Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Benjamin Vetter

Sicherheit in virtualisierten Betriebssystemumgebungen

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Benjamin Vetter

Sicherheit in virtualisierten Betriebssystemumgebungen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Ing. Martin Hübner
Zweitgutachter : Prof. Dr. Ing. Gunter Klemke

Abgegeben am 30. April 2009

Autor

Benjamin Vetter

Thema der Arbeit

Sicherheit in virtualisierten Betriebssystemumgebungen

Stichworte

Sicherheit, Virtualisierung, virtualisierte Betriebssystemumgebungen, VMM, Serverkonsolidierung

Kurzzusammenfassung

Virtualisierte Betriebssystemumgebungen werden zunehmend zu einem Standard um minder- bis mittelmäßig ausgelastete Server zu konsolidieren. Das hohe Konsolidierungspotential soll Kosten einsparen und Virtualisierung soll die Flexibilität beim Umgang mit den Servern erhöhen.

In dieser Arbeit wird die Sicherheit virtualisierter Betriebssystemumgebungen anhand eines fiktiven, mittelständischen Unternehmens aus dem Finanzsektor untersucht, das eine Serverkonsolidierung plant. Hierzu werden konzeptionelle Bedrohungen der Virtualisierung und verschiedener Virtualisierungstechnologien analysiert. Die Risiken der konzeptionellen Bedrohungen werden für das Unternehmen evaluiert und es werden Maßnahmen entwickelt, die die Risiken virtualisierter Umgebungen reduzieren können.

Die Arbeit zeigt, dass die Sicherheit von Systemen durch eine Serverkonsolidierung mithilfe virtualisierter Umgebungen grundsätzlich sinkt. Das untersuchte Unternehmen muss die Flexibilität und das Konsolidierungspotential der Virtualisierung einschränken, um die Sicherheit hochschutzbedürftiger Güter auch in virtualisierten Umgebungen gewährleisten zu können.

Author

Benjamin Vetter

Title of the paper

Security within virtual Operating System environments

Keywords

Security, Virtualization, VMM, Virtual Environment, Server-Consolidation

Abstract

Virtual Operating System environments become more widely accepted as a technological base for consolidating servers to achive higher hardware utilization. Virtualization is expected to offer higher cost-efficiency and flexbility in managing multiple servers rather than conventional environments.

This paper analyzes security concerns of virtual Operating System environments by means of an imaginary, mid-size enterprise from the financial sector, which intends a server consolidation scenario. The conceptual threats of different virtualization technologies and virtualization technology in general get analyzed. The work evaluates the risks of these conceptual threats for the enterprise and develops methods to reduce the risks.

The paper shows that system security basically suffers from consolidation through virtualization technology. The considered enterprise has to lower the flexibility and consolidation potential of virtualized environments to guarantee security of their high-value systems in virtualized environments also.

Inhaltsverzeichnis

Tabellenverzeichnis

Abbildungsverzeichnis

1 Einführung

1.1 Motivation

In den vergangen Jahren hat eine über 30 Jahre alte Technologie einen starken Aufschwung erlebt. Die Virtualisierung von Betriebssystemumgebungen, die es ermöglicht, mehrere Betriebssysteme auf einer einzigen physischen Hardware zu verwenden, hat sich zu einem Hype entwickelt. Aktuell befassen sich rund 95 Prozent der Rechenzentren damit, Server-Virtualisierung einzusetzen [CE09]. Daher ist für diesen Trend auch in den kommenden Jahren noch kein Ende abzusehen. 54 Prozent derjenigen, die Virtualisierung bisher noch nicht für Serversysteme benutzt haben, rechnen in den nächsten 18 Monaten damit [IDC08b]. Damit steigt der Einsatz von Technologien zur Server-Virtualisierung explosionsartig an [IDC09]. Virtualisierung von x86-Serversystemen wird zunehmend zu einem Standard bzgl. einer breiten Palette von Anwendungen. Die Virtualisierungstechnologie befand sich 2008 auf der Top 10 Liste strategischer Technologien der Gartner Group [Gar08a] und auch 2009 ist die Virtualisierungstechnologie auf der Liste der Gartner Group [Gar09] verzeichnet, wenn auch mit Fokus auf storage- und clientseitiger Virtualisierung. Bis 2009 sollen, anhand Prognosen der Gartner Group [Gar08b], mehr als vier Millionen virtuelle Maschinen auf x86-Hardware installiert worden sein.

Antriebsmotor für diesen Prozess sind primär wirtschaftliche Interessen, die Kostenreduktion durch Konsolidierung verfolgen. Ungenutzte Kapazitäten sollen eingespart werden und die Flexibilität soll durch den Einsatz virtueller Umgebungen erhöht werden. Virtualisierung begünstigt den Trend zur Green-IT. Green-IT bezeichnet den verantwortungsvollen Umgang mit Ressourcen in der Informationstechnologie, der nicht nur förderlich für das Image eines Unternehmens ist, sondern auch die Effizienz steigern kann, indem z.B. minder- bis mittelmäßig ausgelastete Server durch Virtualisierung konsolidiert werden. Hierdurch wird höhere Effizienz der eingesetzten Energie, sowie geringerer Verbrauch für die selbe Leistung erreicht [Tho08].

Da die Zahl der Unternehmen, die Server-Virtualisierung im Produktveinsatz verwenden, kontinuierlich steigt, ist die Sicherheit und der sichere Einsatz der Technologie maßgeblich. Mit steigendem Verbreitungsgrad geraten virtualisierte Umgebungen stärker in den

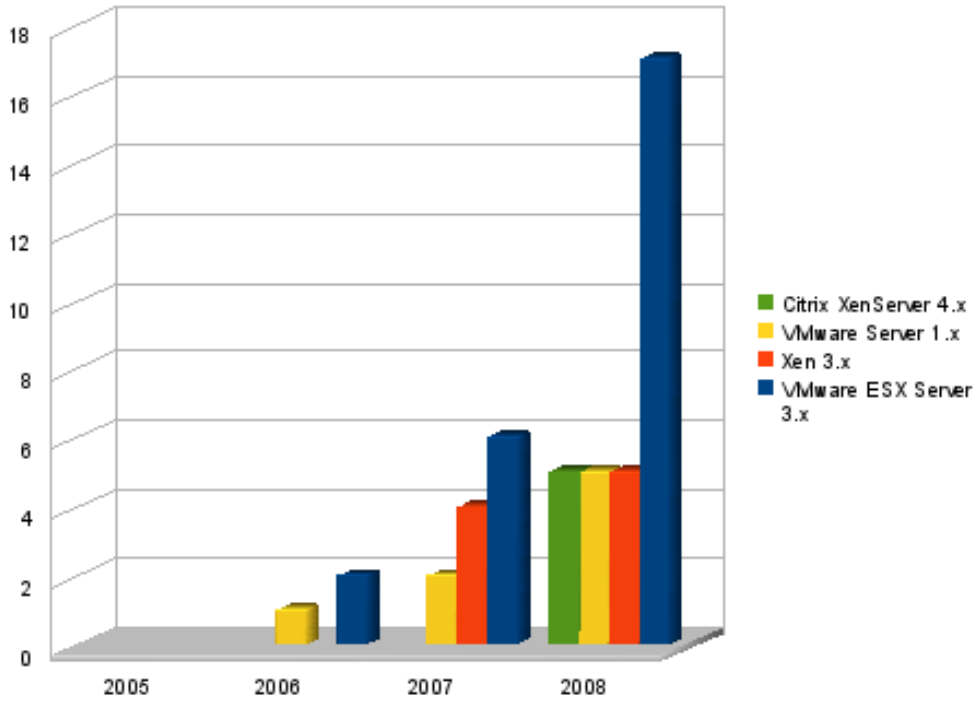

Abbildung 1.1: Secunia-Advisories für gängige Virtualisierungsplattformen

Fokus von Angreifern und Sicherheitsfachleuten, was durch Abbildung 1.1 gezeigt werden soll. Auch der Wirkungsgrad von Sicherheitslücken steigt mit zunehmendem Einsatz der Technologie. Laut S.J. Vaughan-Nichols, können virtualisierte Systeme jedoch nicht immer auf die gleiche Weise abgesichert werden wie ihre physischen Vorbilder [VN08]. IT-Verantwortliche sollten daher mit steigendem Verbreitungsgrad der Virtualisierung auch höhere Priorität auf die Sicherheit der Technologie setzen [ITS08]. Im Zuge der raschen Deployment-Anstrengungen von Unternehmen bzgl. der Server-Virtualisierung tritt die Beurteilung der Informationssicherheit der eingesetzten Virtualisierungstechnologien ggf. hinter den wirtschaftlichen Interessen zurück. Daher führt diese Arbeit eine Sicherheitsanalyse von Virtualisierungstechnologien konzeptionell für das Szenario einer Serverkonsolidierung eines fiktiven mittelständischen Unternehmens durch.

Abbildung 1.1 basiert auf [Sec09b], [Sec09c], [Sec09a] und [Sec09d]. Die Abbildung zeigt die Anzahl an Secunia Security Advisories für vier gängige Virtualisierungsprodukte[1] von 2005 bis 2008. Secunia Security Advisories werden veröffentlicht, wenn Sicherheitslücken in einem IT-Produkt identifiziert werden. Ein Anwender kann mithilfe der Empfehlungen, die in den Advisories zu finden sind, die Sicherheitslücken ggf. schließen [Sec07]. Es wird deutlich,

[1] Vgl. Kapitel 3.3.

dass vor allem das marktführende Produkt, VMware ESX Server, im Fokus der Advisories ist. Aber auch die Anzahl an Advisories für die übrigen Produkte ist angestiegen.

1.2 Problemstellung und Zielsetzung

Das Grundproblem der Arbeit ist die Frage, ob und wie sichere, virtualisierte Umgebungen betrieben werden können und wodurch sie bedroht werden. Daher wird untersucht, wodurch die Sicherheit bzw. Unsicherheit in virtualisierten Umgebungen bedingt ist.

Es existieren verschiedene Konzepte, Techniken und Paradigmen bzgl. der Virtualisierung, wie Paravirtualisierung und vollständige Virtualisierung, die für eine Serverkonsolidierung verwendet werden können. Diese Konzepte unterscheiden sich sehr stark. Daher werden die Konzepte untersucht, um zu zeigen, welche Designentscheidungen die Sicherheit beeinflussen.

Anhand eines fiktiven mittelständischen Unternehmens aus dem Finanzsektor, das über viele personenbezogene Daten verfügt, soll analysiert werden, ob Virtualisierung für den Produktiveinsatz nicht nur bei normalem, sondern auch bei hohem bis sehr hohem Schutzbedarf geeignet ist bzw. unter welchen Bedingungen. Hierzu müssen die konzeptionellen Bedrohungen für die Sicherheit von virtualisierten Betriebssystemumgebungen identifiziert werden. Hierdurch soll untersucht werden, ob konzeptionelle Schwachstellen existieren. Die in der Bedrohungsanalyse erkannten Bedrohungen sollen in einer Risikoanalyse evaluiert werden, um die Relevanz für die Praxis und die Auswirkungen für das Unternehmen zu zeigen. Das Ziel der Arbeit entspricht daher dem Ziel einer Sicherheitsanalyse (vgl. [PE06]). Das Risiko für die schützenswerten Güter des Unternehmens, das mit dem Einsatz virtualisierter Umgebungen verbunden ist, wird abgeschätzt. Die Arbeit soll hierdurch belegen, dass der Einsatz von Virtualisierungsplattformen Sicherheitsfragen aufwirft, die von Verantwortlichen und Unternehmen berücksichtigt werden müssen. Es soll gezeigt werden, dass Virtualisierung konzeptionell die Informationssicherheit steigern, aber vor allem auch verringern kann.

Die Arbeit richtet sich an Fachleute und Wissenschaflter der Informationssicherheit, Fachleute der Informationstechnologie mit Interesse für Informationssicherheit und Verantwortliche für die Durchführung eines Konsolidierungsszenarios hochschutzbedürftiger Güter mithilfe von Virtualisierungstechnologien. Die Arbeit soll zeigen, dass der Einsatz von Virtualisierung Auswirkungen auf die vorhandenen Sicherheitsmechanismen hat und der Einsatz daher sorgfältig bedacht und geplant werden muss. Daher sollen Verantwortliche für die Einführung und Aufrechterhaltung der Sicherheit bzgl. der Virtualisierung durch die Arbeit unterstützt werden, konzeptionelle, d.h. produktunabhängige Bewertungen und adäquate Maßnahmen bzgl. Virtualisierungstechnologien vornehmen zu können.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in sechs Kapitel, die aufeinander aufbauen.

Die Einführung zeigt die Motivation für eine Sicherheitsanalyse von Virtualisierungskonzepten. Eine Sicherheitsanalyse kann den Fokus auf unterschiedliche Aspekte setzen. Deshalb erfolgt in der Einführung auch eine Abgrenzung.

Kapitel zwei erläutert das Szenario, das in dieser Arbeit untersucht wird. Hierdurch wird, anhand eines fiktiven mittelständischen Unternehmens, das mit der Serverkonsolidierung eine typische Motivation für den Einsatz von Virtualisierung hat, der Bezug zur Praxis hergestellt. Es wird gezeigt, dass das Unternehmen hohe Ansprüche an die Sicherheit der eingesetzten IT-Systeme hat. Das Unternehmen möchte mit dieser Arbeit eine Sicherheitsanalyse durchführen lassen, die die Sicherheit in virtualisierten Umgebungen konzeptionell analysiert.

Die Grundlagen ermöglichen das Verständnis der Analyse. Die Grundlagen führen Begriffe ein und unterscheiden grundsätzliche Virtualisierungsarten.

Die Bedrohungsanalyse untersucht, welche Aspekte der Sicherheit von virtualisierten Umgebungen und den Virtualisierungstechnologien beeinflusst werden. Dabei werden die konzeptionellen Bedrohungen virtualisierter Umgebungen identifiziert, um die Evaluation der Bedrohungen zu ermöglichen.

Die Evaluation analysiert das Risiko der konzeptionellen Bedrohungen, die in der Bedrohungsanalyse gewonnen wurden, für das fiktive Unternehmen. Hierdurch werden die, in der Theorie gewonnenen, Erkenntnisse für die Praxis evaluiert und die Relevanz der konzeptionellen Bedrohungen für die Umsetzung einer Serverkonsolidierung gezeigt. Es wird gezeigt, welche Sicherheitsaspekte bei den unterschiedlichen Virtualisierungskonzepten in Kauf genommen werden müssen oder welche konzeptionellen Maßnahmen zu treffen sind um die Sicherheit zu erhöhen. Dem fiktiven Unternehmen werden Empfehlungen gegeben, um eine Serverkonsolidierung sicher durchführen zu können.

Die Schlussbetrachtung blickt kritisch auf die Arbeit zurück, fasst zusammen und bietet einen Ausblick auf die Zukunft des Themas.

1.4 Einordnung

Eine Sicherheitsanalyse findet im Rahmen des Sicherheitsmanagements einer Institution statt. Das Sicherheitsmanagement hat eine Planungs-, Lenkungs- und Kontrollaufgabe, die die Grundlage für eine sinnvolle Umsetzung und Erfolgskontrolle von Sicherheitsmaßnahmen ist [fSidI09].

Dem Autor sind zum Zeitpunkt der Arbeit keine umfassenden, technologieübergreifenden Sicherheitsanalysen der Konzepte der Betriebssystemvirtualisierung bekannt. Die Arbeit stützt sich allerdings auf etliche Veröffentlichungen, die Teilaspekte verschiedener Virtualisierungstechnologien unter Gesichtspunkten der Informationssicherheit analysieren.

Ende der sechziger und während der siebziger Jahre kam dem Bereich der Betriebssystem-Virtualisierung und der Sicherheit der Virtualisierung viel wissenschaftliches Interesse zu. Gerald J. Popek und Robert P. Goldberg haben die formalen Anforderungen an virtualisierbare Architekturen untersucht und Klassifikationskriterien für Virtual Machine Monitors (VMM), auch Hypervisors (HV) genannt, aufgestellt [PG73]. Stuart E. Madnick und John J. Donovan untersuchten den Ansatz, VMMs für die Isolation in Informationssystemen zu benutzen [MD73]. Sie kommen zu der Schlussfolgerung, dass VMMs die Sicherheit von Software substantiell besser gewährleisten können als herkömmliche Multitasking-Betriebssysteme (OS), was in dieser Arbeit kritisch hinterfragt wird.

Die siebziger und achtziger Jahre brachten u.a. zwei Systeme, die den Fokus auf Virtualisierung setzen und ein nachweislich hohes Maß an Sicherheit bieten, hervor. Hierzu zählen KVM/370[2] von IBM, sowie der VMM von DEC für die VAX-Architektur (vgl. [SGLS77], [KZB+90] und [KZB+91]). Der Glaube, dass der Einsatz eines VMM ein System sicherer mache, ist laut Paul A. Karger weitestgehend akzeptiert [Kar08]. Dieser Glaube hat sich ggf. dadurch gebildet, dass Virtualisierung mit den genannten sicheren Systemen assoziiert wird.

In den vergangen Jahren hat die x86-Architektur im Server-Segment enorm an Bedeutung gewonnen und laut IDC im zweiten Quartal 2008 rund 95,5 Prozent der ausgelieferten Geräte ausgemacht [IDC08a]. Der zunehmende Grad der Virtualisierung wird auch von Hardware-Herstellern berücksichtigt. Intel-VT und AMD-SVM sind zwei CPU-Technologien, die die Virtualisierung begünstigen sollen. Diese Arbeit untersucht daher, bzgl. der Sicherheit bei der Prozessor-Virtualisierung, primär die x86-Architektur und deren Nachfolger. Dabei wird untersucht, welche Auswirkungen die x86-Architektur auf die Sicherheit eines VMM hat.

1.5 Abgrenzung

Es existieren verschiedenste Arten der Virtualisierung. Diese Arbeit bezieht sich auf virtuelle Betriebssystemumgebungen, die durch einen Virtual Machine Monitor (VMM) erzeugt werden, wodurch mehrere Betriebssysteme auf einer einzigen physischen Maschine laufen können.

[2] KVM/370 von IBM ist nicht zu verwechseln mit der Kernel-based Virtual Machine (KVM), einer Virtualisierungslösung für den Linux Kernel.

In dieser Arbeit sollen Virtualisierungstechnologien konzeptionell untersucht werden. Es existeren Maßnahmen, die ggf. nach einem Sicherheitsvorfall in einem IT-System einzuleiten sind. Ob und wie diese Maßnahmen durch Virtualisierung begünstigt oder beeinträchtigt werden, ist nicht Gegenstand der Arbeit. Themen wie Backup und Recovery, Intrusion Detection, Sandboxing u.a. werden in dieser Arbeit nicht betrachtet. Konzeptionelle Sicherheit bedeutet weiterhin für diese Arbeit, dass konkrete Virtualisierungsprodukte nur als Exemplar einer bestimmten Virtualisierungstechnologie betrachtet werden. Produktspezifische Schwachstellen dienen nur exemplarisch dazu, die Relevanz der konzeptionellen Schwachstellen zu belegen.

Der Fokus der Arbeit liegt auf einer Sicherheitsanalyse mit Bezug zu einem mittelständischen Unternehmen unter dem Aspekt der Serverkonsolidierung. Daher ist es ggf. nicht immer möglich die Erkenntnisse dieser Arbeit auf größere Unternehmen und andere Szenarien zu übertragen.

Die Arbeit untersucht ausschließlich Aspekte der Informationssicherheit. Andere Aspekte beim Einsatz und der Konstruktion von Virtualisierungstechnologien, wie bspw. die Performance der Virtualisierungsarten, bleiben dabei außen vor.

2 Szenario

2.1 Kygert Micropayment GmbH

Die Kygert Micropayment GmbH (KM) ist ein fiktives, mittelständisches Unternehmen mit Hamburg als Firmensitz. Das junge Unternehmen wurde 2003 gegründet. KM wird von zwei Geschäftsführern geleitet und beschäftigt insgesamt 30 Mitarbeiter, wovon sechs Mitarbeiter ausschließlich für den Betrieb und die Instandhaltung der Informationstechnologie verantwortlich sind.

KM ist im Finanzsektor tätig, mit Micropayment als Geschäftsfeld. Micropayment bezeichnet die Bezahlung geringer Summen von 0,01 bis 5,00 Euro [Wik08]. Marc Hoeft charakterisiert Micropayment dadurch, dass Güter bezahlt werden, deren geringer Wert eine Abrechnung per Kreditkarte unrentabel macht [Hoe02]. Typische, derartige Güter sind bspw. einzelne Musikstücke oder digitale Zeitschriften. KM finanziert sich durch Gebühren, die prozentual bei einer Zahlung erhoben werden. Die Zahlungen werden über die unternehmenseigene Online-Plattform abgewickelt. KM sieht sich als optionales Zahlungsmittel für Online-Shopseiten, Auktions- und anderen Plattformen. Da die Zahlungshöhe 5,00 Euro übersteigen kann reicht das Geschäftsfeld von KM in das so genannte Macropayment hinein. KM zählt die überweisenden Endkunden, in deren Auftrag Zahlungen durchgeführt werden, zu seinen Kunden. Die Akzeptanz der Online-Plattform als Zahlungsinstrument, und damit der wirtschaftliche Erfolg des Unternehmens, wird, neben den Endkunden, durch die Partner von KM bestimmt. Zu den Partnern zählen u.a. die Online-Shop-Betreiber, da der Verbreitungsgrad maßgeblich durch diese bestimmt wird.

Endkunden registrieren ein virtuelles Konto, unter Angabe personenbezogener Daten. Die Begleichung der zu zahlenden Beträge erfolgt durch Vorausüberweisung, Lastschrift oder Kreditkartenzahlung auf das virtuelle Konto. Bei einer Zahlung wird der zu zahlende Betrag von dem persönlichen, virtuellen Konto abgebucht. Die Gebühren, die KM erhebt, werden bei der Abbuchung eines Betrages vom virtuellen Konto des Benutzers erhoben und abgebucht.

2.2 Motive für die Virtualisierung

Dieser Abschnitt soll die Motive, die KM zum Einsatz von Virtualisierung bewegt, zeigen. Eine Anforderungsanalyse oder Controlling soll hierdurch nicht vorgenommen werden.

KM ist, mit 6 Jahren, ein noch junges Unternehmen und schnell gewachsen. Daher ist die IT-Infrastruktur des Unternehmens dementsprechend jung und modern. KM hat keine nennenswerten IT-Altlasten angehäuft. Wie in einem rasch wachsenden Unternehmen üblich, steigt die Fülle notwendiger Dienste schnell an. Die Sicherheitsstrategie von KM schreibt vor, dass jeder Dienst einen eigenen Rechner erhalten soll, was i.d.R. auch von den Softwareherstellern nahegelegt wird. Daher verfügt KM über moderne Rack-Server, die die unterschiedlichen Dienste zur Verfügung stellen und die Online-Plattform bereitstellen. Die Racks sind in einem 150 Kubikmeter großen, modernen Serverraum untergebracht. Die Server sind, aufgrund der Aktualität und physischen Diensttrennung, nicht ausgelastet, was die Kosten des Unternehmens erheblich anhebt und die Effizienz der Geräte senkt.

Die steigenden Energiekosten, sowie der Kostendruck durch die Konkurrenz im Geschäftssegment, zwingt KM die laufenden Kosten zu reduzieren.

KM erwartet vom Einsatz der Virtualisierung eine Kostenreduktion durch Verminderung des Energieverbrauchs. KM möchte ein möglichst hohes Maß an Flexibilität und Konsolidierungspotential durch Virtualisierung gewinnen. Konkret bedeutet dies, dass soviele Server wie möglich, konsolidiert, also auf einem gemeinsamen Host betrieben werden sollen. Eine Konsolidierung der mittelmäßig ausgelasteten Server soll physische Server einsparen, ohne die Sicherheitsstrategie zu verletzen, da jedem Dienst ein logischer Rechner zur Verfügung gestellt werden kann. Hierdurch soll der Betrieb eines neuen Dienstes nicht länger die Anschaffung eines neuen physischen Servers bedingen. Laut Fabian Thorns, können Synergie-Effekte genutzt werden, wie bspw. eine Platzeinsparung im IT-Raum durch den Wegfall von überflüssig werdenden Servern [Tho08].

KM erhofft sich von der Virtualisierung insbesondere gesteigerte Flexibilität und zentraleres Management. Hierdurch soll es möglich werden, schneller auf veränderte Gegebenheiten, wie die Inbetriebnahme neuer Dienste, reagieren zu können und gerüstet zu sein für weiteres Wachstum des Unternehmens. Downzeiten durch bspw. Wartungsarbeiten sollen reduziert werden, da u.a. durch Live-Migration die Verfügbarkeit stets gewährleistet sein soll [VMw09].

2.3 Stellenwert der Informationssicherheit

Kreditinstitute unterliegen besonderen gesetzlichen Regelungen. In der EU-Richtlinie 2000/46/EG werden die Grundregeln für Kreditinstitute auch auf den Teilsektor der E-Geld-Institute übertragen. E-Geld wird als elektronischer Ersatz für Münzen und Banknoten definiert, das elektronisch, bspw. auf einer Chipkarte oder in einem Computer gespeichert wird und dafür gedacht ist, kleine Bezahlungen elektronisch durchzuführen [ePudRdeU00].

Laut EU-Richtlinie 91/308/EWG, sollen Kredit- und Finanzinstitute bspw. mindenstens fünf Jahre eine Kopie der Dokumente bzgl. der Identität der Kunden und Transaktionen aufbewahren, um Geldwäsche zu verhindern [deG91]. Hieraus resultiert ein hoher Schutzbedarf der zur Speicherung benutzten Systeme.

Für den elektronischen Zahlungsverkehr gelten übliche Schutzziele der IT-Sicherheit [DU04]. Hierzu zählen die

Authentizität Die Echtheit und Glaubwürdigkeit von Objekten und Subjekten ist anhand eindeutiger Identitäten und charakteristischen Eigenschaften überprüfbar [Eck08].

Vertraulichkeit Die Vertraulichkeit adressiert die Geheimhaltung von Informationen oder Ressourcen [Bis02].

Integrität Es ist keinem Subjekt möglich, Daten unautorisiert oder unbemerkt zu manipulieren [Eck08].

Verfügbarkeit Authentifizierte und autorisierte Subjekte können in der Wahrnehmung ihrer Berechtigungen nicht unautorisiert beeinträchtigt werden [Eck08].

Die Schutzziele dürfen durch den Einsatz von Virtualisierung maximal derartig beeinträchtigt werden, dass das Risiko noch tragbar ist oder wirtschaftlich vertretbare Maßnahmen ergriffen werden können, die das verbleibende Risiko auf ein tollerierbares Maß senken. Laut Marius Dannenberg und Anja Ulrich, gelten für das Macropayment, extrem hohe Sicherheitsanforderungen, da die Risiken besonders hoch einzustufen sind [DU04]. Für das Micropayment sei die IT-Sicherheit hingegen nicht das hauptsächliche Problem. Da der Übergang zwischen Micro- und Macropayment für KM allerdings nicht einheitlich unterschieden werden kann, bzw. nicht gilt, gelten die hohen Sicherheitsanforderungen für alle Transaktionen und dabei involvierten Systeme von KM.

Personenbezogene Daten sind, laut §3 des Bundesdatenschutzgesetes (BDSG) [Bun09], „Einzelangaben über persönliche oder sachliche Verhältnisse einer bestimmten oder bestimmbaren natürlichen Person (Betroffener).“ KM verfügt über sensitive, personenbezogene Daten ihrer Kunden. Hierzu gehört u.a. der Name, das Alter, der Wohnort, sowie Konto- und Finanzdaten eines Kunden. Anhand des Namens ist ein Kunde eindeutig identifizierbar bzw. der Terminologie des BDSG zufolge: bestimmt. Die Art der personenbezogenen Daten, über

die KM verfügt, ist für kriminelle Absichten von besonderem Interesse, da es sich um Konto- und Finanzdaten der Kunden handelt oder Informationen bzgl. des Kaufverhaltens der Kunden repräsentiert werden.

Zweck des BDSG ist es, laut §1 des BDSG, „... den Einzelnen davor zu schützen, dass er durch den Umgang mit seinen personenbezogenen Daten in seinem Persönlichkeitsrecht beeinträchtigt wird“. Das BDSG definiert ob oder unter welchen Umständen personenbezogene Daten erhoben, verarbeitet und übermittelt werden dürfen, um das Recht auf informationelle Selbstbestimmung zu wahren. Informationelle Selbstbestimmung ist das Recht des Einzelnen, selbst über die Preisgabe und Verwendung der eigenen, personenbezogenen Daten zu bestimmen [TEG04]. Das Gesetz gilt laut §1 des BDSG auch für nicht-öffentliche Stellen, sofern die Daten nicht ausschließlich für persönliche oder familiäre Tätigkeiten verwendet werden. Zu den nicht-öffentlichen Stellen zählen, laut §2 des BDSG, juristische Personen, Gesellschaften und andere Personenvereinigungen des privaten Rechts, also auch das fiktive Unternehmen KM.

In §9 des BDSG heisst es: „Öffentliche und nicht-öffentliche Stellen, die selbst oder im Auftrag personenbezogene Daten erheben, verarbeiten oder nutzen, haben die technischen und organisatorischen Maßnahmen zu treffen, die erforderlich sind, um die Ausführung der Vorschriften dieses Gesetzes, insbesondere die in der Anlage zu diesem Gesetz genannten Anforderungen, zu gewährleisten. Erforderlich sind Maßnahmen nur, wenn ihr Aufwand in einem angemessenen Verhältnis zu dem angestrebten Schutzzweck steht“. Hierdurch werden die Konzepte der IT-Sicherheit in den Dienst des Datenschutzes gestellt [TEG04]. Die Anlage zu §9 des BDSG nennt folgende acht zum Datenschutz zu treffende Maßnahmen:

1. Unbefugten den Zutritt zu Datenverarbeitungsanlagen, mit denen personenbezogene Daten verarbeitet oder genutzt werden, zu verwehren (Zutrittskontrolle),
2. zu verhindern, dass Datenverarbeitungssysteme von Unbefugten genutzt werden können (Zugangskontrolle),
3. zu gewährleisten, dass die zur Benutzung eines Datenverarbeitungssystems Berechtigten ausschließlich auf die ihrer Zugriffsberechtigung unterliegenden Daten zugreifen können, und dass personenbezogene Daten bei der Verarbeitung, Nutzung und nach der Speicherung nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können (Zugriffskontrolle),
4. zu gewährleisten, dass personenbezogene Daten bei der elektronischen Übertragung oder während ihres Transports oder ihrer Speicherung auf Datenträger nicht unbefugt gelesen, kopiert, verändert oder entfernt werden können, und dass überprüft und festgestellt werden kann, an welche Stellen eine Übermittlung personenbezogener Daten durch Einrichtungen zur Datenübertragung vorgesehen ist (Weitergabekontrolle),

5. zu gewährleisten, dass nachträglich überprüft und festgestellt werden kann, ob und von wem personenbezogene Daten in Datenverarbeitungssysteme eingegeben, verändert oder entfernt worden sind (Eingabekontrolle),
6. zu gewährleisten, dass personenbezogene Daten, die im Auftrag verarbeitet werden, nur entsprechend den Weisungen des Auftraggebers verarbeitet werden können (Auftragskontrolle),
7. zu gewährleisten, dass personenbezogene Daten gegen zufällige Zerstörung oder Verlust geschützt sind (Verfügbarkeitskontrolle),
8. zu gewährleisten, dass zu unterschiedlichen Zwecken erhobene Daten getrennt verarbeitet werden können.

Bei der Umsetzung der Maßnahmen muss sich jeweils am aktuellen Stand der Technik orientiert werden [TEG04]. Im Schadensfall des Betroffenen durch unrichtige oder unzulässige Datenverarbeitung gilt für nicht-öffentliche Stellen bei Vorsatz oder Fahrlässigkeit, dass ein Ersatz des materiellen und immateriellen Schadens geleistet werden muss.

Der Diebstahl von personenbezogenen Daten hätte, neben den gesetzlichen Konsequenzen, auch verheerende Konsequenzen auf die Aussenwirkung des Unternehmens. Kunden und Partner können geschädigt werden. Es gäbe einen irreparablen Vertrauensverlust seitens der Partner und Kunden, der im Finanzsektor sehr geschäftsschädigend bis ruinös ist. Sicherheitsvorfälle im Finanzsektor rücken das zughörige Institut stets in ein schlechtes Licht.

Der Schutzbedarf der personenbezogenen Daten wird von KM daher, nach Schutzbedarfskriterien des Bundesamtes für Sicherheit in der Informationstechnologie (BSI), als „sehr hoch“ eingestuft [fSidI09].

Der Begriff der Datenschutzkriminalität bezeichnet Straftaten, die einen spezifischen Bezug zu personenbezogenen Daten aufweisen. Damit richtet sich die Datenschutzkriminalität gegen das Persönlichkeitsrecht der Betroffenen, das durch das BDSG geschützt wird [TEG04]. Für die Rechtssprechung bzgl. der Datenschutzkriminalität gilt insbesondere der Bußgeldkatalog des BDSG (§43), sowie die Paragraphen 202a, 202b, 202c, 303a, 303b und 263a des Strafgesetzbuchs (StGB) [Bun08] bzgl. der Computerkriminalität.

Durch eine Manipulation des Buchungssystems von KM können Kunden geschädigt werden. Partner vertrauen auf die Integrität und Verfügbarkeit des Buchungssystems von KM für den reibungslosen Ablauf der eigenen Geschäftstätigkeit. Das Buchungssystem und die Aufbewahrung personenbezogener Daten obliegt allein dem Verantwortungsbereich und der Sorgfaltspflicht von KM. Daher zöge eine Manipulation des Buchungssystems und der Diebstahl personenbezogener Daten Schadensersatzforderungen gegenüber KM nach sich.

Der Schutzbedarf des Buchungssystems von KM wird daher ebenfalls als „sehr hoch“ eingestuft.

