Michael Roßberg (Autor)

Skalierbare Autokonfiguration sabotageresistenter virtueller privater Netze

https://cuvillier.de/de/shop/publications/319

I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone

(Bjarne Stroustrup)

Kapitel 1

Einleitung und Motivation

Eine hauptsächliche Motivation bei der Entwicklung paketvermittelnder Kommunikationsnetze war deren im Vergleich zu verbindungsorientierten Netzen höhere Robustheit gegen zufällige Ausfälle und Sabotage [Bar64]. Auf dem Gipfel des kalten Krieges sollte in den USA so eine Plattform für den militärischen und behördlichen Nachrichtenaustausch geschaffen werden, welche selbst nach atomaren Angriffen funktionsfähig bleibt. Die dabei entwickelten Prinzipien haben über das ARPANET auch Eingang in die Funktionsweise des heutigen Internets gefunden.

Als Resultat sind die Routing-Mechanismen des Internets daher in der Lage auch größere Ausfälle ohne signifikante Probleme zu kompensieren. So war beispielsweise im New Yorker World Trade Center eine Vielzahl von Internet Providern angesiedelt, um die dort terminierenden Transatlantikkabel in ihre Netze zu integrieren. Trotz der Zerstörung dieser Verbindungen im September 2001 waren weiterhin 99% der Adresspräfixe global erreichbar, und auch die restlichen Netze standen innerhalb weniger Stunden wieder zur Verfügung [Com03].

Im Gegensatz zu den bis dahin aufgebauten Telefonnetzen und anfänglichen Regierungsnetzen, stellt das Internet allerdings kein geschlossenes Netz dar. Diese Offenheit bewirkt zwar, dass ein Zugang nahezu überall auf der Welt preiswert möglich ist; jedoch muss so auch von Angreifern innerhalb des Internets ausgegangen werden. Insbesondere in Bezug auf die Sabotageresistenz, in diesem Kontext teilweise auch als Verfügbarkeit oder Resistenz gegen Denial-of-Service (DoS)-Angriffe bezeichnet, sind die Mechanismen des Internet Protocol (IP) hier noch nicht ausreichend.

1.1 Defizite bei der Konfiguration virtueller privater Netze (VPN)

Um das Internet trotz seiner offenen Architektur zur Realisierung eines schnellen und sicheren, internen Informationsflusses nutzen zu können, schotten Firmen, Behörden, Organisationen aber auch Privatleute ihre Systeme vom Rest des Netzes ab und verbinden einzelne Teile über virtuelle Kanäle. Zur Sicherung von Datenvertraulichkeit, –integrität und –authentizität, sowie zur Realisierung von kontrolliertem

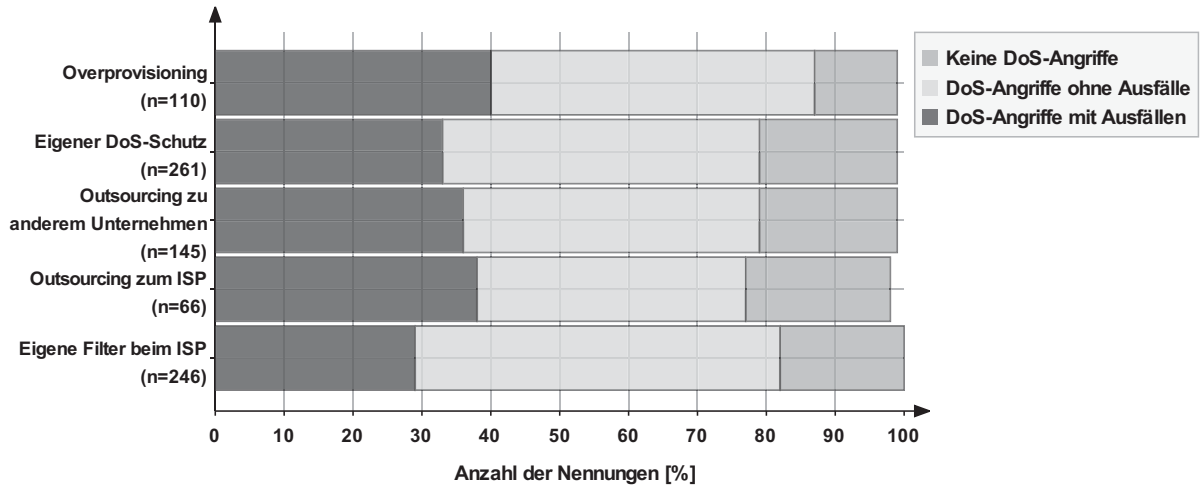

Abbildung 1.1: Umfrage unter 400 IT-Führungskräften zu DoS-Angriffen [For09]: Aufschlüsselung von Gegenmaßnahmen und Auswirkungen auf die IT-Infrastruktur über einen Zeitraum von 12 Monaten

Zugang werden die über die virtuellen Kanäle ausgetauschten Daten kryptographisch gesichert, sodass oberhalb des Internets ein virtuelles privates Netzwerk (VPN) entsteht.

