

Ampel-Schaltung

Schon fast für tot erklärt, erleben Application Level Gateways derzeit eine Renaissance. Nur mit dieser Technik ausgestatteten Firewalls gelingt es, die vielen Dienste zu unterscheiden, die beispielsweise über Port 80 tunnelt. Mit der Zorp-Firewall regelt der Admin sehr detailliert, welche Verbindungen er erlaubt. Christian Ney



Firewall-Hersteller entdecken einen alten Trend neu: Firewalling auf Applikationsebene per Application Level Gateway (ALG), im Marketing-Jargon auch als Intrusion Prevention bekannt. Bereits die ersten Internet-Firewalls nutzten diese Technik, darunter das 1993 veröffentlichte TIS FWTK (Firewall-Toolkit) [7]. Später waren Firewalls jedoch vor allem als Paketfilter aufgebaut (siehe **Kasten „Firewall-Techniken“**).

Heute nennt Hersteller Checkpoint seine ALG-Implementierung Application Intelligence oder kurz AI [8], bei Juniper/Netscreen heißt sie Deep Inspection [9], Cisco verwendet die Bezeichnung Context-Based Access Control [10]. In diesen illustren Kreis dringt die ungarische Firma Balabit mit ihrer Linux-basierten Zorp-Firewall [1] ein. Sie kombiniert den Schutz eines Paketfilters mit einer Sammlung von Proxy-Diensten, die sich – für die Beteiligten unbemerkt – in den Datenstrom einklinken.

Allen ALG-Firewalls gemeinsam ist eine erheblich erweiterte Kontrolle über den Netzwerkverkehr. So kann der Admin detailliert festlegen, welche Kommunikation erlaubt und welche unerwünscht ist. Anders als bei Paketfiltern erstreckt sich die Kontrolle auch auf den Inhalt der Pakete. Ein ALG bestimmt nicht nur, wer mit wem über welches Protokoll kommunizieren darf, sondern auch, welche Kommandos und Daten sich beide zusenden dürfen.

Eine ALG-Firewall erkennt zum Beispiel überlange HTTP-Requests, die einen Buffer Overflow im Webserver auslösen könnten. Auch die Nachfrage nach »cmd.exe«, hervorgerufen vom Code-Red-Wurm, kann ein ALG erkennen und blocken. Oder es lässt alle übertragenen Daten von einem Virens Scanner prüfen.

Modulare Proxys

Balázs Scheidler ist vielen Admins und Linux-Magazin-Lesern durch sein Syslog-NG [11] bekannt. Mit Zorp [1] schuf er ein komplexes ALG. Seine Firma Balabit charakterisiert Zorp als modulares Application Level Gateway, bei dem nicht zwingend für jedes Protokoll ein eigener Proxy zuständig ist. Zorp verschachtelt Dienste ineinander, um getunnelte Protokolle zu behandeln. Beispielsweise ist HTTPS gewöhnliches HTTP, das über eine SSL-Verbindung läuft. Ein Zorp-Modul kümmert sich um SSL, gelangt trickreich an den getunnelten Datenstrom und übergibt ihn an das HTTP-Modul.

Zorp arbeitet mit Hilfe eines Kernelpatch als transparenter Proxy. Clients kommen daher ohne spezielle Einstellungen aus, die Firewall leitet den Verkehr selbst-

ständig zum passenden Proxy um. IP-Tables-Kenner mögen einwenden, dass das Umleiten eines Ports auch ohne Patch gelingt. Das DNAT-Target funktioniert allerdings nur, wenn die Ziel-IP zur Firewall-Maschine gehört.

Ein Prozess kann mit Linux-Bordmitteln allein nicht auf eine fremde IP-Adresse horchen und er kann auch keine Verbindung mit fremder Absenderadresse starten. Ein wirklich transparenter Proxy braucht jedoch beides, damit er von Client und Server nicht bemerkt wird. Die Balabit-Entwickler haben daher das TProxy-Patch [3] entwickelt und unter der GPL veröffentlicht. Die kommer-

Zorp-Firewall



Varianten: Zorp Professional und Enterprise (beide kommerziell) sowie Zorp GPL

Getestete Version: Zorp Professional 3.0.1

Besonderheit: Komplexe Firewall-Architektur mit Paketfilter und transparenten Proxy-Diensten, die dem Admin sehr weit reichende Kontrolle über den Netzwerkverkehr gibt.

Unterschiede: Die kommerziellen Varianten enthalten mehr Proxys sowie ein grafisches Admin-Interface, Enterprise zudem den ZAS. Ein High-Availability-Modul ist optional.

Hersteller: Balabit, Ungarn, [<http://www.balabit.com>]

Lizenzen: Zorp GPL steht unter der GPL. Für Zorp Professional und Enterprise siehe **Kasten „Lizenzierung und Preise“**.

zielle Zorp-Version wird als installierbare CD ausgeliefert. Zusätzlich ist ein Lizenzschlüssel in Form einer Datei erforderlich, die der Lizenz unter anderem ein Haltbarkeitsdatum verpasst.

Ein Installationsprogramm bringt das Zorp Operating System auf die Festplatte, es entpuppt sich als abgespecktes und gehärtetes Debian Woody. Als Paketfilter kommt Netfilter [4] zum Einsatz, zudem enthält der Kernel einige Patches: TProxy für transparente Proxy-Dienste [3], Openwall für zusätzliche Sicherheitsfeatures [12] sowie den Free-swan-Nachfolger Openswan [13] für VPN-Dienste. Für das Logging nutzt Babit das hauseigene Syslog-NS [11].

Zorp GPL

Der wichtigste Unterschied zur GPL-Version (siehe **Kasten „Zorp GPL“**) ist der Aufbau des Systems: Zorp Professional gliedert sich in vier Teile (siehe **Abbildung 1**): Firewall, Zorp Management Server (ZMS), Zorp Authentication Server (ZAS) und Zorp Management Client (ZMC). Per ZMS verwaltet der Admin fast alle Aspekte einer oder mehrerer Firewalls. Der ZMS eignet sich hervorragend, um verteilte Firewallumgebungen zentral zu verwalten; dieses Feature ist sonst nur bei den Großen der Branche zu finden. Darüber hinaus agiert der ZMS als Certificate Authority für die verwalteten Systeme.

