Leveraging SDN for DDoS defenses

Pirmin Blanz
Betreuer: Quirin Scheitle
Innovative Internet-Technologien und Mobilkommunikation SS2015
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: p.blanz@tum.de

KURZFASSUNG

Aktuelle Beispiele [1, 2] zeigen, das noch immer kein Wundermittel gegen großflächig angelegte Angriffe gefunden wurde, welche die Verfügbarkeit verschiedenster Internetdienste bedrohen. Die Rede ist von DDoS-Angriffen (*Distributed Denial of Service*). Initiiert und gesteuert von einigen wenigen Ausgangspunkten wird der Angriff schließlich auf hunderte, tausende oder zehntausende [3] Netzwerkknoten ausgeweitet und auf dem Rücken dieser (meist unwissenden) Helfer ausgetragen. Die Beteiligten bombardieren das Opfer schließlich millionenfach mit Anfragen, was zu einem Angriffsverkehr von über 300Gbps [4] führen kann. Durch Werkzeuge der etablierten Netzwerktechnik können DDoS-Offensiven kaum zeitnah erkannt und dadurch erfolgreich verteidigt werden. Mit dem Einsatz zentraler Steuergeräte liefert das SDN-Paradigma (*Software Defined Networking*) effektive Lösungsstrategien. Nachfolgend werden diese analysiert und herkömmlichen Verteidigungsmaßnahmen gegenübergestellt. Grundlage und Raster bildet hierbei eine Taxonomie, welche bekannte Angriffsmuster abstrahiert und kategorisiert.

Schlüsselworte

DDoS, SDN, DDoS-Verteidigung

1. EINLEITUNG UND MOTIVATION

Ungeachtet der Tatsache, das DDoS-Angriffe und deren Ausmaße seit über 20 Jahren bekannt sind, verursachen diese aktuell immer noch immense Schäden. Akamai [5] zufolge hat sich die Zahl der Vorfälle im Vergleich zum entsprechenden Vorjahreszeitraum sogar mehr als verdoppelt. Generell ist eine stark steigende Tendenz zu verzeichnen.

Dieses Wachstum hat mehrere Gründe. Als Voraussetzung hierfür gilt die Durchführbarkeit von DDoS-Angriffen. So werden Schwachstellen aktiver Netzwerkknoten, wie etwa Sicherheitslücken von Webservern [7] oder Betriebssystemen benutzt, um *Bot*-Netzwerke (kurz: *Botnet*) zu züchten. Von einem Befehlsgeber entsprechend instruiert kann ein Botnet dazu verwendet werden, die für einen DDoS-Angriff notwendige Datenflut auszulösen. Mithilfe von DDoS-Angriffswerkzeugen wie *Stacheldraht* [8] ist es sogar IT-Laien möglich kleinere Webserver zu überlasten. Des Weiteren, wie [6] aufzeigt, entwickelt sich rund um DDoS-Angriffe ein komplett neuer Wirtschaftszweig. Hier können zahlende Kunden DDoS-Dienste wie bspw. den *Lizard Stresser* in Anspruch nehmen, um gezielt Webserver (bspw. von Konkurrenten) in die Knie zu zwingen.

Die leichte Durchführbarkeit ist allerdings nur ein Aspekt, welcher den DDoS-Boom plausibel erscheinen lässt. Was Angreifern darüber hinaus entscheidend in die Hände spielt, ist die Tatsache, dass sich Betroffene oft nicht effizient gegen DDoS-Angriffe verteidigen können [9]. Dies liegt zum einen daran, dass Angreifer anormal große Datenmengen in Richtung des Opfers schicken, die schließlich das durchschnittliche Verkehrsvolumen um ein vielfaches übersteigen. Zum anderen ist die Detektion von Angriffspaketen häufig kompliziert. Die Pakete unterscheiden sich kaum oder überhaupt nicht von denen, die Anfragen realer Kunden repräsentieren. Diesen soll der Dienst natürlich nicht verweigert werden.

Es wurden bereits zahlreiche Lösungsvorschläge zur DDoS-Verteidigung mit konventionellen Methoden veröffentlicht [10]. Dennoch entwickeln Angreifer immer ausgefeiltere Techniken, um bestehende Barrieren zu umgehen. Wie in vielen sicherheitskritischen Bereichen liefern sich Aggressoren und Verteidiger auch hier einen Wettkampf, bei dem Erstere meist die Nase vorn haben. Das führt zu einer raschen Entwicklung in der DDoS-Sparte mit einer scheinbar nicht zu überblickenden Vielfalt an verschiedenen Angriffsvektoren.

Das SDN-Paradigma weist für die Verteidigung von DDoS-Angriffen hervorragende Eigenschaften auf. Durch die Trennung der *Forwarding*-Ebene von der *Control*-Ebene entsteht ein Netzwerk, bei dem eine (replizierte) Kontrollinstanz die Forwarding-Logik zugeordneter Netzwerkknoten plant und verwaltet. Router werden durch simple *SDN-Switches* ersetzt.

Im zweiten Abschnitt der Arbeit wird zunächst eine Taxonomie von DDoS-Angriffen vorgestellt und im dritten Kapitel anhand von Beispielen veranschaulicht. Der vierte Abschnitt ist konventionellen Verteidigungsmechanismen gewidmet. Analog dazu werden im fünften Abschnitt Verteidigungsmechanismen, die sich des SDN-Paradigmas bedienen, erläutert. Im sechsten Kapitel diskutieren wir die Verteidigungsansätze aus beiden genannten Welten. Hier werden Ähnlichkeiten und Unterschiede herausgearbeitet. Den Abschluss bildet Kapitel sieben, welches über die Risiken beim Einsatz von SDN aufklärt.

2. DDOS-ANGRIFFE - EINE TAXONOMIE

Es existieren zahlreiche Fachartikel, die eine Kategorisierung von DDoS-Angriffen vornehmen. So unterscheiden sowohl Douligeris und Mitrokotsa [13] als auch Specht und Lee [12] zwischen zwei verschiedenen Angriffstypen: Jene, welche das Netzwerk des Opfers verstopfen und Angriffen, deren Erfolg die Überlastung wichtiger Systemressourcen (bspw. Hauptspeicher oder CPU) zur Folge haben. Mirkovic und Reiher liefern in [10] eine wesentlich detaillierte Taxonomie. Die

doi: 10.2313/NET-2015-09-1_08

Autoren berücksichtigen in ihrem Artikel bereits Faktoren, welche sich auf den Grad der Automatisierung beim Aufbau einer Botnet-Infrastruktur beziehen. Ähnlich zu [13] werden auch hier semantische- und *Brute-Force*-Angriffe genannt. Weitere Kategorien sind: Art der Quelladresse (*spoofed* oder original), Dynamik in der Masse des Angriffsverkehrs (konstant oder variabel), Auswirkungen des Angriffs auf das Opfer, Verwundbarkeit der Angriffspakete gegenüber Paketfiltern, Dynamik in der Menge der Angriffsknoten und verschiedene Opfertypen. Asosheh und Ramezani [11] erweitern obige Taxonomie noch um die Sparte Angriffsarchitektur.
Ziel dieser Arbeit ist es herkömmliche Verteidigungsmechanismen denen gegenüber zu stellen, welche auf Basis des SDN-Paradigmas arbeiten. Daher verwendet die hier vorgestellte Taxonomie nur Kategorien, welche direkt mit einem DDoS-Angriff in Verbindung gebracht werden können. Kategorien, die etwa Vor- und Nachbereitung eines DDoS-Angriffs thematisieren werden daher nicht in die hier präsentierte Taxonomie mit aufgenommen. Im Speziellen werden die in [10, 11] aufgeführten Klassen ignoriert, welche den Aufbau von Botnets und das Schadensausmaß beim Opfer kategorisieren. Dennoch orientiert sich die in dieser Arbeit vorgestellte Taxonomie (Abbildung 1) stark an den Ausführungen von Mirkovic und Reiher. Im Folgenden sollen die einzelnen Kategorien beschrieben werden.

