Verbesserung der Netzsicherheit in virtualisierten Umgebungen mit Hilfe von OpenFlow

Andreas Brinner
genua mbH
andreas_brinner@genua.de

Rene Rietz
BTU Cottbus–Senftenberg
rrietz@informatik.tu-cottbus.de

Abstract: Viele der klassischen Techniken, mit denen die Netzsicherheit in physikalischen Systemen erhöht werden, lassen sich nicht oder nur mit großem Aufwand in virtualisierten Umgebungen einsetzen. So ist es prinzipbedingt nicht möglich, virtuelle Systeme auf einem Hostsystem physikalisch voneinander zu trennen, um deren Kommunikation durch eine Firewall filtern zu können. Auch Angriffe auf den Layern 2 und 3, wie ARP-Spoofing und Rogue-DHCP-Server, sind in physikalischen Netzen durch entsprechende Switches gut beherrschbar. In virtualisierten Umgebungen sind diese allerdings nicht einsetzbar.

In diesem Beitrag stellen wir einen Ansatz vor, mit Hilfe von OpenFlow und eines speziellen OpenFlow-Controllers die Netzsicherheit in virtuellen Systemen zu erhöhen. Ohne Veränderungen an den Gastsystemen lassen sich ARP- und DHCP-Attacken effektiv verhindern. Zum Schutz von Systemdiensten können die Datenverbindungen für die beteiligten Systeme transparent durch Firewalls, Application-Level-Gateways oder Intrusion-Detection-Systeme geroutet werden. Mit Hilfe einer Client-Authentifizierung lassen sich die definierten Sicherheitsregeln auch nach der Migration von virtuellen Instanzen weiter einhalten.

1 Einleitung

Virtualisierte Systeme sind in ihrer Standardkonfiguration oft besonders anfällig für Angriffe auf den Layern 2 oder 3. Die Linux-Bridge bietet zum Beispiel keinen Schutz gegen ARP-Attacken oder Rogue-DHCP-Server. Des Weiteren lassen sich virtuelle Gäste auf einem Host-System nicht ohne weiteres durch Firewalls voneinander trennen. Falls dies doch geschieht, dann werden die Firewalls meist ebenfalls auf dem virtuellen System ausgeführt. Dies kann aber dazu führen, dass eine Kompromittierung der Firewall den gesamten Host mitsamt aller Gastsysteme gefährdet [WDWY10].

Um die Sicherheit von virtuellen Systemen zu erhöhen, haben wir ein OpenFlow-basiertes Verfahren entwickelt, das es ermöglicht, physikalische Firewalls in die Kommunikation zwischen virtuellen Instanzen einzubinden und grundlegende Layer 2 und 3 Attacken zu unterbinden. OpenFlow [MAB+08, Ope11, Ope13b] wurde an der Stanford University entwickelt mit dem Ziel, die Möglichkeit zu erhalten, Control- und Data-Plane innerhalb

eines Switches zu trennen. Dadurch lässt sich die Switch-Logik in einen separaten Controller auslagern. Entscheidungen, beispielsweise zum Paket-Forwarding, werden nicht mehr eigenständig vom Switch, sondern vom zuständigen Controller getroffen, der dadurch die vollständige Kontrolle über das Netz und das Datenrouting erhält. Für das hier vorgestellte Verfahren wurde ein OpenFlow-Controller entwickelt, der ARP- und DHCP-Pakete an den OpenFlow-Switches nicht weiterleitet, sondern selber beantwortet. Außerdem können Regeln definiert werden, nach denen Datenverbindungen für die Clients transparent durch Firewalls umgeleitet werden können.

Unsere Lösung lässt sich zentral verwalten. Sie bedarf keiner Änderungen und keines Eingriffs auf den zu schützenden Gastsystemen. Einzig die verwendeten Switche müssen OpenFlow-fähig sein. In Linux-Hostsystemen kann dazu der Open vSwitch verwendet werden, das die Linux-Bridge ersetzt. Im Gegensatz zu einer vollständigen (virtuellen) Firewall stellt dies eine wesentlich kleinere Trusting-Computing-Base da, was Angriffe auf diese zentrale Komponente erheblich erschwert.