Die CA-Funktion ist sinnvoll, da die gesamte Kommunikation zwischen den

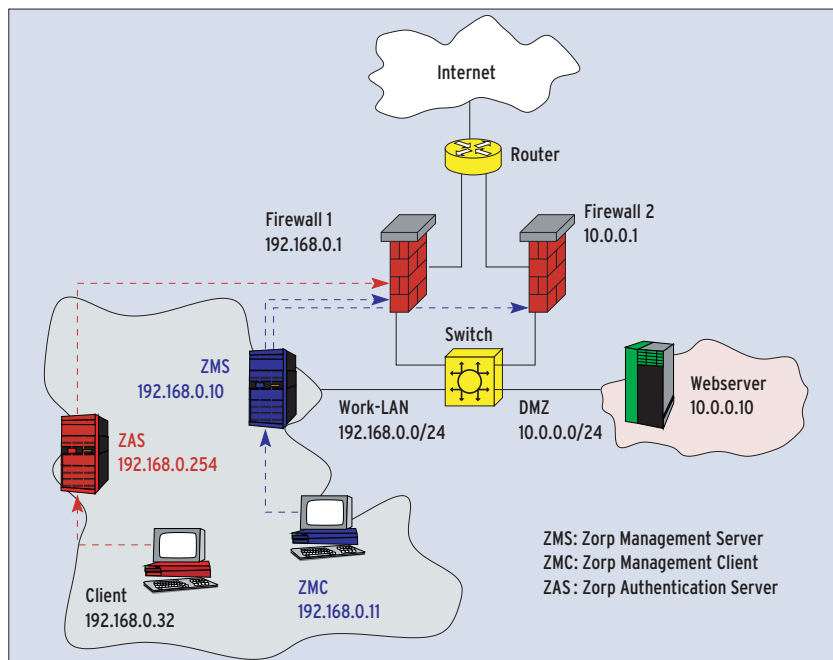


Abbildung 1: Der Admin verbindet sich per ZMC mit dem ZMS, um die Firewalls zu konfigurieren. Firewall 1 gewährt dem LAN Zugriff auf Internet und DMZ, die Clients müssen sich aber zunächst am ZAS authentifizieren. Firewall 2 ist über den Switch per VLAN mit der DMZ verbunden und schützt den Webserver vor Angriffen.

Zorp-Komponenten SSL-verschlüsselt abläuft. Zudem erstellt der ZMS Zertifikate für Anwender, mit deren Hilfe sie sich am ZAS anmelden. Der ZAS kann User dazu zwingen, sich erst zu authentifizieren, bevor sie einen Dienst nutzen. Neben X.509-Zertifikaten können sie auch Passwörter (oder Cryptocard RB-1 oder S/Key) verwenden. ZAS lässt sich auch in bestehende Radius-, LDAP- oder TACAS-Umgebungen einbinden.

Der ZMC (siehe **Abbildung 3**) rundet das Zorp-Paket ab. Mit dieser grafischen

Applikation greift der Admin auf den ZMS zu. Erhältlich ist der ZMC als Debian-Paket und als Windows-Applikation. Das GUI ist ausreichend flexibel, um fast jeden Aspekt der Firewall zu verwalten; angefangen bei Paketfilter- und Proxy-Regeln über die Mailserver-Konfiguration, IPsec und Netzwerkkonfiguration bis zum Systemlogging. Selbst für einen Neustart oder das Herunterfahren des Systems darf der Admin beim Frontend bleiben und muss sich nicht remote einloggen. ▶

Firewall-Techniken

Einfache Paketfilter beziehen ihre Informationen aus dem IP-Header. Er enthält Informationen über Herkunft und Ziel des Pakets. Um Dienste zu unterscheiden (beispielsweise Zielport 80 für Webserverzugriff), lesen sie zusätzlich einen Teil des TCP- oder UDP-Headers. Diese Daten dienen als Entscheidungsgrundlage, um die Kommunikation auf Quell- und Ziel-Adressen und -Ports einzuschränken. Die IPChains [5] im Linux-Kernel 2.2 implementieren diese einfache Technik.

Erweiterte Paketfilter wie der Netfilter im Linux-Kernel 2.4 [4] sind zusätzlich in der Lage, Verbindungen zu erkennen. Sie werten dazu die Flags der TCP-Header aus, um auf Verbindungs-Auf- und -Abbau zu schließen. Die Erkenntnisse über den Zustand einer Verbindung speichert Netfilter, um spätere Pakete

korrekt zuzuordnen. Diese Technik nennt sich „stateful“. Weil erweiterte Paketfilter unterscheiden, ob das Paket Teil einer erlaubten und bereits bestehenden Verbindung ist, muss das Regelwerk nur die Kommunikation in einer Richtung gestatten. In einfachen Paketfiltern müssen weitere Regeln die Antwortpakete erfassen.

Application Level Gateways

Moderne Paketfilter-Varianten prüfen sogar den Inhalt einzelner Pakete gegen frei definierbare Muster (Pattern). Im Idealfall betrachten sie zusätzlich die Folge der Pakete und deren Inhalt während einer Verbindung. So lassen sie sich nicht von gewitzten Angreifern austricksen, die schädliche Daten eines Angriffs auf mehrere Pakete verteilen. Ähnlich

wie Virens Scanner und Intrusion-Detection-Systeme hängt die Zuverlässigkeit des Paketfilters von aktuellen Pattern ab, die auch neuere Angriffstechniken beschreiben.

Statt nur zu filtern bietet es sich an, die Verbindung über einen Proxy zu leiten. Der arbeitet als Stellvertreter des Clients, nimmt dessen Anfrage entgegen und öffnet selbst eine zweite Verbindung zum Zielrechner. Die Netze vor und hinter dem Proxy dürfen getrennt bleiben. Die Firewall muss kein Forwarding durchführen, der Proxy leitet die Daten weiter. Er analysiert nicht einzelne Pakete, sondern den Datenstrom, der durch ihn fließt. Daher ist für jedes Protokoll ein eigener Proxy nötig. Der Vorteil: Ein Proxy kennt sein Protokoll genau und kann daher schädlichen Inhalt erkennen und gezielt aussortieren.

Kümmern sich mehrere Admins um die Firewall, dann sorgt der ZMS dafür, dass immer nur einer Änderungen vornimmt. Ist bereits ein Admin per ZMC angemeldet, lässt der ZMS keine weitere Verbindungen zu und vermeidet so Inkonsistenzen. Als sehr praktisch erweisen sich die Funktionen zum Testen der Änderungen. Da alle Konfigurationen im Klartext vorliegen, kann der ZMC sie auch anzeigen (Abbildung 2).