2.1 Architektur

Diese Kategorie gliedert DDoS-Angriffe gemäß deren Infrastruktur. Angreifer, welche die *Agent-Handler*-Architektur verwenden setzen infiltrierte Netzwerkknoten entweder als *Handler* oder als *Agent* ein. Die Agents werden zumeist ohne deren Wissen missbraucht und daher auch als sekundäre Opfer bezeichnet. Aus Gründen der Sicherheit und Anonymität werden Agents indirekt über die Handler kontrolliert. Jeder Agent wird mindestens einem Handler zugeordnet und ein Handler kontrolliert mindestens einen Agent. Um nun Befehle abzusetzen (bspw. Angriff!) kommuniziert der Angreifer mit den Handlern, welche den Befehl an ihre Agents weitergeben. Die Agents führen den eigentlichen Angriff aus, indem sie die entsprechenden Pakete in Richtung des Opfers senden [12]. So setzt beispielsweise Stacheldraht die Agent-Handler-Architektur ein [8].
Aufgrund einiger negativer Eigenschaften der Agent-Handler-Variante greifen Angreifer heutzutage vermehrt zur IRC-basierten (*Internet Relay Chat*) Architektur [14]. Dort werden anstelle von Handlern IRC-Server oder ganze IRC-Netzwerke eingesetzt. Über *IRC-Channel* kommuniziert der Angreifer mit den Bots. Dieses Vorgehen hat mehrere Vorteile. Da sich Angreifer für gewöhnlich stark frequentierte IRC-Server aussuchen, gewährleisten diese automatisch Anonymität. Befehle des Angreifers unterscheiden sich kaum vom üblichen Nachrichtenverkehr auf dem Server. Agents können daher nur schwer detektiert werden [10]. Darüber hinaus liefert ein IRC-Server bereits die benötigte Infrastruktur und Protokolle welche die Kommunikation zwischen Angreifer und Agents ermöglichen [14].
Beim *Reflector*-Angriff wird eine Infrastruktur verwendet, die der Agent-Handler-Architektur sehr stark ähnelt. Auch hier hat der Angreifer die Möglichkeit, Agents indirekt über Handler zu steuern. Der entscheidende Unterschied ist, dass die Angriffspakete nicht direkt an das Opfer gesendet werden. Vielmehr schicken Agents die Pakete indirekt über Reflector-Knoten (z.B.: *DNS-Server* oder *DNS-Resolver*) an das Opfer. Um die Weiterleitung zu ermöglichen, spoofen Agenten die Quell-IP-Adresse der Angriffspakete, sprich die eigene IP-Adresse wird innerhalb der Pakete mit der des Opfers ersetzt. Die Antwortnachrichten der Reflector-Knoten werden schließlich dem Internet Protokoll entsprechend an das Opfer adressiert. Vorteil dieser Angriffsstrategie ist zum einen, dass durch das Einflechten einer weiteren Delegationsstufe die Rückverfolgbarkeit zusätzlich erschwert wird.
Eine weitere Ausprägung des Reflector-Angriffs missbraucht bestimmte Protokolleigenschaften, wodurch die Antwortnachricht wesentlich größer ausfallen kann, als die vorangegangene Anfrage. Der Quotient (die Datengröße der Antwortnachricht dividiert durch die Datengröße der Anfragenachricht) wird auch als *Amplifikationsfaktor* bezeichnet. Ein hoher Amplifikationsfaktor hat zur Folge, dass mit einer entsprechenden Anfragemenge ein wesentlich größeres Echo generiert werden kann. Während das Netzwerk des Angreifers die geringe Anfragemasse verkraftet, wird das Netzwerk des Opfers mit künstlich aufgeblasenen Antworten konfrontiert und häufig überlastet [15].

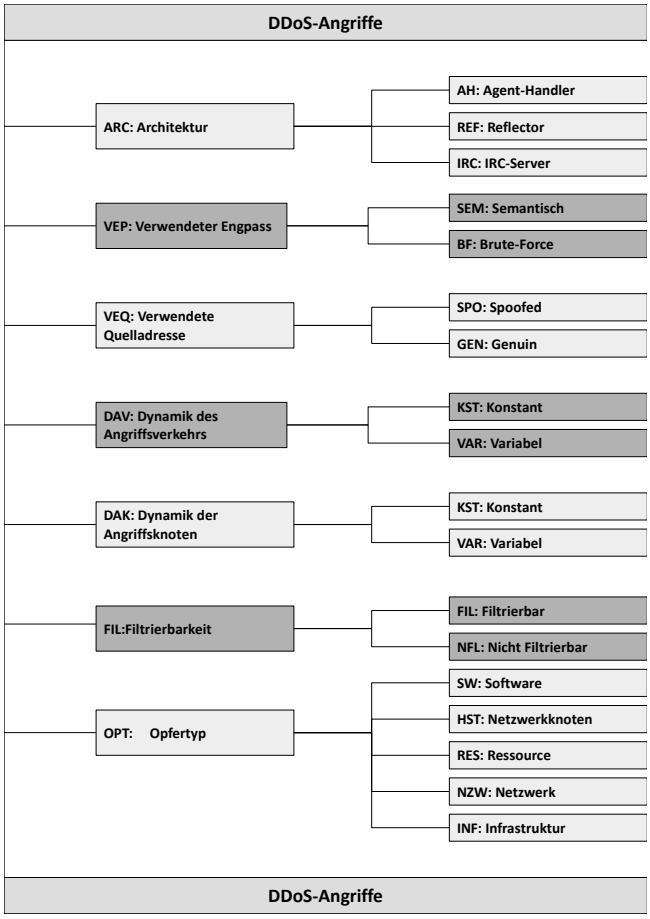

Abbildung 1: Taxonomie DDoS-Angriffe

2.2 Verwendeter Engpass

Bei *semantischen* Angriffen [10] werden gewisse Protokolleigenschaften oder Schwachstellen bzw. Fehler in der Software des Opfers missbraucht um gezielt dessen Ressourcen zu überlasten. In [12, 13] werden sie daher auch als Ressourcenerschöpfungs-Angriffe bezeichnet. Ein prominen-

doi: 10.2313/NET-2015-09-1_08

ter Vertreter dieser Klasse ist der *TCP-SYN*-Angriff. Dabei wird der drei-Wege-Handshake von TCP missbraucht. Jede TCP-Verbindungsanfrage (SYN Nachricht) verschlingt Speicherplatz. Eine große Menge von SYN Paketen führt zur Überlastung des Speichers.
Das Pendant zu semantischen Angriffen sind *Brute-Force*[10]- oder Bandbreitenerschöpfungs-Angriffe [12, 13]. Hierbei sendet der Angreifer eine große Menge unverfälschter Pakete (mit Ausnahme der Quell-IP-Adresse) in Richtung des Opfers um dessen Netzwerk zu überlasten. Zu dieser Kategorie zählt beispielsweise der oben erklärte Reflector-Angriff. Häufig werden aber auch simple UDP- oder ICMP-Pakete verwendet um das Opfernetzwerk zu fluten. Die Diskrepanz zwischen Brute-Force-Angriffen und semantischen Angriffen ist gering. Als wichtiger Unterschied ist hervorzuheben, das letztgenannte durch Protokoll- bzw. Softwarenachbesserungen abgewehrt oder zumindest geschwächt werden können. Semantische Angriffe, deren missbrauchte Lücke geschlossen wurde fallen automatisch in die Brute-Force Klasse. Abschließend ist noch festzuhalten, dass semantische Angriffe durch den Missbrauch von Schwachstellen ein wesentlich geringeres Datenvolumen benötigen als Brute-Force-Angriffe, um vergleichbare Effekte zu erzielen.

