Web browser receives an apparently valid certificate from a Web server, before accepting it, it sends a verification request to the OCSP server specified in the certificate. The OCSP server sends a response that includes a response status (which can be *successful*, *malformedRequest*, *internalError*, *tryLater*, *sigRequired* or *unauthorized*) and a signed response data field.

Marlinspike noticed that even though a fake *successful* response status would fail due to the signature in the response field, the innocent-looking response status *tryLater* does not require a signature and can therefore be faked without difficulties [39] [41]. Most Web browsers, after receiving a *tryLater* response status code, give the certificate the benefit of the doubt, accepting it without alerting the user.

4. CLOUD COMPUTING PRIVACY

Having your personal data stored in a place outside your control is becoming commonplace thanks to cloud computing. In this final chapter we will briefly discuss the privacy challenges that cloud computing faces, together with the relationship between government and the cloud. We will also show how technology can help users regain control of their privacy.

4.1 Privacy challenges in the cloud

In this section we will briefly cover some aspects of cloud computing that may affect users' privacy.

A fundamental characteristic of cloud computing is having users' private data outside their physical control. This can have many consequences. For example, data could be mined by the cloud provider with, or even worse, without authorization. Excessively curious cloud datacenter employees could read users' private (unencrypted) data without their knowledge. A recent case involving Google demonstrates that these privacy risks must be taken seriously: A (now ex-)Google employee was caught spying on teen users, accessing their Google Voice call logs, chat transcripts and contact lists [42].

Another aspect to consider when storing data in a cloud provider datacenter is of a legal nature: Since users' data may be stored in a datacenter located anywhere in the world, their data could be stored in a foreign country without the user ever noticing. A foreign government could, for example, use surveillance techniques to help their companies gain unauthorized access to trade secrets. Leaving covert surveillance aside, users' must take into account that the laws of the government where the datacenter is located may be different from the laws of the country where the user lives.

The challenges presented in this section could be solved through network *and* data encryption [2]. However, in section 4.2 we will introduce a different kind of adversary, one which cannot be defeated solely through encryption.

4.2 Government and the cloud

We will now focus on privacy threats coming not from private actors, such as hackers, but from the government.

4.2.1 Situation in the United States

In the United States the government has been continuously expanding its use of surveillance. Soghoian argues that this is happening because technology has drastically lowered the cost of spying on its citizens [2]. He proposes encryption as the definite solution against government intrusion, but recognizes that the government can force a cloud provider to insert backdoors into its software in order to circumvent the encryption.

One of the most important legal tools used by the U.S. Government to force cloud providers to hand them users' private data is the third-party doctrine. Other relevant laws include the Wiretap Act, the All Writs Act and the Foreign Intelligence Surveillance Act [2].

The Fourth Amendment to the United States Constitution protects U.S. citizens against unreasonable search and seizure, dependent upon a person's reasonable expectation of privacy. Unfortunately, the Fourth Amendment does not protect data stored in the cloud. The third-party doctrine establishes that a person does not have an expectation of privacy regarding information they share with others [2]. Courts consider that a user giving data to a cloud provider is sharing the data with them.

Until now we have not shown an actual example of what information the government can obtain from a cloud provider. Facebook's *Subpoena / Search Warrant Guidelines* [43] and Microsoft's *Global Criminal Compliance Handbook* [44] offer a few glimpses of that. For example, Facebook can supply

doi: 10.2313/NET-2011-05-2_05

law enforcement with a user's complete profile information and uploaded photos, irrespective of her privacy settings.

4.2.2 Situation in Germany

The United States is not alone regarding intrusions into their citizen's privacy. Even though Germany fares very well in Forrester Research's Data Protection Heat Map [45], that map bases its evaluation on each country's data protection laws, which do not cover government surveillance. For a taste of what the German government can do or wants to do, read the following:

§§111 and 112 of the 2004 Telecommunications Act (*Telekommunikationsgesetz* in German) [46] allow the government to force telecommunication service providers (which include cloud service providers like webmail) to hand over information such as a customer's name, address, birthdate, and email address, without a court order, through an automated query system that includes a search function in case law enforcement has incomplete request data. The admissibility of such a query is a decision of the requesting law enforcement authority.

New surveillance laws, such as the Federal Criminal Office Law (*BKA-Gesetz* in German) are explained in layman's terms on *Freedom instead of Fear*'s website (*Freiheit statt Angst* in German) [47]. This law gives the Federal Criminal Office new powers that are usually available only to state police and secret services.

An actual example of court-ordered surveillance in Germany is the Java Anonymous Proxy (JAP), which is an open source software for anonymously browsing websites. A court order obtained by the German Federal Office of Criminal Investigation ordered the JAP developers to add a backdoor to log accesses to an illegal website [2]. In this specific case, the open source nature of JAP allowed a user to discover the backdoor, showing that open source software cannot be modified to insert a backdoor without some technically advanced user noticing it in the source code [2].

4.2.3 Compelled certificate creation attack

This attack, explained in detail in Soghoian and Stamm's article *Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL* [49], is about government agencies forcing certificate authorities (CAs) to issue false TLS certificates to enable intelligence agencies to covertly watch an individual's secure Web communications. They state that all popular browsers and operating systems come pre-installed with more than 100 CA root certificates from entities all over the world, including government-owned CAs. They created a Firefox add-on called *CertLock* that caches certificates from sites that the user visits in order to detect suspicious changes, focusing on changes of the country of the CA that issued the certificate. However, the Firefox add-on has still not been released. An add-on with a more extensive approach than *CertLock* is *Certificate Patrol* [50], which warns the user every time a site sends a certificate different from the cached copy.

4.2.4 Compelled backdoor attack

As we said in section 4.2.1, government can force a cloud provider to insert a backdoor into its application, in order to bypass any encryption the service may provide.

One difficulty users face is that, unlike desktop applications that have a version number and do not automatically update themselves, cloud applications can change anytime without the user noticing [2]. A possible solution for this is Web application fingerprinting. Through the external analysis of the Web application, it may be possible to generate a fingerprint identifying a unique Web application and version pair. Kozina, Golub and Gros developed a fingerprinting method in [51], which compares link patterns, forms and keywords. However, the proposed method has currently some important limitations that prevent it from being ready for prime time.

5. CONCLUSION

It is clear that cloud computing is here to stay [1].

There are many challenges that we can only face if we understand what we are dealing with, how it may affect us and which possible solutions exist. In this article we have covered those points:

We found a concrete definition of the cloud computing concept, which is necessary if we ought to study it. We saw the advantages and disadvantages of the cloud for different actors: cloud service providers, businesses and individual consumers. We covered various security technologies that can solve many cloud computing security challenges. But we must convince cloud providers and users of the importance of implementing available security technologies. A small success in that aspect was achieved when Google was convinced to enable HTTPS by default in its Google Mail service. Unfortunately, the absence of economic incentives for providers to implement effective security measures means that this challenge has yet to be solved. The threat of certain security attacks serves to remind us that security technologies are not perfect.

We have also learned that privacy is more relevant than ever for users of cloud services, and deserves as much attention as security. Users must learn which privacy threats from hackers, data miners and government exist, so that they can take an informed decision when moving their private data to the cloud.

As cloud computing is still in its infancy, much remains yet to be seen. If we inform ourselves of the challenges and solutions that we face, we will be able to tackle them successfully.

6. REFERENCES

[1] *IT Cloud Services Forecast — 2008, 2012: A Key Driver of New Growth*, International Data Corporation, USA, 2008.
Available at `http://blogs.idc.com/ie/?p=224`

[2] C. Soghoian: *Caught in the Cloud: Privacy, Encryption, and Government Back Doors in the Web 2.0 Era*, In Journal on Telecommunications and High Technology Law, vol. 8, no. 2, Boulder, Colorado, USA, 2010.
Available at
`http://www.jthtl.org/articles.php?volume=8`

doi: 10.2313/NET-2011-05-2_05

[3] J. Voas, J. Zhang: *Cloud Computing: New Wine or Just a New Bottle?*, In IT Professional, vol. 11, no. 2, pp. 15–17, March–April, USA, 2009

[4] G. Fowler and B. Worthen: *The internet industry is on a cloud — whatever that may mean*, In Wall Street Journal, March 26, USA, 2009.
Available at `http://online.wsj.com/article/SB123802623665542725.html`

[5] P. Mell, T. Grance: *Effectively and Securely Using the Cloud Computing Paradigm*, version 26, National Institute of Standards and Technology (NIST), USA, 2009.
Available at `http://csrc.nist.gov/groups/SNS/cloud-computing/`

[6] *Facebook*, `http://www.facebook.com/`

[7] *Google Apps*, `http://www.google.com/apps/`

[8] *Google Mail*, `https://mail.google.com/`

[9] *Microsoft Windows Azure Platform*, `http://www.microsoft.com/windowsazure/`

[10] *Google App Engine*, `http://appengine.google.com/`

[11] *Amazon Elastic Compute Cloud (Amazon EC2)*, `http://aws.amazon.com/ec2/`

[12] *Rackspace Cloud*, `http://www.rackspacecloud.com/`

[13] *Microsoft Office Web Apps*, `http://office.microsoft.com/web-apps/`

[14] *Adobe Photoshop Express*, `http://www.photoshop.com/tools`

[15] M. Brandel: *Cloud computing: Don't get caught without an exit strategy*, In Computerworld, March 3, USA, 2009.
Available at `http://www.computerworld.com/s/article/9128665/`

[16] J. D. Lashar: *The Hidden Cost of SaaS*, In destinationCRM.com, May 1, USA, 2008.
Available at `http://www.destinationcrm.com/Articles/Columns-Departments/The-Tipping-Point/The-Hidden-Cost-of-SaaS---48682.aspx`

[17] *About Google Mail: More on Google Mail and privacy*, `https://mail.google.com/mail/help/about_privacy.html`

[18] *Google Gears*, `http://gears.google.com/`

[19] I. Fette: *Hello HTML5*, In Gears API Blog, February 19, USA, 2010.
Available at `http://gearsblog.blogspot.com/2010/02/hello-html5.html`

[20] *HTML5 Web Storage*, `http://www.w3.org/TR/webstorage/`

[21] *IDC Enterprise Panel*, n=244, August, 2008

[22] B. Grobauer, T. Walloschek, E. Stocker: *Understanding Cloud-Computing Vulnerabilities*, In Security & Privacy, IEEE, vol. PP, no. 99, p. 1, 2010

[23] *Yahoo! Mail*, `http://mail.yahoo.com/`

[24] E. Cole: *Network Security Bible 2nd Edition*, p. 130, Wiley, USA, 2009.

[25] Anonymous author: *My National Security Letter Gag Order*, In The Washington Post, March 23, USA, 2007.
Available at `http://www.washingtonpost.com/wp-dyn/content/article/2007/03/22/AR2007032201882.html`

[26] *How Hushmail Can Protect You*, `http://www.hushmail.com/about/technology/security/`

[27] R. Singel: *Encrypted E-Mail Company Hushmail Spills to Feds*, In Wired News — Threat Level, November 7, USA, 2007.
Available at `http://www.wired.com/threatlevel/2007/11/encrypted-e-mai/`

[28] *Mozilla Prism*, `http://prism.mozilla.com/`

[29] T. Dierks, E. Rescorla: *The Transport Layer Security (TLS) Protocol Version 1.2*, IETF RFC 5246, 2008.
Available at `http://tools.ietf.org/html/rfc5246`

[30] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF RFC 5280, 2008.
Available at `http://tools.ietf.org/html/rfc5280`

[31] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, IETF RFC 2560, 1999.
Available at `http://tools.ietf.org/html/rfc2560`

[32] A. Friedlander, A. Mankin, W. D. Maughan, S. D. Crocker: *DNSSEC: a protocol toward securing the internet infrastructure*, In Commun. ACM 50, 6 (June), 44–50, 2007.
Available at `http://doi.acm.org/10.1145/1247001.1247004`

[33] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose: *DNS Security Introduction and Requirements*, IETF RFC 4033, 2005.
Available at `http://tools.ietf.org/html/rfc4033`

[34] R. Arends, R. Austein, M. Larson, D. Massey, S. Rose: *Resource Records for the DNS Security Extensions*, IETF RFC 4034, 2005.
Available at `http://tools.ietf.org/html/rfc4034`

[35] B. Laurie, G. Sisson, R. Arends, D. Blacka: *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*, IETF RFC 5155, 2008.
Available at `http://tools.ietf.org/html/rfc5155`

[36] W. C. A. Wijngaards, B. J. Overeinder: *Securing DNS: Extending DNS Servers with a DNSSEC Validator*, In Security & Privacy, IEEE, vol. 7, no. 5, pp. 36–43, Sept.–Oct., 2009

[37] S. Josefsson: *Storing Certificates in the Domain Name System (DNS)*, IETF RFC 4398, 2006.
Available at `http://tools.ietf.org/html/rfc4398`

[38] M. Marlinspike: *New Tricks for Defeating SSL in Practice*, In Black Hat DC 2009, February, Washington DC, USA, 2009.
Available at `https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf`

[39] M. Marlinspike: *More Tricks for Defeating SSL in Practice*, In Black Hat USA 2009, July, Las Vegas, Nevada, USA, 2009.
Available at
https://www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatSSL-SLIDES.pdf

[40] M. Marlinspike: *Null Prefix Attacks Against SSL/TLS Certificates*, July 29, 2009.
Available at
http://www.thoughtcrime.org/papers/null-prefix-attacks.pdf

[41] M. Marlinspike: *Defeating OCSP With The Character '3'*, July 29, 2009.
Available at
http://www.thoughtcrime.org/papers/ocsp-attack.pdf

[42] K. Zetter: *Ex-Googler Allegedly Spied on User E-Mails, Chats*, In Wired News — Threat Level, September 15, USA, 2010.
Available at
http://www.wired.com/threatlevel/2010/09/google-spy/

[43] *Facebook Subpoena / Search Warrant Guidelines*,
http://cryptome.org/isp-spy/facebook-spy.pdf

[44] *Microsoft Online Services — Global Criminal Compliance Handbook (U.S. Domestic Version)*,
http://cryptome.org/isp-spy/microsoft-spy.zip

[45] *Interactive Data Protection Heat Map*,
http://www.forrester.com/cloudprivacyheatmap

[46] *Telekommunikationsgesetz vom 22. Juni 2004 (BGBl. I S. 1190)*, zuletzt geändert durch Artikel 2 des Gesetzes vom 17. Februar 2010 (BGBl. I S. 78).
Available at
http://www.gesetze-im-internet.de/tkg_2004/

[47] *BKA-Gesetz — Aktion Freiheit statt Angst e.V.*,
http://www.aktion-freiheitstattangst.org/de/themen/polizei-geheimdienste-a-militaer/97-bka-gesetz

[48] T. C. Greene: *Net anonymity service back-doored*, In The Register, August 21, UK, 2003.
Available at
http://www.theregister.co.uk/2003/08/21/net_anonymity_service_backdoored/

[49] C. Soghoian, S. Stamm: *Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL*, Under Submission.
Available at http://www.dubfire.net/#pubs

[50] *Certificate Patrol — a psyced Firefox/Mozilla add-on*,
http://patrol.psyced.org/

[51] M. Kozina, M. Golub, S. Gros: *A method for identifying Web applications*, In International Journal of Information Security, vol. 8, no. 6, pp. 455–467, Springer Berlin/Heidelberg, 2009. DOI: 10.1007/s10207-009-0092-3
Available at http://www.springerlink.com/content/wu204731658576t4/

Digital Signal Modulation Schemes

Philip Daubmeier
Betreuer: Stephan Günther
Seminar Innovative Internet-Technologien und Mobilkommunikation WS2010/11
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: daubmeie@in.tum.de

KURZFASSUNG

Diese Ausarbeitung beschäftigt sich mit den Grundlagen der digitalen Datenübertragung und gibt einen Überblick über existierende Modulationsschemata. Zu diesem Zweck wird der generelle Ablauf der Formung eines analogen Signals und dessen Modulation beschrieben sowie der Mechanismus zur Rückgewinnung der Daten auf der Empfängerseite. In diesem Zusammenhang werden einige einfache Modulationsarten aufgezeigt. Auf dieser Basis werden dann komplexere Modulationsverfahren wie QAM beschrieben, mit denen sich die Datenrate der übertragenen Informationen erhöhen lässt. Abschließend wird ein Bezug zu dem Verfahren des Frequenzmultiplexings hergestellt.

Schlüsselworte

Signal Modulation, ASK, PSK, QAM, FDMA

1. EINLEITUNG

In jüngster Zeit wurden immer neue Protokolle für digitale Datenübertragung entwickelt, die beispielsweise für Funkstrecken bei Mobiltelefonen oder lokalen Rechnernetzen über kurze Strecken mit WLAN genutzt werden können. Ganz ähnliche Verfahren werden jedoch auch für kabelgebundene Übertragungskanäle wie ADSL über Kupferleitungen oder die Übertragung über optische Medien wie Glasfaserleitungen benutzt. Dies sind nur einige Beispiele, wie Daten übertragen werden können. Allen gemeinsam ist die grundsätzliche Art und Weise, wie digitale Daten transferiert werden: Zur Übermittlung muss der Bitstrom erst in ein analoges Signal übersetzt werden, das dann auf Empfängerseite wieder als Folge digitaler Werte interpretiert werden kann.

Die Disziplin der Informatik beschäftigt sich im Gebiet der Rechnernetze vorwiegend mit den Schichten des ISO/OSI Modells [1], die oberhalb der physikalischen Übertragungsebene liegen. Die tatsächlichen Vorgänge des Sendens und Empfangens liegen im Aufgabenbereich der Elektrotechnik und bleiben der Informatik meist verborgen. Diese Ausarbeitung richtet sich damit an Personen aus der Informatik, die einen Einblick in die Welt der physikalischen Schicht bekommen möchten. Sie setzt dort an, wo die digitale Welt auf der Senderseite endet und zu übertragende Daten bereits alle oberen Schichten des ISO/OSI Stapelmodells durchlaufen haben, in Pakete gegliedert und mit Prüfsummen versehen wurden.

Diese unterste Schicht ist die Umgebung, die diese Arbeit betrachtet. In dieser Schicht gibt es keine Konzepte wie Pakete, Zieladressen oder Ähnliches. Dies gewährleistet die gewünschte Modularität oder Austauschbarkeit der darüber liegenden Schichten. Es werden nun einfach alle anstehenden Daten als Bitstrom betrachtet. Die Aufgabe des Senders besteht darin, diesen zeit- und wert-diskreten Strom als analoges Signal auf das Zielmedium zu legen. Beim Empfänger läuft dieser Vorgang ähnlich in umgekehrter Reihenfolge ab: Das ankommende analoge Signal muss dort wieder in einen Bitstrom gewandelt werden, der dann wieder von den höheren Schichten als konkrete Pakete und letztlich als Daten interpretiert wird. Der Vorgang, der hierbei auf der physikalischen Ebene stattfindet, wird im ersten Kapitel dieser Arbeit ausführlich erläutert. Dies bildet die Grundlage auf der verschiedene Modulationsarten erläutert und exemplarisch erklärt werden.

Im anschließenden Kapitel wird dann auf komplexere Modulationsverfahren eingegangen, die es unter Ausnutzung von mathematischen Gegebenheiten ermöglichen, eine höhere Datenrate zu erreichen. Dies wird erzielt, ohne mehr Ressourcen zu verbrauchen, wie zum Beispiel ein größeres Frequenzband einzunehmen, oder mehrere Kanäle parallel zu benutzen. Zusätzlich wird durch diese Techniken die Rekonstruierbarkeit des analogen Signals kaum beeinträchtigt. Sie ermöglichen es somit heutigen Implementierungen, eine schnelle Übertragung von Daten zu gewährleisten.

Im Anschluss wird das Verfahren des Frequenzmultiplexings erläutert, das einen engen Bezug zu den vorhergehend vorgestellten Modulationstechniken besitzt. Es wird kurz auf andere Multiplexing Techniken und Kombinationen dieser Verfahren eingegangen und gezeigt, wo diese eingesetzt werden.

Abschließend werden die Vor- und Nachteile der einzelnen Techniken herausgestellt und diskutiert, wo die Grenzen liegen und wo sich der aktuelle Stand der Technik befindet. Beispiele aus der aktuellen Zeit illustrieren die Sachverhalte, versuchen zum Verständnis beizutragen und einen Einblick zu geben, wie die vorgestellten theoretischen Grundlagen in der Praxis implementiert werden.

Diese Ausarbeitung orientiert sich in den grundlegenden Techniken am Vorlesungsskript Nachrichtentechnik 1 [10] des Lehrstuhls für Nachrichtentechnik der Technische Universität München.

doi: 10.2313/NET-2011-05-2_06

2. GRUNDLEGENDER MECHANISMUS

Um den Ablauf einer Übertragung auf der physikalischen Schicht zu gliedern, illustriert Abbildung 1 die einzelnen Schritte, die dabei durchlaufen werden. Die einzelnen Blöcke der Abbildung werden im Folgenden näher beleuchtet. In diesem Zusammenhang werden zwei grundlegende Modulationstechniken eingeführt, die auch in der Praxis Relevanz besitzen. Die Demodulation wird anhand dieser Techniken gezeigt.

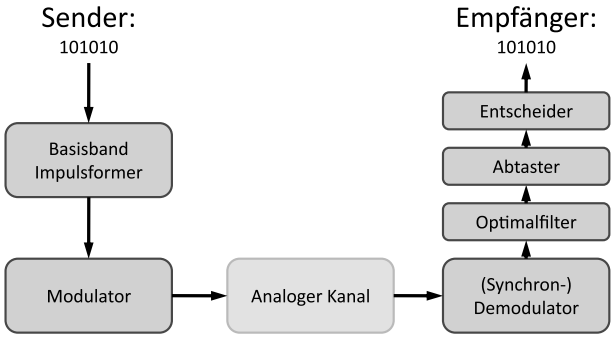


Abbildung 1: Ablauf einer Übertragung

2.1 Basisband Impulsformung

Bevor die eigentliche Modulation stattfinden kann, müssen die digitalen Daten erst in ein modulierbares analoges Signal gewandelt werden. Dies ist die Aufgabe des Basisband Impulsformers. Dazu wird der eingehende Bitstrom in Symbole zerteilt. Ein solches Symbol kann im einfachsten Fall ein Bit sein. Es können jedoch auch mehrere Bits zu einem Symbol zusammengefasst werden. Dies entspricht dann dem Verhalten eines Seriell/Parallel-Wandlers, der den seriellen Bitstrom in n parallele Bitströme zerteilt. Je ein Bit eines jeden solchen parallelen Bitstroms wird dann einem Symbol zugeordnet.

Der Basisband Impulsformer generiert dann aus den eingehenden Symbolen eine Impulsfolge, in dem er jedem Symbol einen Wert zuweist, der sich entweder auf die Amplitude, die Phase oder die Länge des generierten Impulses auswirken kann. In dieser Arbeit wird davon ausgegangen, dass jeder Impuls eine fest definierte Länge und Phase besitzt und die Werte der Symbole ausschließlich auf die Amplitude angewendet werden. Die Symboldauer wird im folgenden als T_s bezeichnet.

Das erzeugte Signal befindet sich im so genannten Basisband, von dem sich der volle Name des Impulsformers ableitet. Dieses Band befindet sich im Frequenzbereich um die Grundfrequenz der Symbole, das heißt $\frac{1}{T_s}$. Diese Frequenz bzw. das Intervall T_s ist zusammen mit der Anzahl der Symbole bestimmend für die Datenrate. Wird eine zu hohe Frequenz gewählt, können die Daten unter Umständen beim Empfänger nicht mehr rekonstruiert werden. Mit einer zu niedrigen Frequenz kann die optimal erzielbare Datenrate nicht erreicht werden.

Die Erzeugung des Datensignals $s(t)$ durch den Basisband Impulsformer besteht im ersten Schritt aus der Generierung eines kurzzeitigen Grundimpulses in jedem Zeitintervall T_s. Der Grundimpuls ist im idealisierten mathematischen Modell ein Dirac-Impuls [5]. Dieser ist eine unendlich dünne und unendlich hohe Kurve, dessen Integral genau 1 beträgt. Ein solcher Dirac-Impuls wird dargestellt als eine Funktion über der Zeit, die zu einem Zeitpunkt $t = 0$ ihren einzigen Wert ungleich Null annimmt, und ist somit definiert als:

$$\delta(t) = \begin{cases} +\infty, & t = 0 \\ 0, & t \neq 0 \end{cases}, \text{ wobei gilt: } \int_{-\infty}^{\infty} \delta(t)\, dt = 1. \quad (1)$$

Im mathematischen Modell wird angenommen, dass die Zeit nicht nach oben oder unten begrenzt ist. Dies bedeutet nun, dass für jedes $n \in \mathbb{N}$ ein solcher Impuls in jedem Zeitintervall T_s generiert wird. Das Signal setzt sich aus der Summe aller $\delta(t - nT_s)$ zusammen. In der Praxis können diese Dirac-Impulse zum Beispiel durch einen sehr kurzen Rechteckimpuls hinreichend genau für diese Anwendung angenähert werden. Diese Grundimpulse werden nun im zweiten Schritt mit dem Wert des anstehenden Symbols d_n multipliziert. Zwar ist diese Multiplikation streng genommen nicht möglich, da ∞ kein Funktionswert ist und das Integral in (1) auch nicht Riemann-integrierbar ist. Es ist allerdings dann unkritisch, wenn "$\delta(t)$ als Faktor vor stetigen Funktionen in bestimmten Integralen auftritt" [8, Seite 78], wie es im späteren Verlauf durch die Faltung der Fall ist. Das Signal $s'(t)$ kann durch folgende Beschreibung dargestellt werden:

$$s'(t) = \sum_{n=-\infty}^{\infty} d_n\, \delta(t - nT_s) \quad (2)$$

Im einfachsten Fall werden hierzu zwei Symbole, bzw. mögliche Symbolwerte, benutzt: Bei jeder binären Eins werden die Grundimpulse nicht verändert, was gleichbedeutend mit der Multiplikation mit 1 ist. Bei einer Null werden sie mit dem Symbolwert 0 multipliziert, was die Grundimpulse an diesen Stellen schlicht löscht. Es ist zu erwähnen, dass in diesem Fall die Symbolwerte rein willkürlich so gewählt sind, dass sie dem zu übertragenden Bit entsprechen und könnten genauso gut exakt anders herum festgelegt werden. Abbildung 2 stellt dieses Signal mit einem Ausschnitt aus einer exemplarischen Bitfolge (`...11010...`) dar. Die Dirac-Impulse werden, wie in der Literatur üblich, als Pfeile dargestellt, die so hoch gezeichnet werden, wie der Wert des Integrals unter ihnen.

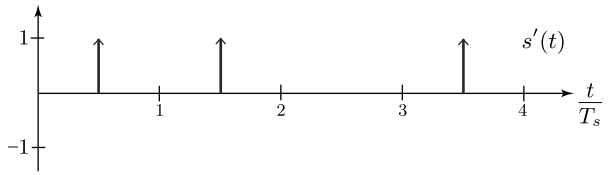


Abbildung 2: Signal $s'(t)$: Grundimpulse mit Symbolwerten Eins und Null

Die Impulse dieses Signals werden dann im letzten Schritt durch ein Filter in zeitlich breitere Ausformungen gestreckt. Ein solches Filter kann mit einer Funktion über die Zeit $g_s(t)$ charakterisiert werden. Eine Faltung dieser Funktion mit dem Signal $s'(t)$ ergibt das Ausgangssignal des Basisband Impulsformers, das heißt $s = s' * g_s$. Der zugrundeliegende Mechanismus der Faltung ist in [8, Seite 345] genauer beschrieben und soll hier nicht näher erklärt werden.

doi: 10.2313/NET-2011-05-2_06

Anschaulich beschrieben wird bei der Faltung ($*$-Operator) mit den Dirac Grundimpulsen der Graph der Funktion g_s überall dorthin projiziert, wo sich ein Dirac-Impuls befindet und auf die Höhe seines Wertes skaliert. Das resultierende Signal $s(t)$ besteht damit aus einer Überlagerung von Funktionen g_s, die je mit dem entsprechenden Symbolwert gestreckt oder gestaucht wurden, und kann daher wie folgt beschrieben werden:

$$s(t) = \sum_{n=-\infty}^{\infty} d_n \, g_s(t - nT_s) \tag{3}$$

Ein einfaches solches Filter kann zum Beispiel ein Rechteckfilter sein. Die Funktion $g_s(t)$, die dieses Filter charakterisiert, ist eine über die Zeit aufgetragene Rechteckfunktion. Diese wird genau so breit gewählt, wie das Intervall T_s zwischen den generierten Impulsen dauert. Dadurch entsteht ein zu einer Rechteckschwingung ähnliches Signal, in dem die einzelnen Rechteckimpulse je die Höhe des Symbolwerts besitzen und direkt aneinander angrenzen. Abbildung 3 illustriert dieses aus dem Bitstrom entstandene Signal.

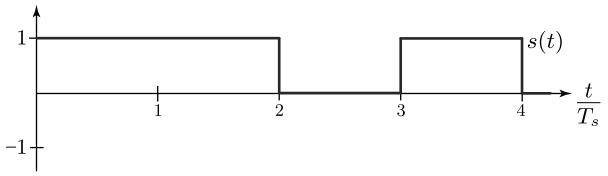


Abbildung 3: Auf $s'(t)$ angewendetes Rechteckfilter

Das Rechteckfilter ist sehr anschaulich und wird in dieser Ausarbeitung in exemplarischen Abbildungen verwendet. Jedoch kann dessen Ausgangssignal in der Praxis zu Problemen führen. Das Frequenzspektrum eines idealen Rechtecks im Zeitbereich ist unendlich. Selbst eine Approximation mit nur wenigen Gliedern belegt bereits ein relativ breites Band. Um diesem Problem entgegenzukommen, werden in tatsächlichen Implementierungen oft andere Filter verwendet, die keine Knicke, also Unstetigkeiten in ihrer Ableitung, besitzen und möglichst flach ansteigende und abfallende Flanken aufweisen. Die beiden bekanntesten, auch in existierenden Standards eingesetzten Filter, sind das (1.) Raised-Cosine-Filter und das (2.) Gauß-Filter. In Abbildung 4 sind die Impulsantworten eines Rechteckfilters und eines Gauß-Filters dargestellt. Dabei ist die Gauß-Glocke typischerweise breiter als T_s und dafür niedriger, damit das Integral unter der Glocke gleich der Fläche des Rechtecks ist. Impulsantworten ergeben sich durch die Faltung mit einem einzigen Dirac-Impuls zum Zeitpunkt 0, das heißt $\delta * g_s$.

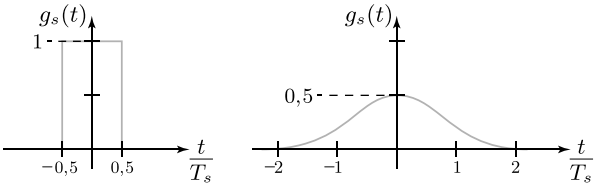


Abbildung 4: Impulsantworten eines Rechteck- und eines Gauß-Filters

Die resultierenden Signale eines Raised-Cosine- oder Gauß-Filters belegen ein wesentlich kleineres Frequenzband, das sich auch auf die Bandbreite das später modulierten Signals auswirkt. Dort resultieren harte Übergänge des von dem Basisband Impulsformer generierten Signals auch in vielen Oberschwingungen des modulierten Signals und damit in einem breiteren Frequenzband. In Abbildung 5 ist das durch ein Gauß-Filter geglättetes Signal mit den Symbolwerten 1 und -1 illustriert. Die grauen Kurven stellen die einzelnen Gauß-Glocken dar, die alle aufsummiert das resultierende Signal ergeben. Hier kann man wieder erkennen, dass eine einzelne Gauß-Glocke breiter als T_s ist und sich mit den angrenzenden überlappt. Werden diese zu schmal oder zu breit gewählt lassen sich die Daten nur schwer wieder rekonstruieren. Das richtige Maß ergibt sich aus einem Kompromiss zwischen möglichst kleiner Bandbreite des Signals und möglichst guter Rekonstruierbarkeit.

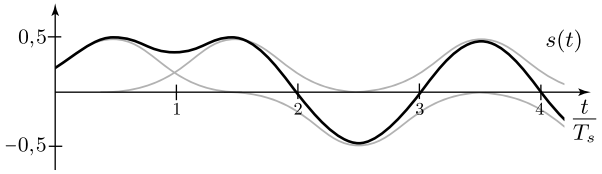


Abbildung 5: Auf $s'(t)$ angewendetes Gauß-Filter

Mit dem geglätteten Signal gehen jedoch auch Nachteile einher. Durch die überlappenden Teilimpulse wird das später modulierte Signal beim Empfänger schlechter rekonstruierbar. Die Dauer der Impulse muss hier gegebenenfalls länger gewählt werden, was in einer geringeren Datenrate für diesen Datenstrom resultiert. Der Fakt, dass das Signal nun aber ein sehr viel schmaleres Frequenzband belegt, wiegt dies allerdings wieder auf, wie ganz zuletzt im Abschnitt 4 – FDMA genauer beschrieben wird.

2.2 Modulation

Die Aufgabe des Modulators ist nun das Signal aus dem Basisband in ein gewünschtes Zielfrequenzband zu verschieben. Dazu wird das Signal aus Grundimpulsen auf ein sogenanntes Trägersignal aufmoduliert. Das Trägersignal ist im Normalfall eine reine Sinusschwingung mit genau der Frequenz innerhalb des Zielbandes. Diese Frequenz wird im Folgenden mit f_0 bezeichnet und muss signifikant höher sein als die Symbolfrequenz $\frac{1}{T_0}$. Der Mechanismus der Modulation selbst besteht im Wesentlichen daraus, das Trägersignal in einer oder mehreren der drei Freiheitsgrade Amplitude, Phase und Frequenz zu verändern. Diese Veränderung wird durch das aufzumodulierende Signal über die Zeit gesteuert. In den folgenden Abschnitten werden nur die beiden Modulationstechniken Amplitudenmodulation und Phasenmodulation betrachtet. Die drei Freiheitsgrade einer Schwingung werden bei der Betrachtung der folgenden Funktion sichtbar. Die Sinusfunktion lässt sich über die drei Parameter Amplitude (a), Frequenz (f) und Phase (p) steuern:

$$y(x) = a \, \sin(fx - p) \tag{4}$$

Weiterhin wird in dem Zusammenhang dieser Arbeit der Begriff Modulation synonym mit digitaler Modulation verwendet. Der feine begriffliche Unterschied der digitalen Modulation zur Modulation eines analogen Signals besteht nicht im Modulator selbst, sondern ist lediglich kontextbezogener

doi: 10.2313/NET-2011-05-2_06

Natur. Wenn digitale Daten vorher durch den Basisband Impulsformer analog gewandelt wurden, und beim Sender wiederum diskretisiert und als digitale Daten interpretiert werden, besitzt dieses System somit einen digitalen Modulator und Demodulator.

2.2.1 ASK - Amplitude Shift Keying

Die einfachste Modulationstechnik, das Amplitude Shift Keying (ASK) beruht auf Amplitudenmodulation. Bei einer solchen Modulation wird das Trägersignal so verändert, dass seine Hüllkurve (siehe gestrichelte Linie in Abbildung 6) dem aufmodulierten Signal entspricht. Dies wird erreicht in dem die Amplitude an jedem Zeitpunkt jeweils auf den Wert des Basisbandsignals gesetzt wird. Das entspricht genau einer Multiplikation des Trägersignals mit dem Basisbandsignal. Die Amplitude des Trägersignal wird mit s_0 bezeichnet, und variiert die maximale Auslenkung des Signals und somit die Sendestärke. Das modulierte Signal $\tilde{s}(t)$ ist somit beschrieben durch:

$$\tilde{s}(t) = s_0\, s(t)\, \sin(2\pi f_0 t) \tag{5}$$

Dies verändert weder die Phase noch die Frequenz des Trägers. Die Nulldurchgänge befinden sich immer noch an den selben Punkten. Mit genau der vorher vorgestellten Symbolwahl mit den Werten Null und Eins ergibt sich das so genannte On-Off-Keying (OOK). Das hierbei resultierende Signal wird in Abbildung 6 dargestellt. Hierbei wurde das Rechteck-gefilterte Basisbandsignal aus Abbildung 3 verwendet.