Die kommerzielle Version enthält erheblich mehr Proxys als die GPL-Variante (siehe **Kasten „Zorp GPL“**), neben FTP, HTTP, SSL, Telnet, Finger und Whois auch POP3, NNTP, IMAP, Printer, Radius, TFTP, PGSQL, Oracle Net8, LDAP sowie das generische »plug«-Plugin. Darüber hinaus sind so genannte native Proxydienste auf spezielle Anwendungsfälle zugeschnitten, etwa Postfix als gehärtetes SMTP-Gateway, ein angepasster Bind als DNS-Server und ein abgespeckter »ntpd«, der auf Wunsch als Zeitserver für das interne Netz dient.

Strikt RFC-konform

Die Proxydienste sind nicht durch einfache Regeln definiert, sie bemühen sich vielmehr um strikte RFC-Konformität. Anders als vergleichbare Produkte lassen sich die Zorp-Module recht leicht an spezielle Bedürfnisse anpassen. Derartige Aktionen in Eigenregie durchzuführen ist aber nicht zu empfehlen. Soll ein Proxy nicht-RFC-konforme Protokollläufe erlauben, etwa weil ein Anwender fehlerhafte Software einsetzen muss, dann implementiert Balabit diese Funktion auf Wunsch selbst.

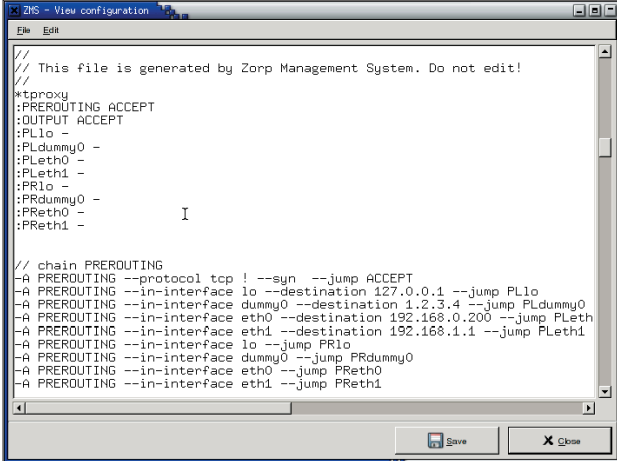
Zorp ist bei seinen Hardware-Ansprüchen recht bescheiden: Ein Rechner der Pentium-Klasse mit 233 MHz und 64 MByte RAM genügt für den Einstieg, auf der Festplatte belegt das System 1 GByte. Welche Anforderungen Zorp in größeren Umgebungen stellt, hängt vor allem vom Netzwerkverkehr und den genutzten Zorp-Komponenten ab. Lokal vorgehaltene Logfiles und Mailspool zum Beispiel benötigen unter Umständen sehr viel Plattenplatz.

Wenn etwa die Monitoring-Fähigkeit (siehe **Abbildung 3**) inklusive PostgreSQL-Datenbank zum Einsatz kommt, ist eine Aufteilung von statischen und dy-

namischen Daten auf mehrere Platten sehr zu empfehlen.

Besonders die Größe des Hauptspeichers beeinflusst die Anzahl der Verbindungen, die das System bewältigt. Bei 64 MByte vertragen das Betriebssystem und Zorp ungefähr 20 gleichzeitige Verbindungen, für jede weitere sind 200 KByte einzuplanen, bei 500 Verbindungen errechnen sich rund 164 MByte. Da Webbrowser gleichzeitig mehrere Verbindungen öffnen, reicht diese Ausstattung für 100 bis 200 Clients, die hauptsächlich HTTP-Verkehr verursachen.

Da eine Firewall meist am zentralen Zugang eines Unternehmens zum Internet steht, sollte die Hardware dem Dauerbetrieb auch gewachsen sein. Wer Wert auf maximale Verfügbarkeit legt, kann seine Zorp-Lizenz um das High-Availability-Paket (HA) erweitern. Ein zweiter Rechner steht dann als Hot Standby für den Ernstfall bereit. Diese Fähigkeit hebt Zorp von vielen anderen Open-Source-Produkten ab.



```

// This file is generated by Zorp Management System. Do not edit!

*protoxy
:PREROUTING ACCEPT
:OUTPUT ACCEPT
:PLto -
:PLdummy0 -
:PLeth0 -
:PLeth1 -
:PRto -
:PRdummy0 -
:PReth0 -
:PReth1 -

// chain PREROUTING
-A PREROUTING --protocol tcp ! --syn --jump ACCEPT
-A PREROUTING --in-interface lo --destination 127.0.0.1 --jump PLto
-A PREROUTING --in-interface dummy0 --destination 1.2.3.4 --jump PLdummy0
-A PREROUTING --in-interface eth0 --destination 192.168.0.200 --jump PLeth
-A PREROUTING --in-interface eth1 --destination 192.168.1.1 --jump PLeth1
-A PREROUTING --in-interface lo --jump PRto
-A PREROUTING --in-interface dummy0 --jump PRdummy0
-A PREROUTING --in-interface eth0 --jump PReth0
-A PREROUTING --in-interface eth1 --jump PReth1

```

Abbildung 2: Das ZMC-GUI erzeugt aus allen vom Admin vorgenommenen Einstellungen eigene Klartext-Konfigurationsdateien. Ein Blick in diese Files gibt zusätzlich Gewissheit, dass alle Optionen auch korrekt sind.

Im Gegensatz zum textbasierten Debian-Installer läuft die Zorp-Installation weitgehend grafisch ab, nur die Hardware-Erkennung arbeitet über ein Textinterface. Bis auf sehr exotische Gerätschaften erkennt Zorp alle Komponenten automatisch. Hat die Software dennoch gepatzt, kann der Admin in einem späteren Schritt GUI-gestützt zusätzliche Kernelmodule aktivieren.

