Personal-Firewalls

Beitrag von Vimes

Einleitung und Begriffsabgrenzung

Ein Thema darf bei keiner zünftigen Diskussion über PC-Sicherheit fehlen: Das der Personal Firewalls. Hier scheiden sich die Geister, hier geht es zur Sache. Die einen würden nie ohne surfen, die anderen lassen solche Software nicht mal in die Nähe des eigenen Rechners.

Solche Diskussionen haben im Schnitt einen hohen Unterhaltungswert, da bei dem dort meist herrschenden Umgangston kein Auge trocken bleibt.

Wer bei einer solchen Diskussion ordentlich mitmischen möchte, ohne am Ende den kürzeren zu ziehen, der braucht ein dort recht selten anzutreffendes Gut: Fachwissen. In erster Linie brauchen wir die Grundlagen TCP/IP der letzten Lektion. Die bilden die Basis, um die weiteren Ausführungen zu verstehen. Der Lohn der Mühen: Man kann den einen oder anderen Streithammel mit Hilfe von sachlichen Argumenten gegen die Wand rennen lassen. Ein lässiger Hinweis auf die technischen Gegebenheiten, ein Wink mit den Protokollen und ihrer Wirkungsweise hat da durchschlagende Wirkung.

In dieser Lektion geht es ausschließlich um sogenannte Personal Firewalls oder Desktop Firewalls (beide Begriffe meinen dasselbe).

Definition:
Zitat:

> "Eine Personal Firewall (PFW, auch Desktop Firewall) ist eine Software, die den ein- und ausgehenden Datenverkehr eines PC auf dem Rechner selbst filtert. Dies soll dem Schutz des Computers dienen."

(Quelle: http://www.de.wikipedia.org/wiki/Personal_Firewall)

Die Definition der Wikipedia ist nicht ganz vollständig. Eine Personal Firewall filtert nicht nur ein- und ausgehenden Traffic.

Die Aufgaben einer durchschnittlichen Personal Firewall sind in etwa folgende:

- Filterung des eingehenden Traffics (inkl. "Stealth")
- Kontrolle des ausgehenden Traffics (Phone Home und Co) - das geht über eine reine Filterung weit hinaus! Der Unterschied wird unten erläutert
- Content Filtering und sonstige Sicherheitsfunktionen

Im folgenden beleuchte ich diese Punkte der Reihe nach. Dabei wird zunächst die Funktion vorgestellt und anschließend anhand der technischen Grundlagen gezeigt, ob die Versprechen der Hersteller überhaupt einzuhalten werden können.

Filterung des eingehenden Traffics

Diese Aufgabe erledigt der Paketfilter einer Personal Firewall. Dieser Paketfilter hängt sich vor den TCP/IP-Stack des Betriebssystems, so daß eingehende Datenpakete aus dem Internet (oder dem LAN) zunächst durch seine Eingangskontrolle müssen. Ein Paketfilter kann dabei üblicherweise nur nach Quell-IP-Adresse und entsprechendem Quellport sowie nach Ziel-IP-Adresse und entsprechendem Zielport filtern. (Dazu bitte in der Lektion TCP/IP über IP und TCP nachlesen!)
Im Klartext: Kommt ein Datenpaket von der IP 65.231.235.12 von Port 80 herein und möchte zur eigenen IP auf Port 2128, dann prüft der Paketfilter, ob dafür eine Regel vorliegt. Ist dieser Zugriff nicht erlaubt, dann gibt es zwei Möglichkeiten:

1.) Der Paketfilter wirft das Paket einfach weg. Der Sender bekommt davon nichts mit. Die meisten Programme warten darauf eine Weile und senden das Paket dann einfach noch einmal. Das geht drei Mal so, dann wird der

Verbindungsversuch abgebrochen. Der Vorgang nennt sich DROP.

2.) Der Paketfilter sendet ein Paket zurück, in dem er dem Sender mitteilt, daß von ihm keine Datenpakete entgegengenommen werden. Diesen Vorgang nennt man REJECT. Ein Standard-konformer Sender stellt daraufhin die Kommunikation sofort ein. Diverse Würmer wie z.B. der Blaster ignorieren diese Meldung allerdings.