2.3 Verwendete Quelladresse

Es existieren die beiden Möglichkeiten Angriffspakete entweder mit der genuinen oder einer gespooften Quelladresse abzuschicken. Die Vorteile für gefälschte Quelladressen liegen klar auf der Hand. Zum einen kann ein Agent vom Opfer nicht durch die Quell-IP-Adresse in den Angriffs-Paketen identifiziert werden. Außerdem kann ein einziger Agent verschiedene Quell-Adressen für den Angriff verwenden, was Paketfiltern die Arbeit erheblich erschwert. Einige Angriffsmuster, wie etwa der Reflector-Angriff, würden ohne IP-Adress-Spoofing gar nicht funktionieren. Originale IP-Quelladressen werden nur eingesetzt, wenn das Spoofen derselben nicht möglich ist. Gründe hierfür sind etwa die Notwendigkeit von Administratorrechten bzw. mangelhafte Unterstützung durch die Betriebssystemebene [10].

2.4 Dynamik des Angriffsverkehrs

Die meisten DDoS-Angriffe erzeugen konstante Verkehrsraten. Der aktuelle Trend geht in Richtung länger andauernder Angriffe mit geringerer Bandbreite [5]. Die starke und plötzliche Abweichung vom regulären Datenvolumen und die Konstanz im Datenfluss ermöglicht es Verteidigern schnell zu reagieren, sprich Angreifer zu detektieren und Paketfilter entsprechend anzupassen. Daher werden viele Angriffe mit einer variablen Datenrate ausgeführt. Ansteigende oder schwankende Volumina sind mögliche Verfahrensweisen [10]. Ansari und Shevtekar [16] beschreiben den (aus ihrer Sicht) perfekten Angriff, welcher konstante und variable Datenraten kombiniert.

2.5 Dynamik der Angriffsknoten

Bei DDoS-Angriffen werden Botnets verwendet, die aus einer großen Menge verschiedener Netzwerkknoten bestehen. Zum Angriffszeitpunkt kann eine Teilmenge von Bots zeitgleich oder verschiedene Teilmengen zeitversetzt aktiviert oder deaktiviert werden. Ständig wechselnde Angreifergruppen erschweren die Rückverfolgung und Detektion der Angreifer zusätzlich [10].

2.6 Filtrierbarkeit

Viele Angriffstypen verwenden Pakete deren äußeres Erscheinungsbild von regulären Paketen abweicht (s. semantische Angriffe). Wenn eine solche Abnormität detektiert wird, können Filterregeln erstellt werden, die speziell auf die entsprechenden Pakete zugeschnitten sind. Diese Angriffe sind demzufolge durch Paketfilter abwehrbar. Im Gegensatz dazu existieren DDoS-Angriffe, die unverfälschte Datenpakete (mit Ausnahme der Quell-IP-Adresse) benutzen (s. Brute-Force-Angriffe). Obwohl diese Teil eines Angriffs sind, können sie von regulären Anfragen legitimer Servicenutzer kaum unterschieden werden und sind damit nur schwer aussiebbar [10].

2.7 Opfertypen

Wie bereits beschrieben existieren Angriffe, die speziell gegen bestimmte Softwareanwendungen gerichtet sind. Ziele könnten etwa ein Webserver oder eine DNS-Anwendung sein. Je nach Ressourcenzuteilung durch das Betriebssystems wird das angegriffene Gerät vollständig überlastet. Obwohl die Auswirkungen in diesem Fall zwar weitreichender sind, kann der Angriff gleichzeitig schneller detektiert werden. Sind der angegriffenen Anwendung lediglich Anteile der gesamten Systemressourcen zugewiesen, können parallel laufende Anwendung ungestört weiter arbeiten.
Eine weitere Klasse von DDoS-Angriffen haben die Überlastung des gesamten Netzwerkknotens zum Ziel. Auch hier ist der TCP-SYN-Angriff ein repräsentatives Beispiel. Ressourcenangriffe gehen gegen Knoten, die einen für das Netzwerk essentiellen Dienst anbieten. Wird bspw. der *Gateway Router* oder lokale DNS-Server außer Gefecht gesetzt, so wird das gesamte Netzwerk gelähmt oder sogar handlungsunfähig gemacht. Da solche Netzwerkknoten in der Regel gut geschützt (z. B. durch Replikation) sind oder über leistungsstarke Systemkomponenten verfügen, benötigt der Angreifer hier einen starken DDoS-Angriff. Diese lassen sich meist einfacher detektieren. Netzwerkangriffe sind ebenfalls gegen ein gesamtes Netzwerk gerichtet. Der dabei erzeugte Datenverkehr überlastet das Netzwerk (bzw. die Bandbreite der Internetleitung), was den Transport legitimer Nachrichten verhindert. Der dafür erforderliche Angriffsverkehr kann auf Grund des großen Datenvolumens einfacher detektiert werden.
Die letzte Unterkategorie der Opfertypen betrifft die gesamte Infrastruktur des Internets. Ähnlich wie bei Ressourcenangriffen im kleinen Rahmen attackiert der Angreifer essentielle Dienste und Systeme des Internets. Dazu gehören unter anderem *Backbone-Router* oder Root-DNS-Server [10].

3. AKTUELLE DDOS-ANGRIFFE UND DEREN EINORDNUNG IN DIE TAXONOMIE

Wie bereits in der Einleitung beschrieben, existiert die Problematik von DDoS-Angriffen nach wie vor. Aktuelle Beispiele belegen dies. Doch auch in der Vergangenheit wurden bereits zahlreiche DDoS-Angriffe ausgeführt. Eine Zuordnung der Angriffe zu der Taxonomie wird in Tabelle 1 gezeigt.

DNS: So waren Oktober 2002 die Root-DNS-Server Ziel eines solchen Attentats (Herkunft unbekannt). Mit einer *ICMP-Ping*-Flut wurden 9 der 13 DNS-Root-Server kurzzeitig lahm gelegt. Eine Messung ergab eine Datenrate von circa 80 Mb-

doi: 10.2313/NET-2015-09-1_08

Tabelle 1: Zuordnung der Angriffe in die DDoS-Taxonomie

Ziel	ARC	VEP	VEQ	DAV	DAK	FIL	OPT
DNS	-	BF	-	KST	-	FIL	INF
SCI	-	BF	GEN	VAR	KST	FIL	HST
SPM	REF	BF	SPO	VAR	KST	FIL	NZW
CLF	REF	BF	SPO	VAR	KST	FIL	NZW
PSN	IRC	SEM	SPO	-	-	NFL	HST
GIT	-	SEM	GEN	KST	VAR	NFL	SW

ARC:Architektur **VEP**:Engpass **VEQ**:Quelladresse
DAV:Angriffsverkehr **DAK**:Angriffsknoten
FIL:Filtrierbarkeit **OPT**:Opfertyp