Installation

Das grafische Installer-Fenster ist recht geschickt aufgebaut. Auf dem Hauptbildschirm läuft die eigentliche Installation ab. Zwei zusätzliche Registerkarten

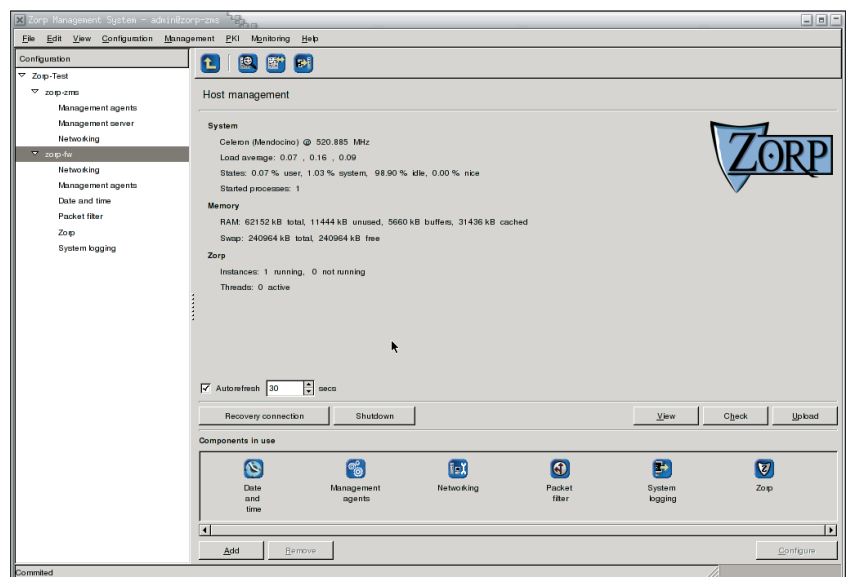


Abbildung 3: Über das ZMS-GUI lassen sich sämtliche Aspekte des Systems verwalten. Es dient auch als Monitoring-Applikation, die den aktuellen Zustand der Firewall darstellt.

```

Mounting: /dev/cdroms/cdrom0, //cdrom, iso9660
Partitioning in full size...
MAXSIZE: 4096 0
Writing out partition table started: 282.
-> [PART]: KEY: /dev/scsi/host0/bus0/target0/lun0
[PART]: START...
[PART]: ===== Writing out partition table!!! =====
[PART]: 0 130 83
[PART]: 131 47 82
[PART]: 179 343 83
[PART]:
[PART]: SFDISK-PARAM: 0 130 83
[PART]: 131 47 82
[PART]: 179 343 83
[PART]:
[PART]: -----
-> [PART]:
[PART]: Disk /dev/scsi/host0/bus0/target0/lun0/disc: 522 cylinders, 255 heads, 63 sectors/track
[PART]: /dev/scsi/host0/bus0/target0/lun0/disc: unrecognized partition
[PART]: Old situation:
[PART]: No partitions found
[PART]: New situation:
[PART]: Units = cylinders of 8225280 bytes, blocks of 1024 bytes, counting from 0
[PART]:
[PART]: Device Boot Start End #cyls #blocks Id System
[PART]: /dev/scsi/host0/bus0/target0/lun0/part1 0+ 129 130- 1044224+ 83 Linux
[PART]: /dev/scsi/host0/bus0/target0/lun0/part2 131 177 47 377527+ 82 Linux s
wap
[PART]: /dev/scsi/host0/bus0/target0/lun0/part3 179 521 343 2755147+ 83 Linux
[PART]: /dev/scsi/host0/bus0/target0/lun0/part4 0 - 0 0 0 Empty
[PART]: Successfully wrote the new partition table
[PART]:
[PART]: Re-read the partition table
  
```

Abbildung 4: Während der Installation gewährt das GUI dem Admin jederzeit Einblick in die Log-Ausgaben. Der »Messages«-Reiter zeigt hier, wie Zorp das Firewall-System partitioniert.

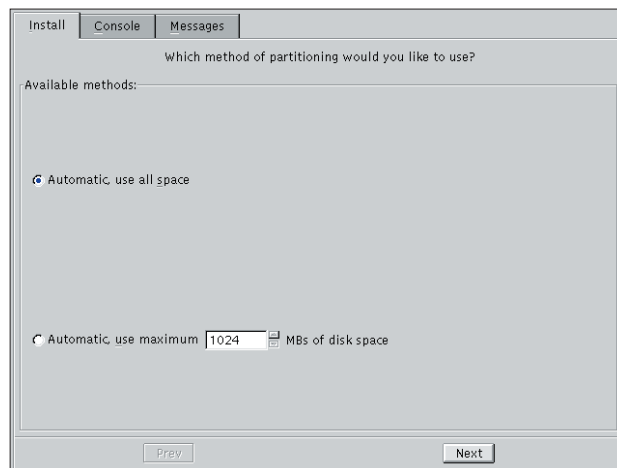


Abbildung 5: Der grafische Installer gibt dem Admin kaum Einfluss auf die Partitionierung der Festplatte. Das ist aber meist unproblematisch, da der Firewall-Rechner exklusiv für Zorp genutzt wird.

gewähren Zugang zur Konsole und zeigen die Systemlogs (siehe [Abbildung 4](#)). Ungeschickt ist dagegen die englische Tastaturbelegung, die sich nicht mal manuell über die Konsole ändern lässt. Dass Zorp aus Ungarn stammt, verraten spätestens die Lizenzbestimmungen, die in Landessprache gezeigt werden. Warum Balabit hier nicht wenigstens englischen Text zusätzlich vorgesehen hat, ist unverständlich.

Bei der Festplattenaufteilung erlaubt die getestete Version 3 im Gegensatz zum Vorgänger nur noch eine starre Partitionierung ([Abbildung 5](#)); der Admin darf lediglich den Platz beschränken, den Zorp für sich reserviert. Der Installer legt

ein Root-Filesystem, Swap-Space und eine eigene Partition für »/var« an. Beim Filesystem setzt Balabit auf das erprobte und robuste Ext 3. Erfahrene Admins haben aber meist eigene Vorstellungen von einer idealen Partitionierung, die sie hier nicht verwirklichen können.

Basissystem

Es folgt die Installation des Basissystems. Hauptunterschiede zum Standard-Woody sind Postfix statt Exim als Standard-MTA sowie Syslog-NG als Systemlogger. Nach der Debian-typischen Anfrage von Zeitzone und Passwort-Variante sind das Root-Passwort zu setzen und optional ein gewöhnlicher User anzulegen. Wer sich später per SSH auf der Maschine einloggen will, braucht diesen zusätzlichen Account: Der SSH-Daemon verweigert Root aus Sicherheitsgründen den direkten Login.