Bewertung:

Das Verfahren funktioniert vergleichsweise zuverlässig. Allerdings besteht die Möglichkeit, daß die Quelladresse gefälscht ist und damit der Schutz unterlaufen wird.

Außerdem stellt sich prinzipiell die Sinnfrage dieses Vorgehens. Warum?

1. Der Paketfilter erhöht die Menge an Code, die ein Paket durchlaufen muß, ganz beträchtlich. Damit steigt logischerweise auch die Möglichkeit, daß bei der Verarbeitung eines Datenpakets ein Fehler auftritt, der dann eventuell ausgenutzt werden kann. Im Vergleich dazu ist der TCP/IP-Stack der gängigen Betriebssysteme mittlerweile "gut abgehangen", d.h. schon älter und auf Herz und Nieren geprüft.

2. Clients müssen nicht weiter geschützt werden. Der TCP/IP-Stack lehnt alle Pakete ab, die nicht vom Client selbst angefordert wurden. Selbst wenn der Stack versagen würde, würde der Client selbst ein nicht angefordertes Paket abweisen.

3. Server bzw. Dienste, die vom Internet (oder LAN) aus erreichbar sein müssen, profitieren nicht von der Personal Firewall, denn sie müssen ja freigeschaltet werden. Sollen sie nicht vom Internet aus erreichbar sein, so schaltet man sie sinnvollerweise ab. Dann verwirft der TCP/IP-Stack alle Anfragen an diesen Port, weil dahinter kein
Programm/Dienst mehr lauscht.

Fazit:

Personal Firewalls erledigen diese Aufgabe recht zuverlässig, ich erlaube mir allerdings, den Sinn in Frage zu stellen. Ein Paketfilter macht dann Sinn, wenn ich einen Dienst / Server laufen habe, der nicht von außen erreichbar sein soll, aber nicht abgestellt werden kann / darf (warum auch immer).

Sonderfall Stealth

Manche Hersteller werben damit, daß ihre Produkte den Rechenknecht "unsichtbar" machen würden.

Das ganze funktioniert technisch folgendermaßen: ein anderer PC sendet ein Datenpaket, z.B. den Versuch, eine TCP/IP-Verbindung aufzubauen. Normalerweise würde der Zielrechner nun entweder den Verbindungsversuch bestätigen und damit den zweiten Teil des 3-Wege-Handschlags einleiten (siehe Lektion TCP/IP) oder aber dem Sender mitteilen, daß eine Kontaktaufnahme nicht erwünscht ist. Stattdessen läßt die Personal Firewall das Datenpaket einfach fallen und sendet keinerlei Antwort zurück. Der Sender bekommt also keinerlei Auskunft, was aus seiner Anfrage geworden ist.

Bewertung:

Stealth ist eine Marketing-Erfindung. Es ist außerdem eine grobe Verletzung der Internet-Standards, die da vorschreiben, daß eine Anfrage nicht einfach kommentarlos fallengelassen wird.

Welchen Sinn macht es, einen Rechner "voll krass stealth" zu machen?

Früher: Absolut keinen. Denn der Internet-Provider würde eine Fehlermeldung zustellen, falls der angesprochene Rechner nicht am Netz hängen würde ("Host unreachable"). Daher wüßte ein Angreifer sofort, daß da jemand lauscht, aber nicht kommunizieren möchte.

Heute: Leider sind einige Provider dazu übergegangen, solche Fehlermeldungen nicht mehr herauszugeben.

Damit würde "Stealth" technisch gesehen funktionieren... wenn nicht. Ja, wenn nicht...

