ComConsult Research

Der Netzwerk Insider

August 2012

Systematische Weiterbildung für Netzwerk- und IT-Professionals

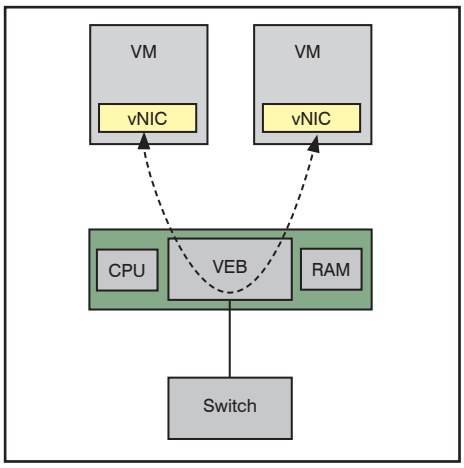
Schwerpunktthema

Virtuelle Netze - Zurück zum Soft-Switch?

von Dipl.-Inform. Damian Lukowski

Die Servervirtualisierung hat sich mittlerweile in vielen Rechenzentren als eine erprobte Technologie etabliert, mit deren Hilfe mehrere virtuelle Maschinen sich die Systemressourcen eines einzelnen physischen Hosts teilen. Werden bestimmte Ressourcen (z.B. CPU-Leistung) von einer virtuellen Maschine zu einem gewissen Zeitpunkt nicht benötigt, können diese Ressourcen von anderen Maschinen genutzt werden, ohne dass es zu Engpässen kommt.

Die Tatsache, dass mehrere virtuelle Systeme auf einem physischen Host betrieben werden, implizierte eine Ausweitung des Netzwerks in den Host hinein, und



damit ein erstes Umdenken im Netzdesign. Da die Anzahl virtueller Maschinen auf einem Host die Anzahl der verfügbaren physischen Schnittstellen mit Leichtigkeit übersteigt, teilen sich im Allgemeinen mehrere VMs einige wenige Host-Schnittstellen zur Kommunikation mit der Außenwelt. Innerhalb des Hosts werden die virtuellen Schnittstellen (vNICs) mehrerer VMs sowohl miteinander als auch mit dem Uplink-Port verschaltet; der Hypervisor implementiert eine virtuelle Ethernet Bridge in Software. VMware hat dem Konstrukt einen schöneren Namen gegeben (virtueller Switch, vSwitch), Citrix und Microsoft nennen es in ihren Produkten ein „virtuelles Netzwerk“ (≠ VLAN).

weiter auf Seite 12

Zweitthema

IPv6 und Sicherheit - technische Ansätze, der Weg zum Sicherheitskonzept

von Dipl.-Inform. Oliver Flüs

IPv6 steht vor der Tür – und die Fragestellung eines sicheren Umgangs damit hat den Fuß mindestens schon in der Tür. Wie in einem früheren Insider-Artikel mit Blick auf IPv6 und Sicherheit dargelegt wurde, gibt es nicht nur die Aufforderung der IETF an Hersteller von IT-Produkten, IPv6 nicht länger als „Option“ zu betrachten, sondern erste Hersteller, die IPv6-fähige Lösungen auch schon anbieten.

Insbesondere Microsoft zwingt mit seinen neuesten Betriebssystemversionen dazu, sich mit der Thematik auseinanderzusetzen, da es einen vor die Wahl stellt, IPv6 eingeschaltet zu lassen und damit eine neue Angriffsfläche für böse Buben zu bieten oder aber IPv6 konsequent zu deaktivieren, dies jedoch auf eigene Gefahr, nämlich unter Inkaufnahme von Seiteneffekten, vor denen auch mehr oder weniger deutlich gewarnt wird.

weiter auf Seite 23

Geleit

Scheitert UC an mobilen Endgeräten?

ab Seite 2

Aktuelle Kongresse

ComConsult Rechenzentrum Infrastruktur-Redesign Forum 2012

ComConsult TK-, UC- und Videokonferenzforum 2012

ab Seite 4

Standpunkt

Neue WLAN auf dem Vormarsch!

auf Seite 22

Neuerscheinung

Moderne Wireless-Technologien

auf Seite 21

Zum Geleit

Scheitert Unified Communications an mobilen Endgeräten?

In der traditionellen TK-Welt hat ein Benutzer einen Desktop-PC und ein Telefon. Beide Systeme haben etwas gemein: der normale Benutzer kann den Funktionsumfang nur sehr beschränkt benutzen, da die Bedienoberflächen mehr eine intellektuelle Herausforderung als eine Hilfe waren.

Dann kam Unified Communications UC mit zwei wesentlichen Zielen. Zum einen sollten noch mehr Funktionen in bereits unbedienbare Systeme integriert werden und zum anderen sollten zwei Benutzerfeindliche Welten vereint werden. Und die Überraschung war groß, als die Benutzer aus kaum nachvollziehbaren Gründen die Nutzung der neuen Funktionen verweigerten. Inkonsistente Clients, verschiedene Clients auf verschiedenen Endgeräten, Aufteilung von Funktionen auf mehrere Clients, Funktionen wild in Untermenüs versteckt, die Liste der Hürden für den Benutzer könnte nicht länger sein.