Informationssicherheit hat, aus den genannten Gründen, den höchsten Stellenwert für KM.

2.4 Ist-Situation

Die Sicherheitsziele des Unternehmens und der Schutzbedarf der eingesetzten IT-Systeme wurden bereits in der Vergangenheit analysiert und dokumentiert. Eine Sicherheitsstrategie ist vorhanden, in Gebrauch, wird regelmäßig überprüft und überarbeitet. Für diese Arbeit sind nur grundlegende Auszüge des, im Unternehmen vorhandenen, Sicherheitsprozesses von Interesse. Daher erfolgt hier eine Betrachtung der für die Analyse und Evaluation notwendigen Teile in entsprechend kurzer Form.

Tabelle 2.1: KM-Schutzbedarf

Schutzbedarf	Schutzziele	IT-Verbünde
sehr hoch	Vertraulichkeit, Integrität und Verfügbarkeit personenbezogener Daten	Datenbankserver
sehr hoch	Integrität und Verfügbarkeit des Buchungssystems	Applikationsserver
hoch	Integrität und Verfügbarkeit der Komponenten, mit denen Kunden in Kontakt kommen	DNS-, Web- und Mailserver
normal	Integrität und Verfügbarkeit der Standarddienste	Interne Kommunikation, VPN, Sekundäre Webserver, u.a.

Tabelle 2.1 zeigt den Schutzbedarf von IT-Verbünden, die KM betreibt, in BSI-Schutzbedarfskategorien. Sekundäre Webserver seien Server von denen die Geschäftsfähigkeit nicht in hohem Maße abhängt. Um was für einen konkreten Datenbankserver es sich handelt ist hier nicht von Belang. Der Datenbankserver sei der Aufenthaltsort der personenbezogenen Daten. Auch der konkrete Applikationsserver ist nicht von Belang. Es sei das System, das Buchungen von Geldbeträgen durchführt.

Die grundlegende Topologie des von KM betriebenen Netzwerkes ist in Abbildung 2.1 ersichtlich. Es wird ein typisches zweistufiges Firewallkonzept verwendet. Die Systeme geringeren Schutzbedarfs verfügen über ein geringeres Schutzniveau als die Systeme höheren Schutzbedarfs. IT-Verbünde verschiender Schutzbedarfskategorien werden daher in verschiedenen demilitarisierten Zonen (DMZ) lokalisiert, um das Risiko zu vermindern, dass ein Sicherheitsvorfall bzgl. den Systemen geringeren Schutzbedarfs die Wahrscheinlichkeit für

einen Sicherheitsvorfall bzgl. den System höheren Schutzbedarfs stark erhöht. Ein Eindringen in die demilitarisierten Zonen ist nur über das Internet oder das lokale Netzwerk unter Überwindung von mindestens einer Firewall möglich.

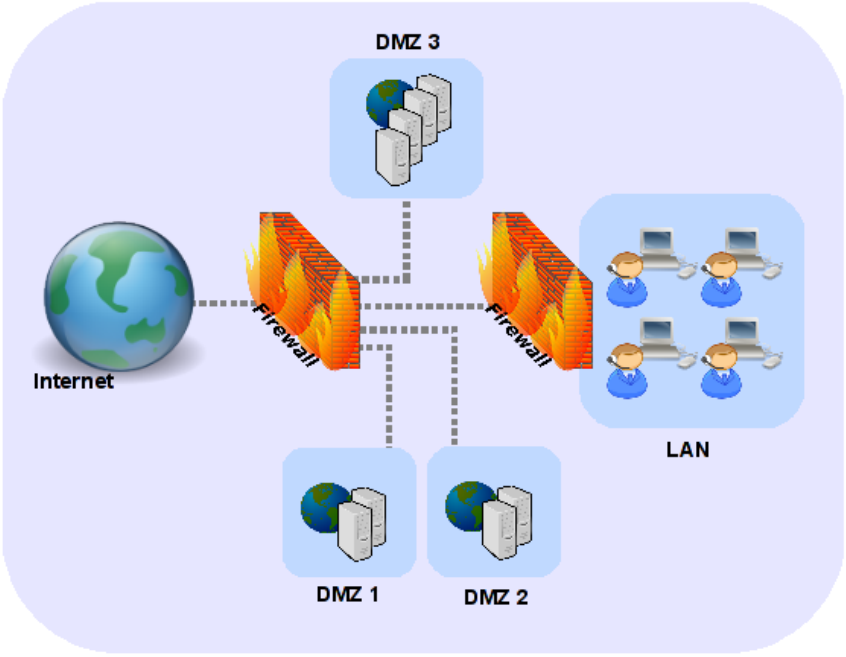

Abbildung 2.1: Topologie des KM-Netzes

Die Serverlandschaft ist heterogen. Es werden unterschiedliche Hard- und Softwarekomponenten, Netzwerklösungen und Betriebssysteme verwendet, da spezifische Softwarekomponenten i.d.R. spezifische Betriebssysteme verlangen.

Authentifikationsdaten, wie Passwörter, dürfen, wie üblich, ausschließlich verschlüsselt über das lokale Netzwerk und das Internet übertragen werden, um die Daten vor Abhörangriffen (Sniffing) zu schützen.

Jeder Netzwerkdienst wird eigens unter einem Betriebssystem betrieben. Das System bietet, soweit es möglich ist, nur diesen Dienst als Netzwerkdienst für das lokale Netzwerk und/oder das Internet an. Daher steht jedem Dienst ein eigener, physischer Rechner zur Verfügung. Die Sicherheitsstrategie sieht jedoch nur vor, dass ein eigener, logischer Rechner zur Verfügung stehen muss.

3 Grundlagen

3.1 Vorgehen

Dieses Kapitel führt grundlegende Definitionen und Begriffe ein, um das Verständnis der darauffolgenden Kapitel zu ermöglichen. Es existieren verschiedene Virtualisierungstechnologien, die in dieser Arbeit untersucht werden. Die verschiedenen Virtualisierungstechnologien münden i.d.R. in unterschiedlichen Virtualisierungsprodukten bzw. -plattformen. Daher werden die zu untersuchenden Virtualisierungstechnologien unterschieden und exemplarische Vertreter dieser Technologien, die für die praktische Umsetzung einer Serverkonsolidierung relevant sind oder in Zukunft relevant werden, kategorisiert. Hierdurch soll jedoch keine Produkt- oder Marktübersicht gegeben werden. Für weiterführende Produktdetails wird auf die entsprechenden Literaturangaben verwiesen.

3.2 Virtualisierte Betriebssystemumgebungen

Eine virtuelle Maschine (Gast) bezeichnet in dieser Arbeit eine virtuelle Umgebung, die von einem Virtual Machine Monitor (VMM) innerhalb einer physischen Umgebung erzeugt wird[1]. Die physische Umgebung entspricht der physischen Rechnerarchitektur. Eine virtuelle Umgebung gleicht einer physischen Umgebung dahingegehend, dass Betriebssysteme und deren Applikationen, die für eine physische Umgebung konzipiert wurden, ohne oder mit geringen Änderungen in der virtuellen Umgebung betrieben werden können [HWF+05]. Hierzu muss die virtuelle Umgebung der physischen Umgebung, bzgl. der Erwartungen der zu virtualisierenden Softwarekomponenten, entsprechen [AA06]. Die virtuelle Rechnerarchitektur stimmt daher mit der physischen Rechnerarchitetkur weitgehend überein. Betriebssysteme erfordern üblicherweise dedizierte Hardware. Durch Virtualisierung können mehrere Betriebssysteme auf einer einzigen, physischen Hardware betrieben werden. Der VMM betreibt jedes Betriebssystem in einer eigenen, virtuellen Umgebung, die auf die vorhandenen physischen Ressourcen abgebildet wird [RG05]. Der VMM stellt jeweils ein Interface, in Form

[1] Ein VMM kann u.U. auch selbst in einer virtualisierten Umgebung betrieben werden, was als Self-Virtualization bezeichnet wird (vgl. [KZB+90]). Self-Virtualization wird in dieser Arbeit nicht untersucht.

einer virtuellen Rechnerarchitektur, für einen Gast bereit. Zugriffe der Gäste auf das virtuelle Interface, wie bspw. der Zugriff auf virtuelle Blockgeräte, übergeben die Kontrolle an den VMM, der den Zugriff auf das Interface der physischen Blockgeräte abbildet. Das Gast-Betriebssystem wird von der physischen Rechnerarchitektur durch den VMM entkoppelt. Der VMM ist eine zusätzliche Indirektionsebene, der auf auf die physische Rechnerarchitektur zugreift und den Zugriff der Gäste auf die physische Rechnerarchitektur koordiniert. Daher ist der VMM selbst als Betriebssystem anzusehen, das sich jedoch von herkömmlichen Betriebssystemen darin unterscheidet, dass die Subjekte eines VMM virtuelle Maschinen und die Objekte virtuelle Ressourcen sind [KZB+90].

Gerald J. Popek et al. klassifizieren einen VMM als Typ-I oder Typ-II-VMM [RI00].

Typ-I-VMM Ein Typ-I-VMM wird direkt auf der Hardware betrieben und als Bare-Metal-VMM bezeichnet. Ein Typ-I-VMM ist ein Betriebssystem mit Mechanismen, die eine Virtualisierung ermöglichen. Der VMM übernimmt hierbei das Scheduling der Gäste und die Verwaltung der Ressourcen selbst.

Typ-II-VMM Ein Typ-II-VMM setzt als Applikation auf einem bestehenden Betriebssystem auf. Das Betriebssystem, das hierbei die physischen Hardwarekomponenten kontrolliert, wird als Host-Betriebssystem bezeichnet. Ein Typ-II-VMM nutzt die Mechanismen, die durch das Host-Betriebssystem bereitgestellt werden.

Emulation schafft ebenfalls virtuelle Umgebungen. Laut David Chisnall, spielt ein emulierendes System vor, ein anderes System zu sein. Ein virtualisierendes System versucht hingegen, zwei oder mehr Systeme des gleichen Typs zu sein [Chi07]. Virtualisierte Umgebungen unterscheiden sich von emulierten Umgebungen außerdem dadurch, dass eine emulierte Umgebung i.d.R. vollständig durch Software erzeugt wird. Instruktionen, die von Softwarekomponenten in einer emulierten Umgebung aufgerufen werden, werden vollständig durch den Emulator auf den physischen Prozessor abgebildet. Jede virtuelle Instruktion benötigt daher mehrere reale Instruktionen, um ausgeführt zu werden, so dass mit Leistungseinbußen zu rechnen ist. Emulation ermöglicht hierdurch, Software auch auf einer Architektur auszuführen, für die die Software nicht konzipiert wurde. Ein VMM versucht hingegen, soweit möglich, d.h. ohne die Isolation der Gäste zu gefährden, Instruktionen direkt auszuführen [PG73]. Emulation und Virtualisierung sind nicht konträr und können daher dort gemeinsam verwendet werden, wo ein VMM die virtuelle Rechnerarchitektur nicht direkt auf die physische Rechnerarchitektur abbilden kann, d.h. wo eine reine Virtualisierung nicht möglich ist [RI00].

3.3 Virtualisierungsarten

In dieser Arbeit wird die Sicherheit verschiedener Virtualisierungsarten untersucht. Die Virtualisierungsarten werden dadurch unterschieden, dass der VMM eine virtuelle Umgebung erzeugt, die der physischen Umgebung gleicht bzw. nicht gleicht und ein Gast-Betriebssystem über die virtualisierte Umgebung in Kenntnis gesetzt ist oder nicht in Kenntnis gesetzt ist.

Vollständige Virtualisierung Wenn die virtuelle Umgebung mit der physischen Umgebung bzgl. der Erwartungen der Gäste vollständig übereinstimmt, handelt es sich um eine vollständige Virtualisierung. Ein Gast-Betriebssystem kann unverändert in dieser virtuellen Umgebung ausgeführt werden, da das Gast-Betriebssystem hierbei konzeptionell auf der Architektur betrieben wird, für die es konzipiert wurde. Das Gast-Betriebssystem wird, bei einer vollständigen Virtualisierung, nicht über die Existenz eines VMM oder der virtuellen Umgebung in Kenntnis gesetzt. Typische Vertreter dieser Virtualisierungsart sind:

- VMware Server [VMw09]
- VMware ESX [VMw08]
- Microsoft Virtual Server 2005 [Arm07]

Eine CPU-unterstützte Virtualisierung erleichtert die Konstruktion eines VMM durch zusätzliche Mechanismen des Prozessors, um bspw. eine vollständige Virtualisierung von Architekturen, die nicht für den Einsatz von Virtualisierungstechnologien konzipiert wurden, zu erreichen (vgl. [UNR+05]). Indem der Instruktionssatz eines Prozessors um Instruktionen bzgl. der Virtualisierung erweitert wird, kann die Konstruktion eines VMM vereinfacht werden [Zab08]. Vertreter einer vollständigen, CPU-unterstützten Virtualisierung sind:

- Xen bzw. Citrix Xen Server [BDF+03] [Cit09]
- Microsoft Hyper-V [RB08]
- Kernel-based Virtual Machine (KVM) [Qum09b] [2]

Diese Virtualisierungsprodukte setzen Prozessoren vorraus, die eine CPU-unterstützte Virtualisierung ermöglichen. Hierzu zählen Prozessoren von Intel mit VT-Technologie [Zab08], sowie Prozessoren von AMD mit AMD-V-Technologie [AMD09]. Eine hardwareunterstützte Virtualisierung bezeichnet in dieser Arbeit allgemein eine Virtualisierung mithilfe von Mechanismen zusätzlicher Hardwarekomponenten.

Paravirtualisierung Ein Gast-Betriebssystem, dass mit einem VMM explizit zusammenarbeitet und daher über die Existenz der virtualisierten Umgebung in Kenntnis gesetzt

[2] Der Linuxdistributor Red Hat plant KVM als Basis der eigenen Virtualisierungsstrategie für Produkte der Server-Virtualisierung zu verwenden [Hat09].

ist, ist ein paravirtuelles Gast-Betriebssystem und wird bei einer Paravirtualisierung verwendet [WCSG05]. Ein Gast-Betriebssystem muss angepasst werden, um paravirtualisiert zu werden. Daher muss das Interface, das ein paravirtueller VMM den zu betreibenden Gästen bereitstellt, der physischen Rechnerarchitektur nicht entsprechen. Wenn der Sourcecode eines Betriebssystems nicht verfügbar ist und der Hersteller dieses Betriebssystems die Anpassung nicht selbst vornimmt, ist eine Paravirtualisierung dieses Betriebssystems nicht möglich, was ein Nachteil der Paravirtualisierung ist. Ein paravirtueller VMM kann ggf. keine beliebigen Betriebssysteme verwenden. Beschränkungen der originalen Rechnerarchitektur bzgl. der Virtualisierung können jedoch durch Paravirtualisierung überwunden werden. Das paravirtuelle Virtualisierungskonzept ist nicht auf das Betriebssystem beschränkt. Werden zusätzliche Komponenten, wie bspw. Gerätetreiber, die mit einem VMM bewusst zusammenarbeiten, innerhalb einer virtualisierten Umgebung betrieben, handelt es sich ebenfalls um Paravirtualisierung (vgl. [BDF+03]). Vertreter der Paravirtualisierung sind:

- Xen bzw. Citrix Xen Server[3] [BDF+03] [Cit09]
- Microsoft Hyper-V [RB08] [Cor09]

Da das Gast-Betriebssystem bei einer Paravirtualisierung angepasst werden muss, fokussiert Paravirtualisierung insbesondere quelloffene Betriebssysteme, ist jedoch nicht auf diese beschränkt [RM06].

Bei der Betrachtung typischer Vertreter bestimmter Virtualisierungsarten ist auffällig, dass gewisse Virtualisierungsplattformen, wie Xen und Microsofts Hyper-V, als Vertreter einer vollständigen Virtualisierung und einer Paravirtualisierung genannt werden. Diese Virtualisierungsprodukte verwenden mehrere Virtualisierungsarten, um die Vorteile verschiedener Virtualisierungstechnologien zu vereinen. Xen fokussiert bspw. eine Paravirtualisierung, erlaubt aber auch eine vollständige Virtualisierung von Betriebssystemen, die nicht angepasst werden können, weil der Quellcode nicht öffentlich verfügbar ist oder Lizenzrechten unterliegt, die eine Anpassung unterbinden.

3.4 Prinzipien sicherer Systeme

Um Informationssicherheit in ein System hineinzukonstruieren können architekturelle Prinzipien herangezogen werden, die zur Folge haben, dass Privilegien eingeschränkt werden und Komplexität vermindert wird, so dass Sicherheitsmechanismen mit geringerer Wahrscheinlichkeit fehlschlagen [Bis02]. Saltzer et al. haben Prinzipien sicherer Systeme, wie das Vollständigkeits- oder Erlaubnisprinzip, entwickelt, die notwendig sind, um Informationen

[3] Xen bzw. Citrix Xen Server erlaubt sowohl eine CPU-unterstützte Virtualisierung, als auch eine Paravirtualisierung.

zu schützen [SS75]. Diese Prinzipien können verwendet werden, um ein sicheres System zu entwerfen oder um zu untersuchen, inwiefern Systeme diese Prinzipien umsetzen. Daher wird in dieser Arbeit analysiert, inwiefern Prinzipien sicherer Systeme in virtualisierten Umgebungen berücksichtigt werden.

Es folgt eine Liste anerkannter Prinzipien sicherer Systeme, die auf [Bis02] basiert. Es werden ausschließlich Prinzipien erklärt, die in dieser Arbeit erwähnt werden. Für eine vollständigere Liste wird auf [Bis02] verwiesen.

Vollständigkeitsprinzip Alle Zugriffe von Subjekten auf Objekte werden ausnahmslos überprüft, um zu garantieren, dass die Zugriffe erlaubt sind.

Prinzip der minimalen Gemeinsamkeiten Mechanismen, um auf Ressourcen zuzugreifen, sollen nicht gemeinsam verwendet werden. Indem Subjekte diese Mechanismen exklusiv verwenden, werden Informationskanäle, die aus einem gemeinsamen Gebrauch von Mechanismen resultieren, unterbunden.

Erlaubnisprinzip Jeder Zugriff auf ein Objekt wird verweigert, sofern der Zugriff nicht explizit erlaubt ist. Falls ein Subjekt nicht in der Lage ist seine Aufgaben zu vollenden, soll ein System Änderungen rückgängig machen, damit das System auch im Falle eines Fehlers sicher bleibt.

Prinzip der minimalen Rechte Ein Subjekt erhält ausschließlich die Privilegien, die notwendig sind, damit das Subjekt seine Aufgaben erfüllen kann. Falls ein Subjekt zusätzliche Privilegien erhält, um eine Aufgabe erledigen zu können, soll das Subjekt diese Rechte wieder abgeben, sobald das Subjekt die Aufgabe vollendet hat.

Keep-It-Simple Sicherheitsmechanismen sollen so einfach wie möglich sein, da ein einfaches Design und eine einfache Implementation weniger Fehlerquellen besitzt. Einfachere Sicherheitsmechanismen sind daher vertrauenswürdiger als komplexere Sicherheitsmechanismen.

Um zu überprüfen, ob ein Subjekt auf ein Objekt zugreifen darf, muss ein System Dienste bereitstellen, die Zugriffsversuche ausnahmslos kontrollieren. Daher benötigt ein System Sicherheitsmechanismen und -dienste, um die Prinzipien sicherer Systeme zu realisieren. Eine Trusted Computing Base (TCB) besteht aus allen Schutzmechanismen eines Computersystems, einschließlich Hardware, Firmware und Software, die dafür verwantwortlich sind, eine Sicherheitsrichtlinie durchzusetzen [Bis02]. Die Ansprüche an die Sicherheit der TCB sind besonders hoch, da Softwarekomponenten oberhalb der TCB auf die Integrität und Verfügbarkeit der TCB vertrauen.

4 Bedrohungsanalyse

4.1 Vorgehen

Die Bedrohungsanalyse und Evaluation erfolgt in Anlehnung an bewährte Methoden der Informationssicherheit. Security Engineering befasst sich mit der Konstruktion sicherer Systeme [Eck08]. In dieser Arbeit soll kein System konstruiert werden, aber Aspekte des Security Engineerings sind für das Vorgehen der Arbeit dennoch sinnvoll und anwendbar, da in beiden Fällen Bedrohungen und deren Konsequenzen analysiert werden. Das Security Engineering ist in Phasen unterteilt. Zunächst erfolgt eine Strukturanalyse und eine Schutzbedarfsermittlung, die in dieser Arbeit jedoch nicht durchzuführen sind, da KM dies bereits durchgeführt hat und sich die bestehenden schützenswerten Güter durch den Einsatz von Virtualisierung nicht grundsätzlich ändern.

Die Bedrohungsanalyse untersucht systematisch, welche Bedrohungen der Virtualisierung die bestehenden schützenswerten Güter gefährden. Die Bedrohungsanalyse untersucht zunächst allgemeine Aspekte der Informationssicherheit die Virtualisierungstechnologie betreffend. Hierzu zählt bspw. die Frage, inwiefern die Informationssicherheit überhaupt durch Virtualisierung adressiert werden kann. Es werden organisatorische Probleme beim Umgang mit virtuellen Maschinen (Gästen) identifiziert, die sich anschließen, da es sich ebenfalls um grundsätzliche Aspekte handelt, die nicht aus unterschiedlichen Virtualisierungstechnologien resultieren. Im Anschluß werden die virtualisierten Komponenten, wie CPU und IO, eines Rechners hinsichtlich der Sicherheit untersucht um die Analyse zu verfeinern. Hierdurch wird gezeigt, wie die Sicherheitsprobleme der Virtualisierung der jeweiligen Komonenten beschaffen sind und wie verschiedene Virtualisierungstechnologien diese Probleme handhaben. Es wird untersucht, inwiefern hinzukommende Bedrohungen der Virtualisierungstechnologien die Angriffe auf schützenswerte Güter von KM ermöglichen oder beeinflussen, um die Bedrohungen evaluieren zu können.

Eine Sicherheitsstrategie, wie für das Security Engineering im weiteren üblich, wird nicht erarbeitet. Es wird jedoch gezeigt, ob Virtualisierung für das Szenario von KM grundsätzlich sicher anwendbar ist bzw. unter welchen Vorraussetzungen[1]. Hierzu werden in der Evaluation

[1] Grundsätzlich ist jede Technologie sicher einsetzbar, indem Maßnahmen getroffen werden um das Risiko

konzeptionelle Kriterien bzgl. der Sicherheit, die an eine Virtualisierungstechnologie zu stellen sind, sowie zu treffende Maßnahmen, auf Basis der Bedrohungsanalyse, entwickelt.

Da KM die Virtualisierung in ein bestehendes Sicherheitskonzept integrieren möchte, muss der Aspekt der Aufrechterhaltung der Sicherheit im laufenden Betrieb in die Bedrohungsanalyse einfließen. Daher wird innerhalb der Bedrohungsanalyse exemplarisch untersucht, ob bestehende Sicherheitsmaßnahmen in einer virtualisierten Umgebung konzeptionell noch greifen können.

4.2 Technologieübergreifende Bedrohungen

4.2.1 Grundsätzliche Sicherheit

In diesem Kapitel werden die grundsätzlichen Probleme bzgl. der Sicherheit der Virtualisierung analysiert. Hierzu wird zunächst in der Abbildung 4.1 gezeigt, wo ein VMM in einem System lokalisiert wird.

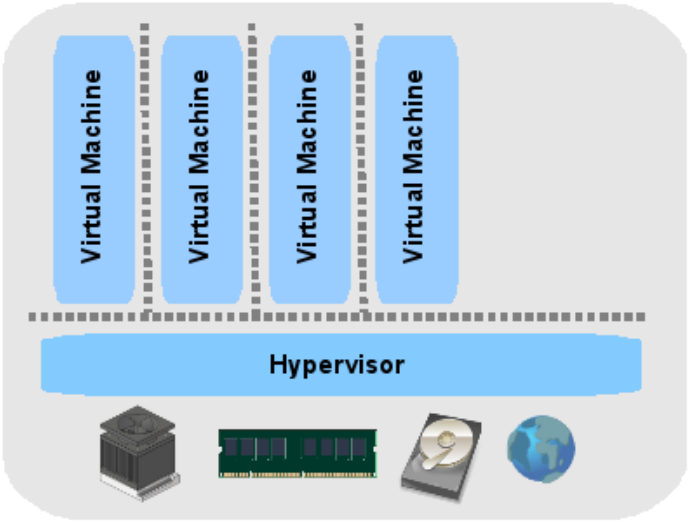

Abbildung 4.1: Aufbau eines Computersystems, das Virtualisierung verwendet

Der VMM befindet sich logisch einen Layer unter den Gästen, also zwischen Gästen und der Hardware. Die gepunkteten Linien stellen, durch den VMM konstruierte, Isolation dar. Nur an diesen Isolationslinien kann der VMM Aspekte der Sicherheit adressieren, da der VMM nur dort in die Ausführung von Instruktionen eingreift.

des Einsatzes einer Technologie auf ein tollerierbares Maß zu senken. Die Maßnahmen können jedoch ggf. sehr hohe Kosten verursachen. Der Aspekt der Kosten soll hier nicht betrachtet werden.

Die Isolation betrifft zum Einen die horizontale Isolation des VMM gegenüber dem darüberliegenden Layer, auf dem sich die Gäste befinden. Hierdurch wird der VMM vor den Gästen geschützt und ein direkter Hardwarezugriff der Gäste, denen der VMM nicht vertrauen kann, unterbunden. Zum Anderen wird eine vertikale Isolation der jeweiligen Gäste untereinander erreicht, die sich gegenseitig nicht vertrauen können, da eine virtuelle Maschine (Gast) ggf. von einem nicht autorisierten Subjekt kompromittiert wurde oder betrieben wird. Subjekte, die in einem Gast tätig sind, sollen Subjekte eines anderen Gasts, sofern nicht explizit gewünscht, nicht wahrnehmen können. Die vertikale Isolation der verschiedenen Gäste voneinander soll daher dazu führen, dass jeder Gast unbeeinflusst von anderen Gästen tätig seien kann.

Physische Isolation schafft Isolation i.d.R. durch räumliche Distanz. Dies betrifft den Betrieb mit dedizierter Hardware. Logische Isolation nutzt die Gesetze der Logik um Isolation zu gewährleisten und zu verifizieren. Hierbei sichern Regeln die Isolation, die durch eine Kombination aus Soft- und Hardware implementiert werden. Ein Verlust von logischer Isolation bzgl. der Virtualisierung hätte zur Folge, dass zwei Subjekte unerwünschten Zugriff auf ein Objekt erhalten, den sie ohne den Einsatz von Virtualisierung nicht hätten. Daher führt ein Verlust von Isolation zu Bedrohungen bzgl. der Informationssicherheit.

Hohe Isolation ist förderlich für die Sicherheit eines Systems, da Isolation das Prinzip der minimalen Gemeinsamkeiten, das Erlaubnisprinzip und das Prinzip der minimalen Rechte umsetzt. Das Prinzip der minimalen Gemeinsamkeiten unterbindet die enstehende Komplexität, die vom gemeinsamen Gebrauch von Komponenten herrührt. Durch den gemeinsamen Gebrauch von Komponenten kann es ggf. zu unerwünschten Informationsflüssen zwischen Subjekten kommen, die nicht exiplizit dazu berechtigt sind Daten auszutauschen. Die Sicherheit wird beeinträchtigt, da ggf. sensitive Informationen gewonnen werden können. Das Erlaubnisprinzip wird dahingehend umgesetzt, dass eine Kommunikation der isolierten Bestandteile eines Systems zunächst vollständig ausgeschlossen wird und erst durch explizite Erlaubnis gestattet wird. Das Prinzip der minimalen Rechte adressiert die Privilegien, die eine Systemkomponente zur Erfüllung seiner Aufgaben benötigt. Diese Privilegien sollen stets minimiert werden, um ein hohes Maß an Sicherheit zu gewährleisten. Im Falle einer total isolierten Komponente können keine Informationen gewonnen werden und die Komponente kann nicht mit anderen Komponenten kommunizieren [Bis02], was zwar der Sicherheit zuträglich ist, aber i.d.R. nicht praktikabel ist.

[SPF+07] Stephen Soltez et al. unterscheiden folgende Isolationsarten:

Fault Isolation Fehler innerhalb eines Gasts beeinflussen nicht den Status und die korrekte Ausführung in anderen Gästen.

Resource Isolation Der Gebrauch von Ressourcen ist fair unter den Gästen aufgeteilt bzw. innerhalb der Vorgaben des Accountings. Zur Resource Isolation gehört auch die Per-

formance Isolation, die sicher stellen soll, dass die Ausführung eines Gasts nicht die Ausführung anderer Gäste übermäßig stark beeinträchtigt.

Security Isolation Das Ausmaß, mit dem virtuelle Systeme den Zugriff auf logische Objekte limitieren. Eine Kompromittierung eines Gasts darf nicht die Wahrscheinlichkeit der Kompromittierung anderer Gäste begünstigen.

Da sich die Gäste, die oberhalb eines VMM betrieben werden, untereinander nicht vertrauen, ist eine Isolation der Gäste voneinander besonders wichtig [SJV+05].

Ohne Virtualisierung entspricht ein virtueller Rechner einem physischen Rechner. Die angesprochene vertikale und horizontale Isolation ist in einer nicht-virtualisierten Umgebung natürlich ebenfalls gewährleistet, da nur ein einziges Betriebssystem auf einem Host betrieben wird. Die logische Isolation wird automatisch durchgesetzt, da ein Sharing der Hardwarekomponenten zwischen mehreren Gast-Betriebssystemen nicht notwendig ist und auch nicht angestrebt wird. Zusätzlich wird eine physische Isolation erreicht, die bei der Virtualisierung i.d.R. nicht erzielt wird. Der Einsatz von Virtualisierung führt daher zu einem Verlust von physischer Isolation und einer Zunahme von Sharing, da verschiedene Rechner die gleiche Hardware benutzen.

Übliche Betriebssysteme müssen eine Reihe schwieriger Aufgaben bewältigen. Sie kontrollieren unterschiedlichste Hardwarekomponenten, verwalten Dateistrukturen, isolieren User-Space Applikationen voneinander und müssen unterschiedlichste Sicherheitsrichtlinien um- und durchsetzen. Laut Stuart E. Madnick et al. sind Betriebssysteme daher üblicherweise sehr komplex und fehleranfällig [MD73]. Laut Tal Garfinkel et al. isolieren herkömmliche Betriebssysteme Applikationen nicht ausreichend voneinander, so dass eine kompromittierte Applikation häufig dazu führt, dass das komplette System kompromittiert werden kann [GPC+03]. Ein Betriebssystem stellt Abstraktionsmechanismen auf hoher Ebene zur Verfügung, um den Zugriff auf hardwarenähere Ebenen zu ermöglichen. Schutzmechanismen werden dabei auf der gleichen, hohen Ebene implementiert. Laut Andrew Whitakter et al. bietet dieses Design Angriffsfläche für Layer-below Angriffe [WSG02]. Ein Angreifer umgeht die Schutzmechanismen auf hoher Ebene, indem er die untere Ebene direkt angreift.

Die Applikationen, die oberhalb eines Betriebssystems lokalisiert werden, vertrauen auf die Integrität des Betriebssystems und aller sicherheitsrelevanten Komponenten des Betriebssystems. Das Betriebssystem ist Teil der Trusted Computing Base (TCB) einer User-Space Applikation. Kleinere TCB gelten unter Sicherheitsaspekten als besser, da eine kleine TCB einfacher zu testen, zu analysieren und auf Vollständigkeit zu verifizieren ist [Bis02].

Betriebssysteme verstoßen, durch die hohe Komplexität und der großen TCB, gegen das Prinzip des Keep-It-Simple bzw. Simplicity Promotes Security. Subjekte, die kaum mit anderen Subjekten interagieren müssen, benötigen i.d.R. keine komplexen Sicherheitsrichtlinien, um Objekte gemeinsam sicher nutzen zu können [Bel06]. Diese Subjekte erfordern i.d.R.

als einzige Policy hohe Isolation, so dass komplexe Schutzmechanismen des Betriebssystems für diese Dienste nicht angemessen sind. Da VMMs klein und simpel konzipiert werden können, gelten VMMs als bessere Alternative für die Sicherheit und Isolation in Informationssystemen. VMMs werden spezifisch konzipiert um Isolation durchzusetzen. Die Funktionssicherheit eines VMM ist sehr hoch, da ein VMM ein sehr kleines Programm ist [HUL06]. VMMs fügen der Hardware lediglich Isolation, Time-Sharing und vertrauenswürdige Treiber hinzu [HPHS04]. Hierdurch sind VMMs weniger anfällig für Fehler, da sich die Fehler eines Systems in Relation zum Umfang des Codes verhalten [MMH08]. Desweiteren ist die Sicherheit von VMMs einfacher formal verifizierbar, was mit zunehmender Komplexität nicht mehr länger der Fall ist [HPHS04]. Ein VMM kann daher Isolation konzeptionell besser als ein übliches Betriebssystem gewährleisten und ist besser dafür geeignet, Mechanismen bereitzustellen um eine Sicherheitsrichtlinie durchzusetzen.

Unter der Annahme, dass ein VMM Isolation besser gewährleisten kann als ein herkömmliches Betriebssystem, lässt sich die Sicherheit eines Systems durch den Einsatz von Virtualisierung konzeptionell dadurch erhöhen, dass Subjekte, die gemeinsam innerhalb eines Betriebssystems betrieben werden, durch den Einsatz von Virtualisierung jeweils in einen eigenen Gast relokalisiert werden und damit ein eigenes Betriebssystem erhalten. Hierdurch sind diese Subjekte stärker als zuvor voneinander isoliert, wodurch Wechselwirkungen zwischen diesen Subjekten und Komplexitätserscheinungen bzgl. der Konfiguration und dem Betrieb dieser Subjekte unterbunden werden.