1.) die allermeisten aktuellen Würmer einfach drauflosballern und sich um fehlende Antworten einen feuchten Kehricht scheren würden

2.) das die Cracker von heute auch wüßten und deswegen trotzdem automatisiert ihre Exploits ausprobieren

3.) man sich damit wunderbar ins eigene Knie schießen kann. Ist mir einmal selbst passiert: ich wollte testen, ob meine Internetverbindung steht und sandte daher ein ping an www.yahoo.de. Die Pakete gingen verloren, ich suchte wüst fluchend eine Viertelstunde lang den Fehler (unter Gentoo Linux). Bis ich auf die Idee kam, mal eine andere Adresse anzupingen. Lösung: yahoo.de ließ pings einfach unter den Tisch fallen. (Anmerkung: Dieses hirntote Verhalten wurde mittlerweile behoben.) Genau dasselbe kann auch im Heimnetzwerk bei der Fehlersuche passieren.

Fazit:

Stealth bringt technisch gesehen keinen Vorteil, behindert die Fehlersuche im Heimnetzwerk und verstößt gegen technische Standards. Im schlimmsten Falle signalisiert man damit einem technisch versierten Angreifer, daß hier jemand auf Marketing-Lügen reingefallen ist und damit wahrscheinlich nicht viel von PC-Sicherheit versteht.

Kontrolle des ausgehenden Traffics

Wir müssen hier genau hinsehen. Eine Personal Firewall kann auf dem "Rückweg" auch das, was sie auf dem "Hinweg" kann: Traffic filtern. D.h., wenn ein Datenpaket nach draußen möchte, dann wird überprüft, ob es diese IP auf diesem Port ansprechen darf. So könnte man z.B. eine Liste von gesperrten IPs anlegen und dazu ggf. den Port mit angeben. Oder generell den Port 1048 für ausgehenden Traffic sperren. Das funktioniert recht zuverlässig, weil diese Regeln für alle Programme auf dem Rechner greifen.

Leider kann dieser "Schutz" recht einfach dadurch umgangen werden, daß ein Programm einfach einen anderen Port öffnet und ggf. eine andere IP anspricht - die Schreiber von Schadsoftware sind flexibel.

Die Verkäufer von Personal Firewalls (im folgenden nur noch "PFW" genannt) versprechen aber noch mehr als die reine Filterung nach Ziel-IP und Port. Sie versprechen eine gezielte Kontrolle des ausgehenden Datenverkehrs. Damit soll es möglich sein, das "Nachhause-Telefonieren" von Programmen unterbinden zu können, sei es das Office-Paket, eine Freeware, der man nicht recht über den Weg traut oder im schlimmsten Falle der Grieche in einem trojanischen Pferd, der mit seinem Herrn und Meister Kontakt aufnehmen möchte.

Das Prinzip ist dabei, daß die PFW überwacht, welche Programme gerne auf das Internet zugreifen möchten. Im Zweifel wird nachgefragt, ob der Benutzer das erlaubt. Über Bibliothekenfunktionen kann auch erfaßt werden, ob ein Programm klammheimlich ein anderes Programm startet, um über dieses ins Internet zu gehen. (Anmerkung: diese Überwachung ist nicht perfekt. Meines Wissens existieren auch hier zumindest unter Windows Wege, das ganze auszuhebeln, ohne die unten aufgeführten Trick zu benutzen.)

Klassische Schadsoftware

Der Chaos Computer Club Ulm hat in diesem Zusammenhang die gängigen PFWs getestet.

Das Video dazu gibt es hier:

http://ulm.ccc.de/chaos-seminar/personal.../recording.html

Die Vorträge finden sich hier:

http://copton.net/vortraege/pfw/index.html

Im folgenden stelle ich das Szenario kurz vor. Zunächst die Grundlagen:

Die Test-PCs waren mit allen Wassern gewaschen, d.h. alle Sicherheitsupdates waren eingespielt worden und der

angemeldete Benutzer verfügte nicht über Administratorrechte.

Die verwendeten PFWs waren "state-of-the-art", also nicht etwa völlig veraltete Modelle.

Angriff Nr. 1: Wer drückt hier Tasten?

Der Angriff wurde von Alexander Bernauer mit einem Programm namens wwwsh durchgeführt.

Die Funktionsweise ist denkbar simpel. wwwsh simuliert eine Eingabe des Anwenders über die Tastatur und startet auf diese Weise den Internet Explorer. Über die eingegebene URL werden dann Daten verschoben, d.h. die Nutzdaten werden in der URL eingebaut. Natürlich kann man damit nicht auf einen Schlag gigantische Datenmengen übertragen, aber der Angriff ist auch nur ein Proof of Concept.