Dann kam der Tag, der die Industrie veränderte. Apple führte das iPhone ein und zeigte einer Jahrzehnte alten Industrie, dass ihre Produkte aus Sicht der Benutzer in den Mülleimer gehören. Wo sie dann auch konsequenter Weise seitdem landen. In der Mobilfunk-Welt ist seitdem nichts mehr wie es vorher war. Apple, Samsung und HTC dominieren die Welt. Weitere Android-Anbieter wie Huawei werden ggf. in den nächsten Jahren folgen. Nokia sucht seine Rettung bei Microsoft, das sagt alles.

Und was passiert in der UC-Welt? Obwohl mittlerweile 5 Jahre vergangen sind seitdem Steve Jobs das iPhone vorstellte passierte lange Zeit nichts. Dann kam Microsoft mit Lync und für alle überraschend einem Produkt, das den Client und das Benutzererlebnis in den Mittelpunkt stellt. Zwar bisher nicht perfekt, aber doch deutlich besser als bei den meisten Mitbewerbern. Und endlich begannen die Hardware-zentrischen TK-Anbieter inklusive Cisco zu reagieren. Und so können wir jetzt in den nächsten 18 Monaten endlich eine Welle neuer Clients mit einem Benutzer-zentrischen Design erwarten.

Reicht das? Die Antwort ist nein. Es ist schön, dass die TK-Hersteller jetzt auf eine Entwicklung reagieren, die vor 5 Jahren begonnen hat. Aber dummerweise ist die Technologie mobiler Endgeräte in der Zwischenzeit nicht stehen geblieben.

Und speziell für UC war die Einführung des iPads ein ganz gravierender Meilenstein. Ein iPhone oder auch ein Samsung Galaxy sind zu klein, um alle UC-Funktionsbereiche sauber abzudecken. Speziell für die Webkonferenz und das Lesen von Folien und Texten reicht ein Smartphone nicht aus, überhaupt der gesamte Dokumenten- und Videobereich machte auf dem Smartphone nur Sinn, solange es kein iPad gab. Wer schon lange auf das universelle mobile Kommunikationsgerät gewartet hat, der weiß nun wie es in Zukunft eventuell aussehen wird. Es fehlen vernünftige HD-Kameras und die Integration der Telefonie, aber das ist eine Frage der Zeit.

Aha, dann brauchen wir jetzt also nur noch den passenden Client für Smartphone und iPads und UC ist gerettet? Leider nein, wir haben ein neues Problem. Mit iOS und Android kam ein neues Konzept von intuitiver Bedienbarkeit. UC war entstanden, um die Komplexität verschiedener aufwendiger Funktionsbereiche mit einem leicht bedienbaren Client zu beseitigen. Wie wir inzwischen wissen, ist das gescheitert, weil die Entwickler wohl dieses Ziel nicht gekannt haben. Aber für iOS und Android hat dieses Ziel an Bedeutung verloren. Die meisten Apps sind intuitiv bedienbar, es gibt keinen messbaren Mehrwert in der Integration dieser Apps in einer zentralen App.

Ist UC damit am Ende? Das hängt davon ab, wie wichtig der Kommunikations-Übergang zwischen Apps ist. Besteht das Ziel aus einer Sprach-App lückenlos in eine Video-Kommunikation zu wechseln, dann braucht man ein System, das beide zusammen führt. Sollen Erreichbarkeits-Stati über verschiedene Apps integriert werden, braucht man ein entsprechendes System. Sollen Apps mit einem Hardware-Telefon und einem Desktop-PC integriert werden, braucht man auch dafür eine Lösung, die den Funktionsrahmen der Einzel-App eventuell sprengt. Aber auch hier ist Vorsicht geboten. Immerhin haben wir Standards wie SIP, denen auch die Apps folgen können, um dieses Ziel zu erreichen.

Zumindest haben die TK-Hersteller den Zahn der Zeit erkannt. Hinter den Kulissen herrscht aufgeregte Hektik, alle arbeiten an neuen Endgerätetypen und Clients. Cisco Cius und Avaya Flare sind Beispiele dafür.

Also doch Hoffnung? Eine Frage bleibt weiterhin im Raum stehen: wissen die TK-Hersteller eigentlich, dass Unternehmen ihr Geld mit Kunden verdienen? Die Effizienz und Qualität der Kommunikation mit Kunden ist der entscheidende Punkt. Dem gegenüber ist es schon peinlich, wenn die Anbieter die Sekunden-Bruchteile ausrechnen, die Unternehmen intern sparen können, wenn es eine bessere interne Erreichbarkeit gibt. Was könnte das Versagen einer ganzen Industrie besser zum Ausdruck bringen! Dabei gab es Lösungsansätze. Wir haben immer noch das Video eines UC-Forums, auf dem Cisco eine Lösung zur Kommunikation über Unternehmensgrenzen vorgestellt hat. Die sah auch ziemlich gut aus. Sie verschwand so schnell wieder vom Markt, dass man nur spekulieren kann, warum das so war. Aber wer wird schön Übles dabei denken, wenn die Provider mit die größten Kunden von Cisco sind. Und so wird vermutlich die alles entscheidende Technologie der Einbeziehung der Kunden vom Diktat einiger weniger weltweit dominanter Anbieter blockiert.