Die These, dass Virtualisierung Isolation besser gewährleisten kann als ein herkömmliches Betriebssystem basiert auf dem Vergleich eines herkömmlichen Betriebssystems und dem Einsatz eines VMM. Der VMM ist jedoch in der Praxis ein zusätzlicher Abstraktions-Layer. Für den Betrieb eines Gasts ist, neben dem VMM, zusätzlich ein übliches Betriebssystem in jedem Gast notwendig, so dass die TCB einer User-Space Applikation in Wirklichkeit größer wird, da der VMM zusätzlich in die TCB jedes Gasts mitaufgenommen werden muss. Die herkömmlichen Angriffe auf Sicherheitslücken von Betriebssystemen kommen auch in einer virtualisierten Umgebung zum Tragen [VN08].

Da der VMM direkt über der Hardware lokalisiert ist, erfüllt ein VMM das Vollständigkeitsprinzip, engl. Complete Mediation. Alle Instruktionen müssen ausnahmslos den VMM-Layer durchqueren, um auf die Hardware zugreifen zu können. Das Ziel eines VMM ist es jedoch, möglichst viele Instruktionen direkt, d.h. ohne Intervention, auszuführen. Laut Gerald J. Popek et al., wird eine statistisch dominierende Teilmenge des Instruktionssatzes eines Prozessors direkt ausgeführt [PG73]. Daher beschränkt sich der VMM i.d.R. auf die Rolle des Vermittlers zwischen den jeweiligen Gästen und der Hardware, d.h. der VMM ist vorrangig dafür zuständig die Hardware unter den Gästen zu partitionieren und zu sharen. Werden keine expliziten, zusätzlichen Sicherheitsmechanismen im VMM-Layer implementiert, die die Sicherheit der Layer über dem VMM adressieren, bleiben alle Sicherheitsprobleme und -eigenschaften der höheren Layer erhalten. Die Sicherheit eines Gasts kann maximal

die Sicherheit erreichen, die auch ohne einen VMM erreicht werden kann, wenn ein Gast von dedizierter Hardware identisch in eine virtualisierte Umgebung überführt wird und der VMM keine zusätzlichen Sicherheitsmechanismen bereitstellt. Der bloße Einsatz von Virtualisierung erhöht daher in diesem Fall die Sicherheit konzeptionell grundsätzlich nicht, da die logische Isolation im Vergleich mit dedizierter Hardware nicht erhöht wird. Eine Konsolidierung von Servern vermindert in diesem Fall tatsächlich die Sicherheit, da die physische Isolation nicht mehr gegeben ist und eine erhöhte logische Isolation nicht erreicht wird. Im Falle der Serverkonsolidierung wird der Verlust der physischen Isolation sogar explizit gewünscht. Der VMM agiert in diesem Fall nur als zusätzlicher Software-Layer, der das Risiko eines Sicherheitsvorfalls erhöht. Es geht im Falle der Serverkonsolidierung daher grundsätzlich nicht darum, dass die Sicherheit durch den Einsatz eines VMM erhöht wird. Vielmehr geht es darum, dass der VMM und die virtualisierte Umgebung möglichst nah an die Sicherheit, die durch dedizierte Hardware erreicht wird, herankommt.

Das Confinement-Problem ist das Problem, zu verhindern, dass ein Service-Anbieter Informationen preisgibt, die ein Service-Nutzer als vertraulich einstuft [Bis02]. In der Praxis bedeutet das Confinement-Problem stets einen Kompromiss zwischen Sharing, der gemeinsamen, koordinierten Verwendung von Ressourcen und der Isolation. Der Einsatz von Virtualisierungstechnologien zur Serverkonsolidierung hat zur Folge, dass Subjekte (Gäste), die isoliert voneinander operieren sollen, in noch stärkerem Ausmaß gemeinsame Ressourcen verwenden. Sharing erleichtert erfolgreiche Angriffe auf schützenswerte Güter [Eck08]. Über gemeinsam verwendete Ressourcen können Prozesse, die nicht ausreichend voneinander isoliert sind, Informationen austauschen. Informationskanäle, die nicht für die Kommunikation konzipiert wurden, aber als solche benutzt werden können, werden als Covert Channels bezeichnet. Je höher die Bandbreite eines Covert Channels ist, desto größere Gefahr geht von ihm aus. Es ist möglich, Covert Channels mit formalen Methoden aufzuspüren [1], was allerdings i.d.R. mit viel Aufwand verbunden ist. Ihre Existenz kann niemals vollständig ausgeschlossen und verhindert werden, aber die Bandbreite kann durch methodische Systemkonstruktion beschränkt werden [Eck08]. Isolation wirkt der Existenz und Nutzung von Covert Channels entgegen. Sharing und Isolation sind daher konträr. Ein höheres Maß an Sharing hat einen Verlust von Isolation zur Folge. Da der Einsatz von Virtualisierung i.d.R. ein hohes Maß an Sharing bedeuetet, ist ein virtualisiertes System besonders anfällig für Covert Channels.

Je mehr Gäste auf einem Host betrieben werden, desto größer wird das grundsätzliche Risiko eines Sicherheitsvorfalls für diesen Host. Der VMM ist ein Single Point Of Failure. Das Schadensausmaß steigt, was am Beispiel des technischen Versagens leicht zu erkennen ist. Fällt der Host-Rechner aufgrund technischen Versagens aus, fallen sämtliche Gäste aus, die auf diesem Host betrieben werden. Werden die gleichen Systeme jeweils auf de-

[1] Hier seien die Shared Resource Matrix Methode, Covert Flow Trees und die Informationsflussanalyse genannt [Bis02].

dizierter Hardware betrieben, ist ein gleichzeitiges, technisches Versagen aller Systeme unwahrscheinlicher. Das Schadensausmaß eines technischen Defekts steigt. Die Eintrittswahrscheinlichkeit für ein technisches Versagen steigt ebenfalls, da die Hardwarekomponenten stärker beansprucht werden und Hardwarekomponenten üblicherweise Abnutzungserscheinungen zeigen. K. Bellam et al. identifizieren die Beanspruchung von Festplatten als Faktor für deren Zuverlässigkeit [BMR+08]. Eduardo Pinhero et al. zeigen, dass dieses Verhalten vorwiegend bei jungen[1] und älteren[2] Festplatten festzustellen ist [PWB07]. Das Risiko wird nicht nur durch technisches Versagen gesteigert, sondern auch durch Vorfälle wie die Infektion durch Viren oder die Kompromittierung eines Gasts oder mehrerer Gäste, da mit steigender Anzahl von Gästen auch die Eintrittswahrscheinlichkeit für diese Vorfälle bzgl. des Hosts steigt.

Ein Pure-Isolation-VMM teilt die verfügbare Hardware exklusiv unter den Gästen auf und unterbindet jegliches Sharing von Ressourcen [Kar05]. Bei einem Pure-Isolation-VMM findet daher kein direktes Sharing von Hardwarekomponenten statt. Ein Pure-Isolation-VMM kann ein ähnlich hohes Maß an Sicherheit bieten, wie der Einsatz mehrerer Rechner und gehört zur sichersten Klasse von VMMs [KS08]. Andere Typen von VMMs verfolgen das Ziel, eine Hardwarekomponente mehreren Gästen zur Verfügung zu stellen, was das Sharing erhöht und die Isolation vermindert und somit die Sicherheit beeinträchtigt.

Durch den Einsatz von Virtualisierung werden vorhandene, physische Barrieren entfernt. Die Kommunikation zwischen zwei Gästen oder einem Gast und dem VMM ist, sofern auf dem gleichen Host lokalisiert, ggf. nicht den physischen Beschränkungen der Netzwerkkomponenten ausgesetzt. Damit unterliegt die Kommunikation der Geschwindigkeit des Hauptspeichers. Daher wird auch die Geschwindigkeit bestimmter Angriffe erhöht und bspw. der Befall durch Schadsoftware (Malware) kann sich schneller innerhalb eines Hosts ausbreiten.

Eine virtualisierte Umgebung ist eine logische Umgebung. Fehler in der Logik oder eine Zerstörung der Mechanismen, die die Isolation zwischen VMM und den Gästen aufrecht erhalten sollen, ermöglicht, dass die Isolation von VMM und den Gästen aufgehoben wird. Ein unprivilegierter, unvertrauenswürdiger Gast kann in diesem Fall Code mit den Privilegien des VMM ausführen und erhält ggf. Zugriff auf Objekte, die dem VMM vorbehalten sind. Hohe Komplexität der Logik, um die Isolation sicherzustellen, erhöht die Wahrscheinlichkeit für derartige Fehler. Man spricht in diesem Fall von einem Ausbruch, engl. Escape, aus der virtualisierten Umgebung [Reu07]. Der Virtualisierungs-Layer und dessen Mechanismen zur Isolation werden hierbei von einem Gast umgangen, so dass der Gast Zugriff auf den physischen Rechner und damit auf andere Gäste des gleichen Hosts erhält. Hierdurch sind der Host und alle Gäste des Hosts als kompromittiert anzusehen.

[1] Festplatten, die weniger als ca. ein Jahr alt sind.
[2] Festplatten, die älter als ca. fünf Jahre alt sind.

In diesem Kapitel wurde gezeigt, dass Virtualisierung die physische Isolation beseitigt. Virtualisierung kann zwar die logische Isolation konzeptionell erhöhen, was aber abhängig vom Einsatzzweck ist und im Falle einer reinen Serverkonsolidierung grundsätzlich nicht eintritt. Die Probleme der nicht vorhandenen physischen Isolation kommen daher bei einer Serverkonsolidierung voll zum Tragen. Die Sicherheit der Systeme von KM wird also grundsätzlich verringert. Die weiter zu analysierende Frage ist daher, ob bzw. unter welchen Bedingungen die verbleibende logische Isolation von Virtualisierungstechnologien ausreicht, um dennoch ein derart hohes Maß an Sicherheit zu erzielen, das für die Komponenten mit hohem und sehr hohem Schutzbedarf von KM notwendig ist.

4.2.2 Organisatorische Probleme

Sicherheitsaspekte der Virtualisierung haben nicht nur mit der jeweiligen Technologie zu tun. Virtualisierung verändert fundamental die IT-Architektur, Handlungsmuster, das Deployment und Management der Server [VN08]. Diese organisatorischen Probleme entstehen primär durch die Dynamik, die die Virtualisierungstechnologie ermöglicht. Die Dynamik ist konträr zu Sicherheitsarchitekturen, die statische Systeme verfolgen. Daher hat die, durch Virtualisierung angestrebte, Erhöhung der Flexibilität auch negative Auswirkungen.

[GR05] Tal Garfinkel et al. nennen folgende, organisatorische Probleme:

Scaling Da Virtualisierung die Inbetriebnahme ganzer, logischer Rechner um ein Vielfaches vereinfacht, ist damit zu rechnen, dass wesentlich mehr Rechner betrieben werden. Hierdurch werden organisatorische Aufgaben, wie das Patch-Management und die Konfiguration der Rechner, die manuelles Eingreifen erfordern, erschwert und multipliziert. Das Ausmaß von Attacken kann daher um ein Vielfaches vergrößert werden, da ggf. die gleichen Sicherheitslücken in vielen Gästen vorhanden sind.

Transience Der Einsatz von Virtualisierung kann dazu führen, dass viele Rechner vorrübergehend im Netzwerk auftauchen und wieder verschwinden. Die Vergänglichkeit von Teilnehmern des Netzwerkes steigt demzufolge. Durch automatisierte Angriffe, wie Würmer und Viren, können in kurzer Zeit viele Rechner infiziert werden. Eine Bereinigung des entstandenen Schadens ist nunmehr kaum vollständig durchführbar, da u.U. Rechner vorrübergehend nicht mehr im Netzwerk vorhanden sind und nicht zentral erfasst wurden. Werden die infizierten Rechner, die bei der Bereinigung nicht berücksichtigt wurden, wieder erneut innerhalb des Netzwerks betrieben, ist eine erneute Infektion nicht auszuschließen. Hierdurch erreicht das gesamte System ggf. nur unter hohem Aufwand und hohen Kosten wieder einen wünschenswerten, definierten Zustand.

Software Lifecycle Gäste können üblicherweise durch einen Rollback-Mechanismus in einen vergangenen Zustand zurückgeführt werden. Hierzu wird zu einem beliebigen Zeitpunkt ein Schnapschuss des gesamten Gasts erzeugt. Zu diesem Zustand kann zu einem späteren Zeitpunkt zurückgekehrt werden. Dieses Vorgehen steht im Widerspruch zu der linearen Lebenslinie eines üblichen, nicht-virtualisierten Rechners, von der Applikationen ausgehen. Bereits gepatchte Sicherheitslücken werden ggf. wieder geöffnet, gesperrte Accounts werden wieder verfügbar oder zurückgezogene kryptographische Schlüssel sind wieder gültig. Auf derartigen Systemen neu generierte Zufallszahlen und Session-Keys wurden ggf. bereits in der Vergangenheit generiert und benutzt, ohne das dieser Missstand zu Tage tritt.

Diversity Der Einsatz von Virtualisierung kann dazu führen, dass vermehrt, verschiedenste OS-Versionen, gepatcht oder ungepatcht betrieben werden. Der Grad an Heterogenität steigt, was das Patch-Management erschwert, da viele Gäste unterschiedliche Update-Zyklen benötigen. Eine vollständige Automatisierung des Patch-Managements ist bis heute nicht möglich. Durch den Einsatz von Virtualisierung wird dieses Problem verschärft.

Mobility Die Mobilität ist beim Umgang mit virtuellen Gästen wesentlich größer. Gäste können innerhalb kurzer Zeit auf andere Host-Rechner live-migriert oder manuell, samt Dateisystem, kopiert werden. Die TCB eines üblichen Rechners besteht aus einem Software- und einem Hardware-Stack. Im Gegensatz dazu besteht die TCB eines Gasts aus allen Hosts, auf denen der Gast jemals betrieben wurde, was ggf. zu langen, undokumentierten Vertrauensbeziehungen führt. Eine Kompromittierung oder Virus-Infektion örtlich einzuschränken und physisch zu isolieren wird hierdurch erschwert. Bei einer Live-Migration wird i.d.R. der Inhalt des Hauptspeichers eines Gasts über das Netzwerk zu einem anderen VMM übermittelt, der daraufhin den Gast in dem gleichen Zustand starten kann. Im Hauptspeicher legen Gäste jedoch sensitive Daten ab, die bei einem Transfer über ein unsicheres Netzwerk abgehört werden können.

Identity Die Identität eines Gasts ist, durch die Vergänglichkeit und Mobilität eines Gasts, schwieriger zu bestimmen. Eine, für einen Gast, verantwortliche Person ist ebenfalls schwieriger zu identifizieren. Eine ad-hoc assozierte Identität ist bei Gästen nicht vorhanden, da ein VMM i.d.R. zufällig MAC-Adressen vergibt und Gäste in verschiedenen Zuständen im Netzwerk erscheinen und verschwinden. Dieser Zustand beeinträchtigt die Sicherheit, da die Ursache von Sicherheitsvorfällen, schwieriger zu ermitteln ist.

Data Lifetime Virtualisierung fügt sich als zusätzliche Indirektionsebene in ein System ein. Betriebssysteme gehen i.d.R. davon aus, die Hardware exklusiv zu kontrollieren. Daher gehen Betriebssysteme weiterhin von einem direkten Pfad zur Hardware aus, dem Aufenthaltsort der verwendeten Daten. Der VMM-Layer wird i.d.R. bei der Softwareentwicklung nicht als zusätzlicher Layer berücksichtigt. Durch den Einsatz von Virtualisierung wird der Aufenthaltsort und die Aufenthaltsdauer sensitiver Informationen

intransparent für die beteiligten Komponenten. Dieser Umstand widerspricht dem Prinzip, dass sich sensitive Informationen stets nur so kurz wie möglich im System aufhalten sollen. Passwörter oder kryptographische Schlüssel können ggf. in Caches des VMM über längere Zeit, als zwingend erforderlich, verweilen.

In einem Unternehmensnetzwerk existieren Komponenten mit unterschiedlichem Schutzbedarf. IT-Systeme besitzen unterschiedliches Schutzniveau und haben unterschiedliche Ansprüche an die Sicherheit. Durch Virtualisierung werden ggf. Systeme mit unterschiedlichem Schutzniveau auf dem gleichen Host, d.h. oberhalb des gleichen VMM, betrieben. Nicht jeder verfügbare VMM, der den Betrieb von Gästen ermöglicht, verfügt auch über das notwendige Schutzniveau, das von Gästen mit unterschiedlichem Schutzbedarf benötigt wird [VN08]. Der Schutzbedarf des VMM ist mindestens so hoch, wie der höchste Schutzbedarf der Gäste. Heutige VMMs bieten i.d.R. keine Möglichkeiten dem unterschiedlichen Schutzbedarf der Gäste gerecht zu werden[1].

Der Administrator eines VMM hat grundsätzlich uneingeschränkten Zugriff auf jeden Gast, der oberhalb des VMM betrieben wird, da der Administrator uneingeschränkten Zugriff auf alle Komponenten hat, die von einem Gast benutzt werden. Daher dürfen nur Personen als Administratoren eines Host-Rechners eingesetzt werden, die auch als Administratoren der Gäste anzusehen sind, was als schwierig einzustufen ist. Durch die gesteigerte Mobilität von Gästen, kann sich dies mit zunehmender Zahl betriebener Gäste, weiter zu einem Problem entwickeln, da die Komplexität der Reglementierungen schnell steigt. In den Gästen können sich sensitive Daten befinden, die allen Administratoren des VMM zugänglich sind. Um auch bei hoher Mobilität von Gästen Schutzziele für die Gäste aufrecht erhalten zu können, müsste bspw. global eine Policy bzgl. eines Gasts definiert werden können, die von Mechanismen des jeweiligen VMM unüberwindbar durchgesetzt wird. Hierdurch muss nicht darauf vertraut werden, dass die organisatorischen Richtlinien von Administratoren eingehalten werden, da die Richtlinie vom VMM durchgesetzt wird. Bzgl. der Vertrauenswürdigkeit des Administrators eines VMM ist auch eine Kompromittierung eines VMM zu nennen. Ein Angreifer, dem es gelingt den VMM zu kompromittieren, hat uneingeschränkten Zugriff auf alle Gäste.

4.3 Bedrohungen der Ressourcen-Virtualisierung

Die zwei letzten Kapitel haben die grundsätzlichen Sicherheitsaspekte und Bedrohungen der Virtualisierung analysiert. In diesem Kapitel wird die Bedrohungsanalyse verfeinert, indem die verschiendenen Virtualisierungstechnologien analysiert werden.

Um eine virtuelle Umgebung zu schaffen müssen die üblichen Komponenten bzw. Ressourcen eines Rechners virtualisiert werden. Hierzu zählen die CPU, die Speicherverwaltung

[1] Als Ausnahme sei hier sHype für die Xen-Plattform genannt [SJV+05]

und das IO-Subsystem. Zusätzlich wird die Virtualisierung von Netzwerkkomponenten analysiert, die den Gästen eine Netzwerkanbindung ermöglicht. Die CPU, der Arbeitsspeicher, der Massenspeicher und das Netzwerk-Interface werden auch als die „Core Four“ bezeichnet [Tho08]. Die Komponenten müssen geschützt, der vorhanden Anzahl an Gästen verfügbar gemacht werden. Unterschiedliche Virtualisierungstechnologien versuchen dieses Problem auf verschiedene Weise zu lösen. Daher werden die konzeptionellen Lösungsansätze der verschiedenen Technologien bzgl. der Sicherheit untersucht.

Der Fokus liegt bei der Analyse der zu virtualisierenden Komponenten auf den Hürden bzgl. der Sicherheit, die von Virtualisierungstechnologien zu überwinden sind. Es wird untersucht, welche Hardwaremechanismen sich vorteil- bzw. nachteilhaft auf die Sicherheit bzgl. der Virtualisierung auswirken und welche zusätzlichen Bedrohungen für die Sicherheit der virtuellen Umgebung hierdurch entstehen.

Der VMM muss die logische Isolation gewährleisten, um ein hohes Maß an Sicherheit zu erzielen, da Virtualisierung die physische Isolation beseitigt. Die logische Isolation eines VMM äußert sich im isolierten Zugriff auf die zu virtualisierenden Komponenten. Die Analyse der zu virtualisierenden Komponenten soll daher zeigen, ob die logische Isolation durch den Einsatz von Virtualisierung unter Verwendung der x86-Architektur ausreicht oder welche Aspekte der Architektur dies beeinträchtigen.

4.3.1 CPU-Virtualisierung

Das Grundproblem, die Ressource CPU mehreren, nicht-vertrauenswürdigen Nutzern zur Verfügung zu stellen, ist bei der Konstruktion von üblichen Betriebssystemen und VMMs gleich. Die privilegierten Instruktionen eines Prozessors dürfen nicht direkt von Layern aufgerufen werden, die oberhalb des Betriebssystem-Kernels bzw. VMM lokalisiert sind. Diese Layer sind nicht vertrauenswürdig. Die privilegierten Instruktionen greifen direkt auf die Hardware zu und manipulieren die Hardware. Privilegierte Instruktionen verändern Einträge in sensitiven Registern, greifen direkt auf den physischen Arbeitsspeicher oder IO-Geräte zu [BDR97]. Durch den Zugriff auf privilegierte Instruktionen des Prozessors erhält die aufrufende Instanz Kontrolle über die Hardware und damit über das System, da eine Intervention des VMM nicht möglich ist. Ein unvertrauenswürdiger Gast, der direkten Zugriff auf die Hardware hat, hat daher Zugriff auf sensitive Informationen, die nicht für ihn bestimmt sind. Daher können Schutzmechanismen, die auf Layern zwischen der Hardware und dem User-Space durchgesetzt werden, nicht wirken oder aber auch zerstört werden.

Um die Unterscheidung zwischen Kernel- und User-Mode treffen zu können, verfügt eine CPU i.d.R. über mindestens zwei hierarchische Schutzringe, wie in Abbildung 4.2 zu sehen ist. Der Betriebssystem-Kernel wird im Kernel-Mode betrieben, die User-Applikationen

im User-Mode. Mit einem Schutzring sind bestimmte Privilegien verbunden, über die Programmcode in diesem Ring verfügt. Äussere Ringe haben weniger Privilegien als die inneren Ringe, da ein weiter außen liegender Ring nur eine Teilmenge des Instruktionssatzes eines weiter innen liegenden Rings ausführen kann. Dem innersten Ring steht der komplette Instruktionssatz des Prozessors zur Verfügung [PG73]. Der Versuch eine Instruktion aus einem Ring heraus auszuführen, die nicht den Privilegien des Ringes entspricht, scheitert.

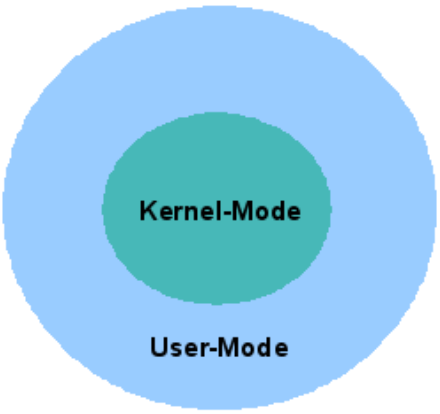

Abbildung 4.2: Schutzringe

Im Falle der Virtualisierung sind vollständige Betriebssysteme und die darunter liegenden Schichten der Gäste nicht-vertrauenswürdiger Code. Der Betriebssystem-Kernel eines Gasts geht jedoch davon aus, privilegierte Instruktionen direkt ausführen zu können, da das Betriebssystem üblicherweise auf dedizierter Hardware betrieben wird und privilegierte Instruktionen ausführen darf. Der VMM muss dem Betriebssystem, das in einem Gast betrieben wird, bei einer vollständigen Virtualisierung daher vortäuschen, dass es die privilegierten Instruktionen ausführen darf. Um den Erwartungen eines Betriebssystems gerecht zu werden existieren verschiedene Techniken.

Gerald J. Popek et al. definieren die formalen Anforderungen an eine virtualisierbare Architektur [PG73]. Die privilegierten Instruktionen sind jene Instruktionen, die eine Trap verursachen, wenn sie im User-Mode ausgeführt werden. Es wird keine Trap ausgelöst, wenn sie im Kernel-Mode ausgeführt werden [PG73]. Eine Trap bewirkt, dass die Kontrolle an eine festgelegte, vertrauenswürdige Routine übergeben wird und der Prozessor-Mode gewechselt wird. Eine Instruktion ist sensitiv, wenn die Instruktion auf sensitive Register oder Arbeitsspeicherbereiche lesend oder schreibend zugreift [RI00]. In einer virtualisierbaren Architektur muss die Menge der sensitiven Instruktionen eine Teilmenge der Menge der privilegierten Instruktionen sein, also eine Trap auslösen. Die Ausführung wird dann an den VMM übergeben, so dass hierdurch der VMM stets die Kontrolle über die sensitiven Instruktionen und damit die Hardware hat. Ein Gast kann nicht direkt auf die Hardware zugreifen, sondern muss durch

die Indirektionsebene des VMM auf die Hardware zugreifen. Der VMM kann, wenn eine Trap ausgelöst wird, das vom Betriebssystem des Gasts erwartete Verhalten emulieren. Versucht bspw. ein Gast auf sensitive Daten des Systems zuzugreifen, würden üblicherweise sensitive Daten über die reale Maschine gelesen. Dadurch, dass die Kontrolle an den VMM übergeben wird, kann der VMM stattdessen die sensitiven Daten der virtuellen Maschine liefern, so dass der Zustand der realen Maschine vor den Gästen verborgen bleibt. Der VMM pflegt zu diesem Zweck sog. Shadow-Strukturen für die Gäste. Der VMM muss den Zugriff auf sensitive Daten explizit gestatten bzw. emulieren, was dem Erlaubnisprinzip entspricht. Diese Technik nennt sich Trap & Emulate.

Damit Traps ausgelöst werden muss der Gast und damit das Betriebssystem innerhalb des Gasts mit niedrigeren Privilegien ausgeführt werden als üblich. Daher wird der Gast und das Betriebssystem des Gasts im User-Mode betrieben, was als Ring Deprivileging bezeichnet wird[1]. Der VMM verfügt über uneingeschränkte Privilegien und ist im innersten Schutzring lokalisiert. Da durch den VMM eine zusätzlich zu schützende Komponente hinzukommt, werden für die Virtualisierung mit *n* Schutzringen *n+1* Schutzringe benötigt. Prozessoren bieten jedoch nur eine begrenzte Anzahl von Schutzringen an. Dieses Problem ist allerdings in der Praxis nicht sichereheitsrelevant, da Prozessorarchitekturen häufig über mehr Schutzringe verfügen als tatsächlich benutzt werden. Falls dennoch nicht genug Schutzringe zur Verfügung stehen, können Schutzringe u.U. virtualisiert werden, indem *n+m* virtuelle Schutzringe auf *n* reale abgebildet werden [KZB+90].

Virtualisierung durch Trap & Emulate ist allerdings nur bei Prozessorachitekturen möglich, die die genannten Kriterien erfüllen. Bevor die x86-Architektur detailliert untersucht wird, soll diesbzgl. eine historische Architektur untersucht werden. Hierzu wird exemplarisch die VAX-Architektur der Digital Equipment Corporation (DEC) herangezogen, da die VAX-Architektur die genannten Kriterien nicht erfüllt. Auf der VAX-Architektur existieren Instruktionen, die sensitiv aber nicht privilegiert sind [KZB+90][2]. An der VAX-Architektur und am Microcode des Prozessors können jedoch Erweiterungen vorgenommen werden, um die Architektur vollständig mittels Trap & Emulate virtualisierbar zu machen, was bei der Konstruktion des nachweislich sehr sicheren VMM für die VAX-Architektur gemacht wurde [KZB+90]. Durch das Hinzufügen eines VM-Bits zu einem Prozessorregister kann singalisiert werden, ob der Prozessor augenblicklich VM-Code ausführt. Hierdurch können die problematischen Instruktionen so modifiziert werden, dass eine Trap ausgelöst wird, wenn VM-Code eine sensitive, nicht-privilegierte Instruktion aufruft. Komplexere Virtualisierungstechniken können hierdurch vermieden werden.

[1] Betriebssysteme werden jedoch i.d.R. für den Betrieb mit den höchsten Privilegien eines Systems konzipiert. Die daraus resultierenden Probleme werden mit Ring Aliasing bezeichnet [Zab08].

[2] Bspw. die Instruktion MOVPSL, die den Prozessor-Mode aus dem Prozessor-Status-Longword-Register liest, ist sensitiv, aber nicht-privilegiert [HR91]. Page Table Entries sind auf der VAX-Architektur sensitive Datenstrukturen, die mit nicht-privilegierten Instruktionen gelesen und geschrieben werden können [KZB+90].

Bis heute wurden verschiedene Techniken konzipiert, die eine Virtualisierung auch bei Architekturen ermöglichen, die die genannten Kriterien nicht erfüllen. Aus heutiger Sicht sagen die Kriterien von Gerald J. Popek et al. daher nicht mehr aus, ob eine Architektur grundsätzlich virtualisierbar ist. Vielmehr sagen die Kriterien aus, ob eine Architektur mittels reinem Trap & Emulate virtualisierbar ist [AA06]. Sichere Systeme sollten auf kleinen, isolierten Komponenten aufsetzen [HPHS04]. Virtualisierung durch Trap & Emulate ist sehr einfach zu implementieren und mit wenig Komplexität verbunden. Die Umsetzung dieser Technik beschränkt sich bzgl. der CPU-Virtualisierung darauf, die Trap-Handler zu implementieren. Traps sind bestehende Hardwaremechanismen, die gut erforscht sind und sich über lange Zeit bewährt haben. Diese bestehenden Hardwaremechanismen können für die Virtualisierung genutzt werden, wodurch ein hohes Maß an logischer Isolation erzielt werden kann und komplexere Methoden vermieden werden. Der entstehende VMM ist daher sehr klein, simpel und formal verifizierbar. Die TCB wächst nur unwesentlich, so dass diese Technik ein hohes Maß an Sicherheit gewährleisten kann. Gesteigerte Komplexität vermindert hingegen das Vertrauen in die Sicherheit des resultierenden Codes [BLRS08]. Dies wird bei der folgenden Betrachtung der Virtualisierung der x86-Architektur deutlich.

Die weit verbreitete x86-Architektur verfügt über vier Schutzringe (vgl. Abbildung 4.3). Ein Betriebssystem wird auf Ring null betrieben, die User-Space Applikationen i.d.R. auf Ring drei. Die x86-Architektur erfüllt ebenfalls nicht die Kriterien, die durch Gerald J. Popek et al. definiert wurden. Von insgesamt etwa 250 Instruktionen sind 17 sensitive, nicht-privilegierte Instruktionen vorhanden. Diese 17 Instruktionen lösen keine Traps aus und nehmen einem VMM für die x86-Architektur die Möglichkeit, das vom Betriebssystem eines Gasts erwartete Verhalten zu emulieren [RI00][1]. Man spricht davon, dass die Instruktionen still und leise abbrechen, anstatt eine geeignete Trap auszulösen [BDF+03]. Hierdurch kann ein Gast bspw. herausfinden, dass er nicht innerhalb des gewünschten Schutzrings ausgeführt wird [AA06] bzw. virtualisiert wird. Eine Virtualisierung nur mittels Trap & Emulate ist auf der x86-Architektur daher nicht möglich [AA06], so dass die x86-Architektur direkte Auswirkungen auf die Virtualisierungstechnologien hat.

Die Technik der Binary-Translation versucht die Beschränkungen der x86-Architektur dadurch zu umgehen, dass die senstiven, nicht-privilegierten Instruktionen explizit im Binärcode on the fly gefunden und umgeschrieben, d.h. übersetzt, werden. Somit kann bspw. eine Trap in den Binärcode eingefügt werden, so dass die sensitiven, nicht-privilegierten Instruktionen dem VMM daraufhin die Möglichkeit geben, das erwartete Verhalten zu emulieren. Die unproblematischen Instruktionen der x86-Architektur werden hingegen unverändert ausgeführt. Binary-Translation vereint somit das Paradigma der vollständigen Emulation und Virtualisierung. Binary-Translation erhält als Eingabe den Binärcode einer VM. Die problematischen Instruktionen können beliebig im Binärcode verteilt sein. Um die problematischen Instruktionen im Binärcode zu finden, wird der Binärcode nach diesen Instruktionen vor der

[1] [RI00] untersucht alle Instruktionen des Intel Pentium Prozessors auf Virtualisierbarkeit.

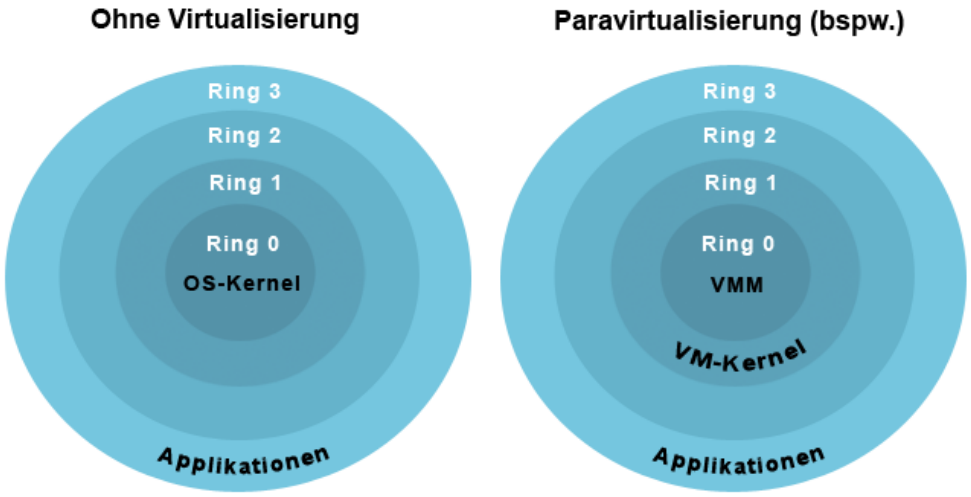

Abbildung 4.3: Schutzringe der x86-Architektur

Ausführung durchsucht. Nur überprüfter Code darf ausgeführt werden, da sich andernfalls problematische Instruktionen darin befinden können, die das gesamte System kompromittieren könnten.