2 Problemstellung

Abbildung 1 zeigt ein einfaches Mustersystem, bestehend aus einem physikalischen Host und zwei virtuellen Gästen. Diese sind über ein virtuelles Netz untereinander und mit dem externen Netz verbunden. Ein Angreifer kann sich entweder im externen Netz (A) oder auf einem der Gastsysteme (B) befinden. Von extern ist ein Angriff auf das interne Netz, den Host oder eines der Gastsysteme möglich. Von einem der Gastsysteme aus, kann zusätzlich noch das externe Netz als Angriffsziel dienen. Angreifer (B) kann dabei legal Zugriff auf das Gastsystem 2 erhalten haben, zum Beispiel durch Anmieten einer virtuellen Instanz bei einem Hosting-Provider. Die Stadttore symbolisieren diejenigen Stellen, an denen idealerweise Firewalls eingesetzt werden müssten, um alle Systeme voreinander zu schützen.

Schon an diesem minimalen Mustersystem wird ersichtlich, dass für einen idealen Schutz eine große Anzahl an Firewalls benötigt wird. Folgende Fragen stellen sich und sollen als erstes geklärt werden. (1) Welchen Gefahren sind die Gastsysteme überhaupt ausgesetzt? (2) Was sind die für unsere Untersuchung relevanten Bedrohungen? Im Rahmen dieser Untersuchungen sind nur solche Bedrohungen relevant, die sich zumindest theoretisch mit Hilfe einer Firewall abwenden lassen. Dazu gehören zuallererst die klassischen Netzbedrohungen, denen auch ein physikalisches, nicht virtualisiertes System ausgesetzt ist. Danach ist (3) die Frage zu klären, ob sich für die virtuellen Umgebungen neue, systematische Angriffsmöglichkeiten ergeben, die sich mit Hilfe einer Firewall absichern lassen.

2.1 Klassische Netzbedrohungen

Im Folgenden sind typische Schwachstellen Ethernet-basierter Netze aufgeführt. Diese Schwachstellen ermöglichen es einem Angreifer vor allem, den Datenverkehr anderer Sys-

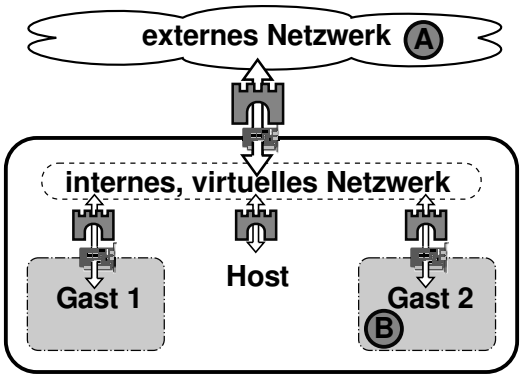

Abbildung 1: Hostsystem mit zwei virtuellen Gästen und den Positionen, an denen Firewalls zum optimalen Schutz eingesetzt werden müssten.

teme umzuleiten und mitzulesen. Sie greifen auf den Layern 2 und 3 des OSI-Modells an.

- *MAC-Spoofing.* Ein Angreifer täuscht eine falsche MAC-Adresse vor, um damit zum Beispiel per DHCP eine fremde IP-Adresse zugeteilt zu bekommen.
- *ARP-Spoofing.* Ein Angreifer versucht durch gefälschte ARP-Pakete die Kommunikation der Opfer zu unterbrechen oder derart umzuleiten, dass er sie mitlesen kann.
- *ARP-Flooding.* Durch gefälschte ARP-Pakete soll der ARP-Cache einfacher Switches geflutet werden, sodass diese in den Broadcast-Modus umschalten und alle Daten an allen Ports ausgeben. Auch hier ist wiederum das Ziel, die Kommunikation der anderen Systeme mitlesen zu können.
- *Rogue-DHCP-Server.* Der Angreifer richtet einen eigenen DHCP-Server im Netz ein und versucht auf DHCP-Requests schneller zu antworten, als der eigentliche DHCP-Server. Gelingt dies, können dem Opfer gefälschte DHCP-Pakete untergeschoben und somit dessen Konfiguration geändert werden. Durch Angabe eines gefälschten Standardgateways kann das Opfer dazu gebracht werden, den Angreifer als Standardgateway anzusehen.