Bei der Konfiguration der Netzwerkkarten erwiesen sich zwei Karten gleichen Typs als problematisch. Der Installer erkannte im Test nur eine Karte und konfigurierte sie als »eth0«. Die zweite Schnittstelle ließ sich nur manuell nach der Installation in Betrieb nehmen.

Nach der Eingabe von IP-Adresse, Host- und Domainnamen sowie des zuständigen Nameservers sind die gewünschten Zorp-Komponenten zu installieren. Pflicht ist das Zorp Operating System, also die Distribution. Wer den Rechner in einer verteilten Firewall-Umgebung einsetzt, die er von einem zentralen Ma-

agement-Server administriert, braucht noch einen Management Agent auf der Maschine.

Die künftige Aufgabe des Systems ist als Firewall, Management- oder Authentication-Server oder als Monitor festzulegen. Ohne Management-Server lässt sich Zorp nur manuell konfigurieren. Wer mit der Lizenz auch den kommerziellen Virens Scanner Virusbuster [6] erworben hat, befördert diesen gleich mit auf die Festplatte.

Lizenzdatei erforderlich

Im nächsten Schritt verlangt Zorp die Lizenzdatei. Sie befindet sich normalerweise auf einer Diskette, der Installer kann sie aber auch per HTTP von einem Webserver holen. Wer aus Sicherheitsgründen kein Diskettenlaufwerk besitzt, muss sich auf das potenziell gefährliche Netz verlassen. Balabit sollte weitere Varianten vorsehen, um das File in den Rechner zu befördern. Der optionale Monitoringserver legt die anfallenden Daten in einer PostgreSQL-Datenbank ab – sie kann lokal oder auf einem eigenen Server liegen.

Die nächsten Fragen beziehen sich auf die Postgres-Konfiguration. Vorsicht ist bei der ersten Frage angebracht: Bei einem »Yes« als Antwort verschwindet die gesamte Datenbank im Falle einer Deinstallation des Pakets per »purge« im Daten-Nirwana. Ein Nein wäre daher der sinnvollere Standardwert, den der Installer auch ohne Rückfrage setzen

Zorp GPL

Neben der kommerziellen Variante gibt es Zorp auch als GPL-Software [2]. Der Support beschränkt sich hier auf Mailinglisten. Außerdem muss der Admin die Firewall manuell konfigurieren, das Konfigurations-GUI steht nicht unter der GPL. Leider sind in Zorp GPL weit weniger Proxys enthalten als in der kommerziellen Version. FTP, HTTP, SSL, Telnet, Finger und Whois sollten für den Privatgebrauch aber ausreichen. Wer andere Protokolle nutzen will, kann den generischen »plug«-Proxy einsetzen, der allerdings das jeweilige Protokoll durchreicht und nur grundlegend analysiert.

Zorp GPL funktioniert mit praktisch jeder Distribution. Es setzt folgende Software voraus: Glibc ab Version 2.0, Python ab Version 2.1, Libcap ab 1.10 und OpenSSL ab 0.9.6g. Wer Kernel 2.4 nutzt, benötigt zusätzlich Balabits TProxy-Patch.

sollte. Nach einigen Fragen zur IPsec- und zur SSH-Konfiguration ist die Rolle des Systems in der Zorp-Umgebung festzulegen: »Firewall«, »ZMS-Host« oder »None«. Von der Wahl hängt ab, wer sich auf welche Weise mit dem Rechner verbinden darf. »Firewall« etwa erlaubt nur Verbindungen vom Management-Server. Eine entsprechende Firewall-Policy wird automatisch erstellt.

Wer Firewall- und Management-Server auf demselben Rechner betreibt, wählt »Firewall« als Verwendungszweck und gibt als ZMS-Adresse »127.0.0.1« an. Für den GUI-Zugriff auf den ZMS ist später noch der Client-Rechner in der Firewall-konfiguration freizuschalten.

Die Konfiguration des ZMS beginnt mit dem Firmennamen, der ihn später eindeutig identifiziert. Auch der Hostname lässt sich passend setzen. Als wichtigster Schritt folgt das Erzeugen des CA-Schlüssels für den Management-Server. Mit ihm beglaubigt der ZMS später die Schlüssel für die anderen beteiligten Rechner. Für den Admin-User des ZMS und für die CA sind getrennte Passwörter zu setzen, aus Sicherheitsgründen ist das auch empfehlenswert. Abschließend fragt der Installer noch die Schlüssellänge und Gültigkeitsdauer der Zertifikate ab und erzeugt dann die benötigten Schlüssel.

Zorp-Spezial

Bei der ersten Konfiguration eines Zorp-Systems erweist sich die Zorp-eigene Terminologie als kleine Barriere. Was der Begriff Zone meint, lässt sich noch gut erschließen, bei einem Chainer hilft aber nur noch der Blick in das (sehr gute) Handbuch.

Zorp reguliert den Datenstrom mit Hilfe von Zonen. Zonen sind im Vorfeld definierte Subnetze, mit denen das Netzwerk die Richtung des Netzwerkver-

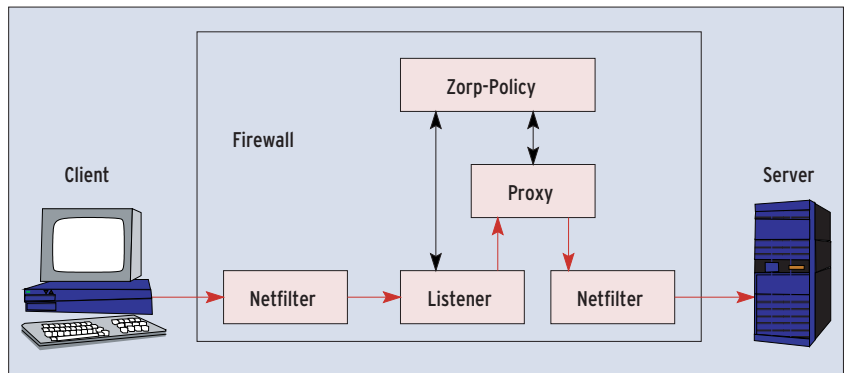


Abbildung 6: Die Zorp-Firewall filtert eingehende Pakete zunächst per Netfilter-Regel. Listener verteilen den Datenstrom an die zuständigen Proxys. Die vom Proxy gesendeten Daten müssen weitere Netfilter-Regeln passieren, bevor sie die Firewall verlassen.

kehrs festlegt. Die Internet-Zone ist bereits vorkonfiguriert, ihr Symbol dient für sämtliches Unbekannte. Um ein Subnetz genauer kontrollieren zu können, bietet es sich an, die zugehörige Zone weiter zu unterteilen. Untergeordnete Zonen erben die Eigenschaften der nächsthöheren.

