Sicherheitsaspekte bei Deployment virtueller Netzwerkinfrastrukturen

Gerhard Gröschl, Miran Mizani
{gerhard.groeschl, miran.mizani}@campus.lmu.de

Seminar: Trends in Mobilen und Verteilten Systemen
Wintersemester 2016/2017

Lehrstuhl für Mobile und Verteilte Systeme
Institut für Informatik
Ludwig-Maximilians-Universität München

Betreuer: Michael T. Beck
eingereicht am 25. Januar 2017

Abstract: Durch die äußerst effiziente Nutzung von Hardware mittels Virtualisierung steigt die Nachfrage nach virtualisierten Infrastrukturen enorm. Serviceprovider müssen die Hardwarebasis für ihre Dienste nicht mehr selbst unterhalten und geben ihre dahingehende Verantwortung an Infrastrukturanbieter weiter. Um die Sicherheit dieser Strukturen nicht zu vernachlässigen, arbeiten viele Forscher in diesem Bereich und versuchen effiziente Algorithmen mit integrierter Beachtung der Sicherheitsaspekte zu finden. Diese Arbeit gibt einen Überblick und eine Klassifizierung der Gefahren, die solche Konstrukte betreffen, sowie eine Analyse zweier unterschiedlicher Ansätze zur Vermeidung möglichst vieler Risiken zum Zeitpunkt der Planung.

Inhaltsverzeichnis

1 Einleitung

Ein Konzept dem Internet Impasse mit flexibler Architektur und Handhabbarkeit zu begegnen, wurde in der Netzwerkvirtualisierung (NV) gefunden. [APST05, BOB+12, FBB+13] Sie basiert auf Knoten- (z.B. Xen) und Linkvirtualisierung und erlaubt so von der tatsächlichen physischen Hardware und Netzinfrastruktur (NI) beinahe unabhängige logische bzw. virtuelle Netzwerke einzurichten, welche nach außen hin den Anschein physischer Netzwerke erwecken. Die Möglichkeit mehrere virtuelle Maschinen (VMs) pro physischem Host und verschiedene heterogene virtuelle Netzwerke (VNs) auf demselben physischen Substratnetz zu betreiben befördert die Flexibilität der Netzwerkstruktur und wirkt einer Beschränkung auf bestehende Architekturen (Internet Ossification Problem) entgegen. [APST05]

Die großen Vorteile der NV liegen in der Abstraktion von der eingesetzten Hardware. Das Erstellen, Verändern, Migrieren, Zurücksetzen und Löschen von Maschinen funktioniert genauso einfach wie der Umgang mit Dateien, was eine dynamische Nutzung des Netzwerkes erlaubt. Virtuelle Maschinen und Netzwerke eigenen sich auch als Testumgebung. Einerseits werden bestehende Systeme im Fehlerfall nicht direkt beeinträchtigt. Andererseits kann neuer Code leicht in verschiedenen Umgebungen (Windows, Linux, verschiedengroßer RAM, mit oder ohne Software-Developer-Kits etc.) ohne zusätzliche Hardware getestet und später einfach ausgerollt werden.

NV eröffnet eine Unterteilung des klassischen Internetserviceproviders (ISP) in Service-Provider (SP) und Infrastructure-Provider (InP). Damit gewonnene Freiheiten durch z.B. jeweils unabhängige Technologieentscheidungen sind besonders für Unternehmen interessant, die die Hardwarebasis ihrer Dienste nicht selbst unterhalten wollen. [WCC16]
Das Anbieten von Software und Hardware als on-demand Ressourcen wird wegen des geringen Wartungsaufwand, verminderter Hardwarekosten durch Koexistenz mehrerer Mieter, aber v.a. wegen Automatisierbarkeit in der Programmierung der Netzwerkumgebung vereinfacht. Dass InPs nicht mehr streng durch Hardware limitiert sind, begünstigt Skalierbarkeit und bspw. lastbedingte Migration von VMs auf andere physische Hosts.
Auch für den Kunden bietet NV Vorteile: Unternehmen bezahlen nur noch für diejenigen Ressourcen, die gerade in Anspruch genommen werden. Hochqualitative Hardware kann so zu einem Bruchteil ihres Preises erworben und ungenutzte Hardware reduziert werden. Durch dynamisches Skalieren (z.B. in Zeiten hoher Last) kann die eigene IT-Landschaft mühelos vergrößert werden.

Die durch NV gewonnene Flexibilität und Kostenreduzierung begünstigen z.B. auch das Outsourcing von Rechenleistung, Speicher, Inhalten und Netzwerk. Dadurch wird Soft- und Hardware einfacher nutzbar gemacht und Geschäftsprozesse befördert. Die damit einhergehende Verantwortungsübertragung erfordert eine Anpassung des Risikomanagements und IT-Sicherheitstechnische Arbeiten zur Erhaltung der klassischen C.I.A.-Aspekte. Wegen der gemeinsam genutzten Hardware kommt aus Sicht des Kunden besonders der Isolation und dem Datenschutz eine wichtige Rolle zu.

Essentielle Komponente der NV ist die Wahl der Zuordnung von virtuellen zu physischen Knoten und Links, das Virtual Network Embedding (VNE), welches auf Basis ver-

schiedener Kriterien geschehen kann. Das theoretische Problem des VNE wurde bislang hauptsächlich unter Performanceaspekten optimiert und Sicherheitsbelange dabei weitgehend außer Acht gelassen.
Bekannte Sicherheitsmechanismen wie Verschlüsselung, Firewalls, Intrusion Detection Systeme etc. können zwar auf den virtuellen Komponenten des Netzwerks implementiert werden. Die Sicherheit von Nutzerdaten lässt sich dadurch aber wegen der heterogenen und stark dynamischen Struktur virtueller Umgebungen jedoch nicht garantieren. Zusätzlich gehen Vorteile der Netzwerkvirtualisierung durch den zusätzlichen Overhead verloren. [GCH+16]
Eine mögliche Lösung hierzu ist das Integrieren von Sicherheitsaspekten bereits in den VNE-Prozess, was eine der größten Herausforderungen in der Netzwerkvirtualisierung darstellt. [FBB+13] Werden virtuelle Netzwerke entsprechend ihrer Sicherheitsanforderungen bereits auf Substratknoten mit hinreichender Schutzfunktion wie beispielsweise Firewall abgebildet, so kann Overhead durch zusätzliche Sicherheitstechnik im laufenden Betrieb des VNs reduziert werden. Nicht allen Problemen bzw. Sicherheitsrisiken lässt sich allerdings auf diese Weise begegnen. [BOB+12, GCH+16, WCC16]

Diese Arbeit klassifiziert Gefahren im Kontext virtualisierter Netzwerke und analysiert zwei unterschiedliche Ansätze zur Schaffung eines möglichst hohen Sicherheitsniveaus bereits zum Zeitpunkt des VNE-Prozesses.
Dazu wird zuerst das VNE-Problem in Kapitel 2 dargestellt. Kapitel 3 untersucht Sicherheitsanforderungen an virtuelle Netzwerkstrukturen und klassifiziert Sicherheitsrisiken, die sich in deren Kontext ergeben. Zwei Möglichkeiten zur Vermeidung von Gefahren, denen bereits im VNE-Prozess begegnet werden kann, werden im Kapitel 4 betrachtet. Nach einer Diskussion in dieser Arbeit offengebliebener Probleme in Kapitel 5 wird mit einem Ausblick abgeschlossen.