Diese Schwachstellen existieren sowohl für physikalische als auch für virtualisierte Netze. Jedoch sind physikalische Systeme schon wesentlich weiter entwickelt und es existieren Lösungen, die sie besser dagegen schützen. Eigene Versuche an einem Debian-System mit KVM-Virtualisierung haben gezeigt, dass ARP-Spoofing zwischen virtuellen Gastsystemen ohne großen Aufwand möglich ist. Die standardmäßig verwendete Linux-Bridge bietet keinen Schutz gegen diese Angriffe. Ein erfolgreicher Angriff eines Rogue-DHCP-Servers basiert darauf, dass er schneller eine Antwort liefern kann, als der eigentliche DHCP-Server. Aufgrund der kurzen Wege zwischen den Gästen eines Hostsystems, ist das Einrichten eines Rogue-DHCP-Servers gerade dort sehr erfolgsversprechend.

2.2 Besondere Gefahren in virtuellen Systemen

Um herauszufinden, ob in virtualisierten Systemen neue, systematische Schwachstellen existieren, die sich durch eine Firewall absichern ließen, wurden die CVEs (Common Vulnerabilities and Exposures [MIT13]) verschiedener Virtualisierungslösungen untersucht. Dazu wurde jede Schwachstelle einem Angriffsvektor und einer Auswirkung zugeordnet. Der Angriffsvektor sagt aus, ob sich diese Schwachstelle über das Netz (”Netz”) ausnutzen lässt oder nur auf direktem Weg (”Direkt”), nachdem schon ein Zugriff auf das System besteht. Für den direkten Zugriff wird also ein Login auf dem System benötigt. Die Auswirkung unterscheidet darüber, ob ein erfolgreicher Angriff einen Zugriff auf das System ermöglicht (”Zugriff”) oder nur die Verfügbarkeit einschränkt (”Verfügbarkeit”). Je nach Anwendungsfall sind diese Auswirkungen unterschiedlich stark zu gewichten. In unserem Fall sind aber beide Auswirkungen unerwünscht und sollen verhindert werden.

	Angriffsvektoren	
Auswirkung	Netz	Direkt
Zugriff	10	41
Verfügbarkeit	8	30

Tabelle 1: Ergebnisse der CVE-Untersuchung

Tabelle 1 zeigt die Ergebnisse dieser Auswertung. Von den untersuchten 177 CVEs konnten die meisten aufgrund fehlender Informationen nicht bewertet werden. Die restlichen Schwachstellen fallen überwiegend in die Gruppe ”Direkt”, die sich nicht durch eine Firewall blocken lassen und somit für unsere Untersuchung uninteressant sind. Gerade einmal 18 Schwachstellen können über das Netz ausgenutzt werden. Dies sind unter anderem Schwachstellen im virtuellen Netztreiber, dem VNC-Zugriff auf die Gastsysteme und im DHCP-Code, die bei VMware aufgetreten sind. Die VNC- und DHCP-Schwachstellen hätten sich dabei durch eine restriktiv eingestellte Firewall schützen lassen. Die restlichen acht Schwachstellen können durch spezielle Datenpakete ausgenutzt werden und führen in sieben Fällen zu einer Denial-of-Service Attacke. Da diese Schwachstellen allerdings sehr speziell sind und sich keine Systematik dahinter erkennen lässt, dürfte es schwer bis unmöglich sein, generische Firewallregelen zu finden, die ein System proaktiv dagegen schützt.

