André König, Ralf Steinmetz: *Identifikation von Migrationsprozessen virtueller Maschinen*. Presented at 5. Essener Workshop "Neue Herausforderungen in der Netzsicherheit" (EWNS 2011), April 2011.

# Identifikation von Migrationsprozessen virtueller Maschinen

TECHNISCHE UNIVERSITÄT DARMSTADT

**André König, Ralf Steinmetz — Fachgebiet Multimedia Kommunikation (KOM)**
{andre.koenig, ralf.steinmetz}@kom.tu-darmstadt.de

Rechnervirtualisierung, also das Einführen einer zusätzlichen Abstraktionsschicht (Hypervisor) zwischen Hard- und Software, bildet die Grundlage aktueller Anwendungen wie Cloud Computing und energieeffizientem Betrieb von Netzen und Rechenzentren. Hypervisor-Systeme wie Proxmox [1] ermöglichen den parallelen Betrieb mehrerer virtueller Maschinen auf dem selben physikalischen Rechner. Durch die Entkopplung von Hardware und Software können virtuelle Maschinen im laufenden Betrieb zwischen physikalischen Rechnern migriert werden. Aus Sicherheitssicht bieten Virtualisierung und Migration virtueller Maschinen neue Möglichkeiten zur Realisierung von Sicherheitsmechanismen, aber auch zur Durchführung von Angriffen [2, 3, 4]. Neben Angriffen auf Ebene des Hypervisors müssen, insbesondere im Kontext der Migration virtueller Maschinen, Angriffe auf Ebene des Kommunikationsnetzes betrachtet werden. Eine wichtige Basis sowohl für Angriffe auf, als auch für Sicherheitsmechanismen für virtualisierte Umgebungen ist dabei das Erkennen von Migrationsprozessen. Sicherheitsmechanismen können anhand dieser Information entscheiden, ob ein Migrationsprozess legitim ist, oder ob z.B. ein unberechtigtes Abbild einer virtuellen Maschine erstellt wird. Angriffsmechanismen können das Wissen über einen ablaufenden Migrationsprozess nutzen, um diesen gezielt, z.B. durch einen Denial of Service-Angriff zu beeinflussen. Zusätzlich zur Erkennung im migrierenden Hypervisor selbst ist eine Erkennung ausserhalb des Hypervisors sinnvoll, um z.B. Migrationsprozesse eines kompromittierten Hypervisors erkennen zu können.

Dieser Beitrag konzentriert sich auf grundlegende Schritte zur Erkennnung von Migrationsprozessen virtueller Maschinen ausserhalb des entsprechenden Systems und des Hypervisors. Vorgestellt werden erste Untersuchungen mit Hilfe der German-Lab Experimentalplattform [5]. Ergebnisse aus verschiedenen Szenarien, in denen die Migration eines Linux Systems im laufenden Betrieb durch den Proxmox Hypervisor betrachtet wurde, zeigen dass die Roundtrip Time von ICMP Paketen eine vielversprechende Metrik zur Erkennung von Migrationsprozessen ist. In weiteren Schritten muss bestimmt werden, welche zusätzlichen Metriken benötigt werden, um eine eindeutige Erkennung zu ermöglichen.

## Verwandte Arbeiten

Verwandte Arbeiten im Bereich der Sicherheit in virtualisierten Rechnerumgebungen datieren zurück bis in das Jahr 1976. Attanasio et al. stellen in [6] eine Sicherheitsanalyse des IBM VM/370 Hypervisors vor. Darauf aufbauend beschreiben Gold et al. in [7] Erweiterungen zur Absicherung von VM/370. Angriffsmöglichkeiten auf aktuelle Hypervisor-Systeme präsentiert Ferrie in [8].

Hypervisor-basierte Malware (Virtual Machine Based Rootkits) wie SubVirt [9] nutzt Rechnervirtualisierung um Systeme illegitim, z.B. zur Vorbereitung weiterer Angriffe, in eine virtuelle Maschine zu migrieren. Dem entgegen stehen Ansätze wie GuardHype [10], die als 'Hypervisor for Hypervisors' den Zugriff eines Hypervisors auf den physikalischen Rechner kontrollieren.

Verwandte Arbeiten zur Entscheidung ob ein System in einer virtuellen Maschine läuft, wie z.B. von Quist et al. in [11] beschrieben, basieren im Allgemeinen auf dem Auslesen von Interrupt- oder Speichertabellen. Diese Unterscheiden sich je nachdem, ob das System in einer virtuellen Umgebung oder direkt auf dem physikalischen Rechner ausgeführt wird. Zwar kann so (nachträglich) festgestellt werden, ob ein System, das bisher auf einem physikalischen Rechner ausgeführt wurde, in eine virtuelle Maschine verschoben wurde; eine Migration von virtuellen Maschinen zwischen physikalischen Rechnern kann aber nicht erkannt werden. Direkt verwandte Arbeiten zur Erkennung von laufenden Migrationsprozessen (ausserhalb des betroffenen Systems) existieren nach bestem Wissen der Autoren nicht.