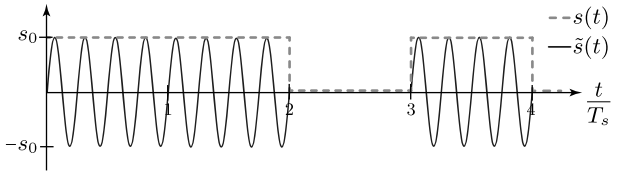


Abbildung 6: On-Off-Keying moduliertes Signal $\tilde{s}(t)$ und Basisband Signal $s(t)$ ($s(t)$ ist aus Gründen der Anschaulichkeit auf s_0 skaliert)

2.2.2 PSK - Phase Shift Keying

Beim Phase Shift Keying bleibt die Frequenz wiederum unangetastet. Statt der Amplitude wird nun jedoch nur die Phase korrespondierend zum Basisbandsignal verändert. Das Trägersignal wird an den Stellen, an denen das aufmodulierte Signal $s(t)$ seine volle Amplitude s_0 annimmt, nicht verändert und an den Stellen, an denen $s(t) = 0$ ist, um genau 180° phasenverschoben. Für den Fall bei dem das Signal $s(t)$ nicht durch einen Rechteckfilter entstand, sondern geglättet wurde, nimmt $s(t)$ beliebige Werte zwischen 0 und s_0 an. Hier wird das resultierende Signal nur zu dem entsprechenden Anteil verschoben. Bei $s(t) = \frac{s_0}{2}$ wäre dies zum Beispiel eine Phasenverschiebung von genau 90°.

Um dieses Verhalten zu erreichen, wird in folgender Formel der Faktor $s(t)$, der bei ASK vor dem Trägersignal stand, als Summand in die Sinusfunktion gezogen. Die Phasenverschiebung des Trägers hängt nun von dem Wert des Basisbandsignals $s(t)$ (durch $\frac{1}{s_0}$ normiert auf 1) zum Zeitpunkt t ab. Somit gilt folgende Gleichung für die vorherige Definition von PSK:

$$\tilde{s}(t) = s_0\, \sin\left(2\pi f_0 t - (1 - s(t))\, \pi\right) \tag{6}$$

Ein Sonderfall tritt ein, wenn ein mit PSK aufzumodulierendes Basisbandsignal nur genau zwei Symbolwerte aufweist. Für diesen kann vereinfachend ein Signal $s(t)$ benutzt werden, das nicht die Symbolwerte 1 und 0, sondern 1 und -1 annimmt. Dieses Binary PSK oder BPSK genannte Verfahren lässt sich dann durch ASK darstellen. Diese Technik, in der Literatur auch 2-ASK genannt, kann BPSK exakt nachstellen. Dies wird erreicht durch den Umstand, dass eine Multiplikation der Amplitude mit -1 eine Spiegelung an der Abszisse darstellt. Eine solche Spiegelung erzeugt genau die selbe Funktion wie eine Phasenverschiebung um 180°. Mathematisch betrachtet stellt dies die folgende Identität dar: $-\sin(x) = \sin(x - \pi)$. Abbildung 7 zeigt das mit 2-ASK modulierte Signal, dass genau einem BPSK Signal entspricht.

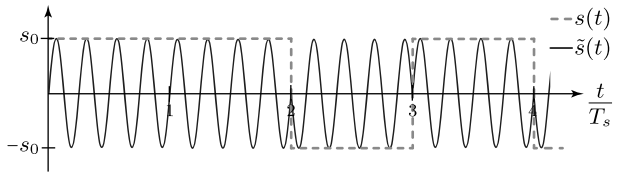


Abbildung 7: 2-ASK moduliertes Signal $\tilde{s}(t)$ und Basisband Signal $s(t)$ ($s(t)$ ist erneut auf s_0 skaliert)

2.3 Übertragungskanal

Nachdem das Signal im Basisband erzeugt und durch die Modulation auf ein Trägersignal in das Zielband verschoben wurde kann es nun auf das eigentliche Übertragungsmedium gelegt werden. Dies kann eine Übertragung per Funk sein, ein Kupferkabel oder ein Lichtwellenleiter. Jedes Medium hat eigene physikalische Eigenschaften und damit einhergehende Störeinflüsse.

Diese Störungen können kurze und unvorhersehbare Einflüsse sein, etwa ein anderes Gerät, das die Übertragung beeinflusst. Dies können aber auch natürliche Einflüsse sein, wie zum Beispiel ein Unwetter, das auf die Verbindung eines Satelliten zur Erde einwirkt. Auf diese Art von Störungen soll nicht näher eingegangen werden. Je nach Implementierung ist die Erkennung solcher Fehler entweder die Aufgabe der physikalischen Schicht beim Empfänger nach der Analog-Digital-Wandlung, oder der darüber liegenden Netzwerkschichten. Die Behandlung besteht je nach Größe des Störeinflusses und der verfügbaren Paritätsinformationen entweder in dem Versuch, die verlorenen Daten zu rekonstruieren, sie neu beim Sender anzufordern oder die Verbindung aufzugeben.

Viel typischer sind jedoch nicht Störungen, die unvorhersehbar sind, sondern vielmehr diese, die charakteristisch für das Medium sind und schlicht durch physikalische Grenzen entstehen. Dies ist ein Rauschen, das das eigentliche Nutzsignal überlagert und somit verfälscht. Um sich von diesem Grundrauschen abzuheben muss das Signal mit ausreichender Energie ausgesendet werden. Das Verhältnis dieses Störsignals zum Nutzsignal bezeichnet man auch als Signal-Rausch-Verhältnis. Es kann auf ein und dem selben Medium pro Frequenz variieren und ist maßgeblich dafür, welche

doi: 10.2313/NET-2011-05-2_06

Kapazität dieses Frequenzband theoretisch besitzt. Es legt damit die theoretische Obergrenze der Datenrate in einem Band fest [12, Seite 158 ff.].

Des weiteren ist ein realer Kanal auch immer in der Frequenzbandbreite begrenzt. Wenn dies nicht der Fall wäre, könnte eine theoretisch unendlich hohe Datenrate erreicht werden, selbst mit sehr schlechtem Signal-Rausch-Verhältnis über das ganze Spektrum hinweg, da man auf beliebig vielen Frequenzbändern gleichzeitig senden könnte (siehe auch Abschnitt 4). Es existiert immer eine Obergrenze im Frequenzbereich, ab der das Signal so stark gedämpft wird, dass in diesen Bändern nichts mehr übertragen werden kann. Dies hängt stark vom Medium ab, und der Länge des Kabels bzw. der Funkstrecke. Ein Kupferkabel besitzt hier typischerweise einen Tiefpass-Charakter, ist also hauptsächlich nur nach oben begrenzt. Funkübertragungen haben dagegen Bandpass-Charakter und sind damit zusätzlich nach unten begrenzt.

Das Shannon-Hartley-Gesetz [11, 6] gibt eine obere Schranke für die Menge der Daten die über einen Kanal mit gegebenem Signal-Rausch-Verhältnis und gegebener Bandbreite theoretisch übertragen werden können. Es gibt allerdings keinen Algorithmus, mit dem die geeignete Übertragungskodierung errechnet werden kann. Hier sind viele Tests und Erfahrung notwendig.

Im Folgenden wird ein idealer Übertragungskanal angenommen, um die Demodulation anschaulich zu beschreiben. Für das empfangene Signal $\tilde{r}(t)$ bedeutet dies: $\tilde{r}(t) = \tilde{s}(t)$. Das Signal wurde also keinem Rauschen ausgesetzt, nicht gedämpft und besitzt die Amplitude s_0.

2.4 Demodulation

Die Demodulation ist das Gegenstück zur Modulation und arbeitet ihr entgegengesetzt. Das bedeutet, dass das Signal im Frequenzbereich von seinem Frequenzband wieder in das Basisband zurückgeschoben werden muss. Idealerweise sollte exakt das Signal $s(t)$ wiederhergestellt werden. Allerdings ist $s(t)$ nicht unbedingt notwendig, um die Daten zu rekonstruieren. Es reicht bereits, ein Signal zu erhalten, das sich beim Abtasten ähnlich wie $s(t)$ verhält. So kann man etwa an einem mit On-Off-Keying, oder allgemein mit ASK, übertragenen Signal bereits an seiner Hüllkurve den Verlauf des Basisbandsignals erkennen. Hier kann man den Schritt der Demodulation gänzlich überspringen und sofort zur Detektion übergehen (siehe nächsten Abschnitt).

Bei einem mit PSK modulierten Signal gestaltet sich dies schon schwieriger. Dort ist die Hüllkurve konstant und auch die Frequenz, auf die man beim Abtasten Rückschlüsse ziehen könnte, verändert sich nie. Um Daten aus solch einem Signal zu rekonstruieren bedarf es zuerst einer Demodulation. Diese wird im Folgenden anhand der Binary-PSK erläutert.

Hierzu muss der Empfänger mit einem Oszillator eine Schwingung generieren, die exakt synchron mit dem empfangenen Signal $\tilde{r}(t)$ läuft. Die Frequenz f_0 ist hierbei bekannt, da sie durch das Protokoll des Übertragungsverfahrens festgelegt ist. Die Phase jedoch muss exakt mit der Grundphase von $\tilde{r}(t)$ übereinstimmen. Ist dies gewährleistet, kann dieses generierte Signal mit $\tilde{r}(t)$ multipliziert werden. Daraus entsteht das demodulierte Signal $b(t)$. Abbildung 8 zeigt diesen Schritt: Das $\tilde{r}(t)$ wird mit dem synchronen in Trägerfrequenz schwingenden Signal multipliziert und es entsteht $b(t)$. In der Abbildung wird ein Trägersignal mit einer Amplitude von 2 verwendet, um den Faktor $0,5$ wieder auszugleichen. Dieser Faktor entsteht durch die Multiplikation der beiden Signale, da folgender trigonometrischer Zusammenhang gilt: $\cos^2(a) = 0,5\ [1 + \cos(2a)]$. In der Abbildung sieht man deutlich, wie die Form der Hüllkurve von $b(t)$ nun exakt der des 2-ASK Basisband Signals aus Abbildung 7 entspricht. Der Grund hierfür ist, dass bei der Multiplikation der frequenz- und phasenrichtigen Anteile je Wellenberg mit Wellenberg und Wellental mit Wellental multipliziert werden. Es entsteht ein $\sin^2(t)$-förmiges Signal, das komplett oberhalb der Abszisse liegt. In den Abschnitten in denen $\tilde{r}(t)$ jedoch um 180° phasenverschoben ist, also das zweite Symbol kodiert ist, wird je ein Wellental mit einem Wellenberg multipliziert und eine $-\sin^2(t)$-förmige Komponente wird erzeugt. Diese liegt dann unterhalb der Abszisse, und hebt sich in der Hüllkurve deutlich von phasenrichtigen Anteilen ab. Dieses so entstandene Signal kann nun ebenso wie OOK modulierte Signale in einem nächsten Schritt detektiert werden.

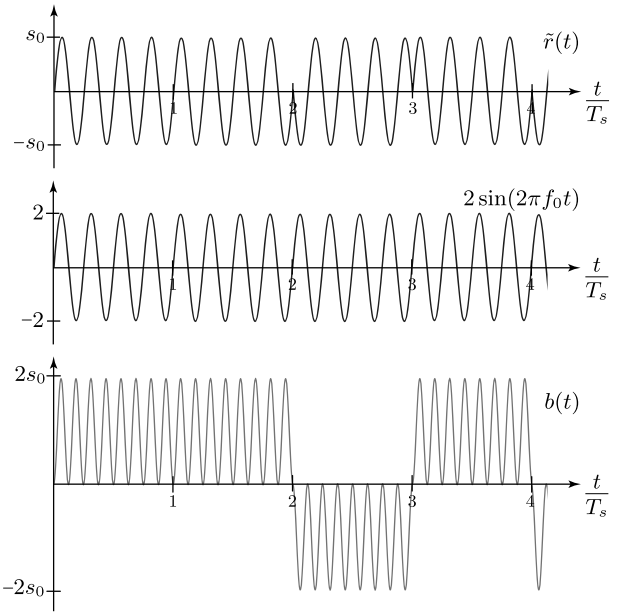


Abbildung 8: Demodulation des empfangenen Signals $\tilde{r}(t)$

Um die Synchronisierung zu gewährleisten gibt es verschiedene Verfahren. Eine Möglichkeit besteht darin, ein Frequenzband fest zu reservieren, in dem ein so genannter Pilotton gesendet wird. Bei einem Nulldurchgang des Pilotton Signals wird der Oszillator synchronisiert. Weiterhin gibt es die Möglichkeit eine so genannte Costa-Schleife [4] einzusetzen. Dies ist eine Regelschleife, die eine Taktabweichung feststellen und diese rückführen kann, um den Oszillator nachzuregeln.

doi: 10.2313/NET-2011-05-2_06

2.5 Detektion

Die Detektion eines demodulierten Signals setzt sich aus drei Teilschritten zusammen: (1.) einem Optimalfilter, (2.) einem Abtaster und (3.) einem Entscheider.

Das Optimalfilter, im englischen Matched-Filter genannt, ist ein Filter, das entsprechend des Frequenzbandes und des Modulationsverfahrens möglichst gut an das übertragene Signal angepasst ist. Es unterdrückt das Rauschen im übertragenen Signal und maximiert damit das Signal-Rausch-Verhältnis. Das Optimalfilter geht auf North [9] zurück, der dieses bereits 1943 untersuchte. Für ein ideal übertragenes Signal ist solch ein Filter nicht notwendig, daher soll seine Funktionsweise in dieser Arbeit nicht genauer erläutert werden.

Nach dieser Aufbereitung, die im Falle eines theoretisch idealen Übertragungskanals nicht nötig ist, kann das Signal ausgewertet werden. Dazu wird es durch einen Abtaster über die Zeit diskretisiert. Dieser liest den aktuellen Wert des Signals in einem regelmäßigen Intervall von Zeitpunkten aus. Dieses Intervall muss mindestens so klein gewählt werden wie $\frac{T_s}{2}$. Häufig wird es jedoch wesentlich öfter, also in kleineren Intervallen abgetastet, und ein Durchschnittswert gebildet der dann diesen Symbolwert repräsentiert.

Da digitale Daten nicht nur Zeit- sondern auch Werte-diskret sind, folgt der letzte Schritt der Detektion: Der Entscheider weist jedem dieser kontinuierlichen Werte nun einen diskreten Symbolwert zu. Damit sind die digital übertragenen Daten wieder vollständig rekonstruiert. Jeder Symbolwert kann nun wiederum auf ein oder mehrere Bits abgebildet werden. Dies ist genau die Rückrichtung der Abbildung der Bits auf einen Symbolwert. Diese können zeitlich nacheinander zu einem Bitstrom zusammengefasst werden, der im Falle einer fehlerlosen Übertragung genau dem gesendeten Bitstrom entspricht.

3. KOMPLEXERE MODULATIONSARTEN

Die zwei bisher vorgestellten grundlegenden Modulationsarten ASK und PSK verwenden jeweils nur einen Parameter des zweidimensionalen Symbolraums. Dies ist der Raum in dem einer Bitfolge ein Symbolwert zugewiesen wird. Graphisch lässt sich dies veranschaulichen, in dem man für jedes Symbol in der komplexen Ebene Punkte aufträgt. Die Länge $|\vec{v}|$ des Vektors $\vec{v}$ vom Ursprung zu einem solchen Punkt stellt dann die aufzumodulierende Amplitude dar und der Winkel φ die Phase. Dieser Phasenwinkel zeigt von der Abszisse aus im Gegenuhrzeigersinn auf den Vektor. Im Zusammenhang eines solchen Symbolraums bezeichnet man die Realkomponente auch als Inphase-Anteil und die Imaginärkomponente als Quadratur-Anteil.

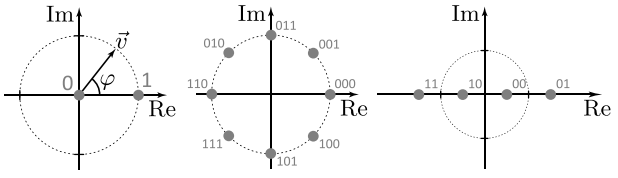


Abbildung 9: Symbolräume des OOK, 8-PSK und 4-ASK

In der linken Grafik der Abbildung 9 ist dies zusammen mit den beiden Symbolen des On-Off-Keyings dargestellt. Die Beschriftung der Punkte steht hier für die Bitfolge, die diesem Symbol zugeordnet ist. Da bei PSK nur die Phase, also der Winkel φ, verändert wird und nicht die Amplitude, liegen die Symbole immer auf einem Kreis um den Ursprung. Da alle Symbole die gleiche Priorität beim Senden besitzen werden sie gleichmäßig auf diesem Kreis verteilt. Um nun zum Beispiel 3 Bit gleichzeitig in einem Symbol zu kodieren werden somit 8 Symbole verwendet, wie in der mittleren Grafik bei dem Symbolraum des 8-PSK zu sehen ist. Bei ASK ist dies ganz ähnlich, jedoch liegen hier alle Symbole auf der Realachse und sind somit immer in Phase oder genau 180° phasenverschoben. Hier wird nur die Amplitude verändert um die Symbole voneinander zu unterscheiden. Die rechte Grafik zeigt dieses am Beispiel des 4-ASK mit 4 Symbolwerten. Die Verwendung beider Parameter gleichzeitig, um den Symbolraum besser auszunutzen, bildet die Basis der im Folgenden vorgestellten Modulationsart.

3.1 QAM

Bei der Quadratur-Amplituden-Modulation wird der gesamte zweidimensionale Symbolraum ausgenutzt, indem man zwei getrennte Signale für je Inphase- und Quadratur-Anteil moduliert und diese anschließend kombiniert. Dies soll im Folgenden anhand der einfachsten Form der QAM gezeigt werden, der so genannten 4-QAM, mit der je 2 Bit pro Symbol übertragen werden können.

Zu diesem Zweck wird der serielle Bitstrom durch einen Seriell/Parallel-Wandler in Blöcke von je 2 Bits geteilt. Diese werden dann dem korrespondierendem Symbol zugeteilt, wie auch bereits bei ASK und PSK beschrieben wurde. Ein solches Symbol entspricht nun aber nicht länger nur einem Symbolwert, sondern zwei Werten: Dem Inphase-Wert d_{In} und dem Quadratur-Wert d_{Qn}. Der Symbolraum der 4-QAM kann somit als eine parallele Kombination von 2-ASK und einer um 90° phasenverschobenen 2-ASK gesehen werden. Abbildung 10 illustriert diesen Zusammenhang mit den Symbolräumen dieser zwei Modulationsarten und dem kombinierten Symbolraum.

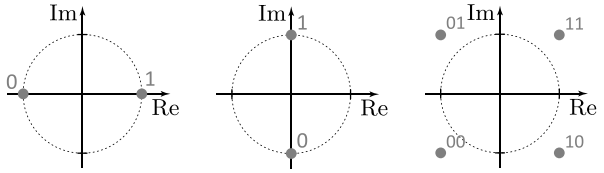


Abbildung 10: Symbolräume des 2-ASK, des 90° phasenverschobenen 2-ASK und deren Kombination zu 4-QAM.

Man kann am Symbolraum des 4-QAM eine Analogie zum 4-PSK sehen. Beim 4-PSK liegen die Symbole auf den vier Schnittpunkten des Einheitskreises mit den beiden Achsen. Wenn man die Vektoren des Symbolraums also um $\sqrt{2}$ streckt und um 45° rotiert, erhält man genau den Symbolraum des 4-QAM. Es wird später gezeigt, dass ein solch gestrecktes und phasenverschobenes 4-PSK Signal genau dem 4-QAM Signal entspricht. Dies ist ein Sonderfall, der nur bei 4-QAM funktioniert, nicht bei QAM mit mehr als 4 Symbolen.

doi: 10.2313/NET-2011-05-2_06

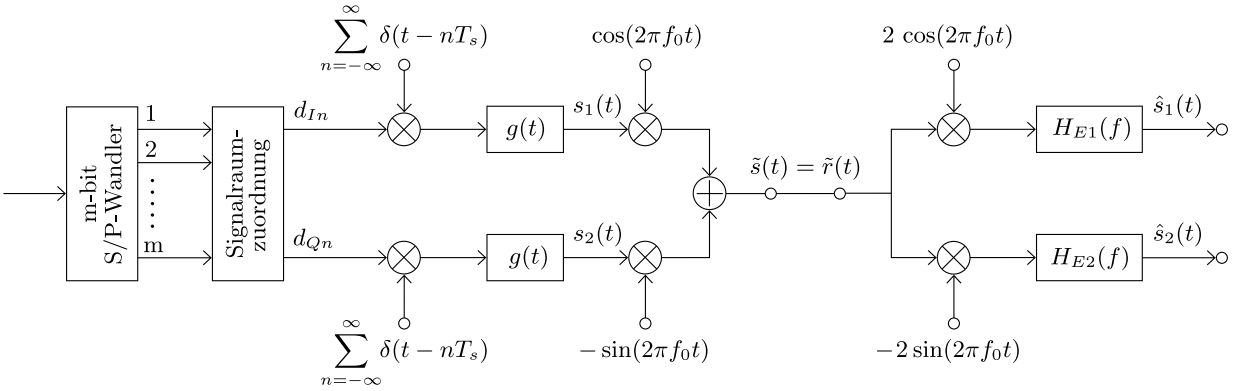


Abbildung 11: Ablauf einer QAM Modulation und Demodulation (nach [10])

Abbildung 11 beschreibt nun detailliert den Ablauf einer m-QAM Modulation. Dabei werden m verschiedene Symbolwerte verwendet. Die S/P-Wandlung sowie die Symbolraumzuordnung mit ihren 2 Ausgängen wurde bereits beschrieben. Die eigentliche Technik der QAM besteht nun darin, diese beiden Symbolwerte getrennt von einander weiter zu verarbeiten. Es werden hierfür sowohl der Inphase- und der Quadratur-Wert auf einen Grundimpuls multipliziert und mit $g(t)$ gefiltert um je ein Basisband Signal zu erhalten. Die Grundimpulse werden in der Abbildung durch die Summe der Dirac-Impulse dargestellt, die mit den Werten d_{In} und d_{Qn} multipliziert werden. Das Basisband Signal für die Folge von Inphase-Werten wird im Folgenden als $s_1(t)$, das für die Quadratur-Werte als $s_2(t)$ bezeichnet. $s_1(t)$ wird nun auf ein Kosinus-Trägersignal moduliert, $s_2(t)$ auf ein Minus-Sinus förmiges, also einen genau 90° phasenverschobenen Träger. Diese beiden modulierten Signale $\tilde{s}_1(t)$ und $\tilde{s}_2(t)$ werden nun addiert und ergeben damit das zu übertragende Signal $\tilde{s}(t)$.

$$\begin{aligned}\tilde{s}(t) &= \tilde{s}_1(t) + \tilde{s}_2(t) \\ &= s_1(t)\cos(2\pi f_0 t) - s_2(t)\sin(2\pi f_0 t)\end{aligned}$$

Nach der Übertragung über einen idealen Kanal erreicht den Sender das Signal $\tilde{r}(t) = \tilde{s}(t)$. Dieses wird ganz ähnlich zur Demodulation eines 2-ASK Signals mit einem synchronen Trägersignal multipliziert. Im Falle von QAM wird $\tilde{r}(t)$ allerdings zwei mal parallel demoduliert. Einmal mit einem Kosinus-synchronem Signal und einmal mit einem Minus-Sinus-synchronem Signal, und damit exakt den beiden Trägersignalen, die zur Modulation verwendet wurden. Dabei werden, wie schon bei in Abbildung 8 gezeigt, je die Wertanteile nach oben und unten geklappt, die zuvor mit dem entsprechenden Träger moduliert wurden. Wenn beide demodulierten Signale noch durch die Optimalfilter $H_{E1}(f)$ und $H_{E2}(f)$ gegeben werden, besitzen die Signale $\hat{s}_1(t)$ und $\hat{s}_2(t)$ genau die Hüllkurven, die der Form von $s_1(t)$ und $s_2(t)$ entsprechen. Nach Abtastung und Entscheidung ergeben sich wieder die Inphase- und Quadratur-Werte d_{In} und d_{Qn}. Je ein solches Wertepaar kann wieder einem Symbol und damit einer Bitfolge zugeordnet werden. Es ergeben sich pro Symbol wieder die m Bit-Werte, die zu diesem Zeitpunkt gesendet wurden und nun seriell auf einen Bitstrom gelegt werden können.

Mit dem folgenden, aus Formelsammlungen bekanntem trigonometrischen Zusammenhang

$$a\sin(x) + b\cos(x) = \sqrt{a^2 + b^2}\,\sin\left(x + \arctan\left(\tfrac{b}{a}\right)\right)$$

lässt sich $\tilde{s}(t)$ abschnittsweise in die Form eines Sinusoid bringen (Beweis: [3]). Für den speziellen Fall des 4-QAM, in dem $a, b \in \{-1, 1\}$ gilt, können somit die folgenden Fälle auftreten:

$$\begin{aligned}&\pm\cos(2\pi f_0 t) \pm \sin(2\pi f_0 t) \\ &= \pm\sqrt{2}\,\sin\left(2\pi f_0 t \pm \frac{\pi}{4}\right)\end{aligned}$$

Hierbei wurde der Zusammenhang $\arctan(\pm 1) = \pm\frac{\pi}{4}$ benutzt. Man sieht in der resultierenden Form, dass sich die Frequenz durch die Addition der beiden Sinus- und Kosinusfunktionen nicht verändert. Die Amplitude wird gegenüber den beiden Teilfunktionen jedoch um den Faktor $\sqrt{2}$ gestreckt. Durch die beiden ±-Variationen, sowohl in der ursprünglichen als auch der Sinusoid-Form, ergeben sich 4 Möglichkeiten. Bei genauerer Betrachtung sieht man, dass dies die Phasenverschiebung je nach kodiertem Symbol um 45°, 135°, 225° oder 315° darstellt. Dies entspricht dem vorher vorgestellten Symbolraum, und deckt sich damit exakt mit der Analogie zum gestreckten und phasenverschobenen 4-PSK.

Im Folgenden wird beschrieben, wie sich mathematisch zeigen lässt, dass sich sowohl der Inphase- als auch der Quadratur-Anteil getrennt von einander rekonstruieren lassen. Das mit dem Kosinus demodulierte Signal ergibt zusammen mit den beiden folgenden trigonometrischen Zusammenhängen

$$\sin(x)\ \cos(x) = \frac{1}{2}\sin(2x) \tag{7}$$

$$\cos^2(x) = \frac{1}{2}\Big(1 + \cos(2x)\Big) \tag{8}$$

die folgende Formel:

$$\begin{aligned}\hat{s}_1(t) &= 2\cos(2\pi f_0 t)\,\tilde{s}(t) \\ &= 2\cos(2\pi f_0 t)\left[A\,\cos\left(2\pi f_0 t\right) - B\,\sin\left(2\pi f_0 t\right)\right] \\ &= 2A\cos^2(2\pi f_0 t) - 2B\cos(2\pi f_0 t)\sin(2\pi f_0 t) \\ &= A + A\cos(4\pi f_0 t) - B\sin(4\pi f_0 t)\end{aligned}$$

doi: 10.2313/NET-2011-05-2_06

Damit wird das Signal immer um die Konstante A in der Ordinate versetzt. In dieser Konstante steckt damit die Information des Inphase-Anteils des Symbolwerts. Analog gilt dies für das demodulierte Signal des Quadratur-Anteils.

Dass das Inphase Signal sich nicht mit dem Quadratur Signal überschneidet, und sich damit gegenseitig beeinflusst, ist der Tatsache zu verdanken, dass die beiden Trägersignale Kosinus und Minus-Sinus orthogonal zu einander sind. Was so eben durch die Berechnung gezeigt wurde, verdeutlicht Abbildung 12 graphisch: Die Minima und Maxima der beiden Funktionen, dort wo die Information über die Amplitude steckt, liegen jeweils über den Nulldurchgängen der anderen Funktion. Damit besitzt die Funktion nach dem Aufsummieren an diesen Stellen immer noch genau diese Werte. Damit interferieren sie nicht, und ermöglichen damit erst die Quadratur-Amplituden-Modulation.

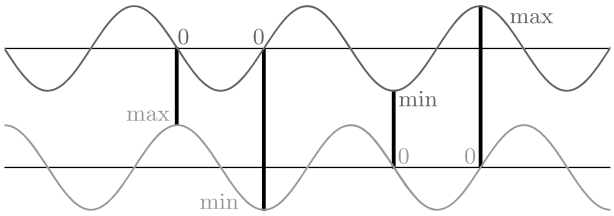


Abbildung 12: Orthogonalität der Sinus und Kosinus Funktionen

Das Grundprinzip der QAM, das gerade anhand der 4-QAM gezeigt wurde, kann mit der Tatsache, dass sowohl im Inphase- als auch im Quadraturanteil mehr als nur ein Amplitudenwert benutzt werden kann, verbunden werden. Damit können sogar noch mehr Bit pro Symbol auf der Trägerfrequenz untergebracht werden. In der Spezifikation des DVB-C2 wird sogar eine 4096-QAM vorgeschlagen, die 2^{12} Symbole besitzt [7]. Sie kann 12 Bit pro Symbol und damit ganze 3 Byte mit nur zwei Symbolen übertragen.

4. FREQUENZMULTIPLEXING

Als Multiplexing, oder Multiple Access, bezeichnet man die Technik, mehrere separate Datenströme auf einem Medium zu transportieren. Dies ist insbesondere für die drei folgenden Anwendungsfälle interessant. Zum einen, wenn mehrere Benutzer ohne sich gegenseitig zu stören, über ein Medium miteinander kommunizieren wollen. Zum anderen ist es oft erwünscht, ein Medium für mehrere separate Datentransfer Protokolle zu nutzen. Dies ist beispielsweise bei der Funkübertragung der Fall, bei der Mobilfunk, terrestrisches Fernsehen, Radio und viele Andere parallel senden und empfangen können ohne sich dabei gegenseitig zu stören. Es wird jedoch auch eingesetzt um die Datenrate für eine einzige Verbindung zu erhöhen, in dem ein Datenstrom in mehrere Teilströme aufgeteilt wird, diese über den Kanal gesendet werden, um dann beim Empfänger wieder in einen Strom zusammengefasst zu werden.

Die naheliegendste Form ist das Multiplexing über die Zeit, auch TDMA (Time Division Multiple Access) genannt, bei der jeder Sender zeitlich nacheinander auf dem Medium sendet. Es gibt diverse Techniken, wie dieses implementiert werden kann und wie Kollisionen erkannt oder vermieden werden können. Auf diese soll hier nicht näher eingegangen werden. Diese wird, neben anderen Techniken, oft für den Anwendungsfall eingesetzt, bei dem mehrere Benutzer verschiedene Verbindungen über ein Medium benutzen wollen.

Um jedoch zeitgleich mehrere Übertragungsströme auf einem einzigen Medium zu realisieren, ist TDMA wenig geeignet. Die verschiedenen Datenströme müssten eine gemeinsame TDMA Technik implementieren, was alleine schon dem Gedanken widerspricht, verschiedene Protokolle einsetzen zu können. Für diesen Zweck wird eine andere Technik eingesetzt, die durch die Modulation erst möglich gemacht wird: FDMA (Frequency Division Multiple Access). Dabei wird der Umstand ausgenutzt, dass das Nutzsignal durch die Modulation auf eine fast beliebige Frequenz gebracht werden kann. Weiterhin belegt das Signal idealerweise ein möglichst kleines Frequenzband, was durch die vorher vorgestellten Raised-Cosine-, Gauß- oder andere Filter sichergestellt werden kann. Durch diese Voraussetzungen können nun mehrere solcher modulierter Datenströme parallel auf ein Medium gelegt werden, wenn sie auf verschiedene Frequenzen moduliert wurden, und ihre Frequenzbänder sich nicht überschneiden. Auf Empfängerseite können ein oder mehrere Frequenzbänder von Interesse durch einen Bandpass wieder herausgeschnitten und die enthaltenen Signale demoduliert werden. Dadurch entsteht, im Gegensatz zu anderen Multiplexing-Techniken, kein höherer Aufwand auf Sender- oder Empfängerseite: Der Sender muss zum Beispiel keine Strategien zur Kollisionsvermeidung einsetzen. Der Empfänger muss nicht den ganzen Datenverkehr mithören und Anhand diesem entscheiden, ob Daten an ihn adressiert ist.

Abbildung 13 zeigt einen Frequenzbereich einer ISDN-Telefonleitung auf der mit ADSL Daten übertragen werden. FDMA macht es hier möglich gleichzeitig über das ISDN Frequenzband zu telefonieren und sich über ADSL ins Internet zu verbinden. Die beiden Frequenzbänder werden im Haushalt über einen ADSL-Splitter mit zwei Bandpässen getrennt und dem ADSL-Modem bzw. der Telefonanlage zugeführt. Das ADSL Band ist wiederum in zwei Zuständigkeitsbereiche für den Up- bzw. Downstream geteilt, um parallel Daten senden und empfangen zu können.

Das elektromagnetische Spektrum, das bei der Übertragung über Funk zur Verfügung steht, ist vielfach von verschiedensten Analogen und Digitalen Übertragungsprotokollen belegt. Lokale Rechnernetze mit WLAN liegen im Band von ca. 5,7-5,9 GHz. GSM Mobilfunk belegt Frequenzen um 900 MHz und 1800 MHz, UMTS für schnellen Internetzugang von Mobiltelefonen den Bereich von etwa 1,9-2,2 GHz. Digitales Satellitenfernsehen wird im Bereich zwischen 10,7 GHz - 12,7 GHz ausgestrahlt [2]. Dies ist jedoch nur eine sehr kleine Auswahl von Protokollen. Ohne FDMA wäre es praktisch nicht denkbar so viele Verfahren gleichzeitig einsetzen zu können.

FDMA kann auch sehr gut für den dritten Anwendungsfall eingesetzt werden: Die Erhöhung der Datenrate durch Multiplexing von mehreren Teilströmen. Diverse Übertragungsprotokolle nutzen diese Technik um ihre Datenrate zu erreichen. Dazu werden mehrere, meist benachbarte Frequenzbänder genutzt und zu einer Übertragung zusammengefasst. Je nach Signal-Rausch-Verhältnis wählen viele mo-

doi: 10.2313/NET-2011-05-2_06

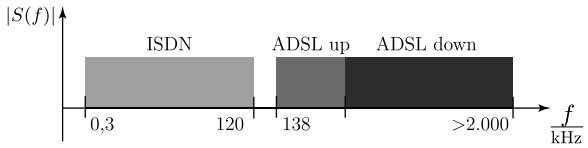


Abbildung 13: Frequenzbereich einer ISDN-Leitung mit ADSL

derne Übertragungsverfahren dazu pro Teilfrequenzband adaptiv die geeignete Symbolanzahl für die Modulation aus. Dies ist im Idealfall die größte noch detektierbare Anzahl für die genutzte Modulationstechnik. Je nach Qualität und Länge der Leitung oder Funkstrecke können bestimmte Frequenzbänder eine unterschiedliche Übertragungsgüte aufweisen. Auf schlechteren Frequenzbändern werden daher nur weniger gültige Werte pro Symbol genutzt, auf besseren entsprechend mehr. Die Summe der Datenraten der einzelnen genutzten Frequenzbänder ergibt dann die Gesamtdatenrate der Verbindung.

Viele Multiplexing-Verfahren lassen sich zusammen mit FDMA bzw. darauf aufsetzend nutzen. So benutzt etwa GSM Mobilfunk ein anderes Multiplexing Verfahren um Kollisionen von Verbindungen mehrerer Nutzer auf dem GSM-Frequenzband zu erkennen und zu vermeiden. Eben dieses Band befindet sich aber wiederum zusammen mit anderen Protokollen über FDMA auf dem selben Übertragungskanal. So entsteht eine Hierarchie, in dem das Übertragungsmedium zwischen den verschiedenen Protokollen, Nutzern, Verbindungen und einzelnen Datenströmen immer weiter unterteilt wird.

5. AUSBLICK

Nachdem nun die wichtigsten digitalen Modulationsverfahren beschrieben wurden, stellt sich die Frage, wie diese in der Praxis eingesetzt werden. Hierzu gibt es für jede Implementierung dieser Techniken viele Parameter, die zu klären sind. Das Frequenzband wird bei kabelgebundener Übertragung meist durch die physikalischen Eigenschaften bestimmt. Je länger das Kabel sein darf, desto mehr leidet hier das Signal-Rausch-Verhältnis und die Bandbreite und damit die erreichbare Datenrate. Bei Funkübertragungen spielt hierbei noch zusätzlich eine Rolle, welche Frequenzbänder frei verfügbar sind und können sich für jedes Land stark unterscheiden.