ps. Der Angriff dauerte in etwa eine Stunde, wobei der normale Internetnutzer davon kaum tangiert wurde. Um den Angriff abzuschwächen wurden vorgeschaltete Paketfilter instruiert IMCP-Pakete zu blocken [17, 31].
SCI: Im Januar 2008 war *Scientology's* Internetauftritt Ziel eines DDoS-Angriffs. Zu der Tat bekannten sich *Anonymous*-Aktivisten. Ausgeführt wurde die Tat mittels mehrerer Angriffswellen. Hierfür wurde das DDoS-Tool LOIC (*Low Orbit Ion Cannon*) verwendet, welches UDP- und TCP-Pakete generiert. Mit Spitzenwerten von 220 Mbps hat es Anonymous geschafft den Dienst der Webseite zu blockieren. Seitens Scientology sind keine nennenswerten Abwehrmaßnahmen bekannt [32, 33, 34]. Da die Datenrate selbst für einen DDoS-Angriff dieser Zeit nicht überdurchschnittlich hoch war, und LOIC-Angriffe mittels einfacher Firewall-Regeln abzuwehren sind [35], ist anzunehmen, dass Scientology keine Abwehrmaßnahmen bereitstellte. Als Reaktion auf den Angriff nahm Scientology die DDoS-Abwehrdienste von *Prolexic Technologies* in Anspruch.
SPM: Der nächste vorgestellte DDoS Angriff spielt größentechnisch in einer weit höheren Liga. Im März 2013 wurde die Webseite der anti-spam Organisation *Spamhaus* von einem DDoS-Angriff (Herkunft unbekannt) erschüttert und die Webseite unerreichbar gemacht. Die Angreifer benutzten mehrere verschiedene Angriffsvektoren. Der mit Abstand größte Datenverkehr wurde mit einem DNS-Reflector-Angriff erzeugt. Hierfür missbrauchten die Täter über 30000 offene *DNS Resolver*. Ein weiterer Bestandteil der Angriffsmasse wurde mit *ACK-Reflector*-Angriffen generiert. Auch hier sind keine Verteidigungsmaßnahmen seitens Spamhaus bekannt. Diese wären jedoch ohnehin unwirksam gewesen, da der Angriffsverkehr mit Spitzenwerten bis zu 75 Gbps die Internetleitung der Webseite komplett überlastete. Einen Tag später kontaktierten die Spamhausbetreiber den CDN-Anbieter (*Content Delivery Network*) *Cloudflare*. Dieses Unternehmen verfügt über 23 Datenzentren die weltweit verteilt sind. Mithilfe der Adressierungsart *Anycast* kann der Datenverkehr auf die verschiedene Datenzentren verteilt werden, die wiederum Duplikate der Webseite speichern. So wird der Angriff zwar nicht abgeschwächt, aber die Last auf verschiedene Netzwerkknoten verteilt. Durch die Unterstützung von Cloudflare war der Internetauftritt von Spamhaus kurz darauf auch wieder erreichbar [36]. Eine detaillierte Analyse des Angriffs kann in [37] gesichtet werden.
CLF: Fast genau ein Jahr später berichteten diverse Quellen von einem DDoS-Angriff der alle zuvor erlebten Angriffe in den Schatten stellte. Ziel des Angriffs war ein Cloudflare Kunde. Die Täter verwendeten einen NTP-Reflector-Angriff (*Network Time Protocol*). Hierbei wird in der Anfrage der *MONLIST* Befehl verwendet. Der NTP-Server Antwortet mit einem Paket, das die IP-Adressen der letzten 600 Anfragen enthält. Der Amplifikationsfaktor beträgt dabei 206. Mit dieser Vorgehensweise schafften es die Angreifer einen Datenverkehr von über 400 Gbps zu generieren. Dafür wurden 4529 verschiedene NTP-Server benutzt. Cloudflare konnte den Angriff mit den oben beschriebenen Methoden erfolgreich verteidigen [38].
PSN: Im Dezember des selben Jahres gelang es der Gruppe *Lizard Squad* das PSN (*PlayStation Network*) mehrere Stunden lahmzulegen [40]. Über den genauen Tathergang ist nach bestem Wissen des Autors nichts bekannt. Aller Wahrscheinlichkeit nach wurde ein *TCP-Flag-DDoS*-Angriff ausgeführt [39]. Quellen zu Datenraten oder zu etwaigen Verteidigungsmaßnahmen Seitens Sony's konnten nicht gefunden werden. Dennoch ist grob bekannt, wie Lizard Squad generell operiert. Die Gruppe verwendet ein Botnet, das aus mit *Malware* infizierten Routern besteht. Diese können schließlich für einen DDoS-Angriff instrumentalisiert werden [41].
GIT: Der letzte hier vorgestellte DDoS-Angriff betraf den Hosting-Dienst *GitHub*. Der Angriff stammte aus China. Dabei wurde Internetnutzern, die bestimmte Webseiten aufriefen, bösartiger Javascript Code gesendet. Das Skript fordert den Browser in einer Endlosschleife auf, zwei spezielle Internetseiten der GitHub-Domäne aufzurufen. Quellen zu Datenraten oder zu etwaigen Verteidigungsmaßnahmen sind auch hier nicht bekannt [42].

4. KONVENTIONELLE VERTEIDIGUNGSMAßNAHMEN

Bei DDoS-Verteidigungsmaßnahmen verhält es sich ähnlich wie bei DDoS-Angriffsvektoren: Es existiert eine kaum zu überblickende Vielfalt verschiedener Ansätzen, Ideen und tatsächlichen Implementierungen. Um dieses Kapitel übersichtlich zu gestalten werden die Verteidigungsansätze in drei verschiedene Kategorien unterteilt. Genauer gesagt wird hierbei nur die Standortzugehörigkeit berücksichtigt. Es existieren drei Möglichkeiten eine DDoS-Verteidigung zu platzieren:

1) Auf der Netzwerkseite des Opfers
2) Im Kernnetzwerk, welches Angreifer und Opfer verbindet
3) Im Quellnetzwerk des DDoS-Angriffs

Im Folgenden sollen für jede Kategorie Verteidigungsmaßnahmen vorgestellt werden.

4.1 Verteidigungsmaßnahmen im Netzwerk des Opfers

Hier angesetzte Verteidigungsmechanismen sind der geballten Wucht von DDoS-Angriffen vollständig ausgeliefert. Anstelle von präventiven Maßnahmen bleibt nur die Reaktion. Wie das Beispiel von Spamhaus [36] zeigt, ist die Verteidigung zu diesem Zeitpunkt oft bereits zu spät. Dennoch existieren Ansätze, die DDoS-Angriffe detektieren und verteidigen können.
Wang et al. [43] stellen in ihrer Arbeit eine Methode vor, die DDoS-Angriffe mit gefälschten IP-Adressen erkennt: *Hop Count Filtering* (HCF). Der Ansatz verwendet das *TTL*-Feld im Header von IP-Paketen. Dabei wird die Tatsache benutzt, dass zwischen Angreifer und Opfer und gespoofter IP-Adresse und Opfer meist unterschiedlich viele Netzwerk-

knoten (*Hops*) liegen. Sprich, viele gespoofte IP-Packete besitzen einen TTL-Wert der nicht zur gefälschten Adresse passt (Betriebssysteme verwenden Stan-dard-Werte für den initialen TTL-Wert). HCF erstellt während einer Lernphase eine *IP-zu-Hop-Count* (IP2HC) Tabelle, in der Netzwerkpräfixe bereits eingegangener IP-Pakete mit den korrespondierenden *HopCounts* gespeichert werden. Wenn ein Angriff detektiert wird (etwa über die stark angestiegene Datenrate) wechselt HTF in den Filtermodus und verwirft mit Hilfe der IP2HC-Tabelle gespoofte Pakete. Experimente zeigen, dass der HCF-Algorithmus gespoofte Pakete mit einer Wahrscheinlichkeit von ca. 90% erkennt. Voraussetzung ist eine ausgiebige Lernphase. Etwa 10% der Pakete werden als *False-Positives* erkannt.
Eine weitere Strategie ist die Verwendung von *SYN-Cookies*. Diese Methodik wird im RFC 4987 präsentiert [44] und bezieht sich auf einen Artikel von Bernstein [45]. SYN-Cookies können als Verteidigungsmaßnahme gegen SYN-Angriffe eingesetzt werden. Die Cookies werden in das *Sequenznummernfeld* von TCP-Paketen gespeichert. Da (bis auf die Länge von 32 Bit) keine konkreten Vorgaben für die initiale Sequenznummer existieren, kann dieses Feld auch dazu verwendet werden Verbindungsinformationen wie etwa Quell--IP-Adresse, Quell-Port-Nummer, Ziel-IP-Adresse und Ziel-Port-Nummer zu speichern. Da diese Informationen zu groß sind, werden Sie mittels einer kryptographischen *Hash*-Funktion auf die gewünschte Länge abgebildet. Diese Verbindungsinformationen müssten sonst in einer Tabelle verwaltet werden. Der Server antwortet(SYNACK) mit dieser speziell gewählten Sequenznummer. Da die Sequenznummer der folgenden Nachricht nur um 1 größer ist (Ausnahme: *Piggyback ACKs*), kann der Server das SYN-Cookie verifizieren. Das Wegfallen der Verbindungstabelle verhindert Speicherüberläufe und schwächt daher die Auswirkungen von SYN-Angriffen stark ab.

