Einführung

Dieser Artikel wendet sich besonders an neue Benutzer von Ubuntu, die zu ihrer Sicherheit eine sogenannte *Personal Firewall* installieren wollen. Gerade Umsteiger von anderen Betriebssystemen haben oftmals auf die harte Tour lernen müssen, dass ein ungesicherter Rechner im Internet nicht lange unkompromittiert bleibt, oder sie haben in der Presse gelesen, dass der Einsatz von *Personal Firewalls* auf Desktop-Systemen überlebenswichtig sei. Selbst Linux-Fachzeitschriften springen hin und wieder auf diesen Zug auf und bescheinigen Ubuntu daher, seine User *"den Gefahren des Internets schutzlos ausgesetzt"* zu überlassen. *(Linux User 03.2006, S. 56)*

Tatsächlich sind solche Programme aber nur deswegen so populär, weil in gängigen Betriebssystemen durch fehlerhafte Designentscheidungen in der Vergangenheit oft große Sicherheitsprobleme entstanden sind, die durch *Personal Firewalls* notdürftig abgedeckt werden. Anstatt eine Anleitung für die Installation einer solchen "Firewall" auf einem Einzelbenutzersystem zu beschreiben, möchte ich daher erklären, warum Ubuntu standardmäßig keine solche "Firewall" installiert hat, und warum das auch gar nicht nötig ist.

Sicheres Design - keine offenen Ports

Die Installation eines Paketfilters a.k.a. "Firewall" unnötig zu machen, ist ein Designprinzip von Ubuntu. Schon bei der Auswahl der Software wird darauf geachtet, dass so wenig Serverdienste wie möglich installiert werden. Die übrigen werden konsequent so installiert, dass sie in der Grundeinstellung nur vom eigenen Rechner erreicht werden können. (Loopback-Schnittstelle) Das ist bei anderen Betriebssystemen oft anders, die anscheinend vorwiegend für das Einsatzszenario geschützter lokaler Netzwerke konstruiert werden.

Eine normale Ubuntu-Desktop-Installation öffnet deswegen **keinen einzigen Port nach außen** und ist deswegen unangreifbar für alle Angriffsszenarien, vor denen ein Paketfilter Schutz bieten könnte.

Sollte jemand doch mal Software in den Ubuntu-Repositories entdecken, die ungefragt Ports nach außen öffnet, so sollte das sofort dem Paketmaintainer als Bug (critical, security) gemeldet werden. Wenn die Ubuntu-Policy lautet, *"keine offenen Ports nach außen"*, dann müssen sich die Maintainer auch daran halten.

Das gilt natürlich nicht für echte Server. Wer sich Samba, ssh, apache, etc. auf dem Rechner installiert, der möchte im Allgemeinen den Zugriff von außen erlauben. Wer das nicht will, (und bspw. den Apache-Webserver als Testumgebung für Webdesign nutzen möchte,) der sollte die jeweiligen Konfigurationsmöglichkeiten nutzen, um die Server nur an die Loopback-Schnittstelle (127.0.0.1) zu binden. Das ist auch nicht schwieriger, als eine "Firewall" zu installieren.

Es stimmt übrigens nicht, dass Ubuntu keinen Paketfilter mitliefert. Das Linux-Firewall-System **netfilter/iptables** wird sehr wohl installiert, es werden bloß nicht von vorneherein restriktive Regeln aufgestellt. Ebenso gibt es zur Konfiguration einer Ubuntu-Firewall kein grafisches Werkzeug im **main**-Repository, wohl aber in **universe**, z.B. das Programm **firestarter**.

Mythen demystifiziert

Attacken gegen Client

Manche Leute glauben, eine "Firewall" würde sie beim Internetsurfen oder Email lesen schützen. Dazu Folgendes: Kein Client-Programm (Web-Browser, Email-Programm, etc.) kann von Hackern aus dem Netz heraus direkt angegriffen werden. Niemals. Diese Programme sind nur dazu

konzipiert, selbst Verbindungen aufzubauen, und diese zu nutzen. Eine solche TCP/IP-Verbindung zu *hijacken* und als Außenstehender zu nutzen, ist zwar theoretisch möglich, aber eindeutig nicht-trivial, und würde dann auch wohl von keiner Firewall entdeckt werden.