Zusätzliche Faktoren sind, welche Modulationsart und Symbolraum eingesetzt wird, wie viele Bit pro Symbol kodiert werden, welche Filter zur Grundimpuls-Generierung verwendet werden, mit welcher Sendeleistung gesendet wird, und viele andere mehr. Jeder dieser Punkte geht mit weiteren Entscheidungen einher, wie zum Beispiel die Auswahl von geeigneten Parametern, mit denen die Grundimpuls-Filter modifiziert werden. Viele Techniken können die Symbolrate, die Anzahl der Bits pro Symbol und die Sendeleistung adaptiv an die Qualität des Mediums anpassen. Wie und wann solche Anpassungen genau geschehen muss ebenfalls durch einen geeigneten Algorithmus festgelegt werden. Mögliche Anforderungen tragen zusätzlich zur Komplexität bei. So kann es etwa für einen ADSL Anbieter wünschenswert sein, die Datenrate eines Kunden bei einem günstigen Vertrag auf ein bestimmtes Maximum zu drosseln. Solche Fälle müssen im Protokoll vorgesehen werden, und beim Verbindungsaufbau eines ADSL-Modems mit der Gegenstelle ausgehandelt werden.

Man sieht, dass sich eine konkrete Implementierung einer digitalen Datenübertragung weitaus komplexer gestaltet als in der Theorie, die in dieser Arbeit vorgestellt wurde. Hierfür ist viel Entwicklungsarbeit, viele Tests und Erfahrung notwendig. Neben diesen Problemen gibt es ein noch weitaus grundlegenderes: Die informationstheoretische maximale Datenrate, die durch das vorher erwähnte Shannon-Hartley-Gesetz [11, 6] festgelegt wird. Es besagt nichts anderes, als dass die Datenrate, die immer weiter anwächst, durch das Medium grundsätzlich begrenzt ist. Ist diese Grenze erreicht, gibt es, um die Datenrate weiter zu erhöhen, keine andere Möglichkeit als mehrere Leitungen zu bündeln, oder auf eine physikalisch günstigere Leitung zu wechseln. Bei Funkübertragungen etwa besteht dann nur die Möglichkeit, die einzelnen Funkzellen zu verkleinern und damit weniger Benutzer pro Zelle bedienen zu müssen.

6. LITERATUR

[1] Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model. *International Standard ISO/IEC 7498-1*, November 1994.

[2] Frequenznutzungsplan. *Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen*, April 2008.

[3] G. Cazelais. Linear Combination of Sine and Cosine. Februar 2007.

[4] J. P. Costas. Synchronous Communications. *Proceedings of the IRE*, pages 1713–1716, Dezember 1956.

[5] P. Dirac. *Principles of quantum mechanics (4th ed.).* Oxford at the Clarendon Press, 1958.

[6] R. V. L. Hartley. *Transmission of Information.* Bell System Technical Journal, 1928.

[7] P. Hasse, D. Jaeger, and J. Robert. DVB-C2 - A new Transmission system for Hybrid Fibre Coax Networks. *ICT Mobile Summit Conference Proceedings, Santander*, Juni 2009.

[8] K. Meyberg and P. Vachenauer. *Höhere Mathematik 2: Differentialgleichungen, Funktionentheorie, Fourier-Analysis, Variationsrechnung.* Springer-Lehrbuch Series. Springer, 2001.

[9] D. O. North. Analysis of the Factors which determine Signal/Noise Discrimination in Radar. *IEEE, RCA Laboratories, Princeton, Technical Report, PTR-6C*, 51, Juni 1963.

[10] G. Söder and C. Hausl. Vorlesungsskript Nachrichtentechnik 1 Sommersemester 2009. *Lehrstuhl für Nachrichtentechnik, Technische Universität München.*

[11] C. E. Shannon. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[12] M. Werner. *Nachrichtentechnik - Eine Einführung für alle Studiengänge - 6. Auflage.* Vieweg + Teubner.

doi: 10.2313/NET-2011-05-2_06

MGRP - passive aggressive measurements

Ferdinand Mayet
Supervisors: Dipl.-Ing. Dirk Haage, Dipl.-Inf. Johann Schlamp
Advanced Seminar - Innovative Internettechnologien und Mobilkommunikation WS2010/2011
Chair for Network Architectures and Services
Department of Informatics, Technische Universität München
Email: mayet@in.tum.de

ABSTRACT

This work presents the Measurement Manager Protocol (MG RP), an in-kernel service supporting flexible, efficient and accurate measurements. MGRP schedules probe transmissions on behalf of active measurement tools and reduces the monitoring overhead by reusing application traffic. A small benchmark experiment demonstrates the potential of this passive aggressive measurement before an evaluation is carried out. In this context, another sophisticated approach, namely TCP Sidecar, is presented and compared with MGRP and other traditional methods. At the end, some analysis about the usage and application of both concepts are discussed.

Keywords

active, passive, aggressive, network, traffic, measurements, MGRP, TCP Sidecar

1. INTRODUCTION

The Internet evolved in the last thirty years from text based utilities to a platform used for multimedia streaming, online conferencing and other services of the World Wide Web. The majority of the users grasp the Internet as a medium which provides the connectivity between their applications and distributed information and data. The end-users do not need to know any background of how the Internet works, such as the processes that are triggered after the user clicks on a hyperlink or how packages are routed on their way through the network [1].

However, network researchers aim to understand the networks infrastructure and the protocols used to communicate with other instances of the network. A major methodology researchers use to collect and analyse information about networks are end-to-end measurements. Due to measurements, interesting network properties could be estimated which help to improve applications and protocols in order to gain a good user experience. A good user experience could for example be reached by selecting the nearest and fastest server to download from. A different application might need a low round-trip-time (RTT) and a high path capacity. Therefore network applications need to discover the current network conditions and adapt accordingly. Since there is no possibility to gather information about the state of the network by asking other network devices, traffic analysis has to be carried out between the endpoints.

The research area of network traffic measurements also aims at evaluating a given network in order to be able to understand its topology and to identify the available bandwidth between different hosts. This topic is steadily gaining popularity since video streaming becomes more and more important to individuals (e.g. watching videos on YouTube) as well as to companies using the Internet as a online meeting platform. Detailed information about the network would enable applications or even protocols to change their behaviour and adapt to the networks state. For example, if a video is hosted on multiple servers the application could choose the best connection between client and server. This might be the nearest server but it could also be the case that this specific one is too busy to satisfy the users requirements. In order to achieve a higher user experience the application should be able to discover such shortages and choose an appropriate way to solve them. Furthermore, measurement techniques are used to reveal security issues like firewall misconfigurations or to locate problems occurring during communication.

In the following, a short introduction to the topic of network measurements will be outlined and a variety of traditional measurement approaches will be discussed. Thereafter, two sophisticated techniques will be presented and analysed. At the same time, their specific advantages and drawbacks will be elaborated. Finally, some related work are presented.

2. BACKGROUND

Network measurements are applied whenever information about a network and its current state is necessary. Therefore, four main reasons for network measurements will be presented and their benefits will be explained:

Network Troubleshooting The purpose concerning traffic measurements in the area of network troubleshooting is to discover defective hardware and misconfigurations of endpoints and intermediate devices. For example, the Internet Control Message Protocol (ICMP) can be used to send messages to a desired endpoint. If these messages do not arrive at the endpoint an error message is triggered which indicates that something is wrong in the network. More precise evaluations in combination with other protocols can then be used to identify defects or misconfigurations in the network [1].

Protocol Debugging Protocol Debugging is necessary if new protocols are developed. Thereby, measurement techniques ensure the standard compliance of a protocol by for example analysing the traffic. Furthermore,

doi: 10.2313/NET-2011-05-2_07

it is possible to prove the backward compatibility of a newer protocol version to its predecessor. In order to prove backward compatibility a variety of approaches are available such as establishing a communication between two endpoints with different protocol versions and examining the transferred messages [1].

Workload Characterisation Another area where network measurements are applied is the field of workload characterisation. This domain analyses the exchanged traffic between endpoints and creates a ranking of the protocols which transfer data. On the basis of this ranking applications can be optimised for the most frequently exchanged data. This is very important to multipurpose applications which use several protocols to communicate with other hosts. Workload characterisation also focuses on the protocol layer and supports the improvement of newer protocol versions with regards to the monitored workload [1].

Performance Evaluation Another important usage of measurements is performance evaluation. Network traffic measurements are utilized to determine the performance of the network. A detailed analysis may help to identify performance bottlenecks. Once these problems are identified the results might be used to further improve protocols or the network infrastucture itself. Performance evaluation is often used in combination with the workload characterisation process described earlier [1].

These four reasons are just a few examples of motivations for monitoring network traffic. Detailed information about protocols and processes of a network are very important to achieve a higher usability and to enhance the users experience. Hence, network measurements must be performed. However, the crucial part is to pay attention to the realization of such measurements because they should be transparent to the user and should not change the network. In the following, a short introduction to traditional methods for monitoring the network is given.

3. METHODOLOGY

Software network measurement tools can be classified in two major monitoring concepts, passive and active. These two concepts can again be subdivided in offline and online measurements. In this case online describes a technique where packets are analysed on the fly whereas offline denotes a mechanism that first captures information and evaluates the data afterwards. A common representative of offline monitoring are log files or dump files, for example created by *tcpdump*[1].

3.1 Passive Measurements

Passive measurement describes a mechanism which collects the observed traffic of the network. The term "passive" states that no additional workload is introduced into the network and only available traffic is captured and analysed. In order to obtain information about a given link, such as time dependent references, the observation of the traffic has to be applied at different network locations (see Figure 1) [2].

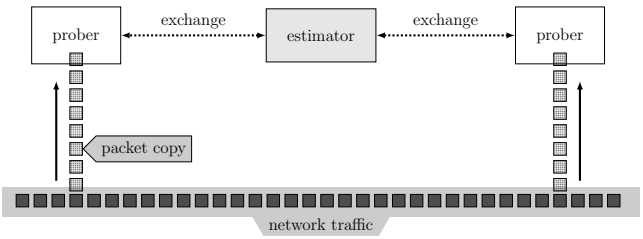


Figure 1: Concept of passive measurements

Claise ([3]) categorises passive measurements into two groups:

Full collection This process captures every single packet which passes the metering point. The main advantage of a full collection is accuracy as the collected data is exactly equal to the passed traffic. However, a drawback is the large number of packets that have to be stored and analysed, which may require very fast metering hard- and software.

Partial collection Most of the time it is not possible to perform a full collection due to high speed interface technologies which send a huge amount of data in a very short time period. Therefore a partial collection process which filters or samples the collected data is necessary. For example, filtering mechanisms may select a specific flow of data (e.g. TCP traffic) to reduce the workload of the monitoring unit. In contrast, sampling uses statistical methods (e.g selecting 1 of N packets) in order to reduce the load of the measurement systems.

Passive measurements are applied if the exact network state is important and interference with live traffic is not wanted. However, the disadvantage of monitoring is that desired traffic types may not be present in the traffic passing the observation point. This purpose could be solved using active measurement techniques.

3.2 Active Measurements

In contrast to passive measurement, active measurements require explicit requests that generate synthetic traffic with a desired type and workload [2]. Active measurements involve two systems into the process, a sender and a receiver. The sender creates the desired traffic and sends it to the receiver which collects all packets at their arrival and evaluates each (see Figure 2).

[1]tcpdump: a Linux tool which dumps traffic on a network `http://www.tcpdump.org/`

doi: 10.2313/NET-2011-05-2_07

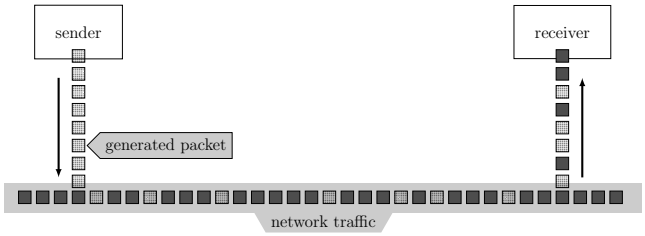


Figure 2: Concept of active measurements

Performance evaluations, network troubleshooting and protocol debugging is mostly done using active measurements since it is possible to generate an extremly high workload, malformed packets and special traffic. This might be necessary if special information about the network must be collected. Compared to passive techniques active probing has the advantages of maximum traffic control, independency of the current traffic and the ability to detect the maximum available bandwidth. Furthermore, active measurements are easier to implement than passive ones. Nevertheless, synthetic traffic may cause collsions in the network und thus change the network's behaviour. Hence, active measurements are commonly an estimation of the real network state and throughput [3].

To conclude, both techniques have multiple advantages but also several disadvantages. Obviously, new mechanisms are necessary to minimise the newly introduced traffic on the one hand and to keep the control of the transferred traffic as high as possible on the other hand. In the following sections two more sophisticated measurements techniques are presented which try to counter the observed problems and combine all advantages.

4. MGRP - PASSIVE AGGRESSIVE MEASUREMENTS

This section introduces the Measurement Manager Protocol (MGRP) which addresses the shortcomings of traditional approaches by using a *hybrid* concept. MGRP is an in-kernel service that enables probes to reuse application traffic transparently and systematically. As described in section 3 passive probing is efficient but unable to detect improvements of network conditions and active probing affects the current traffic on the link. MGRP permits the user to write measurement algorithms as if they are *active* but be implemented as if they are *passive*. Hence MGRP can be more aggressive without harming the performance of an application [4].

MGRP piggybacks application data into probes in order to minimise newly introduced traffic. Piggybacking is a process that is aware of probes which mostly consist of empty padding. Empty padding is necessary because probes have to reflect the behaviour of real application traffic that carries useful payload. The piggybacking mechanism replaces the empty padding of a single probe with payload that should be transmitted to the receiver and thus prevents the prober from sending unnecessary packets.

4.1 MGRP architecture

MRGP is a kernel-level service which extends the transport layer of the ISO/OSI model. This protocol is basically accessed using two application programming interfaces (APIs), the probe and payload API. The payload API extracts useful data from other transport protocols like TCP or UDP and hands it to the MGRP service. In collaboration with the probe API MGRP generates a *hybrid* of a probe and application data. Instead of sending single probes directly to the receiver, the sender uses the probe API to specify an entire *train*. A *train* is defined by the size of each probe, the number of probes, the amount of padding and the gap between probes [5].

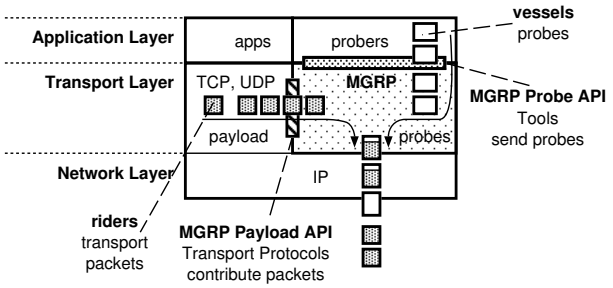


Figure 3: MGRP architecture [4]

Once defined a *train* MGRP starts piggybacking application traffic and sends these merged packets to the receiver. By filling most of the empty padding with payload MRGP nearly behaves like a passive algorithm since it omits the overhead generated by active measurements. Figure 4 indicates the transition from active mechanisms with probes and empty padding (white/black checkerboard) to a MRGP like traffic with no padding. Figure 4 also shows the newly introduced MGRP header (illustrated as small light gray box).

Figure 4: Transition from active measurement traffic (left) to MGRP traffic (right)

At the receiver side the payload is separated from the measurement data. The payload output is handed over to the standard transport layer and the measurement data is transferred to the monitoring system. The receiver side adds a second timestamp to the MGRP header and delivers the packet to the prober. The MGRP header mainly consists of two timestamp header fields, one timestamp is entered by the sender when the packet is sent. The other one is entered by the receiver and contains the reception time [4].

4.2 Probe Transaction by Example

The example described in this section is illustrated in Figure 5 and will be walked through from step ① to ⑧. Consider the following case: The sender is streaming multimedia data to a destination D and at the same time MGRP is used

doi: 10.2313/NET-2011-05-2_07

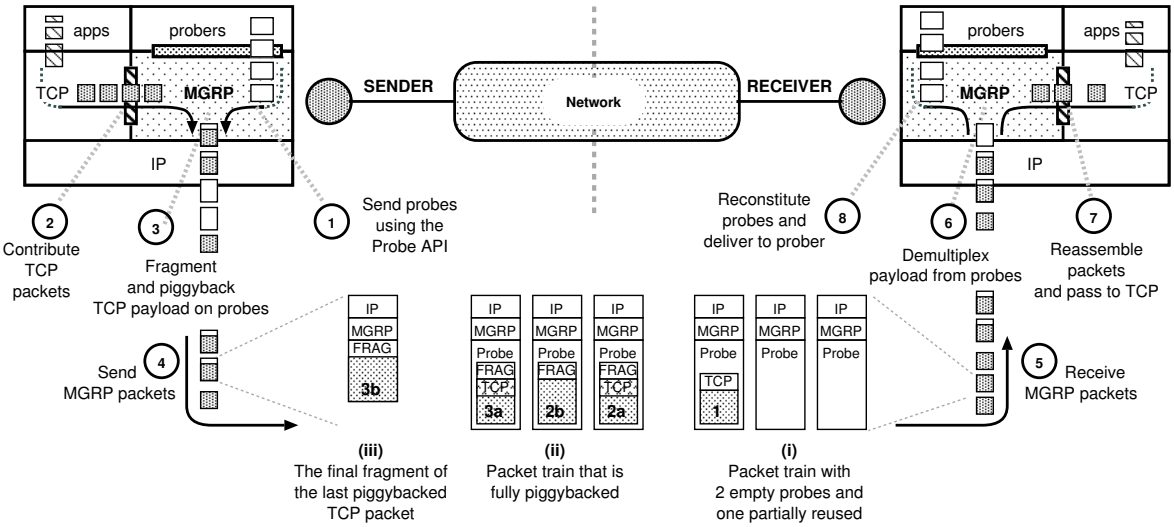


Figure 5: Demonstration of MGRP operation [5].

to measure the network condition between the sender and D.

At step ① the prober calls the `sendmsg` function to send the first probe. The first `sendmsg` call also defines several options and the ancillary data. For example, the gap between probes and the barrier flag is set. The barrier flag is used to indicate that a whole *train* of probes should be created. As long as the barrier flag is set to 1 all probes are buffered until the flag is unset. Afterwards the probes are sent as a *train*. Packets generated by the streaming application with same destination D as the probes are collected by the payload API ②. At step ③ the TCP packets are fragmented and piggybacked into the probe. In some cases fragmentation might be necessary as the payload could exceed the MGRP payload size. Due to additional header information of MGRP the payload size is smaller then the one of e.g TCP. Afterwards MGRP sets the kernel timestamp in the MGRP header field and hands it to the IP layer for transmission ④. During transmission three different kinds of packets may occur: As illustrated in (i) of Figure 5 MGRP provides the possibility to completely disable piggybacking or only partially. In the partially disabled case, MGRP sends out probes with empty padding if no suitable rider was found. If it is completely disabled MGRP behaves like a traditional active measurement tool. The ideal case of piggybacking is shown in (ii) where all probes are reused to transport application data. The label 2a and 2b indicate that the original TCP packets had to be fragmented to fit into the probes data field. If MGRP was unable to piggyback the payload before the buffer timeout exceeds additional MGRP packets containing the remaining chunks are sent (iii). The buffer timeout determines the available time for buffering application data until it must be transmitted by MGRP. As the packets arrive at destination D ⑤ the payload is demultiplexed ⑥ from the measurement data. Next, the original TCP packets are reassembled and delivered to the application ⑦. At the same time, MGRP reconstructs the probing packets by zeroing the padding and setting the reception time ⑧. Finally, MGRP buffers the probes for delivery to the prober [5].

4.3 Experiments with MGRP

This section elaborates on an experimental setup in order to demonstrate the behaviour and performance of MGRP. The network topology is given in Figure 6.

In the experiment a constant 4 Mbps stream is transmitted from $m3$ to $m5$ representing a multimedia stream hosted on $m3$ and requested by $m5$. The data rate of 4 Mbps was chosen because it is commonly used for high definition multimedia streams [4].

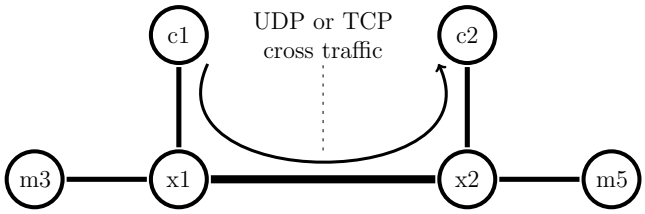


Figure 6: Experimental setup (based on [5])

While streaming from $m3$ to $m5$ MGRP is used to determine the actual bandwidth between those two nodes. In order to obtain more realistic results, the link $x1x2$ is throttled to a maximum data rate of 10 Mbps as there are only 4 participants in the network.

As shown in Figure 7 the multimedia stream in disturbed by UDP cross traffic which is transmitted from $c1$ to $c2$. The cross traffic is stepwise increased until it reaches the maximum spare throughput of 6 Mbps and is decreased afterwards. Each interval lasts 45 seconds and transmits constant

doi: 10.2313/NET-2011-05-2_07

rates of 1, 3, 5, 6, 4 and 2 Mbps. The most interesting interval is between 135 and 180 seconds because at this point in time no additional traffic like measurement probes are able to pass from on side to the other without harming one of the two streams on link $x1x2$ [4].

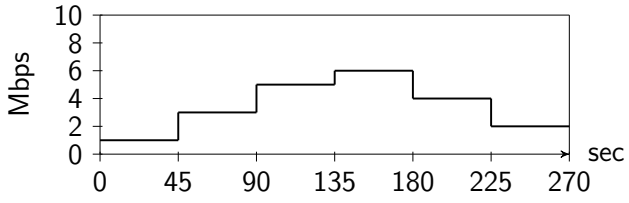


Figure 7: Cross traffic from $c1$ to $c2$ (based on [5])

Figure 8 presents the results of the experiment described before. Two cases are considered in the plots:

1. active probing with piggybacking disabled (upper plot)

2. reuse of application payload as a rider using MGRP (lower plot)

Case 1: After about 70 seconds the two streams and the additionally introduced payload of the packets start to interfere as there is not enough bandwidth available to satisfy all network participants. Unfortunately, UDP has no congestion control and keeps sending as much cross traffic as possible. However, the data flow between $m3$ and $m5$ uses TCP and recognises that the link is overwhelmed which forces TCP to enter the congestion avoidance phase. The algorithm of this phase decreases the maximum segment size (MSS) (e.g. $MSS = \frac{MSS}{2}$) and therefore limits the transmission rate to 3 Mbps respectively to 2 Mbps later on [6]. If both hosts would have used UDP as transport protocol a dramatic packet loss would have occurred. Figure 8 also indicates that the monitoring traffic consumes approximately 2 Mbps. To conclude, the user viewing the multimedia data on location $m5$ will experience stuttering or in the case that the hosting application is aware of the congestion the stream quality will be downgraded.

Case 2: In this case MGRP utilises the application traffic as riders and nearly all probes carry application data. Only approximately 0.2 Mbps are used to send probes without piggybacked payload. Hence, the TCP connection experiences only little congestion and the stream nearly stays at 4 Mbps. To sum up, MGRP reduces the measurement overhead to a minimum and lowers interference with other network communications while monitoring constantly. The viewer of the video stream might only suffer small or even no changes.

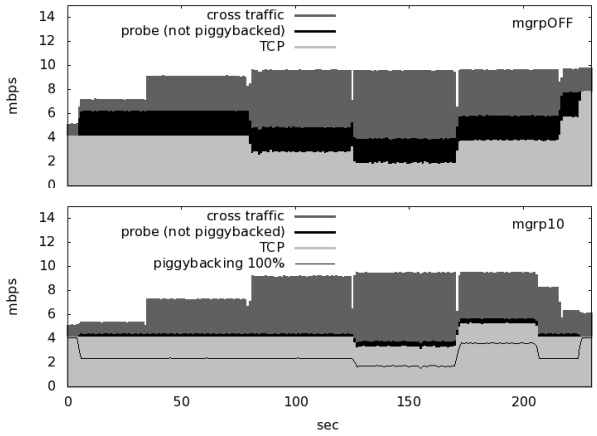


Figure 8: Results of the experiment [5]

4.4 Evaluation

This section provides a overview on MGRP and discusses problems that may occur using MGRP for measurement purposes. As described earlier, MGRP facilitates the reuse of existing traffic. Hence, the networks condition is just slightly modified and the amount of newly introduced collisions is minimal. Furthermore, MGRP is traffic independent since it switches to an active measurement like mode if no riders are available. Due to the fact that the number of probes can be specified, MGRP is able to determine the maximal available bandwidth.

But there are several reasons why MGRP is not used every and all the time: First of all, MGRP inserts delay into the network as application traffic is buffered and multiplexed at the senders side and demultiplexed at the receivers side. Secondly, MGRP changes the behaviour of all TCP connections on the link as collisions may appear more likely. Since TCP implements a congestion avoidance algorithm multiple TCP connections influence each other and try to share the maximum available bandwidth equally. In a scenario where MGRP is used to measure the network additional overhead is added to a single connection which leads to a worsening of all other connection sharing the same link. Even if these modifications are small an excessive usage of MGRP on multiple connections might yield a large overhead which restricts the performance of the network in contrast to passive measurements. Furthermore, sophisticated delay calculation have to be done by the prober since there are several timeouts and buffering/fragmentation delays have to be considered. Additionally, measurement packets are not able to traverse firewalls and NATs (network address translations) which also prevents MGRP from being used universally. This problem occurs as firewalls and NATs do not understand the MGRP header format. However, the most important drawback relates to the implementation of MGRP. As described in section 4 MGRP is an in-kernel service. This indicates that the network stack has to be modified in order to add MGRP to the transport layer. Currently no operating systems integrates MGRP by default. Hence, the monitoring systems stack has to be changed which might not be feasible in many cases. The fact of the matter is that this prevents MGRP from being used by peers globally. Streaming or peer-to-peer applications like Skype can not use it as

doi: 10.2313/NET-2011-05-2_07

well, since there is no guarantee of having MGRP available on all participatory systems.

The following section will present another measurement technique. This approach is based on the traditional active measurement process and extends it in several ways.

5. TCP SIDECAR

TCP Sidecar is a monitoring platform for injecting packets into a network and follows the principle that the network provides enough bandwidth to handle additional traffic caused by probes. The main goal of TCP Sidecar is to circumvent intrusion detection systems (IDSs) and firewalls since synthetic traffic is most often considered being extraordinary and potentially malicious. Hence, most firewalls will block measurement traffic and IDSs will trigger alerts and abuse reports. Therefore, carefully designed measurement probes and responses have to be generated. Thereby, TCP Sidecar does not restrict the source and destination nor the time of measurement since the platform does not want to force any extraordinary behaviour [7].

5.1 Architecture and Probe Transaction

The concept behind TCP Sidecar is to generate probes consisting of replayed data segments. Therefore Sidecar uses passive measurements to collect traffic that is passing by and retransmits it to destination. Figure 9 outlines the standard procedure of a Sidecar measurement. The prober (Sidecar) can be positioned freely in the network. Often it is placed at the senders side but it is possible to place it at every other node in the network as long as both the forward and reverse path are observed by Sidecar [8].

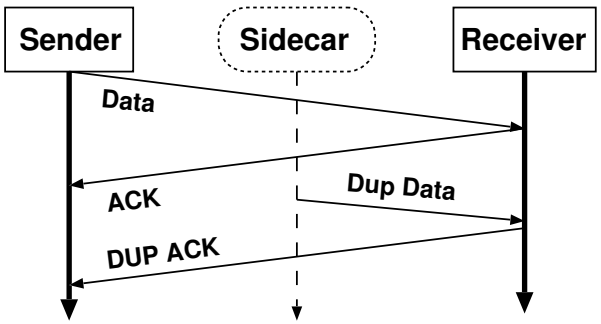


Figure 9: TCP Sidecar procedure [7]

TCP Sidecar is also used to modify the captured data before replaying it. This has to be tampered very accurately as firewalls and IDSs should not notice any difference between real and replayed packets.

For example, the time-to-live (TTL) option of the TCP header can be adjusted freely because the receiver does not necessarily need the information contained in the replayed packets. Hence, these packets can be dropped before they reach the receiver. This modification might be used inside TCP Sidecar to detect NATs. The detection mechanism simply uses ICMP and varies the TTL. If a TTL exceeded message is returned by a network node and the source address of this error message is the same as the original destination address of the packet, a NAT can be assumed [8].

Once a duplicate packet arrives at the receiver no warning or even error message is generated since TCP considers the reception of duplicate packets. Shortly after the reception the receiver generates a duplicate acknowledgement (ACK) and sends it to the source address. This packet is then again captured by TCP Sidecar for measurement purposes [8].

5.2 Evaluation

TCP Sidecar is a platform for unobtrusive measurements and enables measurements throughout firewalls, and NATs. Furthermore, Sidecar is able to detect NATs and perform several kinds of measurements without alerting IDSs. Especially large network service like PlanetLab [8], CoDeeN [9], OpenDHT [10], Meridian [11], and CoralCDN [12] might benefit from TCP Sidecar since most of these services already perform network measurements and might struggle with altering IDSs.

However, Sidecar has several drawbacks: Firstly, the platform depends on existing traffic which might be applicable to large networks but may be a problem in smaller networks. Since no measurement will be forced (only the amount of replayed data can be set) the results in smaller network may not be as accurate as with using other active measurement tools. A second problem might be the placement of the monitoring system as both communication channels must pass the metering point. The most significant problem is the fact that duplicate ACKs are generated. A duplicate ACK can be regarded as network problems by the sender. Therefore TCP adjusts the congestion window size which is a state variable that limits the maximal amount of unacknowledged TCP packets. For example, if the congestion window size is 2, TCP can only send two TCP packets with an outstanding acknowledgment. Each duplicate ACK decrements the congestion window by one. This might get even worse if a third duplicate ACK reaches the sender because in this case TCP enters the slow start phase and halfs the maximum segment size (MSS). Hence, the data rate is reduced and the communication is violated by measurements. Nevertheless, this problem might be solved by selectively grabing duplicate ACKs and discarding them if they are not important to the sender. Regarding to Figure 9 the Sidecar node must be able to not only generate duplicate data but also to analyze duplicate ACKs on their way back to the sender. If the the duplicate ACK is considered being non-essential to the sender (e.g a duplicate ACK with the same sequence number has already been transferred to the sender) the packet must be dropped by the Sidecar node. However, the classification of these packets into categories like important to the sender or not might quite challenging but would potentially increase the performance of TCP Sidecar.

6. RELATED WORK

MGRP is similar to many approaches (Periscope [13], Scriptroute [14], pktd [15]) in that it serves the possibility to define measurement probes and schedules. MGRP differs from these approaches as it is the first tool, which is fully integrated on layer 4 in the IP protocol stack. Thereby, MGRP reduces the measurement overhead by reusing probes as rid-

doi: 10.2313/NET-2011-05-2_07

ers for application data. Furthermore, MGRP is a protocol and not a standalone application. Thus, it can be integrated into a big amount of existing applications and help to improve their performance by reducing unnecessary overhead.

The following collection of related work should give a short overview of similar project:

- Sidecar has the advantage that it support ICMP messages which enable NAT detection. But as a consequence of the problems described in 5.2 the measurement intervals have to be kept low which makes the usage of Sidecar difficult.
- MAD [16] is a Multi-user Active Measurement service that generates probes on behalf of probers and is also implemented in the Linux kernel in order to gain a higher accuracy. In contrast to MGRP MAD does not use piggybacking but provides a interface for self-measurements of the system which again enhances the accuracy.
- Scriptroute [14] is a public Internet measurement facility that conducts remote measurements for users. Measurements are written in a special script language and uploaded to a server. Afterwards, the server performs the desired measurement in a secure way by providing several mechanisms to the user which ensure that a measurement does not exceed a given bandwidth or no bad packets are generated.

Obviously, there is a large amount of measurement tools available online [17, 18]. Combining the features of these tools with the Manager Protocol might lead to even more sophisticated applications for network measurements.

7. CONCLUSION

This work presented traditional measurement techniques and the Measurement Manager Protocol, a flexible and efficient monitoring protocol. Based on an experiment MGRP's performance was demonstrated and proved the potential of this approach. Furthermore, TCP Sidecar was introduced which presented a security oriented way of measuring networks and introduced new features like NAT detection.

Both, MGRP and TCP Sidecar provide a interface to collect information about the network. Subsequently, this feedback can be used to improve applications and protocols. Especially MGRP has great potential to be used by streaming applications to enhance the user experience without harming the network. Furthermore, both mechanisms are able to detect improving network conditions. This information is very important to all kinds of applicatons since it enables them to leave the congestion avoidance phase earlier. The information could also be used to replace the slow start phase of TCP after a congestion occured because the application is aware of the maximal available bandwidth.

To conclude, measurements have the ability to solve a large range of problems - e.g. performance issues. However, the resulting measurement overhead has to be taken into consideration to prevent network exhaustion and co-occuring delays. Moreover, the whole measurement process must be transparent to the user to keep Internet usage as simple as possible.

8. REFERENCES

[1] C. Williamson, "Internet traffic measurement," *IEEE Internet Computing*, vol. 5, no. 6, pp. 70–74, 2001.

[2] D. Verma, *Principles of Computer Systems and Network Management*. Springer-Verlag New York Inc, 2009.

[3] B. Claise, "Network Management - Accounting and Perfomance Strategies," *Cisco press*, p. 672, 2007.

[4] P. Papageorgiou, "The Measurement Manager: Modular and Efficient End-to-End Measurement Services," *Doctor*, no. 1, 2008. [Online]. Available: http://drum.lib.umd.edu/handle/1903/8900

[5] P. Papageorge, J. McCann, and M. Hicks, "Passive aggressive measurement with MGRP," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, p. 279, Aug. 2009.

[6] W. Stevens, M. Allman, and S. Paxson, "RFC 2581: TCP Congestion Control," 1999.

[7] R. Sherwood and N. Spring, "Touring the internet in a TCP sidecar," *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC '06*, p. 339, 2006.

[8] R. Sherwood, "A platform for unobtrusive measurements on PlanetLab," *Proceedings of the 3rd conference on*, 2006.

[9] L. Wang, K. Park, R. Pang, V. Pai, and L. Peterson, "Reliability and Security in the CoDeeN Content Distribution Network," in *Proceedings of the USENIX 2004 Annual Technical Conference*. USENIX Association, 2004, p. pp 14.

[10] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "OpenDHT: A public DHT service and its uses," *Interface*, vol. 35, no. 4, pp. 73–84, 2005.

[11] B. Wong, A. Slivkins, and E. G. Sirer, "Meridian: a lightweight network location service without virtual coordinates," in *Proceedings of the 2005 conference on Applications technologies architectures and protocols for computer communications*, R. Guérin, R. Govindan, and G. Minshall, Eds., vol. 35, no. 4. ACM, 2005, pp. 85–96.

[12] M. J. Freedman, E. Freudenthal, and D. Mazières, "Democratizing content publication with Coral," in *NSDI*. USENIX Association, 2004.

[13] K. Harfoush, A. Bestavros, and J. Byers, "An Active Internet Probing and Measurement API," 2002.

[14] N. Spring, D. Wetherall, and T. Anderson, "Scriptroute: A Public Internet Measurement Facility," *Proc USENIX Symp Internet Technologies and Systems USITS Mar*, 2002.

[15] J. Gonzalez, "pktd: A packet capture and injection daemon," *Passive and Active Measurement Workshop*, 2003.

[16] J. Sommers and P. Barford, "An active measurement system for shared environments," *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement IMC 07*, p. 303, 2007.

[17] Caida, "Performance Measurement Tools Taxonomy."

[Online]. Available: http:
//www.caida.org/tools/taxonomy/performance.xml
[18] L. Cottrell, “Network Monitoring Tools,” 2010.
[Online]. Available: http:
//www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html

doi: 10.2313/NET-2011-05-2_07

Dependency analysis via network measurements

Philip Lorenz
Betreuer: Dipl.-Inform. Lothar Braun
Hauptseminar: Innovative Internet-Technologien und Mobilkommunikation WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme und Systemarchitektur
Fakultät für Informatik, Technische Universität München
Email: lorenzph@in.tum.de

ABSTRACT

Large scale computer networks consist of a vast number of interoperating services. Often, the interchange between those services is not documented leading to a variety of issues. Network dependency analysis aims to automate service dependency discovery. In this work several different approaches to network dependency analysis, ranging from active to passive approaches, will be introduced and evaluated.

Keywords

Network Dependency Analysis, Sherlock, Orion, Active Dependency Discovery, Traffic Dispersion Graphs

1. INTRODUCTION

Modern enterprise IT infrastructures consist of thousands of participants using a large amount of different applications. A survey performed by the Wall Street Journal in 2008 ([11]) reports that in large companies such as HP more than 6000 different applications are in use. A lot of the applications require access to one or more network services making availability of these services crucial. Often, these services are even dependent between themselves further increasing complexity. For example, a seemingly simple task such as opening a web site has at least two dependencies - the DNS server for name resolution and the web server itself which returns the page. In some networks a proxy server may also be required introducing an additional dependency.

Dependency documentation of network services is often not readily available - e.g. if the product was developed within the company without documentation guidelines. Even if the documentation is available, extracting dependency information for the multitude of services can be very time consuming and error prone. Especially, as the network evolves over time, documentation may become outdated.