## Identifikation von Migrationsprozessen

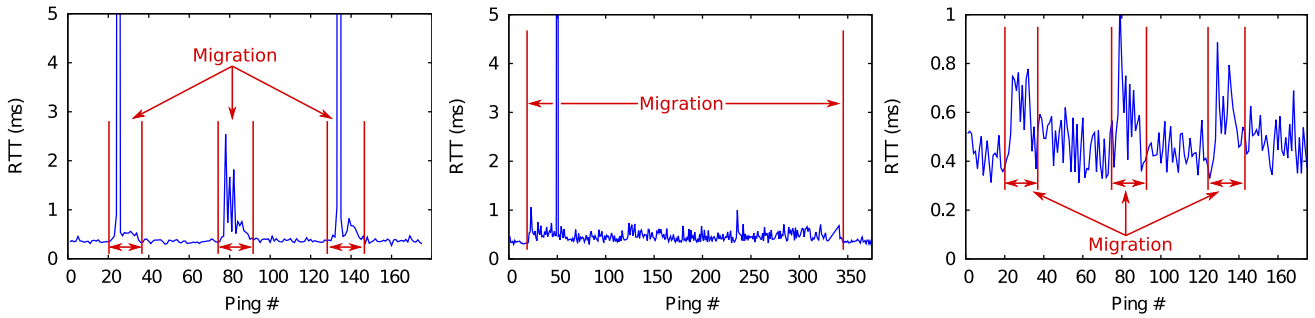
Erste Versuche zur Erkennung von Migrationsprozessen wurden mit Hilfe der German-Lab Experimentalplattform durchgeführt. Auf fünf identisch ausgestatteten SUN Fire X4150 Servern wurde dafür ein Proxmox 1.7 Cluster eingerichtet. Die Server waren über einen Cisco 4500 L3 Series Switch mit einer Bandbreite von 1 Gb/s verbunden. Innerhalb des Clusters wurde eine virtuelle Maschine mit einem 32-bit Ubuntu 10.10 Desktop System installiert, das während der Versuche zwischen Server 1 und 2 migriert wurde. Die virtuelle Maschine war ausgestattet mit einem Prozessorkern, 512 MB Speicher, 8 GB IDE Festplatte und einer rtl8139-basierten Netzwerkkarte mit direktem (TAP) Zugriff auf das Netz des Hypervisors; der Migrationsprozess wurde manuell über das Proxmox Webinterface ausgelöst. Auf Server 3 befand sich eine identische virtuelle Maschine, die als NFS-Server für die (virtuelle) Festplatte der zu migrierenden Maschine diente.

Vor, während und nach dem Migrationsprozess wurde mittels ICMP die Roundtrip Time zwischen der migrierten virtuellen Maschine und einer identischen virtuellen Maschine, die auf Server 4 installiert war, ermittelt. Dafür wurden ICMP Pakete mit einer Größe von 64 Byte in einem Abstand von 0.5 Sekunden gesendet.

Um, im Hinblick auf geplante Versuche zur Migration virtueller Maschinen über die Grenzen lokaler Netze, die Auswirkungen beschränkter Bandbreiten in Weitverkehrsnetzen zu untersuchen, wurde in einer virtuellen Maschine auf Server 5 ein Emulator für Weitverkehrsnetze (WANem 2.3 [12]) installiert. Die Kommunikation zwischen Proxmox auf Server 1 und 2, also die Migration der virtuellen Maschine, wurde durch Anpassen der Routing-Tabellen des Hypervisors über die WANem virtuelle Maschine geleitet.

### Migration in einem unbelasteten Netz mit 1 Gb/s Bandbreite

Abbildung 1a zeigt die Roundtrip Time für die gesendeten ICMP Pakete zwischen der virtuellen Maschine auf Server 4 und der virtuellen Maschine, die zwischen Server 1 und 2 migriert wurde für ein ansonsten unbelastetes Netz mit der vollen zur Verfügung stehenden Bandbreite von 1 Gb/s. Der Migrationsprozess wurde jeweils zu ICMP Paket Nr. 20, 75 und 130 ausgelöst. Die Auswirkungen der Migration auf die Roundtrip Time im laufenden Betrieb der migrierten virtuellen Maschine sind deutlich erkennbar. Auffällig ist der hohe Aussschlag der Roundtrip Time bis zu 160ms (aus Gründen der Darstellung in der Grafik nicht erkennbar) beim ersten und dritten Migrationsprozess. Dieser Höchswert entsteht während die Prozessoraktivität der virtuellen Maschine zwischen Server 1 und 2 verschoben wird. Auch bei