Angriffe gegen Client-Programme laufen deswegen immer über den Inhalt - das was der Benutzer (oder sein Programm) sich freiwillig runterlädt. Es gibt zwar Firewall-Systeme, die unter Umständen auch gegen sowas schützen. Aber das sind dann nicht die Paketfilter, die so gerne liebevoll "Firewall" genannt werden, sondern inhaltsbezogene Proxy- und ähnliche Systeme (z.B. squidguard, mailserver+virenscanner, intrusion detection/prevention systeme), deren Einrichtung für Unerfahrene eher nicht anzuraten ist. Wenn man da nicht genau weiß, was man tut, kann man leicht viel größere Lücken aufreißen, als man schließen möchte. Sowas lohnt sich nur, wenn man für die Administration von einer mittleren bis größeren Anzahl Büroarbeitsplätze verantwortlich ist, oder wenn man sich wirklich für IT-Sicherheit als Hobby entscheidet.

"Kaputte Pakete"

Auch die Behauptung, ein Paketfilter würde "kaputte Pakete" zurückweisen, und dadurch das System schützen, ist falsch. "Kaputte Pakete" werden auch ohne Filter vom Netzwerkstack verworfen, und erreichen die Anwendung (den Server) nicht. Theoretisch könnte es natürlich Fehler im Netzwerkstack geben, die durch solche Pakete getriggert werden. Das könnte aber auch im **netfilter-/iptables**-Modul passieren, beides gilt jedoch als relativ sicherer Code. Die Zeiten, wo schlechte IP-Implementierungen reihenweise zu Denial-of-Service-Angriffen einluden, sind jedenfalls schon seit längerem vorbei.

"Unsichtbarmachen"

Das System "unsichtbar" zu machen, indem man Pakete nicht regelgerecht abweisen, sondern kommentarlos verwerfen lässt (`DROP`), ist ebenfalls nicht sinnvoll. Ein System, das keine Ports offen hat, wie ein Standard-Ubuntu-Desktop, hat keinen einzigen Grund, "unsichtbar" zu sein. Im Gegenteil: Sendet ein System eine Verbindungsanfrage (bspw. weil es sich vor kurzem mit einem anderen System ausgetauscht hat, das zu dem Zeitpunkt diese IP-Adresse hatte,) weiß es sofort, dass es dort keinen Dienst gibt, und bricht ab (wenn es sauber programmiert ist). Bei "unsichtbaren" Systemen versucht es dagegen noch eine halbe Ewigkeit, die Daten zuzustellen. Wenn ein Rechner aber sowieso einen Dienst offen hat, bspw. ssh, dann kann keine "Firewall" der Welt diesen Port "unsichtbar" machen.

"Unsichtbarmachen" kann sogar Probleme verursachen. Bspw. versuchen einige FTP- und IRC-Server, bei deinem Login eine ident-Abfrage zu machen (Port 113). Wenn dieser Port nun "unsichtbar" ist, wird dein Login verzögert, bis der Timeout kommt. Andere Probleme kann man bekommen, wenn irgendeine Software wahllos ICMP-Pakete verwirft, wie es viele Windows-Firewalls machen. Dann funktioniert nämlich unter Umständen die sogenannte Path-MTU-Discovery nicht mehr, und es kann zu "rätselhaften" Verbindungsproblemen kommen, die sich der Laie nicht erklären kann.

Personal Firewalls "vereinfachen" die Administration

Manche Leute denken, die Installation eines Paketfilters mit Hilfe eines grafischen Tools wäre einfach, würde keinen Schaden anrichten und vielleicht doch mal irgendeinen Nutzen bringen. Insbesondere vereinfache sich dadurch die Administration, da man bei der Installation von zusätzlichen Diensten in Bezug auf die Sicherheit sorgloser sein kann.

Tatsächlich macht eine Personal Firewall auf dem lokalen Rechner jedoch nichts einfacher, sondern erschwert die Administration eher, weil bei jeder Serverinstallation erstmal neue Regeln erstellt werden müssen. Das kann zu schwer durchschaubaren Fehlern führen, wenn man bspw. im Falle von Samba einen der vielen Ports vergisst freizugeben.

Wann macht ein Paketfilter doch Sinn?

Wer Serverdienste nur in seinem lokalen Netzwerk zulassen möchte, braucht in den allermeisten Fällen auch keinen Paketfilter, da er wohl meistens einen Router (Hardware oder PC) sein eigen nennt, der für ihn *Network Address Translation (NAT)* durchführt. Der Server trägt dann eine lokale Adresse wie 192.168.0.5, die aus dem Internet überhaupt nicht zu erreichen ist, außer der Router führt Forwarding aus, dann dürfte das aber wohl beabsichtigt sein.