Es bleibt also festzuhalten, dass auch virtuelle Systeme vor allem vor den klassischen Netzschwachstellen geschützt werden müssen, denen auch physikalische Systeme ausgesetzt sind. Virtuelle Systeme sind zwar vielen, neuen Angriffsmöglichkeiten ausgesetzt, es konnten allerdings keine neuen, systematischen Schwachstellen gefunden werden, die sich durch den Einsatz von Firewalls abblocken lassen. Einzig der Schutz der VNC-Konsole erscheint als sinnvoll und praktikabel.

3 Existierende Lösungsansätze

Sowohl für physikalische, als auch für virtuelle Netze existieren zumindest Teillösungen, um diese vor Angriffen zu schützen. Im Folgenden werden wir einige dieser Techniken für physikalische Netze vorstellen und auch einen Blick auf Lösungen für virtuelle Systeme werfen.

3.1 Lösungen für physikalische Netze

Um ein klassisches, physikalisches System abzusichern, werden die Netze oft in kleinere Segmente unterteilt. Dies kann zum einen auf einer logischen Ebene durch Konfiguration von IP-Subnetzen geschehen, was allerdings nur eine sehr schwache Aufteilung ist, oder zum anderen physikalisch durch entsprechende Verkabelung der Systeme. An den Übergangspunkten können dann die Daten gefiltert und untersucht werden. Dazu werden entsprechend den Sicherheitsanforderungen unterschiedliche Systeme verwendet, die allerdings auch unterschiedliche Performanceanforderungen stellen. Angefangen beim Einsatz von Routern, über einfache Paketfilter und Firewalls bis hin zu Application-Level-Gateways (ALGs) und Intrusion Detection Systemen (IDS) bieten sie einen immer besseren Schutz, indem sie immer tiefer in die zu untersuchenden Datenströme hineinschauen. Je genauer die Daten untersucht werden, desto größer wird allerdings auch der Rechenaufwand. Außerdem können alle Systeme nur solche Daten untersuchen, die durch sie hindurch gehen. Daten, die im selben Subnetz bleiben, können nicht untersucht werden.

Eine andere Möglichkeit ist der Einsatz von intelligenten Switches, die einen gewissen Schutz gegen Rogue-DHCP-Server und ARP-Attacken bieten [CIS13]. Dies wird erreicht, indem diese Switches bestimmte Datenpakete an ihren Ports verbieten. Da normalerweise jeder Netzteilnehmer an einen Switch angeschlossen ist, bietet diese Lösung eine ziemlich flächendeckende Sicherheit. Auch virtuelle LANs (VLANs) können zum Separieren von Clients eingesetzt werden, sodass nur die Systeme, die demselben VLAN zugeordnet sind, miteinander kommunizieren können. Allerdings gibt es auch für VLANs schon Angriffsmethoden, über die ein sogenanntes VLAN-Hopping erreicht werden kann [CIS13]. Außerdem ist der Einsatz von VLANs recht unflexibel, da ein Client entweder einem VLAN zugeordent ist oder nicht. Sollen mit Hilfe von VLANs alle virtuellen Gäste voneinander separiert und die gegenseitige Kommunikation gefiltert werden, so entsteht letztendlich ein sternförmiges System mit der entsprechenden Firewall im Mittelpunkt.

Soll auf den Einsatz spezieller Hardware verzichtet werden, können gewisse Konfigurationen auch statisch hinterlegt werden. Statische IP-Adressen und ARP-Tabellen können gegen die oben genannten Attacken schützen. Allerdings geht dabei ein großes Stück an Flexibilität verloren und der Administrationsaufwand steigt enorm an, da jede Änderung an allen Systemen manuell nachvollzogen werden muss.

3.2 Lösungen für virtuelle Systeme

Viele der oben genannten Lösungen lassen sich nicht ohne weiteres auf virtuelle Systeme übertragen, da sich Gäste auf einem Hostsystem nicht physikalisch voneinander trennen lassen. Auch der Einsatz von intelligenten Hardware-Switches ist dabei nicht möglich. Daher werden von den Herstellern der Virtualisierungslösungen spezielle, meist proprietäre Softwarelösungen angeboten. In [Bel99, IKBS00] wurde das Konzept der Distributed-Firewall vorgestellt. Diese besteht aus virtuellen Firewallmodulen auf den zu schützenden Systemen, die zentral administriert werden können. Die einfachere Administration wird dabei allerdings mit einem erhöhten Einrichtungsaufwand erkauft, da auf jedem zu schützenden System diese Firewallmodule installiert werden müssen. Außerdem können nur solche Systeme geschützt werden, für die ein passendes Firewallmodul existiert.