Sehr hilfreiche Werkzeuge sind die Zorp-Instanzen. Mit ihnen ist es möglich, mehrere virtuelle Firewalls auf einer Maschine zu betreiben, die eigenständige Aufgaben wahrnehmen. Der Admin kann Regeln somit voneinander trennen, sie beeinflussen sich auch bei größeren Änderungen nicht gegenseitig. Zum Beispiel funktioniert die DMZ weiter, auch wenn das interne Netz nach einem Konfigurationsfehler keinen Zugriff mehr auf das Internet erhält.

Dienste integriert

Die Zorp-Services legen fest, welcher Proxydienst welchen Netzwerkverkehr bearbeitet und wie die Firewall Source- und/oder Destination-NAT für diese Verbindung handhaben soll. Innerhalb der Services legen Router und Chainer fest, wie der Proxydienst mit dem Datenstrom verfährt. Er kann den Verkehr

vollkommen transparent durchschleifen oder ihn auf eine bestimmte IP und/oder einen bestimmten Port leiten.

Als weitere Besonderheit gibt es Listener (Abbildung 6). Ein Listener horcht auf dem ihm zugewiesenen Port und gibt hereinkommende Verbindungen an seine Services weiter. Während sich die Listener ausschließlich um TCP kümmern, sind Receiver für UDP-Verbindungen zuständig.

Application Level Gateway und Paketfilter

Trotz Application Level Proxy bildet ein Paketfilter die Basis der Zorp-Firewall. Als Grundschutz sorgt er dafür, den Zugriff von bestimmten Rechnern auf bestimmte Dienste zu erlauben oder zu verbieten. Zusammen mit dem TProxy-Patch [3] leitet der Paketfilter zulässige Verbindungen durch die Proxys.

Nach der Installation setzt Zorp eine Standard-Policy um. Sie besteht aus vier Regeln, die nur den bereits konfigurierten Clients Zugriff auf die benötigten Ports der Firewall gestatten. Ist der Rechner als Zorp-Firewall installiert, darf nur der ZMS-Host per SSH und per ZMS-Dienst zugreifen. Der ZMS läuft dabei eventuell auf dem lokalen Rechner. Jeden Zugriff auf andere Ports protokolliert und verwirft der Paketfilter (Drop). Ein kurzer Portscan per Nmap bringt das Ergebnis in Listing 1 zustande: Der Scanner erkennt den Rechner und stellt fest, dass er läuft, aber alle Ports filtert. Das Betriebssystem errät Nmap nicht.

Durch das Dropen der Pakete bemerken Außenstehende also recht schnell,

Listing 1: Nmap-Scan

```
01 Starting nmap 3.70 ( http://www.insecure.org/nmap/ ) at 2004-09-28 16:33 CEST
02 Initiating SYN Stealth Scan against 192.168.1.1 [1660 ports] at 16:33
03 The SYN Stealth Scan took 333.64s to scan 1660 total ports.
04 Warning: OS detection will be MUCH less reliable because we did not find at least 1 open
and 1 closed TCP port
05 Host 192.168.1.1 appears to be up ... good.
06 All 1660 scanned ports on 192.168.1.1 are: filtered
07 Too many fingerprints match this host to give specific OS details
```

dass auf dem Rechner ein restriktiver Paketfilter arbeitet. RFC-konformes Rejection wäre unauffälliger. Gefährlich ist die fehlende Limitierung der erzeugten Logs: Ohne weitere Maßnahmen könnte ein Angreifer die Firewall durch einen Denial of Service lahmlegen. Allerdings stehen zu diesem Zeitpunkt ohnehin noch viele Anpassungen ins Haus, bei denen der Admin auch diese Defaults durch sinnvollere Einstellungen ersetzen sollte. Mit dem ZMS-GUI sind diese Schnitzer auch schnell behoben.

Nach dem ersten Einloggen am ZMS präsentiert sich der GUI-Client spartanisch. Einzig der ZMS-Host bietet bereits diverse Optionen, sie sind mit guten Standardwerten belegt. Bevor er die Konfiguration einer oder mehrerer Firewalls beginnt, muss der Admin die Komponenten in das Managementsystem einbinden. Das geht entweder manuell oder – schneller und bequemer – per Bootstrapping (Abbildung 7). Diese Funktion führt den Admin in typischer Wizard-Technik durch die einzelnen Schritte und benutzt hilfreiche Templates für die wichtigsten Optionen.

Bootstrapping

Alle Bestandteile der Zertifikate bis hin zur Schlüssellänge legt der Admin beim Bootstrapping fest. Zur Beglaubigung des Schlüssels braucht er noch das CA-Passwort, das er bei der Installation festgelegt hat. Da der Rechner bei diesem Stand der Konfiguration noch kein Zertifikat hat, bedient sich Zorp eines kleinen Tricks: Zuerst basiert die Kodierung nur auf einem bei der Installation des Management Agent angegebenen One-Time-Passwort. Es soll sicherstellen, dass kein Angreifer das Zertifikat während der Übertragung manipulieren kann.

Als sehr hilfreich erweist sich, dass

der Wizard im Fehlerfall direkt an jenen Punkt zurückspringt, der wahrscheinlich zum Fehler geführt hat.

Instanzen, Zonen und Dienste

Um das Regelwerk der Firewall möglichst übersichtlich zu halten, sollte der Admin den Netzaufbau in Zonen abbilden. Das einfachste Beispiel kommt mit zwei Zonen aus: dem Internet sowie dem internen, geschützten Netz. Als dritte Zone ist meist eine DMZ einzurichten, es sind aber auch deutlich komplexere Netzstrukturen möglich. Das Zonenmodell sollte sorgfältig geplant sein, da es die weitere Konfiguration entscheidend prägt.

Die gewünschte Firewall-Policy ist bei planvollem Vorgehen bereits im Vorfeld ausgearbeitet und dokumentiert. Von ihr hängt ab, wie stark die vorgefertigte Paketfilterkonfiguration anzupassen ist. Für die Grundkonfiguration genügt eine sehr einfache Policy. Die Feinheiten der TProxy-Umleitungen auf die definierten Services erzeugt das Management-Tool automatisch, sobald der Admin einen Dienst konfiguriert.