Wichtig wird eine Firewall erst, wenn man entweder eigene Serverdienste aus einer DMZ heraus im Netz anbietet, oder wenn ein internes Netz gar öffentliche IP-Adressen besitzt. Davon sollten allerdings Neulinge besser erstmal die Finger lassen und sich informieren. Ein grafisches Tool könnte in dem Fall einem Profi vielleicht helfen, die Konfiguration schneller hinzukriegen, als manuell "iptables zu hacken", sollte aber nicht die Kenntnisse über Netzwerksicherheit ersetzen.

Richtig sinnvoll auf einem Einzelplatzrechner ist ein Paketfilter nur, wenn man nicht die Ports, sondern die Herkunft der Pakete einschränken will. Bspw. möchte man einem Kumpel ermöglichen, den eigenen FTP-Server zu benutzen, oder man will vom Arbeitsplatz aus ssh nach Hause benutzen. So etwas lässt sich oft mit iptables einfacher konfigurieren, als über hosts.allow/.deny oder dienstspezifische Konfigurationsdateien. Besonders, wenn es sich um eine ganze Reihe Dienste handelt, die alle für denselben Zugang vorgesehen sind. Das ist allerdings auch nur dann praktikabel, wenn keine dynamischen IP-Adressen involviert sind. Da die Möglichkeit des Adress-Spoofing besteht, sollte das jedoch niemals starke Authentifizierungsmechanismen des betreffenden Dienstes ersetzen. Gute Passwörter (oder noch besser Public-Keys) sind auch dann unverzichtbar, wenn "nur" der eigene Arbeitsplatzrechner freigegeben ist. Besser könnte es dann wohl sein, gleich auf VPN zu setzen. Es ist außerdem daran zu denken, dass man bei gelegentlichem Zugriff auf einen zu Hause befindlichen Testwebserver o.ä. auch ssh-Tunnel benutzen kann.

In seltenen Fällen mag es vorkommen, dass ein Dienst nicht die Möglichkeit bietet, ihn auf bestimmte Netzwerkschnittstellen zu begrenzen. Insbesondere könnte das bei Software der Fall sein, die nicht von Ubuntu unterstützt wird. In diesem Fall kann einem nichts anderes übrig bleiben, als diesen Qualitätsmangel mit der Implementation von Filterregeln zu kompensieren.

Anhang

Network Address Translation

Network Address Translation (NAT) ist eine Technik, die ursprünglich das Problem knapper IP-Adressen lösen sollte. (Und das auch hinreichend tut.)

Wenn man sich bei einem Dial-Up-Provider einwählt, egal ob per DSL, ISDN, Analog-Modem oder wie auch immer, bekommt man im Normalfall genau eine IP-Adresse für die Dauer der Verbindung zugewiesen. Das wird zu einem Problem, wenn man den Zugang für mehrere Rechner gleichzeitig nutzen will, da nur ein Rechner gleichzeitig diese Adresse nutzen kann. Um dieses Problem zu lösen, braucht man *NAT* und *private Adressbereiche*.

Private Adressbereiche sind Bereiche von IP-Adressen, die für die private Nutzung ohne vorherige Registrierung vorgesehen sind. Jeder darf in seinem Netzwerk diese Adressen benutzen, ohne jemanden vorher fragen zu müssen. Da diese Adressen im Gegensatz zu herkömmlichen IP-Adressen also nicht einzigartig auf der Welt sind, gibt es im öffentlichen Internet keine Möglichkeit, Informationen an diese Adressen zu *routen*, also zu leiten - nicht einmal Antworten auf Anfragen. Die vorgesehenen Bereiche für die private Nutzung sind in einem Dokument festgelegt, dass "RFC-1918" heißt, und umfassen die Adressen **10.0.0.0-10.255.255.255**, **172.16.0.0-172.31.255.255** und **192.168.0.0-192.168.255.255**.

Um also diesen vielen Rechnern in privaten Netzwerken den Zugang zum Internet zu ermöglichen,

wurde die *Network Address Translation*-, genauer gesagt die *Masquerading*-Technik erfunden. Hierbei nimmt ein NAT-Gerät, ein Router, Pakete an einer Schnittstelle zum lokalen Netzwerk entgegen, und leitet sie über die Dial-Up-Verbindung ins Internet weiter, wobei die private Absenderadresse mit der offiziellen IP-Adresse des NAT-Routers überschrieben wird. Gleichzeitig merkt sich der Router, an welchen Rechner er die Antwortpakete für diese spezielle Verbindung zurück leiten muss.