Ein in diesem Zuge häufig realisierter Ansatz setzt ein oder mehrere zentrale VPN-Konzentratoren ein, welche Verbindungen von Außenstandorten oder mobilen Teilnehmern entgegennehmen und zwischen ihnen Daten weiterleiten. Diese Architektur ist leicht zu verwalten, und die Realisierung von Vertraulichkeit, Authentizität, Integrität und Zugriffskontrolle wird in diesem Kontext hinreichend gut beherrscht. Skalierbarkeit in Bezug auf Teilnehmer und Bandbreite, Robustheit sowie Sabotageresistenz stellen hingegen Probleme dar, da diese VPN mit den Konzentratoren einen dedizierten Ausfallpunkt und Flaschenhals besitzen. Auch wenn zurzeit relativ wenig über erfolgreiche Sabotageangriffe auf VPN berichtet wird, liegt dies vermutlich nicht an deren Abwesenheit sondern an den folgenden drei Gründen:

1. Die großen VPN wichtiger Institutionen werden mittels kostenintensiver und relativ unflexibler Service Level Agreements (SLAs) durch einen Internet Service Provider (ISP) vor ungezielten Angriffen geschützt (beispielsweise Verizon DOS Defense [Ham09, Pas06] oder Verisign Internet Defense Network [Ver10]).

2. Angriffe auf VPN erzeugen ferner in der Regel keinen messbaren *Backscatter*, also Pakete die von VPN-Teilnehmern an die Inhaber gefälschter IP-Adressen zurückgesendet werden. Dadurch können DoS-Angriffe auf VPN nicht von dritten Parteien gemessen werden.

3. Aufgrund des mit erfolgreichen Angriffen verbundenen Image-Schadens und der Angst vor Trittbrettfahrern, werden von DoS-Angriffen betroffene Organisationen diese von sich aus nicht an die Öffentlichkeit bringen.

Trotz fehlender Statistiken ist aber gerade bei VPN eine gewisse Immunität gegen DoS-Angriffe extrem wichtig, da sonst mit relativ geringem Aufwand sofort organisationsinterne Arbeitsabläufe vollständig verhindert werden können. Ein Schutz gegen Sabotageangriffe durch SLA funktioniert auch nur begrenzt, da zum einen kleine Institutionen kein solches Abkommen haben. Zum anderen besitzt nur der Betreiber eines VPN die nötigen kryptographischen Schlüssel und kann so legitime VPN-Pakete sicher von Angriffsverkehr unterscheiden. Soll ein VPN potentiell aus dem gesamten Internet erreichbar sein oder kennt der Angreifer die IP-Adressen der Gegenstellen, können die Filter des Internet Provider vergleichsweise leicht umgangen werden. Das Ermitteln von Filterregeln ist bereits für herkömmliche DoS-Angriffe sehr schwer, wie die in Abbildung 1.1 dargestellte Umfrage zeigt. Keine der aufgeführten Methoden zum Schutz gegen DoS-Angriffe kann Ausfälle auf ein akzeptables Niveau senken.

Alternativ können im Kontext von VPN nur sabotageresistentere Topologien erzeugt werden. Ein Beispiel sind die vollvermaschten Topologien, bei denen jeder Standort mit allen anderen verbunden wird, und Angreifer so keinen zentralen Punkt sabotieren können. Die Konfiguration ist jedoch aufwendig und damit potentiell fehleranfällig, erlaubt keine mobilen Teilnehmer, und die Skalierbarkeit in Bezug auf die Anzahl der VPN-Endpunkte ist problematisch, da die Anzahl der einzurichtenden Sicherheitsbeziehungen quadratisch mit ihnen wächst. Wird ein Teilnehmer eines vollvermaschten VPN angegriffen und erhält durch alternative Internet-Anbindung eine andere IP-Adresse, müsste nun in allen Geräten der anderen VPN-Teilnehmer eine Rekonfiguration vorgenommen werden. Die Angreifer können ferner die Kommunikation zwischen zwei Punkten stören, wobei durch die statische Konfiguration in diesem Fall keine Umleitungen von Verkehr innerhalb des VPN möglich sind, obwohl so unter Umständen eine Kommunikation über andere Standorte noch zu realisieren wäre.

Die einzige Möglichkeit auf eine statische Konfiguration zu verzichten, stellt der Einsatz eines Systems zur automatischen Konfiguration dynamischer VPN dar, wobei die existierenden Ansätze allerdings weder Sabotageresistenz noch Robustheit adressieren. Darüber hinaus erfüllen die wenigen skalierbaren Systeme selbst einfache funktionale Anforderungen nicht, wie etwa den Einsatz privater IP-Adressen innerhalb des VPN. Einige Ansätze schwächen zusätzlich die Sicherheit des VPN signifikant, indem beispielsweise unsichere kryptographische Verfahren oder Gruppenschlüssel eingesetzt werden (siehe Abschnitt 3.2).