4.2 Verteidigung im Kernnetzwerk

DefCom ist eine Verteidigungsstrategie die von Oikonomou et al. [47] entwickelt wurde. Dabei verwenden sie bereits etablierte Mechanismen der DDoS-Detektion und -Verteidigung. Das Verfahren operiert nicht ausschließlich mit Knoten des Kernnetzwerks (aber größtenteils), sondern setzt auch Entitäten des Quell- und Opfernetzwerks ein. Alle beteiligten Netzwerkknoten bilden ein dynamisches *Overlay*-Netzwerk. Den Knoten werden verschiedene Rollen zugeteilt: 1) *Classifier*-Knoten werden dazu eingesetzt Angriffspakete zu detektieren und zu markieren. 2) *Rate-Limiter*-Knoten die Angriffe abschwächen (z.B.: Verwerfen von Paketen) 3) *Alarmgeber*-Knoten detektieren DDoS-Angriffe und propagieren diese im DefCom-Netzwerk.
Wobei 1) und 2) ausschließlich im Kernnetzwerk implementiert werden können, ist die Installation von Alarmgebern auch im Opfer- bzw. Quellnetzwerk möglich. Wenn nun ein DDoS-Angriff detektiert wird, aktiviert der Alarmgeber die beiden anderen Knotentypen (1 und 2). Aktive Classifier markieren Angriffspakete, die dann von Rate-Limitern entsprechend behandelt (gemäß der eingesetzten Policy) werden.

4.3 Verteidigung im Quellnetzwerk

Das Quellnetzwerk ist theoretisch der effektivste Ansatzpunkt für eine DDoS-Verteidigung, da Angriffspakete schon gefiltert werden können, bevor sie überhaupt ins Netz gelangen.
RFC 2827[48] spezifiziert Richtlinien für den Einsatz von Filterregeln um Angriffe mit gespooften IP-Adressen zu verhindern. Die wichtigste Maßnahme ist das Blockieren von ausgehenden Paketen, deren Quell-IP-Adresse nicht aus dem Quell-Subnetz stammt.
D-WARD [49] ist ein Paketfilter, der auf dem Gateway-Router des Quellnetzwerks installiert wird. Dieser überwacht eingehenden und ausgehenden Verkehr und speichert Statistiken (Anzahl gesendeter Pakete, Anzahl gesendeter Bytes, etc.) zu den *Flows*. Anomalien (z.B.: Zu hohe Senderate von Paketen mit einer IP-Adresse als Ziel) in den Flows werden detektiert und eingestuft. Flows die zu einer als Angriff eingestuften Anomalie gehören werden blockiert (bzw. die Verkehrsrate wird limitiert). Mit dieser Vorgehensweise ist D-WARD in der Lage ausgehende TCP-, UDP- und ICMP-Angriffe zu blockieren. Sowohl in durchgeführten Experimenten als auch unter realen Bedingungen erzielt D-WARD sehr gute Ergebnisse.

5. VERTEIDIGUNGSMECHANISMEN AUS DER WELT DES SDN

Heutzutage etablierte Netzwerkgeräte werden mit einem vorinstallierten Software-Satz ausgeliefert, der wenig Konfigurationsspielraum für den Endkunden lässt. Von außen betrachtet sind diese Geräte autonome Blackboxen, die Forwarding- und Kontrolllogik in sich vereinigen und mithilfe weniger Stellschrauben an die eigenen Bedürfnisse angepasst werden können. Diese Herangehensweise ist in vielerlei Hinsicht problematisch. Zum einen ist die Schnittstellenlandschaft sehr vielseitig. Jeder Hersteller verwendet proprietäre Software für seine Netzwerkgeräte. Daher existiert keine einheitliche Schnittstellensprache, mit deren Hilfe die Geräte konfiguriert werden können.
Des weiteren, um die Netzwerkgeräte zu konfigurieren steht Administratoren nur ein vorgefertigter Befehlssatz zur Verfügung. Inzwischen gibt es Lösungen die das Ausführen von Scripten erlauben [18]. Nichtsdestotrotz müssen auch diese in einer systemnahen Programmiersprache angefertigt werden. Das Umsetzen abstrakter Aufgabenstellungen (wie bspw. Netzwerk-Policies) ist daher ein schwieriger und fehleranfälliger Prozess [20]. Die Vereinigung von Kontroll- und Forwardinglogik in einem geschlossenen System bringt weitere Herausforderungen mit sich. Obwohl Netzwerkgeräte mithilfe von Routingprotokollen Informationen austauschen, entscheidet die Kontrollschicht jedes Geräts individuell über die Verfahrensweise mit eingehenden Paketen. Dadurch gestaltet sich das Umsetzen von Änderungen auf globaler Ebene schwierig. Man spricht daher auch von der *Verknöcherung* [19] des Internets.
Wie bereits erwähnt tauschen Netzwerkgeräte Routinginformationen um möglichst optimale Routingentscheidungen auf Kontrollebene zu treffen. Diese lassen sich allerdings nur fällen, wenn jeder Router eine globale Netzwerksicht besitzt. Netzwerke sind allerdings hochdynamische Systeme deren Stellschrauben sich ständig verändern. Solche Änderungen werden durch Routingprotokolle propagiert. Auf globaler Ebene skaliert diese Verfahrensweise allerdings nur langsam.
SDN löst viele dieser Probleme durch die Trennung von Forwarding- und Kontroll-Ebene. SDN-Switches werden lediglich für die Weiterleitung von Paketen eingesetzt und sind über wohldefinierte Schnittstellen zugreifbar. Die Kon-

trolleinheit befindet sich physisch getrennt von den Switches auf einem anderen Netzwerkgerät. Über diese zentrale Instanz werden alle Switches gesteuert. Genauer gesagt bietet eine programmierbare Schnittstelle der Kontrolleinheit Nutzern die Möglichkeit, das konkrete Verhalten von Switches zu manipulieren. Die Schnittstelle wird von einem Netzwerk-Betriebssystem bereitgestellt. Für dieses Betriebssystem können Anwendungen erstellt werden, welche die Steuerung des gesamten Netzwerks auf einem sehr abstraktem Niveau erlauben [20]. Es existieren bereits konkrete Implementierungen für Netzwerk-Betriebssysteme. Die bekannteste Ausprägung ist *NOX* [21]. *OpenFlow* [19] ist der de facto Standard für das Protokoll das die Kommunikation zwischen Kontrollgerät und Switch reglementiert.

5.1 DDoS-Detektion im Kernnetzwerk mithilfe von Self Organizing Maps (SOM)

Rodrigo, Edjard und Passito [22] präsentieren eine Methode zur effizienten DDoS-Detektion im Kernnetzwerk. Dabei verwenden sie eine NOX-Kontrolleinheit und OpenFlow-Switches. Letztere verwalten Statistiken jedes aktiven Flows: 1) Die Anzahl der empfangenen Pakete 2) Die Menge der empfangenen Daten (in Bytes) 3) Die Zeitdauer die sich der Flow bereits in der Flow-Tabelle befindet. NOX fordert in regelmäßigen Abständen die Tabellendaten aller Switches an. Somit befinden sich die Flow-Information des gesamten Netzwerks an einer zentralen Stelle. Ein *Feature Extractor* erkennt schließlich Merkmale, welche auf einen DDoS-Angriff hindeuten können. Diese Merkmale werden mithilfe von SOM [23] ausgewertet und kategorisiert. Durchgeführte Experimente zeigen mit ca. 99% eine hohe Zuverlässigkeit bei der Detektion von DDoS-Angriffen (TCP/SYN-Flood, UDP-Flood, ICMP-Flood). Dabei liegt die Rate von False-Positives unter 1%.