Historically systems ([3, 9]) which automatically detect network topology and services were developed. However, these systems did not extract relations between the different components of a network. They rather relied on expert and business knowledge to formulate application dependencies. Other approaches rely on instrumenting the software stack in order to extract dependencies. Pinpoint ([5]) integrates into the J2EE stack, a platform for developing Java-based enterprise server applications, enabling tracing of individual requests. X-Trace ([7]) is a tracing framework supporting a number of different OSI layers (typically the network, transport and application layer). Both approaches are of limited scope as they require detailed implementation knowledge of the software stack and therefore may be troublesome to deploy.

Bahl et. al ([1]) identify several areas which would benefit from the availability of network dependency information:

Fault localisation: Consider a service that is not functioning properly. Dependency information can be used to determine the root cause of the problem as all components which may be responsible for the failure are known. Applied to the web browsing example introduced above, the proxy server may be load balanced - e.g. several physical servers are responsible for fetching the website. If one of those servers fails dependency information may be used to track down the actual machine.

Reconfiguration planning: Companies usually run a lot of servers which sometimes have been in use over the course of several years. Sometimes the tasks of a specific server are not known by the administrators. Imagine that a server which use was not documented, running a backup database, is removed during IT reorganisation. In the best case backups are available but those may be several hours old. And even if those are not too old there is a downtime which might restrict employees from performing their work. In this case dependency information aids system planners in making choices when reorganising the IT infrastructure.

Help desk optimisation: In case a component fails many different applications may be affected. For example, the failure of an Active Directory server may affect a vast amount of the users in an organisation. As dependency graphs allow the extraction of all affected components help desk employees can troubleshoot problems more quickly and efficiently as a specific problem description can be mapped to the actual root cause. This not only avoids unnecessary problem mitigation strategies (e.g. please reboot your computer) but also allows ticket prioritisation if a single failure created a lot of support requests.

Anomaly detection: Dependency graphs show the relations between network components at a given point of time. A rapid change of dependencies may be a sign of an anomaly in the system. If such a change is detected

doi: 10.2313/NET-2011-05-2_08

human supervisors may be alerted to further inspect the found issue.

This paper presents various systems which are used to extract dependency information from a network. In section 2 terminology used throughout the paper will be introduced and explained. Section 3 presents Active Dependency Discovery, an approach which actively influences the network in order to derive dependencies. In section 4 several non-invasive systems are introduced.

2. BACKGROUND

Network dependency analysis attempts to recognise dependencies between members of a network. For example, a system administrator might be interested whether Host A depends on Host B or vice versa. This is a high level viewpoint as the interchange between services is not of interest. In this work a host based dependency between Host B on Host A will be expressed as $(A) \rightarrow (B)$. Note that this relation is not symmetric ($(A) \rightarrow (B) \nRightarrow (B) \rightarrow (A)$) - e.g. if a call to host A depends on B it does not necessarily follow that a call to host B depends on A.

In other cases one may be interested in the actual dependencies between different services. A service can be described by its IP address and the port it provides its services on. Formally, this can be expressed as the three-tuple $(IPaddress, port, protocol)$. For example, the web server at www.in.tum.de can be expressed as $(131.159.0.35, 80, TCP)$. A dependency between two services can then be described using a similar notation as above, by replacing the host with the service part. It is important to realise that a dependency between service A and B does not necessarily mean that every access to service A also triggers an invocation of service B.

Dependencies can be split into two groups - remote-remote (RR) dependencies and local-remote (LR) dependencies. A remote-remote dependency describes that in order to access service B service A has to be invoked first. A typical example for a RR dependency is browsing the web. Before the web browser contacts the web server, the domain name of the website has to be resolved . In order to do so, the DNS service is queried and as soon as the name has been resolved the web server can be contacted on its IP address. This example also illustrates that a dependency is not always visible. Due to caching at the operating system level DNS lookups do not happen every time the web server is contacted.
On the other hand, LR dependencies are triggered by an incoming service call resulting in an outgoing service call. A web server accessing a database to provide the information for a web page is an example for this type of dependency.

Another important issue, when dealing with services, is the distinction between persistent and dynamic services. A persistent service is a long-lived service often using a well-defined port for its purpose. Examples include web or mail servers. In contrast dynamic services are usually short-lived and not meant to serve more than a couple of other clients. Peer to peer applications such as Skype typically fall into this category.

The results of a network analysis can be evaluated using three metrics:

True positives The dependencies found which are ground-truth dependencies - e.g. dependencies which have been verified to be real dependencies.

False positives Classified dependencies of the system which are not actual dependencies of the network. For example, regular background traffic may be misclassified by the system as a dependency.

False negatives Actual dependencies which were not detected by the network analysis system. There are several reasons which lead to false negatives such as a sampling rate which is too high or a lack of traffic which triggers the dependency.

In the case of a perfect analysis engine the true positives will exactly match the actual dependencies of the host or service. However, it is unlikely for a system to correctly detect all and only the correct dependencies, hence the other two metrics play an important role as well. A large number of false positives may be cumbersome if manual inspection of the results has to be performed. In contrast, even a small number of false negatives may have an impact on the operation of the system if the missing dependencies are not found by manual inspection.

3. ACTIVE SYSTEMS

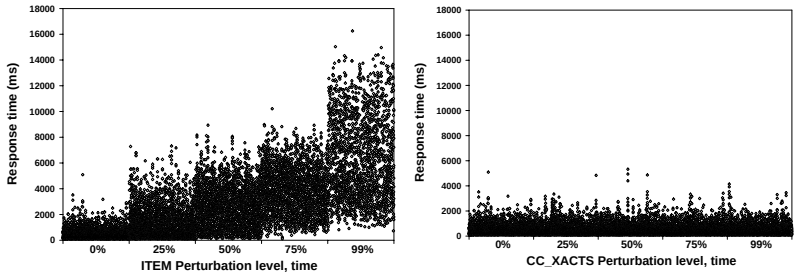
Dependency analysis systems can be categorised into two groups: active and passive systems. Active systems attempt to determine dependencies by modifying the observed system. Modification includes changing parameters of components as well as injecting network traffic generated by the dependency analysis into the system. This means that the analysis process may influence the system behaviour or, in the worst case, disturb the operation of the system.

3.1 Active Dependency Discovery

Brown et. al ([4]) introduce a system called Active Dependency Discovery (ADD) which determines dependencies by actively perturbing services in the network. This approach focuses on fine grained dependency analysis hence some information about the observed system should already be available. The dependency analysis process is split into four major steps:

1. **Node/Component identification:** In this step hosts or components that are relevant to the analysed system are identified. The list of potential components may come from various data sources such as inventory management software or from coarser grained dependency models.
2. **System instrumentation:** Probes and other components are installed to measure the effect of the perturbation within the network. Potential metrics include availability or performance data (e.g. response time).
3. **System perturbation:** In order to measure the effects of the perturbation, a specific workload should

doi: 10.2313/NET-2011-05-2_08



(a) (b)

Figure 1: Perturbation intensity vs response time of an intranet portal ([4])

be chosen and then be continuously applied to the system. In case of an Internet portal, one possible workload could be a list of frequently visited subpages. As soon as the workload is applied, components should be perturbed at varying intensity. For example, one could simulate network loss ranging from 0%, meaning no packet loss, to 100%, resulting in complete loss of connectivity. During this step the instrumentation systems set up in step 2 are used to log the system response to the perturbation. It is also possible to perturb multiple components at the same time which enables the discovery of complex dependencies such as load balancers or replicated components.

In their work, Brown et. al use a e-commerce system to perform fine-grained dependency analysis. This system had a database backend which the authors perturbed by locking individual tables making data retrieval impossible until the lock expired. The authors then recorded the effect on the response time while requesting different parts of the site (e.g. viewing a list of products). Figure 1 shows the effect on the website response time after applying the perturbation at varying intensities to two tables of this e-commerce system (ITEM containing the items available for sale and CC_XACTS which contains credit card transactions).

4. **Dependency extraction:** After the perturbation step is completed, models of the logged data can be created. In this step, the various metrics recorded through instrumentation are related to the perturbation settings at the given point in time. The goal is to identify dependencies by determining the statistical significant correlations. This does not only allow the extraction of dependencies but also the strength of the dependency for the given workload. When looking at the example in Figure 1, it can be clearly seen that a longer locking time on table ITEM leads to a higher response time of the web server. This indicates that a dependency between the sample workload and the given database table exists. On the other hand, no unusual increase in response times for the CC_XACTS table can be seen. This suggests that the workload is independent of the given table. While in their work, perturbation is not performed on the network layer, other perturbation methods such as simulating packet loss, can be used to find the dependency on the database server. In order to lower the cost of dependency extraction, the raw results can be aggregated.

Active approaches such as ADD have several disadvantages when applying them to real world networks. Due to their invasive nature the performance of the network may be negatively affected, simply due to adding additional load to the network. In the case of ADD, perturbation is bound to negatively affect the network services if it is applied to the production environment. Hence, ADD is best used in development environments which simulate the actual network. However, this may cause additional problems if the development environment does not exactly behave like the production environment. Another problem of ADD is its dependency on domain knowledge. In the best case only the workload has to be created but generating an exhaustive workload may prove to be difficult, potentially leading to false positives. Additionally, ADD requires the installation of probes throughout the network hence a priori knowledge about the network topology is required. The acquisition of network topology is not within the scope of ADD but will likely require manual intervention which filters candidate instrumentation targets.

4. PASSIVE SYSTEMS

In contrast to active systems, passive dependency analysis does not interfere with normal system behaviour. In order to derive dependencies only information produced by the network itself is used - e.g. no traffic is generated in order to determine the dependencies within a network.

4.1 Sherlock

Bahl et. al ([2]) introduce a system which aims to aid IT administrators in troubleshooting problems. In order to achieve this goal, a dependency analysis component was developed which passively monitors the network and attempts to automatically create a graph describing the network components and the services provided within the network.

4.1.1 Architecture

Sherlock consists of a centralised *Inference engine* and several distributed *Sherlock agents*. The agents sniff network packet data and compute the dependencies for their attached network segments and the corresponding response time distributions. This data is then relayed to the inference engine which uses the information to perform fault localisation. An agent may be installed as a system service on single hosts

doi: 10.2313/NET-2011-05-2_08

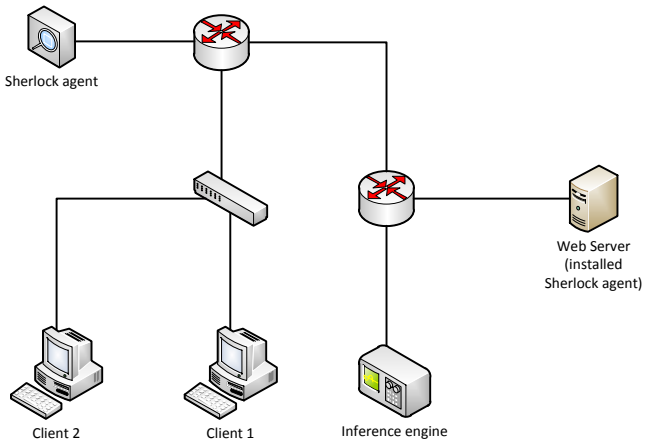


Figure 2: Sample of a Sherlock deployment

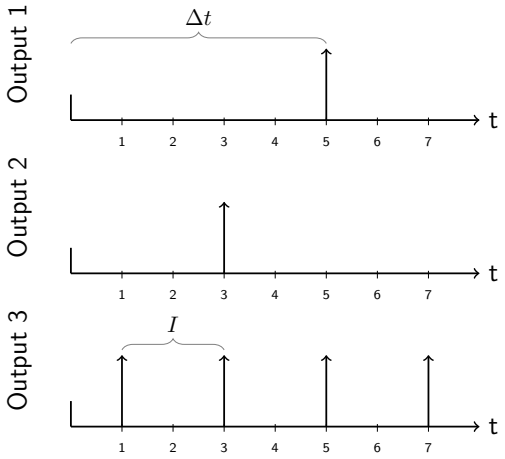


Figure 3: Exemplary packet flow

but can also be used to process data received from a monitoring port at a router or other network hardware. Figure 2 illustrates a sample deployment of Sherlock within a network. It includes an agent connected to a router and one agent directly installed on a web server. Data is collected at an independent host.

4.1.2 Dependency extraction

Sherlock analyses the packets captured, trying to find correlations between single packets directed towards a service. Rather than interrelating all packages, which would result in a severe performance loss, a time-window based approach is used: Let t_0 be the time at which an outgoing service request to service B is observed. Sherlock will choose all other outgoing service requests within the time window Δt before t_0 as dependency candidates. The remote-remote dependency probability that a host accessing service B is dependent on service A can then be expressed as the conditional dependency $Pr[A|B]$ - e.g. the number of times within the trace that A was accessed within the time window before seeing an invocation of B divided by the total number of invocations of B. Figure 3 illustrates such a packet time line. In this case Output 1 is the packet which is analysed. Let the time window Δt be set to 5 seconds. Both, Output 2 and Output 3, have sent packets indicating potential dependencies.

In order to deal with chance co-occurrence, which may be falsely assumed if another service is called often during the trace, Sherlock applies a simple heuristic to filter the results. Let I be the average invocation time interval of the noisy service. Only if the conditional probability is a lot larger than $\frac{\Delta t}{I}$, the dependency is assumed to be valid. Applying this technique to the example introduced above will exclude Output 3 as a dependency (the average interval I for this output is 2) as the resulting chance co-occurrence factor is larger than 1.

The dependency extraction process can solely be controlled by the selection of the time window length Δt. Choosing this value too high may introduce false positives as services called with a relatively high frequency will be falsely classified as dependencies. On the other hand, picking a value which is too low may result in false negatives. According to the authors, a time window of 10ms has proven to be a good choice detecting the majority of the service dependencies.

Data generated by the agents is then transmitted to the central inference engine which further aggregates the data eliminating potential false positives. For example, a client which always relies on a proxy server to perform its networking tasks may introduce false positives. Additionally, this aggregation enables the discovery of seldom accessed service dependencies as the combination of multiple data sources may provide enough data points to mark the dependency as statistically significant.

4.2 Orion

Orion by Chen et. al ([6]) is a dependency analysis engine sharing many basic concepts with the Sherlock system. However, several changes were made to improve the quality of the dependency detection.

To identify a single service invocation, Sherlock groups all contiguous packets with the same source and destination address and port without considering other transport layer attributes. In contrast, Orion aggregates individual packets depending on the protocol headers into flows. In the case of UDP, a stateless protocol, a timeout mechanism is used to determine the flow boundaries. For TCP packets, header flags are used. Example flags include the SYN, FIN, RST but in case of long-living connections the KEEPALIVE messages can be used as well. The reason for including KEEPALIVE messages is simple: If they were not used, the length of flows may include too many packets negatively influencing the dependency extraction performance. The utilisation of flows offers several advantages over a raw packet based approach: (i) the computational overhead is kept low as the number of samples decreases (consider 1 flow vs at least 3 packets for a TCP handshake) (ii) avoid redundancy and therefore skewed results which may occur if multiple packets are transmitted for a single service invocation.

Orion supports both remote-remote and local-remote dependency detection. For each potential dependency, a delay distribution is built. Hence, a system offering n local services and accessing m remote services will have $n \times m$ LR delay

doi: 10.2313/NET-2011-05-2_08

distributions and $m \times m$ RR delay distributions. Similar to Sherlock, Orion uses a time window in order to further reduce processing overhead. However, in the case of Orion the time window is significantly larger (3 seconds) but flows are grouped into smaller intervals, named bins.

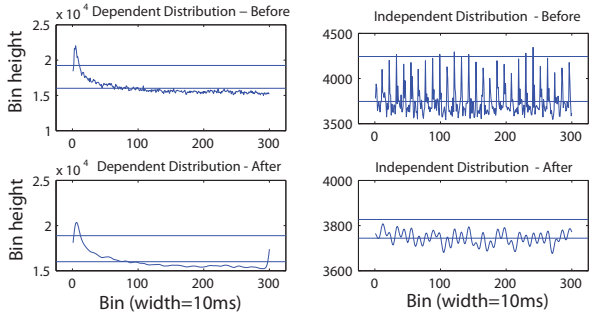


Figure 4: Delay histograms before and after the application of a low pass [6]

Similarly to Sherlock, Orion has to deal with chance co-occurrence which may introduce false positives in the dependency extraction results. As these independent packets do not follow a specific pattern they introduce random spikes in the delay distribution histogram (Figure 4). Orion treats the delay histogram as a signal and uses a common signal processing technique to eliminate the random noise. First the signal is transformed into the frequency domain (e.g. by using the Fast Fourier Transform). Afterwards, a low pass filter is applied, removing the high frequencies from the signal. This results in a smoothened signal as shown in the graphs at the bottom of Figure 4. Orion decides the validity of a dependency based on a specific bin-height threshold (indicated as a horizontal line in the graphs) - e.g. if there is at least one bin with a height above that threshold the dependency is regarded as valid. The impact of filtering can be seen on the right hand side graph where it is applied to a true negative. While without filtering, several peaks were above the threshold, these are eliminated after the application of the low pass, preventing false positives.

Similarly to Sherlock, Orion performs aggregation of the client data sets. However, not only service invocations are aggregated but also services themselves. In corporate networks, frequently used services such as DNS servers and proxy servers are load balanced in order to improve performance. Orion allows those clusters to be represented as a single server through manual input. Aggregation of these clusters may be semi-automated if a logical pattern is available to group the hosts providing these services. An example of such a pattern are reverse DNS names such as $dns-x.network.com$, where x is a number for a specific host part of the cluster.

Due to their operating system independent design Orion and Sherlock offer great flexibility. There are several deployment possibilities which ease the integration within the network. Additionally, this approach enables the detection of exotic dependencies as the amount of logging data generally exceeds those of other solutions. However, this comes at the cost of accuracy. The time-window based approach leads to a trade-off between false and true positives. In addition, as any statistical approach, these systems are highly dependent on the amount of sampling data. This means that the more samples are available the better the detection will become. Another limitation of the approach stems from the layer 4 and below restriction. Both systems do not attempt to parse application payload and will therefore always be restricted in the dependencies they can find.

4.3 Macroscope

Popa et. al introduce Macroscope ([10]) which levitates some of the problems solely packet based dependency analysis systems have due to statistical uncertainties. Macroscope follows a similar architecture as Sherlock and Orion. Network traces and application data is collected at multiple *tracers* deployed on end-systems. The *tracers* relay the data to a central *collector* which aggregates and preprocesses the data and passes it on to the *analyzer* for dependency extractions.

Macroscope uses operating system knowledge about active connections in order to identify RR dependencies of single applications. Most operating systems allow querying active connections using either system calls (e.g. on Windows `GetExtendedTCPTable` or `GetExtendedUDPTable`) or through the filesystem (e.g. on Linux in `/proc/net`). These lists contain the source IP and port, as well as the target IP and port, and the unique process identifier of the process owning the connection. Rather than constantly polling for connection information, Macroscope samples this data periodically in order to minimise resource usage. However, choosing a sampling interval which is too large may result in missed dependencies, especially if the connection duration is always lower than the interval.

Macroscope distinguishes between transient relations and static dependencies. These dependencies are similar to the concept of persistent and dynamic services introduced in section 2 - e.g. a transient relation corresponds to a dynamic service call while a static dependency involves a persistent service call. In order to generate the dependency output the system first classifies all applications into two groups (i) applications with only static dependencies (ii) applications with static dependencies as well as transient connections. Mathematically, the classification into the two groups can be expressed as follows: Let N^a be the number of application instances a within the trace, N_s^a the number of instances of application a using service s, $V_s^a = N^a - N_s^a$ (i.e. the number of application instances which did not use service s) and S^a the number of services contacted by all application instances of type a. The transient dependency metric is then calculated as follows:

$$M^a = \sqrt{\sum_s \frac{{V_s^a}^2}{S^a}} \tag{1}$$

When M^a is 0 all applications instances use all services (as V_s^a is 0). However, if an application only uses transient connections the value of M^a will be maximal at $N^a - 1$ as for each service V_s^a will be $N_a - 1$ (i.e. each instance is the only one using service s). This means that the closer the value of M^a is to 0, the more likely it is that the application only has static dependencies. Using this metric, the authors classify all applications which have a value less than a certain

doi: 10.2313/NET-2011-05-2_08

percentage of the maximum value (e.g. $M^a \leq T \times (N^a - 1)$) into group (i). If an application belongs to this group all of the services it invokes are regarded as dependencies. If the metric is above the threshold an application falls into group (ii) and requires further processing before the static dependencies can be extracted: First of all, all invocations targeting a port below 1024 are considered to be static dependencies[1]. For all other static dependencies the following two conditions must hold:

$$\frac{U_s^a}{U^a} \geq U \wedge \frac{N_s^a}{N^a} \geq I \tag{2}$$

where U^a are the number of users using application a, U_s^a is the number of users that have connected to service s through application a. Note that N^a is the number of application sessions (e.g. the number of unique *(Process Identifier, Application, Source IP)* tuples) while U^a is the number of active application installations a (e.g. the number of unique *(Application, Source IP)* tuples). Essentially, U is the relative number of users using application a which access service s. This metric prevents biased results if a single application installation, making up a large part of the sample set, uses service s frequently. On the other hand, I captures the relative amount of application sessions accessing service s. Dependencies are regarded as static if both values exceed 10% (based on experimental results of the authors) - i.e. at least 10% of the application installations, as well as application sessions, accessed service s.

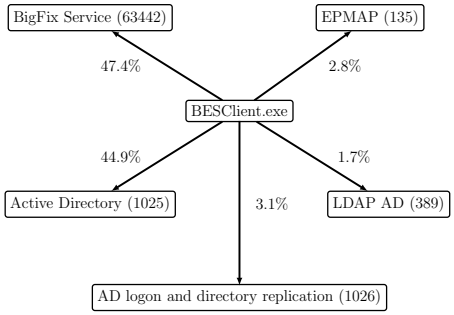


Figure 5: Macroscope sample output (based on [10])

Next to dependency extraction, Macroscope also offers dependency profiling mechanism. This enables detailed dependency analysis, for example by inspecting the amount of traffic generated by a dependency. Also, causal relations between dependencies can be derived. Figure 5 shows a graphical output by the Macroscope system for the BigFix Enterprise Suite, an application for remote system administration. The edges are labelled with the relative traffic usage of each dependency. It can be clearly seen that the majority of traffic is directed towards the BigFix service itself followed by the Active Directory server.

Due to the use of operating system knowledge, Macroscope has a better dependency detection ratio than other solutions such as Orion or Sherlock. Other than the identification of transient relations, there are no further statistical based steps in this approach. This leads to a low number of false positives as completely independent connections are not even considered to be a dependency. However, this comes at the price of flexibility. Macroscope requires the installation at each endpoint in order to determine application dependencies. While this may be feasible in some environments, such as in homogeneous setups, it may become more difficult if a multitude of platforms has to be supported. Additionally, the overhead of deploying Macroscope on every system may be problematic. Another issue of the operating system based approach is, that some dependencies will be simply missed as messages invoking them are not dispatched by the application itself. One popular example are DNS name queries which are usually handled directly by the operating system and therefore are not directly associated to the querying application in the connection table.

[1]These are well known ports specified by IANA at `http://www.iana.org/assignments/port-numbers`

4.4 Traffic Dispersion Graphs

Iliofotou et. al ([8]) introduce Traffic Dispersion Graphs (TDG) in order to extract dependency information from a network. While their work concentrates on identifying peer to peer applications many of the heuristics are also applicable to generic dependency analysis.

A TDG is a graph $G = (V, E)$, where the vertices, V, represent the nodes within a network and the edges, E, connect two nodes only if a flow between the two nodes exists. Edges are directed so that the initiator of the connection is represented. In the case of TCP connection, SYN or SYN/ACK packets are used to derive the direction while in the case of UDP the first packet of the data sample is used.

The resulting graphs can be filtered using *edge filters*. For example, the destination port of a connection could be used to filter for a specific service such as HTTP on port 80. These filters are named *TDG port filters*. Other potential filters include filtering by traffic rate or by traffic count.

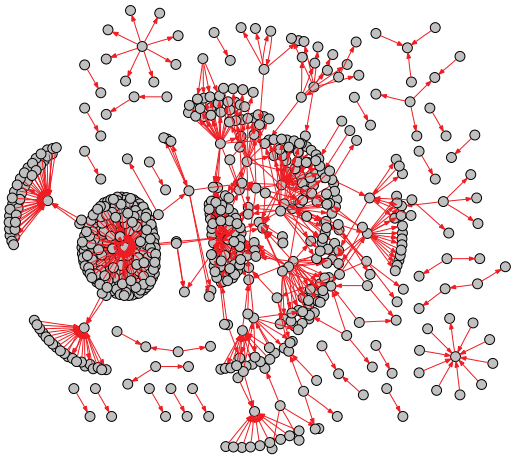


Figure 6: Visualisation of a TDG with a DNS port filter [8]

Visualising TDGs is a very expressive way of showing dependencies of a network. By using a graph layouting algorithm which places connected nodes within the same area, single dependencies can be spotted quickly. Distinguishing servers from clients can be done by looking at the degree of a node and the direction of the edges. For example, Figure 6 shows a TDG with a DNS port filter set. The DNS servers can be easily recognised due to the amount of incoming edges.

doi: 10.2313/NET-2011-05-2_08

Mathematically several graph related metrics can be used to inspect TDGs. For example, a TDG can be analysed using the average degree of nodes. The average degree is the number of incoming and outgoing edges of a node. Graphs with a large number of high average degree nodes are typically tightly connected. Another metric available is the In-and-Out degree (InO). The InO is the percentage of nodes which have a non-zero in-degree as well as a non-zero out-degree. Typically, servers will have a low InO as they generally do not use many outgoing connections.

TDGs show their strength when used for detection of single services. Using Traffic Dispersion Graphs, network phenomena can be either inspected visually or identified using graph metrics. While the authors already introduce various metrics which can be used to categorise filtered TDGs, automatic dependency discovery based on these graphs still has to be part of further research. However, TDGs can already be used to detect certain applications or for anomaly detection within networks.

5. SUMMARY

This work introduces several different dependency analysis systems ranging from active to passive approaches. In the following section the different approaches will be compared, highlighting their strength and weaknesses.

Active Dependency Discovery enables fine grained dependency detection at the cost of generality. Because of the active approach some knowledge about the service tested, such as its external communication protocol, must be known. Additionally, a typical workload has to be created before dependency extraction can begin. Depending on the type of service, this can be automated. In other cases, manual creation of the workload may be needed. However, the results of the dependency extraction are more fine grained than those of the other systems. For example, certain parts of the workload can be related to specific dependencies.

In contrast Orion and Sherlock do not need detailed knowledge of the system itself. As they work on raw packets, data can be collected anywhere in the network. This enables large-scale deployment and dependency detection for any application in the network without manual user intervention. However, as with any statistical approach results do not always match ground truth dependencies. While both systems can be configured for specific workloads there is a trade-off between the number of false positives and true negatives potentially requiring analysis by human operators after dependency extraction.

Macroscope tries to levitate this problem by utilising operating system knowledge to extract dependencies. As a result, both the number of false positives and especially the number of true negatives are significantly lower compared to Orion and Sherlock. However, due to the endpoint installation dependency, deployment within the network, especially in heterogeneous networks, is more difficult.

Traffic Dispersion Graphs define a methodology for capturing network communication in a graph data structure. This data structure makes it possible to visualise relations between nodes in a network but also enables the application of graph metrics for computational feature extraction. TDGs aim to detect the presence of specific applications rather than automatically inferring all dependencies between services. They provide a high level viewpoint of the network communication structure enabling the detection of network anomalies.

6. REFERENCES

[1] P. Bahl, P. Barham, R. Black, R. Ch, M. Goldszmidt, R. Isaacs, S. K, L. Li, J. Maccormick, D. A. Maltz, R. Mortier, M. Wawrzoniak, and M. Zhang. Discovering dependencies for network management. In *In Proc. V HotNets Workshop*, pages 97–102, 2006.

[2] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2007.

[3] R. Black, A. Donnelly, and C. Fournet. Ethernet Topology Discovery without Network Assistance. In *ICNP*, pages 328–339, 2004.

[4] A. Brown, G. Kar, and A. Keller. An active approach to characterizing dynamic dependencies for problem determination in a distributed environment. In *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, pages 377–390. IEEE, 2002.

[5] M. Y. Chen, E. Kiciman, E. Fratkin, A. Fox, and E. Brewer. Pinpoint: Problem Determination in Large, Dynamic Internet Services. *Dependable Systems and Networks, International Conference on*, 0:595, 2002.

[6] X. Chen, M. Zhang, Z. Mao, and P. Bahl. Automating network application dependency discovery: Experiences, limitations, and new solutions. In *Proceedings of the 8th USENIX conference on Operating systems design and implementation*, pages 117–130. USENIX Association, 2008.

[7] R. Fonseca, G. Porter, R. Katz, S. Shenker, and I. Stoica. X-trace: A pervasive network tracing framework. In *Networked Systems Design and Implementation*, number April, 2007.

[8] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, and G. Varghese. Network traffic analysis using traffic dispersion graphs (TDGs): techniques and hardware implementation. 2007.

[9] B. Lowekamp, D. O'Hallaron, and T. Gross. Topology discovery for large ethernet networks. In *SIGCOMM*, SIGCOMM '01, New York, NY, USA, 2001. ACM.

[10] L. Popa, B. Chun, I. Stoica, J. Chandrashekar, and N. Taft. Macroscope: end-point approach to networked application dependency discovery. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 229–240. ACM, 2009.

[11] J. Scheck. Taming Technology Sprawl, 2008.

doi: 10.2313/NET-2011-05-2_08

Understanding and using Balanced Security Scorecards

Aurelia Stöhr
Advisor: Johann Schlamp
Seminar Innovative Internet Technologies and Mobile Communications WS2010/2011
Chair for Network Architectures and Services
Faculty of Computer Science, Technical University of Munich
Email: aurelia.stoehr@mytum.de

ABSTRACT

In recent years, information security became more and more important due to the increased use and continuous development of information technologies. Especially companies have a great need to guarantee information security to customers, partners and within the firm. Today, it takes a great role in achieving business goals and thus it is essential to include this aspect in strategic planning and decision making. For this purpose, a Balanced Security Scorecard could be adopted by the IT department to communicate their goals to the upper management and align its strategy with the overall vision of the company. This security metrics is deduced from a business framework called Balanced Scorecard which considers financial, customer, internal business, and innovation and leading factors for performance measurement and long-term success. These perspectives are transferable to information security issues and encourage a holistic view of the people and processes that underlie sustainable success. However, this approach is strongly unfamiliar to IT employees and differs from normally used methods, which involves uncertainty and acceptance problems. Nevertheless, this framework is required to ensure the realisation of information security objectives in the context of the company's business strategy.

Keywords

Balanced Scorecard (BSC), Balanced Security Scorecard (BSSC), information security, performance metrics

1. INTRODUCTION

The Balanced Scorecard (BSC) is a business framework that has gained popularity over the last eighteen years as a holistic approach to evaluate quantitative and qualitative information with respect to the organisation's strategic vision and goals. Today, this performance metrics is effectively applied in many organisations and was labelled as one of the 75 most influential ideas of the 20th Century by the Harvard Business Review in 1997 [4]. The BSC is often considered as a solely business management tool but in fact it should be used in every business unit and in some cases even on an individual project level to develop a cross-company strategy. Hence, it is not just applicable for such obvious divisions like the marketing or controlling department but also for IT or information and communications technologies. As the paper will show, the BSC is not just a stand-alone performance measurement tool but a strategic performance management system.

Focusing on information security aspects, this paper shows how the BSC can be used within the IT department to assess its own performance as well as to integrate the IT strategy to the business strategy of the organisation as a whole. Starting with a general description of the BSC model and the process of developing a BSC in a company, this paper gives an idea of how a Balanced Security Scorecard (BSSC) could be derived and set up in concrete terms.

2. THE BALANCED SCORECARD AS A BUSINESS FRAMEWORK

The BSC was given its name by *Arthur M. Schneiderman* [15] who first created this metrics in 1987 for a semi-conductor company as an independent consultant. Nevertheless, the two men who are primarily known as the developers of the BSC are the Harvard professors Robert Kaplan and David Norton who published most of the initial specialist articles which lead to the present-day popularity of the technique. Their first paper "The Balanced Scorecard: Measures that Drive Performance" [7] emerged from a study in which 12 companies were analysed in order to develop an innovative performance measurement system. It was first published in the Harvard Business Review in 1992 and revolutionised the thinking about performance metrics. They introduced a business framework to align business activities to the firm's strategy and to monitor the progress on organisational goals. The traditional way of evaluating a firm exclusively by financial measures of performance was supplemented by operational value drivers to provide an all-encompassing strategic management and control system focusing on the progress of achieving strategic objectives in the long run. This enables a company to link its long-term strategy with its short-term actions. Therefore, Kaplan and Norton defined four primary perspectives that should help managers to focus on their strategic vision:

- Financial
- Customer
- Internal business
- Innovation and learning

These four perspectives create a balance between internal and external measures and show the trade-offs between the identified key success factors. The BSC considers past performance measures, respectively outcome measures, by regarding financial key figures as well as future performance indicators as internal processes and innovation and learning. So a balance between leading and lagging indicators is achieved whereas customer satisfaction can be interpreted in both ways [7]. Referring to this, a fitting comparison from *Martinson* [12] says: "Medical doctors typically examine a patient's health by checking their heartbeat, blood pressure, and the composition of body fluids. Similarly, a few key indicators provide a reliable guide to corporate health."

After Kaplan and Norton sketched the general methodology of the BSC, they wanted to go into more detail and published several related empirical studies, enhancements and of course their most

doi: 10.2313/NET-2011-05-2_09

renowned book "The Balanced Scorecard" (1996). They tried to provide a manual which describes the optimal usage of the BSC. Formally, many companies managed to execute only about 10% to 30% of their formulated strategies which strongly improved in firms that implemented the BSC.

According to a survey executed by *Bain & Company* [13] in January 2009, about 53% of 1,430 questioned international executives regularly use the BSC being the sixth popular management tool.

A lot of research is done concerning representative examples of successful implementations of the BSC. *The Balanced Scorecard Institute* [3] maintains a list of famous BSC adopters including German firms like *Siemens AG*, *BMW* and *Daimler Chrysler* who have succeeded in overcoming strategic problems with the help of the BSC. Many of them introduced this framework over 10 years ago and use it consistently. There are several listings called 'BSC Hall of Fame' from different publishers including profit and non-profit organisations as well as governmental institutions. *Balanced Scorecard Collaborative* [2] for example quoted Brisbane City in Australia as a young city on the move owning its success to the long-standing use of the BSC since 1997.

This framework is of course no guarantee for success but it has proven its value [4].

2.1 The four perspectives of the Balanced Scorecard

2.1.1 How does the company look to shareholders?

The **financial** perspective includes the traditional measures of growth, shareholder value and profitability as profit and loss, return on invested capital or earnings before interest and taxes (EBIT) [6]. "Financial performance measures indicate whether the company's strategy, implementation, and execution are contributing to bottom-line improvement." [7] This perspective is essential to the shareholders of a company and monitors if the organisation is able to make money, to generate growth and to reduce risk. Even if financial measures are often criticised as only providing a short-term view and causing no operational improvements, they should be employed to give periodic feedback and motivation to the operators [7].

2.1.2 How do customers see the company?

The **customer** perspective inquires the customers' needs and the success of the company in satisfying those needs. Furthermore, it identifies profitable market segments and target groups that should be addressed in order to stay competitive. The most common measures to accomplish the typical mission statement of delivering value to customers are time, quality, performance and service, and cost. The company has to make out the most important issues to their customers either by a customer survey, by finding the best practice through a benchmark or by experience. After figuring out the main goals, they are specified and translated into specific measures such as customer loyalty, market share or profit per customer [6]. Each company or industry sector might have different measures which could also vary within a timeframe or because of the necessity to quantify the effect of a recent amendment [7]. Kaplan and Norton emphasised that the BSC "is not a template that can be applied to businesses in general or even industry-wide." [8]

2.1.3 What must the company excel at?

The **internal business** perspective points out the critical processes the company has to excel at to meet the customers' expectations and reach the financial goals and thus satisfy the shareholders. This perspective also includes a demand for identifying the firms' core competencies to guarantee continuing market leadership. Typical measures regarding this perspective are order-to-cash ratios, employment of labour or product development cycle time. Hewlett-Packard for example uses breakeven time as a measurement parameter for the effectiveness of its product development cycle [7].