Ein besonderes Problem stellt bei der Binary-Translation sich selbst modifizierender Code dar. Sich selbst modifizierender Code kann, nachdem der Code durch den VMM gescannt wurde, problematische Instruktionen einschleusen. Trap & Emulate ist resistent gegen dieses Problem, da unmittelbar bevor eine sensitive Instruktion ausgeführt werden soll, eine Trap ausgelöst wird. Bei der Binary-Translation dagegen, muss der Code explizit auf dieses Phänomen hin untersucht werden, um sicherzustellen, dass die Software das gleiche Ergebnis bei der Ausführung erhält, als würde sie ohne Intervention des VMM auf der Hardware ausgeführt. Ein derartiges Vorgehen stellt eine Bereinigung des Codes dar, um ihn vertrauenswürdig zu machen und gefahrlos ausführen zu können (engl. sanitising). Den nicht-vertrauenswürdigen Code explizit zu bereinigen steht im Widerspruch zu Security by Design und dem Erlaubnisprinzip. Jeder Fehler in der Programmlogik führt zu einer Bedrohung der Systemintegrität. Die sensitiven, nicht-privilegierten Instruktionen werden explizit herausgefiltert und übersetzt, anstatt gezielt erlaubt zu werden. Der Anteil des Codes, der für eine Bereinigung zuständig ist, ist in sicheren Systemen jedoch zu minimieren, da ein Fehler in diesem Code mit hoher Wahrscheinlichkeit dazu führt, dass die TCB kompromittiert werden kann [MMH08].

Da der gesamte Quell-Binärcode überprüft werden muss, werden i.d.R. performancesteigernde Maßnahmen ergriffen, was zusätzliche Komplexität des Verfahrens bedingt.

Bspw. kann versucht werden, den Binärcode wie ein Compiler zu optimieren oder es kann ein Trace-Cache während der Ausführung aufgebaut werden, der bereits überprüfte Binärcodeblöcke enthält, so dass bei erneuter Ausführung des gleichen Binärcodeblocks CPU-Zyklen gespart werden können [RG05].

Laut Kevin Lawton, werden bei der Binary-Translation Breakpoints in den Binärcode eingefügt, um die Ausführung des Codes gezielt unterbrechen zu können [Law99]. Hierdurch kann Code, der ausgeführt wird und selbst Teile seines Binärcodes liest auch die Breakpoints lesen. Auch Sprunganweisungen können durch sich selbst modifizierenden Code manipuliert werden. Der durch Binary-Translation veränderte Code muss vor Veränderungen geschützt werden und es dürfen keine sensitiven Informationen gewonnen werden können [RI00]. Daher muss sehr viel Aufwand betrieben werden um diese Ziele zu erreichen.

Binary-Translation ist ein äußerst komplexes Verfahren. Zwar ist der Quell- und Zielinstruktionssatz identisch, aber es muss ein hoher Aufwand betrieben werden um wenige problematische Instruktionen aus dem Binärcode zu filtern, die i.d.R. nicht häufig ausgeführt werden [WSG02] und nicht durch Schutzmechanismen der x86-Architektur behandelt werden. Laut Kevin Lawton, ist die Komplexität der Binary-Translation derartig hoch, dass ein sehr sicherer VMM möglicherweise unerreichbar ist [Law99]. Die Vertrauenswürdigkeit des Verfahrens wird hierdurch beeinträchtigt.

Paravirtualisierung verfolgt einen anderen Weg als Binary-Translation, um das Problem der sensitiven, nicht-privilegierten Instruktionen der x86-Architektur zu lösen. Da die problematischen Instruktionen primär vom Gast-Betriebssystem benutzt werden, liegt es nahe das Gast-Betriebssystem anzupassen. Bei der Paravirtualisierung werden die Betriebssysteme daher so angepasst, dass auf diese Instruktionen verzichtet wird und das Gast-Betriebssystem in einem weiter außen liegenden Schutzring betrieben werden kann. Hierbei wird einer virtuellen Maschine eine idealisierte Abstraktion der x86-Architektur bereitgestellt, die der darunter liegenden Hardware ähnelt, aber nicht mit ihr identisch ist [BDF+03] und besser für die Virtualisierung geeignet ist als die x86-Architektur. Da ein paravirtualisierter Gast durch Ring-Deprivileging bei der x86-Architektur, wie in Abbildung 4.3, auf Schutzring eins betrieben wird, ist es unmöglich sensitive Instruktionen am VMM vorbei auszuführen. Zugriffe der Gäste auf die Hardware müssen erst vom VMM gestattet werden, was dem Erlaubnisprinzip gerecht wird. Die User-Space Applikationen können auf der x86-Architektur weiterhin auf Ring drei betrieben werden.

Da die angepassten Betriebssysteme sozusagen wissen, dass sie virtualisiert betrieben werden, ist die Implementation des paravirtuellen VMM bzgl. der CPU-Virtualisierung wesentlich einfacher als es bei Binary-Translation der Fall ist, da der Aufwand entfällt den Gast zu illusionieren. Die TCB, die für Gäste durch Paravirtualisierung hinzukommt, kann daher bzgl. der CPU-Virtualisierung klein ausfallen. Es besteht kein Zwang, den Anforderungen

der Gäste an eine x86-CPU gerecht werden zu müssen, was die Komplexität der Technologie vermindert. Die Tatsache, dass ein Betriebssystem innerhalb eines Gasts betrieben wird muss nicht vor dem Betriebssystem verborgen werden. Die Anforderung, die ein paravirtuelles Betriebssystem an einen paravirtuellen VMM stellt, ist, eine schmale Schnittstelle bereitzustellen, die die problematischen Instruktionen der x86-Architektur kapselt. Der VMM ähnelt daher einem herkömmlichen Betriebssystem, das eine Schnittstelle für User-Space Applikationen bereitstellt, aber wesentlich schmaler als ein herkömmliches Betriebssystem ist. Über diese Schnittstelle kann das Gast-Betriebssystem kontrolliert und logisch isoliert auf den Teil der Hardware zugreifen, der dem Betriebssystem vom VMM zugesichert wird. Nicht-sensitive Instruktionen können gefahrlos ausgeführt werden. Führt ein Gast sensitive Instruktionen aus, wird die Systemintegrität nicht bedroht, da das System durch die Schutzmechanismen der x86-Architektur auf Hardwareebene gesichert wird und eine Trap zum VMM ausgelöst wird oder die Instruktionen lautlos abbrechen. Ein unvertrauenswürdiger Gast wird vom VMM gezwungen bzgl. der sensitiven, nicht-privilegierten Instruktionen zu kooperieren, um Zugriff auf sensitive Systemressourcen zu erlangen. Kooperiert der Gast nicht, kann der Gast nicht auf die Ressourcen zugreifen. Ein paravirtueller VMM ist daher, bzgl. der CPU-Virtualisierung, vertrauenswürdiger als ein VMM, der Binary-Translation verwendet und vergleichbar mit einem VMM der ein reines Trap & Emulate verwendet.

Eine Paravirtualisierung erlaubt i.d.R. die User-Space Applikationen unmodifiziert auf dem modifizierten Gast-Betriebssystem zu betreiben [BDF+03]. Sicherheitspatches, die Sicherheitslücken in Softwareprodukten schließen, müssen i.d.R. für jede Architektur erzeugt werden, für die die Softwareprodukte verfügbar sind. Sicherheitspatches, die veröffentlichte Sicherheitslücken für User-Space Applikationen einer paravirtuellen Architektur schließen, werden nicht mit zeitlicher Verzögerung veröffentlicht, da Sicherheitspatches nicht explizit für die User-Space Applikationen der paravirtuellen Architektur erzeugt werden müssen. Für das modifizierte Gast-Betriebssystem gilt dies jedoch nicht. Der Aufwand, das Betriebssystem anzupassen, entspricht i.d.R. dem Aufwand, das Betriebssystem auf eine andere Prozessorarchitektur zu portieren[3] [WCSG05] [BDF+03]. Daher müssen Sicherheitspatches, die Sicherheitslücken des Gast-Betriebssystems adressieren, explizit für die paravirtuelle Architektur erzeugt werden. Sicherheitspatches werden hierdurch ggf. erst mit zeitlicher Verzögerung für das paravirtuelle Gast-Betriebssystem veröffentlicht. Ein Angriff auf eine Sicherheitslücke zum Zeitpunkt ihrer Veröffentlichung wird als Zero Day Angriff bezeichnet [M"u07]. Ein paravirtuelles Gast-Betriebssystem ist daher stärker durch Zero Day Angriffe gefährdet als ein Gast-Betriebssystem bei einer vollständigen Virtualisierung.

Neue Modelle von x86-CPUs verfügen über unterstützende Technologien[4] , die der Virtualisierung dienen. Dabei wurde nicht die bestehende Architektur abgeändert, sondern ergänzt,

[3] Änderungen sind primär am Hardware Abstraction Layer (HAL) eines Betriebssystems vorzunehmen [WCSG05] [BS06].

[4] Die Technologien werden als AMD-SVM [AMD09] und Intel VT-x [Zab08] bezeichnet.

um Kompatibilität zu wahren. Ein Design-Ziel der Technologien ist, die Notwendigkeit für Binary-Translation und Paravirtualisierung, die durch nicht vorhandene „Virtualisierbarkeit“ der x86-Architektur ensteht, zu beseitigen [UNR+05]. Dabei werden von der CPU Container konstruiert, die jeweils eine virtuelle CPU darstellen und neue Prozessor-Modi einführen. Die Container entsprechen zusätzlichen Schutzringen. Die Prozessor-Modi identifizieren, ob der Container eine virtuelle oder die reale Umgebung ist und welche Berechtigungen mit dem Container verbunden sind. Der VMM wird in einem Container mit den höchsten System-Privilegien ausgeführt, was konzeptionell einem Schutzring -1 entspricht. Die Gäste werden innerhalb eines Containers mit niedrigeren Privilegien ausgeführt, was aber konzeptionell dem Schutzring null entspricht. Das Gast-Betriebssystem kann daher in dem Schutzring betrieben werden, für den es entwickelt wurde [UNR+05]. Ein Gast befindet sich in einer, durch Hardwaremechanismen unterstützten, virtuellen Umgebung. U.a. wenn ein Gast problematische Instruktionen der x86-Architektur aufruft, wird der Container niedriger Privilegien verlassen und eine Trap zum Container des VMM, dem realen Container, ausgelöst [Zab08][5]. Dem VMM wird es hierdurch möglich, das erwartete Verhalten auch bei sensitiven, nicht-privilegierten Instruktionen zu emulieren und eine Virtualisierung ohne Binary-Translation und Paravirtualisierung zu erzielen.

Durch den Einsatz der Virtualisierungstechnologien einer CPU ist eine Virtualisierung mittels reinem Trap & Emulate bzgl. der CPU auf der x86-Architektur möglich. Die Isolation wird mit Hardwaremechanismen erzielt. Die Sicherheitseigenschaften des Trap & Emulate Konzeptes gelten daher auch für die CPU-unterstützte Virtualisierung. Die CPU-Unterstützung kann die Komplexität beim Design eines VMM reduzieren. Der VMM kann bzgl. der CPU-Virtualisierung schmal und einfach implementiert werden. Die aufwendigere Binary-Translation wird vermieden.

4.3.2 Memory-Virtualisierung

Zu virtualisierende Systeme verfügen nur über eine begrenzte Menge an Ressourcen. Der Hauptspeicher bildet hierbei keine Ausnahme. Der Hauptspeicher muss grundsätzlich unter den handelnden Subjekten eines Systems verteilt und zugänglich gemacht werden. Bei der Virtualisierung sind die handelnden Subjekte der VMM und die Gäste. Subjekte dürfen den Hauptspeicher nur isoliert voneinander verwenden, da die Subjekte sensitive Daten, wie Passwörter, kryptographische Schlüssel und Programmcode im Hauptspeicher halten. Der Zugriff der Gäste auf den Hauptspeicher muss daher isoliert geschehen. Die sensitiven Daten dürfen den anderen Subjekten einer virtuellen Umgebung nicht oder nur definiert zuänglich sein, damit die Subjekte keine Informationen gewinnen können, die nicht für sie bestimmt sind. Bei der Virtualisierung wird i.d.R. angestrebt, dass die sensitiven Daten im

[5] Das Verlassen des Containers wird als Exit bezeichnet.

Hauptspeicher ausschließlich einem Gast zugänglich sind, indem die Speicherbereiche der Gäste voneinander isoliert werden.

Um die logische Isolation des Hauptspeichers unter Verwendung von Trap & Emulate erzielen zu können, definieren Gerald J. Popek et al., dass eine Architektur einen Schutzmechanismus oder eine Umrechnung von Adressen bereitstellen muss [PG73] um das reale System und die Gäste vor dem jeweils aktiven Gast zu schützen [RI00]. Derartige Mechanismen werden durch Paging und/oder Segmentation bereitgestellt.

Segmentation teilt den eindimensionalen Speicherbereich, dessen Speicheradressen von null bis zu einem Maximum *n* reichen, in mehrere unabhängige und unterschiedlich große Speicherbereiche auf [S01], die als Segmente bezeichnet werden. Die Segmente können mit Zugriffsberechtigungen versehen werden oder werden durch Privilegien die mit den Schutzringen des Prozessors eng verknüpft sind geschützt.

Paging ermöglicht einen größeren linearen Adressbereich ohne in mehr physischen Hauptspeicher investieren zu müssen [S01]. Paging implementiert eine Umrechnung von virtuellen Speicheradressen in physische Speicheradressen durch Einsatz einer Indirektionsebene in Form einer Seitentabelle. Hierdurch können virtuell zusammenhängende Speicherseiten beliebig fragmentiert im physisischen Hauptspeicher lokalisiert sein und auf Sekundärspeichermedien ausgelagert werden, was als Swapping bezeichnet wird. Abbildung 4.4 zeigt die Abbildung von virtuellen auf physische Speicherseiten.

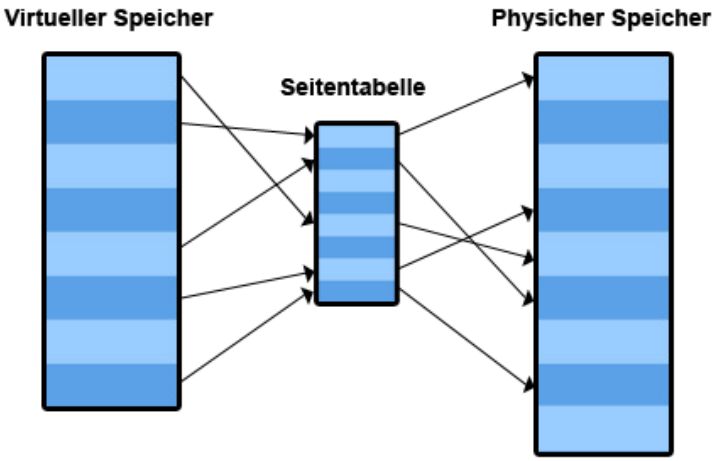

Abbildung 4.4: Paging

Unter Einsatz von Paging, kann ein VMM den Hauptspeicher virtualisieren, indem eine zusätzliche Indirektionsebene in Software implementiert wird, die den Erwartungen eines Gasts

gerecht wird. Ohne die Indirektionsebene bestünde das Problem, dass die Gäste Speicherseiten selbst, unter Berücksichtigung der anderen Gäste, registrieren müssten. Eine Zusammenarbeit von Gästen ist jedoch ausgeschlossen, da sich die Gäste gegenseitig nicht vertrauen und vollständige Virtualisierung vorraussetzt, dass die Betriebssysteme der Gäste nicht wissen, dass sie virtualisiert werden. Die Betriebssysteme gehen also davon aus, dass sie über den gesamten, zusammenhängenden Hauptspeicher verfügen. Dies ist jedoch nicht möglich, da ein Gast eine Speicherseite registrieren könnte, die bereits von einem anderen Gast oder dem VMM benutzt wird. Durch die zusätzliche Indirektionsebene kann die Illusion, über den gesamten, zusammenhängenden Hauptspeicher zu verfügen, aufrecht erhalten werden. Gäste können nur isoliert auf Teile des Hauptspeichers zugreifen.

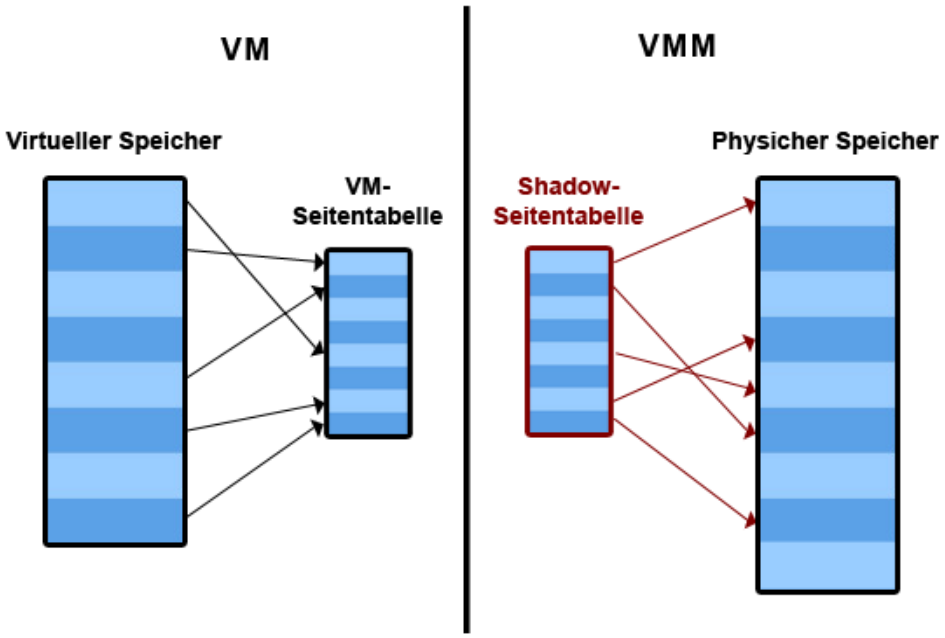

Abbildung 4.5: Shadow-Paging

Beim Einsatz einer zusätzlichen Indirektionsebene unterscheidet sich die Sicht eines Gast-Betriebssystems auf den physischen Arbeitsspeicher von der Sicht des VMM. Shadow-Seitentabellen, werden vom VMM für jeden Gast geplegt und mit der Seitentabelle des Gasts konsistent gehalten. Hierzu sind zwei Techniken verbreitet [AMD08]. Versucht ein Gast eine Speicherseite in der Seitentabelle zu registrieren, wird eine Trap zum VMM ausgelöst, weil die Seitentabelle schreibgeschützt betrieben wird. Der VMM registriert ein Mapping in der Shadow-Seitentabelle, das nicht mit der Seitentabelle des Gasts übereinstimmen muss, aber konsistent mit ihr ist. Zur tatsächlichen Adressierung des Hauptspeichers durch einen Gast wird daher die Shadow-Seitentabelle verwendet, die die virtuellen Speicherseiten eines

Gasts auf die physischen Speicherseiten des Rechners abbildet [RG05]. Die andere Technik lässt Gast-Betriebssysteme ihre Seitentabelle ohne Intervention manipulieren. Seitenfehler, die durch Verstöße gegen die Schutzrichtlinie einer Speicherseite ausgelöst werden, müssen zu den Gästen weitergeleitet und von den Gästen behandelt werden. Seitenfehler, die durch nicht vorhandene Einträge in der Shadow-Seitentabelle ausgelöst werden, müssen behandelt werden, indem neue Speicherseiten in der Shadow-Seitentabelle registriert werden [AA06].

Durch den Einsatz von Shadow-Seitentabellen kann der VMM den Zugriff auf Speicherseiten granuliert kontrollieren [RG05] und den Zugriff auf Speicherseiten, die nicht dem jeweiligen Gast zuzurechnen sind, verbieten. Das Paging stellt bei der Virtualisierung hierdurch eine transparente Schutzfunktion dar, so dass Gäste nur auf ihren Teil des Speichers zugreifen können ohne es zu bemerken. Shadow-Seitentabellen zu pflegen stellt keine wesentliche Erhöhung der Komplexität dar, da die zusätzliche Indirektionsebene der Shadow-Seitentabelle eine ähnliche Funktionalität anstrebt wie eine herkömmliche Seitentabelle bzgl. der Komplextiät. Häufig stellt eine Architektur die wesentlichen Mechanismen, wie Paging und Segmentation, die für das Shadow-Paging notwendig sind, auf Hardwareebene bereit. Die Abbildung der Speicherseiten in Software ist, wie gezeigt wurde, einfach zu implementieren und kann auch direkt in Hardware implementiert werden. Grundsätzlich kann daher mit Shadow-Seitentabellen ein hoher Grad an logischer Isolation in virtualisierten Umgebungen erzielt werden (vgl. [KZB+90], [HR91]).

Die Kontrolle des VMM über den Zugriff auf Speicherseiten und die Pflege der Shadow-Seitentabellen ermöglicht dem VMM, bei starker Speicherauslastung, Speicherseiten auf Sekundärspeicher auszulagern. Der VMM kann transparent, wie ein herkömmliches Betriebssystem, Speicherseiten auf die Festplatte schreiben, wodurch physischer Hauptspeicher frei wird. Hierdurch können allerdings auch sensitive Daten eines Gasts ausgelagert oder extern modifiziert werden. Mechanismen des Gast-Betriebssystems, um zu verhindern, dass sensitive Daten nicht-flüchtigen Speicher, also i.d.R die Festplatte, erreichen, können außer Kraft gesetzt werden [GR05]. Sensitive Daten können zwar auch durch das übliche Swapping eines herkömmlichen Betriebssystems ausgelagert werden, aber bei der Integration in den VMM wird die Komplexität des VMM erhöht und zusätzliche Bedrohung geht von Thrashing aus [BDF+03]. Der VMM, der Speicherseiten auslagert, muss sowohl den Gast als auch die Speicherseiten des Gasts auswählen, die er auslagern möchte. Der VMM verfügt jedoch über weniger Informationen über die auszulagernden Speicherseiten als ein herkömmliches Betriebssystem, das detaillierte Informationen darüber hat, welche Speicherseiten zu einem bestimmten Zeitpunkt nicht notwendigerweise gebraucht werden [Wal02]. Werden beim Swapping Speicherseiten ausgewählt die zum Workload eines Gasts oder zum Workload von Applikationen eines Gasts gehören, müssen regelmäßig Speicherseiten ein und wieder ausgelagert werden. Daher kann es zu einer Beeinträchtigung dieses Gasts oder anderer Gäste durch unzureichende Performance Isolation kommen, so dass die

Hardwarekomponenten primär mit dem Swapping von Speicherseiten eines Gasts beschäftigt sind und CPU-Zyklen verschwendet werden, statt Sie anderen Gästen zur Verfügung zu stellen. Die Bedrohung des Thrashings wird durch nicht-vertrauenswürdige Gäste, die explizit versuchen ein unerwünschtes Verhalten herbeizuführen, verstärkt. Verzichtet ein VMM auf Swapping sinkt die Flexibilität bzw. das Konsolidierungspotential des VMM, da Teile des Hauptspeichers statisch einem einzigen Gast zugewiesen werden, der ggf. nicht dauerhaft die statisch zugewiesene Menge des Hauptspeichers benötigt.

Die Zuweisung des Hauptspeichers an Gäste geschieht i.d.R. durch eine der folgenden drei Arten:

Statisch Wenn eine VM erzeugt wird, erhält sie einen festgelegten Anteil statischer Größe des Hauptspeichers. Die Größe des Hauptspeichers ist während der Lebenszeit eines Gasts unveränderlich. Dieses Konzept wird bei Pure-Isolation-VMMs verwendet und garantiert ein hohes Maß an Isolation.

Dynamisch Der verfügbare Hauptspeicher wird dynamisch, je nach Bedarf, zwischen den Gästen aufgeteilt. Die dynamische Aufteilung kann dazu führen, dass Hauptspeicher vielfach zwischen den Gästen hin- und hergeschoben werden muss. Dieser Umstand kann durch schadhafte Gäste ausgenutzt werden.

Minimum/Maximum Wenn ein Gast erzeugt wird, wird statisch festgelegt, wieviel Hauptspeicher eine VM stets mindestens und maximal zur Verfügung hat. Innerhalb dieses Bereichs wird der verfügbare Hauptspeicher zwischen den Gästen, je nach Bedarf, dynamisch verteilt.

Um Hauptspeicher zwischen Gästen hin- und herzubewegen ohne die Sicherheitsnachteile des Swappings in Kauf nehmen zu müssen, kann die paravirtuelle Technik des Memory-Balloonings eingesetzt werden. Hierbei ist ein Prozess direkt in einem Gast aktiv und erzeugt Speicherdruck. Hierdurch wird das Gast-Betriebssystem angehalten Hauptspeicher freizugeben, der dann durch den Balloon-Prozess an den VMM übergeben wird. Da das unvertrauenswürdige Gast-Betriebssystem detaillierte Informationen über den Speicherbedarf hat, wird das Thrashing-Problem hierbei umgangen. Wird der Ballooning-Prozess kompromittiert, ist nur das jeweilige Gast-Betriebssystem betroffen. Ein schadhaftes Verhalten wird in dem schadhaften Gast isoliert.

Die x86-Architektur verfügt ab dem 80368-Modell über Paging und Segmentation [Chi07]. Somit erfüllt die x86-Architektur die Anforderungen von Gerald J. Popek et al. bzgl. der Memory-Virtualisierung. Vollständige Virtualisierung des Hauptspeichers der x86-Architektur kann daher durch Binary-Translation in Verbindung mit Shadow-Seitentabellen implementiert werden. Durch die Memory-Virtualisierung wird die Sicherheit von Systemen, die Binary-Translation zur vollständigen Virtualisierung verwenden, daher konzeptionell nicht beeinträchtigt, sofern auf Swapping verzichtet wird.

Die Paravirtualisierung muss auch bei der Virtualisierung des Hauptspeichers nicht den üblichen Erwartungen eines Betriebssystems gerecht werden. Ein Betriebssystem wird angepasst um mit dem VMM zusammenzuarbeiten. Einem paravirtuellem VMM steht es offen, ebenfalls Shadow-Seitentabellen zu verwenden. Ein Gast kann hierbei bspw. über eine definierte Schnittstelle Speicherseiten vom VMM anfordern, dem die volle Kontrolle über den Hauptspeicher obliegt. Neben dem Einsatz von Shadow-Seitentabellen ermöglicht die Paravirtualisierung, die Gäste selbst für das Paging verantwortlich zu machen, was als Self-Paging bezeichnet wird [BDF+03]. Ein Gast-Betriebssystem verwendet eine Speicherseite aus seinem reservierten Speicherpool und registriert die Seite über eine Schnittstelle des VMM. Dabei wird vom VMM sichergestellt, dass nur Speicherseiten registriert werden, die dem Gast vom VMM zugewiesen wurden. Die Schreibrechte auf die Seitentabelle werden dem Gast-Betriebssystem abgetreten. Aktualisierungen werden vom VMM verifiziert, damit der Gast nicht nachträglich Speicherseiten unautorisiert manipulieren kann. Speicherseiten zu lesen ist ohne Intervention des VMM möglich.

Komplexität der Seitenverwaltung wird durch Self-Paging aus dem VMM entfernt [BDF+03]. Der VMM muss ausschließlich eingreifen um die Isolation zu gewährleisten. Daher ist durch den Einsatz paravirtueller Speicherverwaltung ein hohes Maß an Sicherheit möglich. Prinzipiell ermöglicht der Einsatz von Paravirtualisierung auch, die Erwartung des Gast-Betriebssystems so zu verändern, dass die Notwendigkeit eines linearen, zusammenhängenden Adressbereichs entfernt werden kann [Chi07]. Hierdurch würde weitere Komplexität im VMM eingespart werden, da der Aufwand den Gast zu illusionieren entfällt.

Eine CPU-unterstützte Virtualisierung verwendet üblicherweise Shadow-Seitentabellen, da die CPU-unterstützte Virtualisierung einem Trap & Emulate entspricht. Die Seitentabelle wird bspw. schreibgeschützt betrieben, wodurch eine Trap zum VMM ausgelöst wird, sobald ein Gast versucht die Seitentabelle zu manipulieren. Der VMM kann dann die Shadow-Seitentabellen pflegen und das vom Gast-Betriebssystem erwartete Verhalten emulieren. Die Sicherheitseigenschaften der Shadow-Seitentabellen gelten daher auch bzgl. der CPU-unterstützten Virtualisierung.

Die AMD-V-Technologie führt eine zusätzliche Indirektionsebene zur Adressübersetzung auf MMU-Ebene ein. Die physischen Adressen des Gasts werden hierbei, mittels sog. Nested Page Tables, auf die physischen Adressen des Hosts abgebildet und stellen daher Shadow-Seitentabellen auf Hardwareebene dar. Die Nested Page Tables werden vom VMM gepflegt [AMD08]. Wenn die Nested Page Tables etabliert sind, muss der VMM nicht mehr intervenieren. Hierdurch wird Komplexität aus dem VMM entfernt, der die zur Verfügung stehenden Datenstrukturen der Hardware verwenden kann und auf die Buchhaltung über die Speicherseiten verzichten kann, so dass der VMM im Vergleich zum Einsatz von Shadow-Paging auf Softwareebene noch schmaler werden kann. Intel plant eine äquivalente Technologie namens Extended Page Tables [Int09a], so dass ein VMM mit Hardware-Unterstützung der Memory-Virtualisierung in Zukunft noch schmaler bzw. weniger komplex werden kann.

4.3.3 IO-Virtualisierung

Die x86-Architektur wird auch als offene Architektur bezeichnet [SVL01]. Verschiedenste Hersteller können IO-Komponenten für die x86-Architektur entwickeln. Es existiert daher eine unüberschaubare Menge von IO-Komponenten für die x86-Architektur.

Diese IO-Komponenten müssen bei der Virtualisierung von mehreren Betriebssystemen gleichzeitig nutzbar sein. IO-Komponenten im Mainframe-Bereich werden häufig mit Fokus auf Virtualisierung entwickelt [KS08]. Die Komponenten für die x86-Architektur werden i.d.R. für den Betrieb unter einem einzigen Betriebssystem entwickelt. Daher stehen der Virtualisierungstechnologie, aufgrund möglicher Beschränkungen durch die Hardware, nur beschränkte Mechanismen zur Verfügung, um die Sicherheitsrichtlinie der Isolation durchzusetzen.

Die Fülle an Komponenten bewirkt, dass viele Treiber benötigt werden, um die Komponenten verwenden zu können. Treiber für die x86-Architektur werden häufig schlecht konzipiert und gelten als eine der größten Fehlerquellen in einem System [FHN+04]. Fehler in Gerätetreibern haben oft fatale Auswirkungen auf die Systemstabilität, da Gerätetreiber zu den privilegiertesten Komponenten eines Systems gehören und i.d.R. nicht von den anderen hochprivilegierten Komponenten isoliert werden.

Bei der Virtualisierung haben die Gäste keinen direkten Hardwarezugriff. Stattdessen muss der VMM den Zugriff auf die Hardware koordinieren. Der VMM bietet den Gästen hierzu virtuelle Hardwarekomponenten an. Die Zugriffe der Gäste auf die virtuellen Hardwarekomponenten münden in koordinierten Zugriffen auf die realen Hardwarekomponenten. Hierzu ist der VMM auf Gerätetreiber angewiesen und der VMM muss den Treibern vertrauen. Die Treiber sind daher Teil der TCB des VMM.

Es liegt im Interesse der Sicherheit die Treiber aus dem VMM zu entfernen und nicht auf ihre korrekte Funktionsweise vertrauen zu müssen. Ohne Virtualisierung ist jedes Betriebssystem selbst für den Betrieb der IO-Komponenten zuständig und verwendet eigene Treiber. Da ein Pure-Isolation VMM jedem Gast eigene IO-Komponenten zuweist, die nicht zwischen mehreren Gästen geshared werden, kann ein Pure-Isolation VMM die Treiber in die Gäste verlagern. Der VMM weist jedem Gast eigene IO-Komponenten zu. Hierdurch sind die Gäste selbst für den Betrieb der IO-Komponenten zuständig und der VMM vertraut den Treibern nicht [KS08]. Die Gäste verwenden nicht gemeinsam den gleichen Software-Stack bzgl. der unvertrauenswürdigen Treiber. Die TCB des VMM wird hierdurch erheblich schmaler, was sich positiv auf die Sicherheit und Stabilität des VMM auswirkt.

IO-Komponenten für die x86-Architektur verfügen jedoch über Direct Memory Acess (DMA). IO-Komponenten kommunizieren hierbei direkt, d.h. ohne Adressübersetzung, mit dem Arbeitsspeicher und schreiben Daten an programmierbare Adressen im Arbeitsspeicher. Eine Intervention durch den VMM ist nicht möglich. Die Treiber der IO-Komponenten legen fest, an welche Adresse im Arbeitsspeicher die Daten geschrieben werden [BYM05]. Daten der

Gast-Betriebssysteme oder des VMM können überschrieben werden, wodurch ein Treiber die Verfügbarkeit des Systems beeinträchtigen kann oder das System gezielt manipulieren und kompromittieren kann. Die Treiber für IO-Komponenten der x86-Architektur in die Gäste zu verlagern hätte zur Folge, dass ein Gast beliebige Treiber einschleusen könnte. Hierdurch wäre es unmöglich den Zugriff des Gastes auf Speicherbereiche einzuschränken. Ein Angreifer, der den Gast kompromittiert hat, könnte hierdurch eine IO-Komponente so programmieren, dass der VMM zerstört würde oder das gesamte System vom Angreifer kompromittiert würde. Ein VMM für die x86-Architektur kann daher nicht die Gäste für den Betrieb der IO-Komponenten verantwortlich machen, da er hierzu den Gästen vertrauen müsste. Der VMM muss die Treiber daher in seine TCB aufnehmen. Der VMM muss den Treibern vertrauen, obwohl die Treiber unvertrauenswürdig sind, so dass der VMM die Sicherheitheit konzeptionell nicht gewährleisten kann. Bei einem Fehler des Treibers sind ggf. alle Gäste von den Auswirkungen betroffen. Ohne Virtualisierung wäre dagegen nur eine Maschine betroffen. Die logische Isolation wird beeinträchtigt.