Macht UC dann überhaupt noch Sinn? Keine Sorge, wir haben ja noch Microsoft. Microsoft muss sich dem Diktat nicht unterwerfen. Und die Übernahme von Skype deutet an, wo es hingeht. Eine Federation zwischen Lync und Skype könnte den gesamten Markt umdrehen. Zum Ende des Jahres kommt die neue Lync-Version und Microsoft wird seine Ansätze auf dem ComConsult TK-, UC- und Videokonferenz-Forum vorstellen.

Die spannende Frage lautet nun: muss

Schwerpunkthema

Virtuelle Netze - Zurück zum Soft-Switch?

Fortsetzung von Seite 1

Dipl.-Inform. Damian Lukowski ist seit 2010 Mitarbeiter bei der ComConsult Beratung und Planung GmbH. Zu seinen Schwerpunkten zählen dort die Planung von Virtualisierungsumgebungen und Speicherlandschaften sowie die Bewertung relevanter Technologien im Bereich Data Center.

Werden alle virtuellen Maschinen an die gleiche Ethernet Bridge angeschlossen, sind sie Teil einer gemeinsamen Layer-2 Domäne, und das ist natürlich nicht immer gewünscht. In der Regel werden daher mehrere Ethernet Bridges verwaltet, die eine Teilmenge gehosteter VMs anbinden. Um die Netzsegmentierung auch in das physische LAN zu exportieren, werden entweder unterschiedliche Uplinks genutzt oder die Netze mittels VLAN-Tagging unterschieden.

Nach einiger Zeit hatte man Optimierungspotenzial erkannt, was die Konfiguration der vSwitches und ihrer (VLAN-) Segmentierung betraf. Dies war in erster Linie der Live-Migration und der High-Availability-Funktionalität für virtuelle Maschinen geschuldet. Diese Funktionen setzen voraus, dass eine VM auf einem beliebigen Host eines Clusters lauffähig ist. Für den Betreiber des Clusters bedeutet dies, dass die Netzanbindung aller Hosts einheitlich zu konfigurieren ist – die gleichen Konfigurationsschritte sind für jeden Host und jeden Access-Port (Freischalten der VLANs) durchzuführen.

Unter VMware vSphere stellte der „vNetwork Distributed Switch“ die erste Möglichkeit dar, alle Hosts eines Virtualisierungsclusters homogen zu konfigurieren, ohne die Konfigurationsaufgaben für jeden Host wiederholen zu müssen. Basierend auf der API für den vDS hat Cisco zudem den Nexus 1000v vermarktet, der einen Teil der netzwerkspezifischen Konfigurationsaufgaben wieder in die Verantwortung der Netzwerkadministratoren geben sollte. Über die NX-OS Konsole werden virtuelle Ports (vEth) oder Port-Profile erstellt und bezüglich VLANs, ACLs, QoS, etc. konfiguriert. Der vSphere-Administrator ist lediglich dafür zuständig, eine virtuelle Maschine dem passenden Port-Profil zuzuordnen.

Die bis hierhin beschriebenen vSwitch-Varianten sind reine Software-Konstrukte und mittlerweile nichts Neues. Mit zunehmender Konsolidierung immer mehr virtueller Maschinen auf einzelnen Hosts und der steigenden Verbreitung von 10Gigabit (zukünftig 40G) Ethernet-Adaptern steigt die Wahrscheinlichkeit, dass die theoretisch verfügbare Netzwerkperformance auch real abgerufen werden soll. Das bedeutet insbesondere, dass softwarebasierte Switches eine höhere CPU-Belastung des physischen Hosts nach sich ziehen. (siehe Abbildung 1)

In diesem Zusammenhang existieren einige neue Technologien, die zumindest teilweise damit motiviert sind, den Hypervisor von der Switching-Last zu befreien. Um zu verstehen, welche Verbesserungen diese Neuentwicklungen mit sich bringen, wollen wir uns kurz die Architektur gängiger Soft-Switch-Implementierungen anschauen.

Hypervisor- und Soft-Switch-Architektur

Abbildung 1 stellt eine abstrakte Sicht eines Virtualisierungshosts dar. Virtuelle Maschinen nutzen virtuelle NICs über einen „Front-End“ Treiber. Bei der vNIC kann es sich um die emulierte Version eines echten Ethernet-Adapters handeln (z.B. Intel 82545EM), der über den vom Hersteller angebotenen Treiber genutzt wird oder es handelt sich um einen paravirtualisierten Treiber, der speziell zur Nutzung in virtuellen Umgebungen angeboten wird.