Wieso merkt das die PFW nicht?

Ganz einfach, es gibt unter Windows exakt keine Möglichkeit, eine Eingabe über die Tastatur (kinetische Energie wirkt auf Tasten ein) von einer Simulation derselben zu unterscheiden. Es ist technisch nicht machbar. Letztlich werden Tasteneingaben ja auch nur vom PC verarbeitet. Es existieren keine gesicherten Schnittstellen.

Sämtliche PFWs versagten bei diesem Angriff kläglich. Das Programm benötigte auch keine Administratorenrechte.

Angriff Nr. 2: Nervöser Zeigefinger und nervende PopUps

Fast alle PFWs fragen nach, ob ein Programm Daten versenden darf. Der Anwender kann das dann über einen Mausklick auf das aufpoppende Fenster bestätigen.

Die Gemeinheit daran: Das kann eine Schadsoftware auch! Zunächst löst sie über einen ganz normalen Zugriffsversuch das Popup-Fenster aus. Anschließend bestätigt sie die Anfrage einfach selbst. Sie benötigt dafür auch keine Administratorenrechte! Schuld ist wiederum das System, mit dem Windows interne Ereignisse verarbeitet. Die Problematik ist damit prinzipbedingt und läßt sich nicht beheben.

Auch hier waren sämtliche getesteten PFWs gegen den Angriff wehrlos.

Fairerweise muß man sagen, daß man dieses Nachfragen den meisten PFWs abgewöhnen kann. In der Standardeinstellung fragen sie allerdings munter nach.

Ich bin root, ich darf das...

Das ist noch nicht alles. Häufig genug arbeiten Anwender eben nicht als eingeschränkte Benutzer, sondern als Administratoren. Daß das keine sonderlich gute Idee ist, merkt man spätestens dann, wenn eine Schadsoftware die lästige PFW kurzerhand beendet. Selbst wenn das einem aufmerksamen Anwender rasch auffällt, kann es bereits zu spät sein.

Fazit:

Prinzipbedingt ist es nicht möglich, das "raustelefonieren" von Schadsoftware zu verhindern.

Davon ab: Ist Schadsoftware auf dem Rechner zur Ausführung gelangt, ist der Rechner nicht mehr als vertrauenswürdig anzusehen. Damit gehört er sowieso formatiert und neu aufgesetzt (bzw. ein sauberes Image zurückgespielt). Damit erübrigt sich eigentlich die
Notwendigkeit, ausgehenden Datenverkehr kontrollieren zu wollen.

Nach Hause telefonieren - Dein Office hat Heimweh

Aber es bleiben ja noch die klassischen Phone-Home-Programme, die keine Schadsoftware im eigentlichen Sinne

sind.

Hier gilt: solche Software läßt sich in aller Regel so konfigurieren, daß sie nicht mehr telefoniert. Ist das nicht möglich, sollte man über die Notwendigkeit des Einsatzes nachdenken. Hat man für die Software bezahlt, empfiehlt es sich, den Hersteller zu treten.

MS spioniert mir nach...

Zu guterletzt gibt es noch die Fraktion, die Windows selbst am Nach-Hause-telefonieren hindern möchte.

Dazu überlege man sich folgendes: Man führt eine Software auf der Basis eines Betriebssystems aus, um dieses Betriebssystem dann zu kontrollieren. Klingt irgendwie so, als hätte die Sache einen massiven Haken, nicht? Richtig. Das Betriebssystem stellt der Software überhaupt erst alle Schnittstellen und sonstige Mittel zur Verfügung, damit diese ihrer Aufgabe nachgehen kann.

Es gilt: Traust Du Deinem Betriebssystem nicht, schalt den Rechner aus... alles andere ist Augenwischerei.