4 Lösung mit Hilfe von OpenFlow

Mit Hilfe von OpenFlow wurde eine Lösung für virtuelle Systeme entwickelt, die diese gegen die zuvor genannten Angriffsformen absichert. Sie erfordert keine Änderungen an den Gastsystemen, ist flexibel, zentral administrierbar, verhindert effektiv Angriffe auf den Layern 2 und 3 und ermöglicht es, externe physikalische Firewalls in die virtuellen Systeme einzubinden. Dazu wird in einem ersten Schritt die Linux-Bridge durch den Open vSwitch [OPE13a] ersetzt (siehe Abbildung 2). Dies ist ein OpenFlow-fähiger virtueller Switch, der Switchingfunktionalität in virtuellen Systemen zur Verfügung stellt. Werden auch die physikalischen Switches durch OpenFlow-fähige Modelle ersetzt, entsteht ein wesentlich homogeneres Netz, in dem die Grenzen zwischen physikalischer und virtueller Umgebung verschwimmen. Abhängig von definierbaren Sicherheitsregeln können einzelne oder alle Datenverbindungen eines Systems für dieses transparent durch eine Firewall geroutet werden. Dafür kann sowohl eine virtuelle als auch eine physikalische Firewall eingesetzt werden.

Die Switches haben vollständige Kontrolle über alle Datenströme im Netz und werden ihrerseits vom zentralen OpenFlow-Controller gesteuert. Dieser ist somit die zentrale und kontrollierende Instanz, die außerdem eine globale Sicht auf das Netz hat. Einige Sicherheitsfunktionen können auch direkt von den Switchen bzw. dem Controller übernommen werden. Im Folgenden werden die weiteren Ideen und Maßnahmen vorgestellt, auf denen das Prinzip des IP-Switches beruht.

4.1 ARP- und DHCP-Pakete nicht mehr broadcasten

Eines der großen Probleme in virtuellen Netzen besteht darin, dass ARP- und DHCP-Pakete im Netz gebroadcastet werden und somit jeder Netzteilnehmer mithören und auch antworten kann. Es ist also naheliegend, in den OpenFlow-Switches entsprechende Regeln zu hinterlegen, die dafür sorgen, dass alle ARP- und DHCP-Pakete an den Control-

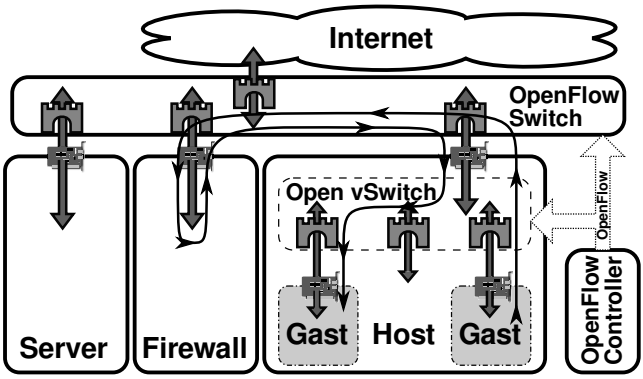

Abbildung 2: Virtueller und physikalischer Switch mit OpenFlow Controller. Die Kommunikation zwischen den beiden virtuellen Instanzen wird über die physikalische Firewall umgeleitet.

ler umgeleitet und nicht mehr weiter verteilt werden. Um berechtigte Anfragen dennoch beantworten zu können, wird im Controller ein entsprechender ARP- und DHCP-Server integriert. Ein Großteil der Schwachstellen lässt sich alleine mit dieser einfachen Maßnahmen beheben. ARP-Spoofing wird unterbunden, indem (bösartige) APR-Pakete nicht länger im Netz gebroadcastet, sondern vom Controller selbst beantwortet oder verworfen werden. Das gleiche gilt für DHCP-Pakete, was Rogue-DHCP-Server verhindert. Ein Angreifer sieht nicht einmal mehr die DHCP-Requests der anderen Systeme, was notwendig wäre, um im richtigen Augenblick eine bösartige Antwort senden zu können.