5.2 DDoS-Detektion im Quellnetzwerk mithilfe von Frequent Sets Analyzern (FSA)

Mehdi et al. [24] beleuchten den Einfluss des SDN-Paradigmas auf die Netzwerksicherheit im Allgemeinen. In ihrer Arbeit beschreiben sie zudem ein DDoS-Detection System. Anders als beim vorangehenden Ansatz dient die hier vorgestellte SDN-Anwendung zur Detektion von DDoS-Angriffen im Quellnetzwerk. Dafür wird die Idee von FSA herangezogen. FSA ist eine Methode des Datamining zur Detektion von Anomalien. In der Arbeit wird FSA zur Detektion von verdächtigen Verkehrsflüssen verwendet. Essentielle Voraussetzung ist die Möglichkeit des Zugriffs auf Flow-Informationen des gesamten Netzwerks. Genau das leistet SDN. Switches senden ihre Flow-Tabellen an die Kontrollinstanz. Diese analysiert die empfangenen Informationen und filtert *Frequent-Data-Sets* heraus. Das sind Teilmengen der Flow-Tabellen die auf DDoS-Verkehrsflüsse hindeuten. In der Arbeit werden keine konkreten DDoS-Angriffe genannt oder Experimente durchgeführt, welche die Effizienz der vorgestellten Methode belegen.

5.3 DDoS-Detektion und Verteidigung in Echtzeit im Kernnetzwerk

Krishnan und Durrani präsentieren in [25, 26] ein System, das DDoS-Angriffe in Echtzeit erkennt und verteidigt. Dabei konzentrieren sie sich auf lang andauernde Angriffe mit großen Datenraten (*Long-Lived Large Flows* bzw. LLL-Flows), die Protokolle der Transport- und Netzwerkschicht missbrauchen. Unter Verwendung der *sflow*-Technologie [27] senden SDN-Switches Flow-Informationen an *inMon sFlow-RT*-Kontrolleinheiten [28]. Aus den Flow-Informationen werden Metriken generiert, die eine Einstufung der Flows zulassen. Eine Anwendung auf dem SDN-Steuergerät bestimmt Regeln für den Umgang mit verdächtigen Flows (z.B.: *Drop, RateLimit*) und schreibt die Regeln mittels OpenFlow in die Flow-Tabellen der SDN-Switches. Das System wird mit einem NTP-Reflector-Angriff getestet und ist in der Lage, den Angriff binnen weniger Sekunden zu detektieren. Die Angriffs-Pakete werden von den SDN-Switches im Kernnetzwerk verworfen und gelangen gar nicht erst bis zum Netzwerk des Opfers.

5.4 DDoS-Detektion im Netzwerk des Opfers und Verteidigung im Kernnetzwerk

Saray et al. beschreiben in [29] ein System bei dem ein ISP (*Internet Service Provider*)und dessen Endkunden eng zusammen arbeiten. Beide Netzwerke setzen SDN-Steuergeräte ein. Die Switches des ISP-Netzwerks, die eine direkte Verbindung mit dem Kunden-Netzwerks besitzen (Egress-Switches) füttern die Kontrolleinheit des Kunden ständig mit neuen Informationen aus deren Flow-Tabellen. Mit diesen Informationen kann das Steuergerät des Kunden verdächtige Informationen detektieren (etwa unter Verwendung von [30]). Entdeckungen dieser Art berichtet die Kontrolleinheit des Kunden and das Steuergerät des ISP. Dadurch, dass die Angriffs-Pakete über das ISP-Netzwerk zu den Kunden gelangen, existieren in den Flow-Tabellen aller involvierter Switches Tabelleneinträge, welche die Angriffs-Flows repräsentieren. Als Konsequenz auf die Meldung des Kunden markiert das Steuergerät des ISP die verdächtigen Flows. Pakete die zu einem so markierten Flow gehören werden vom ISP genauer untersucht. Ein entsprechendes Modul siebt Pakete die zu einem DDoS-Angriff gehören aus und leitet legitime Pakete zurück in das Kernnetzwerk.

6. EIN VERGLEICH HERKÖMMLICHER UND SDN-BASIERTER VERTEIDIGUNG

Die Diskussion soll gemäß der im 4. Kapitel vorgenommenen Kategorisierung unterteilt werden.

6.1 Verteidigung im Quellnetzwerk

Generell gilt: Je näher das Verteidigungssystem an der Quelle des Angriffs sitzt, desto besser kann ein DDoS-Angriff auch verteidigt werden. Im Extremfall sind bereits Verteidigungsanwendungen auf dem Netzwerkgerät installiert, das die Pakete entsendet. Wahrscheinlicher ist allerdings, dass solche Anwendungen im Knotenpunkt angebracht sind, der das interne Netzwerk mit dem Internet verbindet. Dazu wurden im Punkt 4.3 bereits etablierte Verteidigungsstrategien vorgestellt, namentlich Egress-Filter und D-WARD. Die in Kapitel 5.2 vorgestellte SDN-Variante verfährt ähnlich. SDN-Switches im Quellnetzwerk verwalten Flow-Tabellen die sie in regelmäßigen Abständen Nachrichten an die Kontrolleinheit senden. Dadurch besitzt das Steuergerät eine globale Sicht über das Quellnetzwerkes. Allerdings sitzt der bei D-WARD verwendete Router an der Schnittstelle zum Internet und leitet alle Pakete des internen Netzwerks weiter. Damit ist die globale Sicht auch hier gegeben. Die Detek-

doi: 10.2313/NET-2015-09-1_08

tionsalgorithmen und Verteidigungsmaßnahmen sind dann rein theoretisch austauschbar. Sprich D-WARD könnte auch FSAs zur Detektion von Anomalien verwenden während der SDN-Controller Egress-Filtering einsetzen könnte.
Anhand dieses Beispiels kann gefolgert werden, dass das SDN-Paradigma bei einer Verteidigung auf der Opferseite seinen größten Vorteil nicht ausspielen kann. Herkömmliche Verteidigungsstrategien verfügen ebenfalls über eine globale Netzwerksicht. Daher bringt das SDN-Paradigma hier keine bahnbrechenden Neuerungen. Ein Vorteil von SDN könnte allerdings die einfache Portierbarkeit der Verteidigungsanwendungen sein. Aufgrund standardisierter Protokolle (OpenFlow) und Schnittstellen (NOX) könnte eine solche Anwendung unabhängig von Geräteherstellern auf verschiedenen Systemen installiert werden.

6.2 Verteidigung im Opfernetzwerk

Werden Angriffe gegen die Bandbreite gefahren, ist die Verteidigung auf der Opferseite oft schon zu spät (s. Spamhaus [36]). Lässt man diesen Angriffsvektor außer Acht, existiert noch eine weitere große Gefahr: IP-Spoofing. Bei Angriffen dieser Art steht das Opfer vor einem Dilemma. Zum einen möchte ein Serviceanbieter seine Kunden bedienen, zum anderen sollen Pakete mit gespoofter IP-Adresse geblockt werden. Die Detektion von Angriffen mit gespoofter IP-Adressen ist normalerweise nicht die Schwierigkeit. Diese kündigen sich in Form eines plötzlichen, massiven Anstiegs der Datenrate von selbst an. Die Problematik ist viel mehr: Wie können Pakete mit gespooften IP-Adressen von legitimen Paketen realer Kunden unterschieden werden? HCF (in 4.1) stellt hierfür einen Lösungsansatz parat. Der Algorithmus verwendet Unstimmigkeiten bei den HopCounts eingehender Pakete zur Detektion von Spoofing-Angriffen. Der Ansatz weist allerdings einige Schwachstellen auf. Beispielsweise könnten Angreifer IP-Adressen innerhalb des selben Subnetzes verwenden. Der HopCount wäre bei solchen Paketen stimmig. Mit SYN-Cookies werden gespoofte Pakete zwar nicht detektiert, aber die negativen Auswirkung von SYN-Angriffen (welche oft im Zusammenhang mit gespooften IP-Adressen verwendet werden) auf das Opfer abgeschwächt.
Im Opfernetzwerk allein kann das SDN-Paradigma nicht viel zur Verbesserung der momentanen Situation beitragen. Ähnlich wie bereits im Quellnetzwerk liefert das SDN-Paradigma auch hier nicht den Vorteil der globalen Netzwerksicht. Gegen Brute-Force-Angriffe bzw. IP-Spoofing kann die neue Technologie auf der Opferseite nichts ausrichten. Diese Angriffe müssen bereits vorher (im Kernnetzwerk oder bei der Quelle) vereitelt werden.
In Kapitel 5.5 wird ein hybrider Ansatz (mit Verteidigungskomponenten im Opfer- und Kernnetzwerk) beschrieben. Egress Switches füttern das Steuergerät des Kunden kontinuierlich mit Flow-Informationen. Mit diesen Informationen kann der Kunde verdächtige Flows detektieren und dem ISP melden. Der ISP filtert die verdächtigen Pakete und schwächt den Angriff damit ab. Dieser Ansatz funktioniert bei Angriffen mit "wenigen"(z.B.: 10000) Quellen, die jeweils große Datenmengen generieren. Ein NTP-Reflection-Angriff passt in das Schema. Damit lassen sich also Angriffe gegen die Bandbreite des Opfers verhindern. Allerdings ist nur schwer vorstellbar, wie dieses Verteidigungssystem SYN-Angriffe mit gespooften IP-Adressen verhindern soll. Schließlich erzeugen diese Millionen verschiedene Flows.