Selbst um NAT-Regeln braucht er sich nicht zu kümmern. In vielen Fällen ist NAT überflüssig, da die Firewall selbst die Verbindung zum Zielrechner aufbaut. Diese Verbindung verwendet dann – sofern es nicht explizit anders konfiguriert ist – eine IP-Adresse der Firewall als

Abbildung 7: Der Bootstrapping-Wizard fragt vom Admin schrittweise alle Konfigurationsparameter ab. Dazu gehören auch Informationen über die Zertifikate, die Zorp für die SSL-geschützte Kommunikation benutzt.

Absender. Wer dennoch spezielle Anforderungen an NAT stellt, setzt dafür einen weiteren Service auf oder erledigt die Aufgabe über den Paketfilter.

Der schwierigste Teil der Zorp-Konfiguration betrifft den Einsatz der Zorp-Instanzen und -Services. Während eine einfache Firewall-Konfiguration mit einer Instanz und wenigen grundlegenden Services auskommt, entsteht bei komplexen Netzen mit vielen Sonderfällen schnell ein unübersichtliches Gebilde. Durch klug gewählte Instanzen gelingt es aber meist, den Überblick zu behalten und Nutzergruppen oder Serverfarmen sauber zu trennen. Wer diese Möglichkeiten aber zu intensiv nutzt, überlastet leicht den Rechner.

Wahl des besten Proxy

Die Auswahl der passenden Proxykomponente trägt entscheidend zur Sicherheit des Systems bei. Für den scheinbar einfachen HTTP-Zugriff auf einen Webserver stehen diverse Optionen zur Wahl: Den Verkehr transparent über einen Proxy leiten oder den Proxy im Client fest eintragen; die Anfrage unverändert durchreichen oder bestimmte Teile (etwa die User-Agent-Angabe) umschreiben; in SSL getunneltes HTTP ungeprüft durchreichen oder durch Protokolltricks in den HTTP-Verkehr eingreifen.

Der letztgenannte Punkt ist besonders interessant. Sinn und Zweck der SSL-Verschlüsselung ist unter anderem, solche Man-in-the-Middle-Manipulationen zu verhindern. Wer seinen Nutzern und deren Browser-Software so weit misstraut, dass er auch HTTPS-Verkehr durch die Firewall kontrollieren lassen möchte, kombiniert dazu zwei Proxys. Einer knackt den SSL-Schutz, der zweite prüft wie gewohnt den HTTP-Verkehr

SSL ausgetrickst

Um in den SSL-verschlüsselten Verkehr einzugreifen, behauptet die Firewall, dass der Server keine SSL-Verbindungen zulässt. Zum Server baut Zorp stellvertretend für den Client eine verschlüsselte Verbindung auf, deren Netzwerkverkehr dann die Firewall analysiert. Wer diesen Trick einsetzt, muss seinem internen Netz vertrauen, da er dort unverschlüs-

selten Verkehr mit meist sicherheitsrelevantem Inhalt erlaubt. In vielen Umgebungen ist es besser, SSL-Verbindungen ungeprüft durch die Firewall zu schleusen. Mit dem einfachen »plug«-Proxy ist das auch schnell erledigt.

Viel sicherer arbeitet der SSL-Proxy, wenn er Zorps CA nutzt, um Kopien der Server-Zertifikate für sich selbst auszustellen und zu beglaubigen. Diese Zertifikate gelten für den Schlüssel der Firewall, enthalten sonst aber die Angaben aus den Originalzertifikaten. Damit nimmt der SSL-Proxy zum Client eine verschlüsselte und authentifizierte Verbindung auf, in der er sich als Server ausgibt. Der Client bemerkt jedoch diese Manipulation und gibt eine Sicherheitswarnung aus, da er die Zorp-CA nicht kennt und ihr nicht vertraut. Abhilfe schafft der Import von Zorps CA-Schlüssel in die Browser. Leider ist diese Technik zurzeit in der Dokumentation nicht zu finden, da das GUI sie erst in der aktuellen Testversion vorsieht.

Zum Schluss steht die Konfiguration des Listeners an. Erst er versorgt den Proxy, der einen Service erfüllt, mit Daten. Listener sind leicht zu definieren, es stehen kaum Optionen zu Wahl. Die schwierigste Entscheidung betrifft den Listener-Typ. Die Varianten unterscheiden sich darin, welche Zonen sie bedienen.

Der einfachste Typ bearbeitet alle Anfragen gleichwertig, egal aus welcher Zone sie stammen. Mit Hilfe eines »Zone Listener« lässt sich die Herkunft der Anfrage eingrenzen. Der »CSZone Listener« erlaubt es dem Admin zusätzlich, das Ziel des Datenstroms zu spezifizieren.

So kann er Zugriffe sehr gezielt einschränken. Er muss darüber hinaus nur noch den Service nennen, den der Listener bedienen soll, Adresse und den Port konfigurieren, auf dem der Listener Verbindungen abgefangen soll, und auswählen, ob dieser Vorgang transparent ablaufen soll.

Die Änderungen gelangen per Commit und Upload vom GUI über den ZMS auf die Firewalls. Um auch die Regeln des Paketfilters zu aktivieren, ist noch bei dessen Konfiguration ein so genanntes Skeleton zu erzeugen. Es enthält die benötigten Regeln. Ist auch dieser Teil der Konfiguration auf die Firewall übertragen, werden die Änderungen durch einen Restart des Paketfilters und der Zorp-Komponente aktiv.

Sollen zusätzlich zu den Zorp-Proxys native Dienste wie Bind oder der »ntpd« auf der Firewall laufen, muss der Admin diese auch in der Paketfilter-Konfiguration eintragen. Sie nehmen auf ihrem Port Anfragen entgegen und beantworten sie – ganz ohne Umleitung oder Proxy-Dienst.