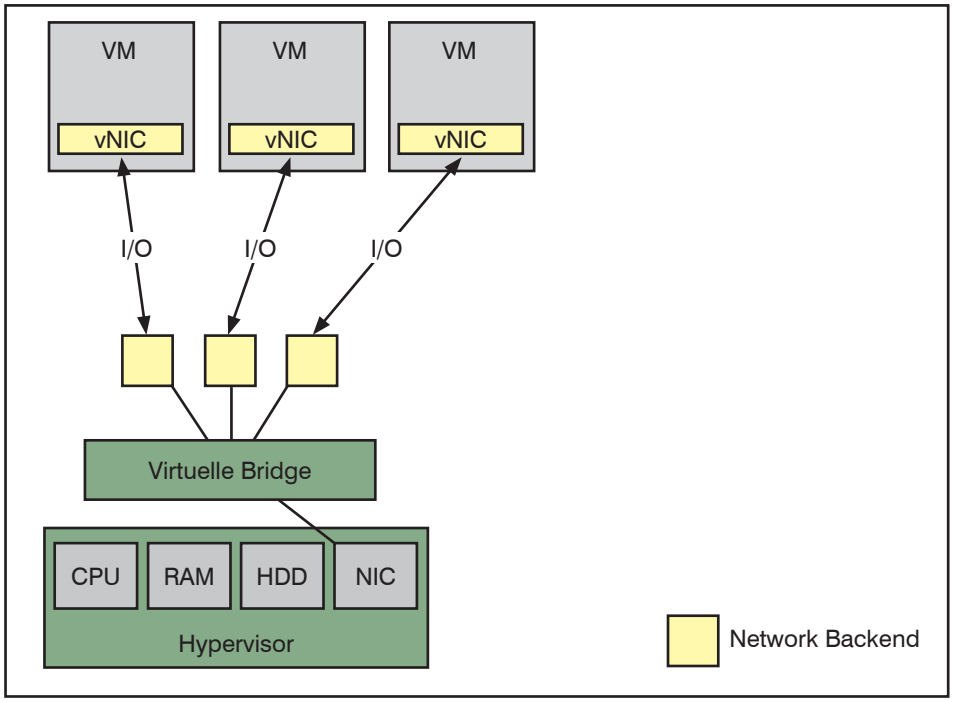


Abbildung 1: Abstraktes I/O Modell bei der Servervirtualisierung

Virtuelle Netze – Zurück zum Soft-Switch?

Zu jeder vNIC gehört ein Gegenspieler im „Back-End“, bildlich die andere Seite des virtuellen Kabels zwischen VM und der virtuellen Bridge. Die heutigen Virtualisierungslösungen unterscheiden sich in ihrer Implementierungsform, insbesondere in der Hinsicht, an welcher Stelle die Elemente innerhalb des punktierten Bereichs (Back-End-Ports und virtuelle Bridge) betrieben werden. Unter Xen und Hyper-V sind die Back-End-Ports, die Bridge und die Treiber der physischen NIC Teil einer privilegierten VM („Domain-0“ bzw. „Parent-Partition“). ESX(i) nutzt eine direktere Architektur und verwaltet Treiber, Back-End und Bridging innerhalb des Hypervisors. KVM folgt einer dritten Variante und verleiht dem Linux-Kernel selbst Hypervisor-Funktionalität, sorgt durch diese Architektur aber immer wieder für hitzige Debatten, ob es sich dabei eigentlich noch um einen Bare-Metal-Hypervisor handelt.

Wie auch immer die konkrete Architektur aussieht, wird die Host-CPU durch das Bridging belastet. Im Falle eines extern einkommenden Datenpakets ist der Host auf einer Low-Level-Ebene mit folgendem Ablauf konfrontiert:

1. Der physische Ethernet-Adapter des Hosts empfängt ein Paket vom angeschlossenen physischen Switch, das für den virtuellen Adapter einer VM bestimmt ist.

2. Der Adapter schreibt das empfangene Ethernet-Paket mittels DMA (Direct Memory Access) in einen vom Hypervisor vorgesehenen Speicherbereich.

3. Der Adapter erzeugt einen Hardware-Interrupt.

4. Ein Host-CPU-Kern unterbricht seine aktuelle Aufgabe und wechselt in den Hypervisor-Kontext („VMX root“ Modus auf VT-x fähigen Hosts), genauer in den entsprechenden Interrupt-Handler. Dieser Handler ist Teil des Hardware-Treibers für den Ethernet-Adapter, bzw. bereitet den Hypervisor darauf vor, die Treiber-Logik zeitnah auszuführen.

5. Das Paket durchläuft den Netzwerk-Stack, die Bridging-Logik und letztendlich den korrekten Back-End-Port. Es findet mindestens einmal ein Kopiervorgang des Pakets in eine Speicherregion statt, auf die die virtuelle Maschine Zugriff hat.