Insgesamt erweist sich somit die manuelle Konfiguration großer VPN als zu aufwendig, fehleranfällig, unflexibel und bietet keinen ausreichenden Schutz vor bandbreitenerschöpfenden DoS-Angriffen. Die bestehenden VPN-Autokonfigurationssysteme vereinfachen zwar die Einrichtung und sind potentiell weniger anfällig gegen menschliche Fehler, berücksichtigen aber in der Regel lediglich wenige Szenarien, und sehen keine Mechanismen zur Realisierung von Robustheit und Sabotageresistenz vor.

1.2 Beiträge der Arbeit

Ausgehend von dieser Situation wurden im Rahmen der Dissertation und der vorangegangen Diplomarbeit [Ros07] zunächst umfangreiche Anforderungen für die automatische Konfiguration von VPN ermittelt. Wie bereits angedeutet, ergab ein systematischer Vergleich des Standes der Wissenschaft und Technik mit den Anforderungen eine relativ große Lücke [RS11].

Infolgedessen ist ein eigener Ansatz mit dem Namen Secure OverLay for IPsec Discovery (SOLID) entstanden, der von beherrschten Peer-to-Peer-Prinzipien ausgehend, ein vollständiges System zur Konfiguration von VPN mit folgenden Eigenschaften bereitstellt:

- Im Gegensatz zu bestehenden Ansätzen zeichnet sich SOLID durch die Realisierung umfangreicher funktionaler Anforderungen aus, wie der Möglichkeit verschachtelte Topologien oder VPN mit privaten IP-Adressbereichen zu konfigurieren.
- Durch die Einrichtung verschachtelter kryptographischer Tunnel wird die Sicherheit der übertragenen Daten selbst bei Anwesenheit interner Angreifer gewährleistet.
- Der konsequente Verzicht auf exponierte VPN-Teilnehmer und die Nutzung der VPN-Topologie zur Ableitung von Routing-Informationen ermöglichen einen robusten Umgang mit Ausfällen, aber auch eine schnelle Reaktion auf Sabotageangriffe.

Um die Resistenz gegen mögliche Sabotageangriffe weiter zu erhöhen, wurden im Kontext dieser Arbeit Ansätze zur Steigerung der Verfügbarkeit von VPN entwickelt und in SOLID integriert. Hierbei werden VPN-Teilnehmer in so genannte Verfügbarkeitszonen eingeteilt und eine direkte Kommunikation nur zwischen vordefinierten Zonen erlaubt. Auf diese Weise wird es für potentielle Angreifer schwieriger, Ziele für Sabotageangriffe zu identifizieren, und so die Resistenz des Netzes insgesamt erhöht.

SOLID ist ferner im Rahmen einer Simulationsumgebung implementiert worden und kann so auch für große Netze untersucht werden. Zusätzlich werden große Teile des Quellcodes wiederverwendet, um einen Prototypen auf Linux-Basis zu realisieren. Die vollautomatische Einrichtung von Kommunikationsinfrastrukturen auf Basis von IP security (IPsec) kann so simulativ und experimentell mit nahezu identischen Ergebnissen untersucht werden.

1.3 Übersicht

Die vorliegende Arbeit gliedert sich in acht Kapitel. Je nach Wissensstand und Interesse empfiehlt es sich, einige Passagen zu überspringen oder genauer zu lesen. Abbildung 1.2 gibt einen Überblick über einige mögliche Reihenfolgen.

Im folgenden zweiten Kapitel werden zunächst einige grundlegende Fakten zu Zielen der Netzwerksicherheit sowie aktuell eingesetzten VPN-Architekturen und Protokollen gegeben. Anschließend erfolgt ein Überblick über strukturierte Peer-to-Peer-Systeme und ihren möglichen Einsatz zur Weiterleitung von Unicast- und Multicast-Verkehr. Zusätzlich wird auf Netzwerkkoordinatensysteme eingegangen, dort im Speziellen auf den Vivaldi-Ansatz, und es werden mögliche Angriffe auf Peer-to-Peer-Systeme skizziert.

Kapitel 3 beginnt mit einem Überblick über funktionale und nichtfunktionale Anforderungen an VPN-Konfigurationssysteme. Sicherheitsanforderungen, im Speziellen in Bezug auf Verfügbarkeit, werden in eigenen Abschnitten detailliert dargelegt. Die erarbeiteten Anforderungen werden in Abschnitt 3.2 mit den 17 existierenden Ansätzen zur VPN-Autokonfiguration verglichen. Im Anschluss erfolgt ein kurzer Überblick über Overlay-Netze, die explizit konzipiert wurden, um die Sabotageresistenz angebotener Dienste zu erhöhen. Als dritte Komponente des Standes der Wissenschaft werden verschiedene Verfahren zur Realisierung von Routing in strukturierten Overlay-Netzen in Abschnitt 3.4 vorgestellt.