**(a)** Unbelastetes Netz, 1 Gb/s Bandbreite **(b)** Unbelastetes Netz, 10 Mb/s Bandbreite **(c)** 10 Mb/s Hintergrundverkehr, 1 Gb/s Bandbreite

**Abbildung 1:** Ping-Profil von Online-Migrationsprozessen eines Ubuntu 10.10 Systems in Proxmox

zweiten Migrationsvorgang ist ein solcher Ausschlag, allerdings mit geringerer Ausprägung erkennbar. Dies lässt sich durch die mit 0.5 Sekunden Abstand zwischen den ICMP Paketen relativ geringe Abtastrate des Migrationsprozesses erklären. Für alle drei Migrationsprozesse ist eine erhöhte Roundtrip Time während des Gesamtprozesses erkennbar.

#### Migration in einem unbelasteten Netz mit 10 Mb/s Bandbreite

Abbildung 1b zeigt die analog zum ersten Versuch gemessene Roundtrip Time für ein ansonsten unbelastetes Netz mit einer (emulierten) beschränkten Bandbreite von 10 Mb/s zwischen Server 1 und 2. Im Gegensatz zum ersten Versuch ist nur ein Migrationsvorgang dargestellt. Deutlich erkennbar ist die im Vergleich zu einer unbeschränkten Bandbreite stark verlängerte Migrationsdauer. Diese verlängert sich jedoch nicht umgekehrt proportional zum Begrenzungsfaktor (0.1) der Bandbreite von etwa 10 auf 1000 Sekunden im Vergleich zum ersten Versuch, sondern bleibt mit etwa 160 Sekunden deutlich unter diesem Wert. Wie auch im ersten Versuch ist ein starker Ausschlag der Rountrip Time während der Migration der Prozessoraktivität erkennbar.

#### Migration mit 10 Mb/s Hintergrundverkehr und 1 Gb/s Bandbreite

Abbildung 1c zeigt die analog zu den ersten beiden Versuchen gemessene Roundtrip Time für ein Netz mit unbeschränkter Bandbreite und 10 Mb/s Hintergrundverkehr. Der Hintergrundverkehr wurde durch ICMP Pakete mit einer Größe von 12500 Byte erzeut, die in einem Intervall von 0.01 Sekunden direkt vom Hypervisor auf Server 2 an den Hypervisor auf Server 1 geschickt wurden. Wie im ersten Versuch wurde der Migrationsprozess jeweils zu ICMP Paket Nr. 20, 75 und 130 ausgelöst. Auch in der im Vergleich zu den ersten beiden Versuchen in einem unbelasteten Netz erhöhten Roundtrip Time auch ausserhalb der Migrationsphasen ist die Auswirkung der Migrationsphasen noch gut erkennbar. Der Ausschlag der Roundtrip Time während der Migration der Prozessoraktivität ist beim zweiten Migrationsvorgang deutlich erkennbar. Die geringere Ausprägung bzw. das Fehlen des Ausschlags bei den anderen beiden Migrationsprozessen wird auf die geringe Abtastrate zurückgeführt.

### Literatur


[1] Proxmox Homepage. *http://www.proxmox.com*.

[2] T. Garfinkel et al.. When Virtual is Harder than Real: Security Challanges in Virtual Machine Based Computing Environments. In *Proc. of HotOS '05*.

[3] M. Price. The Paradox of Security in Virtual Environments. *IEEE Computer Magazine*, 41:22 – 28, 2008.

[4] Doug Hyde. A Survey on the Security of Virtual Machines. Tech. report, Dept. of Computer Science & Engineering, Washington University in St. Louis, 2009.

[5] G-Lab Project Homepage. *http://www.german-lab.de*.

[6] C. R. Attanasio et al.. Penetrating an operating system: a study of VM/370 integrity. *IBM Systems Journal*, 15:102 – 116, 1976.

[7] B. D. Gold et al.. A security retrofit of VM/370. In *Proc. of the National Computer Conference*, 1979.

[8] P. Ferrie. Attacks on Virtual Machine Emulators. In *Proc. of AVAR '06*.

[9] S. T. King et al.. SubVirt: Implementing malware with virtual machines. In *Proc. of S&P '06*.

[10] M. Carbone et al.. Taming Virtualization. *IEEE Journal on Security and Privacy*, 6:65 – 67, 2008.

[11] D. Quist et al.. Further Down the VM Spiral - Detection of full and partial emulation for IA-32 virtual machines. http://www.offensivecomputing.net, 2006.

[12] M. Nambiar, H. K. Kalita, D. Mishra, and S. Rane. WANem: The Wide Area Network Emulator. *http://wanem.sourcefourge.net*, 2011.