6. Der Hypervisor generiert einen virtuellen Interrupt, um den NIC-Treiber innerhalb der virtuellen Maschine zu aktivieren.

7. Der Host-CPU-Kern kann den Hypervisor-Kontext verlassen, und steht für die Gast-VMs wieder zur Verfügung.

Schritte 1 bis 3 sind Aufgaben, die vom Adapter ohne Zuhilfenahme der CPU übernommen werden können. Der Hardware-Treiber des Ethernet-Adapters programmiert im Vorhinein zukünftige DMA-Transfers mit der „Host Physical Address“, welche dem Paketpuffer des Treibers im Host-Speicher entspricht.

In den Schritten 4 bis 6, insbesondere in Schritt 5, ist ein CPU-Kern mit relativ komplexen Aufgaben betreut, wenn man bedenkt, dass heutige 10G Ethernet-Adapter 250.000+ IOPS, und dementsprechend viele Interrupts pro Sekunde erzeugen können. Diese Komplexität gilt es aufzulösen.

Device Passthrough / Direct-IO

Über sogenanntes Device Passthrough bzw. Direct-I/O ist es möglich, PCI/PCI-e Funktionen an eine virtuelle Maschine durchzureichen, so dass die Funktion exklusiv durch die besagte Maschine genutzt werden wird. Was eine Funktion ist, werden wir später noch genauer spezifizieren.

Abbildung 2 illustriert den Datenpfad zwischen Host-Hardware und virtueller Maschine bei der Nutzung von Direct-I/O, falls es sich bei dem durchgereichten Device um einen Ethernet-Adapter handelt. Es ist anzumerken, dass Device Passthrough nicht auf Netzwerk-Adapter beschränkt ist. Eine essenzielle Voraussetzung für Passthrough ist die Nutzung einer IOMMU (Input/Output Memory Management Unit, z.B. bei Intel VT-d), welche virtuelle Adressen in Host-Adressen umrechnen kann. Wird nämlich der Ethernet-Adapter vom Front-End Treiber innerhalb der VM betrieben, werden DMA-Adressen initialisiert, die das virtuelle Betriebssystem für die tatsächliche physische Adresse hält. Man spricht hier von der „Guest Physical Address“, im Gegensatz zur Host Physical Address, die vom physischen Ethernet-Adapter eigentlich angesteuert werden müsste. Die Zustellung eines Ethernet-Pakets mittels Direct-I/O und VT-d sieht dann wie folgt aus:

1. Der physische Ethernet-Adapter des Hosts empfängt ein Paket vom physischen Switch.

2. Der Adapter initiiert einen DMA-Transfer unter Nutzung der Guest Physical Address, die vom Treiber der Gast-VM programmiert wurde.

3. Die IOMMU übersetzt die Guest Physical Address in eine Host Physical Address; das Ethernet-Paket wird vom Adapter an die Stelle im Host-Speicher geschrieben, die dem Paket-Puffer des virtuellen Treibers innerhalb der virtuellen Maschine entspricht.

