Virtualisierung komplexer Netzwerke

Prof. Dr. -Ing. Evren Eren, Dipl. -Inf. Markus Breitländer

Fachhochschule Dortmund
Emil-Figge-Str. 42
44227 Dortmund
eren@fh-dortmund.de
markus@breitländer.de

Abstract: Der Begriff Virtualisierung hat viele Ausprägungen und wird in verschiedenen Kontexten der Informationstechnik benutzt. Der wesentliche Gesichtspunkt von Virtualisierung ist es, die Abhängigkeit von Hardware und Software aufzutrennen. Diese Abstraktion führt dazu, dass vorhandene IT-Ressourcen flexibel genutzt und eine höhere Auslastung erzielt werden kann.

Die Virtualisierung von Betriebssystemen hat sich bereits seit geraumer Zeit etabliert. Mit Hilfe moderner Virtualisierungssoftware bestehen heute Möglichkeiten, sogar komplexe IT-Infrastrukturen und deren Komponenten (Subnetze, Router, Switches, Firewalls, DMZ, etc.) zu virtualisieren.

Dieser Beitrag stellt das Projekt "VISA - Virtual IT-Security Architectures" vor, dass im LISA-Labor (www.lisa.fh-dortmund.de) des Fachbereiches Informatik an der FH-Dortmund entwickelt wurde. In diesem Projekt wurde auf Basis von Open Source Software eine komplexe IT-Sicherheitsinfrastruktur für mittelständische Unternehmen virtualisiert. Hierbei wurde die Software KVM (GPL) eingesetzt, welche Vollvirtualisierung auf x86-Hardware ermöglicht. KVM nutzt die Befehlssatzerweiterungen der neueren Intel- und AMD-Prozessoren (Intel VT und AMD-V). Vollvirtualisierung erlaubt den parallelen Betrieb mehrerer Betriebssysteme in einem unveränderten Modus. Die virtuellen Maschinen "wissen" also nicht, dass sie in einer virtuellen Umgebung laufen und müssen nicht angepasst werden. KVM ist eine Abspaltung vom Emulator „QEMU“.

Zur Realisierung der Netzwerkkommunikation zwischen den virtuellen Maschinen dient die Software VDE (Virtual Distributed Ethernet). VDE stellt wichtige Elemente einer virtuellen Infrastruktur, wie Switches und Kabel zur Verfügung.

Im Weiteren wird der Einsatz speziell konfigurierten virtuellen Maschinen, sowie von Hardware-Emulatoren, als virtuelle Router und Firewalls beschrieben. In diesem Zusammenhang wird die Router- und Firewall-Distribution Vyatta und der Router-Emulator Dynamips vorgestellt.

Durch die Kombination der beiden Softwarelösungen KVM und VDE sowie Emulatoren wie Dynamips lassen sich komplexe Netzwerkstrukturen flexibel und "realitätsnah" virtualisieren.

1 LISA (Laboratory for IT-Security Architectures)

Das „Laboratory for IT-Security Architectures – LISA" im Fachbereich Informatik der Fachhochschule Dortmund stellt eine modulare Entwicklungs- und Evaluations-Plattform für IT-Sicherheitsarchitekturen zur Verfügung. Hier können Sicherheitsmodelle und -architekturen exemplarisch diskutiert, erprobt und validiert werden. LISA wird sowohl für die praxisorientierte Lehre und Forschung, als auch als Demo-Center für externe Unternehmen eingesetzt. Dabei werden insbesondere Sicherheitsprobleme und -architekturen von Klein- und Mittelständischen Unternehmen adressiert.

2 VISA

Im Rahmen des Projektes VISA wurde eine komplexe IT-Sicherheitsinfrastruktur virtualisiert. VISA soll in diesem Beitrag als ein Beispiel für eine komplexe virtuelle Netzwerktopologie dienen.

Im Folgenden werden die verschiedenen Softwarelösungen und Techniken vorgestellt, die zur Virtualisierung der verschiedenen Komponenten der LISA-Infrastruktur eingesetzt wurden. Zunächst ging es darum, die bestehenden Systeme samt ihrer Konfiguration in virtuelle Maschinen (VM) zu überführen. Bei diesem Verfahren spricht man von der sog. Physical-to-Virtual Migration (P2V). Hierbei wird eine exakte Kopie der Festplatte erstellt, welche dann später als virtueller Datenträger in die VMs eingebunden wird. In der Regel können hierfür gängige Imaging-Tools wie Clonezilla oder Acronis True Image genutzt werden. Es existieren allerdings auch spezielle P2V-Tools wie beispielsweise Platespin Powerconvert oder VMware Converter, mit denen es möglich ist, Systeme während des laufenden Betriebs zu migrieren. Die so erstellten Images lassen sich von modernen Vollvirtualisierungslösungen meist ohne weitere Anpassungen gebootet. In VISA wurden die Images der Linux-Systeme mit Clonezilla erstellt und zur Migration der Windows-Systeme kam VMware Converter zum Einsatz.