2 Das Virtual Network Embedding Problem

Um den Anforderungen der heutigen Gefahren zu genügen, ist es unumgänglich sämtliche Grundsätze der IT-Sicherheit bereits in die Infrastrukturplanung miteinzubeziehen. Nicht nur, weil das nachträgliche Schließen von Sicherheitslücken und Hinzufügen von Sicherheitskomponenten in finanzieller und zeitlicher Hinsicht zehnmal so teuer ist wie die initiale Beachtung dieser Aspekte, sondern weil die vollständige Sicherheit eines nachgerüsteten Systems kaum gewährleistet werden kann [Col11].„Security-by-Design“ ist einer der wichtigsten Begriffe bei der Planung öffentlich zugänglicher Netzwerkstrukturen. Dementsprechend groß ist die Nachfrage nach VNE-Algorithmen, die bereits beim Prozess des Mappings viele Sicherheitsaspekte beachten und abdecken. Die vorausgehende Klassifizierung der bekannten Gefahren in „VNE-relevant“ und „nicht-VNE-relevant“ bildet die Grundlage für unsere weiteren Untersuchungen. Um zuvor noch einen genaueren Einblick in die Problematik von VNE zu gewähren, beginnen wir zunächst mit der Erläuterung des VNE-Problems sowie einem Überblick über die verschiedenen erfolgreichen Strategien bestehender VNE-Algorithmen, welche die Sicherheitsaspekte noch nicht beachten.

Betrachtet man das zugrundeliegende physische System als einen ungerichteten Graphen und extrahiert die vorhandenen Elemente sowie deren Werte, erhält man die Basismenge in welche die virtuellen Strukturen eingebettet werden sollen. Siehe Abbildungen 1, 2 und 3.

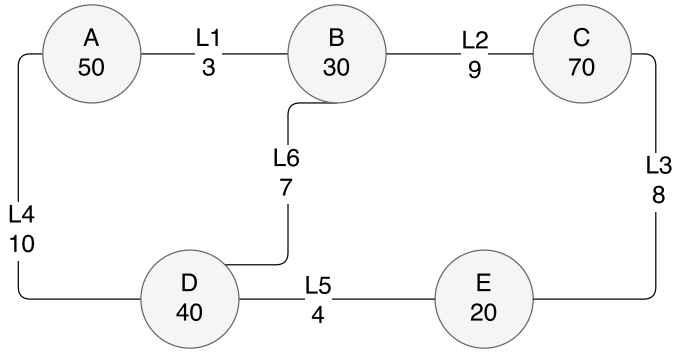

Abbildung 1: Ungerichteter Graph der physischen Infrastruktur

$$G^S = \{N^S, L^S\}$$

Abbildung 2: N steht für die Menge der physischen Knoten, L für die Menge der physischen Links

$$G^S = \{\{ (A^S,50),(B^S,30),(C^S,70), (D^S,40),(E^S,20)\},$$
$$\{ (L1^S,3),(L2^S,9),(L3^S,8), (L4^S,10),(L5^S,4),(L6^S,7)\}\}$$

Abbildung 3: Netz aus Abbildung 1 in vereinfachter Mengennotation

An dieser Stelle muss darauf verwiesen werden, dass die Mengen in der Realität um einiges komplexer sind. Beispielsweise kann aus der Menge aus Abb. 3 nicht mehr rekonstruiert werden, welcher Link mit welchen Knoten verbunden ist.

Ein „virtual network request“ (in Folge „VNR“ genannt) kann hierbei ebenfalls als ein ungerichteter Graph gesehen und genauso in eine Menge übersetzt werden. Abbildungen 4, 5 und 6

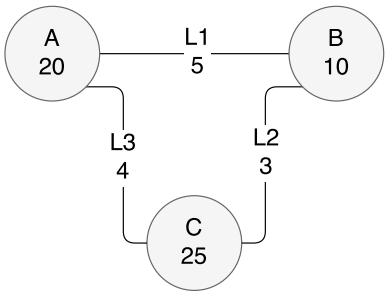

Abbildung 4: VNR als ungerichteter Graph

$$G^V = \{N^V, L^V\}$$

Abbildung 5: N steht für die Menge der virtuellen Knoten, L für die Menge der virtuellen Links

$$G^V = \{\{ (A^V,20),(B^V,10),(C^V,25)\} , \{ (L1^V,5),(L2^V,3),(L3^V,4)\}\}$$

Abbildung 6: VNR aus Abbildung 4 in vereinfachter Mengennotation

Gesucht ist nun eine Abbildungsfunktion $f : G^V \rightarrow G^S$

In der Regel ist es üblich, nicht nur eine einzelne Requests auf ein physisches System abzubilden, sondern gleich mehrere auf einmal. Ebenso wäre es theoretisch möglich, dass ein Infrastruktur-Provider gleich mehrere getrennte unabhängige physische Strukturen betreibt, und somit mittels VNE das beste System für die Abbildung der Menge von VNRs eroiren möchte. Dies könnte beispielsweise der Fall sein, wenn ein Serviceprovider ein VNR mit der Bedingung „alle Elemente mögen sich am selben Standort befinden, egal an welchem“ in Auftrag gibt, und der Infrastruktur-Provider über mehrere Hardware-Standorte verfügt.

Die Attributmenge bei den grundlegenden VNE-Algorithmen beschränkt sich zumeist auf CPU- und Netzwerk-Leistung. Lokalität, GPU-Leistung und RAM-Menge wären einige weitere mögliche Standard-Attribute. Ein großes Problem bei der Berechnung des optimalen Mappings findet sich bei der Berechnungszeit. Die endliche Beschränkung der Knoten- sowie Link-Resourcen und die „on-line nature“ von VNRs stellen zusätzliche Hindernisse dar. Da selbst Lösungen für einfache Anfragen (geringe Anzahl von Knoten und Links), welche wenige Attribute beinhalten, exponentiellen Rechenaufwand benötigen, steigt der Aufwand sowohl mit der Anzahl der abzubildenden Knoten und Links, als auch mit steigender Attributanzahl dementsprechend. Teilweise gilt das Problem als rechnerisch unlösbar, grundsätzlich aber befinden wir uns im Komplexitätsbereich der NP-Vollständigkeit [MRR13]. Da die Erhöhung der Attributmenge - wie bereits genannt - grundsätzlich nicht positiv zur Laufzeit der existierenden Algorithmen beiträgt, werden wir den Komplexitätsbereich beim Hinzufügen von Sicherheitsanforderungen nicht verlassen. Dennoch gibt es Möglichkeiten, die den Rechenaufwand reduzieren können. Zunächst folgt jedoch eine Übersicht über die Gefahren, welche bei VNE eine Rolle spielen.

3 Sicherheitsaspekte virtueller Netzinfrastrukturen

3.1 Neue Verwundbarkeiten in virtualisierten Umgebungen

Wie in der Einleitung angedeutet bietet Netzwerkvirtualisierung einige Vorteile gegenüber bisherigen Netzarchitekturen. Durch die Einführung einer weiteren Schicht zwischen Hardware und Anwendungssoftware („Ebene virtueller Netze” in Abbildung 7) und dem Hosten verschiedener virtueller Netzwerke auf einem gemeinsamen Substratnetz tun sich aber auch verschiedene – im Gegensatz zu herkömmlichen, nicht-virtualisierten Architekturen – neue Verwundbarkeiten auf. [GCH+16, NW, WDWY10, GR05, DMT11] haben bereits einige Sicherheitsrisiken analysiert, welche im Folgenden klassifiziert und ergänzt werden.

