Firewall selbst gebaut

Türen schließen

Michael Plura

Zum Erweitern der Firewall-Kenntnisse, als heimischer DSL-Router mit Firewall auf einem alten oder Mini-PC sowie beim Aufbauen virtueller Netze leistet eine Minivariante des bei OpenBSD vorinstallierten Paketfilters *pf* ausgezeichnete Dienste.

VirtualBox bietet ein rein virtuelles „internes“ Netzwerk. Eine virtuelle Firewall kann es mit dem LAN des Host verbinden und für Experimente dienen – etwa mit OpenBSD und *pf*.

Eine OpenBSD-VM ist mit 128 MByte RAM und 8 GByte Festplattenplatz ausreichend bestückt. In den Optionen der Maschine ist es sinnvoll, unter „System“ die Option „Hardware-Uhr in UTC“ zu aktivieren, Audio darf abgeschaltet sein. Als Nächstes geht man daran, zwei virtuelle Netzwerkadapter (vNICs) zu definieren: Der erste soll die Verbindung zum neuen virtuellen LAN herstellen, angelegt als „Internes Netzwerk“ („intnet“, den Namen darf man ändern).

Der zweite vNIC, gekoppelt ans Host-LAN, stellt die „Internet-Seite“ dar und muss daher als „Netzwerkbrücke“ zu *eth0* des Hosts dienen. Es empfiehlt sich, beide Adapter als „Intel PRO/1000 MT Server“ im Promiscuous-Modus („erlauben für alle VMs und den Host“) zu definieren. Weitere VMs in dem neuen virtuellen LAN erhalten jeweils einen vNIC, der nur mit dem internen Netz „intnet“ verbunden wird. Sie werden später die virtuelle Firewall als Gateway ins Host-LAN und damit ins Internet nutzen.

Besonderheiten von OpenBSD

Dank seiner Ausrichtung auf Sicherheit und Einfachheit zusammen mit einer streng auditierten Code-Basis liefert OpenBSD die ideale Grundlage für eine Firewall. Die aktuelle Version 5.7 ist auf openbsd.org verfügbar; als Basis genügt das nur 7 MByte kleine „cd57.iso“. Per „i“ startet die Installation, „de“ wählt ein deutsches Tastaturlayout, der Hostname kann „fw01“ heißen.

Die Netzadapter scheint VirtualBox nach eigenem Gusto zu sortieren, denn *em0* ist nicht zwangsläufig der erste definierte vNIC. Daher muss die Konfiguration flexibel bleiben: Ist ein DHCP-Server im LAN vorhanden, sollte man *em0* zuerst per *dhcp* konfigurieren. Funktioniert das, ist *em0* das externe Netzwerk der Firewall. Führt das zu einer Fehlermeldung, ist *em0* mit „intnet“ verbunden und erhält von Hand die Adresse 192.168.1.254. Bei „Which network interface… [done]“ richtet man durch das Eingeben von *em1* den zweiten vNIC ein – als *dhcp*, wenn *em0* fix ist, und mit fester IP-Adresse, wenn *em0* per *dhcp* konfiguriert wurde –, „done“ beendet den Vorgang. Nach Vergabe des *root*-Passworts aktiviert man *sshd* samt *ntpd* und legt das X Window System mit „no“ still. Bei der Partitionierung reicht eine Bestätigung, bei den zu installierenden Sets werfen *–x** und *–g** unnötigen Ballast über Bord. Einige Augenblicke später ist OpenBSD 5.7 installiert – einen VirtualBox-Snapshot und das Aushängen des ISOs sollte man nicht vergessen.

Wurde kein Benutzer außer *root* angelegt, kann sich *root* direkt via *ssh* einloggen. Linux-Administratoren finden mit *pkg_add nano* einen vertrauten Editor, weitere Software verwalten *pkg_remove* und *pkg_info*; *pkg_add –u* aktualisiert die Pakete. Die Firewall *pf* gehört zum Basissystem und ist daher bereits aktiv.

Software verwalten ohne Umstände

Damit OpenBSD Pakete nach einem Neustart durchleitet, muss *net.inet.ip.forwarding =1* in */etc/sysctl.conf* stehen; *sysctl net.inet.ip.forwarding=1* aktiviert das Forwarding im laufenden System. Ein *ifconfig* listet alle erkannten Interfaces auf, unter OpenBSD bestehen die Bezeichnungen aus einem Herstellerkürzel plus einer laufenden Nummer. OpenBSDs *pf* lässt sich ganz simpel über */etc/pf.conf* konfigurieren, einen neuen Regelsatz generiert man per *pfctl –nf /etc/pf.conf*. Die Option *–n* führt nur zu einem syntaktischen Prüfen, erst *–f* veranlasst das Laden, *–e* aktiviert, *–d* deaktiviert *pf*. Den aktuellen Regelsatz löst *pfctl –sr* auf, *–si* zeigt Informationen, *–ss* die aktuellen Verbindungen. *sysctl pf* oder das nachzuinstallierende *pftop* geben weitere Zustandsmeldungen in Echtzeit aus.

Die einfachste Regel ist „pass“ in */etc/pf.conf*, sie lässt jeglichen Datenverkehr durch, „block“ schließt alles, sinnvolle Regelwerke liegen dazwischen. Für einen einfachen NAT-Router etwa:

```
extern_if = "em1"
intern_if = "em0"
localnet = $intern_if:network
match out on $extern_if inet from $localnet ⤸
                              nat-to ($extern_if)
block all
pass from { self, $localnet }
pass in proto tcp to $extern_if port ssh
```

In den ersten zwei Zeilen erscheinen das externe und interne Interface, *localnet* steht für das interne Netz. Die *match*-Zeile markiert Pakete für NAT und erkennt wegen der Klammern dynamische IP-Änderungen von *extern_if*. Geschweifte Klammern in der *pass*-Zeile interpretiert *pf* als Liste, die letzte Zeile lässt *ssh*-Zugriffe vom Host-LAN aus zu – im LAN okay, im WAN gefährlich.

Infos zu OpenBSD und *pf* stehen in den Links, empfehlenswert sind die Bücher „Absolute OpenBSD“ von Michael W. Lukas und „Book of PF, 3rd Edition“ von Peter N. M. Hansteen, ebenfalls unter „Alle Links“ zu finden. (rh)

Michael Plura

lebt in Schweden und ist freier Autor mit den Schwerpunkten IT-Sicherheit, Virtualisierung und freie Betriebssysteme.

Alle Links: www.ix.de/ix1509145