6.3 Verteidigung im Kernnetzwerk

In 4.2 wird DefCom beschrieben, welches ein Overlay-Verteidigungs-Netzwerk aus Knoten des Quell-, Kern- und Opfernetzwerks erstellt. Dies erfordert zum einen die Zusammenarbeit von Kunden mit ihrem ISP, was nicht vorausgesetzt werden kann. Zweitens müssten Netzwerkrouter mit neuer Funktionalität versehen werden. Zu guter Letzt werden neue Protokolle benötigt, welche die Interaktion der beteiligten Parteien definieren. Soviele Neuerungen zu etablieren ist in einem Netzwerk mit größtenteils vertikal integrierten Routern schwierig.
Die Gründe hierfür werden weitestgehend im 5. Kapitel erläutert: Für jede Routersoftware werden individuelle Lösungen benötigt. Neue Sicherheitsprotokolle müssen entwickelt und eingebettet werden. Durch die hochverteilte Infrastruktur sind Änderungen auf globaler Ebene nur mühsam umsetzbar. Selbst wenn DefCom oder ähnliche Strategien in die Realität umgesetzt werden könnten, enstehen dadurch weitere Probleme. Neue Technologien sind für gewöhnlich verletzlich (z.B.: DNSSEC [50]). Um deren Widerstandsfähigkeit zu stärken müssen Wartungs- und Nachbesserungsarbeiten durchgeführt werden. Dieser Entwicklungsprozess findet im heutigen, starren Internet nur sehr langsam statt. Sind die Verteidigungsstrategien veraltet, müssten unter größten Anstrengungen neue Konzepte implementiert werden. Diese Beschreibung lässt erahnen, weshalb eine grundlegende Änderung der konventionellen Architektur vorteilhaft sein könnte.
Im 5. Kapitel werden SDN-Verteidigungssysteme vorgestellt, die im Kernnetzwerk operieren. Lässt man deren Feinheiten außer Acht und betrachtet stattdessen die grundlegende Architektur und Funktionsweise, so erkennt man die Ähnlichkeit der Ansätze. SDN-Switches im Kernnetzwerk senden Statusberichte an einen zentralen Controler. Dieser detektiert mithilfe installierter Softwareanwendungen Anomalien. Entsprechend der zugrundeliegenden Policies sendet die Kontrolleinheit Befehle an die Switches, wonach verdächtige Flows blockiert oder eingeschränkt werden können. So eine Architektur kann durch Alarmgeber auf der Seite des Opfernetzwerks unterstützt werden.
Unter einem noch abstrakteren Blickwinkel existiert also ein zentraler Befehlsgeber, welcher eine globale Netzwerksicht besitzt und mithilfe von Softwareanwendungen den Netzwerkfluss analysieren und steuern kann. Diese Architektur ist sehr flexibel, da Detektions- und Steuerungsalgorithmen durch einfache Softwareupdates verbessert und gewartet werden können. Auch ist die Installation neuer Verteidigungsstrategien ist wesentlich unkomplizierter. Die entsprechende Anwendung kann in einer hohen Programmiersprache entwickelt und auf dem betreffenden Controler installiert werden. Hier werden keine Individuallösungen benötigt, welche auf jedem Router installiert werden müssen.

7. SDN-CONTROLER - SINGLE POINT OF FAILURE

Neben vielen bereits beschriebenen Vorteilen des SDN-Paradigma existieren natürlich auch Nachteile. Das größte Sicherheitsrisiko, das Experten bei der Einführung von SDN sehen, ist das Problem des *Single Point of Failure*. Bisher waren Aufgaben (wie etwa das Routing) über viele Netzwerkknoten hinweg verteilt. Redundanz und Vielseitigkeit machen Netzwerke wie das Internet zwar schwerer Kontrol-

lierbar aber auch robuster gegenüber Angriffen. Schließlich hat es noch kein Angreifer seit bestehen des Internets geschafft, dieses auch nur ansatzweise außer Kraft zu setzen. SDN-Controller auf der anderen Seite steuern Netzwerke von einem einzigen Punkt aus. Dies wäre ein ideales Angriffsziel um großen Schaden anzurichten. Denkbar wären etwa DDoS-Angriffe, welche die Handlungsunfähigkeit des Controllers bewirken. Da SDNs bisher noch nicht in der Öffentlichkeit eingesetzt werden, konnte die konkrete Verwundbarkeit des Controlers nicht festgestellt werden. Die Gefahr eines SSingle Point of Failureßollte bei der Planung und Entwicklung von SDNs stets berücksichtigt werden. Die Installation von Redundanz durch Controler-Replikate könnte beispielsweise die Widerstandsfähigkeit erhöhen.

8. FAZIT

In dieser Arbeit wurden SDN-basierte Verteidigungsansätze und konventionelle Lösungen untersucht und miteinander verglichen. Dabei zeigen sich die Stärken und Schwächen beider Verteidigungswelten. Sowohl auf der Opferseite, als auch im Quellnetzwerk bietet das SDN-Paradigma keine maßgeblichen Vorteile gegenüber der konventionellen Verteidigungsvariante. Anders verhält es sich im Kernnetzwerk. Dort besticht das SDN-Paradigma durch seine Flexibilität. Die heutige, starre Struktur des Internets lässt Erweiterungen kaum zu. Natürlich können auch SDNs nicht von dem einen Tag auf den anderen installiert werden. Der flächendeckende Einsatz dieses neuen Netzwerkparadigmas stellt vor allem für ISPs eine gewaltige Aufgabe dar. Dennoch zeigen aktuelle Beispiele ([51, 52]) die Umsetzbarkeit und deren Vorteile. Der ISP AT&T plant bis zum Jahr 2020 für 75% seiner Netzwerke das SDN-Paradigma einzusetzen [53]. Es bleibt abzuwarten wie sich SDN-basierte DDoS-Verteidigung in einer realen Umgebung schlägt. Erste Experimente zeigen bereits vielversprechende Resultate.

9. LITERATUR

[1] *PlayStation Network, Xbox Live hit by DDOS attacks*, Abgerufen: 21.05.2015, `http://www.gamesindustry.biz/articles/2014-12-29-playstation-network-xbox-live-hit-by-ddos-attacks`,2015

[2] *Hackerattacke auf Merkel und Regierung*, Abgerufen: 21.05.2015, `https://www.tagesschau.de/inland/kanzlerin-bundestag-101.html`,2015

[3] *Large DDoS Botnet Powered by Routers Infected With "Spike" Malware*, Abgerufen: 21.05.2015, `http://www.securityweek.com/large-ddos-botnet-powered-routers-infected-%E2%80%9Cspike%E2%80%9D-malware`,2015

[4] *Arbor Networks Detects Largest Ever DDoS Attack in Q1 2015 DDoS Report*, Abgerufen: 21.05.2015, `http://www.arbornetworks.com/news-and-events/press-releases/recent-press-releases/5405-arbor-networks-records-largest-ever-ddos-attack-in-q1-2015-ddos-report`,2015

[5] Akamai: *Q1 2015 State of the Internet — Security Report*, 2015

[6] Akamai: *Q4 2014 State of the Internet — Security Report*, 2014

[7] *DDoS-Malware auf Linux-Servern entdeckt*, Abgerufen: 26.05.2015, `http://www.golem.de/news/botnetze-ddos-malware-auf-linux-servern-entdeckt-1409-109028.html`,2014

[8] David Dittrich: *The "stacheldraht"distributed denial of service attack tool*,1999

[9] Saman Taghavi Zargar et al.: *A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks*, Communications Surveys & Tutorials, IEEE 15.4 (2013): 2046-2069.