Sicherheitsrisiken in virtualisierten Netzinfrastrukturen lassen sich auf verschiedene Weisen wie z.B. nach ISO/OSI-Schicht, nach Verletzung der klassischen C.I.A.-Aspekte, nach Schicht in der NV-Architektur, oder aus Sicht des SPs bzw. InPs klassifizieren. Da dieses Kapitel sich aber auf durch Netzvirtualisierung gegenüber herkömmlichen Netzinfrastrukturen neu hinzukommende Risiken konzentriert, liegt der folgenden Klassifizierung zur Verdeutlichung der Angriffsrichtungen die in Abbildung 8 dargestellte Struktur zugrunde, die sich auf der Drei-Schichtenarchitektur der Netzwerkvirtualisierung (Abbildung 7) ableitet.

Sicherheitsrisiken werden zunächst nach solchen technischer, organisatorischer bzw. unternehmerischer und rechtlicher Art geordnet. Im Zentrum der Betrachtung stehen dabei die technischen Risiken, welche wiederum nach Angriffsrichtungen ‚*von NI ausgehend*‘, ‚*von VN/VM ausgehend*‘ und ‚*vom User ausgehend*‘ gegliedert werden. In jeder dieser Kategorien wird nach Angriffsziel ‚*gegen NI*‘, ‚*gegen VN/VM*‘ und ‚*gegen User*‘ unterteilt.
Da die Kategorien ‚*I.1 von NI ausgehend gegen NI*‘ und ‚*III.3 vom User ausgehend gegen User*‘ in virtualisierten Netzwerkumgebungen keine neuen Angriffsszenarien eröffnen,

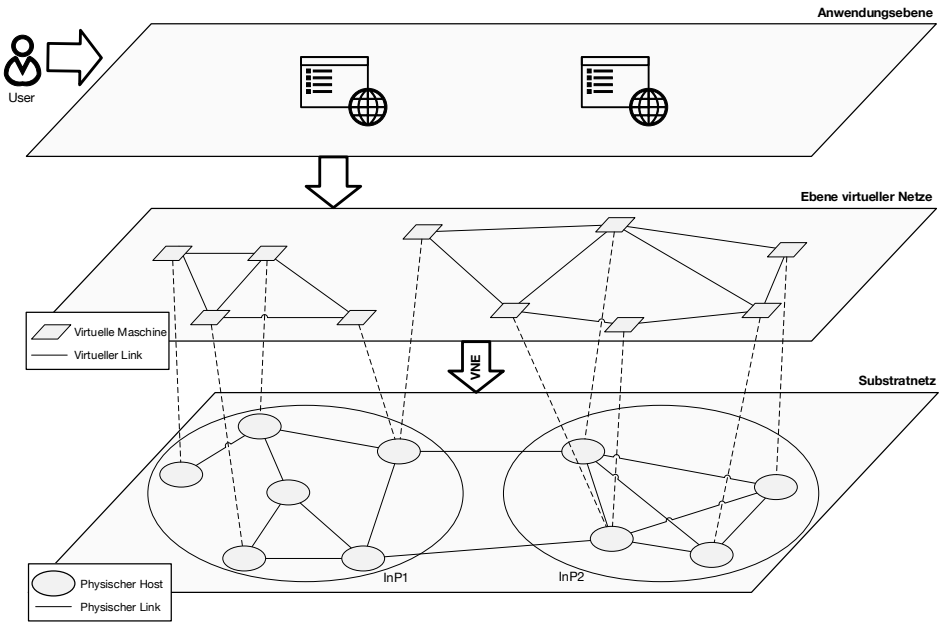

Abbildung 7: Drei-Schichtenarchitektur der Netzwerkvirtualisierung.
Das Substratnetz zweier InP hostet zwei virtuelle Netze.

werden sie hier dem Ziel des Kapitels entsprechend nicht tiefergehend ausgeführt.

3.1.1 Technischer Art

Dieses Kapitel analysiert sich durch NV ergebende Sicherheitsrisiken technischer Art. Der Kapitelaufbau folgt der Klassifizierung in Abbildung 8.

I.2 & I.3 Von NI ausgehend gegen VN/VM und User. Physische Hosts bieten ihren VMs Ressourcen an. Alle Dienste und Anwendungen der VMs werden letztlich auf dem physischen Host ausgeführt und auch alle Daten auf ihm gespeichert. Dies eröffnet für den physischen Host prinzipiell die Möglichkeit eines Monitorings der VM-Aktivitäten, was ab einer gewissen Intensität über Verwaltungsbelange hinausgehen und Privatsphärenanforderungen widersprechen wird. Auf demselben Weg lassen sich auch vertraulichkeitsverletzende Sniffing- oder Spoofing-Attacken gegen VM bzw. VN starten.
Da alle ihre Rechenoperationen letztlich auf dem physischen Host ausgeführt werden, ist es für eine VM nur schwer möglich, sich gegen solche Angriffe zu wehren. Eine Verschlüsselung der eigenen Daten kann einer VM nicht ohne Weiteres helfen, da auch die Schlüsseldatei innerhalb der VM und damit auf dem physischen Host gespeichert werden muss.

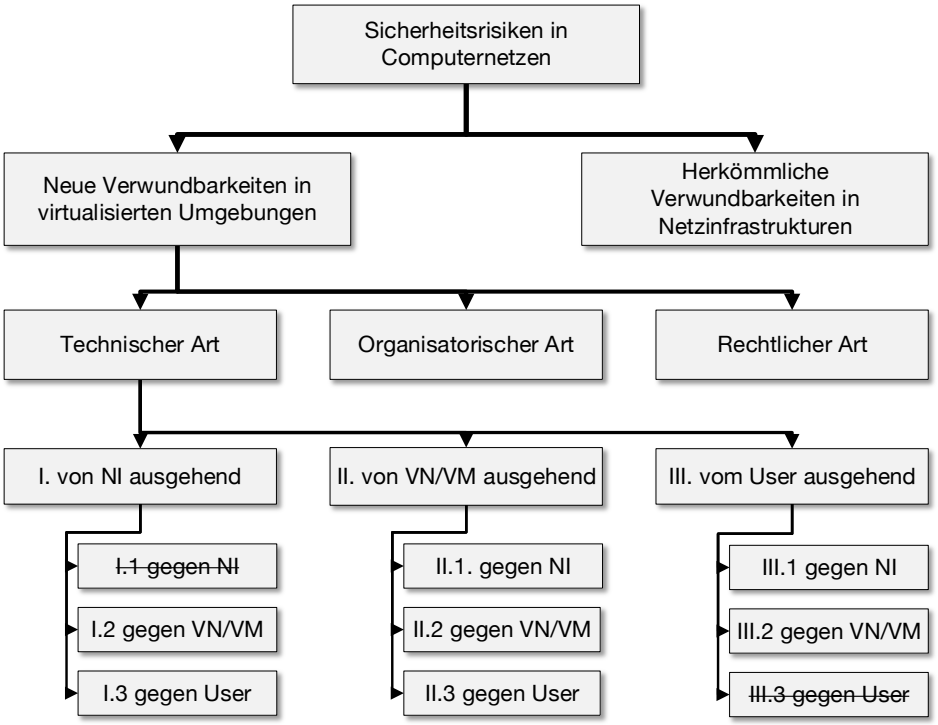

Abbildung 8: Klassifizierung von Gefahren in virtualisierten Netzwerken
NI = Netzwerkinfrastruktur / Substratnetz. Der physische Host einer VM ist Teil der NI.
User = Endsystem bzw. in VNs implementierter Service oder Nutzer
Kategorie I.1 und III.3 beinhalten keine durch NV neu hinzukommenden Risiken.