4. Der Host-Ethernet-Adapter erzeugt einen Hardware-Interrupt.

5. Ein Host-CPU-Kern unterbricht seine aktuelle Aufgabe und wechselt in den Hypervisor-Kontext.

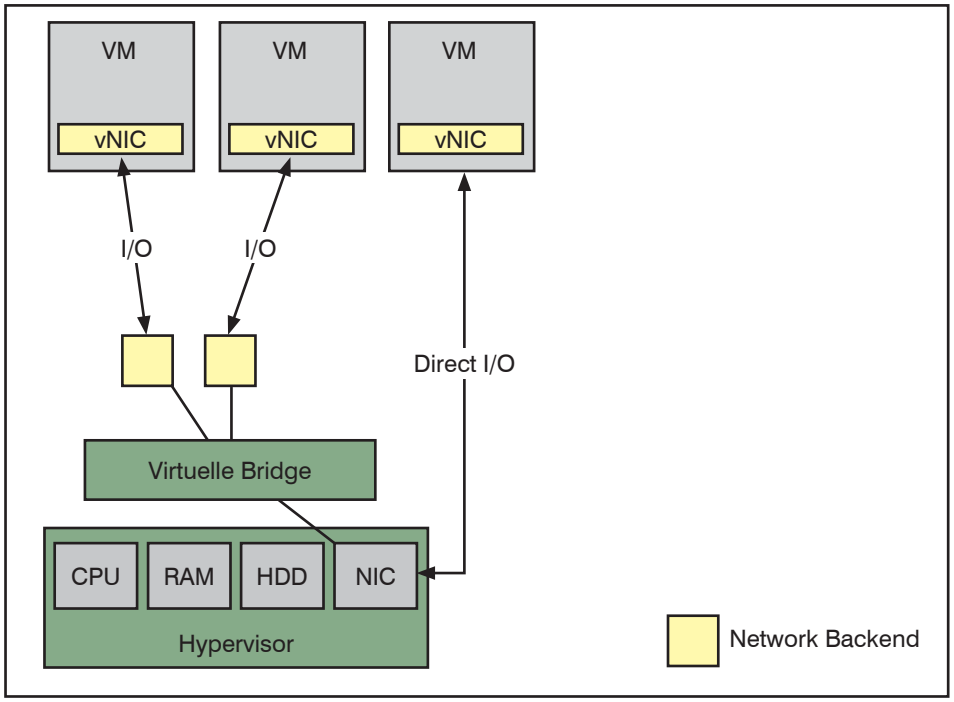


Abbildung 2: vSwitch-I/O im Vergleich zu Direct I/O

Zweitthema

IPv6 und Sicherheit - technische Ansätze, der Weg zum Sicherheits-konzept

Fortsetzung von Seite 1

Dipl.-Inform. Oliver Flüs verfügt über langjährige Kenntnisse im Betrieb von IT-Infrastrukturen. Als Leiter des Competence Center IT-Service der ComConsult Beratung und Planung GmbH bearbeitet er seit Jahren Projekte in den Bereichen Services im IT-Bereich. Zu diesen Themengebieten ist er regelmäßig als Referent bei der ComConsult Akademie tätig, unter anderem als Schwerpunktreferent zu TCP/IP-Aspekten, in der Trouble Shooter-Seminarreihe sowie im Rahmen der Sicherheitsseminare.

Da auch die Verfügbarkeit und damit das störungsfreie Funktionieren von Lösungen mittlerweile als „Sicherheitsgrundwert“ behandelt wird, haben Sicherheitsbeflissene somit keine Zeit mehr, dem Thema IPv6 und Sicherheit noch länger auszuweichen und eine Situation mit günstigeren Rahmenbedingungen abzuwarten. Dabei ist Microsoft nicht etwa besonders „kundenunfreundlich“ mit seiner Strategie, IPv6 als verbindlichen Lösungsbestandteil zu behandeln, sondern (wie so häufig mit Blick auf Umsetzung von RFC-Spezifikationen) nur Vorreiter – weitere Hersteller werden folgen.

Andererseits wurde in dem früheren Artikel an einigen wichtigen Beispielen erläutert, dass es für die Aufgabenstellung „sicherer Umgang mit IPv6“ (noch) keine einfachen Standardantworten gibt. Dies ist teilweise auf die produkteseitige Umbruchsituation zurückzuführen, zum Teil auch darauf, dass manche aus Sicherheitsgründen verlockend erscheinende Ansätze betriebstechnisch nicht unbedingt für jede Umgebung ideal sind (z.B. ein Rückgriff auf „private“ IPv6-Adressen für interne Adressierung).

Insgesamt sind eine Reihe von Aspekten unter Berücksichtigung von Umgebungsbedingungen und IPv6-Fähigkeit präferierter Produktlinien zu beleuchten und zu entscheiden (siehe Abbildung 1).

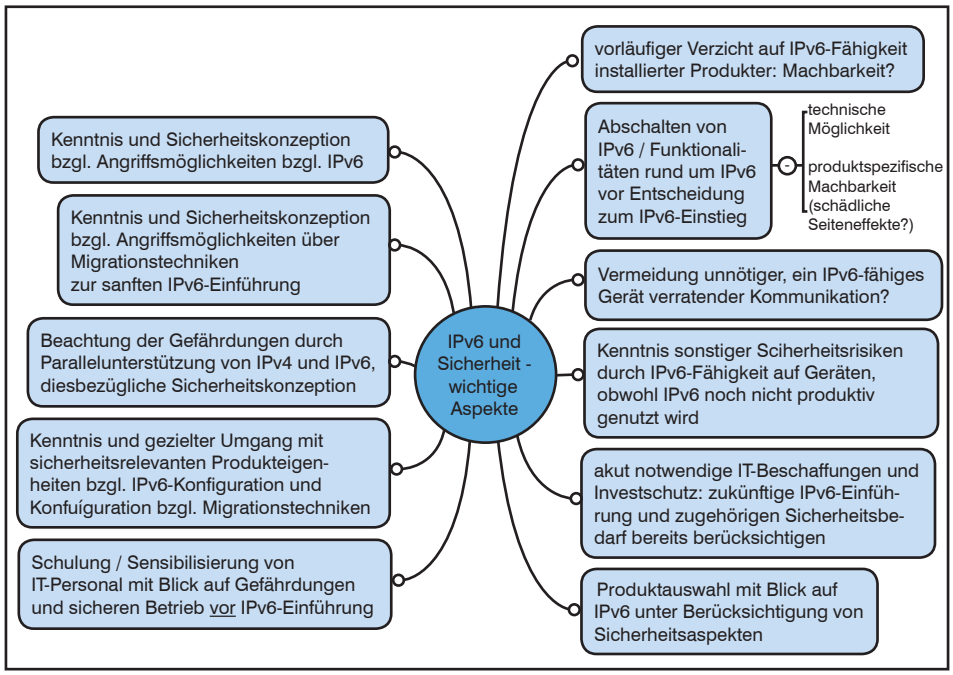


Abbildung 1: Übersicht: wichtige Elemente einer IPv6-Einführung unter Sicherheitsgesichtspunkten