Da der VMM die Treiber beinhaltet, muss er den Gästen Zugriff auf die Geräte ermöglichen. Unterschiedliche Virtualisierungstechnologien verwenden hierbei verschiedene Techniken.

Eine vollständige Virtualisierung der x86-Architektur emuliert typische IO-Komponenten, damit das Gast-Betriebssystem nicht angepasst werden muss. Ein Gast kommuniziert mit dem emulierten Gerät, wofür das Gast-Betriebssystem i.d.R. bereits Treiber bereitstellt. Die Daten, die das emulierte Gerät erreichen, werden über den Treiber im VMM an das reale Gerät übermittelt. Da das Gast-Betriebssystem bereits über Treiber für das emulierte Gerät verfügt, muss kein zusätzlicher, nicht-vertrauenswürdiger Treibercode geschrieben werden. Das emulierte Gerät muss allerdings den Erwartungen des Treibers im Gast gerecht werden. Das gesamte Gerät muss virtuell exakt in Software nachgebildet werden. Zwar ist es dem VMM möglich sich auf je ein zu emulierendes Gerät pro Gerätetyp zu beschränken, so dass bspw. nur ein ganz bestimmter Network Interface Controller (NIC) emuliert wird, aber es muss dennoch eine große Menge komplexen Emulationscodes geschrieben werden. Die TCB eines Gasts steigt hierdurch im Vergleich zu einem Betrieb ohne Virtualisierung erheblich an, da der Emulationscode Teil der TCB eines Gasts ist. Ein Fehler im Emulationscode gefährdet im Mindesten Schutzziele, die für den Gast gelten.

Ein paravirtueller VMM exportiert hingegen eine schmale Schnittstelle für verschiedene Gerätetypen, wie Blockgeräte oder Netzwerk-Adapter. Das Treiber-Konzept wird als Split-Driver-Model bezeichnet [Chi07]. Im Gast werden sehr einfache unvertrauenswürdige Nutzer der Schnittstelle des VMM als Treiber implementiert, die hierdurch mit dem VMM kommunizieren. Es müssen keine realen Geräte in Software nachgebildet werden. Der Bedarf für komplexen Emulationscode entfällt hierdurch. Die Komplexität des VMM steigt, im Gegensatz zu einer vollständigen Virtualisierung mit Emulation, nur unwesentlich. Fehler in der Schnittstelle sind unwahrscheinlicher. Die Schnittstelle kann formal verifiziert werden.

Der VMM muss auf die IO-Komponenten, die er den Gästen anbietet, direkt zugreifen können. Der VMM muss daher potentiell Treiber für alle IO-Komponenten bereitstellen, die in einem x86-System verbaut werden. Ein VMM ist ein Betriebssystem und für ein Betriebssystem müssen Treiber für IO-Komponenten je neu geschrieben werden. Daher muss der VMM, durch die Fülle von IO-Komponenten, eine Fülle von Treibern neu implementieren. Um diesen enormen Implementationsaufwand zu vermeiden, kann der VMM auf ein bestehendes Betriebssystem aufsetzen. Da die gängigen Betriebssysteme bereits Treiber für die häufigsten IO-Komponenten bereitstellen, erhält auch der VMM die Geräteunterstützung durch das Betriebssystem auf das er aufsetzt. Diese Architektur wird, wie in Kapitel 3.3 gezeigt wurde, auch als Hosted-Architektur bzw. Typ-II-VMM bezeichnet [SN05] und wird in Abbildung 4.6 dargestellt.

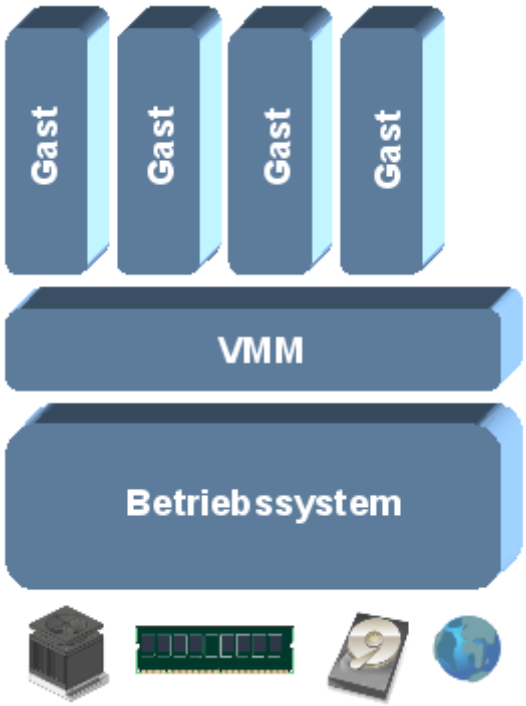

Abbildung 4.6: Hosted-Architektur

Die Hosted-Architektur hat gravierende Auswirkungen auf die Sicherheit des VMM. Der VMM nimmt ein komplettes, herkömmliches Betriebssystem in seine TCB auf. Hierdurch wird die Grundannahme von Stuart E. Madnick et al. [MD73], dass ein VMM besser als ein herkömmliches Betriebssystem dazu geeignet sei, die Isolation in einem System zu gewährleisten, da ein VMM wesentlich schlanker als ein herkömmliches Betriebssystem sei, ungültig. Ein derartiger VMM ist nicht kleiner und weniger komplex als ein herkömmliches Betriebssystem, sondern erheblich größer. Die TCB eines Gasts wächst erheblich, denn statt eines einzigen Betriebssystems besteht die TCB eines Gasts hierbei aus zwei gesamten Betriebssystemen

und dem VMM. Der VMM erbt alle Sicherheitsprobleme und -lücken eines herkömmlichen Betriebssystems. Besonders bzgl. des Szenarios der Serverkonsolidierung wird die Isolation hierdurch beeinträchtigt, da für alle konsolidierten Maschinen die Sicherheitsprobleme des Host-Betriebssystems zusätzlich gelten. Auch eine Erhöhung der Sicherheit durch die Trennung von Diensten durch Virtualisierung ist, da die Grundannahme ungültig ist, nicht zu erwarten.

Ein VMM kann auch direkt auf der Hardware betrieben werden, ohne auf ein herkömmliches Betriebssystem aufzusetzen. Ein derartiger VMM wird, wie in Kapitel 3.3 gezeigt wurde, als Bare-Metal-VMM bzw. Typ-I-VMM bezeichnet und ist in Abbildung 4.7 dargestellt.

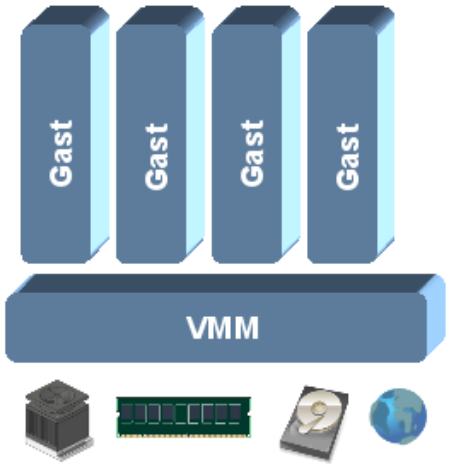

Abbildung 4.7: Bare-Metal-VMM

Um die Treiber und das Betriebssystem aus dem VMM zu entfernen, können die Treiber in *n* Gästen ausgelagert werden. Abbildung 4.8 zeigt diese Architektur. Die Treiber-Gäste bzw. -Domains sharen die Geräte mit den anderen Gästen durch den VMM [KS08]. Hierdurch soll eine Isolation des Treibercodes vom restlichen System erreicht werden. Fehler der Treiber sollen auf die jeweiligen Domains beschränkt werden. Die Domains verfügen über spezielle Privilegien, über die andere Gäste nicht verfügen, da eine Domain direkt auf die Hardware zugreifen können muss. Eine derartige IO-Partition bzw. -Domain kann ebenfalls den DMA der Hardware programmieren. Jeder Domain mit direktem Hardwarezugriff muss daher vertraut werden, da sie durch DMA den gesamten Hauptspeicher überschreiben kann. Eine tatsächliche Security-Isolation der Treiber kann daher durch IO-Domains auf der x86-Architektur nicht erreicht werden. Dem Treiber bzw. der gesamten Domain ist weiterhin zu vertrauen, da der Treiber direkt in Kontakt mit sensitiven Daten der anderen Gäste kommt, die die Gäste in den jeweiligen IO-Komponenten aufbewahren wollen. Konzeptionell steigt die Sicherheit durch den Einsatz von privilegierten IO-Domains im Vergleich mit der Hosted-

Architektur daher nicht. Die TCB des VMM wird nicht kleiner, indem die Treiber einfach aus dem VMM entfernt werden. Die TCB des VMM nimmt alle Bestandteile der IO-Partionen auf. Eine IO-Domain beinhaltet i.d.R. ebenfalls ein komplettes, herkömmliches Betriebssystem, so dass bei *n* IO-Domains *n* Betriebssysteme in die TCB des VMM aufgenommen werden müssen. Daher hat der Einsatz von IO-Domains i.d.R. zur Folge, dass ggf. noch mehr Betriebssysteme in die TCB des VMM aufgenommen werden müssen als bei der Hosted-Architektur, obwohl der Kern des VMM schmaler wird. Der Informationsfluss vom Gast zur realen Hardware wird durch IO-Domains komplexer, da der Zugriff auf die Hardware stets den Umweg über IO-Domains bedingt. Der VMM leitet alle Zugriffe auf IO-Komponenten an die jeweilige IO-Domain weiter [KS08].

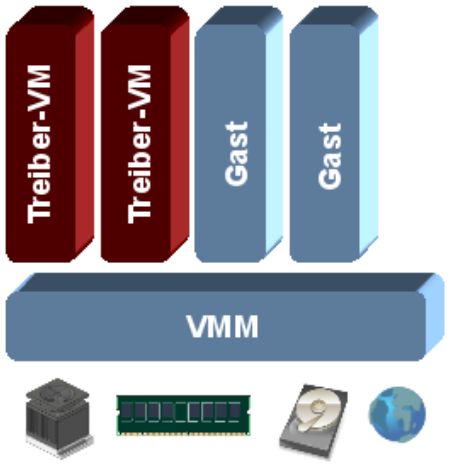

Abbildung 4.8: Treiber-Gäste

Die Sicherheit eines VMM kann durch die Hosted-Architektur und die Domain-Architektur das Maß der Sicherheit der Betriebssysteme, denen der VMM vertrauen muss, nicht überschreiten. Laut John Scott Robin et al., ist es unklug, einen sicheren VMM auf einem unsicheren Host-Betriebssystem aufsetzen zu wollen [RI00].

Um das Problem der schlecht konzipierten Gerätetreiber einzugrenzen und die Systemstabilität zu erhöhen, kann sich ein VMM auf bestimmte Hardwarekomponenten beschränken, die exklusiv vom VMM unterstützt werden. Hierdurch kann die Anzahl notwendiger Treiber auf einige zertifizierte Treiber reduziert werden, denen der VMM jedoch immernoch vertrauen muss. Die Flexibilität bei der Auswahl und Wiederverwendbarkeit bestehender Hardwarekomponenten geht hierdurch zwar verloren, aber die Größe der TCB des VMM wird geringer.

Ein VMM benötigt für den Zugriff auf die Hardware grundsätzlich kein komplettes Betriebssystem, sondern nur eine begrenzte Anzahl an Treibern [KS08]. Aus Perspektive der Sicherheit ist ein VMM vorzuziehen, der direkt auf der Hardware betrieben wird und eigene Treiber

verwendet. Hierdurch vermindert sich die Größe der TCB und die Sicherheit des VMM wird nicht durch das Maß an Sicherheit des Host-Betriebssystems beschränkt. Die nicht notwendigen Bestandteile eines herkömmlichen Betriebssystems werden nicht in die TCB eines VMM aufgenommen.

Um den DMA der Hardware zu beschränken kann eine IO Memory Management Unit (IOMMU) eingesetzt werden. Eine IOMMU ist eine zusätzliche Hardwarekomponente und erfüllt zwei Aufgaben bzgl. der Virtualisierung. Eine IOMMU führt eine zusätzliche Indirektionsebene bzgl. der Hardwareadressen ein. Die DMA- bzw. virtuellen Geräte-Adressen werden von der IOMMU auf physische Adressen abgebildet [BYMX+06]. Eine IOMMU erfüllt daher die gleiche Aufgabe bzgl. der DMA-Adressen der IO, wie die MMU bzgl. des virtuellen Speichers. Es werden unabhängige Adressbereiche erzeugt, durch die der DMA von IO-Komponenten auf Speicherbereiche beschränkt wird, die den IO-Komponenten zugewiesen werden. Die Geräte können hierdurch nur noch transparent in vordefinierte Speicherbereiche schreiben. Die zweite Aufgabe ist, den DMA auf andere Speicherbereiche nicht zu gestatten, um andere Speicherbereiche und damit Gäste zu schützen. Der VMM kann durch den Einsatz einer IOMMU den Zugriff der Hardware bzw. der Treiber auf den Speicher kontrollieren. Die Treiber können daher, wie bei IO-Domains, in Gästen betrieben werden, so dass die Treiber vom restlichen System isoliert werden. Die Security-Isolation ist jedoch deutlich höher als bei einfachen IO-Domains, da die Treiber nicht in beliebige Speicherbereiche schreiben können. Die Isolation wird durch die IOMMU garantiert. Ein Gast kann den Zugriff auf IO-Komponenten implementieren ohne andere Gäste oder den VMM beeinträchtigen zu können [PvDS08], indem der VMM sicherstellt, dass DMA nur in Speicherbereiche des jeweiligen Gasts zugelassen wird. Die IO-Komponenten werden Gästen offengelegt, denen der VMM nicht länger vertrauen muss. Hierdurch muss der VMM auch den Treibern für Hardwarekomponenten, auf die der VMM nicht selbst angewiesen ist, nicht mehr länger vertrauen. Die TCB des VMM wird wesentlich schmaler und einfacher. Ein Fehler in den Treibern, denen der VMM nicht vertraut, hat nicht mehr zur Folge, dass das komplette System, also auch der VMM, kompromittiert werden kann.

Erst der Einsatz einer IOMMU macht es möglich, einen Pure-Isolation-VMM für die x86-Architektur zu konstruieren. Jedem Gast können exklusiv Hardwarekomponenten zugewiesen werden, die auch mit direktem Hardwarezugriff (DMA) die Integrität anderer Gäste nicht gefährden können. Hierdurch erreicht der Pure-Isolation-VMM ein ähnlich hohes Maß an Isolation wie es durch vollständig dedizierte Hardware möglich ist. Nichtsdestotrotz wird ein Pure-Isolation-VMM i.d.R. nicht gewüscht, da die Skalierbarkeit und Flexibilität eines Pure-Isolation-VMM nicht sehr hoch ist. Für jeden Gast müssen dedizierte Hardwarekomponenten verbaut werden.

Zur Zeit sind IOMMUs in physischen Maschinen noch nicht die Regel, um die Virtualisierung zu unterstützen. Die Hersteller von Hardwaretechnologien zur Unterstützung der Virtuali-

sierung bieten jedoch IOMMUs an [AMD07] [Int08]. Hierdurch ist davon auszugehen, dass IOMMUs in der Zukunft die Regel werden [BYXO+07].

4.3.4 Netzwerk-Virtualisierung

Die Netzwerkanbindung von Gästen wird als eigener Punkt, neben der IO-Virtualisierung, analysiert, da die Netzwerkanbindung von enormer Bedeutung für den Betrieb von Gästen ist und sich die Virtualisierung der Netzwerkanbindung stark von der reinen IO-Virtualisierung unterscheidet. Bei einer Serverkonsolidierung sind Server ohne Netzwerkanbindung bedeutungslos. Die Sicherheit der Netzwerkanbindung hat daher hohen Stellenwert.

Ohne Virtualisierung muss ein Netzwerkpaket einen physischen Host verlassen, damit der Host mit einem beliebigen anderen Host kommunizieren kann. Physische Netzwerkkomponenten, wie Switches, vermitteln dabei die Kommunikation. Virtuelle Rechner benötigen eine virtuelle Netzwerkanbindung. Bei der Kommunikation in einer virtualisierten Umgebung zwischen zwei Gästen, die auf dem gleichen, physischen Host lokalisiert sind, müssen die Pakete den physischen Host nicht zwangsläufig verlassen. Hierzu wird, neben dem virtuellen Network Interface Controller (NIC), eine zusätzliche, virtuelle Netzwerkkomponente unterhalb der Gäste implementiert, wie in Abbildung 4.9 gezeigt wird.

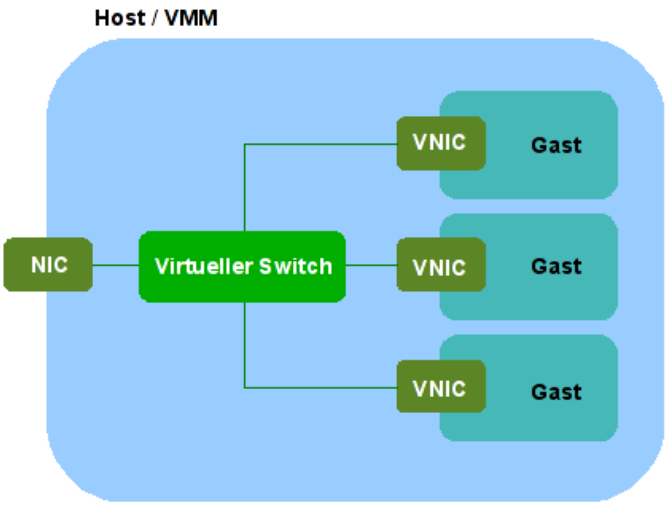

Abbildung 4.9: Virtual-Switch

Die virtuelle Netzwerkkomponente verbindet die Gäste eines Hosts und das physische Netz, an das der Host angeschlossen ist, miteinander. Bei einer Kommunikation zwischen zwei

Gästen kann die virtuelle Netzwerkkomponente das Netzwerkpaket direkt an den entsprechenden Gast weiterleiten, anstatt das Paket in das physische Netzwerk zu leiten. Die virtuelle Netwerkkomponente ist üblichen, physischen Netzwerkkomponenten nachempfunden. Die Komponente kann als Switch oder als Hub implementiert werden. Ohne Virtualisierung werden heute i.d.R. Switches für die Kommunikation im lokalen Netzwerk verwendet. Wird die virtuelle Netzwerkkomponente in einem VMM als Hub implementiert, wird es für einen Angreifer einfacher, den Netzwerkverkehr abzuhören (sniffen), da der Hub jedes Paket per Broadcast an jeden Gast leitet. Ein schadhafter Gast kann hierdurch, mit geringem Aufwand, den gesamten Netzwerkverkehr der anderen Gäste auf diesem Host sniffen [Wol07]. Der Angreifer muss die Netzwerkkomponenten nicht erst dazu bringen die Pakete an den Host zu leiten, zu dem der Angreifer Zugang besitzt.

Ein Unternehmensnetzwerk ist üblicherweise in verschiedene Segmente aufgeteilt, die physisch voneinander isoliert sind, wie auch in Abbildung 2.1 zu sehen ist. Die virtuellen Netzwerkkomponenten bieten üblicherweise in einem VMM zusätzlich Routing-Funktionalität an, wodurch mehrere, beliebige Netzwerk-Segmente von Gästen innerhalb eines Hosts betrieben werden können. Bei einer Serverkonsolidierung verändert sich hierdurch die physische Netzwerktopologie, was exemplarisch für KM in Abbildung 4.10 gezeigt wird. Gäste, die in unterschiedlichen Segmenten betrieben werden, können oberhalb des gleichen VMM existieren, wodurch die Isolation der Segmente nicht länger auf gleichem Niveau gewährleistet ist. Die Nutzung gemeinsamer Hardware bedingt, dass die Daten physisch durch die gleichen Netze fließen [Gei07], so dass sich die Daten unterschiedlicher Netze in der gleichen Hardware befinden. Ein Ausbruch aus einem Gast ermöglicht unter diesen Bedingungen die Kompromittierung über Netzsegmente hinaus. Hierdurch wird es möglich, dass Malware oder ein Angreifer, der einen einzigen Host, den VMM, kompromittiert, zusätzlich ganze Netzwerke und DMZ kompromittiert.

Die virtuellen Netzwerkkomponenten müssen nicht Bestandteil des VMM sein, da die Komponenten nicht direkt mit der Hardware kommunizieren müssen, sondern über den VMM auf die Harware zugreifen können, so dass die Komplexität des VMM-Kerns nicht steigt. Die virtuelle Netzwerkkomponente kann jedoch, abhängig von der bereitgestellten Funktionalität, sehr komplex werden, da ggf. Routing und Switching oberhalb des VMM integriert wird. Die virtuellen Netzwerkkomponenten gehören zur TCB eines jeden Gasts des Hosts. Die Gäste benutzen einen gemeinsamen Netzwerk-Stack, was dem Prinzip der minimalen Gemeinsamkeiten widerspricht.

Der Netzwerkverkehr wird ohne Virtualisierung i.d.R. durch physische Netzwerkkomponenten analysiert und kontrolliert. Da ein Netzwerkpaket bei der Virtualisierung ggf. den Host nicht mehr verlässt, kann der Netzwerkverkehr nicht durch die physischen Schutzmechanismen auf Netzebene kontrolliert werden. Der Netzwerkverkehr zwischen Gästen des gleichen Hosts wird für die herkömmlichen Mechanismen der Netzwerksicherheit, wie die Firewall in

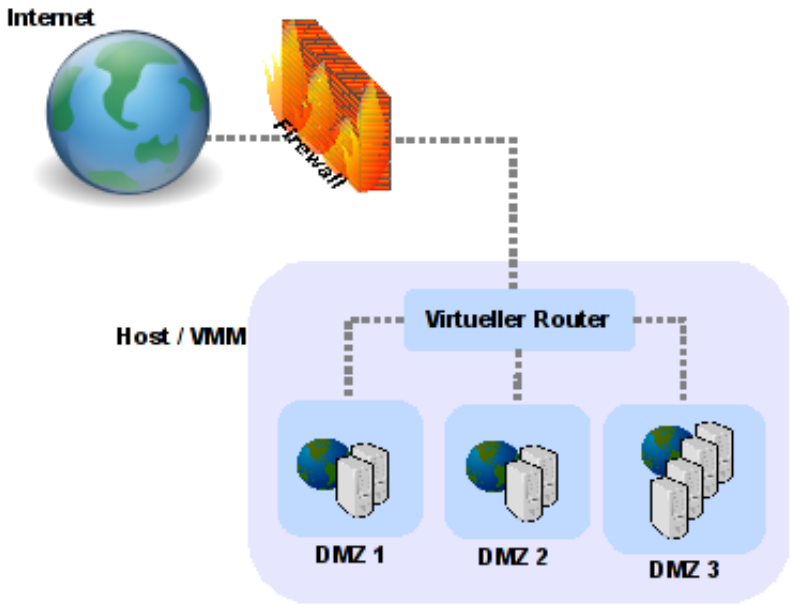

Abbildung 4.10: Mehrere logische DMZ innerhalb eines Hosts

Abbildung 4.10, unsichtbar [VN08][6]. Es bilden sich Inseln, die sich nicht transparent in ein Netzwerk einfügen.

Ein schadhafter Gast kann andere Gäste auf dem gleichen Host angreifen, was als Inter-VM-Angriff bezeichnet wird. Herkömmliche Schutzmechanismen auf Netzebene können hierbei, aufgrund der Unsichtbarkeit des Netzwerkverkehrs, nicht greifen. Durch Virtualisierung wird es in diesem Fall einfacher einen Gast zu kompromittieren, wenn ein anderer Gast des gleichen Hosts bereits kompromittiert wurde. Virtualisierung verringert daher in diesem Fall die Security Isolation aus Kapitel 4.2.1.

Da der Netzwerkverkehr unsichtbar wird, sind auch Tools, die einen Administrator den Netzwerkverkehr bspw. innerhalb eines Routers analysieren lassen, unbrauchbar. Neue Tools müssen erst entwickelt werden und sich etablieren. Eine Fehlersuche auf Netzebene ist hierdurch ggf. schwieriger durchführbar, was die Verfügbarkeit beeinträchtigen kann und den Netzwerkverkehr auch bzgl. der Wartung intransparent macht, da die Analyse ein falsches Bild des Netzwerkverkehrs wiedergibt. Der unsichtbare Netzwerkverkehr verhält sich konträr zu den Erwartungen an die Virtualisierungstechnologie. Es wird davon ausgegangen, dass sich eine virtuelle Maschine wie eine reale Maschine in das Netzwerk einfügt, was nicht der Fall ist.

[6] Hierzu zählen bspw. Firewalls oder Intrusion Detection- (IDS) und Prevention-Systeme (IPS)

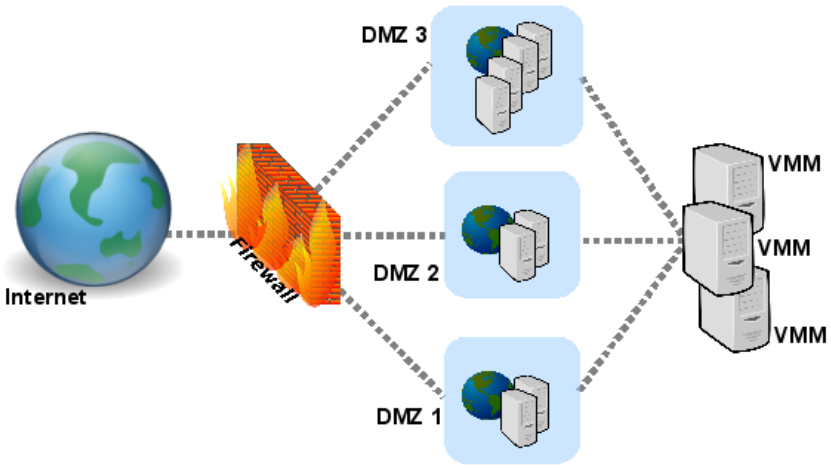

Abbildung 4.11: Konsolidierung von DMZ

Um herkömmliche Schutzmechanismen auf Netzebene auch für virtuelle Netzwerke anwenden zu können, müssen die physischen Netzkomponenten in Software virtuell neu implementiert werden. Die virtuellen Komponenten der Netzwerksicherheit werden hierbei zwischen dem VMM und den Gästen lokalisiert. Die Virtualisierung beschränkt sich hierdurch jedoch nicht auf die wesentlichen, funktionalen Aspekte der Virtualisierung. Die Ebenen der Virtualisierung beinhalten Paketfilter und andere Mechanismen um die Netzwerksicherheit zu bewahren. Die Komplexität der Virtualisierungsebenen steigt und die TCB eines jeden Gasts wächst. Die physischen Schutzmechanismen haben sich über lange Zeit bewährt, während die virtuellen Komponenten noch jung und ggf. nicht ausgereift sind. Das Risiko eines Sicherheitsvorfalls wird erhöht.

Werden verschiedene Netzwerke im Zuge einer Serverkonsolidierung innerhalb eines VMM-Hosts konsolidiert, muss ein VMM an jedes Netz angeschlossen werden, aus denen Gäste innerhalb eines VMM konsolidiert werden sollen, um die Netztopologie aufrecht zu erhalten. Dies wird in Abbildung 4.11 für KM gezeigt. Die Isolation der Netze wird hierdurch vermindert, da neue Kommunikationswege in die Netzwerke geschaffen werden, die an den physischen Netzkomponenten, wie Firewalls, vorbei führen. Verschiedene Netze kommunizieren nicht mehr ausschließlich über die vorhandenen physischen Netzkomponenten, die die Kommunikation kontrollieren. Um die Kontrollmechanismen aufrecht zu erhalten muss der VMM-Layer auch in diesem Fall virtuelle Netzkomponenten aufnehmen, allerdings als Kopie der physischen Netzkomponenten. Es muss ein identisches Abbild einer Netzkomponente, wie bspw. einer Firewall, innerhalb des VMM-Layers konstruiert werden, um die Isolation der Netze auch nach der Serverkonsolidierung auf ähnlichem Niveau zu gewährleisten. Die Komplexität der Inbetriebnahme und Wartung eines VMM steigt diesbzgl. stark an, da jede Änderung in der Konfiguration der physischen Netzkomponenten eine Änderung der Konfi-

guration der virtuellen Kopien notwendig macht. Auch wenn die virtuellen Netzkomponenten aufgenommen werden um den Netzwerkverkehr zu kontrollieren, existieren dennoch mehr Zugriffspfade, die in sichereren Systemen jedoch zu minimieren sind [WSG02].

Um eine Vergrößerung der TCB zu verhindern und die vorhandenen physischen Komponenten der Netzwerksicherheit weiterhin zu verwenden, kann der gesamte Traffic der Gäste durch die herkömmlichen physischen Netzwerkkomponenten hindurch geleitet werden. Das Rerouting des Netzwerkverkehrs erhöht jedoch auch die Komplexität des Administrationsaufwands, sofern dieses Verhalten nicht das Standardverhalten des Virtualisierungsproduktes ist. Routing-Regeln müssen für jeden Gast konfiguriert werden, was mit zunehmender Anzahl von Gästen schwieriger und somit fehleranfälliger wird [VN08]. Außerdem geht die physische Isolation von Netzen auch bei einem Rerouting verloren, da verschiedene Netze auf einem Host konsolidiert werden können und nur durch den VMM isoliert werden.

4.4 Angriffe

Die Bedrohungsanalyse hat in den letzten Kapiteln konzeptionelle Bedrohungen der Virtualisierung analysiert. Die Komplexität der Virtualisierungstechnologien wurde als Kriterium für die Sicherheit in virtualisierten Umgebungen belegt. Die Komplexität der Virtualisierungstechnologie hat Einfluss auf die Wahrscheinlichkeit mit der Sicherheitslücken existieren [Orm07]. Die Isolation zwischen dem VMM und Gästen oder mehereren Gästen untereinander kann durch diese Sicherheitslücken beeinträchtigt werden. Ein Angriff der speziell eine virtualisierte Umgebung fokussiert ist ein Angriff auf die logische Isolation. Um die Konsequenzen für das fiktive Unternehmen KM evaluieren zu können, muss das Risiko der jeweiligen Bedrohungen abgeschätzt werden. Von besonderem Interesse sind dabei Bedrohungen, die ein hohes Risiko zur Folge haben, da die Schutzziele eines Unternehmens besonders durch Bedrohungen mit hohem Risiko gefährdet werden. Bedrohungen, die ein hohes Risiko haben, besitzen eine hohe Eintrittswahrscheinlichkeit und/oder ein hohes Schadensausmaß.

Automatisierte Angriffe sind einfach durchführbar und werden ggf. im Internet, bspw. durch Hacker, öffentlich verfügbar gemacht. Automatisierte Angriffe basieren auf Sicherheitslücken und deren Exploits. Ein Exploit besteht aus Angriffscode, der eine Sicherheitslücke ausnutzt, um bspw. privilegierten Zugriff auf ein System zu erlangen oder das System zum Absturz zu bringen (vgl. [Eck08]). Falls vollständig automatisierte Angriffe noch nicht existieren, kann sich ein Angreifer die Exploits und Tools, die hierzu notwendig sind und ebenfalls im Internet i.d.R. frei verfügbar sind, leicht beschaffen. Die Sicherheitslücken werden bei diesen Angriffen automatisch ausgenutzt, um bspw. viele Ziele in kurzer Zeit anzugreifen. Mit wenig Aufwand und geringem Know-How können hierdurch schützenswerte Güter, durch bspw.

Script-Kiddies, angegriffen werden [Eck08]. Die Eintrittswahrscheinlichkeit für Sicherheitsvorfälle durch automatisierte Angriffe ist daher hoch einzustufen.

Sicherheitslücken resultieren aus dem Design, der Implementation, der Instandhaltung oder dem Betrieb von Computer-Systemen [Bis02]. Diese Arbeit untersucht konzeptionelle Aspekte der Sicherheit in virtualisierten Betriebssystemumgebungen. Daher sind hier Sicherheitslücken und Angriffe, die aus den konzeptionellen Bedrohungen und dem Design von Virtualisierungstechnologien resultieren, von Interesse.

Um zu zeigen, dass aus den konzeptionellen Bedrohungen Angriffsvektoren resultieren und welche Bedrohungen für eine Serverkonsolidierung von KM besonders zu berücksichtigen sind, werden in Tabelle 4.1 die automatisierten oder automatisierbaren Angriffe belegt, die aus aus den konzeptionellen Bedrohungen der Virtualisierung resultieren. Dabei wird exemplarisch Bezug zu Common Vulnerabilities and Exposures (CVE) [MIT09b] genommen, um konkrekte Schwachstellen in die konzeptionellen Bedrohungen einzuordnen. Alle in dieser Arbeit genannten CVE-Bezeichner können unter [MIT09b] eingesehen werden. CVEs sind allgemein anerkannte, eindeutige Bezeichner für öffentlich bekannte Sicherheitslücken [MIT09a]. Die CVEs zeigen Sicherheitslücken, die bereits in gängigen Virtualisierungsplattformen entdeckt und veröffentlicht wurden. Diese Sicherheitslücken kommen in einer virtualisierten Umgebung neu hinzu. Auf Basis der in CVEs veröffentlichten Sicherheitslücken können mit geringem Aufwand automatisierte Angriffe konstruiert werden. Die konkreten Angriffe und Sicherheitslücken sind ggf. nicht aktuell, so dass Sicherheitspatches für diese Sicherheitslücken ggf. existeren, aber die Sicherheitslücken belegen exemplarisch die Relevanz der konzeptionellen Bedrohungen die von der Virtualisierung ausgehen und in der Bedrohungsanalyse nachgewiesen wurden. Hierdurch soll gezeigt werden, dass die Eintrittswahrscheinlichkeit und das Schadensausmaß von Angriffen potentiell in virtualisierten Umgebungen in einem beliebigen Szenario der Serverkonsolidierung steigt.