Auch Manipulation des legitimen Datenverkehrs, (gezieltes) Verwerfen von empfangenen Paketen bzw. das Einschleusen schadhafter Nachrichten bieten eine weitere Möglichkeit der Kompromittierung, gegen die VNs und Endsysteme wohl schutzlos ausgeliefert sind. Durch derartige Aktionen oder auch unzureichende Sicherungsmaßnahmen gegen Datenabfluss kann der physische Host in den SLAs vereinbarte Bestimmungen verletzen, was gerade bei Drittanbietern als Hostingpartner eine Rolle spielt.

Von VN/VM ausgehend Die gemeinsame Nutzung von Ressourcen wie Speicher und Netzwerkkarten eröffnet VMs neue Angriffsmöglichkeiten gegen ihr Substratnetz, gegen andere VNs/VMs und gegen User.

- **II.1 Von VN/VM ausgehend gegen NI/ihren physischen Host.** Das Bereitstellen von Ressourcen für VMs ist auch für den physischen Host nicht ohne Risiko. Schadhafte oder bösartige VMs können Verwundbarkeiten ihres physischen Host über zugeteilte Ressourcen angreifen. Ohne hinreichende Restriktionen kann eine

VM über ihr zugeteiltes Kontingent hinaus bspw. wichtige Speicherbereiche manipulieren oder z.B. durch übermäßige Reservierung von CPU-Zeiten eine Denail of Service Attacke gegen den physischen Host bzw. das Substratnetz fahren. Da Host und virtuelle Netztopologie aus der Ferne konfigurierbar sind, stellt das Einschleusen von konstruierten Nachrichten des verwendeten Netzwerkmanagementprotokolls auf oder durch die Netzwerkkarte des physischen Hosts einen weiteren gefährlichen Angriffsvektor dar.
Nach Eindringen in oder Übernahme des Hosts – einem „break of isolation“ im ersten Sinne [WDWY10] – könnte eine schadhafte VM dann ihr Kontingent an Ressourcen beeinflussen, netztopologische Informationen sammeln, andere Netzwerkressourcen oder -infrastrukturkomponenten angreifen und so beispielsweise Dienste anderer VMs oder VNs behindern.

- **II.2 Von VN/VM ausgehend gegen VN/VM.** Neben den herkömmlichen Angriffsszenarien zwischen Maschinen im selben Netzwerk, ergeben sich v. a. aus der gemeinsamen Nutzung von Ressourcen neue Angriffsvarianten.[1] Ein Angreifer kann sich nun gezielt Ressourcen auf denjenigen physischen Maschinen mieten, auf denen auch sein Angriffsziel gehostet wird, um so erleichterten Zugang zu deren Verwundbarkeiten zu erlangen. Durch Eindringen oder Übernehmen gewisser Ressourcen des gemeinsamen physischen Hosts kann eine schadhafte VM ggfs. Verwundbarkeiten anderer VMs ausnutzen oder durch Cross-VN-side-channel-Attacken vertrauliche Informationen gewinnen und Daten manipulieren. Ein Beispiel einer Integritätsverletzung mittels einer solchen Attacke findet sich in [RTSS09].
  Da i.d.R. nur virtuelle Netzwerkkarten zugeteilt werden, kann jede VM potentiell den gesamten Datenverkehr aller VMs bzw. VNs auf derselben physischen Netzwerkkarte lesen. Ein Monitoring anderer VMs auch aus anderen VNs, ein „break of Isolation“ im zweiten Sinne bedroht deren Vertraulichkeit. Durch Belauschen des Netzwerkverkehrs lassen sich ggfs. auch Dienste gemeinsam gehosteter VN reproduzieren und dadurch bspw. ein live Videostreaming breiter zugänglich machen. [NW]
  Sollte es einer VM gelingen kritische Teile ihres physischen Hosts zu übernehmen, so stehen ihr zusätzlich die im Abschnitt *I.2 & I.3 Von NI ausgehend gegen VN/VM und User* aufgeführten Angriffsvektoren offen.

- **II.3 Von VN/VM ausgehend gegen User.** Auch ein virtuelles Netzwerk kann mit herkömmlichen Methoden wie Monitoring der Nuteraktivitäten oder dem Einschleusen von konstruierten Nachrichten zur Störung oder Abbruch von Peer-to-Peer-Verbindungen Einfluss auf den Nutzverkehr seiner User nehmen.

Vom User ausgehend Einem Benutzer oder schadhaften Anwendungsprogramm stehen auch in virtualisierten Netzwerkumgebungen die bekannten Methoden der Störung durch z.B. Herbeiführen von Überlastsituationen in VN oder NI offen.

[1] Da die Angriffstechniken von VMs gegen VMs auf anderen physischen Hosts vergleichbar mit herkömmlichen Angriffen in nicht-virtualisierten Netzinfrastrukturen sein dürften, werden hier hauptsächlich VMs auf demselben physischen Host betrachtet.

- **III.1 Vom User ausgehend gegen NI.** Da sich die virtuelle Netztopologie im VNE-Prozess laufend ändert, müssen Netzwerkkomponenten wie Switches und Router dynamisch umprogrammierbar sein. Dies jedoch ermächtigt Angreifer solche ggfs. mit Codeexpliots wie Bufferoverflows o. Ä. zu kompromittieren und für ihre Zwecke zu nutzen oder einen Denail of Service herbeizuführen.
  Daneben besteht die Chance auch Netzwerkknoten anzugreifen. Gelingt es z.B. mit einem Rootkit wie BluePill [RT08] – als Vorbereitung für weitere Angriffe – einen Hypervisor zu übernehmen, wird so gleichzeitig die Kontrolle über alle gehosteten VMs erlangt. Auch eine VM lässt sich als Rootkit instrumentalisieren. [WDWY10]

- **III.2 Vom User ausgehend gegen VN/VM.** Aus der dynamischen Natur virtueller Netzwerktopologien ergeben sich neue Verwundbarkeiten: Während der Migration im Livebetrieb eines VNs ist eine Man-in-the-Middle-Attacke möglich, mit der Informationen über und Inhalte des migrierenden VNs erlangt werden können. [NW]
  Auch die Manipulation von Speicherbereichen der VMs ist während der Migration umsetzbar und lässt sich sogar automatisieren. [OCJ08]
  Die Notwendigkeit die gesamte virtuelle Netzwerkstruktur aus der Ferne umkonfigurieren zu können, erschließt weitere Angriffsziele: Attacken auf die VN-Managementtools durch z.B. Cross-Site-Scripting, SQL-Injection etc. werden lohnend, da auf diese Weise effizient Kontrolle über das gesamte Netzwerk gewonnen werden kann.

Durch Virtualisierung eröffnen sich also eine Reihe neuer Verwundbarkeiten und Angriffsmöglichkeiten technischer Art, die hauptsächlich in der gemeinsamen Nutzung von Ressourcen begründet liegen.

3.1.2 Organisatorischer Art

Unter ‚organisatorischen' Risiken werden hier Risiken für Unternehmen verstanden, die im Zusammenhang mit Virtualisierung stehen.