3 Kernel-based Virtual Machine (KVM)

KVM [KVM] ist eine Open Source Virtualisierungslösung (GPL) für Linux, welche Vollvirtualisierung auf x86-Hardware ermöglicht. Hierbei werden die Befehlssatz-erweiterungen der modernen Intel- und AMD-Prozessoren (Intel VT und AMD-V) genutzt. Das Verfahren der Vollvirtualisierung ermöglicht es, mehrere unveränderte Linux- und Windows-Betriebssysteme parallel auf einem System zu betreiben.

KVM wurde 2006 von der israelischen Firma Qumranet veröffentlicht und bereits ein halbes Jahr später in den Linux-Kernel (> 2.6.20) aufgenommen. Qumranet wurde im September 2008 von Red Hat gekauft [HEI]. KVM ist heutzutage in den meisten Distributionen als Standard-Virtualisierungslösung integriert. Sie ist die einzige wirklich „freie“ Virtualisierungslösung, welche die volle Unterstützung der Linux-Community erfährt und profitiert direkt von Weiterentwicklungen am Linux-Kernel. Auch in der Industrie erfährt KVM eine breite Unterstützung. Es kooperieren die Firmen AMD, IBM, Intel, Suse und RedHat bei der Weiterentwicklung der Software.

KVM baut auf dem Emulator QEMU [WIK0] was verschiedene Prozessorarchitekturen emulieren kann. Hierzu zählen PowerPC, ARM, Alpha, m68k, MIPS und Sparc. Er stellt weiterhin die virtuelle Hardware (wie bspw. virtuelle Netzwerkinterfaces) den VMs zur Verfügung.

KVM besteht aus den ladbaren Kernel-Modulen kvm.ko sowie den prozessorabhängigen Modulen kvm-intel.ko und kvm-amd.ko. Bei entsprechender Hardwareunterstützung ermöglicht KVM das äußerst performante Verfahren der hardwarebasierten Vollvirtualisierung. Hierbei werden CPU-Instruktionen direkt auf der Hardware des Hostsystems ausgeführt. Performanz-kritische I/O-Zugriffe lassen sich zusätzlich durch angepasste (paravirtualisierte) Gerätetreiber für Netzwerk- sowie Blockdevices weiter optimiert. Hierbei baut KVM auf dem offenen Standard VirtIO [LIN] auf.

Für Netzwerkverbindungen zwischen VMs sowie zum Anschluss an reale Netzwerke bietet KVM mehrere Netzwerkoptionen. Zur vollständigen TCP/IP-Anbindung wird in der Praxis meist auf TUN/TAP-Interfaces zurückgegriffen. TUN und TAP sind virtuelle Netzwerk-Kerneltreiber, die Netzwerkgeräte über Software simulieren. TUN simuliert ein Point-to-Point-Netzwerkgerät, während TAP ein Ethernet-Gerät darstellt. Für jedes virtuelle Netzwerkinterface einer VM wird im Hostsystem jeweils ein TAP-Interface erstellt. Diese können dann über Netzwerk-Bridges (IEEE 802.1d) des Hostsystems verbunden werden. An eine Bridge lassen sich wiederum auch physikalische Netzwerkschnittstellen des Hostsystems anschließen, was eine Kommunikation der virtuellen Maschinen mit dem Netzwerk des Hostsystems ermöglicht.

Eine weitere Netzwerkoption stellt VDE (Virtual Distributed Ethernet) dar, welche im nächsten Kapitel vorgestellt wird. KVM-VMs können direkt mit VDE-Netzwerken verbunden werden.

4 Virtual Distributed Ethernet (VDE)

Die Software Virtual Distributed Ethernet (VDE) [WIK1] stellt eine allgemeine, virtuelle Infrastruktur zur Verbindung verschiedener Softwarekomponenten zur Verfügung.

VDE ist Ethernet-konform und stellt virtuellen Infrastrukturen virtuelle Switche und virtuelle Kabel zur Verfügung. Es lassen sich virtuelle Maschinen verschiedener Virtualisierungslösungen, Emulatoren, reale Betriebssysteme und Netzwerke miteinander verbinden. Auf Basis von VDE lassen sich sehr einfach und flexibel virtuelle Netzwerke erstellen. Teilbereiche solcher virtuellen Netzwerke lassen sich auf mehrere physikalische Rechner verteilen.