Daraus folgt andererseits, dass der Router keine Möglichkeit hat, Pakete zuzustellen, die keine Antworten auf Verbindungsanfragen sind. Solche Pakete landen dann direkt beim Router, der im Allgemeinen (haarsträubende Konfigurationsfehler mal ausgeschlossen) keine Dienste nach außen anbietet und diese Pakete deswegen zurückweist bzw. verwirft. So ganz nebenbei ist *Masquerading* also ein sehr wirksamer Sicherheitsmechanismus für lokale Netzwerke.

Es gibt auch noch andere Formen des *NAT*, bspw. das Port-Forwarding. Das dient dazu, dass Rechner hinter einem NAT-Gerät doch von außen zu erreichen sind, z.B. Server. Hierzu muss der NAT-Router so konfiguriert werden, dass Anfragen an bestimmte Ports doch in das interne Netz geroutet werden, auch wenn es keine "Antworten" sind. Der Rechner, an den diese Pakete geschickt werden sollen, muss allerdings vom Administrator explizit festgelegt werden. So kann man bspw. alle Pakete, die an Port 80 oder 443 gerichtet sind, an einen internen Webserver weiterleiten lassen.

Filesharing-Tools

Diese öffnen in den meisten Fällen auch Ports nach außen, wie ja schon die Bedeutung des Wortes "Sharing" suggeriert. Dass die meisten inzwischen auch hinter einer Firewall funktionieren, liegt an einem Trick. Systeme, die hinter einer Firewall liegen, fragen einfach in gewissen Abständen ihren Server nach, ob jemand etwas von ihnen haben will, und "pushen" diese Datei dann, indem sie selber eine Verbindung zum Tauschpartner ausführen. Wenn dieser allerdings auch hinter einer Firewall sitzt, funktioniert das nicht, so dass man als Filesharer hinter einer Firewall heutzutage doch ziemlich eingeschränkt ist. BitTorrent bestraft dich dafür sogar mit geringerer Downloadrate. Deswegen öffnen viele Filesharer die Ports extra, oder stellen Port-Forwarding auf ihrem Router ein. Sollte man also auf einem Einzelplatzsystem so eine Software einsetzen, aber der Qualität der Software nicht soweit trauen, dass es nur die freigegebenen Daten rausrückt, könnte man einen Paketfilter einsetzen wollen, der diese Ports für den Zugriff von außen sperrt. Oben beschriebene Nachteile sind dann allerdings unvermeidlich.

Diagnose

Wer sich dafür interessiert, welche Ports auf seinem System gerade offen sind, kann einen Befehl wie

netstat -an --inet | egrep '(Proto|0.0.0.0/*)' | grep -v 127.0.0.1

benutzen.

(Oder

sudo netstat -anp --inet | egrep '(Proto|0.0.0.0/*)' | grep -v 127.0.0.1

dann werden die Prozesse, die für die Ports verantwortlich sind, mit ausgegeben.)

Oder man ruft das grafische Werkzeug **Netzwerkdiagnose** aus dem Gnome-Menü **Anwendungen/Systemwerkzeuge** auf. Im Reiter **Netzwerkstatus** kann man sich dort die **Aktiven Netzwerkdienste** anzeigen lassen. Dabei ist zu beachten, dass dort auch lokale Dienste, zu erkennen an der Adresse **127.0.0.1**, angezeigt werden. Diese sind nicht für andere Rechner erreichbar, und stellen deswegen keine Gefährdung dar.

Um sicher zu gehen, kann man ein System auch mit Hilfe eines *Port-Scanners* von außen testen, wenn man ein eigenes Netzwerk besitzt, oder auch über das Internet. Einen einfachen Port-Scanner beinhaltet das oben erwähnte Tool **Netzwerkdiagnose**. "Profis" benutzen dagegen das

Kommandozeilentool **nmap** von insecure.org, das auch im **main**-Repository von Ubuntu vorhanden ist, und das unzählige Optionen bietet. Auch für dieses Tool gibt es eine grafische Oberfläche namens **nmapfe**, zu finden im **universe**-Repository.