Wie im Kapitel 3.1.1 dargestellt, eröffnet Netzvirtualisierung eine Reihe neuer Verwundbarkeiten für gehostete Systeme. Gerade für Unternehmen stellt die teils deutliche Gefährdung der Vertraulichkeit und Integrität von Firmen- oder Kundendaten ein ernstzunehmendes Problem dar. Da Netzvirtualisierung oftmals via Cloudcomputing abgewickelt wird, erhöht sich das Risiko eines Datenlecks nochmals durch den Up- und Downloadprozess von Daten.
Mögliche Richtlinien zur Beschränkung von Anzahl, Art oder Eigentümer gemeinsam gehosteter VNs/VMs, welche sich z.B. als komplexe Sicherheitsvektoren mit dem in Kapitel 4.2 vorgestellten Algorithmus realisieren ließen, schränken Flexibilität der Netzarchitektur und Kostenvorteil der NV zwar ein, können jedoch bei ausreichendem Sicherheitslevel der physischen Hosts einen gewissen Schutz gewährleisten. Vor- und Nachteile eines solchen Ansatzes werden in Kapitel 4.2 und 4.3 diskutiert.

Ein großer Vorteil virtueller Maschinen besteht in der Leichtigkeit Momentaufnahmen (Snapshots) für Backup- oder Migrationszwecke zu erstellen. Beim Wiedereinspielen dieser werden jedoch ggfs. zwischenzeitlich deaktivierte Accounts, veraltete Sicherheitsricht-

linien oder mittlerweile gepatchte Schwachstellen wieder produktiv gesetzt. Organisationen müssen hierfür einen geeigneten Prozess etablieren, der auch das Patchmanagement häufig offline gehender VMs regelt. Der Umgang mit solchen VMs ist schwierig, da Würmer meist relativ schnell alle verwundbaren Systeme infizieren. Geht die VM danach offline wird Malware darin ggfs. nicht entdeckt und die Wurminfektionselle startet erneut beim Onlinegehen der VM. [GR05]

3.1.3 Rechtlicher Art

Da Substratnetze nicht zwingend räumlich eng beschränkt sein müssen, könnte es passieren, dass gewisse Teile des virtuellen Netzes (dynamisch / lastbedingt) auf Knoten oder in ein Land gemappt werden, welches mit den Auflagen des Unternehmens zu Datenschutz, Privatsphäre oder IT-Securitystandards nicht vereinbar ist. Hinzu kommen eventuelle aus technischen Risiken wie Sniffing resultierende Konflikte mit rechtlichen Vorschriften zu bspw. Datenschutz.

3.2 VNE-Relevante Gefahren

[TODO] Benennung von (Kategorien an) Gefahren aus dem vorherigen Kapitel, denen mit dem VNE-Prozess begegnet werden kann.

4 Vermeidung von Gefahren via Secure VNE (SVNE)

Da wir nun einen Überblick zu VNE sowie den bestehenden Gefahren gegeben haben, widmen wir uns nun zwei verschiedenen SVNE-Lösungsansätzen. Der Detailgrad wurde so gewählt, dass man einen guten Einblick in die Arbeitsweise der Ansätze bekommt und die Problematik der zusätzlichen Last der Sicherheitsaspekte erkennt.

4.1 Ansatz 1

Hierbei beschäftigen wir uns mit dem Lösungsansatz aus [WCC16]. Das Hauptaugenmerk bezüglich der Sicherheitsaspekte legen Wang et. al. auf Traffic-Verschlüsselung und die Separierung von VMs unterschiedlicher Trust-Levels. Die drei folgenden strukturellen Aspekte werden betrachtet und in dementsprechende Levels eingeteilt.

- Netze:
  Ein VNR wird als Netzwerk betrachtet und seinem Level entsprechend isoliert. „High“ fordert und beansprucht ein gesamtes Netz bzw. Subnetz der physischen Infrastruktur für sich. Hiermit soll die Wahrscheinlichkeit für DOS-Angriffe oder ähnliche vermindert werden, da „Mehr-Parteien-Netzwerke“ die Hauptschwachstel-

len für solche Angriffe darstellen [Liu10]. Auch Sniffing durch kompromittierte Hosts im selben Netz wird durch diese Maßahme verhindert. Während „high“ Resourcenteilung somit komplett verweigert, lässt der Level „medium“ zumindest ein gemeinsam genutztes Netz für VNRs vom selben Eigentümer zu.

- Knoten:
  Die einzelnen virtuellen Knoten eines VNR stellen Isolierungsanforderungen an die physischen Knoten. Wie auch beim Netzaspekt, werden die Levels „high”, „medium“ und „none“ definiert und umgesetzt.„high“ fordert die alleinige Existenz eines VNR-Knotens, „medium“ lässt VNR-Knoten vom selben Eigentümer auf ein und demselben physischen Knoten zu. Durch diesen Plan sollen Angriffe von VM zu VM über gemeinsam genutzte Resourcen unterbunden werden. Zusätzlich wird das Risiko eines Angriffs vom physischen Host verringert, da der Angriffsvektor „VM zu physischem Host“ eliminiert wird.

- Links:
  End-to-End(E2E), Point-to-Point(P2P) und „none“ sind die hier wählbaren Levels. Während E2E nur an den Endpunkten Verschlüsselungskapazitäten zu Verfügung stellen muss, benötigt P2P diese Kapazitäten an allen Hops des abgebildeten Links. Die heutzutage sehr gängigen man-in-the-middle Angriffe sollen dadurch wesentlich erschwert werden.

VNRs werden nun, wie in Kapitel 2 bereits gezeigt, in ungerichtete Graphen mit Standardanforderungen transformiert. Zusätzlich werden hier auch noch Sicherheitsanforderungen integriert. Siehe Abbildung 9.

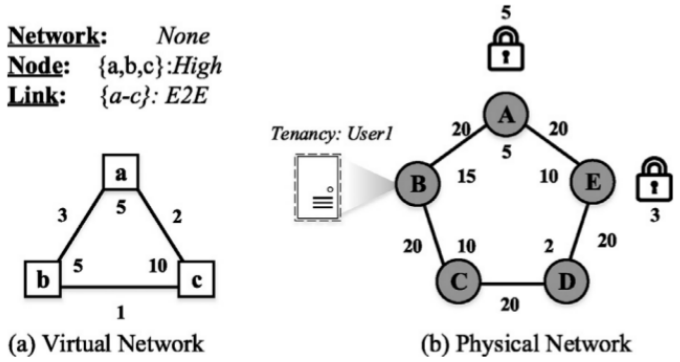

Abbildung 9: VNR und physisches System mit Sicherheitsmerkmalen [WCC16]

Nun wird ein vier-stufiges Pre-Processing durchgeführt, um die Berechnung der optimalen Abbildung der VNRs zu vereinfachen. Hierbei werden als ersten die Standardanforderun-

gen der virtuellen Knoten mit den zu Verfügung stehenden Kapazitäten der physischen Knoten verglichen. Sollten physische Knoten bestimmte Anforderungen nicht erfüllen, werden sie aus der Berechung ausgeschlossen. Der zweite Schritt ist den Netzen gewidmet. Sollte ein physischer Knoten, welcher den Anforderungen des Netzes nicht genügt sich in einem Kandidaten-Netzwerk befinden, wird das Subnetz aus den Berechnungen ausgeschlossen. Im dritten Schritt werden die Sicherheitsanforderungen der einzelnen Knoten betrachtet und nicht entsprechende physischen Knoten werden abermals entfernt. Der letzte Schritt widmet sich den Sicherheitsanforderungen aus der Links und streicht Links aus den weiteren Berrechnungen, welche den Verschlüsselungsanforderungen der virtuellen Links nicht genüge tun. Siehe Abbildung 10.