VDE-Netzwerke bestehen aus den folgenden Hauptkomponenten:

VDE-Switch: In der virtuellen Umgebung übernimmt ein VDE-Switch die gleiche Funktion wie ein physikalischer Ethernet-Switch eines realen Netzwerkes. Er verfügt über mehrere Ports, über die verschiedenen Systeme (Computer, Router und andere Ethernet-kompatiblen Endgeräte) einer virtuellen Infrastruktur miteinander verbunden werden können. Intern arbeitet ein solcher virtueller Switch wie ein realer Layer-2-Switch. Er „lernt“ durch Analyse der Paketheader und verwaltet an Hand einer Adresstabelle (Source Address Table) dynamisch die Verbindungen zwischen Hardwareadressen (MAC) und den Switch-Ports.

VDE-Plug: Ein VDE-Plug (dt. Stecker) ist ein Hilfsprogramm, welches sich mit einem Port eines VDE-Switches verbinden kann. Ein VDE-Plug stellt ein universelles Tool dar, mit dem sich der Datenstrom des virtuellen Netzwerkes auf die Betriebssystem-Schnittstellen „stdin“ und „stdout“ umleiten lässt.

VDE-Wire: Als VDE-Wire (dt. Leitung) können Programme eingesetzt werden, die in der Lage sind, Datenströme bidirektional miteinander zu verbinden. Für diesen Zweck hat VDE das Tool dpipe. Jedoch sind auch andere Tools wie bspw. netcat oder ssh einsetzbar.

VDE-Cable: Ein VDE-Cable (dt. Kabel) besteht aus der Kombination zweier VDE-Plugs und einem VDE-Wire. Dieses Konstrukt stellt das virtuelle Pendant zum Netzwerkkabel eines realen Netzwerks dar.

Die o.g. Hauptkomponenten stellen die Basis zum Aufbau von VDE-Netzwerken dar. Die Terminologie orientiert sich hierbei an den Komponenten realer Netzwerke.

VDE-Switches lassen sich durch VDE-Cables untereinander verbinden. Um Loop-Verbindungen zwischen Switches zu verhindern, unterstützten VDE-Switches das Fast-Spanning-Tree Protocol. Da, wie beschrieben, ein VDE-Wire sich auch über Netzwerk-protokolle realisieren lässt, können Verbindungen zwischen VDE-Switchen auch über ein physikalisches Netzwerk hergestellt werden. Somit sind virtuelle Netzwerke, bspw. über eine verschlüsselte SSH-Verbindung sehr einfach zu verbinden. Auch lassen sich somit VPNs realisieren. Darüber hinaus unterstützt VDE die Realisierung von VLANs nach dem IEEE 802.1q-Standard.

Da sich TUN/TAP-Interfaces an einen VDE-Switch anschließen lassen, besteht die Möglichkeit, virtuelle Netzwerke mit dem realen Netzwerk des Hostsystems zu verbinden.

Mit dem VDE-Tool wirefilter besteht eine weitere interessante Möglichkeit, verschiedene Eigenschaften einer Netzwerkverbindung zu simulieren. Dabei werden die Eigenschaften eines VDE-Cables herangezogen. Solchen virtuellen Kabeln können QoS-Parameter wie Bandbreite, Verzögerung, Paketverlust, etc. zugeordnet werden.

Alle wichtigen Komponenten von VDE sind User-Mode-Prozesse. Somit sind auch unprivilegierte Benutzer in der Lage, eigene virtuelle Netzwerke zu erstellen, ohne die Netzwerkkonfiguration des Hosts verändern zu müssen. Über die Rechteverwaltung des Betriebssystems des Hosts kann zudem geregelt werden, welche Benutzer sich mit welchen virtuellen Netzwerken verbinden dürfen. (vgl. [WIK2])

5 Virtuelle Router

Wichtige Komponenten einer Netzwerkinfrastruktur stellen Router- und Firewall-Systeme dar. Je nach Sichtweise dienen sie dazu, Netzwerke bzw. Netzwerksegmente miteinander zu verbinden oder zu trennen. Neben spezieller Hardware können auch normale PCs als sogenannte Software-Router eingesetzt werden, da viele moderne Betriebssysteme über entsprechende Routingdienste verfügen. Gleiches gilt auch für den Einsatz als Software-Firewall. Für den Einsatz als Software-Router sowie als Software-Firewall existieren allerdings auch spezielle, vorkonfigurierte Distributionen mit teilweise mächtigem Funktionsumfang. Bekannte Open Source Lösungen sind beispielsweise SmoothWall, Edian Firewall, pfSense oder Vyatta.