Wie die Eintrittswahrscheinlichkeit und das Schadensausmaß konkret zu bemessen ist, wird in Kapitel 5, der Evaluation, im Gesamtkontext des Szenarios gezeigt, da bspw. die Eintrittswahrscheinlichkeit eines spezifischen Angriffs auch von spezifischen Vorbedingungen abhängig ist.

Weiteres Interesse wird durch bereits bekannte Angriffe erweckt, die in virtualisierten Umgebungen höhere Eintrittswahrscheinlichkeiten besitzen, so dass das Risiko dieser Angriffe steigt. Tabelle 4.2 belegt exemplarisch drei derartige Angriffe.

Angriffe, die viel Schaden verursachen, gefährden Schutzziele in besonderem Maße. Daher sind, neben Bedrohungen, die eine hohe Eintrittswahrscheinlichkeit besitzen, Bedrohungen, die in virtualisierten Umgebungen ein höheres Schadensausmaß besitzen, besonders zu berücksichtigen. Diese Angriffe werden in Tabelle 4.3 belegt.

Tabelle 4.1: Durch den Einsatz von Virtualisierung hinzukommende, automatisierte und automatisierbare Angriffe mit potentiell hoher Eintrittswahrscheinlichkeit

konzeptionelle Bedrohung	konkrete Bedrohung	CVE	Erläuterung
Unzureichende Isolation u.a.	Ausbruch	CVE-2007-4993 CVE-2008-0923 CVE-2007-1744 CVE-2008-1943	Exploits für diese Sicherheitslücken sind dokumentiert und automatisiert bzw. automatisierbar [Tec08], [Woj08]. Einem Angreifer ist es möglich, aus der virtuellen Umgebung auszubrechen und Code innerhalb des Host-OS auszuführen.
Komplexität der Binary-Translation	Privilege Escalation	CVE-2008-4915 CVE-2008-4279	Ein Angreifer kann seine Privilegien in einem Gast ausweiten.
Komplexität der Hardware-Emulation	Typische Sicherheitslücken	CVE-2009-0177 CVE-2008-4279 CVE-2008-1952 CVE-2007-6416 CVE-2007-1320	[Orm07] Travis Ormandy belegt Sicherheitslücken, die aus der Komplexität der Hardware-Emulation resultieren. Würde ein VMM auf Hardware-Emulation verzichten, wären diese Sicherheitslücken nicht existent.
Unzureichende Isolation	Thrashing		In der Literatur wurden keine dokumentierten Angriffe gefunden, die auf Thrashing-Verhalten des VMM abzielen. Um die praktische Relevanz des Thrashings zu zeigen wurde daher ein Angriff exemplarisch für die Kernel-based Virtual Machine (KVM) entwickelt. Der Angriff ist in Anhang A.1 dokumentiert. Der Angriff zeigt, dass Thrashing-Verhalten durch unvertrauenswürdige Gäste hervorgerufen werden kann. Eine Performance Isolation wird hierdurch zerstört.

Tabelle 4.2: Bereits bestehende Angriffe und Bedrohungen mit potentiell höherer Eintrittswahrscheinlichkeit als in nicht-virtualisierten Umgebungen

konzeptionelle Bedrohung	konkrete Bedrohung	CVE	Erläuterung
Unzureichende Isolation u.a.	DoS	CVE-2008-4914 CVE-2008-1340 CVE-2007-2491 CVE-2007-1337 CVE-2008-1952 CVE-2008-1944 CVE-2008-1943 CVE-2007-5907 CVE-2007-5906	Denial of Service Angriffe (DoS) haben höhere Erfolgsaussichten in virtualisierten Umgebungen. Durch erhöhtes Sharing von Hardwarekomponenten, hohe Komplexität des VMM und erhöhte Auslastung der Hardware steigt die Wahrscheinlichkeit für einen erfolgreichen DoS-Angriff. Zahlreiche CVEs belegen das Risiko von DoS-Angriffen in virtualisierten Umgebungen.
Komplexität des VMM	Typische Sicherheitslücken & Angriffe		Sofern der VMM auf einem herkömmlichen Betriebssystem aufsetzt, gelten alle Sicherheitsprobleme und -lücken auch für den VMM. Automatisierte Angriffe sind zahlreich vorhanden.
Verminderte Isolation durch virtuelle Netzwerkkomponenten	Sniffing		[Wol07] Chris Wolf zeigt, dass es ggf. einfacher möglich ist, sensitive Daten von Gästen auf dem gleichen Host im virtuellen Netzwerk abzuhören. In diesem Fall ist die virtuelle Netzwerkkomponente, die die Gäste an das physische Netzwerk anbindet, als Hub implementiert. Spezialisierte Tools für das Abhören des Netzwerkverkehrs sind frei verfügbar und hierfür einsetzbar. Das Sniffing greift die Vertraulichkeit von Informationen an.

Tabelle 4.3: Angriffe und Bedrohungen mit potentiell hohem oder höherem Schadensausmaß durch den Einsatz von Virtualisierung

konzeptionelle Bedrohung	konkrete Bedrohung	CVE	Erläuterung
Unzureichende Isolation	DoS	CVE-2007-5906 CVE-2007-5498 CVE-2008-4914 CVE-2008-1340	DoS-Angriffe gegen den VMM bzw. das Host-OS haben ein potentiell hohes Schadensausmaß, da die Verfügbarkeit aller Gäste des Hosts beeinträchtigt wird, wenn die Verfügbarkeit des VMM oder des Host-OS beeinträchtigt wird. Durch unzureichende Isolation kann ein DoS-Angriff oder andere Angriffe auf einen Gast ggf. die Verfügbarkeit anderer Gäste beeinträchtigen.
Verminderte Isolation	technisches Versagen		Die für DoS-Angriffe genannten Kriterien gelten auch für ein technisches Versagen.
Unwirksamkeit herkömmlicher Schutzmaßnahmen	Inter-VM-Angriffe		Aufgrund der Unsichtbarkeit des Netzwerkverkehrs zwischen Gästen des gleichen Hosts, ist das Schadensausmaß eines erfolgreichen Angriffs ggf. größer, da unbemerkt weitere Gäste angegriffen werden können.
Komplexität des VMM	Typische Sicherheitslücken & Angriffe	S. Tabelle 4.1	S. Tabelle 4.1. Ein Einbruch in ein Host-OS gefährdet Schutzziele, die für Gäste gelten. Das Schadensausmaß ist potentiell hoch, da die Kompromittierung eines Hostes potentiell viele Gäste gefährdet.
Unzureichende Isolation	Thrashing		Thrashing-Verhalten eines Gasts gefährdet die Verfügbarkeit aller Gäste, die auf dem gemeinsamen Host betrieben werden.

In virtualisierten Umgebungen sind die handelnden Subjekte, die Gäste, nur durch logische Isolation voneinander abgeschottet. Eine Verletzung der logischen Isolation in virtualisierten Umgebungen hat daher ein potentiell grundsätzlich höheres Schadensausmaß als in herkömmlichen Umgebungen, da eine physische Isolation nicht existiert und das Schadensausmaß beschränken kann [1].

Aus den gezeigten Tabellen wird in der Evaluation die jeweilige Eintrittswahrscheinlichkeit, das Schadensausmaß und das Risiko von Bedrohungen und deren Vorbedingungen für das fiktive Unternehmen KM bestimmt.

[1] Von Pure-Isolation-VMMs sei hier abgesehen.

4.5 Zusammenfassung

Der Einsatz von virtualisierten Umgebungen kann die Isolation steigern, aber auch verringern. Bei einer Serverkonsolidierung wird die Isolation jedoch ausschließlich verringert, wodurch die konsolidierten Systeme insbesondere den konzeptionellen Bedrohungen der Virtualisierung unterliegen.

Virtualisierte Umgebungen bereiten eine Reihe organisatorischer Probleme bzgl. der Sicherheit, da der Umgang mit virtualisierten Umgebungen nicht mit dem Umgang herkömmlicher Umgebungen übereinstimmt. Virtualisierung bringt daher eine Reihe grundsätzlicher Bedrohungen mit sich, die unabhängig von bestimmten Technologien sind. Virtualisierte Umgebungen besitzen ein hohes Maß an Dynamik, Sicherheitsstrategien sind jedoch eher statisch. Durch Virtualisierung werden einige, herkömmliche Sicherheitsmechanismen, bspw. auf Netzwerkebene, wirkungslos.

Die x86-Architektur ist nicht entwickelt worden, um virtualisierte Umgebungen zu betreiben. Die x86-Architektur beinhaltet daher Stolpersteine bzgl. des Designs von Virtualisierungstechnologien die Sicherheit betreffend. Die Stolpersteine werden durch kreative Methoden umgangen. Diese kreativen Methoden haben sehr unterschiedlich hohe Komplexität der Virtualisierungstechnologie zur Folge.

Hohe Komplexität von IT-Systemen ist schädlich für die Sicherheit der Systeme. Hohe Komplexität eines VMM verstößt gegen die grundlegende Annahme, dass ein VMM besser als ein herkömmliches Betriebssystem für die Isolation in Informationssystemen geeignet sei. Hohe Komplexität bewirkt, dass die konzeptionellen Bedrohungen der Virtualisierung mit höherer Wahrscheinlichkeit in konkreten Sicherheitslücken resultieren. Die Eintrittswahrscheinlichkeit und das Schadensausmaß von Angriffen gegen virtualisierte Umgebungen steigt potentiell aufgrund der konzeptionellen Bedrohungen. Sehr komplexe Systeme sind nur unter hohem Aufwand oder gar nicht verifizierbar, wodurch das Vertrauen in diese Systeme geschmählert wird. Ein sehr komplexer VMM ist daher nicht in sehr hohem Maße vertrauenswürdig.

5 Evaluation

5.1 Vorgehen

In diesem Kapitel werden die Bedrohungen, die in der Bedrohungsanalyse identifiziert wurden, für das fiktive Unternehmen KM evaluiert. Es wird untersucht, wie sich das Risiko für die Schutzziele des Unternehmens durch den Einsatz virtualisierter Umgebungen verändert. Hierzu wird ein, für KM spezifisches, Konsolidierungsszenario betrachtet. Die Bedrohungen werden hierbei im Kontext, d.h. samt Vorbedingungen, in Bedrohungsbäumen betrachtet, um die spezifische Eintrittswahrscheinlichkeit einer Bedrohung und damit das Risiko bemessen zu können. Aufbauend auf der Bedrohungs- und Risikoanalyse werden zu treffende, konzeptionelle Maßnahmen für das Unternehmen identifiziert, um das Risiko virtualisierter Umgebungen für KM zu reduzieren. Hierdurch soll es für KM möglich werden, eine Serverkonsolidierung mithilfe virtualisierter Umgebungen unter Sicherheitsaspekten durchführen zu können.

5.2 Konsolidierung

KM strebt ein möglichst hohes Maß an Flexibilität und Konsolidierungspotential durch Virtualisierung an (vgl. Kapitel 2.2). Das größte Konsolidierungspotential ist gegeben, wenn jeder Gast auf jedem verfügbaren VMM betrieben werden kann und darf. Die Hardware, die zur Virtualisierung eingesetzt wird, kann in diesem Fall effizient und flexibel unter den Gästen aufgeteilt werden. Da jeder Gast potentiell auf jedem VMM betrieben werden kann, wird die Konsolidierung von Servern nicht beschränkt. Daher wäre es konzeptionell möglich, dass alle Gäste auf einem einzigen VMM betrieben werden, wenn von der Performance und Beschränkheit von Ressourcen abstrahiert wird.

Um jeden Gast, zu einem beliebigen Zeitpunkt, auf jedem VMM betreiben zu können, muss bei KM jeder VMM an jede der drei DMZ aus Abbildung 2.1 angebunden werden, um die Konfiguration der Gäste auch bei hoher Mobilität von Gästen beibehalten zu können. Um die Risikoanalyse zu vereinfachen wird davon ausgegangen, dass die verschiedenen VMMs konzeptionell gleich konfiguriert sind und über die gleiche Hardware verfügen, da ein Gast

sonst ggf. angepasst werden muss, um auf einem bestimmten VMM betrieben werden zu können. Um bspw. eine CPU-unterstützte Virtualisierung anbieten zu können, muss jeder VMM über Prozessoren verfügen, die eine CPU-unterstützte Virtualisierung ermöglichen. Hier kann jeder VMM jede Virtualisierungstechnologie der x86-Architektur einsetzen, d.h. CPU-unterstützte Virtualisierung, vollständige Virtualisierung ohne CPU-Unterstützung und Paravirtualisierung. Eine beliebiger Gast kann auf einem beliebigen VMM, aus einem Pool von VMMs, betrieben werden.

Die Topologie der von KM betriebenen Netze gleicht in diesem Fall der Abbildung 5.1. Die Risikoanalyse findet anhand dieser Topologie bzw. dieses Konsolidierungsszenarios statt.

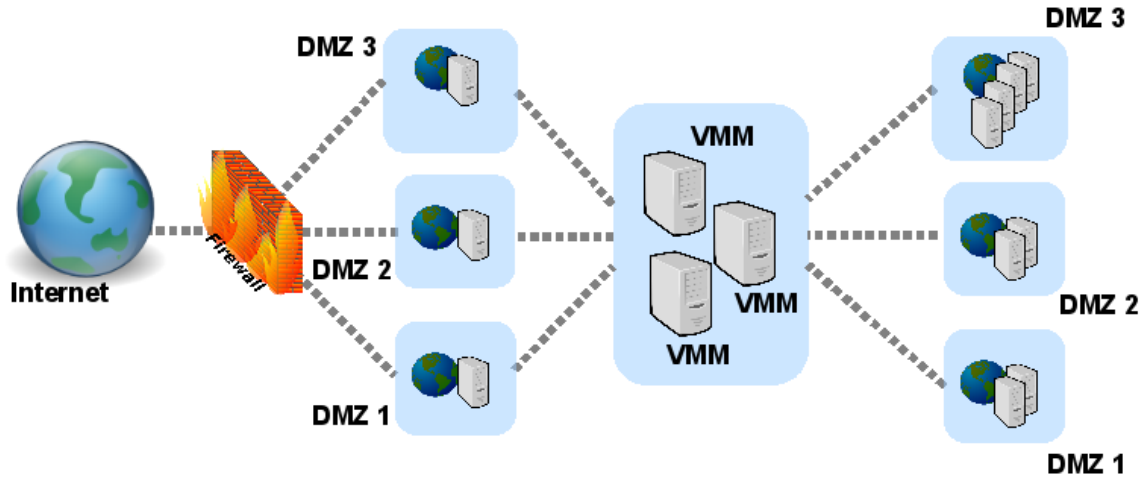

Abbildung 5.1: Konsolidierung

5.3 Risikoanalyse

Um das Risiko zu bemessen, werden die Eintrittswahrscheinlichkeiten der notwendigen Angriffsschritte bewertet, die eine Bedrohung eintreten lassen. Zusammen mit dem Schadensausmaß der Bedrohungen lässt sich das Risiko abschätzen [Eck08]. Auf eine numerische Berechnung des Risikos wird hier verzichtet, da das Risiko tendenziell bewertet wird und eine numerische Bewertung üblichweise schwer korrekt zu treffen ist.

Die Risikoanalyse betrachtet die schützenswerten Güter des höchsten Schutzbedarfs des Unternehmens. Den höchsten Schutzbedarf hat, wie in Tabelle 2.1 gezeigt wurde, die Vertraulichkeit, Integrität und Verfügbarkeit personenbezogener Daten, sowie die Integrität und

Verfügbarkeit des Buchungssystems. Hierzu werden für die drei Schutzziele Bedrohungsbäume bzgl. der Systeme mit höchstem Schutzbedarf untersucht.

Der Bedrohungsbaum in Abbildung 5.2 zeigt, inwiefern die Vertraulichkeit für die personenbezogenen Daten von KM durch Angriffe, die in der Bedrohungsanalyse identifiziert wurden, bedroht werden. Es werden ausschließlich Bedrohungen, die durch die Virtualisierung hinzukommen oder ein höheres Risiko haben, betrachtet. Die Angriffsschritte sind mit Indizes versehen. Die Kanten des Bedrohungsbaums sind mit den Eintrittswahrscheinlichkeiten der Angriffsschritte attributiert. Die Attributierung befindet sich rechts, neben den Angriffsschritten. Die Eintrittswahrscheinlichkeit wird in den Kategorien „normal“, „hoch“ und „sehr hoch“ bemessen. Die Kategorien bemessen sich anhand des geschätzten Aufwands, den ein Angreifer für einen Angriffsschritt aufwenden muss, und des Nutzens, den er daraus ziehen kann (vgl. [Eck08]). Das Schadensausmaß wird in den gleichen Kategorienbezeichnungen angegeben und bemisst sich anhand des erwarteten Schadens für das Unternehmen. Es wird davon ausgegangen, dass hier für den VMM bzw. das Host-OS zunächst keine zusätzlichen Schutzmaßnahmen ergriffen werden.

Als Zugriff wird in allen Fällen ein nieder- bis hochprivilegierter Zugang zu dem jeweiligen System bezeichnet. Erhält ein Angreifer Zugriff auf das Datenbank-System, so dass personenbezogene Daten eingesehen werden können, geht die Vertraulichkeit der personenbezogenen Daten verloren.

Kann ein Angreifer den Netzwerkverkehr abhören, kann er ggf. personenbezogene Daten mitschneiden. Besonders bei einer unverschlüsselten Übertragung personenbezogener Daten kann der Angreifer diese Daten verwerten. Erhält ein Angreifer Zugang zu einem Gast, der an einen virtuellen Hub angeschlossen ist, ist der Aufwand für das Abhören des gesamten Netzwerkverkehrs gering und der Nutzen immens. Die Eintrittswahrscheinlichkeit des Angriffsschritts 2.1 wird daher, auf Basis von Kapitel 4.4, mit „hoch“ bemessen. Eine „sehr hohe“ Eintrittswahrscheinlichkeit wird nicht gewählt, da die höhere Eintrittswahrscheinlichkeit von der Implementation der virtuellen Netzwerkkomponente abhängt.

Aus Angriffsschritt 3 und 4 folgt die Möglichkeit des Diebstahls personenbezogener Daten, da sich personenbezogene Daten temporär, bspw. im Hauptspeicher, der in Schritt 3 und 4 genannten Systeme, befinden müssen, um Buchungen oder andere Aufgaben erfüllen zu können. Erlangt ein Angreifer Zugang zu diesen Systemen, erlangt er auch Zugang zu den temporär vorgehaltenen personenbezogenen Daten.

Die Eintrittswahrscheinlichkeit des Angriffsschritts 1.1.1.1 wird mit „sehr hoch“ bemessen, da grundsätzlich jeder Gast als Ziel für diesen Angriffsschritt in Frage kommt, d.h. auch Gäste deren Schutzbedarf und daher ggf. Schutzniveau nur „normal“ ist[1]. Zusätzliche, starke Sicherheitsmaßnahmen werden für diese Gäste i.d.R. nicht ergriffen. Ein Gast behält in einer

[1] Hierzu zählen bspw. die Standard-Dienste aus Tabelle 2.1.

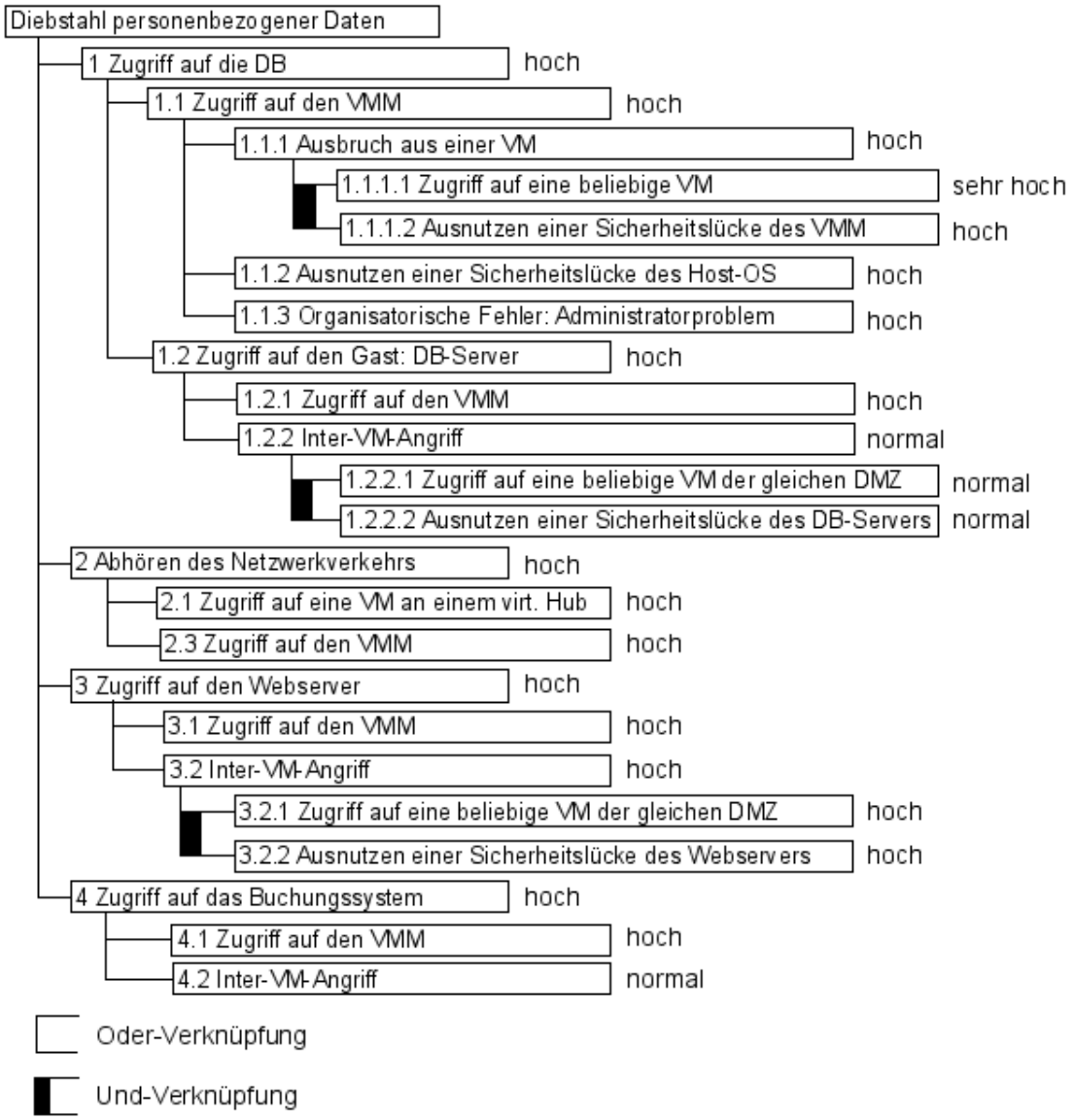

Abbildung 5.2: Diebstahl personenbezogener Daten

virtualisierten Umgebung grundsätzlich alle Sicherheitslücken, die auch in einer herkömmlichen Umgebung gelten, so dass automatisierte Angriffe, Insider- u.a. Angriffe weiterhin wirken und der Aufwand für diesen Angriffsschritt gering ist. Administratoren von niederschutzbedürftigen Gästen sind ggf. nicht in sehr hohem Maße vertrauenswürdig oder sicherheitsbewusst. Da die Anzahl der Dienste von KM hoch ist und die Systeme heterogen sind, ergibt sich die „sehr hohe“ Eintrittswahrscheinlichkeit für diesen Angriffsschritt.

Die Eintrittswahrscheinlichkeit des Angriffsschritts 1.1.1.2 wird mit „hoch“ bemessen, da in Kapitel 4.4 gezeigt wurde, dass Sicherheitslücken, die einen Ausbruch aus einem Gast erlauben, existieren und mit steigender Komplexität des VMM wahrscheinlicher sind. Der Nutzen, den ein Angreifer aus einem Ausbruch aus einem Gast ziehen kann, ist immens, da mit einem Zugriff auf den VMM weitreichende Privilegien verbunden sind. Im Falle von dokumentierten Sicherheitslücken ist der Aufwand für einen Ausbruch gering, wenn der Zugang zu einem beliebigen Gast erlangt wird.

Die Eintrittswahrscheinlichkeit des Angriffschritts 1.1.2 wird mit „hoch“ bemessen, da ggf. ein vollständiges, herkömmliches Betriebssystem für den Betrieb eines VMM notwendig ist. Der VMM unterliegt in diesem Fall allen Sicherheitslücken des Host-OS, weshalb der Aufwand für einen Angreifer gering ist. Der Nutzen, den ein Angreifer daraus ziehen kann, ist immens. Da ein VMM jedoch i.d.R. keine Netzwerkdienste bereitstellen muss, wird die Eintrittswahrscheinlichkeit für Angriffschritt 1.1.2 geringer eingeschätzt als die Eintrittswahrscheinlichkeit des Angriffsschritts 1.1.1.1.

Bei Angriffsschritt 1.1.3 handelt es sich um einen Insider-Angriff. Insider-Angriffe stellen die häufigste Art von Angriffen dar [Eck08]. Die Eintrittswahrscheinlickeit des Angriffsschritts 1.1.3 wird daher mit „hoch“ bemessen. In diesem Szenario ist jeder Administrator eines VMM bzw. Host-OS als Administrator aller Gäste anzusehen. Daher ist jeder Administrator bzgl. jedes Gasts als vertrauenswürdig einzustufen, was i.d.R. nicht möglich ist bzw. Insider-Angriffe wahrscheinlicher werden lässt. Der Aufwand für weiterführende Angriffsschritte ist gering, da eine derartige Person, bereits über ausgedehnte Privilegien im Host-OS verfügt. Die Eintrittswahrscheinlichkeit von Angriffsschritt 1.1.3 wird allerdings nicht mit „sehr hoch“ bemessen, da die Anzahl der verantwortlichen Personen für die IT-Infrastruktur bei KM, mit sechs Personen, zwar hoch, aber noch überschaubar ist.

Die Eintrittswahrscheinlichkeit des Angriffsschritts 1.2.2.2 wird mit „normal“ bemessen, da der Datenbankserver einen sehr hohen Schutzbedarf (vgl. Kapitel 2.4) und daher auch ein sehr hohes Schutzniveau hat, so dass die Eintrittswahrscheinlichkeit dieses Angriffsschritts geringer ist als die Eintrittswahrscheinlichkeit von Angriffsschritt 1.1.2 und 3.2.2.

Die Eintrittswahrscheinlichkeiten der Angriffsschritte 1.2.2.1 und 3.2.1 sind unterschiedlich, da der Webserver einen geringeren Schutzbedarf und und daher ein geringeres Schutzniveau hat, als der Datenbankserver.

Aus Abbildung 5.2 folgt, dass die Eintrittswahrscheinlichkeit für den Diebstahl personenbezogener Daten als „hoch“ einzustufen ist, da gezeigt wurde, dass die Eintrittswahrscheinlichkeit der Angriffschritte 1, 2, 3, und 4 „hoch“ ist. Das Schadensausmaß des Diebstahls personenbezogener Daten wird mit „sehr hoch“ bemessen, da die Kapitel 2.3 und 2.4 belegt haben, dass der Diebstahl jeglicher, personenbezogener Daten verheerende Auswirkungen auf die Aussenwirkung des Unternehmens hat, das außerdem für die Sicherheit der personenbezogenen Daten gesetzlich verantwortlich ist. Das Risiko für einen Diebstahl personenbezogener Daten in der virtualisierten Umgebung dieses Szenarios muss daher als „hoch“ bis „sehr hoch“ eingestuft werden.

Das Risiko für einen Verlust der Verfügbarkeit personenbezogener Daten wird in dem Bedrohungsbaum der Abbildung 5.3 untersucht. Dabei handelt es sich um den Verlust der Verfügbarkeit des Datenbankservers, der die personenbezogenen Daten für andere Systeme vorhält.

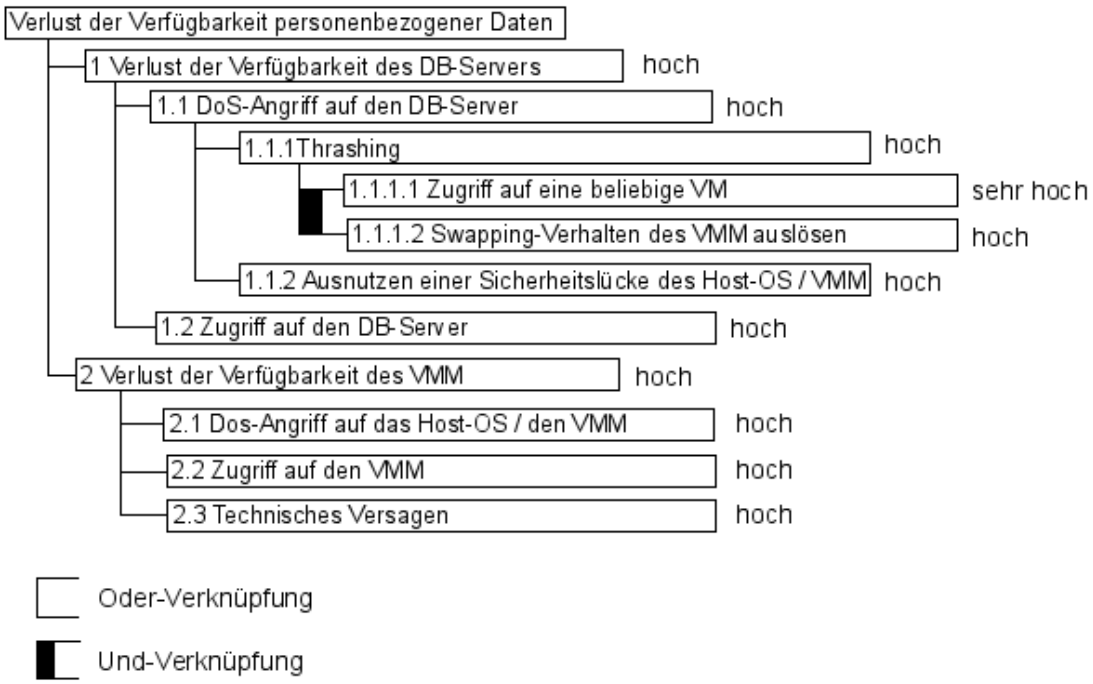

Abbildung 5.3: Verlust der Verfügbarkeit personenbezogener Daten

Die Verfügbarkeit der personenbezogenen Daten kann in der virtualisierten Umgebung auf zwei Wegen angegriffen bzw. beeinträchtigt werden. Wird die Verfügbarkeit des Datenbankservers beeinträchtigt, geht die Verfügbarkeit der personenbezogenen Daten verloren. Wird

die Verfügbarkeit des zugehörigen VMM beeinträchtigt, wird ebenfalls, mindestens die Verfügbarkeit der personenbezogenen Daten beeinträchtigt.

Die Eintrittswahrscheinlichkeiten der Angriffsschritte 1.1.1.1, 1.2 und 2.2 wurden bereits argumentiert.

Bei Angriffsschritt 1.1.2 handelt es sich um eine Sicherheitslücke, die einen automatisierten DoS-Angriff ermöglicht. Die Eintrittswahrscheinlichkeit für einen derartigen Angriff wird, auf Basis von Kapitel 4.4, mit „hoch“ bemessen, da der notwendige Aufwand gering ist.

Kapitel 4.4 hat gezeigt, dass Thrashing-Verhalten hervorgerufen werden kann. Da die Eintrittswahrscheinlichkeit jedoch von der Implementation des VMM abhängt, wäre eine „sehr hohe“ Eintrittswahrscheinlichkeit nicht angemessen. Die Eintrittswahrscheinlichkeit des Angriffsschritts 1.1.1.2 wird daher mit „hoch“ bemessen.

Die Eintrittswahrscheinlichkeit des Angriffsschritts 2.1 wird, auf Basis der Argumentation aus Kapitel 4.4, mit „hoch“ bemessen. Der Aufwand für einen DoS-Angriff gegen das Host-OS ist gering, da diese i.d.R. automatisierbar sind und Sicherheitslücken, die einen DoS-Angriff ermöglichen, häufig auftreten.

Die Eintrittswahrscheinlichkeit für ein technisches Versagen des VMM wird in Punkt 2.3 mit „hoch“ bemessen, da die Hardwarekomponenten potentiell in diesem Szenario sehr stark beansprucht werden.

Die Eintrittswahrscheinlichkeit für den Verlust der Verfügbarkeit der personenbezogenen Daten wird daher insgesamt ebenfalls mit „hoch“ bemessen. Das Schadensausmaß wird, wie der Diebstahl personenbezogener Daten, mit „sehr hoch“ bemessen, da die Geschäftsfähigkeit von KM stark beeinträchtigt und die Aussenwirkung des Unternehmens geschädigt wird [2]. Das Risiko für die Schutzziele der personenbezogenen Daten wird daher durch Virtualisierung für dieses Szenario deutlich verstärkt und muss, bzgl. der Verfügbarkeit der personenbezogenen Daten, mit „hoch“ bis „sehr hoch“ bemessen werden.

Für die Integrität des Buchungssystems gilt der in Abbildung 5.4 gezeigte Bedrohungsbaum.

Die Eintrittswahrscheinlichkeiten der Angriffsschritte wurden bereits argumentiert. Werden personenbezogene Daten, bspw. Accountdaten von Kunden, gestohlen, können unerwünschte Buchungen durchgeführt werden (Angriffsschritt 3).

Die Eintrittswahrscheinlichkeit für diese Bedrohung ist „hoch“. Das Schadensausmaß ist abermals als „sehr hoch“ anzusehen, so dass das Risiko „hoch“ bis „sehr hoch“ ist.

[2] Geht die Verfügbarkeit des VMM verloren, wächst das Schadensausmaß noch erheblich an, da hierdurch weitere Schutzziele betroffen sind.