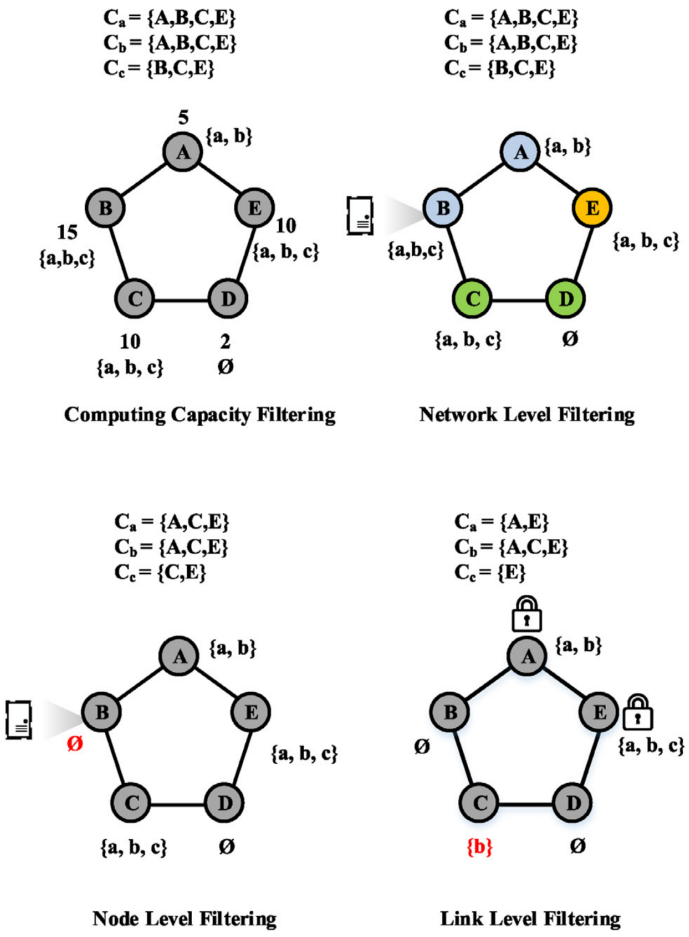

Abbildung 10: 4-stufiges Pre-Processing[WCC16]

Mit den aus dem Pre-Processing gewonnen Information bildet man nun einen Hilsgraphen, welcher die übrig gebliebenen Möglichkeiten der Einbettung zeigt. Siehe Abbildung 11.

Dieses Pre-Processing reduziert das SVNE-Problem auf ein „multi-commodity-flow“ Problem. [RA93] Im weiteren Vorgehen werden nun zwei Fälle unterschieden.„Path-splitting”, im weiteren SVNE-PS und „no-path-splitting“, im weitern SVNE-NPS, sind zusätzliche Sicherheitsvorgaben, welche im Vorfeld definiert werden müssen, um die Wahl des Algorithmus zu ermöglichen. Sollte SVNE-PS gewählt werden, beziehen sich die Gleichungen nur auf die Erfüllung der geforderten Attribute. Sollte SVNE-NPS gewählt werden, werden die für Link-Abbildungen verantwortlichen Gleichungen ersetzt.

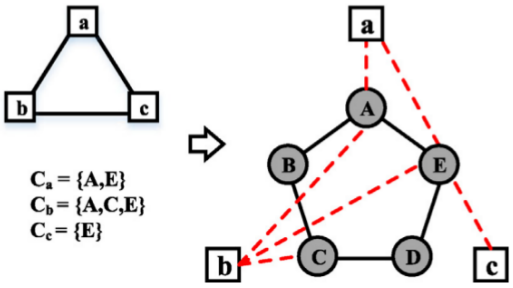

Abbildung 11: Hilfsgraph[WCC16]

Die Berechnungen wurden unter Verwendung von CPLEX und durch k-shortest-path anhand aller abzubildenden Elemente limitiert[hi]. Für die Testumgebungen wurden zufällig erzeugte VNRs mit 2 bis 10 Knoten und halbsovielen Links erzeugt. Auch die Rechenkapazität wie Bandbreite wurden durch Zufallsgeneratoren mit Werten im Bereich von 1 bis 10 gewählt und verteilt. Die Anzahl der VNRs beträgt einer Poisson-Verteilung nach einen Durchschnitt von 4 VNR pro 100 Zeiteinheiten, mit einer jeweiligen durchschnittlichen Einsatzzeit von 1000 Zeiteinheiten.

Die zugrundeliegenden physischen Netze wurden ebenfalls zufällig erzeugt und beinhalteten 10 bis 50 Knoten, sowie halbsoviele Links. Die Rechen- sowie Bandbreiten-Kapazitäten wurden gleichmäßig verteilt und enthielten Werte zwischen 1 und 50. Die Verschlüsselungskapazitäten der Knoten wurde über die gesamte Testreihe ebenfalls gleichmäßig verteilt.

Die folgende Statisk zeigt einen Durchschnittsvergleich zwischen dem, als Standard gewählten, VNE-Algorithmus und den beiden SVNE-Varianten PS und NPS[QH13]. Dazu sei gesagt dass sämtliche Algorithmen ihre Berechnungen anhand des Hilfsgraphen durchgeführt haben und durch k=3 bzw. k=4 limitiert wurden. Abbildung 12

Man sieht hier einen deutlichen Zeitvorsprung von PS und NPS gegenüber dem Standard-Algorithmus, was auf eine Optimierung der Algorithmen auf den Input des Pre-Processings schließen lässt. Da hier jedoch auch der Standard-Algorithmus vom Pre-Processing profitiert, wäre ein Vergleich, welcher die Dauer des Pre-Processings mit aufnimmt und den Standard-Algorithmus ohne Pre-Processing arbeiten ließe, anschaulicher und aussagekräftiger. Dennoch ist ein Punkt durchaus überraschend und bemerkenswert: Der Standard-Algorithmus benötigt deutlich länger, obwohl er keine Sicherheitsaspekte mitbeurteilt, im Gegenzug zu seinen Kontrahenten.

In welchem Komplexitätsbereich sich das Pre-Processing befindet wird hier leider nicht erwähnt.

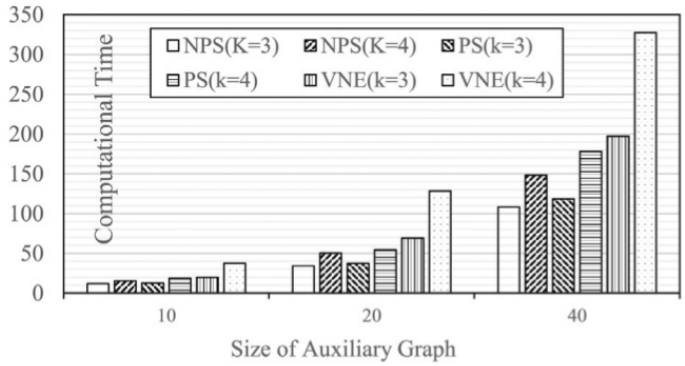

Abbildung 12: Evaluationsergebnis [WCC16]

4.2 Ansatz 2

Im zweiten Ansatz widmen wir uns dem Modell aus [SL15]. Dieser Ansatz arbeitet mit abstrakten Sicherheitslevels. Im Folgenden werden zwar Skalare hierfür verwendet, was allerdings nicht zwingend so vorgesehen ist. Stattdessen wäre eine Verwendung komplexerer Sicherheitsvektoren möglich, welche bei weitem detailliertere Sicherheitsmerkmale beschreiben könnten. Des weiteren verfügt jeder physische wie auch virtuelle Knoten über zwei Sicherheitswerte: Anforderungslevel und (eigenes) Sicherheitslevel. Der Anforderungslevel definiert das Minimum-Level des Gegenübers, der Sicherheitslevel definiert die eigenen gewünschten Sicherheitsmerkmale. Virtuelle Links verfügen über Anforderungslevels, physische Links nur über Sicherheitslevels. Die vorausgesetzten Basisanforderungen, welchen alle VNRs unterliegen, beschränken sich auf 4 Regeln, welche mittels der zuvor genannten Merkmale umgesetzt werden:

- Ein physischer Knoten sollte einen Sicherheitslevel garantieren, der höher ist als die Anforderungen der darauf abzubildenden virtuellen Knoten.
- Der Sicherheitslevel des virtuellen Knotens sollte höher sein, als das Anforderungslevel des physichen Knotens.
- Alle virtuellen Knoten, welche auf den selben physischen Knoten abgebildet werden, sollten über einen ausreichenden Sicherheitslevel verfügen.
- Der Anforderungslevel des virtuellen Links sollte stets niedriger sein, als das Sicherheitslevel des physischen Links.