Durch den Einsatz von Virtualisierungssoftware können virtuelle Router und Firewalls in Form von entsprechend konfigurierten virtuellen Maschinen realisiert werden. Hierbei bieten sich die erwähnten Distributionen als Betriebssysteme solcher VMs an.

Vyatta

Im Projekt VISA wurde hierfür die Software Vyatta [VYA1] eingesetzt. Vyatta ist eine umfangreiche Open Source Router- und Firewall-Distribution auf Basis eines angepassten Debian-Linux Systems. Vyatta bietet zahlreiche Netzwerkfunktionen, welche auch auf professionellen Hardware-Routern zum Einsatz kommen. Unterstützt werden diverse Routingprotokolle wie bspw. OSPF, BGP und RIP. Vyatta bietet jedoch nicht nur Routingfunktionen, es lässt sich auch als VPN-Server und Firewall einsetzen. Weitere Features sind u.a. DHCP, NAT, PPPoE, QoS, site-to-site IPsec VPN, SSL-based OpenVPN, RADIUS, 802.1q VLANs, Stateful Inspection Firewalling, High Availability (VRRP), SNMP sowie Intrusion Detection/Prevention und Anti-Virus. (vgl. [VYA2])

Das System lässt sich komfortabel über eine Management-Konsole, sehr ähnlich der von Juniper JUNOS oder Cisco IOS, sowie über eine Web-basierte GUI und üblichen Linux-Systembefehlen konfigurieren und verwalten. Nicht nur virtualisiert stellt Vyatta eine leistungsfähige und erstzunehmende Alternative zu Hardwarerouter wie die von Cisco oder Juniper dar.

Dynamips

Eine weitere Möglichkeit ist der Einsatz eines Hardware-Emulators, um Router in einer virtuellen Infrastruktur abzubilden. Hierbei sei erwähnt, dass sich Durchsatz und Performanz nicht mit Hardware-Routern und virtualisierten Software-Routern vergleichen lassen. In diese Kategorie fällt die Software Dynamips. Dynamips (GPL) wurde im Jahr 2005 zur Emulation von Cisco Router-Plattformen entwickelt. Sie emuliert die Hardware verschiedener Cisco Router und ermöglicht es, unangepasste Cisco IOS Betriebssysteme auf x86-Hardware zu betreiben. Der Emulator wurde mit der Zielsetzung entwickelt, Anwendern die Möglichkeit zu bieten, sich mit dem weitverbreiteten Router-Betriebssystem Cisco-IOS vertraut zu machen, ohne teure Cisco-Hardware zu benötigen. Er stellt somit eine freie Test- und Trainingsplattform für Cisco-IOS zur Verfügung. Die Software kann auch bspw. dazu genutzt werden, Konfigurationen in der emulierten Umgebung zu erstellen und zu testen, um diese dann später auf realen Routern einzusetzen.

Mit der Software GNS3 [GNS] existiert ein komfortables grafisches Frontend für Dynamips. Auch wird die Emulation von Cisco PIX- und ASA-Firewalls sowie das Juniper-Betriebssystem JunOS unterstützt. Hierzu werden speziell veränderte Versionen des Emulators Qemu benutzt. Diese inoffiziellen Versionen von Qemu sind auch unter den Namen PEMU [PEM] und JEMU (Codename Olive) bekannt [JUN].

6 Fazit

Dieser Beitrag zeigt auf, dass sich mit Hilfe moderner Virtualisierungssoftware sogar komplexe IT-Infrastrukturen bis auf Layer 1 des ISO-OSI-Modells virtuell abbilden lassen. Das Projekt "VISA - Virtual IT-Security Architectures" im LISA-Labor (www.lisa.fh-dortmund.de) des Fachbereiches Informatik an der FH-Dortmund entwickelte hierzu eine Lösung auf Basis von KVM sowie VDE. Des Weiteren wurde die Router- und Firewall-Distribution Vyatta und der Router-Emulator Dynamips eingesetzt.

Literaturverzeichnis

[KVM] http://www.linux-kvm.org/page/
[HEI] http://www.heise.de/open/Red-Hat-investiert-in-Virtualisierung-Update--/news/meldung/115452
[LIN] http://www.linux-kvm.org/page/Virtio
[WIK0] http://wiki.qemu.org/Main_Page
[WIK1] http://wiki.virtualsquare.org/index.php/Introduction#Virtual_Square_Networking
[WIK2] http://wiki.virtualsquare.org/index.php
[VYA1] http://www.vyatta.org/
[VYA2] http://www.vyatta.com/downloads/datasheets/vyatta_virtual_datasheet.pdf
[GNS] http://www.gns3.net/
[PEM] http://www.blindhog.net/pemu-cisco-pix-emulator/
[JUN] http://juniper.cluepon.net/index.php/Olive