Mit der Aktivierung von IPv6, vorerst typisch parallel zu IPv4 („dual stack-Betrieb“) kommen zu den bekannten Sicherheitsrisiken neue hinzu, teils durch IPv6 an sich, teils durch die parallele Verfügbarkeit von IPv4 und IPv6 – ein Angreifer kann über beide Protokollwelten ein Gerät schlimmstenfalls „in die Zange“ nehmen. Abschalten hat aber auch seine Gefahren (Seiteneffekte bzgl. Verfügbarkeit), und die Hersteller von Produkten mit Sicherheitsfunktionalitäten können auch noch nicht am Ende der Entwicklung notwendiger Gegenmaßnahmen sein: Dazu fehlt noch die ausreichend lange Praxisphase mit IPv6, während der so langsam alle grundlegenden „Schweinereien“ bzw. Implementierungsschwächen ausprobiert, entdeckt und durch Gegenmaßnahmen entschärft werden konnten.

Also: Man kann nur verlieren und ist hilflos?

Ganz so ist es nicht, auch wenn der Weg zu einem „umfassenden“ IPv6-Sicherheitskonzept und dessen erfolgreicher Umsetzung mühselig und sein Ende noch nicht absehbar ist. Entsprechend wäre die Ankündigung, mit dem vorliegenden Artikel ein „best practice“-Sicherheitskonzept zu IPv6 anzubieten, natürlich Schwindelei. Eindrücke, Beispiele und über diese auch erste konkrete Informationen für die IPv6-Sicherheitspraxis können und sollen aber gegeben werden.

Die Aufgabe wird sein, schrittweise unter Berücksichtigung jeweils neuester Erkenntnisse zu Angriffsformen, deren Ansätzen, Ideen zur Erkennung oder gar

IPv6 und Sicherheit – technische Ansätze, der Weg zum Sicherheitskonzept

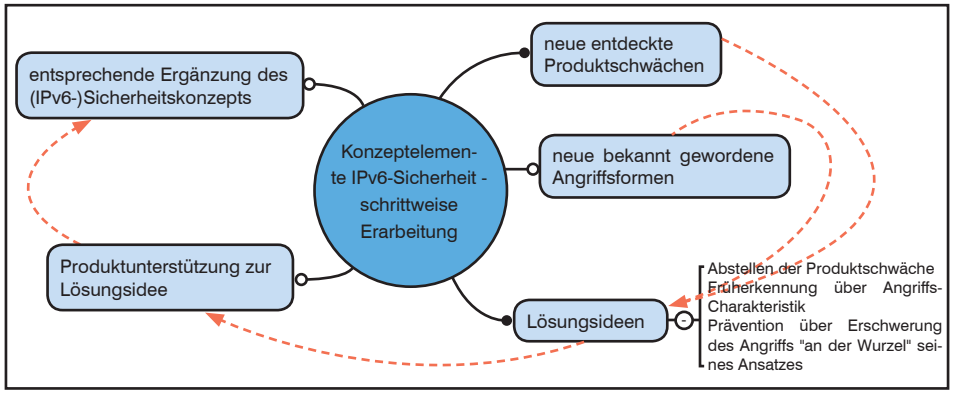


Abbildung 2: Typische Abfolge von Teilschritten auf dem Weg zu ausreichender IPv6-Sicherheit

Prävention und entsprechender Machbarkeit auf Produktebenen immer neue Puzzle-Steine zum Sicherheitskonzept zu legen, bis mittelfristig ein geeignet umfassender Schutz aufgebaut ist (siehe Abbildung 2).

Diese vorerst eher reaktive Erarbeitung eines IPv6-Sicherheitskonzepts wird im Zuge einer schrittweisen Einführung von IPv6 mehrfach anfallen. Dabei heißt es, aufmerksam zu bleiben und jeweils mit Ideen auf neue Herausforderungen zu reagieren.

Auch wenn dies mühselig ist und die Produktsituation jeweils zumindest anfangs aus Sicherheitsperspektive noch Wünsche übrig lassen wird, heißt dies nicht, dass man nichts tun kann.

Die nachfolgend diskutierten Beispiele sollen zeigen, dass man ohne große Genialität sowohl Ideen für Sicherheitsvorkehrungen mit Blick auf neue Angriffsformen bzw. Schwachstellen entwickeln als auch ganz konkrete Produktanforderungen daraus ableiten kann – bzw. zumindest mit gegebenen Produktmöglichkeiten Übergangslösungen finden kann, die das Risiko zumindest reduzieren.