Der erste und zweite Punkt dienen der Risikominimierung zwischen physischen und virtuellen Hosts in beide Richtungen. Der dritte Punkt stellt sicher, dass keine zu schwach

gesicherten VMs zusammen mit stark gesicherten VMs abgebildet werden und somit keine offentsichtlichen Schwachstellen den physichen Knoten samt seiner vituellen Hosts gefährden. Punkt 4 sorgt für ausreichende Sicherheitmechanismen der physischen Links.

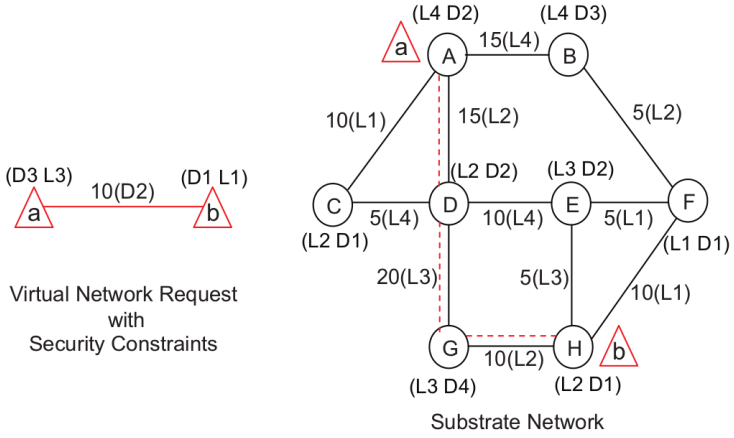

Abbildung 13: VNR und physisches System mit Sicherheitsmerkmalen[SL15]

Wie man in Abbildung 13 sieht, werden auch bei diesem Modell VNRs und physische Netze in ungerichtete Graphen transformiert. Die in Klammern gestellten Parameter beschreiben Anforderungslevel(D) und Sicherheitslevel(L). Die hier durchgeführte Abbildung berücksichtigt nicht nur die Sicherheitsanforderungen aller Parteien unter Berücksichtigung der Basisanforderungen, sondern achtet zusätzlich noch auf Kostenminimierung. Insgesamt existieren drei Abbildungsmöglichkeiten, die gewählte ist jedoch die günstigste, unter dem Aspekt keine Sicherheitsresourcen zu verschwenden. Dieser VNR hätte auch auf EDGH abgebildet werden können, wobei der Link ED über einen Sicherheitslevel verfügt, welcher höher als nötig ist. Würde dies nicht beachtet werden, könnte es zu unnötigen Engpässen bei der Behandlung von VNRs mit höheren Anforderungslevels kommen. Um ein solches Verhalten zu erreichen, wurden zusätzlich Kosten- und Nutzen-Funktionen in die Berechnung integriert.

Das außergewöhnliche bei diesem Ansatz ist die Verwendung zweier unterschiedlicher Algorithmen, welche sich gegenseitig ergänzen: uSAV und cSAV.

- uSAV
  uSAV, der unkooridinierte zwei-Phasen-Algorithmus, behandelt Knoten- und Link-Abbildungen getrennt voneinander. Dementsprechend liegt die Schwäche von uSAV in der Abbildung von non-splittable-links. uSAV ist ein terminierender Algorithmus, welcher die Abbildung der Knoten priorisiert und dahingehend ein sehr gutes Ergebnis mit wenig Aufwand erzielt. Jedoch kann es, durch die Priorisierung der Knoten,

zu hohen Kosten bei der Link-Abbildung kommen.

- cSAV
  cSAV, der koordinierte Algorithmus, behandelt Knoten und Links gemeinsam, und kann zur Optimierung, des bereits von uSAV gelieferten Ergebnisses verwendet werden. cSAV liefert zwar optimalere Ergebnisse als uSAV, benötigt aber auch mehr Zeit. Hier muss Zeitaufwand mit gelieferter Leistung abgewogen werden.

Beide Varianten nutzen die selben Heuristiken. Der Author schlägt für ideale Ergebnisse eine kombinierte Benutzung der beiden Varianten vor.

Die Testumgebung dieses Ansatzes wurde mittels GT-ITM-Tool erstellt [EWZ].

Die physischen Netze wurden in der Größenordnung eines mittleres ISP angesetzt, und betrugen 100 Knoten und 500 Links. Die Bandbreite und Rechenkapazität wurden, wie auch im vorherigen Ansatz, gleichmäßig verteilt und betrugen Werte zwischen 50 und 100. Die abstrakten Sicherheitslevels der physischen Elemente wurden zwischen 0 und 4 gleichmäßig verteilt. Die Anforderungslevels wurden ebenfalls aus dem Bereich 0-4 gewählt, und so verteilt, dass kein physisches Element ein höheres Anforderungslevel als Sicherheitslevel besitzt.

Die VNRs beinhalteten 2 bis 20 Knoten sowie halbsoviele Links. Die Bandbreite- und Rechenkapazitäts-Forderungen wurden zwischen 0 und 50 gewählt und ebenfalls gleichmäßig verteilt. Die zeitlichen Parameter wurden auf durchschnittlich (Poisson-Verteilung) 5 Anforderungen pro 100 Zeiteinheiten begrenzt. Die Verwendungszeit eines VNRs folgt einer Exponential-Verteilung und betrugt im Durchschnitt 500 Zeiteinheiten. Eine einzelne Simulation erhielt 1500 VNRs und dauerte 30000 Zeiteinheiten.

Als Vergleichswert wurde der Algorithmus aus [MY08] verwendet. Die Tests wurden für drei, anhand der Link-Split-Ratio unterschiedenen Szenarien ausgelegt: High-LSR(mehr als 80% der Links dürfen gesplittet werden), Low-LSR(weniger als 20%) und VLSR(varied LSR). Da letzteres Testszenario am meisten Bezug zur Realität hat, werden wir hier nur diese Ergebnisse vorstellen.

Man sieht in Abbildung 14 die drei, am schwersten gewichteten Aspekte dieses Ansatzes. Der Kosten-/Nutzen-Wert sinkt mit zunehmender LSR, während die Akzeptanz und der Nutzen gegenteiliges Verhalten zeigen. Die vorhandenen Laufzeitanalysen der Algorithmen bestätigen die Aussagen der Authoren. uSAV führt hier mit Werten zwischen 6 und 10 Minuten, während cSAV erst nach 14 bis 25 Minuten brauchbare Ergebnisse für einzelne VNRs liefert. Hinzuzufügen sei hier, dass für alle Simulationen der einfachste Sicherheitsvektor verwendet wurde und bei Verwendung eines ausgeprägten Vektors die Rechenzeit mit Sicherheit andere Maße annimmt.