4.2 Verwenden von IP- anstelle von MAC-Adressen

Eine weitere Idee ist, zum Ansprechen der Zielsysteme anstelle von MAC-Adressen IP-Addressen zu verwenden. Viele der genannten Schwachstellen beruhen darauf, dass zur Adressierung eines Systems zwei unterschiedliche Ebenen zuständig sind (Layer 2 und 3), die miteinander in Einklang gebracht werden müssen. Um zum Beispiel die zu einer IP-Adresse zugehörige MAC-Adresse zu erhalten, wird ARP verwendet. Genau an diesem Punkt kann ein Angreifer ansetzen, um das System in einen nicht konsistenten Zustand zu bringen und Verbindungen mitzulesen oder zu unterbinden.

Dazu wird der ARP-Server im Controller so implementiert, dass er alle ARP-Anfragen der Clients mit einer virtuellen, nicht existierenden MAC-Adresse (im Beispiel FF:...) beantwortet. Beim Versand von Ethernet-Frames verwenden die Clients diese MAC-Adressen zusammen mit den IP-Adressen der Zielrechner. Die OpenFlow-Switches leiten diese Informationen an den Controller weiter, der anhand der IP-Adresse eine passende Route bestimmt und entsprechende Match-Einträge in den Switches erstellt. Der letzte Switch auf dieser Route erhält die Aufgabe, die MAC-Adressen so umzuschreiben, dass sie für das

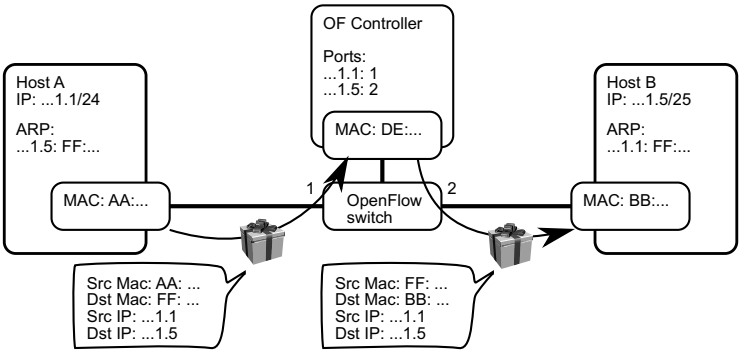

Abbildung 3: Kommunikation mit virtuellen MAC-Adressen und OF-Switch.

Zielsystem gültig sind. Dieser Vorgang ist beispielhaft in Abbildung 3 dargestellt. Aufgrund der virtuellen MAC-Adressen kennt jeder Client nur seine eigene, reale Adresse. Von den anderen Systemen ist ihnen nur deren IP-Adresse bekannt und auch die eigentliche Netztopologie bleibt ihnen verborgen. Jedes System kennt somit nur all jene Informationen, die für eine erfolgreiche Kommunikation zwingend erforderlich sind.

4.3 Einbinden von Firewalls

Einfache Paketfilterregeln können direkt durch entsprechende Regeln auf den OpenFlow-Switches implementiert werden. Auch Firewalls mit zustandsbehafteter Verbindungsverwaltung lassen sich mit Hilfe des Controllers und der Regeln auf den Switches realisieren. Im Prinzip entspricht ein Flow-basiertes Switching einer solchen Firewall. Für jede neue Datenverbindung wird eine Anfrage an den Controller gesendet, der dann im Einzelfall entscheiden kann, ob diese Verbindung zugelassen werden soll oder nicht. Danach wird eine spezifische Regel auf den Switches hinterlegt. Sollen Application-Level-Gateways oder Intrusion Detection Systeme eingesetzt werden, so ermöglicht es dieselbe Technik, die zu filternden Verbindungen dynamisch und transparent zu einem entsprechenden System zu routen. Erst nach erfolgreicher Filterung werden die Daten dann dem Zielsystem zugestellt. All diese Sicherheitsregeln lassen sich im Controller für jeden Client individuell hinterlegen und werden unabhängig davon, an welchem Port sich das System mit dem Netz verbindet, durchgesetzt. Jeder Port an jedem OpenFlow Switch wird somit zur Firewall (siehe auch Abbildung 2).