Content Filtering und sonstige Sicherheitsfunktionen

Die Idee ist an sich ist gut: Die PFW soll verhindern können, daß bestimmte Inhalte übertragen werden. In beide Richtungen. So könnte man z.B. verhindern, daß jemand von außen über eine Webseite schädlichen Code auf den Rechner schleust oder ähnliches.

Dafür muß die PFW allerdings die verwendeten Protokolle (s. bitte TCP/IP und Protokolle) auch alle sprechen, d.h. den übertragenen Inhalt verstehen. Das ist der erste Punkt. Der zweite ist der, daß eine PFW mangels hellseherischer Fähigkeiten nicht wissen kann, ob der Anwender nicht bestimmten Code ausgeführt haben möchte. Da kämen wieder die bunten Fenster ins Spiel...

PERSÖNLICHES FAZIT

Im ersten Teil habe ich gezeigt, daß eine PFW eingehenden Datenverkehr filtern kann, das aber in aller Regel überhaupt nicht notwendig ist.

Im zweiten Teil zerplatzten die Versprechen, eine PFW könnte ausgehenden Datenverkehr verhindern oder filtern, wie eine Seifenblase.

Da stellt sich natürlich die Frage: Haben PFWs überhaupt einen Sinn und Zweck?

Wer darauf angewiesen ist, eingehenden Datenverkehr zu filtern, aber aus irgendwelchen Gründen keine bessere Lösung dafür nehmen kann, der kann dafür eine PFW benutzen.
Man sollte dabei allerdings im Hinterkopf behalten, daß PFWs durchaus auch eigene Schwachstellen aufweisen können (Stichwort: Witty-Wurm). Im Klartext: Bei dem Versuch, eingehenden Datenverkehr zu filtern, reiße ich mir eventuell Sicherheitslöcher auf. So öffnen manche PFWs Ports nach außen - d.h., sie lauschen am Internet und tun damit genau das, was sie versprechen, zu unterbinden.

Eine bessere Alternative ist der bei Windows XP eingebaute Paketfilter, der erheblich schlanker ist und still seine Aufgabe erledigt, nämlich eingehende Pakete zu filtern, ohne den Anwender zu belästigen.

Wer ausgehenden Datenverkehr filtern oder begrenzen möchte, der sollte darauf achten, welche Programme er installiert. Aus den oben angeführten Gründen sind PFWs dafür absolut untauglich.

Wer sich vor Scripten etc. schützen möchte, der nutze einen sicheren Browser, konfiguriere diesen richtig und klicke nicht auf alles, was bei drei nicht verschwunden ist.

Eine letzte Schwäche von PFWs habe ich bisher bewußt verschwiegen.

PFWs müssen konfiguriert werden. Dafür benötigt man aber zumindest solide Grundkenntnisse in Sachen TCP/IP. Wer die nicht hat, der weiß nicht, was er tut. Die Behauptungen der Hersteller, man müssen einfach nur die Software installieren, ein paar Schieberegler betätigen und dann wäre man sicher, sind genau das - Behauptungen.
Beweis: Wie soll ich Filterregeln für Ports und IPs aufstellen können, ohne einen blassen Schimmer davon zu haben, was das eigentlich ist?
Und noch schlimmer: Was nützt es mir, wenn PFW mich fragt, ob die svchost.exe auf das Internet zugreifen können soll, wenn ich keinen Plan habe, was dieses Programm ist und was es macht?

Wer weiß, was er tut, der kann mit einer PFW arbeiten. Wer das aber weiß, möchte es fast immer eigentlich nicht, da eine PFW Schwachstellen aufweisen kann, das System ausbremst - und weil so jemand sein System so einstellen kann, daß es keiner PFW mehr bedarf...

Durch längeres Mitlesen in einschlägigen Newsgroups habe ich mittlerweile folgende persönliche Meinung über PFWs:

Eine PFW ist mit dem Befall durch einen Schädling gleichzusetzen - in beiden Fällen hilft nur noch "platt machen" und neu aufsetzen.
Denn häufig versaut eine PFW den TCP/IP-Stack des Betriebssystems so sehr, daß man mit dem "Resultat" kaum noch vernünftig arbeiten kann. Von den Sicherheitslücken gar nicht zu reden.