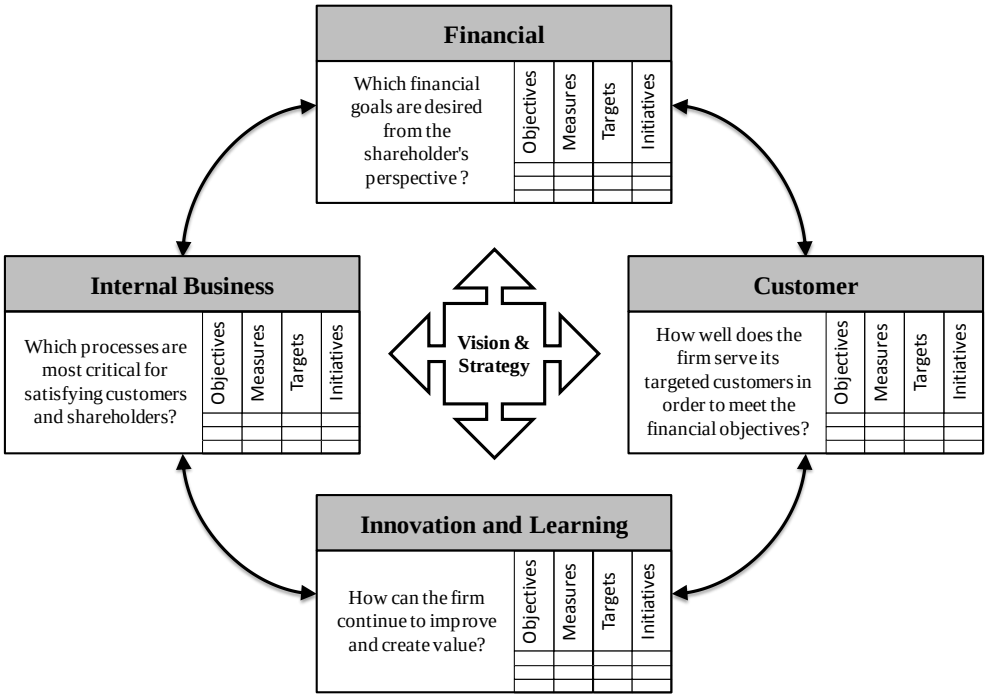


Figure 1: The four perspectives of the Balanced Scorecard

doi: 10.2313/NET-2011-05-2_09

2.1.4 Can the company improve and create value?

The **innovation and learning** perspective shows the critical factors influencing employees, systems and processes in which the company should invest to ensure long-time growth and continuous improvements. It helps to identify the optimal infrastructure and knowledge the people have to be equipped with in order to attain the appropriate objectives. Therefore the parameters that involve competitive success have to be identified while considering that these factors keep changing over time due to global pressure of competition [7]. These factors drive improvements and successes in the other three perspectives. "A company's ability to innovate, improve, and learn ties directly to the company's value. That is, only through the ability to launch new products, create more value for customers, and improve operating efficiencies continually can a company penetrate new markets and increase revenues and margins – in short, grow and thereby increase shareholder value." [7]

2.2 The cause-and-effect relationships

The vision and strategy of a company is translated in specific objectives (optimum 15-20) which are measured by selected parameters. Hereby it is very important to choose parameters with strong cause-and-effect linkages to the relative objective. Additional to the parameters, the objectives themselves have to be verified to avoid vague formulations and ensure practicable operational goals [6]. Consequently, to develop an applicable BSC for the company, it is essential to involve senior management who has the best overview of the company's strategy and priorities [7].

2.3 Establishing the Balanced Scorecard as a strategic management system

As an evolution to the theoretical idea of the BSC, Kaplan and Norton additionally developed an advisable process to develop and execute an individual BSC for any company. In their paper "Using the Balanced Scorecard as a Strategic Management System" [9] of 1996, they described a spiral of

- Translating the vision,
- Communicating and linking,
- Business planning and
- Feedback and learning

that has to be passed through. This process leads to an applicable management system to implement a value-added, customer-intensive strategy and vision. Thus, after the senior management clarified the overall strategy, the following steps should be executed.

2.3.1 Translating the vision

Translating the vision is the first step of realising a company's vision. The management needs to communicate its goals and measures in terms that employees can understand them and know what precisely they should do. It often occurs that the operators do not know which appropriate action to take and what exactly is expected from them. Therefore, it is recommended to define concrete

- **Goals**: What needs to be achieved?
- **Measures**: Which parameters are used to measure success?
- **Targets**: What is the desirable quantitative value of the chosen measures?
- **Initiatives**: What needs to be done to achieve the goals?

and a status for every objective in every perspective [13]. Another typical phenomenon is that people have different definitions of words and so it is important to clarify the exact meaning of every strategic statement.

2.3.2 Communicating and linking

The process of **communicating and linking** helps to communicate the strategy and procedure to everyone up and down the ladder and also link it to their individual objectives. The key to succeed the managements' objectives is to align the departments and employees with the overall strategy. Therefore, the senior executives prescribe the objectives for the financial and customer perspective of the BSC and let the lower levels of management formulate the other two operative perspectives. Their task is to work out beneficial performance measures and gain a better understanding of the company's long-term strategy by that. This process holds the advantage of more effective measures on the one site and stronger commitment on every management level on the other side. But of course this system needs guidance by the upper management. The responsible employees should be educated and regular informed about their mission and current status. They should also be able to discuss their ideas and results with the senior management at an interim stage. Another method to align operating units and individuals is to let them develop a personal scorecard so that they could set their own goals. Some companies even have the approach to link the compensation of their employees to BSC measures which also aligns them to the strategy but in addition bears high risks if the measures are not chosen right or the data for these measures are not reliable.

2.3.3 Business planning

Business planning is another important aspect that must be considered. Most companies have separate procedures for

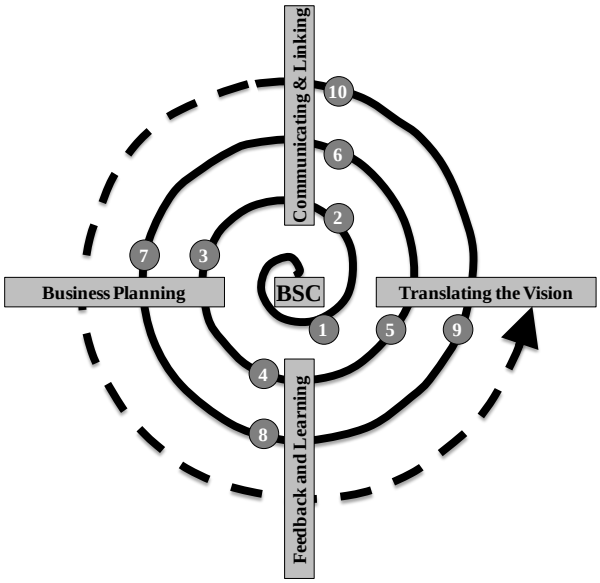


Figure 2: The cycle of developing a Balanced Scorecard

strategic planning and resource allocation. But to ensure that the budgeting system supports the BSC strategy, this process has to be integrated in strategic planning. One way to achieve that would be to set targets for every strategic measure and define short-term

doi: 10.2313/NET-2011-05-2_09

milestones to make financial budgeting on a BSC basis possible and thus move towards a long-term strategy. It can also be helpful to add another column with budget or costs to the description of the particular objective in order to allocate financial resources.

2.3.4 Feedback and learning

Finally, the BSC is accomplished by a constant operational and tactical **feedback and learning** process or in other words strategic learning. The company should periodically ask itself if their strategy was successful and if their measures and initiatives are still valid. Considering changing business conditions and new challenges, some strategy statements or measures maybe have to be adjusted and the hypotheses about cause-and-effect relationships might have to be reconsidered. There are also possible internal reasons like changing or wrong estimated capabilities that could cause the need to improve the BSC and the business planning [9].

2.4 Challenges and benefits of the Balanced Scorecard

What can be avoided using the BSC is the typical control bias of traditional performance measurement systems based on financial aspects. The BSC does not aim to control the employees but provide them an overall view of the company's strategy and vision to encourage them to contribute their share to reach common goals. Although, it can be very difficult to agree on corresponding objectives including the problem that fulfilling everyone's wishes is impossible for a complex organisation. Additionally, the already mentioned ambiguity of some goals can lead to communicational problems.

Furthermore, the BSC forces an orientation towards the future and therefore achieves a long-term perspective. Other metrics often focus on past events and financials which fade out all other aspects of acquiring a successful strategy. The BSC, however, provides a holistic view of all factors that influences organisational performance [7].

One further challenge is to avoid an overdevelopment of the BSC including too much information, details and aspects which results in a loss of utility. An organisation should never lose focus on the essential performance measurements. However, a very high grade of detailed planning can be reached by the employment of BSC software that provides real-time display of performance indicators and an overall framework to manage complexity and communication [4].

According to a survey of *2GC* [1], there are more than 100 BSC reporting software packages available provided by SAP (*SAP Business Objects Strategic EPM Solutions*), Oracle (*Oracle EPM System*) and other specialists. About 26% of the surveyed institutions used one of those tools and even 37% employed their existing standard Microsoft office software which could be enhanced by *Microsoft Office Business Scorecard Manager*. The other 37% had their own bespoke software which is very expensive and most often not necessary. Another outcome of this report is that 60% of the respondents rated their BSC as "very" or "extremely" valuable. One dominant reason why the companies were not convinced of the BSC was that they did not update the metrics for years and had not used it on a frequent level.

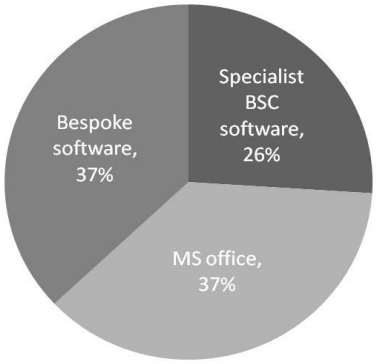


Figure 3: BSC reporting software used [1]

Even if the BSC is a suitable intern method of communicating the strategy of a company, it is not the right way of presenting it to shareholders or the public in general. It contains much too detailed information and therefore gives a deep insight into advantageous competitive structures and future activities. Generally, the main findings and strategic statements should be taken from the BSC but published in another form without confidential company data [9].

3. THE BALANCED SCORECARD AS AN INFORMATION SECURITY METRICS

What makes the BSC interesting for IT measures is that it is very convertible. The organisation is able to choose individual objectives and measures and even rename or add overall perspectives [6]. Consequently the BSC is adjustable for every form of organisation or department and includes potential applications for IT domains. The development of a specific metrics for information security is a part of the communicating and linking process (see 2.3.2). The BSC is a suitable tool to manage an appropriate integration of the IT strategy to the enterprise-standard metrics and hence the company's long-term strategy [10].

3.1 Transferability of the Balanced Scorecard to the domain of information security

At first glance it volunteers to deploy the flexibility of the BSC in order to match it with information security purposes. It would be reasonable to change the four perspectives of the BSC for example to (1) threats, (2) vulnerabilities, (3) identify and access management and (4) policies and compliance. This would adjust the BSC to a more operational metrics matching information security concerns. Even though, this would be a convenient solution for the IT department, this model could not be brought into line with other BSCs and the overall business objectives. The balanced security scorecard (BSSC) should provide a framework that the executives could easily understand and be able to consider in the unified BSC.

Another approach could be to involve an existing security taxonomy like ITIL (IT Infrastructure Library) or ISO 17799. But this would be too complex including too many areas and would be far too profound in the sense of a BSC. It is also not practical to reduce the control areas of the taxonomies to a few main concerns to form a BSSC [6].

doi: 10.2313/NET-2011-05-2_09

Martinsons et al. [11] suggested four adapted perspectives for a BSSC based on Kaplan's and Norton's framework: (1) business value, (2) (internal) user orientation, (3) internal processes, and (4) future readiness. They made this adjustment to the original BSC because it is their opinion that the information security department is an internal service supplier, in the first place.

Nevertheless, as *Jaquith* [6] proposes, this paper will stick to the original perspectives of the BSC and discusses possible relations to the objectives of information security. It will also be described how cause-and-effect measures can be defined from the viewpoint of security and can be assigned in a nontechnical language to financial, customer, internal business and innovation and learning issues.

Even if the connection between information security and the four standard BSC perspectives is not always obvious at first sight, security is only a subsector of the company and influences the success of the whole enterprise, for better or worse. Poor information security can destroy value. Sometimes, it takes just one public breach failing regulatory or data security requirements to lose shareholder confidence and to inhibit future growth. It can even result in costly legal repercussions. Problems with information security are often caused by the fact that security controls are decoupled from the mission of the company. By ignoring the needs of the company's management and workforce, they are forced to use their computers in an unintuitive way causing errors. Consequently, it is necessary to take account of the users on an individual basis and take a holistic, strategic view of information security issues [5]. Information security practitioners have to recognise that they are sales people selling their applications and service to internal staff and external customers.

3.2 Creating the Balanced Security Scorecard

3.2.1 Contributing to the value of the company

Initially focusing on the **financial** perspective, the security efforts have a strong impact on the total revenues of a firm. The ultimate aim of the BSSC would be to improve the overall financial outcomes of the enterprise. So in order to fill the financial section of the BSSC that ought to be developed, the security department should work out:

- How is information security related to the ability to generate revenues?
- In what way does effective security encourages growth by meeting the demands of customers?
- Which investments are tied directly to the security units and can they be reduced?
- Which security controls are necessary to decrease the risk of the organisation?

While the typical objectives of the BSC are the increase of revenues, return on invested capital and profit as well as reducing risk, the BSSC should support those goals by

- Increasing the usage and decreasing the risk of systems that generate revenue. These systems could be applications, tools, servers, people or other infrastructure like for example e-commerce websites, track & trace software, order management systems or payment systems
- Increasing the integrity and decreasing the risk of systems that account for revenue such as spreadsheets, accounting software or enterprise resource planning (ERP) systems
- Reducing security and compliance costs and those costs caused by downtime and other incidents

Typical financial measures would be revenues, return ratios, EBIT, earnings before interest, taxes, depreciation, and amortisation (EBITDA), cost per sales, cash flows etcetera. These parameters are amongst others influenced by security control which could be measured for example by the following key figures based on the above mentioned objectives for information security:

- Order or transaction rate or value
- System uptime, respectively costs of downtimes
- Information security expenses per employee
- Average cost of a security incident
- Cost incurred to deal with known threats
- Number of controls per transaction or accounting event
- Risk indices for those systems [6]

3.2.2 Delivering value-added products and services to end-users

More challenging is the perspective of the **customer**. As customers could easily be defined as the recipient of a product or service in the sense of a common BSC, it is not necessarily transferable to the security metrics because the security aspect is normally not directly sold to the customer. This service is rather employed by internal staff and managers of the company. But this possible redefinition of the customer perspective does not consider the impact that the security system might have to the external customers even if it is indirect. To mix those both views would be also a mistake because the interests of external customers might be diluted. Since the internal users are already included in the internal business perspective it is more important to focus on external users at this point. Hence, the task for the developers of the BSSC is to identify how the security program affects external parties:

- Does the security system encourage the customers to do business with the company or even make it possible at all?
- Does the security system satisfy binding and non binding external requirements?

Consequently, the objectives of the BSC and the BSSC are very similar. The BSC aims at creating added value for the customers with sophisticated products, competitive prices and new innovative products and thereby attract new customers to increase market share. Likewise the BSSC targets

- The increase of attractiveness of products and services and thus the number of customer orders
- The increase of electronic business transactions with customers and partners
- The guarantee of transactional availability and above all integrity
- The preservation of the reputation of the company concerning security criterions. The quality of information security should become a competitive advantage

The customer perspective is typically measured by market share, brand recognition, customer satisfaction, customer loyalty, customer acquisition rates, annual sales per customer or the like. Whereas specific measures for the BSSC could be:

- Number or percentage of customers' wins or losses due to security reasons

doi: 10.2313/NET-2011-05-2_09

- Percentage of strategic partner agreements with documented security requirements
- Volume of electronic transactions
- Cycle time to grant access to company system
- Toxicity rate of customer data
- Downtime of critical operations due to security incidents
- Number of data privacy escalations

Here, the data privacy and protection of customer information plays a very important role. Nevertheless, one should keep in mind, while regarding the objectives and measures for all of the four perspectives, that every company needs to find their individual emphasis and parameters [6].

3.2.3 Developing efficient and effective internal processes

The overall goal is to satisfy the customers' needs because this leads to profitability. Emerging from that, the propos of the **internal business** perspective is to reveal supporting operational activities to achieve that goal. Possible objectives of a company related to internal processes are the development of high-quality and innovative products, the reduction of costs, for example materials or logistics costs, the minimisation of cycle time and the optimisation of the customer service. What the security system should provide is:

- Protection of the company's information system
- Access to permitted resources
- Maximal availability of the system
- Technological agility including collaboration with the other business units

The internal processes of the information security team like project planning, application development, and operation and maintenance of current applications have to be optimised. To guarantee an effective and responsive security system at high-quality and lowest possible costs, the following objectives could be deployed:

- Ensuring the safety and security of people and information assets
- Decreasing the number and impact of security incidents
- Maximising cooperation between the security team and other business units
- Identifying security vulnerabilities and control gaps
- Minimising access privileges
- Optimising the effectiveness and reach of security control

These objectives should be monitored by empirical measures like:

- Password strength
- Estimated damage from all security incidents
- Percentage of authorised users accessing security software
- Percentage of communications channels controlled in compliance with policy
- Average time required to address an end-user problem
- Time taken to enroll a new employee
- Ratio of business units shadowed by a security team to security team staff

considering both lag indicators (oriented towards the past) and leading indicators (towards the future) [6].

3.2.4 Ensuring continuous improvement and preparing for future challenges

The last perspective that needs to be transferred is the **learning and growth** perspective. It considers the affordable skills and ability of the employees to meet the expectations and ensure future success. So, the objective of a company is to support the skills and knowledge of its workforce and to provide a productive work environment. The performance of the employees strongly correlates with the company culture, quality of internal communication and the tools they are given. To form a flexible and forward-looking security department the following targets should be focused:

- Spread and partly decentralise responsibility for security issues amongst the security team and several business units to consider differentiated user perspectives and gain a holistic view on security problems
- Offer technical training and certifications in order to evolve the required knowledge and skills of the employees
- Provide a safe environment for every business unit
- Be prepared and aware of new threats respectively requirements and develop quick countermeasures and solutions

The concrete objectives for the BSSC should therefore be related to these interests:

- Delegate the responsibility for authoring user activities directly the business units
- Promote the awareness of security threads in the whole organisation and increase collaboration between IT employees and the single business units
- Ensure continual enhancement of the skillset of the IS specialists to prepare them for future and current changes and challenges
- Put effort into the development and update of the applications portfolio and technologies that could be of value to the organisation [11]

Typical measures of the original BSC are training hours per employee, knowledge management metrics, employee productivity or participation in professional organisations. These measures can be directly transferred to information security:

- Percentage of staff with security responsibilities or certain security roles and expertise
- Percentage of employees who have completed security awareness training or professional certifications
- Training and development budget as a percentage of the overall IT budget [6]
- Technical performance of applications portfolio
- Time to implement a regulatory requirement
- Speed of dealing with a new threat [16]

3.2.5 Implementing the BSSC

The implementation of the BSSC is similar to the already described process of the traditional BSC. But this time the vision and strategy needs to be clarified, translated and communicated up the ladder to senior management and later, as a second step, to other business units and of course the IT employees. Afterwards, the resource allocation should be oriented on strategic goals and finally the performance should be periodically reviewed including necessary adjustments of the strategy. As mentioned before, it is very important to point out the cause-and-effect relationships of

doi: 10.2313/NET-2011-05-2_09

the targets and possible linkages with other perspectives. The information security strategy should especially be aligned with the overall IT management including budgeting and assigning employees to tasks and roles.

4. CONCLUSION

To conclude, it can be said that the BSSC brings substantial benefits in pursuit of aligning IT and the overall business strategy. The objectives and concerns of information security are formulated in such a way that the non-technical senior management is able to understand it easily and to foresee problems or quickly identify them as they arise. At the same time, the IT department is forced to be mindful of different drivers of its performance outside the security team. It is able to see itself in the context of the company and as an important part of the implementation of a joint strategy. It prevents a deviation of information security management from firm missions. Although, it is very difficult for IT stuff to transcribe their common technical oriented taxonomy and to think in the four perspectives of the economic-oriented BSC framework. The connections between those perspectives and information security issues are not always obvious at first sight and therefore are prone to misinterpretation and overlooking of indirect explanatory variables. But that is exactly why the IT division should work with the BSSC. They need to work out these hidden cause-and-effect relations to reveal possible causes of current problems, needs for adjustments and future chances.◆

5. REFERENCES

[1] 2GC Limited (2009): GC Balanced Scorecard Usage Survey 2009, p. 3, including www.2gc.co.uk/resources-swdb (13.01.2011).

[2] Ascendant Strategy Management Group: Balanced Scorecard Hall of Fame - Brisbane City, www.bscwiki.com/wiki/index.cfm/Balanced_Scorecard_Hall_of_Fame (13.01.2011).

[3] Balanced Scorecard Institute: Balanced Scorecard Adopters, www.balancedscorecard.org/BSCResources/AbouttheBalancedScorecard/BalancedScorecardAdopters/tabid/136/Default.aspx (13.01.2011).

[4] Bible, L. / Kerr, S. / Zanini, M. (2006): The Balanced Scorecard: Here and Back, Management Accounting Quarterly, Summer 2006, Vol. 7 Issue 4, pp. 18-21.

[5] Farshchi, J. / Douglas, A. (2010): Information security and the balanced scorecard, Computer world UK, Sep. 2010.

[6] Jaquith, A. (2007): Security Metrics – Replacing Fear, Uncertainty, and Doubt, 1st Ed., Addison-Wesley Pearson Education Inc., Boston 2007, pp. 260-289.

[7] Kaplan, R. S. / Norton D. P. (1992): The Balanced Scorecard – Measures That Drive Performance, Harvard Business Review, Jan/Feb92, Vol. 70 Issue 1, pp. 71-79.

[8] Kaplan, R. S. / Norton D. P. (1993): Putting the Balanced Scorecard to Work, Harvard Business Review, Sep/Oct93, Vol. 71 Issue 5, pp. 135.

[9] Kaplan, R. S. / Norton D. P. (1996): Using the Balanced Scorecard as a Strategic Management System, Harvard Business Review, Jul/Aug07, Vol. 85 Issue 7/8, pp. 152-160.

[10] Keyes, J. (2005): Implementing the IT Balanced Scorecard – Aligning IT with Corporate Strategy, 1st Ed., Auerbach Publications 2005, Chapter 4, p. 92.

[11] Martinsons, M. / Davison, R. / Tse, D. (1999): The balanced scorecard: A foundation for the strategic management of information systems, Decision Support Systems, Vol. 25 Issue 1, Feb99, pp.75, 81.

[12] Martinsons, M. (2002): The Balanced Scorecard: A Tenth Anniversary Review, Department of Management at the City University of Hong Kong, p. 2, www.cb.cityu.edu.hk/mgt/index.cfm?category=community (19.11.2010).

[13] Rigby, D. / Bilodeau, B. (2009): Management Tools and Trends 2009, Bain & Company, p. 7.

[14] Scherer, D. (2002): Balanced Scorecard Overview, Core Paradigm Article, 17. June 2002, p. 1.

[15] Schneiderman, A. M. (1992): Analog Devices: 1986-1992 – The First Balanced Scorecard©, www.schneiderman.com (13.01.2011).

[16] Sethuraman, S. (2006): Framework for Measuring and Reporting Performance of Information Security Programs in Offshore Outsourcing, ISACA Journal, Vol. 6, 2006, p. 5.

doi: 10.2313/NET-2011-05-2_09

Sicherheit bei Cloud Computing

Eugen Wachtel
Betreuer: Heiko Niedermayer
Seminar Innovative Internettechnologien und Mobilkommunikation WS2010/11
Lehrstuhl Netzarchitekturen und Netzdienste, Lehrstuhl Betriebssysteme und Systemarchitektur
Fakultät für Informatik, Technische Universität München
Email: wachtel@in.tum.de

KURZFASSUNG

Cloud Computing ermöglicht die Bereitstellung von IT-Diensten über das Internet, die dynamisch skaliert und kosteneffizient abgerechnet werden. Damit ist Cloud Computing für Organisationen und Service-Anbieter von großem Interesse, erübrigt sich dadurch schließlich eine eigene Hardwarelösung für die Dienste. Doch wie ist es um die Sicherheit von Cloud-Lösungen bestellt? Können Unternehmen mit ihren Systemen in der Cloud die Sicherheitsstandards ohne weiteres einhalten? Dieser Artikel beschäftigt sich mit diesen Fragen und zeigt mögliche Problembereiche sowie einige Lösungsansätze und Best Practices, die in der Praxis Verwendung finden.

Schlüsselworte

Cloud Computing, Sicherheit, Informationssicherheit, Datensicherheit, IT-Sicherheit

1. EINLEITUNG

Die Anforderungen an IT-Lösungen im Internet können sich bezüglich Skalierbarkeit im Betrieb dynamisch ändern, wodurch die Bereitstellung von zusätzlichen oder aber auch die Abschaltung unnötiger Ressourcen schnell und zuverlässig erfolgen muss, um die Wirtschaftlichkeit einer IT-Lösung zu gewährleisten. *Cloud Computing* stellt diesbezüglich eine aktuell sehr populäre Lösung bereit, die dynamisch auf den Ressourcenbedarf reagiert und gleichzeitig eine nutzungsabhängige Abrechnung bereitstellt. Damit ist Cloud Computing für die IT-Welt in vielen Hinsichten relevant. Es unterstützt neben den klassischen auch innovative IT-Dienste und ermöglicht neuartige Geschäftsmodelle im Internet. Das Wachstum für den globalen Cloud Computing-Markt soll aufgrund dieser Vorteile bis 2013 auf einen Wert von 150 Mrd. US-Dollar anwachsen [7]. Demgegenüber sind laut [11, 5] allerdings die Aspekte der Verfügbarkeit, Sicherheit und des Datenschutzes die wesentlichen Gründe für viele Unternehmen eine große Skepsis gegenüber Cloud Computing aufrecht zu erhalten.

1.1 Cloud Computing

Cloud Computing-Systeme (kurz: *Cloud*) setzen sich zum einen aus den Anwendungen (*Services*) und zum anderen aus der Soft- und Hardware, die diese bereitstellt, zusammen. Der Hauptunterschied zwischen konventionellen Systemen und Cloud Computing liegt in der Entkopplung der Daten und Software von den Servern und der Bereitstellung solcher als Services. Abbildung 1 vermittelt den Unterschied in der Architektur-Sicht. Bei konventionellen Client-Server-Systemen sind die verfügbaren Ressourcen unmittelbar sichtbar sowie deren Standort bestimmbar. Beim Cloud Computing hingegen kann von einem unendlichen Ressourcenpool ohne bestimmbare Lokalität der einzelnen Komponenten ausgegangen werden. Dieser Unterschied erlaubt eine flexible und kostengünstige Skalierung sowie Abrechnung der Cloud-Dienste, bringt aber auch problematische Aspekte wie die Erfüllung der Compliance Anforderungen mit sich. Cloud Computing wird mittels moderner Virtualisierungs- sowie Automatisierungs- und Bereitstellungstechnologien umgesetzt [12].

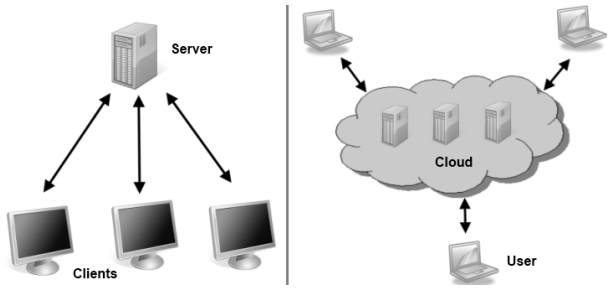


Abbildung 1: Client-Server vs. Cloud Computing-Architektur

Hintergrund. Cloud Computing wird erst in letzter Zeit in der Breite populär, doch die Idee dahinter ist nicht neu und findet bereits seit längerer Zeit Anwendung. Großunternehmen, wie Microsoft oder Amazon, setzen unternehmensintern seit Jahren darauf. Neu ist heutzutage die Verfügbarkeit solcher Systeme für die Öffentlichkeit, die durch die *Cloud Provider* bereitgestellt wird. Wir unterscheiden diesbezüglich zwischen

- *Public Clouds* stellen öffentlich zugängige Clouds dar, bei denen der Cloud Provider die Richtlinien bezüglich Sicherheit festlegt.
- *Private Clouds* gehören einem Unternehmen, das die Kontrolle über die Cloud hat und diesbezüglich auch seine Festlegungen und Standards bezüglich Sicherheit aufstellen kann.
- *Hybrid Clouds* setzen sich aus Public und Private Clouds zusammen.

doi: 10.2313/NET-2011-05-2_10

Wir fokussieren in diesem Artikel insbesondere die Public Clouds.

Genutzt werden Clouds durch Unternehmen sowie IT-Lösungsanbieter, bezeichnet als die so genannten *Cloud User*, deren Dienste ihrerseits den *Service Usern* zur Verfügung gestellt werden (siehe auch Abbildung 2).

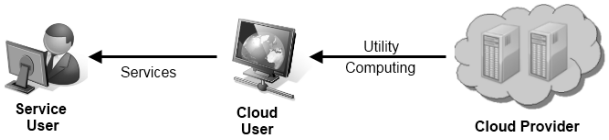


Abbildung 2: Benutzer und Anbieter beim Cloud Computing [4]

Servicemodelle. Cloud Computing ist eine Architektur zur Bereitstellung von Dienstleistungen. Diese können, je nach Cloud Provider, unterschiedlich sein (siehe auch Abbildung 3):

Software as a Service (*SaaS*) bezeichnet die Bereitstellung von Applikationen als Service über das Internet. Der Betrieb und die Instandhaltung der Applikationen wird seitens des Anbieters übernommen, wie beispielsweise bei den Google Apps.

Platform as a Service (*PaaS*) ist durch die Bereitstellung von Entwicklertools und Schnittstellen zur Entwicklung von Webapplikationen gekennzeichnet. Die Dienste werden basierend auf der Infrastruktur des Anbieters programmiert und betrieben. Bekannte Beispiele sind Google App Engine, Microsoft Azure und Salesforce.com.

Infrastructure as a Service (*IaaS*) stellt Speicherplatz oder Rechenleistung zur Verfügung. Prominente Beispiele sind Amazon EC2 und Microsoft Azure.

Chancen. Sowohl für den Cloud User als auch für den Cloud Provider ergeben sich durch das Cloud-Modell Chancen [14]:

Cloud User Der Aspekt der Kostenreduzierung aufgrund der Skalierung bei Bedarf und der Bezahlung nach Verbrauch wurde bereits angesprochen. Hinzu kommen die höhere Flexibilität bei der Wiederverwendung bestehender Systeme sowie eine Verkürzung der Entwicklungszeit zur Marktreife von Diensten. Schließlich werden die Innovationsmöglichkeiten hinsichtlich der Gestaltung neuer Systeme erweitert.

Cloud Provider Neben den wirtschaftlichen Aspekten sind die technischen Verbesserungsmöglichkeiten als Chancen zu sehen. So kann der Betrieb des Rechenzentrums optimiert und die administrativen Prozesse automatisiert werden. Dadurch können verkürzte Innovationszyklen bei der Bereitstellung von neuen Diensten erreicht werden.

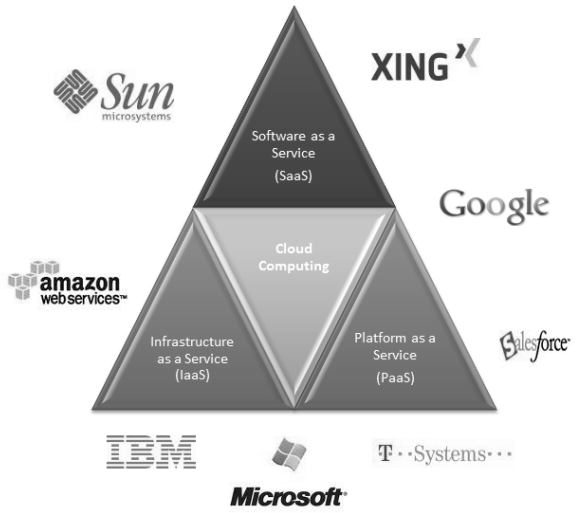


Abbildung 3: Überblick über die Cloud Anbieter sowie die angebotenen Dienstleistungen.

Sicherheit. Cloud Computing ist eine Plattform, die den Unternehmen viele Vorteile bietet, jedoch nach wie vor auf eine große Skepsis bezüglich der Sicherheit stößt. Sicherheitsrisiken können beim Cloud Computing an unterschiedlichen Stellen auftreten, sei es aufseiten des Service Users, bei der Programmierung und Bereitstellung von Diensten seitens des Cloud Users oder aber auch beim Cloud Provider. Im Folgenden diskutieren wir vor allem die Sicherheitsrisiken für den Entwickler von IT-Lösungen, die auf öffentlichen Clouds laufen, weisen allerdings auch auf mögliche Problembereiche bezüglich des Cloud Providers hin.

1.2 Verwandte Arbeiten

Zum Cloud Computing gibt es eine Vielzahl an Literatur, die das Thema der Sicherheit in der Cloud aus unterschiedlichen Blickwinkeln diskutiert. So wird in [6] vor allem die Sicherheit privater Clouds aus Sicht des Unternehmers und des Cloud Providers angesprochen sowie einige Best Practices aufgelistet. Ein Entwurf der BSI [8] listet die Mindestsicherheitsanforderungen auf, die von Cloud Providern erfüllt werden sollen. Die Studie [14] des Fraunhofer-Instituts für Sichere Informationstechnologie liefert neben Erkenntnissen und Lösungsmöglichkeiten zu Sicherheitsbedenken auch eine Taxonomie, die einen Rahmen zur Bewertung der Cloud-Sicherheit schafft.

1.3 Aufbau der Arbeit

Der strukturelle Aufbau dieser Arbeit ergibt sich wie folgt:

Kapitel 2 beschäftigt sich mit dem Thema Sicherheit bei Clouds und betrachtet dieses aus unterschiedlichen Sichten. Zunächst spricht Abschnitt 2.1 die Risiken für den Service User, sprich für den Benutzer der Cloud-Dienste, an. Folgend setzt sich Abschnitt 2.2 mit den Sicherheitsbedenken für den Cloud User auseinander, der die Dienste in der Cloud anbietet. Abschnitt 2.3 vertieft diese Thematik anhand der Cloud Storage-Systeme. Schließlich listet Abschnitt 2.4 problematische Sicherheitsaspekte auf der Seite des Cloud Providers auf.

doi: 10.2313/NET-2011-05-2_10

Kapitel 3 fasst abschließend die Arbeit zusammen und spricht zudem offene Fragen zur Thematik an.

2. CLOUD COMPUTING UND SICHERHEIT

Cloud Computing stellt eine immer populärer werdende Lösung, skalierbare IT-Dienste kosteneffizient im Internet anzubieten. Jedoch bringt die moderne Technologie des Cloud-Modells auch neuartige Sicherheitsprobleme mit sich und eröffnet neue Angriffsmöglichkeiten. In diesem Kapitel wollen wir uns mit Sicherheitsrisiken beim Cloud Computing beschäftigen und diese jeweils aus den Sichten der Service und Cloud User betrachten. Dabei fokussieren wir vor allem die Risiken, die den Cloud User betreffen, im Detail.

2.1 Risiken für den Service User

Der Service User greift über das Internet auf einen bestimmten Dienst zu. Demzufolge gelten hier die gleichen Risiken wie bei jedem IT-Dienstzugriff über das Internet. Genauer genommen ist es für einen Benutzer nicht direkt erkennbar, ob ein Dienst in der Cloud läuft oder nicht. Es ist für ihn aber auch nicht weiter von Bedeutung. Das Problem des Identifikationsbetrugs ist auch in der Cloud ernst zu nehmen. Generell ist *Fraud Protection* eine herausfordernde Aufgabe, die täglich neuen Angriffsarten ausgesetzt ist. Behelfen kann man sich beispielsweise durch Risikoauthentifizierung, bei der neben der Eingabe eines Benutzernamens und Passwords noch zusätzliche benutzerspezifische Informationen abgefragt werden.

Somit sind beim Zugriff auf einen Dienst unter anderem Brute-Force Attacken auf beispielsweise Benutzerpasswörter, Trojaner, Malware oder Phishing die wesentlichen Angriffe/Risiken für einen Service Benutzer. Weitere Informationen diesbezüglich können [3] entnommen werden.