4.4 MAC-Adressen der Clients authentifizieren

Um die Clients eindeutig identifizieren zu können, kann der Controller diese mit Hilfe von 802.1x oder durch Überprüfung des SSH-Hostkeys authentifizieren. Dies geschieht nachdem sich ein Client am Netz angemeldet hat. Mit Hilfe der Client-Authentifizierung kann MAC-Spoofing verhindert werden, was wiederum notwendig ist, wenn im Controller für jeden Client individuelle Sicherheitsregeln hinterlegt werden sollen. Dadurch wird ein freies Roaming oder Migrieren von Client-Systemen ermöglicht, wobei alle Sicherheitsregeln mitwandern und beibehalten werden.

5 Umsetzung und Evaluation

Im Rahmen des Projekts ist mit Hilfe des Ryu-Frameworks [RYU13] ein OpenFlow-Controller entstanden, der diese Funktionalität umsetzt. Während der Entwicklungsphase wurde der Controller intensiv mit Mininet [LHM10] getestet. Später kam ein eigenständiges Testnetz mit Debian/KVM-Server, mehreren virtuellen Instanzen und einer Anbindung an zwei OpenWRT Linksys Router hinzu. Dieses System ist in Abbildung 4 schematisch dargestellt. Auf dem KVM-Server sind drei virtuelle Gastsysteme mit unterschiedlichen Aufgaben eingerichtet. Das NAT-Gateway ist über das Interface eth1 mit dem externen Netz verbunden und verbindet die Teilnehmer im OpenFlow-Netz mit diesem. In der Firewall-Instanz ist eine genuscreen-Firewall installiert, die für die Clients transparent den ein- und ausgehenden Datenverkehr filtert. Der virtuelle Client dient als Testinstanz eines normalen Netzteilnehmers. Eine zweite genuscreen-Firewall wurde als externes, physikalisches System angeschlossen.

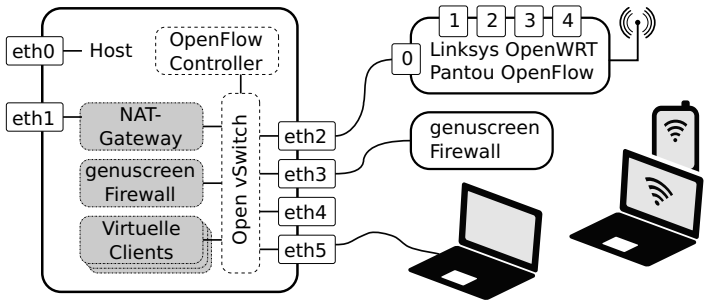

Abbildung 4: OpenFlow Testsystem bestehend aus Debian/KVM Host mit virtuellen Instanzen, Linksys Router, physikalischer Firewall und Endgeräten.

Über den Open vSwitch sind neben diesen Instanzen noch vier weitere, physikalische Netzschnittstellen angebunden, die somit als OpenFlow-Switch arbeiten. An diese Ports können direkt weitere Endgeräte angeschlossen werden, die sich dann wie normale Teilnehmer des OpenFlow-Testnetzes verhalten. Außerdem können weitere OpenFlow-Swit-

ches kaskadiert werden. In Abbildung 4 ist ein Linksys-Router mit einer Pantou OpenWRT Firmware eingezeichnet, der sich wiederum als OpenFlow-Switch verhält und das Testnetz auch für WiFi-Teilnehmer zur Verfügung stellt. Darüber konnten weitere Clients, wie Laptops oder Smartphones, erfolgreich ins OpenFlow-Testnetz eingebunden und vom OpenFlow-Controller gesteuert werden. Das Routing der Datenverbindungen durch die Firewall und/oder das NAT-Gateway wird durch den Controller gesteuert und geschieht für die Clients völlig transparent.