In Kapitel 4 wird SOLID — als Hauptansatz der Dissertation — vorgestellt. Beginnend mit einem kurzen Abriss über Prinzipien, Annahmen, potentielle Angreifertypen und zu erfüllenden Aufgaben, wird anschließend auf die Realisierung einzelner Dienste, wie etwa Bootstrapping und Topologiekontrolle eingegangen. Den letzten Teil des Kapitels bildet die Beschreibung von Multicast-Routing in SOLID-Netzen.

Die in SOLID durch eine dynamische Topologiekontrolle gewonnene Flexibilität kann zum Schutz vor DoS-Angriffen eingesetzt werden. Kapitel 5 geht dazu zunächst auf die Modellierung von VPN und DoS-Angriffen mittels Graphen ein. Anschließend werden optimale, externe DoS-Angriffe vorgestellt, sowie das Problem auf diese bestmöglich zu reagieren. Das Kapitel schließt mit einer Betrachtung interner DoS-Angriffe, sowie der Übertragung der vorgestellten Techniken auf SOLID.

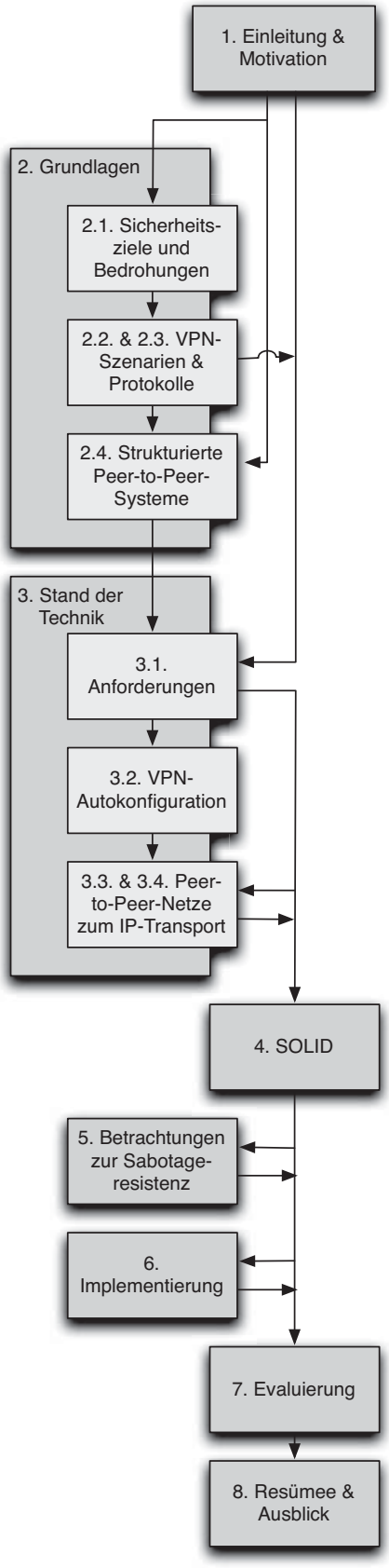

Abbildung 1.2: Empfohlene Lesereihenfolgen

Das sechste Kapitel beschäftigt sich mit der Umsetzung des vorgestellten Systems in Simulationsumgebung und Prototyp. In einem ersten Abschnitt werden dazu weitere die Implementierung betreffende Anforderungen hergeleitet, bevor die Umsetzungsstrategie und Software-Architektur vorgestellt werden. Die Realisierung ausgewählter Funktionen, wie Bootstrapping und Routing, wird in einzelnen Unterkapiteln detaillierter beschrieben.

Das letzte Hauptkapitel 7 beschäftigt sich mit dem Nachweis erreichter Anforderungen und Ziele. Im ersten Unterkapitel wird SOLID dazu systematisch in Bezug auf die in Abschnitt 3.1 aufgestellten Anforderungen diskutiert. Anschließend wird auf verschiedene Simulations- und Experimentalumgebungen eingegangen, in welchen SOLID quantitativ untersucht und bewertet wurde. Die folgenden Abschnitte beschreiben die eigentlichen Experimente und Resultate in Prototyp und Netzwerksimulator.

Die Arbeit schließt mit einer kurzen Zusammenfassung und einem Ausblick über mögliche, fortsetzende Arbeiten in Kapitel 8. Dabei wird sowohl auf zukünftige forschungsorientierte Aspekte als auch auf eventuelle Produktisierungsschritte eingegangen.