Ein weiteres Risiko ergibt sich für den Service User bei der Nutzung von *Software as a Service* Angeboten[1]. So versprechen Dienste wie beispielsweise Google Mail einen reibungslosen Email-Verkehr, sie sind allerdings in der Regel weder vertraglich verbunden die Dienste immer und jederzeit verfügbar zu machen, noch können sie bei Datenverlusten zur Rechenschaft gezogen werden. Zudem ist die Datenschutz-Thematik bei SaaS-Angeboten mehr als nur beachtenswert. Viele Anbieter formalisieren eine vertragliche Nutzung der Daten für Eigenangebote oder sogar für Drittanbieter. Vielen Benutzern solcher Dienste ist das in der Regel aber nicht bekannt. Diese und weitere SaaS-Problematiken sind jedoch wieder nicht cloudspezifisch, sondern gelten grundsätzlich für die SaaS-Angebote.

2.2 Risiken für den Cloud User

Die Verlagerung der IT-Dienste oder IT-Infrastruktur kann, wie bereits angesprochen, finanziell große Vorteile aufweisen und zusätzlich operative Freiräume in der Gestaltung der IT-Landschaft gewährleisten. So kann die Infrastruktur leicht ohne zusätzliche Hardware/Software-Investitionskosten ausgebaut werden und die überflüssigen Wartungsausgaben entfallen. Der Transfer auf eine cloudbasierte Lösung ist allerdings für den Cloud User mit Sicherheitsbedenken verbunden. So kann die bereits vorhandene Sicherheitsstrategie eines Unternehmens auch in der Cloud verwendet werden, sie muss allerdings an die spezifischen Gegebenheiten des Cloud-Modells angepasst und erweitert werden. Beim Programmieren von Services für die Cloud sind zudem einige Besonderheiten zu beachten, die beispielsweise in der konventionellen Webentwicklung unproblematisch sind. Ressourcen unterliegen in der Cloud einer gemeinsamen Nutzung, Kontrolle, Verwaltung sowie Teilung, so dass auch hier Anpassungen in bewährten Sicherheitsmaßnahmen notwendig sind.

Die administrative Seite des Cloud Computing ist insbesondere bei den Public Clouds ein weiterer Angriffspunkt. In der Regel wird über ein vom Cloud-Anbieter angebotenes Verwaltungsportal die Administration vorgenommen (Speicher/Maschinen hinzufügen oder abbestellen, etc.). Damit ist die administrative Schnittstelle ein lohnendes Angriffsziel, das mit hohen Risiken verbunden ist. Der Cloud User muss unternehmensintern eine klare Trennung der Zugriffsrechte durchsetzen, sodass nur autorisierte Mitarbeiter Änderungen an der Cloud-Plattform vornehmen können. Andererseits muss der Cloud Provider auch Schutzmaßnahmen ergreifen, um die Verwaltungsportale vor Angriffen zu schützen.

Gerade aus wirtschaftlicher Sicht ist die oben angesprochene Angriffsmöglichkeit auf administrative Verwaltungsportale problematisch, da beim Cloud Computing nach einem *Payper-Use* Modell abgerechnet wird. Das bedeutet aber auch, dass generell eine nicht-autorisierte Nutzung einen ökonomischen Schaden verursacht. Das ist beispielsweise bei Dateien gegeben, die unberechtigt geladen werden, aber eigentlich vom System nur gegen Entgelt übers Internet angeboten werden.

In diesem Abschnitt wollen wir uns mit den Risiken für den Lösungsanbieter von IT-Diensten in der Cloud beschäftigen und neben der Auflistung der Probleme auch Lösungsmöglichkeiten vorschlagen. Dabei wird die Thematik der Sicherheit aus der Sicht der Unternehmenssicherheit bei der Nutzung des Cloud-Modells (Verfügbarkeit, Datenschutz, Informationssicherheit) betrachtet.

Damit soll dem Leser ein umfassendes Bild über die Sicherheitsbedenken im Cloud Computing aufseiten des Cloud Users gegeben werden.

2.2.1 Compliance

Unter *IT-Compliance* wird die Einhaltung gesetzlicher und verträglicher Regelungen verstanden, die sich insbesondere auf die Richtlinien des Datenschutzes und -aufbewahrung, Informationssicherheit sowie Verfügbarkeit beziehen. Unternehmen müssen solchen Verpflichtungen, die auch oft landesspezifische Unterschiede aufweisen, nachgehen, denn bei Nichteinhaltung drohen hohe Geldstrafen. Die Studie zur IT-Sicherheit und IT-Compliance [10] von ibi research an der Universität Regensburg unterstreicht die Relevanz der Compliance- und Sicherheitsanforderungen für Unternehmen

[1] Ein Beispiel hiervon lässt sich beim Smartphone „Sidekick“, das von T-Mobile in den USA angeboten wurde, feststellen. Besitzer des Smartphones konnten persönliche Daten im Internet speichern, die nach einem Problem in der zugehörigen Cloud-Lösung größtenteils unwiederbringlich verloren wurden [2].

doi: 10.2313/NET-2011-05-2_10

und zeigt, dass beide Themen auch aktuell einen hohen Stellenwert genießen. Gerade in der Cloud wird aber die Einhaltung der Compliance-Regelungen und Vorgaben für ein Unternehmen erschwert.

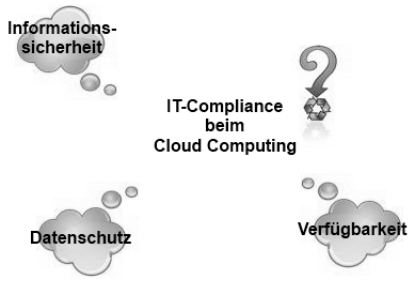


Abbildung 4: Aspekte der IT-Compliance in der Cloud.

Wie werden datenschutzrechtliche Anforderungen vom Cloud-Anbieter erfüllt und wie lässt sich ein verantwortungsvoller Umgang mit IT-Compliance sicherstellen? Gerade diese Fragen beschäftigen viele Unternehmen, die einen Umstieg auf das Cloud-Modell erwägen. Das liegt in erster Linie daran, dass die Verantwortung für personenbezogene Daten auch bei der Nutzung der Cloud-Services nach dem Bundesdatenschutzgesetz beim Cloud User als Auftraggeber liegt. Deshalb sind hier vertrauensvolle Cloud Anbieter wichtig, die datenschutzkonforme Cloud-Services transparent anbieten, so dass seitens des Cloud Users die Einhaltung der gesetzlichen Regelungen überprüft werden kann.

Für kleine bis mittelgroße Unternehmen wird aufgrund der IT-Compliance der Umstieg auf das Cloud-Modell plötzlich problematisch. Sie haben sich in der Regel soweit nur geringfügig mit den gesetzlichen Anforderungen auseinandergesetzt und haben diesbezüglich auch keine Strategie entwickelt. Beim Cloud Computing müssen sie sich aber mit diesem zentralen Thema auseinandersetzen. Anders ist es bei Großunternehmen, die in vielen Ländern vertreten sind und deshalb bereits Strategien zur Einhaltung der Compliance-Vorgaben erarbeitet haben.

In den folgenden Abschnitten werden wir die Aspekte der vertraglichen Regelungen der IT-Compliance näher vorstellen und aus der Sicht des Cloud Users diskutieren.

2.2.2 *Verfügbarkeit*

Hohe Verfügbarkeit und Stabilität der IT-Dienste ist für ein Unternehmen eine Grundvoraussetzung, um einen erfolgreichen Geschäftsbetrieb zu ermöglichen. Bereits der Ausfall von Teilbereichen der IT-Infrastruktur kann zu großen Beeinträchtigungen führen und im größeren Ausmaß sogar existenzbedrohend sein. Das gilt für Klein- als auch insbesondere für Großunternehmen. Damit sind die Anforderungen an Verfügbarkeit vor allem bei Cloud-Modellen seitens der Cloud User entsprechend hoch. Problematisch wird das beim Cloud-Modell gerade aufgrund der Erreichbarkeit der Dienste und der Daten von überall und zu jeder Zeit über das Internet. Hochverfügbarkeit ist beim Cloud Computing damit ein sehr herausforderndes Thema, schließlich sind die meisten Cloud-Systeme vielen Internet-Attacken (beispielsweise *Denial of Service*) ausgesetzt. Die Sicherstellung der Verfügbarkeit liegt hierbei in erster Linie beim Cloud Anbieter, die sich allerdings vertraglich in der Regel dazu nicht binden lassen. Vor allem in Public Clouds sind Ausfälle möglich.

Beispiel: Google Mail - Ausfälle

Google bietet mit der Cloud-Lösung *Google Apps* Dienste für Klein bis Großunternehmen an. Dazu gehört auch Google Mail, das allerdings in der Vergangenheit Ausfälle verzeichnete [13] und über gewisse Zeitspannen nicht erreichbar war (z.B.: Webinterface ca 100 Minuten [1]). Das kann für Unternehmen sehr problematisch sein, vor allem wenn bei derartigen Ausfällen Emails verloren gehen.

Zur Absicherung des Unternehmens bezüglich der Verfügbarkeit von IT-Diensten muss in erster Linie ein vertrauenswürdiger Cloud Anbieter gewählt werden, der einen hohen Stellenwert bei den Kriterien Stabilität und Verfügbarkeit aufweist. Allerdings reicht das insbesondere bei größeren Unternehmen in der Regel nicht aus, denn technische Mängel sind in der Cloud sehr wohl möglich, nicht erreichbare IT-Dienste können aber existenzbedrohend sein. Daher kann die Erhaltung der Teile eigener Infrastruktur als Lösungsvorschlag genannt werden, die zumindest die wesentlichen IT-Services abdecken. Somit können diese beim Ausfall der Cloud die für das Unternehmen essentiell notwendigen Funktionalitäten anbieten.

2.2.3 *Datenschutz*

Der Schutz persönlicher Daten ist für Unternehmen ein sehr wichtiges Thema. Neben den gesetzlichen Bindungen und den damit bei Verletzungen drohenden Strafen, ist auch ein Imageverlust für die meisten Unternehmen ein großes Problem. Gerade in der Cloud ist die Erfüllung gesetzlicher Vorgaben bezüglich Datenaufbewahrung und -übertragung nicht einfach, denn die Lokalität der einzelnen Server kann im Cloud-Serverpool nicht ohne weiteres festgestellt werden.

Beispiel: Datenaufbewahrung- und übertragung in Deutschland [6]

Betrachten wir beispielsweise die Vorgaben zur Datenaufbewahrung und -übertragung in Deutschland (Bundesdatenschutzgesetz), so sind Unternehmen zwingend verpflichtet, persönliche Daten der Bundesbürger auf Server zu übertragen oder speichern, die sich im Gültigkeitsbereich der deutschen Rechtsprechung befinden. Gerade in der Cloud ist das aber nicht unproblematisch, denn hier bilden viele Server einen einzelnen Server-Pool (die Cloud) und die Lokalität einzelner Server kann sehr unterschiedlich sein. Damit könnten tatsächlich persönliche Daten auf Server gelangen, die nicht den Vorlagen der IT-Compliance in Deutschland entsprechen. Die Einforderung der Einhaltung von Compliance-Regeln seitens des Cloud Anbieters ist nicht einfach, schließlich handelt es sich beim Cloud Computing um eine Form des Auslagerns und dieses ist immer mit juristischen Tücken verbunden.

---

doi: 10.2313/NET-2011-05-2_10

Zur Lösung der obigen Problematik stehen Unternehmen unter anderem die folgenden Möglichkeiten zur Verfügung:

1. Nutzung von Private Clouds: Aufgrund der Exklusivität der Benutzung einer Cloud von einem einzelnen Unternehmen lassen sich Regelungen besser verhandeln als bei Public Clouds. Somit können Unternehmen bei vertrauenswürdigen Cloud Anbietern auch IT-Compliance Regelungen durchsetzen und sich vertraglich zusichern lassen.

2. Vertrauenswürde Public Cloud Anbieter: Auch bei Public Clouds gibt es Anbieter, die landesspezifisch die IT-Compliance Vorgaben erfüllen können. Allerdings verlangt die Nutzung der Angebote solcher Anbieter auch nach einem gewissen Grad an Vertrauen und nach einer entsprechenden Transparenz beim Anbieter, so dass die compliancegerechte Datenspeicherung überprüft werden kann.

3. Unternehmerspezifische Lösung: Eine weitere Lösungsmöglichkeit der Problematik kann in der Auslagerung der Datenspeicherung auf ein spezifisch hierfür entwickeltes Eigensystem des Unternehmens thematisiert werden. Allerdings ist das zwangsläufig mit dem Verlust der Cloud Vorteile verbunden.

2.2.4 Informationssicherheit

Die Informationssicherheit beschäftigt sich mit dem Schutz von Systemen, die Information verarbeiten oder speichern. Dabei soll die Integrität, die Vertraulichkeit und die Verfügbarkeit der Daten sichergestellt werden. Im Rahmen des Cloud Computings ergeben sich gegenüber konventionellen Systemlösungen weitere Risiken. So nehmen in der Cloud neben den üblichen Mensch-Maschine Kommunikationen zusätzlich Maschine-Maschine Transaktionen zu. Die Feststellung der Anwenderidentität muss hierbei sichergestellt werden, um nicht zulässige IT-Transaktionen auszuschließen. Problematisch wird das durch die Automatisierung der Transaktionen in der Cloud. Eine Maschine kann nämlich auch für einen Menschen agieren, so dass der Authentifizierung und der Verwaltung von Identitäten eine große Wichtigkeit zukommt.

Auch der Weg des Nachrichtenaustauschs muss abgesichert werden. In einem Mensch-Maschine Szenarium ist das über das Internet beispielsweise mittels Verschlüsselung und der richtigen Benutzung entsprechender Protokolle durchführbar. In der Cloud selbst, bei Maschine-Maschine Transaktionen, muss die Sicherheit durch den Cloud Provider hergestellt werden.

Die Auslagerung von Diensten beziehungsweise von Teilen der IT-Infrastruktur bedeutet gleichzeitig, dass sich Teile der Unternehmensdienste und der damit verbundenen Daten im Besitz Dritter befinden. Somit müssen aufseiten des Cloud Users insbesondere bei vertraulichen Daten Verschlüsselungsalgorithmen angewandt werden, um die Daten zu schützen. Das ist insbesondere bei sensiblen Daten wichtig, die zudem auch separiert werden sollten. Gleichzeig ist beim Cloud Anbieter sicherzustellen, dass der physikalische und der virtuelle Raum geschützt ist und keine Zugriffe seitens benachbarter Systeme möglich sind.

Der Cloud User muss sich bei der Wahl des Cloud Anbieters zudem vergewissern, dass dieser die Standards zur Informationssicherheit einhält. Das kann beispielsweise über den Nachweis von Zertifizierungen erfolgen.

2.3 Risiken bei der Cloud Storage

IT-Anwendungen sowie ihre Benutzer produzieren immer mehr Daten, die geschützt, archiviert und zur Weiterverwendung zur Verfügung gestellt werden müssen. Der Datenzuwachs wird für viele Unternehmen immer problematischer, so dass eine kostengünstige und skalierbare Lösung notwendig ist. Genau in diesem Zusammenhang kommt die *Cloud Storage* ins Spiel. Initial für die Archivierung und Backups gedacht, bietet Cloud Storage mittlerweile sehr viele Formen der Datenspeicherung. So können beispielsweise Datenbanksysteme auch in der Cloud genutzt werden oder aber die Datensicherung in rein tabellarischer Form erfolgen. Für Unternehmen ergeben sich durch die kostengünstige und skalierbare Form der Datenhaltung beim Cloud Storage enorme Vorteile. Sie können durch die Cloud-Nutzung ihre Daten über ein hochverfügbares System anbieten und müssen dafür nur einen Bruchteil des Preises einer traditionellen Lösung bezahlen.

In diesem Abschnitt wollen wir uns mit der Sicherheit von Cloud Storage aus der programmiertechnischen Sicht auseinandersetzen. Generelle Sicherheitsbedenken zum Datenschutz und Verfügbarkeit sowie der Informationssicherheit können auch bei Cloud Storage angewandt werden, so dass hier darauf nicht weiter eingangen wird. Als Beispiele für prominente Cloud Storage Systeme können *Microsoft Windows Azure Storage Services* und *Amazon Simple Storage Service* (S3) genannt werden, anhand derer die Analyse der Sicherheit geführt werden soll. Hierbei beziehen wir uns vor allem auf die Lösung von Microsoft [9], Details zu der Cloud Storage von Amazon können entsprechend auch [9] entnommen werden.

Microsoft Windows Azure Storage. Microsoft bietet mit den Windows Azure Storage Services mehrere Möglichkeiten zur persistenten Speicherung von Daten in einem Cloud-System an:

- *Blob Service*: *Binary Large Objects* (Blobs) sind binäre Objekte, die im Rahmen der Blob Storage gespeichert werden können. Dabei kann es sich beispielsweise um Bilder, Videos oder Musikdateien handeln. Die Blobs können öffentlich zugängig sein oder aber auch nur bestimmten autorisierten Benutzern vorbehalten bleiben. Für jedes Blob ist eine maximale Größe von 1TB festgelegt.

- *Table Service*: Table Services ermöglichen es Anwendungsbesitzern frei-formatierte Daten in Tabellen abzuspeichern. Dabei kann jeder Eintrag mehrere Eigenschaften aufweisen, die zwischen den Einträgen auch unterschiedlich sein können. Die so gespeicherten Daten liegen zwar in tabellarische Form vor, sie sind allerdings nicht mit relationalen Datenbanktabellen zu verwechseln. Konzepte wie *Foreign Keys* sind bei Table Services nicht vorhanden. Die Tabellen selbst können

doi: 10.2313/NET-2011-05-2_10

sehr groß werden (Billionen von Zeilen) und mehrere Terrabyte an Daten beinhalten.

- *Queue Service*: Die Queue Services werden in der Regel für die Kommunikation zwischen Anwendungen genutzt. So können hierbei Nachrichten hinzugefügt und gelesen werden.
- *SQL Azure*: SQL Azure ermöglicht den Zugriff auf eine relationale Datenbank in der Cloud, die über die meiste Funktionalität einer Datenbanklösung wie Microsoft SQL verfügt. Die Kommunikation mit SQL Azure erfolgt über das Protokoll TCP, so dass Software, die zur Verwaltung oder zum Zugriff sowie zur Kommunikation mit der Microsoft SQL-Plattform geschrieben wurde, hier genauso für SQL Azure verwendet werden kann.

Bei der Betrachtung der Sicherheit sowie möglicher Angriffsszenarien beziehen wir uns vor allem auf Blob und Table Services. Für SQL Azure gelten dieselben Sicherheitsbedenken wie bei klassischen Datenbanken, so dass wir sie in diesem Artikel nicht weiter behandeln.

Zugriffs-API. Zum Zugriff auf die Daten der Blob, Table und Queue Services wird die so genannte *REST API* verwendet. Dabei sind die Schreib- und Lesevorgänge einfache POST/PUT und GET-Requests. So kann ein öffentlich zugängliches Blob wie folgt gelesen werden:

```
GET http://accountname.blob.core.windows.net/
    containername/blobname?snapshot=datetime
    HTTP/1.1
```

Ein Lesezugriff auf eine Tabelle kann wie folgt erfolgen:

```
GET http://accountname.table.core.windows.net/
    tablename(PartitionKey='partitionkey',
    RowKey='rowkey') HTTP/1.1
```

Dabei werden die gefundenen Daten in der Tabelle im XML-Format zurückgegeben. Bei geschützten Blobs oder Tabellen sind zusätzliche HTTP-Header notwendig, die die Authentizität sicherstellen.

Zum Schreiben von Blobs oder Tabellen werden PUT-Befehle verwendet. Eine vereinfachte Darstellung einer Schreiboperation bei Blobs sieht wie folgt aus (eine genauere Darstellung kann [9] entnommen werden):

```
PUT http://myaccount.blob.core.windows.net/
    mycontainer/myblockblob HTTP/1.1
*HTTP Headers*
*Content*
```

Bei Tabellen sieht der strukturelle Aufbau von Schreibnachrichten wie folgt aus (siehe auch [9])

```
POST http://myaccount.table.core.windows.net/
    GuestBookEntry HTTP/1.1
*HTTP Headers*
*XML-Content*
```

Neben der REST API kann bei Azure Storage auch eine .NET plattformspezifische Bibliothek verwendet werden (Teil der *Windows Azure SDK*). Durch die Verwendung der .NET API verringern sich viele programmiertechnische Risiken, so dass wir uns im Folgenden bei der Analyse der Sicherheit vor allem auf die REST API konzentrieren.

2.3.1 Klassische Datenbankattacken

In einer Webapplikation ist die eigentliche Datenbank in der Regel für einen Angreifer nicht direkt erreichbar und wird durch die Wahl der Architektur somit geschützt. Allerdings werden Datenbanken über Nachrichten mit Benutzereingaben angesteuert, so dass ein Angriff durch eine sinnvolle Manipulation solcher Eingaben durchaus geschehen kann. Die wohl bekannte Angriffsweise, die genau dieses Prinzip ausnutzt, ist *SQL Injection*. Eine weitere Methode, *XML Injection*, nutzt Mechanismen von XML-Parsern aus, um Nachrichtenein- und -ausgaben zu manipulieren. Beide Angriffsweisen sind nicht cloudspezifisch, können aber in der Cloud sehr wohl auftreten. Weitere Information zu SQL und XML Injection können [9] entnommen werden.

2.3.2 Bedrohungen von Cloud Systemen

In diesem Abschnitt wollen wir Angriffsmöglichkeiten [9] auf Systeme näher diskutieren, die zur Datenspeicherung Azure Storage Services verwenden. Dabei werden, bedingt durch die Nutzung von XML durch die REST API, XML-basierende Attacken thematisiert. Hierzu definieren wir uns zunächst ein Beispiel, anhand dessen die einzelnen Attacken näher erläutert werden können:

Betrachten wir beispielsweise eine einfache MP3-Verkaufsplattform, so können dort beim Kauf einer Musikdatei die Lieferadresse sowie ein Kommentar hinterlassen werden. In Azure Table Store könnte die Schreiboperation eine strukturierte Nachricht wie folgt verwenden:

```
<content type="application/xml">
 <m:properties>
  <d:Address>Meine Adresse...</d:Address>
  <d:UserID>523</d:UserID>
  <d:FileID>123231</d:FileID>
  <d:Comment>Mein Kommentar...</d:Comment>
  ***weitere Felder***
 </m:properties>
</content>
```

Folgend versuchen wir einige Attacken auf diesen einfachen MP3-Shop anzuwenden.

Direct Node Injection. Diese Attacke greift die Operationen der REST API an in der Annahme, dass zum Verarbeiten von XML Sax-Parser eingesetzt werden, die den XML-Datenstrom sequentiell abarbeiten. Welche Parsertypen von Cloud Storage Systemen tatsächlich verwendet werden, ist in der Regel (wie z.B. bei Microsoft oder Amazon) nicht bekannt. Bei der *Direct Node Injection* wird nun versucht, Daten der Eingabefelder zu manipulieren, um beispielsweise ein anderes Verhalten zu erreichen.

Im obigen Beispiel des MP3-Shops ist das Kommentarfeld der Angriffspunkt. Ein manipulierter Kommentarwert wie

doi: 10.2313/NET-2011-05-2_10

```
</d:Comment><d:UserID>100</d:UserID><d:Comment>
    Mein Kommentar...
```

würde die ursprüngliche Nachricht wie folgt verändern

```
<content type="application/xml">
 <m:properties>
  <d:Address>Meine Adresse...</d:Address>
  <d:UserID>523</d:UserID>
  <d:FileID>123231</d:FileID>
  <d:Comment></d:Comment><d:UserID>100</d:
      UserID><d:Comment>Mein Kommentar...</d:
      Comment>
 </m:properties>
</content>
```

und damit den Kauf einer MP3 als Benutzer mit der ID 100 ausführen. Es ist klar, dass diese Attacke in der Regel blind ausgeführt werden muss, denn die einzelnen Datenstrukturen einer Applikation sind im Normalfall nicht bekannt. In der Praxis wurden jedoch solche Attacken auch schon erfolgreich durchgeführt. Bei Windows Azure funktioniert die *Direct Node Injection*-Attacke nicht, es wird ein „400 Bad Request“ als Antwort zurückgegeben, was auf einen DOM-Parser zur Verarbeitung von XML schließen lässt. Dabei ist allerdings nicht zwingend gegeben, dass ein DOM-Parser auf der Cloud Provider-Seite zum Einsatz kommt. Cloud Provider geben in der Regel nicht ohne weiteres bekannt, welche Parsertypen sie verwenden.

CDATA and XML Comment Injection. Diese Attacke greift XML-Parser an, die über das volle XML-Verständnis verfügen und somit auch mit Kommentaren und CDATA-Feldern umgehen können (beispielsweise DOM-Parser). Hierbei wird versucht, zwei manipulierbare Felder so zu füllen, dass sie die dazwischen vorkommenden Felder auskommentieren. Im MP3-Shop Beispiel könnte dazu das Adressfeld mit

```
Meine Adresse...</d:Address><!--
```

und das Kommentarfeld mit

```
--><d:UserID>200</d:UserID><d:FileID>123231</d:
    FileID><d:Comment>Mein Kommentar...
```

gefüllt werden, um somit das folgende XML zu erhalten:

```
<content type="application/xml">
 <m:properties>
  <d:Address>Meine Adresse...</d:Address><!--</
      d:Address>
  <d:UserID>523</d:UserID>
  <d:FileID>123231</d:FileID>
  <d:Comment>--><d:UserID>200</d:UserID><d:
      FileID>123231</d:FileID><d:Comment>Mein
      Kommentar...</d:Comment>
 </m:properties>
</content>
```

Damit können nun für User sowie File beliebige IDs und Lieferadressen festgelegt werden. Diese Attacke *funktioniert* nach [9] bei Azure Storage.

Ein ähnlicher Angriff kann auch mittels des CDATA-Tags formuliert werden, wie in [9] thematisiert wird.

Enumeration. Im MP3-Shop Beispiel haben wir soweit programmiertechnische Lücken im XML-Parser des Cloud Anbieters untersucht. Eine genauso interessante Sicherheitsproblematik ergibt sich beim Anbieten von Dateien für Shop-Benutzer, die nicht jedem zugängig sein sollen und nur von Benutzern geladen werden dürfen, die auch tatsächlich die entsprechenden Dateien erworben haben. Diese Sicherheitsthematik beschränkt sich natürlich nicht nur auf unseren MP3-Shop, sondern ist für jeden Anbieter, der Content (MP3, Ebooks, Videos, ...) kostenpflichtig übers Internet anbietet relevant. Cloud Storage-Systeme bieten eine sehr kostengünstige Art und Weise Content übers Internet anzubieten, doch wie ist es um die Zugriffsicherheit beim Content bestellt?

Zunächst gibt es natürlich die Thematik der Sicherheit durch Verborgenheit. Dateien erhalten sehr lange Namen (bspw. Guids) und entsprechend auch lange Pfade, so dass die einzelnen Dateien nicht von jedermann erraten werden können. Das Problem hierbei liegt allerdings in der illegalen Verteilung solcher Pfade, die das System sehr gefährden kann. Weiterhin kann bei Containern (Bucket bei Amazon S3), die die Blobs und somit den Content enthalten, auch der gesamte Inhalt anhand seines Namens ausgelesen werden, was bei Amazon S3 mit dem so genannten *S3 Ripper* nach [9] tatsächlich möglich ist. Bei Windows Azure hängt es hingegen von den vergebenen Zugriffsrechten auf Container und Blobs ab, so dass im schlimmsten Fall hier eine Enumeration des Inhalt durch einen einfachen Browserbefehl möglich ist[2].

Zur Lösung der oben beschriebenen Sicherheitsproblematiken können unterschiedliche Mechanismen eingesetzt werden. Empfehlenswert nach [9] sind die so genannten *Shared Access Signatures* in Windows Azure und die *Signed URLs* in Amazon S3. Das Prinzip dahinter liegt in der Verwendung einer Zugriffssignatur. Dabei kann festgelegt werden, wie lange ein Zugriff möglich ist und welche Zugriffsrechte erlaubt sind. Über einen *Access Token* wird die Richtigkeit der Daten einer Signatur abgesichert. Im MP3-Shop Beispiel könnten wir so Downloads von Dateien nur für eine Stunde zulassen, so dass hier beim Verteilen der Links nur ein sehr begrenzter Schaden entstehen würde.

2.4 Risiken beim Cloud Provider

Der Cloud Provider bietet Benutzern vordefinierte Dienste nach dem Cloud-Modell an. Sind Sicherheitsprobleme bereits beim Cloud Provider vorhanden, so sind sie gleichermaßen für alle Benutzer aktuell. Hier muss aufseiten des Cloud Providers Aufwand investiert werden, um Probleme von vornherein zu vermeiden. Die folgenden Risiken können unter Anderem auftreten [14]:

Host Zugriff auf die Daten/Applikationen benachbarter Benutzer; Denial of Service; fehlerhafter Ressourcenzuweisung; Zugriffe auf den Host seitens externer Angreifer sowie durch andere Anbieter.

Netz Klassische netzbasierte Angriffe; verteilte Denial-of-Service Angriffe.

Virtualisierung Bedrohung von Privacy durch Verschiebung von Maschinen oder durch Datenreplica; Fehler-

[2]Enumerationen werden durch die REST API unterstützt.

doi: 10.2313/NET-2011-05-2_10

hafte Konfiguration und Sicherheitslücken in der Virtualisierungslösung.

3. ZUSAMMENFASSUNG

Cloud Computing bietet Unternehmen und Lösungsanbietern vielfältige Chancen die Wettbewerbsfähigkeit zu steigern, indem Innovationszeiträume verkürzt, neuartige Geschäftsmodelle ermöglicht und die generelle Wirtschaftlichkeit der eigenen IT-Systeme erhöht wird. Allerdings ist die Nutzung von Cloud Computing mit einer Vielzahl von Sicherheitsrisiken verbunden. So müssen die Sicherheitsproblematiken der IT-Compliance wahrgenommen und die Bedrohungen der Privatheit, Vertraulichkeit, Verfügbarkeit und Integrität entsprechend beachtet werden. Die Wahl des Cloud Providers gestaltet sich hierbei als nicht unproblematisch und muss deshalb wohlüberlegt getroffen werden. Schließlich ist die Auslagerung von IT-Architektur oder die Verlagerung von IT-Abläufen in die Cloud immer mit der Übergabe von Daten/Systemen in Besitz Dritter verbunden und sollte somit mit äußerster Vorsicht vonstattengehen.

Zur Minderung der Vorbehalte gegenüber der Sicherheit beim Cloud Computing wären Standardisierungsmaßnahmen von Vorteil, die cloudspezifisch ausgelegt neben der Risikominimierung auch das Vertrauen in die Cloud-Plattform steigern würden. Damit könnten derzeit zögernde Benutzer umgestimmt sowie weitere potentielle Cloud User gewonnen werden.

4. LITERATUR

[1] Google Mail - Ausfall statt höherer Verfügbarkeit. Golem Online: http://www.golem.de/0909/69510.html, September 2009.

[2] T-mobile, microsoft und das sidekick-desaster. Stern Online: http://www.stern.de/digital/online/cloud-computing-t-mobile-microsoft-und-das-sidekick-desaster-1514865.html, Oktober 2009.

[3] A Joint Report of the US Department of Homeland Security, SRI International Identity Theft Technology Council, and the Anti-Phishing Working Group. The crimeware landscape: Malware, phishing, identity theft and beyond. Online: http://www.antiphishing.org/reports/APWG_CrimewareReport.pdf, Oktober 2006.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. UC Berkeley Reliable Adaptive Distributed Systems Laboratory: http://radlab.cs.berkeley.edu, Februar 2009.

[5] Avanade. Globale Avanade-Studie zeigt: Sicherheitsbedenken beim Cloud Computing bremsen Einzug der Technologie in Unternehmen - trotz wirtschaftlicher Vorteile. Online: http://www.avanade.com/de-de/about/avanade-news/press-releases/Documents/avanadecloudcomputingg250209877376.pdf, Februar 2009.

[6] E. Baize, R. Cloutier, B. Hartman, S. Herrod, C. Hollis, U. Rivner, and B. Verghese. Cloud Computing mit Sicherheit - Best Practices für vertrauenswürdige Umgebungen. RSA Serurity Brief, Online: http://www.rsa.com/innovation/docs/10764_CLWD_BRF_1009_DE.pdf, November 2009.

[7] BITKOM. Cloud Computing - Evolution in der Technik, Revolution im Business. Online http://www.bitkom.org/files/documents/BITKOM-Leitfaden-CloudComputing_Web.pdf, Oktober 2009. Seite 15.

[8] BSI. BSI-Mindestsicherheitsanforderungen an Cloud-Computing-Anbieter. Online: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Sonstige/Cloud_Computing_Mindestsicherheitsanforderungen.pdf?__blob=publicationFile, Oktober 2010.

[9] G. Bugher. Secure Use Of Cloud Storage. Online: http://media.blackhat.com/bh-us-10/whitepapers/Bugher/BlackHat-USA-2010-Bugher-Secure-Use-of-Cloud-Storage-wp.pdf, July 2010.

[10] ibi research. Vorstellung der Studienergebnisse: IT-Sicherheitsstandards und ITCompliance 2010. Online: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Veranstaltungen/3GS_Tag2010/IBI_Kronschnabel.pdf?__blob=publicationFile, Oktober 2010.

[11] IDC. IDC Enterprise Panel, August 2008. n=244.

[12] G. Kaefer. Cloud Computing Architecture. 4th Generation Datacenter IEEE Spectrum, http://www.sei.cmu.edu/library/assets/presentations/Cloud%20Computing%20Architecture%20-%20Gerald%20Kaefer.pdf, Februar 2009.

[13] T. Kleinz. Google Mail für Stunden offline. Focus Online: http://www.focus.de/digital/internet/google/it-ausfall-google-mail-fuer-stunden-offline_aid_374310.html, Dezember 2009.

[14] W. Streitberger and A. Ruppel. Cloud-Computing - eine Herausforderung für die Sicherheit, September 2009.

Clickjacking-Angriff auf Webseiten

Gel Han
Betreuer: Dr. Heiko Niedermayer
Hauptseminar Innovative Internet-Technologien und Mobilkommunikation WS2010/2011
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: hang@in.tum.de

KURZFASSUNG

Das Internet ist heutzutage ein sehr wichtiges Medium zum Austausch von Daten. Mit immer mehr Zugang zu sensiblen Daten steigt gleichzeitig die Anzahl der Angriffsmöglichkeiten im Internet. *Clickjacking* ist eine dieser Möglichkeiten, um einen Nutzer des Internets anzugreifen. Der Angreifer täuscht eine Webseite vor, um Aktionen des Benutzers (wie zum Beispiel Maus-Klicks) abzufangen. *NoScript* oder *ClickIDS* stellen zwei Abwehrmöglichkeiten dar, um sich vor Clickjacking zu schützen. Es gibt jedoch Browser-spezifische Probleme, die weiterhin ungelöst sind. Clickjacking wird zurzeit noch erfolgreich erkannt und geblockt. Sicherheitsprogramme müssen dennoch weiterhin verbessert werden, da Clickjacking in Zukunft ein ernstes Thema bezüglich Sicherheit im Internet sein wird.

Schlüsselworte

Clickjacking, Transparenter `iframe`, Framebusting, NoScript, ClearClick, ClickIDS, `X-FRAME-OPTIONS` Header

1. EINLEITUNG

Das Internet hat sich zu einer Plattform entwickelt, auf der sehr viele persönliche und auch sensible Daten im Umlauf sind. Soziale Netzwerke wie zum Beispiel *Twitter* oder *Facebook* werden immer populärer und gleichzeitig gibt es immer mehr Angriffsziele für einen Hacker. Heutzutage kommen Computer nicht ohne vorinstallierte Antivirenprogramme aus. Weitere Angriffstypen wie Würmer, Trojaner oder eine gezielte Denial of Service-Attacke können einem unwissende Benutzer Schaden hinzufügen. Neben diesen Typen gibt es auch andere Angriffe wie zum Beispiel *Clickjacking* (kurz CJ). Alles was der Angreifer braucht, um sein Opfer anzulocken, ist eine Webseite im Internet. Der Angreifer kann mithilfe eines CJ-Angriffs das Opfer dazu bringen, durch Maus-Klicks einen Account zu löschen, eine Ware zu kaufen oder seine Webcam einzuschalten. Der Benutzer hat dabei keine Ahnung was im Hintergrund geschieht, da alles versteckt abläuft. Bei den sechs Sicherheitszielen kann Clickjacking die Integrität der Daten, die Vertraulichkeit, die Verfügbarkeit, die Verbindlichkeit, sowie die Privatheit verletzen.

Im nächsten Kapitel werden wir etwas genauer auf den Begriff "Clickjacking" und die Vorgehensweise des Angriffs eingehen. Anhand von Beispielen werden erfolgreiche CJ-Angriffe vorgestellt. Um Schwachstellen gegenüber CJ zu beseitigen, wurden Abwehrmethoden entwickelt, die im darauf folgenden Kapitel erklärt werden. Zuletzt folgt ein Überblick über verwandte Arbeiten sowie eine kurze Zusammenfassung und ein Ausblick auf die Zukunft von Clickjacking.