Durch die in 4.1 beschriebenen Maßnahmen konnten Angriffe wie ARP-Spoofing, ARP-Flooding und Rogue-DHCP-Server effektiv verhindert werden. Die im OpenFlow-Controller implementierten ARP- und DHCP-Server verwerfen ungültige APR- und DHCP-Pakete, sodass andere Netzteilnehmer diese nicht mehr erhalten. Angriffe auf Systemdienste konnten durch das Einbinden einer Firewall in die Kommunikationsverbindung verhindert werden. Die Daten wurden dabei für die Netzteilnehmer transparent durch eine der Firewalls geroutet und gefiltert. MAC-Spoofing konnte durch Authentifizierung der Netzteilnehmer mit 802.1X verhindert werden. Dazu wurde der OpenFlow-Controller entsprechend erweitert, sodass er Systeme, die sich neu am Netz anmelden, authentifizieren konnte.

6 Zusammenfassung und Ausblick

Es konnte ein OpenFlow-Controller entwickelt werden, der erfolgreich die aufgezeigten Fähigkeiten umsetzt. Die Funktionalität wurde in unterschiedlichen Test-Setups, unter anderem im Mischbetrieb mit virtuellen und physikalischen Systemen, getestet. Da für den Testbetrieb keine dedizierte OpenFlow-Hardware, sondern Softwareswitches verwendet wurden, kann keine realistische Performanceabschätzung für den Mischbetrieb gegeben werden. Für den reinen virtuellen Betrieb ergab der Einsatz des Open vSwitches nur geringe Performanceeinbußen. Der Einfluss des OpenFlow Controllers auf die Performance wird erst nach einer Optimierung des Quellcodes realistische Ergebnisse liefern können.

In weiteren Untersuchungen soll der Mischbetrieb mit OpenFlow- und Legacy-Switches untersucht werden. Außerdem soll die weitere Einbindung von Middleboxen, wie der Firewall, weiter untersucht und entsprechende APIs definiert werden. Dadurch könnten unter anderem Hardwareshunts auf den Switches realisiert werden, die bei einer erkannten, gutartigen Verbindung die Performance enorm steigern können.

Literatur

[Bel99] Steven M. Bellovin. Distributed Firewalls. *login*, Seiten 37–39, November 1999.

[CIS13] CISCO. VLAN Security White Paper. `http://www.cisco.com/en/US/products/hw/switches/ps708/products\_white\_%paper09186a008013159f.shtml`, 2013.

[IKBS00] Sotiris Ioannidis, Angelos D. Keromytis, Steve M. Bellovin und Jonathan M. Smith. Implementing a Distributed Firewall. *Proceedings of Computer and Communications Security (CCS)*, Seiten 190–199, November 2000.

[LHM10] Bob Lantz, Brandon Heller und Nick McKeown. A Network in a Laptop: Rapid Prototyping for Software-defined Networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, Seiten 19:1–19:6, New York, NY, USA, 2010. ACM.

[MAB+08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker und Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Marz 2008.

[MIT13] MITRE Corporation. CVE – Common Vulnerabilities and Exposures. `http://cve.mitre.org/`, 2013.

[Ope11] OpenFlow Consortium. OpenFlow – Enabling Innovation in Your Network. `http://archive.openflow.org/`, 2011.

[OPE13a] Open vSwitch – An Open Virtual Switch. `http://openvswitch.org/`, 2013.

[Ope13b] Open Networking Foundation. OpenFlow. `http://www.opennetworking.org/sdn-resources/onf-specifications/openflow%`, 2013.

[RYU13] Ryu SDN Framework. `http://osrg.github.io/ryu/`, 2013.

[WDWY10] Hanqian Wu, Yi Ding, C. Winer und Li Yao. Network security for virtual machine in cloud computing. In *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*, Seiten 18–21, 2010.