Beispiel 1: IPv6 ist das Ende von „ping sweep“? Nicht ganz, aber …

Das erste Beispiel beginnt dabei scheinbar nicht mit einer Sicherheitsfrage, sondern einer Antwort auf eine altbekannte Frage: Wie verhindert man das Ausprobieren von Adressen als ersten Angriffsschritt? Häufig ist ein Auskundschaften einer Umgebung ein erster wichtiger Vorbereitungsschritt im Rahmen eines Sicherheitsangriffs; gelingt es, einem Angreifer bereits an so früher Stelle das Leben schwer zu machen, lässt er womöglich von dieser Umgebung ab, oder muss sich zumindest so auffällig verhalten, dass man große Chancen hat, ihn zu entdecken, ehe er mit den nächsten Angriffsschritten fortfahren kann.

```
RFC 4291              IPv6 Addressing Architecture          February 2006

   For all unicast addresses, except those that start with the binary
   value 000, Interface IDs are required to be 64 bits long and to be
   constructed in Modified EUI-64 format.
```

Abbildung 3: Auszug aus RFC 4291 (Standards Track) bzgl. Interface ID-Länge

Eine unter IPv4 übliche Methode der Informationsgewinnung über ein Angriffsziel ist das systematischen Durchführen von Ping-Versuchen an aufeinanderfolgende Adressen („ping sweep“) mit dem Ziel, aktive IP-Adressen ausfindig zu machen und diese dann mit weiteren gezielten Testschritten bzgl. verwendeten Produkten und potenziellen Schwachstellen weiter einzugrenzen. Im Falle von IPv6 scheint dieser brute-force-Ansatz auf den ersten Blick wenig zielführend: Eine Möglichkeit der Adresserzeugung für ein Gerät unter IPv6 ist Autoconfiguration, bei der zu einem gegebenen „Netzwerk-Präfix“ die zur 128-Bit-Adresse fehlenden Bits des „Interface Identifiers“ von der IP-Software des Geräts selbst berechnet werden. Ein Ansatz dabei ist das Überführen der Layer 2-Adresse in den Interface Identifier, wobei Layer 2-Adressen von bis zu 64 Bit-Länge als Ausgangspunkt zu unterstützen sind (Stichwort: modified EUI64-Adressen). Um die gegebenenfalls 64 Bit der Layer-2-Adresse dabei vollständig verwerten zu können, ist etablierte Konvention (ursprünglich sogar: verbindliche Anforderung), dass ein Interface Identifier stets 64 Bit zu umfassen habe (siehe Abbildung 3).

Na und (mal abgesehen davon, dass diese Forderung nicht als „MUST“ formuliert und daher nicht in Stein gemeißelt ist)? Nun, setzt man diese RFC-Passage in die Praxis um, damit der Weg für Autoconfiguration mit modified EUI-Adressen in jedem Fall offen gehalten wird, so hat dies zur Konsequenz, dass jedes IP-Teilnetz die Größenordnung von 2^{64} Adressen aufweist. Wer einen solchen Vorrat der Reihe nach mittels Ping durchtesten will, verliert eine Menge Zeit und setzt sich dabei entsprechend lange der Gefahr aus, entdeckt zu werden.

Allerdings, wenn man sich genauer mit dem Thema beschäftigt, ist der Ping-Sweep-Versuch doch nicht so „sinnlos“ – statt einfach durchzuprobieren, kann man nämlich auch systematisch vorgehen. Die Systematik besteht darin, die praxisrelevanten Alternativen der Generierung von IPv6-Adressen durchzugehen und hierdurch die mit hoher Wahrscheinlichkeit in Frage kommenden Adressen von weniger wahrscheinlichen abzugrenzen. Beispiele:

- Geht man davon aus, dass auf Layer 2 Ethernet verwendet wird, sind die ersten 24 Bit der MAC-Adresse und damit die in Frage kommenden Bitmuster zu Beginn einer EUI64-Interface ID nicht mehr willkürlich, sondern durch die als „Herstellerkennungen“ im Ethernet-Bereich verwendeten Bitfolgen beschränkt.

- Diesen Ansatz über führende „Hersteller-Bits“ den vorderen Teil der Interface ID einzugrenzen, kann man als Angreifer noch gezielter ausnutzen, wenn man es auf bestimmte Gerätetypen abgesehen hat, bei denen man „autokonfigurierte Adressen“ erwartet, z.B. Notebooks oder Drucker.

- Werden Adressen aus einem Pool mittels DHCPv6 zugeteilt („stateful DHCPv6“), und die DHCP-Server-Software vergibt aus den freien Adressen „fortlaufend“, so genügt die Kenntnis einer ersten so verwalteten Adresse, um von dort aus systematisch auf- bzw. absteigend weiter zu probieren.

So oder ähnlich lässt sich zumindest für nach einer bekannten Systematik erzeugten bzw. festgelegten Adressen der Umfang der erfolgversprechenden Testkandidaten für Ping-Sweep durchaus signifikant eingrenzen und leider wieder mit erträglichem Aufwand ein Ping-Sweep als erster Schritt zur Informationsgewinnung über eine „Opfer-Umgebung“ durchführen – man versetzt sich in eine systematisch