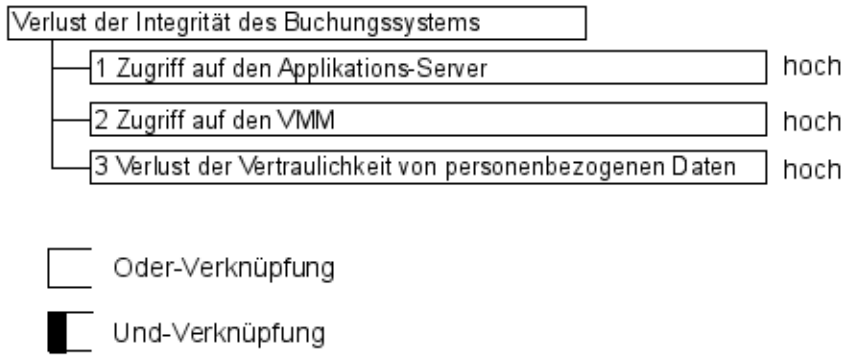

Abbildung 5.4: Verlust der Integrität des Buchungssystems

Das Risiko aller hier gezeigten Bedrohungen muss mit „hoch“ bis „sehr hoch“ bemessen werden. Der hohe Schutzbedarf der schützenswerten Güter des fiktiven Unternehmens KM erfordert Maßnahmen zur Reduktion des Risikos, da das Schutzniveau dem Schutzbedarf nicht gerecht wird.

5.4 Maßnahmen

Die Risiken der Virtualisierung können in diesem Szenario behandelt werden, indem die Sicherheit des VMM verifiziert wird, auf Virtualisierung verzichtet wird, geeignete Schutzmaßnahmen getroffen werden oder die Risiken einfach akzeptiert werden (vgl. [BSI08]). Dem VMM einfach zu vertrauen wäre bei hohem bis sehr hohem Schutzbedarf fahrlässig, zumal die vergangen Kapitel gezeigt haben, dass aufgrund von bestimmten Designkriterien, Implementationshürden und hoher Komplexität, ein VMM i.d.R. nicht in hohem Maße vertrauenswürdig ist. Auf Virtualisierung zu verzichten soll hier nicht betrachtet werden. Eine umfassende Verifikation ist hier ebenfalls ausgeschlossen und wird durch hohe Komplexität des VMM zusätzlich erschwert. Auch ein als hochgradig sicher verifizierter VMM könnte die organisatorischen Probleme der Virtualisierung ggf. nicht vollständig lösen. Um Risiken abzuschwächen sieht das Security Engineering daher vor, adäquate Schutzmaßnahmen zu ergreifen [Eck08]. Hier sollen konzeptionelle Empfehlungen identifiziert werden, um die Risiken der konzeptionellen Bedrohungen zu reduzieren. Daher soll keine konkrete Anpassung der Sicherheitsstrategie von KM durchgeführt werden. Es soll vielmehr gezeigt werden, was dabei zu berücksichtigen ist.

In der Tabelle 5.1 werden von KM zu treffende, konzeptionelle Maßnahmen gezeigt, die sich aus der Bedrohungs- und Risikoanalyse ableiten, um das Risiko beim Einsatz virtualisierter

Umgebungen für KM zu reduzieren. Die Maßnahmen werden im Anschluß erklärt. Dabei wird gezeigt, für welche Systeme von KM die Maßnahmen besonders relevant sind und wie die Maßnahmen das Risiko reduzieren.

Tabelle 5.1: Konzeptionelle Maßnahmen, um das Risiko virtualisierter Umgebungen bzgl. KM zu reduzieren

Index	Maßnahme
M.1	Gäste mit unterschiedlichem Schutzbedarf nicht oberhalb des gleichen VMM konsolidieren
M.2	Die Komplexität von Virtualisierungstechnologien minimieren
M.3	Das Schutzniveau des VMM bzgl. Angriffen maximieren
M.4	Das Konsolidierungspotential eines VMM einschränken
M.5	Die bestehenden Schutzmaßnahmen aufrecht erhalten
M.6	Den Kreis der Administratoren eines VMM reduzieren
M.7	Die Mobilität von Gästen einschränken
M.8	Prozesse für die Inbetriebnahme, den Betrieb und die Wartung von Gästen definieren und einrichten

M.1 Gäste mit unterschiedlichem Schutzbedarf nicht oberhalb des gleichen VMM konsolidieren

Das Risiko eines Sicherheitsvorfalls bzgl. Gästen von KM ist höher, wenn ein erstes Eindringen in ein anderes System des gleichen Hosts erfolgreich verläuft. Bspw. durch einen Ausbruch, das Abhören des Netzwerkverkehrs oder einen Inter-VM-Angriff können Gäste mit hohem und sehr hohem Schutzbedarf daraufhin im Bedrohungsbaum der Abbildung 5.2 angegriffen werden. Da Gäste mit unterschiedlichem Schutzbedarf in diesem Szenario potentiell auf dem gleichen VMM betrieben werden, wird die Isolation der Gäste mit unterschiedlichen Schutzanforderungen vermindert. Daher wird empfohlen, Gäste mit unterschiedlichem Schutzbedarf nicht oberhalb des gleichen VMM zu betreiben. Hierdurch wird das Risiko eines Angriffs eingedämmt, da nur die Systeme gleichen Schutzbedarfs gefährdet sind. Bzgl. KM ist ein VMM daher nicht an jede der drei DMZ anzubinden, sondern stets nur an eine DMZ, da die Systeme in unterschiedlichen DMZ unterschiedliche Schutzanforderungen haben. Hierdurch wird die Isolation der unterschiedlichen Schutzbedarfskategorien aufrecht erhalten. Diese Maßnahme ist als Grundvorraussetzung für eine Virtualisierung jeglicher Systeme von KM anzusehen. Kann diese Maßnahme nicht durchgeführt werden, wird von einer Virtualisierung abgeraten.

M.2 Die Komplexität von Virtualisierungstechnologien minimieren

In dieser Arbeit wurde gezeigt, dass die Komplexität der Virtualisierungstechnologien Auswirkungen auf die Wahrscheinlichkeit mit der Sicherheitslücken existieren hat. Ein konzeptionell einfacher VMM kann die Sicherheit besser gewährleisten als ein komplexer VMM. Die Vertrauenswürdigkeit der Virtualisierungstechnologie wird durch hohe Komplexität geschmählert. Daher sind für den Einsatz im unternehmenskritischen Bereich Virtualisierungstechnologien zu wählen, die eine geringe Komplexität aufweisen. Konkret sollten die Aspekte der Tabelle 5.2 besonders berücksichtigt werden. Hierdurch kann das Risiko eines Ausbruchs, möglichen DoS- und Inter-VM-Angriffen, sowie zusätzlichen Sicherheitslücken des VMM reduziert werden. Eine Virtualisierung bzgl. Gästen mit sehr hohem Schutzbedarf von KM sollte ausschließlich unter Umsetzung aller Empfehlungen der Tabelle 5.2 stattfinden, um das Risiko virtualisierter Umgebungen durch hohe Komplexität zu minimieren. Auch eine Virtualisierung bzgl. der Gäste mit hohem Schutzbedarf von KM sollte alle Aspekte der Tabelle 5.2 berücksichtigen, da diese Systeme für die Außenwirkung des Unternehmens gegenüber Kunden und Partnern verantwortlich sind. Eine Virtualisierung bzgl. Gästen mit normalem Schutzbedarf von KM ist hingegen nicht strikt an die Empfehlungen der Tabelle 5.2 geknüpft und kann daher auch bei Abweichungen stattfinden, sofern die erforderlichen Kosten den Nutzen für die Systeme mit normalem Schutzbedarf übersteigen.

M.3 Das Schutzniveau des VMM bzgl. Angriffen maximieren

Der Schutzbedarf jedes VMM ist bei KM mindestens so hoch, wie der Schutzbedarf der jeweiligen DMZ. Verschafft sich ein Angreifer Zugang zu einem VMM, besitzt er gleichfalls privilegierten Zugang zu allen Gästen. Die Konfiguration eines VMM muss angepasst werden um Isolation zu maximieren und Sharing zu minimieren, indem Gäste insgesamt nur so viele virtuelle Ressourcen erhalten wie auch physisch vorhanden sind. Ist der Einsatz eines Host-OS durch einen VMM bzgl. Gästen mit normalem Schutzbedarf nicht zu vermeiden, können etablierte Best-Practices für das jeweilige OS eingesetzt werden, um das Host-OS gegenüber Angriffen zu härten. Best-Practices stellen Standard-Maßnahmen dar, die sich für ein System etabliert haben und anerkannt sind [SR04]. Bspw. können nicht notwendige Dienste und Komponenten entfernt oder deaktiviert werden. Das Host-OS unterliegt allen aktuellen Sicherheitslücken, die für das OS existieren. Daher müssen Patches für aktuelle Sicherheitslücken des Host-OS und den VMM eingespielt werden, sobald diese veröffentlicht werden. Die Wahrscheinlichkeit, dass offene Sicherheitslücken existieren, die bspw. in DoS-Angriffen, Privilege Escalation oder einem Einbruch in das Host-OS oder den VMM resultieren, wird hierdurch reduziert. Das Risiko eines Ausbruchs aus einem Gast wird ebenfalls durch diese Maßnahme reduziert, da die Isolation zwischen den Gästen und dem VMM maximiert wird. Bei einer Virtualisierung bzgl. Gästen mit hohem und sehr hohem Schutzbedarf

Tabelle 5.2: Maßnahmen, um Komplexität von Virtualisierungstechnologien zu reduzieren

Aspekt	Erläuterung
CPU-Virtualisierung	CPU-unterstützte Virtualisierung und Paravirtualisierung ist der Binary-Translation vorzuziehen, da Binary-Translation eine höhere Komplexität aufweist. Bei einer vollständigen, CPU-unterstützten Virtualisierung existiert, im Gegensatz zur Paravirtualisierung, keine zeitliche Verzögerung bzgl. der Veröffentlichung von Sicherheitspatches für die Gast-Betriebssysteme.
Architektur	Aus Sicherheitsaspekten ist ein Bare-Metal-VMM statt einer Hosted-Architektur und einer Domain-Architektur zu wählen, da kein komplettes herkömmliches Betriebssystem in die TCB des VMM aufgenommen werden muss. Der VMM unterliegt hierdurch nicht den Sicherheitsbeschränkungen der Betriebssysteme auf denen der VMM aufsetzt.
IOMMU	Durch den Einsatz einer IOMMU können Treiber aus der TCB des VMM entfernt werden. Die unvertrauenswürdigen Treiber können in Gäste verlagert werden, denen der VMM nicht vertrauen muss, da die IOMMU den DMA der Geräte isoliert. Der Einsatz einer IOMMU hat daher höchste Priorität um die Komplexität eines VMM zu reduzieren.
Treiber	Ist der Einsatz einer IOMMU nicht möglich sollten nur einige wenige zertifizierte Treiber eingesetzt werden, die vertrauenswürdiger sind als die meisten Treiber für die x86-Architektur. Die Fülle von IO-Komponenten, die vom VMM unterstützt werden muss, wird hierbei reduziert. Hierdurch können jedoch nur noch bestimmte Hardwarekomponenten verwendet werden.
Hardwaremulation	Die vollständige Abbildung einer Hardwarekomponente in Software ist sehr komplex und daher fehleranfällig. Auf Hardwareemulation ist daher, zugunsten einer paravirtuellen Geräteschnittstelle, zu verzichten.
Hardwareunterstützung	Zusätzliche Hardwareunterstützung, wie bspw. Nested-Page-Tables u.a., kann weitere Komplexität aus dem VMM entfernen.
Netzwerkkomponenten	Herkömmliche Komponenten der Netzwerksicherheit und zusätzliche Funktionalität sollten meiner Auffassung nach nicht in den VMM-Layer relokalisiert werden, da sie die Komplexität des VMM-Layers steigern und sich der VMM nicht transparent in eine bestehende Infrastuktur einordnet. Somit sollten sich auch Gäste, wie physische Maschinen, in ein Netzwerk eingliedern. Der Netzwerkverkehr sollte durch vorhandene physische Netzkomponenten geleitet werden, um den Netzwerkverkehr für die Komponenten sichtbar zu machen. Die virtuelle Netzkomponente, die die Gäste an das physische Netzwerk anbindet, ist als Switch zu betreiben, um Sniffing zumindest zu erschweren.
Ressourcen	Ein VMM sollte Ressourcen ausschließlich statisch zuweisen, um Komplexität der Buchhaltung von Ressourcenzuweisungen zu vermeiden. Das Risiko von DoS-, Thrashing-Angriffen und Sicherheitslücken nimmt hierdurch ab.

sollte das höchstmögliche Maß an Isolation erzielt werden. Bspw. sind unerwünschte Kommunikationskanäle[3] zwischen Gästen untereinander, sowie einem Gast und dem VMM, zu entfernen. Bei Systemen mit normalem Schutzbedarf kann das Sharing ggf. höher ausfallen. Sicherheitspatches sind jedoch unverzüglich einzuspielen, da die Eintrittswahrscheinlichkeit von Angriffen auf aktuelle Sicherheitslücken sehr hoch ist.

M.4 Das Konsolidierungspotential eines VMM einschränken

Der VMM ist ein Single Point of Failure. Die Hardware wird im Szenario der Serverkonsolidierung von KM potentiell stark belastet. Daher sind Maßnahmen zu ergreifen, die das Risiko eines Hardwareausfalls reduzieren. Hierzu zählt bspw. Hardwareredundanz. Zusätzlich muss die erlaubte Anzahl von Gästen oberhalb eines VMM reguliert werden um das Schadensausmaß von Sicherheitsvorfällen, die die Verfügbarkeit beeinträchtigen, zu reduzieren. Ein Verlust der Verfügbarkeit eines VMM führt hierdurch nicht mehr zu einem Verlust der Verfügbarkeit von beliebig vielen Gästen. Der Web- und der Mailserver aus Tabelle 2.1 sind Systeme des gleichen, hohen Schutzbedarfs, die in herkömmlichen Umgebungen unabhängig voneinander sind. Diese Systeme benötigen sich nicht gegenseitig, um ihre Aufgaben zu erledigen. In virtualisierten Umgebungen dürfen unabhängige, hochschutzbedürftige Güter ggf. nicht auf dem gleichen VMM betrieben werden, da diese Systeme dann bzgl. der Verfügbarkeit des VMM nicht mehr unabhängig voneinander sind. Das Risiko des Verlusts der Verfügbarkeit eines VMM im Bedrohungsbaum der Abbildung 5.3 kann durch diese Maßnahme reduziert werden. Diese Maßnahme ergänzt die Maßnahme M.1 dahingehend, dass das Konsolidierungspotential auch innerhalb einer Schutzbedarfskategorie eingeschränkt wird. Die Maßnahme ist für Systeme hohen und sehr hohen Schutzbedarfs bei KM besonders relevant. Bei einer Virtualisierung bzgl. Gästen mit normalem Schutzbedarf kann das Konsolidierungspotential eines VMM ggf. höher ausfallen als bei Gästen höheren Schutzbedarfs.

M.5 Die bestehenden Schutzmaßnahmen aufrecht erhalten

Gäste haben die gleichen Sicherheitslücken, die auch ohne den Einsatz virtualisierter Umgebungen existieren. Alle vorhandenen Schutzmaßnahmen für die Gäste müssen daher aufrecht erhalten werden. Patch- und Updatezyklen sind bspw. in virtualisierten Umgebungen auch für die Gäste gleichermaßen einzuhalten. In virtualisierten Umgebungen dürfen Systeme daher nicht länger ungepatcht bleiben als in herkömmlichen Umgebungen. Diese Maßnahme ist zu treffen, damit das Risiko bzgl. der Layer oberhalb eines VMM nicht steigt. Diese Maßnahme ist bei einer Virtualisierung von Systemen jeglichen Schutzbedarfs bei KM

[3] Hierzu gehört bspw. eine Clipboard-Funktionalität, um Daten zwischen Gästen kopieren zu können (vgl. [RI00]).

ausnahmslos umzusetzen. Die Maßnahme hat hohe Relevanz, weil zu erwarten ist, dass in virtualisierten Umgebungen mehr Systeme betrieben werden als in herkömmlichen Umgebungen (vgl. Kapitel 4.2.2).

M.6 Den Kreis der Administratoren eines VMM reduzieren

Der Administrator eines VMM hat privilegierten Zugriff auf jeden Gast, der oberhalb des VMM betrieben wird. In der Sicherheitsstrategie von KM muss daher berücksichtigt werden. dass ein Administrator eines VMM bzgl. allen Gästen, die auf diesem VMM betrieben werden, als vertrauenswürdig einzustufen ist. Daher muss der Kreis der Administratoren eines VMM reduziert werden. Das Risiko von Insider-Angriffen kann durch diese Maßnahme reduziert werden, da nur wenige Administratoren statisch mit Gästen und VMMs assoziiert werden, wodurch eine Ursache für Sicherheitsvorfälle mit geringerem Aufwand zu identifizieren ist. Das Risiko von Insider-Angriffen lässt sich durch diese Maßnahme jedoch nicht vollständig beseitigen. Ein Administrator hat weiterhin uneingeschränkte Privilegien gegenüber allen Gästen, so dass die Eintrittswahrscheinlichkeit und das Schadensausmaß von Insider-Angriffen höher ist als ohne den Einsatz von Virtualisierung. Diese Maßnahme ist bei einer Virtualisierung von Systemen jeglichen Schutzbedarfs bei KM relevant und ausnahmslos umzusetzen.

M.7 Die Mobilität von Gästen einschränken

Diese Maßnahme ist eng mit der Maßnahme M.6 verknüpft. Um das Schadensausmaß von Sicherheitsvorfällen zu begrenzen und Inkonsistenzen bzgl. der Vertrauenswürdigkeit von Administratoren auszuschließen, muss die Mobilität von Gästen eingeschränkt werden. Gäste dürfen nur zwischen VMMs, die dem gleichen Administrator bzw. gleichen Administrationsgruppen unterstehen, relokalisiert werden. Gäste dürfen zusätzlich nur zwischen VMMs relokalisiert werden, die über ein ausreichendes Schutzniveau verfügen. Bzgl. KM darf ein Gast daher maximal zwischen VMMs, die an die gleiche DMZ angebunden sind, relokalisiert werden[4]. Das Risiko von Insider-Angriffen wird hierdurch reduziert. Allerdings wird auch die Flexibilität eingeschränkt, so dass ein Gast nicht beliebig relokalisiert werden kann, was auch die Komplexität der Sicherheitsstrategie erhöht. Diese Maßnahme ist bei einer Virtualisierung von Systemen jeglichen Schutzbedarfs bei KM relevant und ausnahmslos umzusetzen.

[4] Die Mobilität wird zusätzlich durch die Maßnahme M.4 eingeschränkt.

M.8 Prozesse für die Inbetriebnahme, den Betrieb und die Wartung von Gästen definieren und einrichten

Definierte Prozesse für die Inbetriebnahme, den Betrieb und die Wartung von virtualisierten Umgebungen können verhindern, dass Gäste sporadisch, ohne definierte Zuständigkeiten, betrieben werden. Jeder Gast muss von KM zentral und eindeutig erfasst werden, damit im Falle eines Sicherheitsvorfalls alle betroffenen Systeme identifiziert und bereinigt werden können. Es müssen Vorkehrungen getroffen werden, damit aus der Verwendung eines Rollback-Mechanismus keine Sicherheitslücken resultieren. Daher sind die Veränderungen in einem Gast, die zwischen zwei Rollback-Schritten stattfinden, innerhalb eines definierten Prozesses zu dokumentieren und bereits vor einem Rollback zu evaluieren. Das Risiko eines ersten Eindringens in einen beliebigen Gast und das Schadensausmaß von Sicherheitsvorfällen wird durch diese Maßnahme reduziert, da die Dauer und Wahrscheinlichkeit mit der ungepatchte Sicherheitslücken oder kompromittierte Gäste existieren reduziert wird. Diese Maßnahme ist bei einer Virtualisierung von Systemen jeglichen Schutzbedarfs bei KM relevant und ausnahmslos umzusetzen.

5.5 Empfehlung

Es bleibt festzuhalten, dass der Einsatz virtualisierter Umgebungen für eine Serverkonsolidierung die Sicherheit der zu virtualisierenden Systeme grundsätzlich vermindert. Wenn Komplexität der Virtualisierungstechnologie reduziert, die Flexibilität und das Konsolidierungspotential der Technologie eingeschränkt werden, kann das Risiko virtualisierter Umgebungen reduziert werden. Da virtualisierte Umgebungen jedoch gerade Flexibilität und Konsolidierungspotential bzgl. einer Serverkonsolidierung steigern sollen, ist das Maß der Einschränkungen und der hierfür notwendige finanzielle Aufwand entscheidend, ob der Einsatz virtualisierter Umgebungen lohnenswert ist. Virtualisierte Umgebungen unterscheiden sich maßgeblich von herkömmlichen Umgebungen. Daher kann ein Einsatz virtualisierter Umgebungen nur unter sorgfältiger Planung sicher durchgeführt werden. Für die Systeme mit normalem Schutzbedarf von KM ist eine Serverkonsolidierung unter Berücksichtigung der in diesem Kapitel genannten Aspekte als lohnenswert und ausreichend sicher anzusehen. Für die Systeme mit hohem und sehr hohem Schutzbedarf wäre eine Serverkonsolidierung unter Gesichtspunkten der Informationssicherheit ausschließlich unter Berücksichtigung aller in diesem Kapitel genannten Maßnahmen zu empfehlen. Die Risiken der Virtualisierung für die hochschutzbedürftigen Güter von KM lassen sich hierdurch deutlich reduzieren. Die positive Empfehlung für das fiktive Unternehmen KM ist jedoch an starke Bedingungen und Beschränkungen geknüpft, so dass Motive, die das fiktive Unternehmen KM für die Virtualisierung verfolgt, beeinträchtigt werden (vgl. Kapitel 2.2). Die Güter sehr hohen Schutzbedarfs sind i.d.R nicht zahlreich. Da die zugehörigen Systeme nicht mit Systemen niedrigerer

Schutzbedarfskategorien oberhalb eines VMM betrieben werden dürfen, ist ein hoher finanzieller Nutzen der Virtualisierung von hochschutzbedürftigen Systemen unwahrscheinlich. Es muss eine ausführliche Produktrechersche bzgl. der genannten Sicherheitsaspekte, sowie eine Kosten/Nutzen-Schätzung vor dem Einsatz virtualisierter Umgebungen von den Verantwortlichen durchgeführt werden.

6 Schlussbetrachung

In diesem Kapitel wird die Arbeit kritisch reflektiert und zussammengefasst. Es wird ein Ausblick auf die Zukunft des Themas gegeben.

6.1 Fazit

Diese Arbeit sollte zeigen, dass Virtualisierung konzeptionell die Informationssicherheit steigern, aber auch verringern kann. Es wurde gezeigt, dass der Einsatz von Virtualisierung nicht zwangsläufig die Sicherheit steigert. Virtualisierung kann zwar ein kosteneffektives Mittel sein, um die Sicherheit eines Systems zu steigern, indem eine Sicherheitsarchitektur statt eines einzigen Betriebssystems mit *n* Diensten, *n* Betriebssysteme mit *n* Diensten vorsieht. Hierdurch kann eine höhere logische Isolation der Dienste erzielt werden, sofern die Komplexität des VMM gering ist. Bzgl. der Serverkonsolidierung vermindert Virtualisierung jedoch ausschließlich die Sicherheit. Eine Serverkonsolidierung schafft physische Isolation explizit ab und führt nicht zu einer höheren, logischen Isolation.

Virtualisierte Umgebungen führen zu organisatorischen Problemen. Administratoren von VMMs verfügen über sehr weitreichende Privilegien, so dass der Vertrauenswürdigkeit von Personen, die einen Zugang zum VMM besitzen, zentrale Bedeutung zukommt. Die Mobilität, die Inbetriebnahme und der Betrieb von virtuellen Maschinen muss reguliert werden, um die Eintrittswahrscheinlichkeit und das Schadensausmaß von Sicherheitsvorfällen in virtualisierten Umgebungen zu begrenzen.

Ein VMM benötigt als Systemkomponente extrem hohe Privilegien in einem System. Der VMM muss deshalb möglichst schmal ausfallen. Aufgrund architektonischer und hardwareseitiger Beschränkungen, sowie dem Wunsch nach zusätzlicher Funktionalität, fällt der VMM jedoch häufig sehr komplex aus. Die Sicherheit des VMM-Layers kann daher i.d.R. nicht garantiert werden. Mit steigender Komplexität sinkt die Vertrauenswürdigkeit des VMM weiter, da auch eine Verifkation immer schwieriger wird und Sicherheitslücken wahrscheinlicher werden. Daher muss die Flexibilität der Virtualisierung, durch die Ansprüche an die Sicherheit von hochschutzbedürftigen Gütern, weiter eingeschränkt werden.

Bei den Einschränkungen muss ein Kompromiss zwischen vertretbaren Einschnitten bzgl. der Sicherheit und Flexbilität gefunden werden, der von den eingesetzten Systemen abhängig ist. Ziel dieser Arbeit war weiterhin die Frage, ob und wie sichere, virtualisierte Umgebungen betrieben werden können. Die Antwort auf diese Frage ist daher im Ausmaß der Einschränkungen der Flexibilität und des Konsolidierungspotentials zu finden, also gerade denjeningen Eigenschaften, die die Virtualisierungstechnologie ausmachen. Die Risiken der Virtualisierung werden dadurch handhabbar, dass die Flexibilität eingeschränkt wird und die Komplexität der Technologie reduziert wird. Güter geringen Schutzbedarfs ermöglichen höhere Flexibilität. Güter höheren Schutzbedars erlauben nur geringere Flexibilität, so dass nur im Kontext des konkreten Szenarios entschieden werden kann, ob der Einsatz virtualisierter Umgebungen überhaupt lohnenswert ist.

Das Sprichwort von David Wheeler gilt auch bzgl. der Virtualisierung: Jedes Problem der Informatik kann durch eine zusätzliche Indirektionsebene gelöst werden. Aber jede neue Indirektionsebene schafft üblicherweise auch ein neues Problem [Wik09].

6.2 Ausblick

Dieses Kapitel zeigt, wo sich eine weitere Sicherheitsanalyse virtualisierter Umgebungen anschließen könnte und wie sich das Themengebiet ggf. entwickeln wird.

Virtualisierungstechnologien werden zunehmend zu einem Standard und beim Betrieb von Servern allgegenwärtig.

Hierdurch ist einerseits zu erwarten, dass die bereits anfängliche Unterstützung der Virtualisierung durch Hardwaremechanismen weiter zunimmt. Es ist bspw. anzunehmen, dass der Verbreitungsgrad von IOMMUs steigt und Prozessorhersteller ihre Virtualisierungskonzepte verstärkt ausweiten. Hierdurch kann die Komplexität und der Implementationsaufwand von Virtualisierungstechnologien auf Ebene der Software sinken, so dass Isolation mit weniger Aufwand sicher gestellt werden kann und die Technologien schmaler und sicherer werden oder komplexe Virtualisierungskonzepte unnötig werden. Die heutigen Komplexitätsprobleme eines VMM können daher in der Zukunft ggf. gelöst werden. Hierdurch kann höhere Flexibilität und ein höheres Konsolidierungspotential in virtualisierten Umgebungen auch für hochschutzbedürftige Systeme erzielt werden.

Andererseits nimmt auch die Anzahl von Virtualisierungslattformen zu. Die zunehmende Konkurrenz unter VMM-Herstellern kann darin münden, dass zusätzliche Funktionalität in den VMM-Layer aufgenommen wird um Verkaufsargumente zu schaffen und den Umgang mit virtuellen Servern noch einfacher zu gestalten. Die Komplexität eines VMM steigt hierdurch jedoch ggf. weiter an, so dass die Vertrauenswürdigkeit eines derartigen VMM sinkt und der Verbreitungsgrad und die Funktionsvielfalt ggf. schneller steigt als die Sicherheit

nachziehen kann. Ein derartiger VMM erlaubt in diesem Fall nur geringe Flexibilität und ein niedriges Konsolidierungspotential bzgl. hochschutzbedürftigen Gütern.

Die Mobilität von Gästen und andere organisatorische Probleme, die mit virtualisierten Umgebungen einher gehen, können auch durch die genannten Hardwaremechanismen nicht gelöst werden. Da ein VMM ganze Netzwerke konsolidieren kann, muss der VMM die Sicherheitsstrategie einer Institution durchsetzen. Daher müssen Konzepte geschaffen und bzgl. der Sicherheit analysiert werden, die versuchen, diese organisatorischen Probleme der Virtualisierung zu lösen.

A Anhang

A.1 Thrashing-Angriff

In diesem Kapitel soll exemplarisch ein Angriff auf virtualisierte Umgebungen durchgeführt werden, der in Thrashing-Verhalten des VMM mündet. Hierdurch soll die Verfügbarkeit von Gästen beeinträchtigt werden. Der Angriff soll zeigen, dass die Erkenntnisse der Bedrohungsanalyse bzgl. der Memory-Virtualisierung auch in der Praxis sehr relevant sind. Der Angriff ist als „Proof of Concept“ zu verstehen, d.h. es soll gezeigt werden, dass es unter bestimmten Bedingungen grundsätzlich möglich ist, dass unvertrauenswürdige Gäste Thrashing-Verhalten des VMM hervorrufen können.

In der Bedrohungsanalyse wurde gezeigt, dass Thrashing-Verhalten des VMM dann hervorgerufen werden kann, wenn ein VMM Swapping verwendet um Hauptspeicher dynamisch zwischen Gästen zu verschieben. Der VMM kann Gästen hierdurch mehr virtuellen Hauptspeicher anbieten als an realem Hauptspeicher vorhanden ist. Wenn der Gast das Swapping des VMM kontrollieren kann, kann der Gast Thrashing-Verhalten des VMM hervorrufen. Der VMM wählt hierbei Speicherseiten für das Swapping aus, die zum Workload von Gästen gehören, so dass Speicherseiten vielfach ein- und ausgelagert werden müssen. Andere Gäste oberhalb des gleichen VMM werden hierdurch beeinträchtigt, da CPU-Zyklen verschwendet werden und auf den langsamen Sekundärspeicher gewartet werden muss. Für einen Thrashing-Angriff werden daher Virtualisierungsprodukte benötigt, die Swapping verwenden.

Bei der Kernel-based Virtual Machine (KVM) [Qum09b] ist es möglich, Gästen mehr virtuellen Hauptspeicher zuzuweisen als realer Hauptspeicher vorhanden ist, was als Memory-Overcommitment bezeichnet wird. Der offizielle Linux-Kernel enthält KVM seit der Version 2.6.20 des Linux-Kernels als festen Bestandteil [Qum09b]. Qumranet, der Hersteller von KVM, wurde im September 2008 von Red Hat Linux gekauft [Hat09]. Red Hat plant KVM als Basis der eigenen Virtualisierungsstrategie für Server- und Desktop-Virtualisierung anzubieten [Hat09]. KVM ermöglicht eine vollständige Virtualisierung auf der x86-Architektur durch CPU-unterstützte Virtualisierung, d.h. es werden Prozessoren benötigt, die eine CPU-unterstützte Virtualisierung erlauben. Shadow-Seitentabellen werden von KVM verwendet um den Hauptspeicher zu virtualisieren [Qum09a], so dass der VMM Speicherseiten auf

Sekundärspeicher auslagern muss, wenn der virtuelle Hauptspeicher nicht in den realen Hauptspeicher passt. KVM wird daher exemplarisch für einen Angriff auf Virtualisierungsplattformen verwendet, die Memory-Overcommitment zulassen und Swapping verwenden um Hauptspeicher flexibel unter Gästen zu verteilen.

Da ein VMM Speicherseiten transparent auslagert ist es schwierig zu zeigen, dass Performanceeinbußen tatsächlich auf Thrashing-Verhalten zurückzuführen sind. Daher müssen andere Möglichkeiten ausgeschlossen werden, indem die zu untersuchenden Szenarien dementsprechend gewählt werden.

Es wurde das folgende Szenario untersucht:

- Es wurde KVM in der Version „QEMU PC emulator version 0.9.1 (kvm-72), Copyright (c) 2003-2008 Fabrice Bellard“ als Virtualisierungsplattform verwendet.
- Die verwendete Hardware bestand aus zwei Gigabyte Hauptspeicher, einem Intel Core 2 Duo E8400 Prozessor [Int09b], einer 750 Gigabyte großen Festplatte und einem 1-Gigabit Ethernet-Adapter.
- Als Host-Betriebssystem wurde ein Debian GNU Linux 5.0 Lenny (Stand: 18.03.2009) als Minimalinstallation verwendet [Deb09].
- Als Kernel wurde „Linux version 2.6.26-1-amd64 (Debian 2.6.26-13)“ verwendet.
- Der VMM betrieb zwei Gäste, „Good“ und „Evil“, die ebenfalls die genannte Linux-Distribution als Gast-Betriebssystem verwendeten.
- Gast „Evil“ wurde mittels

  ```
  kvm evil.img -m 2048 -curses -k de
  -net nic,macaddr=52:54:00:12:34:56,model=rtl8139
  -net tap,ifname=tap1
  ```

  gestartet und erhielt somit zwei Gigabyte virtuellen Hauptspeicher.
- Gast „Good“ wurde mittels

  ```
  kvm good.img -m 1024 -curses -k de
  -net nic,macaddr=52:54:00:12:34:55,model=rtl8139
  -net tap,ifname=tap0
  ```

  gestartet und erhielt somit ein Gigabyte virtuellen Hauptspeicher.
- Gast „Evil“ betrieb keine zusätzlichen Dienste.
- Gast „Good“ betrieb als typischen Webserver einen „Apache/2.2.9 (Debian)“. Das entsprechende Paket wurde von der Distribution bereitgestellt.