2. DER BEGRIFF "CLICKJACKING"

Der Begriff "Clickjacking" tauchte das erste Mal 2008 auf. "Clickjacking" wurde von Robert Hansen[1] und Jeremiah Grossman[2] geprägt. R. Hansen und J. Grossman stellten CJ-Angriffe und Schwachstellen auf der *OWASP NYC AppSec 2008* Konferenz vor (siehe [5]). Zum Entwickeln einer CJ-Attacke braucht der Angreifer die Markup-Sprachen *HTML* (HyperText Markup Language, siehe [6]) und *CSS* (Cascading Style Sheets, siehe [10]). Der Angreifer erstellt eine Webseite, um das unwissende Opfer anzulocken und auf Buttons oder Hyperlinks zu klicken. Dabei ist dem Opfer nicht bewusst, dass die Webseite eine Falle ist. Der Angreifer hat auf der Homepage eine zweite Webseite mittels `iframe` eingebunden und per CSS transparent gemacht. In Abbildung 1 sieht man ein Beispiel für eine Webseite mit einem eingebundenen `iframe` und einer potentiellen CJ-Attacke.
Clickjacking verwendet Elemente und Attribute, die in den Sprachen von HTML und CSS schon vorhanden sind. `iframes` werden mittels HTML in bestehende, normale Webseiten eingebunden. Um diese dann vor dem unwissenden Opfer zu verstecken, verwendet man CSS. Es gibt das Attribut `opacity`, um Objekte transparent zu machen. Der Benutzer denkt, dass er/sie sich auf einer ganz normalen Homepage befindet und auf Hyperlinks klickt. In Wirklichkeit liegt eine zweite Homepage über der angezeigten Webseite und die Klicks werden alle darauf registriert. In dem folgenden Listing sieht man den HTML-Quellcode, den der Angreifer braucht, um Facebook in einem transparenten `iframe` einzubinden:

```
<iframe src="http://www.facebook.com/home.php?"
     id="frame1" style="opacity:0.0;filter:
    alpha(opacity=0);" width="100%" height="
    100%"/></iframe>
```

[1]Robert Hansen ist CEO von der Sicherheitsconsulting-Firma "SecTheory". Er ist vor allem für seinen Beitrag zum Buch "XSS Attacks: Cross Site Scripting Exploits and Defense" und seiner Tätigkeit in der Hacker- und Sicherheitsszene bekannt.
[2]Jeremiah Grossman ist der Gründer und CTO von der Website-Security Firma "WhiteHat Security". Er gilt als einer der bekanntesten Experten im Bereich "Webapplication Security" und ist neben R. Hansen auch Co-Autor von "XSS Attacks: Cross Site Scripting Exploits and Defense".

doi: 10.2313/NET-2011-05-2_11

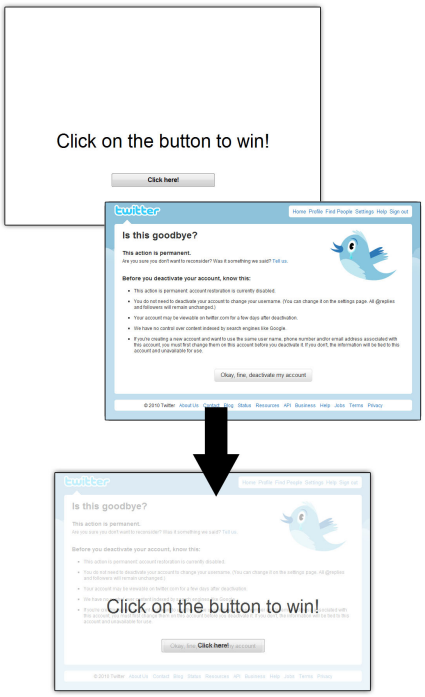


Abbildung 1: Der Angreifer baut sich den Clickjacking-Angriff aus zwei Webseiten zusammen: einer falschen, vom Angreifer erstellten HTML-Seite, die der unwissende Benutzer zu Gesicht bekommt. Die zweite Webseite wird in einem `iframe` eingebunden und als Angriffsziel genutzt. Im dritten Bild sieht man, dass der `iframe` mittels CSS transparent gemacht wurde.

Der CJ-Angriff wurde unter anderem durch Vorfälle auf dem Mikroblogging-Dienst *Twitter* (siehe [13]) und der Social Networking-Webseite *Facebook* bekannt. Im Fall von Twitter haben Angreifer eine Webseite mit einem Button erstellt. Bei einem Klick auf den Button wurde der Twitter-Status des Opfers aktualisiert. Diese CJ-Attacke funktioniert jedoch nur, wenn der Benutzer bei Twitter registriert und gerade angemeldet ist. Die neue Status-Nachricht liest sich wie folgt: "Don't Click: `http://xyz`". Wenn einer der "Followers" (Personen, die Twitter-Nachrichten des Opfers lesen) auf diese URL klickt, so gelangt derjenige auf die Webseite des Angreifers. Die Webseite des Angreifers hatte in einem transparenten `iframe` die Twitter-Homepage mit dem angemeldeten Benutzer geladen und in der Nachrichten-Box die Nachricht "Don't Click: `http://xyz`" kopiert sowie den "Don't Click"-Button auf den "Nachricht absenden"-Button gelegt. Mitarbeiter von Twitter haben auf dieses Problem reagiert (siehe [15]) und die Schwachstelle mittels JavaScript behoben. Diese Abwehrmethode (mit JavaScript) nennt sich *Framebusting* und wird in Kapitel 3.1 genauer erläutert.

Ein anderes Beispiel, wie ein Angreifer Gebrauch von einer Webseite mit transparentem `iframe` machen könnte, wäre mithilfe von Werbebannern. Der Angreifer erstellt eine Homepage mit einer CJ-Attacke, die das unwissende Opfer mehrmals auf einen Button klicken lässt. Jeder dieser Maus-Klicks geschieht auf dem Werbebanner und bringt somit dem Angreifer Geld. Die Webseite könnte ein Spiel vortäuschen, bei dem der Benutzer so schnell und so oft wie möglich auf einen Button klicken soll um in eine Highscore-Liste eingetragen zu werden oder etwas zu gewinnen.

Auf der OWASP NYC AppSec 2008 Konferenz haben R. Hansen und J. Grossman eine Schwachstelle im Adobe Flash Player angesprochen, die durch Clickjacking ausgenutzt werden konnte, um eine installierte Webcam einzuschalten[3]. Mittlerweile hat Adobe diese Schwachstelle behoben. In der Demo wurde eine Webseite mit einem Spiel erstellt, bei dem der Benutzer auf verschiedene Buttons klicken musste. Bei jedem Klick auf einem der Buttons verschwand dieser und ein neuer tauchte an einer anderen Stelle auf. Bei den meisten Klicks geschah nichts; bei vier Klicks wurden jedoch im Hintergrund Einstellungen des Flash Players verändert (siehe Abbildung 2). Der unwissende Benutzer hat auf diese Weise seine Webcam eingeschalten.

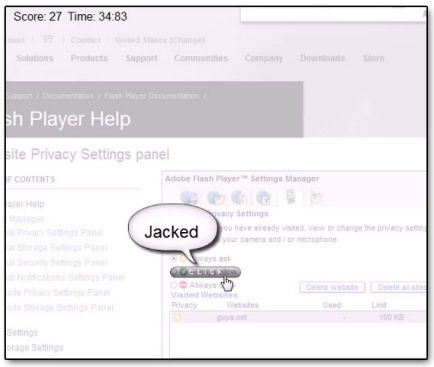


Abbildung 2: Die Abbildung zeigt einen Ausschnitt aus einem Clickjacking-Angriff auf den Adobe Flash Player. Ziel des Angreifer ist es, die Webcam des Benutzers einzuschalten, ohne dass dieser es bemerkt. Quelle: [1].

3. ABWEHRMETHODEN

Grundsätzlich gibt es zwei Möglichkeiten für Abwehrmaßnahmen gegen Clickjacking: 1. Der Benutzer schützt sich mit einem Programm oder 2. die Webseite wehrt Angriffe automatisch mittels Mechanismen ab. Eine Abwehrtechnik, die sehr weit verbreitet ist und in vielen HTML-Seiten angewendet wird, ist *Framebusting*. Diese Technik schützt eine Webseite davor, in einen `iframe` eingebettet zu werden. Framebusting basiert auf JavaScript und ist sehr einfach einzusetzen. Meist genügen ein paar Zeilen Code, um eine Webseite erfolgreich zu schützen. Dennoch gibt es Schwachstellen und Möglichkeiten, Framebusting zu umgehen. Weitere Details werden im Kapitel 3.1 erläutert. In Abschnitt 3.4 gehen wir auf den `X-FRAME-OPTIONS`-Header ein, der auch zur Abwehr von Angriffen wie Clickjacking gedacht ist. Im Anschluss wird das Feature *ClearClick* vom Plugin *NoScript* vorgestellt. NoScript wird hauptsächlich im beliebten Internetbrowser *Firefox* eingesetzt.

[3]Eine Demo des Clickjacking-Angriffs findet man auf `http://guya.net/security/clickjacking/game.html` [09.12.2010]

doi: 10.2313/NET-2011-05-2_11

3.1 Framebusting und Schwachstellen

3.1.1 Framebusting im Allgemeinen

Wie in der Einleitung schon erwähnt wurde, ist *Framebusting* eine Technik, die in Webseiten eingesetzt wird um zu verhindern, dass sie in andere Homepages eingebunden werden. Viele Webseiten, darunter Twitter, Facebook und Yahoo verwenden Framebusting, um sich gegen Clickjacking-Angriffe zu schützen. Der Framebusting-Code von Twitter macht folgendes (siehe Abbildung 1): es wird überprüft, ob das Fenster der Webseite auch das aktuelle Browser-Fenster ganz ausfüllt beziehungsweise das oberste Fenster im Frameset ist. Falls dies nicht der Fall sein sollte, so wurde die Webseite mindestens einmal in einen Frame eingebunden. Der nachfolgende Code ermöglicht es der Webseite, aus einem Frame "auszubrechen" und verhindert somit einen CJ-Angriff (Quelle: `http://www.twitter.com` [09.12.2010]):

```
if (window.top !== window.self) {document.write
    = "";window.top.location = window.self.
    location; setTimeout(function(){document.
    body.innerHTML='';},1);window.self.onload=
    function(evt){document.body.innerHTML
    ='';};}
```

Rydstedt et al. [12] haben im Juli 2010 eine Untersuchung zum Thema "Framebusting" durchgeführt. Es wurden die Top-500 Seiten vom Dienst "Alexa" auf JavaScript-Code untersucht. Dabei wurde festgestellt, dass die meisten Seiten als Abfrage `if (top != self)` und als Statement darauf `top.location = self.location` verwenden. Wie an diesem Beispiel zu sehen ist, besteht ein Framebusting-Code aus zwei Teilen: einem Abfrage-Statement und einer Anweisung, falls ersteres zutrifft. Die Untersuchung lieferte folgendes Ergebnis (für die vollständige Tabelle siehe [12]):

Am meisten verwendete Abfragen für Framebusting:

- `if (top != self)`
- `if (top.location != self.location)`
- `if (top.location != location)`

Am meisten verwendete Statements:

- `top.location = self.location`
- `top.location.href = document.location.href`
- `top.location.href = self.location.href`
- `top.location.replace(self.location)`

`location` ist ein Objekt in JavaScript, das den Standort eines HTML-Dokuments beschreibt. Mit dem Attribut `href` kann man auf die komplette URL zugreifen. In den Statements kommen außerdem die Eigenschaften `top`, `parent` und `self` vor. Diese drei Eigenschaften werden bei einem Verweis auf das jeweilige Browserfenster verwendet. `top` stellt dabei das oberste, `self` das aktuelle und `parent` das übergeordnete Fenster in der Struktur dar. In JavaScript kann man sich mit diesen Eigenschaften durch die verschiedenen Fenster bewegen (jeweils eine Ebene nach oben oder nach unten). Für mehr Informationen zu JavaScript, siehe [3].

Bei der Untersuchung hat sich herausgestellt, dass die meisten Webseiten keinen Framebusting-Code auf ihren Startseiten hatten. Die Abwehrtechnik wurde meist nur bei den "Login"- oder den "Passwort zurücksetzen"-Seiten eingesetzt. In der Ausarbeitung von Rydstedt et al. wird außerdem darauf hingewiesen, dass die meisten Webseiten zwar ihre normale Homepage vor Clickjacking-Angriffen schützen, jedoch aber die mobilen Seiten vernachlässigen. Viele Webseiten bieten dem Benutzer die Möglichkeit, falls dieser über ein mobiles Endgerät verfügen sollte, auf eine mobile Version der Homepage zurückzugreifen. Der Benutzer gelangt jedoch auch mit einem normalen Browser auf die mobilen Seiten. Diese Webseiten bieten meist keinen Schutz vor Clickjacking-Angriffen und können so ein leichtes Ziel von Angreifern werden.

3.1.2 Schwachstellen

In diesem Unterkapitel werden einige Schwachstellen der Framebusting-Technik vorgestellt. Obwohl Framebusting sehr weit verbreitet ist und von Webseiten als *die* Lösung im Bezug auf Clickjacking angesehen wird, ist es sehr einfach, die Abwehrmethoden zu umgehen.

Ein Frame in einem Frame

Nach Rydstedt et al. gibt es Webseiten, die im Framebusting-Code auf Hierarchie prüfen. Dazu wird das Attribut `parent` aus JavaScript verwendet. Eine einfache Methode, diese Technik zu umgehen, ist ein Frame in einem Frame. Ein Beispiel für einen Framebusting-Code, der einen CJ-Angriff verhindern würde:

```
if(top.location != location) {
        parent.location = self.location }
```

Die Webseite würde aus einem `iframe` "ausbrechen" und der Benutzer würde die echte Seite zu Gesicht bekommen (auch bei zwei `iframes`). Falls jedoch dem Attribut `parent.location` etwas anderes zugewiesen wurde und die übergeordnete Webseite schon ein eingebetteter Frame war, so führt der Angreifer den unwissenden Benutzer auf eine zweite, falsche Webseite, auf der der Clickjacking-Angriff stattfindet. Dieser Sachverhalt ist in Abbildung 3 dargestellt.

`onBeforeUnload` und *204 Flushing*

Framebusting-Code führt dazu, dass eine Webseite, die eingebettet wurde, ausbricht und diese anstatt der falschen Webseite des Angreifers angezeigt wird. Diese Weiterleitung kann jedoch vom Angreifer manipuliert werden. Der JavaScript-Handler `onBeforeUnload` wird geladen, wenn der Browser auf die eingebette Seite wechseln will. Wenn der Angreifer dem Benutzer eine Nachricht mit einer Warnung anzeigt und dieser "Abbrechen" klickt, so bleibt das Opfer auf der Webseite mit dem eingebetteten Frame.

doi: 10.2313/NET-2011-05-2_11

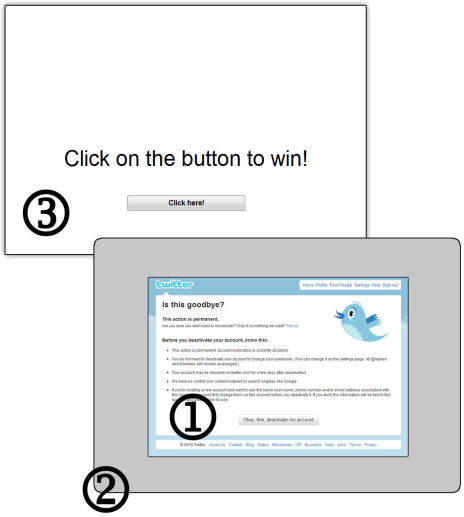


Abbildung 3: Sei (1) die authentische Seite mit Framebusting-Code. Diese erkennt, dass sie in einem Frame eingebunden wurde (2) und "bricht" aus. Die echte Seite (1) wird jedoch in einer weiteren Seite (3) eingebunden und kann somit für einen möglichen Angriff ausgenutzt werden.

Der folgende JavaScript-Code verhindert eine Weiterleitung, falls der Benutzer "Abbrechen" klickt:

```
window.onbeforeunload = function() {
        return ""; }
```

Eine andere Möglichkeit wäre, die Weiterleitung auf eine Webseite zu lenken, die einen *204 No Content*-Fehler zurückliefert. Während `onBeforeUnload` die Interaktion des unwissenden Benutzers erfordert hat, um auf der Webseite des Angreifers zu bleiben, so geschieht mit dieser Methode das "Abschalten" des Framebusting-Codes automatisch. Die Weiterleitung auf die *204 No Content*-Webseite führt dazu, dass alle registrierten Befehle gestrichen werden (daher auch der Name "Flushing"). Somit wird auch die Weiterleitung auf die echte Webseite durch Framebusting ignoriert.

XSS-Filter

Mit XSS-Filtern, die in den Browsern *Google Chrome* und *Microsoft Internet Explorer 8* integriert sind, ist es möglich, Framebusting-Code zu umgehen. Die *reflective XSS Filter* dienen zur Abwehr von XSS-Angriffen. Der Internet Explorer untersucht dabei Abfragen nach möglichen Cross-Site-Scripting-Attacken und falls Übereinstimmungen gefunden werden, so werden diese Abschnitte übersprungen. Dies kann der Angreifer ausnutzen, in dem er im `iframe` einen Teil von einem Framebusting-Code als Übergabe-Parameter angibt. Der XSS-Filter wird beim Überprüfen feststellen, dass sich der Ausschnitt aus dem Code und der JavaScript-Code im Header (der Framebusting-Code zur Abwehr) übereinstimmen. Im Endeffekt wird der übereinstimmende Teil übersprungen. Für den Angreifer bedeutet dies, dass der `iframe` erfolgreich eingebunden werden kann. Während beim Internet Explorer alle *inline*-Scripts übersprungen werden (Teile des JavaScript-Codes und Cookies können trotzdem geladen werden), so kann man bei Chrome explizit Framebusting-Code ausschalten. Ein einfaches Beispiel (aus [12]) würde wie folgt aussehen:

Framebusting-Code:

```
if(top != self) {
        top.location=self.location; }
```

Angreifer:

```
<iframe src="http://www.victim.com/?v=if(top
    +!%3D+self)+%7B+top.location%3Dself.
    location%3B+%7D">
```

Hinter der URL im `iframe` hat der Angreifer den gesamten Framebusting-Code als Übergabeparameter definiert. Chrome stellt eine Übereinstimmung im Code fest und schaltet den JavaScript-Block aus.

Einbetten in Frames

Es gibt Webseiten, die erlauben, dass sie von einer ihrer eigenen Seiten eingebettet werden. Rydstedt et al. haben gezeigt, dass diese Methode bei den meisten Homepages Schwachstellen für einen CJ-Angriff aufweist. Zum Überprüfen, ob eine Webseite von einer bekannten Homepage eingebettet wird, wird der `document.referrer` verwendet. Dieser überprüft, ob die URL, von der die Anfrage zum Einbetten kommt auch mit der aktuellen URL übereinstimmt. In den meisten Fällen wird aber nicht die ganze URL abgeglichen. Es reicht, wenn der Angreifer zum Beispiel eine CJ-Attacke von einer Webseite mit der URL `http://www.walmart.com.badgy.com` startet (siehe [12]). Mit dieser Homepage könnte der Angreifer theoretisch die "Walmart"-Webseite als `iframe` einbinden, da diese zwar auf die URL `walmart.com` überprüft, jedoch nicht den Rest der URL.

Angenommen, eine Webseite wäre gegen Clickjacking-Angriffe geschützt und würde nur vertrauenswürdigen Seiten erlauben, sich selbst einzubetten. Eine einfache Schwachstelle, die ein Angreifer in dem Fall ausnutzen könnte, wäre das Verwenden des Dienstes *Google Images*. Der Angreifer sucht nach einem bestimmten Bild bei Google Images. Die gefundene Seite wird in einem Subframe angezeigt. Die meisten Webseiten vertrauen Google Images und lassen das Einbetten in den Frame zu (siehe Abbildung 4). Laut Rydstedt et al. [12] ist in diesem Fall das Problem Google selbst, da keine Abwehrtechniken gegen Clickjacking bei der Bildersuche verwendet werden. Der Angreifer kann in einen `iframe` Google Images, mit einer Anfrage auf ein Bild (das sich auf der Homepage des Angreifers befindet), einbinden. Im Beispiel von Rydstedt et al. konnte ein CJ-Angriff mittels Google Images und Social-Networking-Seite MySpace durchgeführt werden. Zuerst hat man nach einem Bild von dem MySpace-Profil gesucht und dann beide Webseiten in einen `iframe` eingebettet.

doi: 10.2313/NET-2011-05-2_11

Abbildung 4: Der Angreifer kann Google Images mittels `iframe` in seine Webseite einbinden. Das Ergebnis der Suche wird dabei auch mit eingebunden.

Das Attribut `restricted` beim Internet Explorer
Webseiten, die im *Restricted Zone*-Modus vom Internet Explorer aufgerufen werden, werden in ihrer Funktion eingeschränkt, da aus Sicherheitsgründen kein JavaScript und keine Cookies erlaubt sind. Angreifer können dies ausnutzen und einen `iframe` mit dem Attribut `security=restricted` verwenden. Falls die aufgerufene Seite einen Framebusting-Code enthalten sollte, so wird dieser durch den Internet Explorer ausgeschaltet. Die eingebettete Seite wäre somit ungeschützt vor Clickjacking-Angriffen und könnte überall eingebunden werden.

3.2 ClickIDS

Balduzzi et al. [8] haben in ihrem Paper *A Solution for the Automated Detection of Clickjacking Attacks* ein System vorgestellt, welches automatisch Clickjacking-Angriffe erkennt. Dieses System wurde *ClickIDS* genannt (IDS für *Intrusion Detection System*). ClickIDS überprüft alle anklickbaren Objekte einer Webseite anhand ihrer Koordinaten. Wenn sich zwei oder mehr Objekte bei einem Mausklick überlappen, wird ein auffälliges Verhalten registriert. Das System besteht aus zwei Teilen: einer Testumgebung und einer *Detection Unit*. Das automatisierte System spielt in der Testumgebung Szenarien durch und versucht, alle Objekte anzuklicken. Die Detection Unit analysiert eine Webseite und überprüft sie auf mögliche CJ-Angriffe. Sie besteht aus zwei Browser-Plugins wobei eines davon *NoScript* ist. Es wurden umfangreiche Tests mit dem System durchgeführt (es wurden zirka 70,000 URLs untersucht). Die Resultate haben gezeigt, dass ClickIDS zuverlässig Clickjacking-Angriffe erkennt. Es gibt zwar Grenzfälle und das System schlug oft falschen Alarm, dennoch wurden zwei Clickjacking-Attacken richtig durch das System erkannt.

3.3 NoScript/ClearClick

NoScript ist eine kostenlos erhältliche Erweiterung für den populären Internetbrowser *Firefox*. Die Erweiterung schützt vor Cross-Site-Scripting Angriffen und bietet ein Modul an, das auch Clickjacking-Angriffe erkennt und abwehren kann. Dieses Modul nennt sich *ClearClick*. Dieses Plugin erkennt Maus-Klicks auf transparenten Elementen und bei einem potentiellen Angriff bekommt der Benutzer eine Warnung (siehe Abbildung 5). Die aktuelle Aktion wird vorzeitig gestoppt. ClearClick verursacht sehr viele falsch-positive Resultate laut Balduzzi et al.

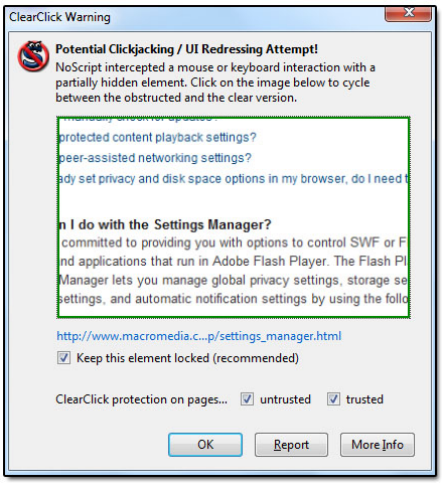


Abbildung 5: ClearClick erkennt potentielle Clickjacking-Angriffe und warnt den Benutzer im Voraus.

3.4 X-FRAME-OPTIONS Header

Das Entwicklerteam rund um Eric Lawrence von Microsoft hat im Internet Explorer 8 einen Mechanismus eingebaut, um Webseiten vor Clickjacking-Angriffen zu schützen (siehe [7]). Beim Erstellen von Webseiten können Entwickler einen `X-FRAME-OPTIONS`-Header angeben. Dieser stellt sicher, dass die Webseite in keiner anderen eingebettet wird und somit für einen Angriff ausgenutzt werden kann. Der Header erlaubt zwei Attribute: `DENY` und `SAMEORIGIN`. Während das erste Attribut allen Seiten verwehrt, die Webseite einzubetten, erlaubt das zweite Attribut der aktuellen Webseite beziehungsweise der gleichen Seite das Einbetten. Laut [12] verwenden nur vier Seiten aus den Top 10,000 der Alexa-Liste den `X-FRAME-OPTIONS`-Header. Dies kann auf die schwierige Einbindung des Headers zurückgeführt werden. Der Header kann in alle Seiten manuell eingebunden werden aber es treten Probleme auf, wenn ein Entwickler mehrere Domänen benutzen möchte, da es keine Liste mit individuell erlaubten Seiten gibt. Eine wesentlich elegantere Lösung wäre das serverseitige Mitsenden des `X-FRAME-OPTIONS` -Headers. Damit wären alle Webseiten mit dem Header ausgestattet.

`X-FRAME-OPTIONS` wird von den Browsern Microsoft Internet Explorer 8+, Apple Safari 4+ und Google Chrome 2+ unterstützt. Firefox unterstützt das Feature ab Version 3.6.9+.

4. VERWANDTE ARBEITEN

R. Hansen und J. Grossman stellten *Clickjacking* auf der OWASP NYC AppSec 2008 Konferenz vor (siehe [5]). Bei der Untersuchung der Angriffstechnik wurde eine Schwachstelle in Adobe's Flash Player gefunden. Adobe hat diese Lücke im Flash Player 10 behoben und das Security Team hat den beiden Experten für die Entdeckung gedankt (siehe [2]).

doi: 10.2313/NET-2011-05-2_11

Es existieren ähnliche Angriffe wie Clickjacking - eine der Angriffe, welche auch von Grossman auf der OWASP Konferenz erwähnt wurde, nennt sich *Cross-Site Request Forgery* (kurz CSRF). Diese Technik nutzt Schwachstellen in HTML aus und bringt den Browser des Opfers dazu, eine Aktion des Angreifers auszuführen. Dazu lockt der Angreifer den unwissenden Benutzer zuerst auf eine aufgesetzte, falsche Webseite. Diese sendet dann eine falsche Abfrage an die echte Webseite, die daraufhin die bösartige Aktion ausführt. Zeller et al. [16] haben in ihrem Paper nachgewiesen, dass es möglich ist, einen CSRF-Angriff auf die Webseite der Zeitung "New York Times" durchzuführen. Mittlerweile wurden die Schwachstellen seitens der New York Times behoben. CSRF-Angriffe sind sehr einfach zu entwickeln aber gleichzeitig auch sehr einfach zu beheben.

J. Grossman, einer der Entdecker von Clickjacking, gab ein ausführliches Interview in dem Artikel *Silver Bullet Talks with Jeremiah Grossman* [9], wo er unter anderem auf den Fall "Adobe" und Gefahren im Netz generell einging. Es wurde CSRF (und auch Cross-Site-Scripting) angesprochen und klar gemacht, dass Clickjacking unter anderem ein Problem ist, auf das Browser-Hersteller reagieren müssen.

Den ersten Report eines möglichen Clickjacking-Falls gab es in 2002. Jesse Ruderman hatte einen Bug-Report für *Mozilla* (siehe [11]) mit dem Titel *iframe content background defaults to transparent* erstellt. In einer kurzen Beschreibung erklärte er einen potentiellen Clickjacking-Angriff auf Yahoo. Sieben Jahre später veröffentlichte Grossman einen Blog-Eintrag (siehe [4]), in dem er voraussagte, dass Clickjacking-Angriffe vermutlich zwischen 2014 und 2017 immer mehr in den Vordergrund rücken werden.

Im White Paper von der Security-Firma *context* (siehe [14]) wird ein *Next Generation Clickjacking* beschrieben. Es handelt sich um einen Clickjacking-Angriff, bei dem der Benutzer eine Drag-and-Drop-Aktion ausführt und dabei zum Beispiel den gesamten Text von einer Facebook-Wall markiert und mit einem Klick auf einen Button an den Angreifer sendet. Das White Paper zeigt, dass es noch sehr viele Schwachstellen gibt und mit Framebusting alleine nicht alle Angriffe abgewehrt werden können.

5. ZUSAMMENFASSUNG

Framebusting sowie das System *ClickIDS* und die Erweiterung *NoScript* mit dem Plugin *ClearClick* haben gute Erfolge bei der Erkennung und Abwehr von *Clickjacking*-Angriffen gezeigt. Während es bei Framebusting einige Schwachstellen gibt, hat ClickIDS gute Ergebnisse geliefert. Es gibt zwar sehr viele falsch-positive Resultate, diese sind jedoch auf Grenzfälle zurückzuführen. Der `X-FRAME-OPTIONS` -Header ist eine Alternative zu den bestehenden Techniken.

Clickjacking-Angriffe waren für eine kurze Zeit populär, vor allem durch Attacken auf die Social Networking-Plattformen Twitter, Facebook und der Schwachstelle im Adobe Flash Player. J. Grossman [4] sagt, dass Clickjacking in den nächsten Jahren kein bedeutendes Sicherheitsrisiko spielen wird. Grossman erwähnt, dass Cross-Site-Scripting erst nach 8 Jahren (in 2005) ein großes Sicherheitsrisiko wurde. SQL Injections wurden erst nach 9 Jahren Entwicklung zu einem größeren Problem. Die erste Schwachstelle bezüglich Clickjacking wurde 2002 gefunden und 2008 wurde der Begriff "Clickjacking" das erste Mal erwähnt. Wenn man diese Entwicklungszeiten in Betracht zieht, so stellt Clickjacking vorerst keine Gefahr dar. Hersteller und Entwickler reagieren sehr schnell auf Schwachstellen und sorgen dafür, dass diese Sicherheitslücken schnell geschlossen werden. Die Resultate der Untersuchung von Balduzzi et al. [8] zeigen, dass es zurzeit noch nicht sehr viele Webseiten mit CJ-Angriffen gibt. Dennoch ist Vorsicht geboten und es ist ratsam, NoScript zu installieren falls man Firefox verwendet. Eine sehr einfache Lösung gegen Clickjacking wäre, dass User sich aus allen Webapplikationen sowie Webseiten nach einer Session richtig ausloggen. Die meisten Angriffe auf Seiten wie Twitter oder Facebook funktionierten, weil die Benutzer noch in den jeweiligen Seiten eingeloggt waren. Clickjacking-Angriffe nehmen von der Tatsache Gebrauch, dass Nutzer in Diensten wie Facebook oder Twitter meist eingeloggt bleiben. Sehr viele Nutzer schließen nur das Browser-Fenster. Dabei bleibt eine Session jedoch aktiv. Eine einfache Lösung wäre, dass der Nutzer nach einer bestimmten Zeit, in der er inaktiv war, automatisch ausgeloggt wird. Für Formulare gäbe es die Möglichkeit der Authentifizierung. Ein Benutzer muss etwas in ein Feld schreiben um das Formular wegschicken zu können, sei es ein Code oder eine Nummernfolge. Dies kann verhindern, dass CJ-Angriffe ausgenutzt werden, um vorgefüllte Formulare ohne Bestätigung des Nutzers abzusenden. Clickjacking-Angriffe sind sehr schnell zu entwickeln und der unerfahrene Benutzer kann schnell Opfer eines Angriffs werden, ohne dass der/diejenige etwas davon bemerkt.

6. LITERATUR

[1] Webcam ClickJacking. `http://www.youtube.com/watch?v=gxyLbpldmuU` [09.12.2010], Oktober 2008.

[2] Adobe (PSIRT). Thanks to Jeremiah Grossman and Robert "RSnake" Hansen. `http://blogs.adobe.com/psirt/2008/09/thanks_to_jeremiah_grossman_an.html` [09.12.2010], September 2008.

[3] D. Goodman. *JavaScript Bible*. Hungry Minds, Inc., 2001.

[4] J. Grossman. Clickjacking 2017. `http://jeremiahgrossman.blogspot.com/2009/06/clickjacking-2017.html` [09.12.2010], Juni 2009.

[5] J. Grossman and R. Hansen. New Zero-Day Browser Exploits - ClickJacking. `http://video.google.com/videoplay?docid=-5747622209791380934&hl=en#` [09.12.2010], 2008.

[6] H. Herold. *Das HTML/XHTML Buch: mit Cascading Style Sheets und einer Einführung in XML*. SuSe-PRESS, 2002.

[7] E. Lawrence. IE8 Security Part VII: ClickJacking Defenses. `http://blogs.msdn.com/b/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx` [09.12.2010], Januar 2009.

[8] M.Balduzzi, M.Egele, E.Kirda, D.Balzarotti, and C.Kruegel. A Solution for the Automated Detection of Clickjacking Attacks. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security (AsiaCCS)*, Beijing, China,

doi: 10.2313/NET-2011-05-2_11

April 2010.

[9] G. McGraw. Silver Bullet Talks with Jeremiah Grossman. *Security Privacy, IEEE*, 7(2):10 –14, 2009.

[10] W. Nefzger. *CSS: Cascading Style Sheets für Profis*. Franzis Verlag GmbH, 2006.

[11] J. Ruderman. iframe content background defaults to transparent. `https://bugzilla.mozilla.org/show_bug.cgi?id=154957` [09.12.2010], Juni 2002.

[12] G. Rydstedt, E. Bursztein, D. Boneh, and C. Jackson. Busting frame busting: a study of clickjacking vulnerabilities at popular sites. In *in IEEE Oakland Web 2.0 Security and Privacy (W2SP 2010)*, 2010.

[13] D. Sandler. Quick explanation of the 'Don't Click' attack. `http://dsandler.org/outgoing/dontclick_orig.html`, Feb. 2009.

[14] P. Stone. Next Generation Clickjacking - New attacks against framed web pages. `http://www.contextis.co.uk/resources/white-papers/clickjacking/Context-Clickjacking_white_paper.pdf` [09.12.2010], April 2010.

[15] Twitter Blog. Clickjacking Blocked. `http://blog.twitter.com/2009/02/clickjacking-blocked.html` [09.12.2010], Februar 2009.

[16] W. Zeller and E. W. Felten. Cross-Site Request Forgeries: Exploitation and Prevention, Oktober 2008.

doi: 10.2313/NET-2011-05-2_11

Stormbot - Ein Botnetzwerk im Detail

Steve Walter
Betreuer: Matthias Wachs
Seminar Innovative Internet-Technologien und Mobilkommunikation WS2010/11
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: walteste@in.tum.de

ABSTRACT

In dieser Ausarbeitung wird das Peer-to-Peer-Bot-Netzwerk Stormnet basierend auf dem Stormbot detailiert behandelt. Es werden allgemeine Eigenschaften des Bots herausgearbeitet und analysiert. Die geschichtliche Entwicklung von Bots wird erläutert und Stormbot in einen zeitlichen Rahmen eingeordnet. Zum besseren Verständnis der Bot-internen Abläufe wird auf das Kademlia Peer-to-Peer-Protokoll eingegangen, auf dem Stormbot basiert. Auf dieser Grundlage wird die Art und Weise der Verbreitung und die Kommunikation innerhalb des Botnetzwerkes genau beschrieben. Darauf basierend werden die Schwachstellen der derzeitigen Implementierung aufgezeigt und auf mögliche Verbesserungen hingewiesen. Des Weiteren wird auf Methoden eingegangen, mit denen das Netzwerk übernommen und zerschlagen werden kann, sowie auf die Möglichkeiten die zur Verfügung stehen dies zu unterbinden.