Virens Scanner eingebaut

Eine Konfiguration für das interne Netz aus **Abbildung 1** ist schnell erstellt. Sie soll HTTP-Verkehr transparent durchreichen und durch Virusbuster [6] prüfen lassen; eine Instanz genügt für dieses Beispiel. Zwei Services kümmern sich um den HTTP-Verkehr. Der erste verwendet einen transparenten HTTP-Proxy, um den korrekten Protokollablauf zu prüfen. Der zweite Service nutzt den

Lizenzierung und Preise

Balabit bietet zwei kommerzielle Zorp-Lizenzmodelle an: Die Professional-Variante enthält die Lizenz für eine Firewall, die von einem ZMS verwaltet wird. Die Anzahl der GUI-Clients ist unbegrenzt. Die Enterprise-Lizenz enthält zusätzlich den ZAS, über den sich eine unbegrenzte Anzahl Clients authentifizieren darf. Nur dieses Lizenzmodell gestattet es, die Firewall-Umgebung durch weitere – zusätzlich lizenzierte – Firewalls zu vergrößern. Beide Modelle enthalten die VPN-Funktionalität. Eine Clusterlizenz, die zwei Firewalls inklusive ZMS und HA-Modul umfasst, ist für beide Modelle verfügbar. Der Preis berechnet sich aus der Anzahl der geschützten Rechner. Die Staffelung beginnt

bei 1 bis 25 Rechnern, ab 50 zu schützenden Hosts geht die Berechnung in 50er Schritten weiter. Kosten von 9780 Euro für eine unbegrenzte Anzahl von Clients in der Professional-Variante liegen im Rahmen der marktüblichen Preise.

Optional erhältlich sind die Updates für einen Zeitraum von drei Jahren. Auch über den 30-Tage-Installationssupport hinausgehenden Support berechnet Balabit gesondert. Der Preis für beide Optionen richtet sich wieder nach der Anzahl der zu schützenden Rechner. Beim Support wirken sich zudem Umfang und Servicelevel auf die Kosten aus. Eine Preisliste ist online unter [<http://www.zorpfirewall.com/howtobuy/pricing/1>] als PDF verfügbar.

VBuster-Proxy. Dieser Dienst wird in den HTTP-Service eingebunden.

Die Konfiguration des Listeners beschränkt sich darauf, die IP-Adresse der Netzwerkschnittstelle festzulegen. Da die Verbindungen aus dem LAN kommen, ist die interne Adresse der Firewall anzugeben. Der für HTTP genutzte Port ist bereits im GUI eingetragen. Auch der schon gesetzte Haken für das transparente Handling der Verbindungen passt zur Beispiel-Policy.

Da die Anfragen der Clients aus dem LAN nicht auf einen speziellen Rechner umgeleitet werden müssen, bleiben die beiden Router-Optionen »Transparent« und »Inband« übrig. »Inband« bezieht alle Informationen über Quelle und Ziel der Verbindung aus dem Protokoll selbst. Solche Adresseninformationen finden sich nur in relativ wenigen Protokollen, beispielsweise FTP und HTTP.

Gefälschte Adresse

Die Option »Forge Address« wäre sinnvoll, wenn externe Clients auf einen Webserver in der DMZ zugreifen. Für die Verbindung zum Server verwendet der Proxy dann die Adresse des Clients als Absender. Damit enthalten die Apache-Logfiles die Adressen der echten Clients und nicht die der Firewall. Das interne Netz im Beispielszenario verwendet private IP-Adressen, daher wäre diese Option hier falsch.

Den Virenschanner trägt der Admin über die Schaltfläche zur Konfiguration des Routers innerhalb des Service ein, und zwar als »Chained Proxy«. Damit defi-

niert Zorp automatisch einen Chainer, der den Datenstrom bei der Analyse an den Virenschanner weitergibt.

Nagelprobe

In der Einarbeitungszeit wirkt die grafische Oberfläche teils verwirrend: Wichtige Optionen verstecken sich in einem erweiterten Optionsfeld – bei ihrem Funktionsumfang kaum zu vermeiden. Wer will, kann auf die Hilfe von ZMS und ZMC auch verzichten und alle Einstellungen manuell vornehmen.

Die Installation verläuft durch das falsche Tastaturlayout riskanter als nötig. Auch das später installierte System benutzt dieses Layout, hier lässt es sich aber umstellen: »dpkg-reconfigure console-data«. Stärker stört die automatische Partitionierung der Festplatte; diese feste Vorgabe passt nicht zur sonst üblichen Flexibilität des Produkts.

Positiv fällt hingegen die Dokumentation auf. Umfangreiche Handbücher zur Installation und Konfiguration erleichtern den Einstieg erheblich, sie enthalten praxisnahe Beispiele und gute Erklärungen. Konnten sie im Test einmal nicht weiterhelfen, war der Support schnell und kompetent zur Stelle.

Balabit hat mit Zorp ein Produkt geschaffen, das sich aufgrund seiner Architektur hervorragend für größere Organisationen mit hohem Schutzbedürfnis eignet. Es ist innovativ und technisch ausgereift, dabei bleibt Zorp jederzeit sehr flexibel. Der Lernaufwand, um diese ALG-Firewall zu beherrschen, ist allerdings höher als bei reinen Paketfil-

tern oder bei den Konkurrenzprodukten. Der Gewinn an Sicherheit rechtfertigt den Aufwand jedoch. (fjl) ■

Infos

- [1] Zorp: [<http://www.zorpfirewall.com>]
- [2] Zorp GPL: [http://www.balabit.com/products/zorp_gpl/]
- [3] TProxy-Patch: [<http://www.balabit.com/products/oss/tproxy/>]
- [4] Netfilter: [<http://www.netfilter.org>]
- [5] IPChains: [<http://people.netfilter.org/~rusty/ipchains/>]
- [6] Virusbuster: [<http://www.virusbuster.hu>]
- [7] TIS Firewall-Toolkit: [<http://www.fwtk.org/main.html>]
- [8] Checkpoint AI: [http://www.checkpoint.com/products/downloads/applicationintelligence_whitepaper.pdf]
- [9] Juniper Deep Inspection: [http://www.juniper.net/products/integrated/firewall_technology.html]
- [10] Cisco Context-Based Access Control: [<http://www.ciscopress.com/articles/article.asp?p=26533>]
- [11] Syslog-NG: [http://www.balabit.com/products/syslog_ng/] und Christian Schmitz, „Systemprotokolle der nächsten Generation“: Linux-Magazin 11/03, S. 61
- [12] Openwall: [<http://www.openwall.com>]
- [13] Openswan: [<http://www.openswan.org>]

Der Autor



Christian Ney arbeitet als Unix- und Firewall-Administrator bei einer Regionalfluggesellschaft und wirkt in seiner Freizeit in mehreren Open-Source-Projekten mit.

1/4 quer A

210x81 mm zzgl. Beschnitt

IMS