Es wurde ein Gigabyte mehr virtueller Hauptspeicher zugewiesen als real verfügbar war, um den VMM zu zwingen Speicherseiten des VMM selbst oder des Gasts „Good“ auszulagern,

wenn der Gast „Evil“ den gesamten virtuellen Hauptspeicher verwendet, der ihm zugewiesen wurde. Es wurde erwartet, dass die Performance des Gasts „Good“ hierdurch in hohem Maße beeinträchtigt werden würde.

Der Verlust der Performance muss konkret bemessen werden, um die Wirkung des Thrashing-Angriffs abschätzen zu können. Hierzu wurde ein Benchmark verwendet, um die verbleibende Leistungsfähigkeit des Webservers zu belegen. Der Benchmark wurde auf einem zusätzlichen Host betrieben, der über dedizierte Hardware verfügt, so dass das Ergebnis nicht verfälscht werden konnte. Alle beteiligten Hosts wurden am selben 1-Gigabit Switch und in einem gemeinsamen Subnetz betrieben, um Auswirkungen der Netzwerkverbindung auf den Benchmark durch bspw. Routing zu unterbinden. Als Benchmark wurde die Software „httperf“ in der Version „httperf-0.9.0 compiled Jun 23 2008“ verwendet [HPDC09]. Der Benchmark versuchte hierbei eine statische HTML-Seite von 3704 Byte Textdaten vom o.g. Webserver des Gasts „Good“ mit einer spezifizierten Frequenz abzurufen.

Damit der Gast „Evil“ seinen gesamten virtuellen Hauptspeicher verwendet und regelmäßig auf ihn zugreift, wurde Listing A.1 entwickelt. Der resultierende Prozess allokiert geringfügig weniger Hauptspeicher als dem Gast zugewiesen wurde, um Swapping des Gast-Betriebssystems zu unterbinden.

Listing A.1: thrashing.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MEMORY_MB 1950
#define MEMORY_MAX (MEMORY_MB*1048576)

main(){
        char* ch = (char*)malloc(MEMORY_MAX);

        memset(ch, 0, MEMORY_MAX);

        printf("%s\n", "Hauptspeicher initialisiert");

        while(1) {
                memset(ch, 1, MEMORY_MAX);
        }
}
```

Dieser Code wurde in Gast „Evil“ mittels

```
gcc -o thrashing thrashing.c
./thrashing
```

kompiliert und gestartet [1]. Nachdem der Hauptspeicher initialisiert wurde, wurde der Benchmark mittels

```
httperf --server good.local --uri /test.html
--num-conn 3000 --num-call 10 --rate 100 --timeout 5
```

von einem anderen Host aus gestartet. Die Parametereinstellungen des Benchmarks sind so gewählt, dass ein Webserver den Benchmark i.d.R. fehlerlos bearbeiten kann. Der Benchmark wurde drei mal durchgeführt um verlässliche Ergebnisse zu erzielen. Von den 3000 Verbindungen stellte der Benchmark beim ersten Versuch 3000 Fehler, beim zweiten Versuch 2269 Fehler und beim dritten Versuch 2995 Fehler durch Timeouts fest, so dass der Webserver nur noch geringfügig in der Lage war Anfragen zu beantworten. Der Benchmark wurde daraufhin vom Anspruch weiter abgeschwächt und erneut durchgeführt. Der Benchmark wurde mittels

```
httperf --server good.local --uri /test.html
--num-conn 500 --num-call 10 --rate 10 --timeout 5
```

gestartet. Von 500 Verbindungen traten jedoch immernoch 97, 184 und 153 Fehler durch Timeouts auf. Daraufhin wurde der thrashende Prozess in Gast „Evil“ beendet und der erste Benchmark erneut mehrfach durchgeführt. Hierbei wurden keine Fehler festgestellt, so dass der Benchmark stets vollständig und korrekt abgearbeitet wurde. Der Webserver war also, durch den thrashenden Prozess nicht mehr in der Lage, zuverlässig Anfragen zu beantworten, obwohl der Benchmark nicht anspruchsvoll war.

Um zu zeigen, dass die Performanceeinbußen ausschließlich durch das Memory-Overcommitment und das darausfolgende Swapping hervorgerufen wurden, wurde daraufhin mehr Hauptspeicher im Host verbaut, so dass vier Gigabyte realer Hauptspeicher vorhanden waren. Hierdurch konnten alle Einstellungen des Szenarios beibehalten werden, so dass nur das Swapping des VMM entfiel. Listing A.1 wurde gleichermaßen gestartet. Der Benchmark wurde erneut mehrfach durchgeführt und wurde trotz des Prozesses durchgängig ohne Fehler von dem Webserver bearbeitet, so dass das Swapping des VMM als Ursache der Performanceeinbußen verbleibt.

Dieser Angriff sollte zeigen, dass unvertrauenswürdige Gäste unter bestimmten Bedingungen Thrashing-Verhalten des VMM hervorrufen können. Der Aufwand für einen derartigen

[1] S. [Fou09].

Angriff ist sehr niedrig wenn Zugang zu einem beliebigen Gast des Hosts besteht. Hierbei werden keine besonderen Privilegien in dem Gast benötigt. Der Nutzen, den ein Angreifer aus diesem Angriff ziehen kann, ist hingegen immens, da ggf. die Verfügbarkeit aller Gäste des Hosts in hohem Maße beeinträchtigt wird. Daher kommt der Angriff einem Denial of Service gleich.

Glossar

AMD-Pacifica S. AMD-SVM.

AMD-SVM AMD-SVM ist eine Technologie von AMD, die die Virtualisierung durch die CPU unterstützen soll. Prozessoren, die diese Technologie anbieten, verfügen über einen erweiterten Instruktionssatz, der die Virtualisierung der x86-Architektur einfacher macht.

Bare-Metal-VMM Ein VMM, der direkt auf der Hardware betrieben wird. Ein Bare-Metal-VMM setzt nicht auf einem herkömmlichen Betriebssystem auf, sondern übernimmt das Scheduling und die Verwaltung der Ressourcen vollständig selbst.

Common Vulnerabilities and Exposures Common Vulnerabilities and Exposures sind ein Standard der Sicherheitsindustrie. Konkrete Sicherheitslücken von Produkten der Informationstechnologie werden hierbei zentral und einheitlich in Form von CVEs u.a. von der MITRE-Corporation erfasst und identifiziert. Jede CVE, die in dieser Arbeit genannt wird, kann unter [MIT09b] nachvollzogen werden.

Demilitarisierte Zone Eine demilitarisierte Zone ist ein Computernetzwerk, auf das nur kontrolliert von Außerhalb des Netzwerks zugegriffen werden kann, da das Netzwerk durch Firewalls von allen anderen Netzen abgeschottet wird. Durch eine demilitarisierte Zone können bspw. Hosts der demilitarisierten Zone Dienste für das Internet bereitstellen ohne das lokale Netzwerk, das an die demilitarisierte Zone angrenzt, im Internet bloßzustellen.

Denial of Service Ein Angriff, der bspw. die Angriffziele veranlasst, alle verfügbaren Ressourcen aufzubrauchen, so dass der Dienst eingestellt werden muss und die Verfügbarkeit des Systems verloren geht.

Domain-Architektur Ein VMM verlagert hierbei die Komponenten eines oder mehrerer Host-Betriebssysteme in virtuelle Maschinen (s. Gast). Hierdurch setzt der VMM nicht direkt auf den Betriebssystemen auf, kann aber dennoch die Funktionalität der Betriebssysteme selbst verwenden.

Emulation Ein System, das ein anderes System nachahmt. Hierdurch wird Kompatibilität für ein System hergestellt, das das nachgebildete System benutzt und davon ausgeht, das originale System zu verwenden.

Exploit Code, der eine spezifische Sicherheitslücke ausnutzt, um Privilegien in einem System zu erlangen oder das System zum Absturz zu bringen.

Extended Page Tables S. Nested Page Tables.

Firewall Eine Firewall kontrolliert den Netzwerkverkehr zwischen zwei oder mehr Netzen. Hierbei wird u.a. entschieden ob bestimmte Pakete die Firewall passieren dürfen oder nicht, so dass die Firewall den Netzwerkverkehr filtert.

Gast Ein Gast ist eine virtuelle Maschine, die oberhalb eines VMM betrieben wird. Ein Gast stellt einen vollständigen, logischen Rechner dar.

Hosted-Architektur Der VMM wird bei einer Hosted-Architektur nicht direkt auf der Hardware betrieben, sondern oberhalb eines herkömmlichen Betriebssystems. Der VMM nutzt daher Komponenten des Betriebssystems, wie bspw. Gerätetreiber, für seine eigene Funktionalität.

Hub Eine Netzwerkkomponente, die in einem lokalen Netzwerk eine Broadcast-Kommunikation in der Sicherungsschicht ermöglicht.

Intel-VT Intel-VT ist eine Technologie von Intel, die die Virtualisierung durch die CPU unterstützen soll. Prozessoren, die diese Technologie anbieten, verfügen über einen erweiterten Instruktionssatz, der die Virtualisierung der x86-Architektur einfacher macht.

Inter-VM-Angriff Eine virtuelle Maschine (s. Gast) wird aus einer anderen virtuellen Maschine des gleichen Hosts angegriffen.

Malware Software, die ein unerwünschtes, schadhaftes Verhalten aufweist. Das schadhafte Verhalten wird i.d.R. vor Benutzern versteckt.

Micro- und Macropayment Die Bezahlung geringer Summen, die i.d.R. digital vollzogen wird.

Nested Page Tables Nested Page Tables stellen konzeptionell Shadow-Seitentabellen (s. Shadow-Seitentabelle) auf Hardwareebene bereit.

Privilege Escalation Privilege Escalation bezeichnet einen Angriff. Ein Angreifer kann durch spezifische Sicherheitslücken oder andere Fehler eines Systems seine Privilegien in dem System ausweiten.

Pure-Isolation-VMM Ein Pure-Isolation-VMM verfolgt die vollständige Isolation bei der Virtualisierung. Hardwarekomponenten werden nicht zwischen Gästen (s. Gast) geshared. Jeder Gast erhält stattdessen dedizierte Hardwarekomponenten. Ein Pure-Isolation-VMM wird i.d.R. nur für teure Mainframesysteme eingesetzt, da sich der Einsatz andernfalls finanziell nicht lohnt.

Sanitising Code, der unvertrauenswürdigen Code überprüft oder modifiziert, um ihn vertrauenswürdig zu machen.

Schutzring Schutzringe werden von einem Prozessor benutzt, um Code mit unterschiedlichen Privilegien ausführen zu können. Code, der innerhalb eines Schutzrings ausgeführt wird, darf nur eine Teilmenge des gesamten Instruktionssatzes direkt ausführen.

Seitentabelle Eine Seitentabelle wird bei der virtuellen Speicherverwaltung des Hauptspeichers benutzt. Die Seitentabelle bildet Speicherseiten aus einem Adressbereich, üblicherweise eines virtuellen Adressbereichs, auf Speicherseiten eines anderen Adressbereichs, bspw. den Adressbereich des physischen Hauptspeichers, ab.

Shadow-Seitentabelle Eine Shadow-Seitentabelle wird bei der Virtualisierung benutzt, um, wie eine herkommliche Seitentabelle, eine zusätzliche Indirektionsebene auf Hauptspeicherebene zu schaffen. Hierdurch kann der gesamte Hauptspeicher transparent und isoliert unter den Gästen verteilt werden.

Sicherheitslücke Fehlfunktion einer Software, die ein Angreifer ausnutzen kann, um Schutzziele anzugreifen.

Sicherheitsstrategie Umfasst alle technischen und organisatorischen Regeln, Verhaltensrichtlinien, Verantwortlichkeiten und Rollen sowie Maßnahmen um angestrebte Schutzziele zu erreichen [Eck08].

Sniffing Bezeichnet einen Angriff auf Netzwerkebene. Der Netzwerkverkehr wird abgehört und mitgeschnitten. Er kann eingesehen werden, wodurch ggf. die Vertraulichkeit des Netzwerkverkehrs verloren geht.

Swapping Ein System, das virtuellen Speicher benutzt, kann Speicherseiten transparent auf Sekundärspeicher, wie bspw. der Festplatte, auslagern. Der Sekundärspeicher ist langsamer als der Primärspeicher, aber i.d.R. bedeutend günstiger und größer.

Switch Eine Netzwerkkomponente, die in einem lokalen Netzwerk eine Kommunikation in der Sicherungssicht ermöglicht. Ein Paket wird nicht per Broadcast vermittelt, sondern i.d.R. direkt an den Adressaten geleitet. Nur wenn ein Paket nicht direkt an den Adressaten geleitet werden kann wird das Paket per Broadcast verschickt.

Thrashing Thrashing kann aus Swapping (s. Swapping) resultieren. Werden Speicherseiten auf langsamerem Sekundärspeicher ausgelagert, die zum Workload des zugehörigen Codes gehören, müssen diese Speicherseiten ggf. sehr häufig aus- und wieder eingelagert werden, da der Code häufig auf diese Speicherseiten zugreift. Hierdurch werden CPU-Zyklen verschwendet und die Gesamtperformance sinkt beträchtlich.

Traffic Sämtliche Daten, die zwischen Teilnehmern eines Netzwerks fließen.

Trusted Computing Base Eine Trusted Computing Base umfasst alle sicherheitskritischen Bestandteile eines Systems oder einer Komponente, auf deren korrekte Funktionsweise und Verfügbarkeit ein anderes System oder eine andere Komponente vertraut. Die Trusted Computer Base spiegelt daher eine Vertrauensbeziehung zwischen Komponenten wider.

Virtueller Speicher S. Seitentabelle.

Abkürzungen

AMD-SVM	Advanced Micro Devices - Secure Virtual Machine Architecture
AMD-V	Advanced Micro Devices - Virtualization Technology
BDSG	Bundesdatenschutzgesetz
BSI	Bundesamt für Sicherheit in der Informationstechnologie
CVE	Common Vulnerabilities and Exposures
DEC	Digital Equipment Corporation
DMA	Direct Memory Access
DMZ	Demilitarized Zone, Demilitarisierte Zone
DoS	Denial of Service
EPT	Extended Page Tables
HTML	Hypertext Markup Language
HV	Hypervisor
Intel-VT	Intel Virtualization Technology
Intel-VT-d	Intel Virtualization Technology for Directed I/O
IOMMU	I/O Memory Management Unit
KM	Kygert Micropayment GmbH
KVM	Kernel-based Virtual Machine
KVM/370	Kernelized Virtual Machine
NIC	Network Interface Controller, Netzwerkadapter
NPT	Nested Page Tables
OS	Operating System, Betriebssystem

StGB	Strafgesetzbuch
TCB	Trusted Computing Base
VM	Virtual Machine
VMM	Virtual Machine Monitor

Literaturverzeichnis

[AA06] Keith Adams and Ole Agesen. A comparison of software and hardware techniques for x86 virtualization. In *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pages 2–13, San Jose, California, USA, 2006. ACM.

[AMD07] AMD. AMD I/O Virtualization Technology (IOMMU) Specification. http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/34434.pdf, 2007. 12.02.2009.

[AMD08] AMD. AMD-V Nested Paging. http://developer.amd.com/assets/NPT-WP-1%201-final-TM.pdf, 2008. 02.02.2009.

[AMD09] AMD. Documentation. http://developer.amd.com/documentation/guides/Pages/default.aspx, 2009. 02.02.2009.

[Arm07] Ben Armstrong. *Professional Microsoft Virtual Server 2005*. Wiley & Sons, 1., auflage edition, April 2007.

[BDF+03] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM.

[BDR97] Edouard Bugnion, Scott Devine, and Mendel Rosenblum. Disco: running commodity operating systems on scalable multiprocessors. In *SOSP '97: Proceedings of the sixteenth ACM symposium on Operating systems principles*, pages 143–156, New York, NY, USA, 1997. ACM.

[Bel06] Steven M. Bellovin. Virtual machines, virtual security? *Commun. ACM*, 49(10):104, 2006.

[Bis02] Matthew A. Bishop. *Computer Security: Art and Science*. Macmillan Technical Publishing, December 2002.

[BLRS08] Sergey Bratus, Michael E. Locasto, Ashwin Ramaswamy, and Sean W. Smith. Traps, events, emulation, and enforcement: managing the yin and

yang of virtualization-based security. In *VMSec '08: Proceedings of the 1st ACM workshop on Virtual machine security*, pages 49–58, New York, NY, USA, 2008. ACM.

[BMR+08] K. Bellam, A. Manzanares, Xiaojun Ruan, Xiao Qin, and Yiming Yang. Improving reliability and energy efficiency of disk systems via utilization control. *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 462–467, July 2008.

[BS06] Uwe Baumgarten and Hans-Jürgen Siegert. *Betriebssysteme: Eine Einführung*. Oldenbourg, 6., überarb., aktualis. u. erw. a. edition, December 2006.

[BSI08] BSI. BSI-Standard 100-1, Managementsysteme für Informationssicherheit (ISMS). http://www.bsi.bund.de/literat/bsi_standard/standard_1001.pdf, 2008. 13.12.2008.

[Bun08] Bund. Strafgesetzbuch. http://bundesrecht.juris.de/bundesrecht/stgb/gesamt.pdf, 2008. 27.04.2009.

[Bun09] Bund. Bundesdatenschutzgesetz. http://bundesrecht.juris.de/bundesrecht/bdsg_1990/gesamt.pdf, 2009. 27.04.2009.

[BYM05] Muli Ben-Yehuda and John D. Mason. The Xen Hypervisor and its IO Subsystem. http://www.mulix.org/lectures/xen-iommu/xen-io.pdf, 2005. 21.11.2008.

[BYMX+06] Muli Ben-Yehuda, Jon Mason, Jimi Xenidis, Orran Krieger, Leendert van Doorn, Jun Nakajima, Asit Mallick, and Elsie Wahlig. Utilizing iommus for virtualization in linux and xen. In *OLS '06: The 2006 Ottawa Linux Symposium*, pages 71–86, July 2006.

[BYXO+07] Muli Ben-Yehuda, Jimi Xenidis, Michal Ostrowski, Karl Rister, Alexis Bruemmer, and Leendert van Doorn. The Price of Safety: Evaluating IOMMU Performance. http://www.xen.org/files/xensummit_4/price-of-safety-xen-summit-apr-07_Muli.pdf, 2007. 21.11.2008.

[CE09] Computer-Economics. Server Virtualization Growth: Slow but Steady. http://www.computereconomics.com/article.cfm?id=1424&tag=rbspot, 2009. 02.02.2009.

[Chi07] David Chisnall. *The Definitive Guide to the Xen Hypervisor*. Prentice Hall International, 1 edition, December 2007.

[Cit09] Citrix. Citrix Xen Server. http://citrix.com/English/ps2/products/product.asp?contentID=683148, 2009. 30.03.2009.

[Cor09] Microsoft Corporation. Hyper-V Feature Overview. http://msdn.microsoft.com/en-us/library/cc768521.aspx, 2009. 30.03.2009.

[Deb09] Debian. Debian – The Universal Operating System. http://www.debian.org/, 2009. 03.04.2009.

[deG91] Der Rat der europäischen Gemeinschaften. Richtlinie des Rates vom 10. Juni 1991 zur Verhinderung der Nutzung des Finanzsystems zum Zwecke der Geldwäsche. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CONSLEG:1991L0308:20011228:DE:PDF, 1991. 27.04.2009.

[DU04] Marius Dannenberg and Anja Ulrich. *E-Payment und E-Billing: Elektronische Bezahlsysteme für Mobilfunk und Internet*. Gabler, 1 edition, February 2004.

[Eck08] Claudia Eckert. *IT Sicherheit*. Oldenbourg Verlang München Wien, 2008. 5. Auflage.

[ePudRdeU00] Das europäische Parlament und der Rat der europäischen Union. Richtlinie 2000/47/EG des europäischen Parlaments und des Rates. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2000:275:0039:0043:DE:PDF, 2000. 27.04.2009.

[FHN+04] Keir Fraser, Steven Hand, Rolf Neugebauer, Ian Pratt, Andrew Warfield, and Mark Williamson. Reconstructing i/o. Technical report, 2004.

[Fou09] Free Software Foundation. GCC, the GNU Compiler Collection. http://gcc.gnu.org/, 2009. 20.04.2009.

[fSidI09] Bundesamt fuer Sicherheit in der Informationstechnologie. Schutzbedarfskategorien. http://www.bsi.bund.de/gshb/webkurs/gskurs/seiten/s4100.htm, 2009. 02.02.2009.

[Gar08a] Gartner. Gartner Identifies the Top 10 Strategic Technologies for 2008. http://www.gartner.com/it/page.jsp?id=530109, 2008. 02.02.2009.

[Gar08b] Gartner. Gartner Says Virtualization Will Be the Highest-Impact Trend in Infrastructure and Operations Market Through 2012. http://www.gartner.com/it/page.jsp?id=638207, 2008. 02.02.2009.

[Gar09] Gartner. Gartner Identifies the Top 10 Strategic Technologies for 2009. http://www.gartner.com/it/page.jsp?id=777212, 2009. 02.02.2009.

[Gei07] Kaspar Geiser. Sicherheit und Kosten der Virtualisierung. http://www.aspectra.ch/uploads/media/Virtualisierung_IT-Security_207.pdf, 2007. 15.11.2008.

[GPC+03] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: a virtual machine-based platform for trusted computing. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 193–206, New York, NY, USA, 2003. ACM.

[GR05] Tal Garfinkel and Mendel Rosenblum. When virtual is harder than real: security challenges in virtual machine based computing environments. In *HOTOS'05: Proceedings of the 10th conference on Hot Topics in Operating Systems*, pages 20–20, Berkeley, CA, USA, 2005. USENIX Association.

[Hat09] Red Hat. Red Hat Outlines Its Virtualization Strategy and Roadmap for 2009. http://www.redhat.com/virtualization-strategy/, 2009. 03.04.2009.

[Hoe02] Von Marc Hoeft. *Zahlungssysteme im Electronic Commerce*. 2002.

[HPDC09] L.P. Hewlett-Packard Development Company. Welcome to the httperf homepage. http://www.hpl.hp.com/research/linux/httperf/, 2009. 03.04.2009.

[HPHS04] Michael Hohmuth, Michael Peter, Hermann Härtig, and Jonathan S. Shapiro. Reducing tcb size by using untrusted components: small kernels versus virtual-machine monitors. In *Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 22, Leuven, Belgium, 2004. ACM.

[HR91] Judith S. Hall and Paul T. Robinson. Virtualizing the vax architecture. *SIGARCH Comput. Archit. News*, 19(3):380–389, 1991.

[HUL06] Gernot Heiser, Volkmar Uhlig, and Joshua Levasseur. Are virtual-machine monitors microkernels done right? *SIGOPS Oper. Syst. Rev.*, 40(1):95–99, January 2006.

[HWF+05] Steven Hand, Andrew Warfield, Keir Fraser, Evangelos Kotsovinos, and Dan Magenheimer. Are virtual machine monitors microkernels done right? In *HOTOS'05: Proceedings of the 10th conference on Hot Topics in Operating Systems*, pages 1–1, Berkeley, CA, USA, 2005. USENIX Association.

[IDC08a] IDC. EMEA Server Market Performs Strongly in 2Q08 as IT Managers Invest in Scalable Servers, Demand Diversifies From x86 Space, Says IDC. http://www.idc.com/getdoc.jsp?containerId=prUK21403008, 2008. 17.04.2009.

[IDC08b] IDC. Server Virtualization Now Firmly Embedded in European Organizations, According to IDC Survey. http://www.idc.com/getdoc.jsp?containerId=prUK21327108, 2008. 02.02.2009.

[IDC09] IDC. IDC-Umfrage: Servervirtualisierung in europäischen Anwenderunternehmen fest integriert. www.idc.com/germany/downloads/pdf/

pm2008/pm09_Server%20Virtualisierung_final.pdf, 2009. 17.04.2009.

[Int08] Intel. Intel Virtualization Technology for Directed I/O. ftp://download.intel.com/technology/computing/vptech/Intel(r)_VT_for_Direct_IO.pdf, 2008. 12.02.2009.

[Int09a] Intel. Future of Intel Virtualization Architecture. http://www.intel.com/technology/itj/2006/v10i3/1-hardware/8-virtualization-future.htm, 2009. 02.02.2009.

[Int09b] Intel. Intel Core 2 Duo Processor. http://www.intel.com/products/processor/core2duo/specifications.htm?iid=prod_core2duo+tab_spec, 2009. 05.04.2009.

[ITS08] ITSecurity. Virtualization Security. http://www.itsecurity.com/features/virtualization-security-061708/, 2008. 02.02.2009.

[Kar05] P.A. Karger. Multi-level security requirements for hypervisors. *Computer Security Applications Conference, 21st Annual*, pages 9 pp.–, Dec. 2005.

[Kar08] Paul A. Karger. Is Your Virtual Machine Monitor Secure? http://grid.hust.edu.cn/aptc08/slides/08_Is%20Your%20Virtual%20Machine%20Monitor%20Secure_Paul%20A.Karger.ppt, 2008. 27.11.2008.

[KS08] P.A. Karger and D.R. Safford. I/o for virtual machine monitors: Security and performance issues. *Security & Privacy, IEEE*, 6(5):16–23, Sept.-Oct. 2008.

[KZB+90] P.A. Karger, M.E. Zurko, D.W. Bonin, A.H. Mason, and C.E. Kahn. A vmm security kernel for the vax architecture. *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pages 2–19, May 1990.

[KZB+91] Paul A. Karger, Mary Ellen Zurko, Douglas W. Bonin, Andrew H. Mason, and Clifford E. Kahn. A retrospective on the vax vmm security kernel. *IEEE Trans. Softw. Eng.*, 17(11):1147–1165, 1991.

[Law99] Kevin Lawton. Running multiple operating systems concurrently on an IA32 PC using virtualization techniques. http://www.floobydust.com/virtualization/lawton_1999.txt, 1999. 21.11.2008.

[MD73] Stuart E. Madnick and John J. Donovan. Application and analysis of the virtual machine approach to information system security and isolation. In *Proceedings of the workshop on virtual computer systems*, pages 210–224, Cambridge, Massachusetts, United States, 1973. ACM.

[MIT09a] MITRE. CVE Identifiers Defined. http://cve.mitre.org/cve/identifiers/index.html, 2009. 02.02.2009.

[MIT09b] MITRE. CVE List. http://cve.mitre.org/cve/cve.html, 2009. 02.02.2009.

[MMH08] Derek Gordon Murray, Grzegorz Milos, and Steven Hand. Improving xen security through disaggregation. In *VEE '08: Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pages 151–160, New York, NY, USA, 2008. ACM.

[M"u07] Klaus-Rainer Müller. *IT-Sicherheit mit System: Sicherheitspyramide - Sicherheits-, Kontinuitäts- und Risikomanagement - Normen und Practices - SOA und Softwareentwicklung*. Vieweg+Teubner, 3., erweiterte und aktualisierte auflage. edition, November 2007.

[Orm07] Travis Ormandy. An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments. http://taviso.decsystem.org/virtsec.pdf, 2007. 27.04.2009.

[PE06] D. E. Atencio Psille and J. Eschweiler. *Security@Work*. Springer, Berlin, 1 edition, July 2006.

[PG73] Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. In *SOSP '73: Proceedings of the fourth ACM symposium on Operating system principles*, page 121, New York, NY, USA, 1973. ACM.

[PvDS08] R. Perez, L. van Doorn, and R. Sailer. Virtualization and hardware-based security. *Security & Privacy, IEEE*, 6(5):24–31, Sept.-Oct. 2008.

[PWB07] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso. Failure trends in a large disk drive population. In *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, pages 2–2, Berkeley, CA, USA, 2007. USENIX Association.

[Qum09a] Qumranet. kvm: Kernel-based Virtual Machine for Linux. http://www.haifux.org/lectures/152/kvm-external.pdf, 2009. 22.03.2009.

[Qum09b] Qumranet. Status - KVM. http://www.linux-kvm.org/page/Status, 2009. 22.03.2009.

[RB08] Dirk Rzepka and Uwe Bünning. *Microsoft Windows Server 2008. Einrichten und Verwalten von Unternehmensnetzwerken*. Hanser Fachbuch, 1 edition, June 2008.

[Reu07] Jenny Susan Reuben. A Survey on Virtual Machine Security. http://www.tml.tkk.fi/Publications/C/25/papers/Reuben_final.pdf, 2007. 27.04.2009.

[RG05] M. Rosenblum and T. Garfinkel. Virtual machine monitors: current technology and future trends. *Computer*, 38(5):39–47, 2005.

[RI00] John Scott Robin and Cynthia E. Irvine. Analysis of the intel pentium's ability to support a secure virtual machine monitor. In *Proceedings of the 9th conference on USENIX Security Symposium - Volume 9*, pages 10–10, Denver, Colorado, 2000. USENIX Association.

[RM06] Andrej Radonic and Frank Meyer. *XEN 3*. Franzis, 1., aufl. edition, October 2006.

[S01] TANENBAUM ANDREW S. *Modern Operating Systems*. Prentice Hall India, 2nd edition, 2001.

[Sec07] Secunia. Corporate Presentation. http://secunia.com/gfx/pdf/Secunia_Corporate_Profile.pdf, 2007. 12.04.2009.

[Sec09a] Secunia. Vulnerability Report: Citrix XenServer 1.x. http://secunia.com/advisories/product/19197/, 2009. 02.02.2009.

[Sec09b] Secunia. Vulnerability Report: VMware ESX Server 3.x. http://secunia.com/advisories/product/10757/, 2009. 02.02.2009.

[Sec09c] Secunia. Vulnerability Report: VMware Server 1.x. http://secunia.com/advisories/product/10733/, 2009. 02.02.2009.

[Sec09d] Secunia. Vulnerability Report: Xen 3.x. http://secunia.com/advisories/product/15863/, 2009. 02.02.2009.

[SGLS77] Marvin Schaefer, Barry Gold, Richard Linde, and John Scheid. Program confinement in kvm/370. In *Proceedings of the 1977 annual conference*, pages 404–410. ACM, 1977.

[SJV+05] Reiner Sailer, Trent Jaeger, Enriquillo Valdez, Ramon Caceres, Ronald Perez, Stefan Berger, John Linwood Griffin, and Leendert van Doorn. Building a mac-based security architecture for the xen open-source hypervisor. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 276–285, Washington, DC, USA, 2005. IEEE Computer Society.

[SN05] James E. Smith and Ravi Nair. The architecture of virtual machines. *Computer*, 38(5):32–38, 2005.

[SPF+07] Stephen Soltesz, Herbert Pötzl, Marc E. Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *EuroSys '07: Proceedings of the*

2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007, pages 275–287, New York, NY, USA, 2007. ACM.

[SR04] Brady R. Stevenson and Gordon W. Romney. Teaching security best practices by architecting and administering an it security lab. In *CITC5 '04: Proceedings of the 5th conference on Information technology education*, pages 182–187, New York, NY, USA, 2004. ACM.

[SS75] J.H. Saltzer and M.D. Schroeder. The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308, Sept. 1975.

[SVL01] Jeremy Sugerman, Ganesh Venkitachalam, and Beng-Hong Lim. Virtualizing i/o devices on vmware workstation's hosted virtual machine monitor. In *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, pages 1–14, Berkeley, CA, USA, 2001. USENIX Association.

[Tec08] Core Security Technologies. Path Traversal vulnerability in VMware's shared folders implementation. http://www.coresecurity.com/content/advisory-vmware, 2008. 24.02.2009.

[TEG04] Marie-Theres Tinnefeld, Eugen Ehmann, and Rainer W. Gerling. *Einfuehrung in das Datenschutzrecht: Datenschutz und Informationsfreiheit in europaeischer Sicht*. Oldenbourg, 4., völlig neubearb. u. erw. a. edition, October 2004.

[Tho08] Fabian Thorns. *Das Virtualisierungs-Buch*. C & l Computer- U. Literaturverlag, 2., aktualisierte und erweiterte auflage. edition, September 2008.

[UNR+05] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M. Martins, A.V. Anderson, S.M. Bennett, A. Kagi, F.H. Leung, and L. Smith. Intel virtualization technology. *Computer*, 38(5):48–56, 2005.

[VMw08] VMware. VMware ESX 3.5. http://www.vmware.com/files/de/pdf/esx_datasheet_de.pdf, 2008. 25.11.2008.

[VMw09] VMware. VMotion-Übersicht, Migration virtueller Maschinen, Virtualisierung. http://www.vmware.com/de/products/vi/vc/vmotion.html, 2009. 02.02.2009.

[VN08] S.J. Vaughan-Nichols. Virtualization sparks security concerns. *Computer*, 41(8):13–15, 2008.

[Wal02] Carl A. Waldspurger. Memory resource management in vmware esx server. In *OSDI '02*. USENIX, 2002.

[WCSG05] Andrew Whitaker, Richard S. Cox, Marianne Shaw, and Steven D. Gribble. Rethinking the design of virtual machine monitors. *Computer*, 38(5):57–62, 2005.

[Wik08] Wikipedia. Micropayment. http://de.wikipedia.org/wiki/Micropayment, 2008. 04.02.2009.

[Wik09] Wikipedia. David Wheeler. http://de.wikipedia.org/wiki/David_Wheeler, 2009. 05.03.2009.

[Woj08] Rafal Wojtczuk. Adventures with a certain Xen vulnerability (in the PVFB backend). http://invisiblethingslab.com/pub/xenfb-adventures-10.pdf, 2008. 24.02.2009.

[Wol07] Chris Wolf. Virtual switch security: VMware, Virtual Server and XenExpress. http://searchservervirtualization.techtarget.com/tip/0,289483,sid94_gci1244407,00.html, 2007. 10.12.2008.

[WSG02] Andrew Whitaker, Marianne Shaw, and Steven D. Gribble. Denali: a scalable isolation kernel. In *Proceedings of the 10th workshop on ACM SIGOPS European workshop*, pages 10–15, Saint-Emilion, France, 2002. ACM.

[Zab08] Matias Zabaljauregui. Hardware Assisted Virtualization Intel Virtualization Technology. http://linux.linti.unlp.edu.ar/kernel/images/f/f1/Vtx.pdf, 2008. 10.12.2008.

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §24(5) ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 30. April 2009	
Ort, Datum	Unterschrift