Keywords

Stormbot, Peer-to-Peer-Netzwerke, Kademlia, Sybil-Angriff

1. EINLEITUNG

Ein Bot ist die Software, welche auf einen einzelnen Rechner installiert ist und diesen Rechner fernsteuert. Kommunizieren und organisieren sich mehrere Bots untereinander und führen somit bestimmte Aufgaben aus, spricht man von einem Botnetzwerk. Dabei bekommen die einzelnen Bots über unterschiedliche Verfahren Befehle vom Botnetzbetreiber mitgeteilt. Botnetzwerke stellen eine zunehmende Bedrohung für die Sicherheit im Internet dar. Durch eine Masse ferngesteuerter Rechner ist es möglich eine Vielzahl von Angriffen durchzuführen, unbefugte Daten zu sammeln oder Spam zu versenden. Da es sich dabei in der Regel um illegale Aktivitäten handelt und hauptsächlich privaten Rechner mit Bots infiziert werden, geschieht dies meist ohne das Wissen des jeweiligen Nutzers. Der Großteil der bestehenden Botnetzwerke arbeiten zentralisiert indem sich die einzelnen Bots zu einem bestimmten Server verbinden von dem sie ihre Aufträge bekommen. Diese Botnetzwerke können durch das Aufspüren und Abschalten des Zentralservers unschädlich gemacht werden. Eine wesentlich robustere Variante ist ein dezentralisiertes Botnetzwerk, bei dem kein zentraler Server existiert, sondern die Kommunikation zwischen den Bots basierend auf einem Peer-to-Peer-Netzwerk stattfindet. Bots die auf einer solchen Technologie basieren nennt man Peer-to-Peer-Bots (P2P-Bots). Im Zuge dieser Arbeit wird Stormbot behandelt, einer der ersten P2P-Bots der zwischen 2006 und 2007 aktiv war.

Im Ersten Abschnitt dieser Arbeit werden allgemeine Eigenschaften des Stormbots behandelt. Darunter fällt die globale Verbreitung sowie die Ziele und Möglichkeiten des Botnetzwerks. Im zweiten Abschnitt wird die Geschichte von Botnetzwerken, begonnen beim ersten IRC-Bot bis hin zum Stormbot, kurz beschrieben. Es folgt ein Abschnitt über das Kademlia-Protokoll, auf dessen Basis Stormbot konstruiert ist. Unter Anderem wird auf den Beitritt in das Netzwerk sowie die Suche innerhalb des Netzwerks eingegangen. Der folgende zentrale Abschnitt dieser Arbeit beschäftigt sich mit Stormbot im Speziellen. Es wird auf die Verbreitung des Bots, die Infektion des Rechners sowie die Befehlsstruktur des Botnetzwerks näher eingegangen. In den letzten Abschnitten wird auf die Schwachstellen des Netzwerkes hingewiesen und ein erfolgreicher Angriff, sowie entsprechende Gegenmaßnahmen geschildert. Am Ende befindet sich eine kurze Zusammenfassung der zentralen Eigenschaften des Stormbots und die finalen Erkenntnisse die aus dem Botnetzwerk gewonnen werden können.

2. ALLGEMEINE INFORMATIONEN

Stormbot verbreitet sich von Rechner zu Rechner in Form eines Trojaners, der den offiziellen Namen “Trojan.Peacomm” trägt. Er befällt ausschließlich die Betriebssysteme Windows 95, 98, ME, 2000, NT und XP. Die Größe des gesamten Botnetzwerks wird auf minimal 5.000 bis 6.000 und maximal 45.000 bis 80.000 infizierte System geschätzt [2]. Infizierte Rechner befinden sich weltweit in mehr als 200 Ländern. Die USA ist mit 31% infizierten Maschinen am stärksten betroffen. Gefolgt von Russland 15% und Indien 9,2%. Deutschland fällt mit 5,1% ins Gewicht. Eine aktuelle Statistik über die globale Verbreitung des Strombots kann in [6] gefunden werden.

Stormbot wurde hauptsächlich mit dem Ziel entwickelt Spam-Mails zu versenden und “Distributed Denial of Service”-Attacken auszuführen. Dies hat zur Folge, dass das Netzwerk ebenso im Stande ist sich durch illegale Informationsbeschaffung, meist in Form von E-Mail-Adressen, zu vergrößern. Des Weiteren kann das Netzwerk in verschiedene Teilnetze gegliedert werden, sodass davon auszugehen ist, dass das Botnetzwerk in Teilen auch vermietet hätte werden können. Somit diente es auch der Kapitalbeschaffung des Betreiber. Es wird davon ausgegangen, dass die Betreiber des Botnetzes entweder durch die Vermietung des Botnetzes zum Versenden von Spam Geld machen oder selbst ein pharmazeutisches Unternehmen führen für dessen Medikamente per Botnetz Spam-Werbung versendet wird.

doi: 10.2313/NET-2011-05-2_12

Bekannte Ziele der DDoS-Attacken[1] des Stormbotnetzwerkes sind unter anderem Wirtschaftsseiten und Anti-Spam-Seiten. Des Weiteren wurden Seiten konkurrierender Botnetzwerke angegriffen um womöglich die Konkurrenz auszuschalten. Auch wurden Analysten die versuchten die ausführbare Datei, welche Stormbot enthält zu analysieren Opfer von DDoS-Attacken. Jedoch ist nicht bekannt, ob diese automatisch erfolgen, wenn der Bot bemerkt, dass er analysiert wird oder ob sie vom Betreiber ausgelöst wurden.

Die Kommunikation von Stormbot erfolgt über Overnet, welches eine konkrete Umsetzung von Kademlia darstellt. Kademlia ist ein Protokoll zur Kommunikation in Peer-to-Peer-Netzwerken, das auf verteilten Hashtabellen basiert. Overnet selbst wurde für die Verwendung von Filesharing-Programmen wie eDonkey2000 und BitTorrent entwickelt, welche nicht über einen zentralen Server kommunizieren, sondern Daten innerhalb des P2P-Netzes übertragen.

3. GESCHICHTLICHE EINORDNUNG

Die meisten zentralgesteuerten Bots basieren auf dem Internet Relay Chat (IRC), welches eine Chatsoftware ist, die leicht zu handhaben ist und sehr gut mit der Anzahl der Nutzer skaliert. Der erste Bot wurde im Dezember 1993 entwickelt und trug den Namen "EggDrop". Die ersten Bots waren ursprünglich dazu entwickelt die Handhabung von IRC weiter zu vereinfachen und zu automatisieren. Aus diesen Grunde hatten die zu dieser Zeit entwickelten Bots alle noch keinen schädlichen Charakter, sondern wirkten den Benutzer unterstützend. Erst in April im Jahr 1998 wurde der erste schädliche IRC-Bot mit dem Namen "Global Thread" in mehreren Varianten entwickelt. Dieser zeichnete sich durch eine modifizierte ausführbare Datei von IRC aus, welche in der Lage war Schadcode in Form von Scripten auszuführen. Im Mai 1999 wurde mit Napster der erste offizielle Peer-to-Peer-Filesharing-Service veröffentlicht. Napster benutzte allerdings einen zentralen Server um die Anfragen der einzelnen Peers zu koordinieren. Erst im März 2000 wurde mit Gnutella der erste P2P-Service veröffentlicht, der keine zentralen Server zur Koordination benötigte. Die Verbindungen in diesen Netzwerk wird von den Knoten selbst erstellt, sodass jeder Knoten bei seinen Nachbarknoten die Adressen anderer Knoten erfragen kann. Damit bekommt jeder Knoten eine Liste von möglichen Knoten bei denen er nach Informationen oder nach anderen Knoten suchen kann. November 2003 wurde dieses Konzept in Form von Kademlia auf verteilte Hashtabellen erweitert. Verteilte Hashtabellen bieten den Vorteil, dass die Suche im Vergleich zu normalen linearen Listen wesentlich effizienter stattfinden kann. Zuvor wurde im März 2003 mit WASTE ein VPN[2] ähnliches P2P-Netzwerk entwickelt, welches zur Verschlüsselung der Verbindung RSA[3] nutzte. WASTE enthält neben der Verschlüsselung noch weitere Funktionen unter anderem einen "Random Traffic Generator" und die Möglichkeit die Benutzung von WASTE zu verschleiern. Es bildet damit die erste Implementierung eines Peer-to-Peer-Netzwerkes welches starken Wert auf Sicherheit legt. Basierend auf WASTE entstand im März 2004 mit "Phatbot" der erste P2P-Bot. Dieser Bot zeichnete sich durch modulares Design aus, was zur Folge hat, dass er sich schnell an neue Sicherheitslücken anpassen kann. So verwendete er zum Beispiel die von Sasser genutzten Sicherheitslücken um sich zu verbreiten. Schließlich entstand am 29. Dezember 2006 die erste Variante des Stormbots.

[1]Distributed Denial of Service-Attacken sorgen dafür, dass ein bestimmter Service oder eine Infrastruktur aufgrund von Überlast nichtmehr verfügbar ist.

[2]Virtual Private Network dienen dazu kleinere, scheinbar unabhängige Netzwerke innerhalb eines großen Netzwerkes zu integrieren

[3]RSA ist ein asymmetrisches Verschlüsselungsverfahren zur Verschlüsselung von Daten

4. KADEMLIA-PROTOKOLL

Da Stormbot auf dem Kademlia-Protokoll basiert, sind die Kommunikationswege bis auf einige Ausnahmen gleich. Um die Funktionsweise von Stormbot zu verstehen wird deshalb ein Grundverständnis von Kademlia benötigt. Kademlia ist ein Service für P2P-Netzwerke, bei dem jeder Knoten der im Netzwerk teilnimmt sowohl Senden als auch Empfangen kann. Für die Kommunikation innerhalb des Kademlia-Netzwerkes wird UDP verwendet. Hauptmerkmal von Kademlia ist, dass die Knoten des Netzwerkes in verteilten Hashtabellen eingeordnet werden. Jeder Knoten besitzt dabei eine eindeutige "Distributed Hash Table ID" (DHT ID), welche durch einen 128-Bit MD4 Hashwert dargestellt wird. Dieser Hashwert wird beim ersten Eintritt in das Netzwerk automatisch generiert. Des Weiteren besitzt jeder Knoten eine Hashtabelle in der die Adressen anderer Knoten sowie deren Distanz gespeichert sind. Distanz meint in diesem Zusammenhang nicht die tatsächliche räumliche Distanz sondern die Distanz innerhalb des Netzwerks. Es kann zum Beispiel vorkommen, dass der direkte Nachbar eines infizierten Rechners in Deutschland ein infizierter Rechner in Australien ist. Die Distanz ergibt sich aus dem Modulo der beiden Hashwerte deren Distanz ermittelt werden soll. Ist zum Beispiel die DHT ID des Knoten A 1101 und die DHT ID des Knoten B 1001 so ergibt sich eine Distanz von $1101 \oplus 1001 = 0100$. Der Vorteil dieses Verfahrens und verteilter Hashtabellen besteht darin, dass die Suche innerhalb dieser Tabellen wesentlich schneller ist als bei linearen Listen. Dazu werden die DHT IDs in k-buckets sortiert, welche den Adressraum in entsprechende Teile spalten.In den k-buckets werden die Verbindungsdaten anderer Knoten gespeichert. Während der Kommunikation wird geprüft ob die in den k-buckets enthaltenen Verbindungsdaten aktuell sind und verfallene Verbindungsdaten werden entfernt. Dies garantiert, dass die Knoten nur eine Liste von aktiven Nachbarn verwalten und das Netzwerk nicht mittels einer DDoS-Attacke zerstört werden kann, bei welcher die Routingtabellen mit falschen Verbindungsdaten geflutet wird. Eine komplette Zusammenfassung über das Kademlia-Protokoll kann in [5] nachgelesen werden.

4.1 Bootstrapping

Um dem Netzwerk beizutreten, benutzt jeder Knoten eine als Bootstrapping bekannte Methode. Die Idee hinter diesem Verfahren ist, dass sich der Knoten von selbst in das Netzwerk einklinkt ohne auf einen speziellen Verbindungspunkt angewiesen zu sein, der ihn in das Netzwerk eingliedert. Dabei muss dem neuen Knoten, der sich mit dem Netzwerk verbinden will, allerdings ein Knoten der bereits Teil des Netzwerkes ist, bekannt sein. Dieser Einstiegsknoten kann jeder beliebige Knoten des Netzwerk sein und muss keine besonderen Eigenschaften erfüllen. Der neue Knoten verbindet sich zu diesem Einstiegsknoten und fordert eine Liste

doi: 10.2313/NET-2011-05-2_12

der dem Einstiegsknoten bekannten weiteren Netzwerkknoten an. Anhand dieser Liste kann der neue Knoten nun seine eigene Knotenliste erweitern und ist nun vom Einstiegsknoten unabhängig.

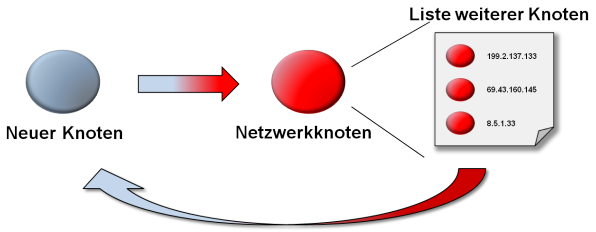


Figure 1: Bootstrapping

4.2 Suche innerhalb des Netzwerks

Um bestimmte Inhalte innerhalb eines DHT basierten Peer-to-Peer-Netzwerks zu finden wird zuerst der Hashwert des gesuchten Inhalts ermittelt. Im Anschluß werden die Knoten in der Hashtabelle mit der geringsten Distanz zum errechneten Hashwert befragt. Die befragten Knoten haben entweder die gesuchte Information, worauf dann eine direkte Verbindung zum Zielknoten aufgebaut werden kann oder sie berechnen ihrerseits die Knoten ihrer Hashtabelle mit der geringsten Distanz zum Hashwert des gesuchten Inhalts. Diese neu gefundenen Knoten werden dann zum suchenden Knoten übermittelt, welcher daraufhin diese Knoten nach den gesuchten Inhalten befragt. Besitzt der neu gefundene Knoten die gesuchten Informationen wird eine separate Verbindung zwischen den beiden Knoten aufgebaut und die Information übertragen. Andernfalls übersendet der neu gefundene Knoten ebenfalls einen Knoten aus seiner Hashtabelle der näher an dem vom suchenden Knoten benötigten Inhalten ist. Mit dieser Methode wird auf iterative Art und Weise der Knoten mit dem gesuchten Inhalt ermittelt.

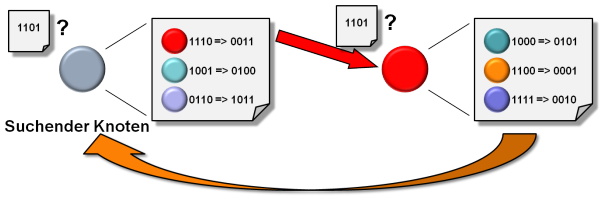


Figure 2: Suche nach dem Hashwert 1101

Ein verdeutlichendes Beispiel zeigt Figur 2. Der suchende, hellblaue Knoten will den Inhalt mit dem Hashwert 1101 erhalten. Dazu überprüft er die Distanz der Knoten seiner Hashtabelle. Die Distanz des roten Knoten zum suchendem Knoten beträgt 1110. Durch den Modulo-Operator wird die Distanz des roten Knoten zum gesuchten Inhalt ermittelt: $1110 \oplus 1101 = 0011$. Des weiteren ermittelt der suchende Knoten auch die Distanz der Anderen Knoten in seiner Hashtabelle und stellt fest, dass der rote Knoten näher an der gesuchten Information ist als alle anderen Knoten, die ihm bekannt sind. Folglich fragt der suchende Knoten den roten Knoten nach dem Inhalt mit den Hashwert 1101. Der rote Knoten wiederholt den Prozess der Distanzermittlung innerhalb seiner Hashtabelle und übersendet dem fragenden Knoten den orangen Knoten.

4.3 Nachrichten zwischen den Bots

Während des Kommunikationsprozess werden folgende Nachrichten verwendet:

hello wird verwendet um zu überprüfen ob ein Knoten existiert oder die umliegenden Knoten über die eigene Existenz zu informieren.

route request wird verwendet um Knoten die näher am gesuchten Hashwert sind von einem Nachbarknoten zu erfragen.

route response ist die Antwort auf einen route request und beinhaltet die empfohlenen Knoten die näher am gesuchten Hashwert sind.

publish request/response werden zum Veröffentlichen bestimmter Inhalte verwendet.

search request/response kommen zum Einsatz bei Suchen nach bestimmten Inhalten.

5. STORMBOT IM DETAIL

5.1 Verbreitung

Der Stormbot-Trojaner verbreitet sich ausschließlich über E-Mails mittels Social Engineering. Aus diesem Grund handeln die Inhalte der E-Mails meist von sozialen Ereignissen wie zum Beispiel Weihnachten, Neujahr oder Halloween. Des Weiteren gibt es Mails, die auf derzeitig aktuelle Ereignisse hinweisen, wie Tag der Arbeit, Beginn der Ferien oder der NFL Saison. So verdankt der Stormbot seinen Namen einer Verbreitungsmail in der vor dem Orkan Kyrill, der 2007 in der USA wütete, gewarnt wird. Auch sind Varianten im Umlauf in denen Spiele angeboten werden. In den Mails befindet sich neben einem Text entweder eine ausführbare Datei im Anhang die den Trojaner installiert oder ein Link von dem eine entsprechende Kopie des Trojaners heruntergeladen wird. Eine Liste der möglichen Anhänge kann in Tabelle 1 eingesehen werden. In Tabelle 2 sind einige Betreffzeilen mit denen Stormbot sich per Mail verbreitete, eingetragen. Es wurden keinerlei Sicherheitslücken im System ausgenutzt und die Infektion fand somit durch den Benutzer, wenn auch unter Vorspiegelung falscher Tatsachen, statt. Die Spam-Mails wurden hauptsächlich in der USA verbreitet. Mithilfe von *spamtraps* konnte die Anzahl der versendeten Spam-Mail, welche die ausführbare Datei des Stormbots enthielten festgestellt werden. Eine *spamtrap* ist ein für den Erhalt von Spam erstelltes E-Mail-Postfach anhand dessen analysiert werden kann, wie viel und welche Art Spam weltweit versendet wird. Vom September 2006 bis September 2007 wurden zwischen 2.200 und 23.900 Spam-Mails pro Tag vom Stormbotnetzwerk verschickt. Das entspricht 8.500 Spam-Mails im Durchschnitt und machte damit 10% aller von den *spamtraps* täglich gesammelten Spam-Mails aus [2].

FullVideo.exe	FullStory.exe	Video.exe
ReadMore.exe	FullClip.exe	GreetingPostCard.exe
MoreHere.exe	FlashPostCard.exe	GreetingCard.exe
ClickHere.exe	FullNews.exe	

Table 1: Namen der Anhänge der Stormbot-Spam-Mails

doi: 10.2313/NET-2011-05-2_12

Naked teens attack home director.
230 dead as storm batters Europe.
Radical Muslim drinking enemies's blood.
Chinese missile shot down Russian satellite.
Saddam Hussein alive!
Venezuelan leader: "Let's the War beginning".
Fidel Castro dead.

Table 2: Betreffzeilen der Stormbot-Spam-Mails

5.2 Infektion

Die Infektion erfolgt über 2 Phasen, die Erste und zweite Injektion genannt werden. Mit dem Ausführen des Anhangs wird die erste Injektion initiiert. In dieser Phase verankert sich der Bot im System. Dazu wird der Systemtreiber "wincom32.sys" im Windows-Stammverzeichnis erzeugt, welcher zur Verbindung mit dem Netzwerk benötigt wird. Dieser Treiber wird in die "service.exe" injiziert, wodurch dieser Service wie ein P2P-Client fungiert und die zweite Injektion herunterladen kann. Die Uhr des infizierten Rechner wird mit dem "Network Time Protocol" synchronisiert, da zur korrekten Kommunikation der Bots untereinander eine zeitliche Synchronisation vorausgesetzt wird (siehe Abschnitt 5.4). Die Synchronisation der Uhr mittels "Network Time Protocol" wird zwar von Windows XP ab Service Pack 2 automatisch vollzogen, jedoch ist dies auf den anderen Zielsystemen (Win 95/98/NT/2000/ME) nicht automatisch der Fall. Dies ist nötig um die Kommunikation zwischen den Bots zu gewährleisten. Des Weiteren werden der Windows Firewall und die Einstellungen der Internetsicherheit deaktiviert. Da es vorkommen kann, dass neben dem Windows Firewall noch weitere Firewalls auf dem infizierten Rechner installiert sind oder sich der Rechner hinter einem NAT[4] verbirgt, werden wie in [1] beschrieben eine Reihe von Ports zusätzlich geöffnet:

TCP:	139, 12474
UDP:	123, 137, 138, 1034, 1035, 7871, 8705, 19013, 40519

Nachdem eine DHT ID generiert wurde, tritt der neue Bot dem Netzwerk wie in 4.1 beschrieben bei. Im Stormbot-Netzwerk sucht der neue Bot nach einem bestimmten Hashwert, der ihm eine verschlüsselte URL zur zweiten Injektion übermittelt. Die URL wird entschlüsselt und die zweite Injektion heruntergeladen und installiert.

Bei der zweiten Injektion handelt es sich um sechs zusätzlicher Komponenten. Diese Komponenten sind optional und müssen nicht alle installiert werden. Sie tragen die Namen "game0.exe" bis "game5.exe". Die zusätzlichen Funktionalitäten sind wie folgt [7]:

- Erweiterung zum Herunterladen weiterer Module oder Befehle
- Rootkit um den Bot vor dem System zu verbergen
- SMTP- und Spam-Client zum versenden von Spam-Mails
- E-Mail-Adressensammler zur Verbreitung des Bots per Mail
- DDoS-Attacken-Komponenten

Nach dem Abschluss der zweiten Injektion beginnt der Bot das Netzwerk nach Befehlen zu durchsuchen.

Zu Forschungszwecken wurde die zweite Injektion von einer Gruppe von Analysten mehrfach von der vom Bot gefundenen URL heruntergeladen um festzustellen, ob sich die der Inhalt oder die Zusammensetzung der zweite Injektion verändert. Dabei stellte sich heraus, dass wenn man die zweite Injektion in einem bestimmten kurzen Zeitintervall (innerhalb weniger Minuten) mehrfach mit der gleichen IP-Addresse herunterläd, nach einem undefinierten Zeitintervall ein DDoS-Angriff auf diese IP-Addresse erfolgt. Dadurch gerieten die Analysten selbst ins Visir des Botnetzwerk. Es ist allerdings nicht bekannt, ob diese Angriffe automatisch erfolgten oder vom Betreiber des Botnetzwerk ausgingen, da die Zeitintervalle die zwischen dem Herunterladen und den DDoS-Angriffen lagen stark varierten und von 10 Minuten bis zu einem halben Tag reichten.

5.3 Verbindung zum Netzwerk

Um den Netzwerk beizutreten, benötigt es wie in im Abschnitt 4.1 beschrieben einen Einstiegsknoten. Zu diesem Zweck wird mit der ersten Injektion im Windows-Stammverzeichnis die Datei "%windir%\system\wincom32.ini" angelegt. Diese enthält eine Liste von über 140 Einstiegsknoten in das Netzwerk. Diese Liste ist in der ersten ausführbaren Datei des Trojaners hartkodiert. Es ist nicht bekannt, wie diese Liste zusammengestellt ist und ob sie unter Umständen aktualisiert wird. Sie hat den folgenden Aufbau:

```
<DHT ID>=<IP><Port>00
<DHT ID>=<IP><Port>00
...
```

Zum Beispiel:

```
D943283AB63746B8E62436682728DDD4=5511238154BD00
D6E46BF02E64D940E37EECCC982584A8=573349B6124A00
...
```

wobei auf der linke Teil die DHT ID darstellt und der rechte Teil die IP und den Port in Hexadezimal:

```
Knoten DHT ID: D943283AB63746B8E62436682728DDD4
0x55.0x11.0x23.0x81:0x54BD = 85.17.35.129:21693
```

Anhand dieser Einstiegsknoten verbindet sich der neue Bot zum Netzwerk. Nach dem ersten Verbinden hat dieser dann eine eigene Liste von Kontaktknoten von den entsprechenden Einstiegsknoten erhalten, sodass er danach von dieser Liste unabhängig ist.

[4]Network Address Translation verändern die Addressdaten in einem Datenpaket. Tritt zum Beispiel bei Routern auf.

doi: 10.2313/NET-2011-05-2_12

5.4 Steuerung des Botnetzes

Damit Befehle an die Bots übermittelt werden können, benötigt es bestimmte Kontrollknoten zu denen sich die einzelnen Bots verbinden und Befehle entgegen nehmen können. Diese Kontrollknoten werden anhand eines bestimmten Suchhashwerts ausfindig gemacht. Dieser wird durch einen Algorithmus berechnet, der in jedem Bot verankert ist. Der Suchhashwert errechnet sich aus dem aktuellem Datum und einen zufälligen Wert im Intervall von $[0 - 32]$. Aus diesem Grund muss während der ersten Injektion die Uhr des infizierten Systems synchronisiert werden. Somit sind täglich 32 unterschiedliche Schlüssel im Umlauf. Diese Schlüssel fungieren als Treffpunkte an denen sich die Bots und die die Kontrollknoten treffen beziehungsweise anhand deren sie sich finden können. Da der Algorithmus sowohl dem Bots als auch den Kontrollknoten bekannt ist, veröffentlichen die Kontrollknoten ihre Inhalte unter dem Hashwerten nach den die Bots suchen oder in naher Zukunft suchen werden. Nach der Berechnung des Hashwerts innerhalb eines Bots wird eine Suche mittels "routing request" gestartet an deren Ende der gesuchte Kontrollknoten gefunden wird. Ist die IP-Adresse und das Port schließlich bekannt, wird eine TCP/IP-Verbindung zwischen Bot und Kontrollknoten hergestellt. Es erfolgt eine kurze Authentifizierung bei welcher der Kontrollknoten eine zufällige Zahl mit einem in jedem Bot hartkodierten Schlüssel XOR-verknüpft und dem verbundenen Knoten übersendet. Dieser errechnet mittels des hartkodierten Schlüssels "`0x3ED9F146`" die vom Kontrollknoten generierte, zufällige Zahl und sendet sie dem Kontrollknoten als Antwort zurück. Damit ist die Authentifizierung abgeschlossen. Der Kontrollknoten übermittelt im Anschluß weitere Befehle wie zum Beispiel Angriffsdaten für deine DDoS-Angriff oder den Inhalt einer Spam-Mail, die verbreitet werden soll. Durch die Kontrollknoten und die Suche der normalen Botknoten nach deren verbreiteten Daten lässt sich das Botnetzwerk direkt steuern und kann so vom Netzwerkbetreiber entsprechend verwendet werden.

In den frühen Versionen des Stormbotnetzwerkes wurde Overnet zur Kommunikation verwendet, welches keinerlei Verschlüsselung unterstützt. Mit fortschreitender Entwicklung und zunehmender Größe des Netzwerkes wurde das Netzwerk erweitert und eine 40-Bit XOR-Verschlüsselung integriert. Dieser 40-Bit-Schlüssel ist in jeden Bot hartkodiert. Alle Nachrichten innerhalb des Netzwerkes werden in neueren Versionen mit diesen Schlüssel verschlüsselt. Die Funktionalität bleibt aber die selbe wie in Overnet/Kademlia.

6. SCHWACHSTELLEN UND ÜBERNAHME

6.1 Schwachstellen

Die gewählte Funktionsweise und der Aufbau des Bots zeigen einige Schwachstellen die unter Anderem dazu führten, dass das große Teile des Botnetzwerk übernommen und zerschlagen werden konnten. Eine der offensichtlichsten Schwachstellen ist die Liste mit den Einstiegsknoten insofern sie nicht aktualisiert wird oder veränderlich ist. Sollte es gelingen alle in dieser Liste angegebenen Startknoten zu lokalisieren und vom Netz zunehmen, findet ein neuer Bot keinen Zugang mehr zum bestehenden Botnetzwerk. Des Weiteren bietet die Suche des Kontrollknotens eine weitere Schwachstelle, da durch ein Abfangen und fehlleiten der Routing-Anfrage die Suche auf einen falschen oder gefälschten Knoten umgeleitet werden kann. Es fehlt weiterhin ein sicheres kryptographisches Verfahren, das die Kommunikation zwischen den Bots verschleiert.

Eine gravierende weitere Schwachstelle stellen die hartkodierten Daten und Algorithmen dar. So konnte durch Reverse Engineering der ausführbaren Datei der Algorithmus zur Berechnung des Suchhashwertes und damit des Treffpunktes des Stormbots gewonnen werden. Die Auswirkungen dieser Erkenntnis werden im folgendem Abschnitt 6.2 behandelt. Ebenso wurde der 40-Bit-Schlüssel für die XOR-Verschlüsselung gefunden und der Hashwert unter dem die verschlüsselte URL für die zweite Injektion angeboten wird. Dadurch ist die Kommunikation zwischen den einzelnen Knoten nicht mehr sicher und kann abgehört werden. Auch der Schlüssel der zur Authentifizierung des Bots beim Kontrollknoten verwendet wird, ist extrahiert wurden.

6.2 Sybil-Angriff

Große Teile des Strombotnetzwerk wurden mithilfe des Sybil-Angriffs erfolgreich infiltriert und übernommen [2, 3]. Dazu werden künstliche Bots in das Netzwerk eingespeist. Künstliche Bots sind Bots, die von den Angreifern des Botnetzwerkes entwickelt wurden um das Botnetzwerk zu infiltrieren. Alle anderen Bots, die von den Entwicklern des Botnetzwerkes entworfen wurden, werden als echte Bots bezeichnet. Diese künstlichen Bots verbreiten falsche Routingeinträge mittels Hello-Nachrichten im Netzwerk. Dazu wird den echten Botnetzknoten in den Hello-Nachrichten vorgetäuscht, dass sich die künstlichen Knoten näher zu den tatsächlichen Botnetzknoten befinden als die echten Botnetzknoten. Durch diesen Umstand übernimmt der echte Botnetzknoten den künstlichen in seine Routingtabelle und entfernt im Besten Fall die Einträge der echten Botnetzknoten. So werden die echten Botnetzknoten sämtliche Anfragen in erster Linie an die künstlichen Bots senden. Da der Algorithmus mit denen die Bots ihren nächsten Treffpunkt berechnen gefunden wurde, können die gesuchten Schlüssel vor berechnet werden. Die künstlichen Bots leiten dann die echten Bots auf einen künstlichen Kontrollknoten, statt auf den Kontrollknoten des Betreibers, um. Dadurch ist die Übernahme gelungen und die Bots können mithilfe des künstlichen Kontrollknotens unschädlich gemacht oder manipuliert werden. Der Vorteil dieser Methode ist neben einer Übernahme des Netzwerk auch die Effizienz. Es werden keine Unmengen an Rechner benötigt um die Vielzahl an künstlichen Bots bereit zustellen. Sondern Dank der geringen Daten, die ein einzelner Knoten benötigt, ist es möglich an einen einzelnen Rechner das Netzwerk mit hunderten von künstlichen Bots zu überschwemmen.

Den Sybil-Angriff kann allerdings durch entsprechende Vorkehrungen verhindert werden[4]. Die sicherste Methode besteht darin eine vertrauenswürdige Autorität in das Netzwerk zu integrieren, die sicher stellt, dass es sich bei allen Teilnehmern des Netzwerkes um vollwertige Mitglieder handelt. Nicht validierte Knoten hätten somit keine Chance das Netzwerk zu zerschlagen. Ein weitere Methode wäre die Knoten einen Ressourcen-Test zu unterziehen. Dazu wird geprüft ob die Knoten die entsprechenden Ressourcen eines normalen Rechners aufweist und zum Beispiel die Rechenleistung oder die Speicherkapazität getestet. Zum Beispiel könnte dem Bot, der sich in das Netzwerk einklinken will, eine sehr rechenintensive Aufgabe gestellt werden, welche

doi: 10.2313/NET-2011-05-2_12

den Prozessor des Rechners auf dem der Bot läuft stark beansprucht. Der Bot müßte diese Aufgabe in einer vertretbaren Zeit lösen können. Wird nun versucht der Sybil-Angriff durch die Simulation mehrer Botknoten auf einen Rechner anzuwenden, wird der Prozesser des Rechners überlastet und die Aufgaben, die jede Simulation eines Bots gestellt bekommt, könnten nicht in vertretbarer Zeit gelöst werden. Die Variante des Sybil-Angriffs die auf einen Rechner ausgeführt werden kann, würde damit enttarnt werden, da ein Rechner nicht die Rechen- oder Speicherkapazität von hunderten Knoten aufbringen kann.

7. ZUSAMMENFASSUNG

Stormbot hat durch seine Peer-to-Peer-Struktur und erste primitive Sicherheitsverfahren in Form einer 40-Bit XOR-Verschlüsselung die Problematik der Botnetzwerke auf eine neue Ebene verlagert. Das Stormbotnetzwerk konnte sich über ein Jahr behaupten und in dieser Zeit Unmengen an Spam-Mails versenden und einige DDoS-Angriffe starten. Obgleich die gewählte Methode der Verbreitung über Spam-Mails keinerlei Sicherheitslücken ausnutzte und nur auf Social Engineering basierte, erwies sie sich dennoch als effektiv. Die Zerschlagung des Botnetzwerkes stellte seine Gegner vor ein größeres Problem aufgrund fehlender Zentralisierung. Da allerdings das verwendete Protokoll des Stormbotnetzwerkes identifiziert und dessen Schwachstellen lokalisiert werden konnten, war es möglich das Botnetz durch die Einspeisung künstlicher Bots zu infiltrieren und die Kontrollstrukturen des Botnetzwerkes zu übernehmen. Durch Reverse Engineering war es möglich die im Stormbotnetzwerk verwendeten Sicherheitsalgorithmen zu umgehen und die Treffpunkte der Knoten untereinander im Voraus zu berechnen. Heute hat Stormbot keinen nennenswerten Einfluss mehr. Dennoch bietet es durch die geführte Pionierarbeit in dem Bereich der Peer-to-Peer-Botnetzwerke eine ausreichende Basis für weitere Generationen von Botnetzwerken.

Peer-to-Peer-Botnetzwerke sind ernst zunehmende neue Herausforderungen im Bereich der Internetsicherheit. Das größte Problem besteht in der Dezentralisierung, welche dazu führt, dass alle Bots unabhängig sind und ein Loch im Netzwerk keine größeren Schäden anrichtet. Die Problematik besteht darin eine Masse von Bots einzeln auszuschalten oder die steuernden Knoten zu übernehmen. Auch gibt es keinen zentralen Server, durch dessen Abschaltung das Netzwerk zerschlagen werden könnte. Zur Verbreitung von Befehlen wird immer ein Treffpunkt für die Netzteilnehmer benötigt. Dieser Treffpunkt könnte die potentielle Schwachstelle aller Peer-to-Peer-Netzwerke darstellen. Ein Reverse Engineering ist zu diesem Zweck sehr hilfreich, kann aber bei intelligenten Bots zu unerwünschten Nebenwirkungen führen, wie Angriffe auf die Analysten. Ein ausführliches und intensives Reverse Engineering benötigt allerdings viel Zeit. Das System des Botnetzwerke konnte in diesem Fall zwar durch eine Sybil-Attacke gebrochen werden, jedoch gibt es bereits Methoden um Sybil-Attacken vorzubeugen, wodurch ein P2P-Netzwerk wesentlich schwieriger zu infiltrieren ist.

Die Struktur von Peer-to-Peer-Netzwerken könnte durch erhöhtes modulares Design und mehr Flexibilität in der Art der Kommunikation verbessert werden. Durch die Verwendung bekannter und bewährter kryptographischer Verfahren kann die Sicherheit eines P2P-Netzwerkes weiter erhöht werden. Dabei können bereits einfache und viel erprobte Sicherheitsverfahren wie ein Diffie-Hellmann-Schlüsselaustausch, synchrone und asynchrone Verschlüsselungsverfahren mittels bekannter Kryptographieverfahren wie AES[5] oder RSA einen wesentlichen Beitrag zur Sicherung des Peer-to-Peer-Netzwerkes beitragen.

8. REFERENCES

[1] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: Overview and case study. Technical report, The Johns Hopkins University, Georgia Institute of Technology, University of Noth Carolinat at Charlotte, 2007.

[2] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and mitigation of peer-to-peer-based botnets: A case study on storm worm. Technical report, Universität Mannheim, Institut Eurècom, 2007.

[3] F. Leder, G. Wicherski, T. Werner, and M. Schlösser. Stormfucker: Owning the storm botnet. 25th Chaos Communication Congress, Dezember 2008.

[4] B. N. Levine, C. Shields, and N. B. Margolin. A Survey of Solutions to the Sybil Attack. Tech report 2006-052, University of Massachusetts Amherst, October 2006.

[5] P. Maymounkovand and D. Mazi'eres. Kademlia: A peer-to-peer information system based on the xor metric. Technical report, New York University, 2002.

[6] McAfee and T. S. tm. Storm tracker, 2010.

[7] SecureWorks. Storm worm ddos attack, Februar 2007.

[5]Advanced Encryption Standard ist ein symmetrischer Verschlüsselungsalgorithmus zur Verschlüsselung von Daten

doi: 10.2313/NET-2011-05-2_12

ISBN 3-937201-19-X
DOI 10.2313/NET-2011-05-2

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