[10] Jelena Mirkovic, Peter Reiher: *A Taxonomy of DDoS Attack and DDoS Defense Mechanisms*, ACM SIGCOMM Computer Communication Review 34.2 (2004): 39-53.

[11] Dr. Abbass Asosheh, Naghmeh Ramezani: *A Comprehensive Taxonomy of DDoS Attacks and Defense Mechanism Applying in a Smart Classification*, WSEAS Transactions on Computers 7.7 (2008): 281-290.

[12] Stephen M. Specht, Ruby B. Lee: *Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures*, ISCA PDCS. 2004.

[13] Christos Douligeris, Aikaterini Mitrokotsa : *DDoS Attacks and Defense Mechanisms: A Classification*, 3rd IEEE International Symposium on Signal Processing & Information Technology,2003

[14] David Dittrich, Sven Dietrech : *command and control structures in malware*, Usenix magazine 32.6,2007

[15] Christian Rossow : *Amplification Hell: Revisiting Network Protocols for DDoS Abuse*, Symposium on Network and Distributed System Security (NDSS),2014

[16] Nirwan Ansari, Amey Shevtekar : *On the new Breed of Denial of Service (DOS) Attacks in the Internet* ,2004

[17] Susan Branowski: *How Secure are the Root DNS Servers?*, Version 1,2003

[18] Cisco Systems, Inc.: *Cisco IOS Scripting with TCL Configuration Guide*,2014

[19] Nick McKeown et al. : *OpenFlow: Enabling Innovation in Campus Networks*, ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[20] Bruno Nunes et al.: *A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks*, Communications Surveys & Tutorials, IEEE 16.3 (2014): 1617-1634.

[21] Natasha Gude et al.: *NOX: Towards an Operating System for Networks*, ACM SIGCOMM Computer Communication Review 38.3 (2008): 105-110.

[22] Rodrigo Braga et al.: *Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow*, Local Computer Networks (LCN), 2010 IEEE 35th Conference on. IEEE, 2010.

[23] T. Kohonen: *The self-organizing map*, Proceedings of the IEEE 78.9 (1990): 1464-1480.

[24] Syed Akbar Mehdi et al.: *SDN Architecture Impact on Network Security*,2011

[25] Ramki Krishnan, Muhammad Durrani: *Real Time SDN and NFV Analytics for DDoS Mitigation*,2014

[26] Ramki Krishnan: *Real Time SDN and NFV Analytics for DDoS Mitigation*, Abgerufen: 11.06.2015, `https://youtu.be/8JBU88dsyks`, 2014

[27] sflow.org : *sflow*, Abgerufen: 11.06.2015, http://www.sflow.org,2015

[28] InMon Corp. sFlow: *sFlow-RT*, Abgerufen: 11.06.2015, http://www.inmon.com/products/sFlow-RT.php,2015

[29] Rishikesh Sahay et al.: *Towards Autonomic DDoS Mitigation using Software Defined Networking*,2015

[30] K. Giotis et al.: *Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments*, Computer Networks 62 (2014): 122-136.

[31] Ryan Naraine: *Massive DDoS Attack Hit DNS Root Servers*, Abgerufen: 11.06.2015, http://www.internetnews.com/dev-news/article.php/1486981/, 2002

[32] Jose Nazario: *Church of Scientology DDoS Statistics*, Abgerufen: 11.06.2015, https://asert.arbornetworks.com/church-of-scientology-ddos-statistics/, 2008

[33] Quinn Norton: *Church of Scientology DDoS Statistics*, Abgerufen: 11.06.2015, http://www.wired.com/2011/12/anonymous-101-part-deux/3/, 2011

[34] PC World: *Hackers Hit Scientology With Online Attack*, Abgerufen: 11.06.2015, http://www.washingtonpost.com/wp-dyn/content/article/2008/01/25/AR2008012503399.html, 2008

[35] Ryan Barnett: *LOIC DDoS Analysis and Detection*, Abgerufen: 11.06.2015, https://www.trustwave.com/Resources/SpiderLabs-Blog/LOIC-DDoS-Analysis-and-Detection/,2011

[36] Matthew Prince: *The DDoS That Knocked Spamhaus Offline (And How We Mitigated It)*, Abgerufen: 11.06.2015, https://blog.cloudflare.com/the-ddos-that-knocked-spamhaus-offline-and-ho/,2013

[37] NSFOCUS: *Analysis of DDoS Attacks on Spamhaus and recommended solution*, 2013

[38] Matthew Prince: *Technical Details Behind a 400Gbps NTP Amplification DDoS Attack*, Abgerufen: 11.06.2015, https://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack/,2014

[39] Bill Brenner: *TCP Flag DDoS Attack by Lizard Squad Indicates DDoS Tool Development*, Abgerufen: 11.06.2015, https://blogs.akamai.com/2015/01/tcp-flag-ddos-attack-by-lizard-squad-indicates-ddos-tool-development.html,2015

[40] Brian Krebs: *Cowards Attack Sony PlayStation, Microsoft xBox Networks*, Abgerufen: 11.06.2015, http://krebsonsecurity.com/2014/12/cowards-attack-sony-playstation-microsoft-xbox-networks/comment-page-1/,2014

[41] Brian Krebs: *Lizard Stresser Runs on Hacked Home Routers*, Abgerufen: 11.06.2015, http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/,2015

[42] Erik Hjelmvik: *China's Man-on-the-Side Attack on GitHub*, Abgerufen: 11.06.2015, http://www.netresec.com/?page=Blog&month=2015-03&post=China%27s-Man-on-the-Side-Attack-on-GitHub,2015

[43] Haining Wang et al.: *Defense Against Spoofed IP Traffic Using Hop-Count Filtering*, IEEE/ACM Transactions on Networking (ToN) 15.1 (2007): 40-53.

[44] W. Eddy: *TCP SYN Flooding Attacks and Common Mitigations*, RFC 4987,2007

[45] D. J. Bernstein: *SYN cookies*, Abgerufen 11.06.2015, http://cr.yp.to/syncookies.html,1997

[46] Information Sciences Institute: *TRANSMISSION CONTROL PROTOCOL*, RFC 793,1981

[47] George Oikonomou et al.: *A Framework for A Collaborative DDoS Defense*, Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual. IEEE,2006.

[48] P. Ferguson: *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*, RFC 2827,2000

[49] J. Mirkovic, P. Reiher: *D-WARD: a source-end defense against flooding denial-of-service attacks*, Dependable and Secure Computing, IEEE Transactions on 2.3 (2005): 216-232.

[50] Roland van Rijswijk-Deij et al.: *DNSSEC and Its Potential for DDoS Attacks*, Proceedings of the 2014 Conference on Internet Measurement Conference. ACM,2014

[51] Marcia Savage: *Google's Infrastructure Chief Talks SDN*, Abgerufen: 08.07.2015, http://www.networkcomputing.com/data-centers/googles-infrastructure-chief-talks-sdn/d/d-id/1320352,2015

[52] Jonathan Vanian: *Facebook shows the promise of SDN with new networking tech*, Abgerufen: 08.07.2015, https://gigaom.com/2014/11/14/facebook-shows-the-promise-of-sdn-with-new-networking-tech/,2014

[53] J Dan Meyer: *Software, NFV, SDN focus bolsters workforce opportunities*, Abgerufen: 08.07.2015,http://www.rcrwireless.com/20141216/telecom-software/att-targets-75-virtualization-software-control-of-network-by-2020-tag2,2014