[GENAUERE ERGEBNISBESCHREIBUNG]

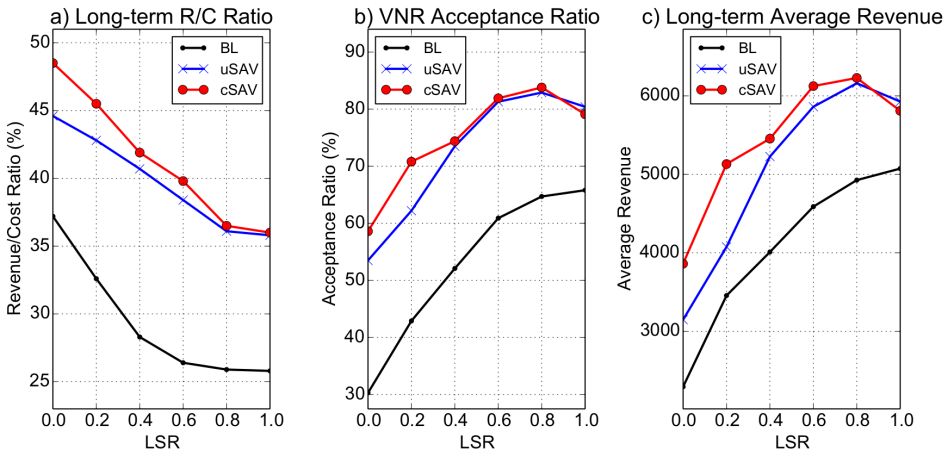

Abbildung 14: VNR und physisches System mit Sicherheitsmerkmalen[SL15]

4.3 Vergleich der beiden Ansätze

Um nun die beiden vorgestellten Ansätze in einen Vergleich zu bringen, möchten wir uns hier nicht auf die Ergebnisse der Testszenarien konzentrieren, sondern auf die unterschiedlichen Herangehensweisen.

[unterschiedliche Testumgebungen... offline vs online]

Der - bei dieser Arbeit wichtigste - Unterschied zwischen den beiden Ansätzen bezieht sich auf Granularität der Sicherheitsanforderungen. Ansatz 1 versucht hier mittels drei, auf Teilelemente bezogene, Sicherheitspläne einen größtmöglichen Schutz zu bieten. Ein Vorteil dieses Vorgehens liegt klar in der Einfachheit der Umsetzungsmöglichkeit und der Limitierung zusätzlicher Parameter, durch welche die Rechenzeit der Algorithmen im Zaum gehalten wird. Nachteile könnten sich aus der Starrheit des Sicherheitssystems und dem Kostenfaktor ergeben. Ansatz 2 bietet hier etwas mehr Flexibilität sowie Präzision. Die entsprechenden Vektoren müssten hierfür zwar erst definiert werden, bieten dafür aber Erweiterbarkeit, was ein Vorteil bezüglich Flexibilität und Nachteil bezüglich Rechen- sowie Zeit-Aufwand darstellen kann. [detailliertere Ausführungen für diesen Punkt folgen]

Ein weiterer Unterschied ist die Herangehensweise bei der Berechnung. Pre-Processing kann das Abbilden der VNRs deutlich verkürzen, muss jedoch sequenziell durchgeführt werden und benötigt bei komplexeren VNRs vermutlich auch eine nicht zu vernachlässigende Inanspruchnahme von Rechenzeit. Ansatz 2 versucht an dieser Stelle mittels Heuristiken und zweier unterschiedlicher Algorithmen das gewünschte Ziel zu erreichen.

5 Schlussfolgerung und Ausblick

Frage: Auf welcher Ebene wird virtualisiert? Auf IP-Ebene? Was ist dann aber mit IP-Support-Protokollen (ARP)? Oder nicht-IP-Protokollen? Will ich ein Netzwerk virtualisieren, oder nur den IP-Verkehr? Verkapselung führt zu Leistungseinbußn. [CDRS07]

Fragen aus den Herausforderungen, die sich aus Sicherheitsrisiken ergeben...

Literatur

[APST05] Thomas Anderson, Larry Peterson, Scott Shenker und Jonathan Turner. Overcoming the Internet impasse through virtualization. *Computer*, 38(4):34–41, 2005.

[BOB+12] Leonardo Richter Bays, Rodrigo Ruas Oliveira, Luciana Salete Buriol, Marinho Pilla Barcellos und Luciano Paschoal Gaspary. Security-aware optimal resource allocation for virtual network embedding. In *Proceedings of the 8th International Conference on Network and Service Management*, Seiten 378–384. International Federation for Information Processing, 2012.

[CDRS07] Serdar Cabuk, Chris I Dalton, HariGovind Ramasamy und Matthias Schunter. Towards automated provisioning of secure virtualized networks. In *Proceedings of the 14th ACM conference on Computer and communications security*, Seiten 235–245. ACM, 2007.

[Col11] Eric Cole. Network Security Bible: Edition 2, 2011.

[DMT11] Kamal Dahbur, Bassil Mohammad und Ahmad Bisher Tarakji. A survey of risks, threats and vulnerabilities in cloud computing. In *Proceedings of the 2011 International conference on intelligent semantic Web-services and applications*, Seite 12. ACM, 2011.

[EWZ] S. Bhattacharjee E. W. Zegura, K. L. Calvert. How to model an internet-work.

[FBB+13] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer und Xavier Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, 15(4):1888–1906, 2013.

[FDM11] Andreas Fischer und Hermann De Meer. Position paper: Secure virtual network embedding. *PIK-Praxis der Informationsverarbeitung und Kommunikation*, 34(4):190–193, 2011.

[GCH+16] Shuiqing Gong, Jing Chen, Conghui Huang, Qingchao Zhu und Siyi Zhao. Virtual Network Embedding through Security Risk Awareness and Optimization. *KSII Transactions on Internet & Information Systems*, 10(7), 2016.

[GR05] Tal Garfinkel und Mendel Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *HotOS*, 2005.

[hi] http://www 01.ibm.com/software. CPLEX.

[Liu10] H. Liu. A new form of dos attack in a cloud and its avoidance mechanism, in: Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, 2010.

[MRR13] Raouf Boutaba Muntasir Raihan Rahman. Surviveable Virtual Network Embedding Algorithms for Network Virtualization, 2013.

[MY08] J. Rexford M. Chiang M. Yu, Y. Yi. Rethinking virtual network embedding: substrate support for path splitting and migration, 2008.

[NW] Sriram Natarajan und Tilman Wolf. Security Issues in Network Virtualization for the Future Internet.

[OCJ08] Jon Oberheide, Evan Cooke und Farnam Jahanian. Empirical exploitation of live virtual machine migration. In *Proc. of BlackHat DC convention*. Citeseer, 2008.

[QH13] X. Cao Q. Hu, Y. Wan. Resolve the virtual network embedding problem: a column generation approach. *INFOCOM, 2013 Proceedings IEEE*, 2013.

[RA93] J.B. Orlin R.K. Ahuja, R.L. Magnanti. Network Flow: Theory, Algorithms and Applications, 1993.

[RT08] Joanna Rutkowska und Alexander Tereshkin. Bluepilling the xen hypervisor. *Black Hat USA*, 2008.

[RTSS09] Thomas Ristenpart, Eran Tromer, Hovav Shacham und Stefan Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, Seiten 199–212. ACM, 2009.

[SL15] Hong Xu Ming Xu Shuhao Liu, Zhiping Cai. Towards Security-aware Virtual Network Embedding, 2015.

[WCC16] Yang Wang, Phanvu Chau und Fuyu Chen. Towards a secured network virtualization. *Computer Networks*, 104:55–65, 2016.

[WDWY10] Hanqian Wu, Yi Ding, Chuck 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. IEEE, 2010.