Sicherung eines heterogenen Netzwerkes mit freien Software–Programmen

by Georges Tarbouriech
<georges.t(at)linuxfocus.org>

About the author:

Georges ist ein langjähriger Unix–Benutzer. Er dankt der Freien–Software–Gemeinschaft für das Bereitstellen großartiger Sicherheits–Werkzeuge.

Translated to English by:
Georges Tarbouriech
<georges.t(at)linuxfocus.org>

Abstract:

Dieser Artikel wurde zuerst in einer Linux Magazine France Spezialausgabe mit Schwerpunkt Sicherheit veröffentlicht. Der Redakteur, die Autoren und Übersetzer erlaubten LinuxFocus freundlicher Weise jeden Artikel dieser Spezialausgabe zu veröffentlichen. Daher wird LinuxFocus sie veröffentlichen, sobald sie ins Englische übersetzt sind. Dank an alle Leute, die an dieser Arbeit beteiligt sind. Dieser Abschnitt wird bei jedem Artikel derselben Herkunft stehen.

Vorwort

Sicherheit in Computer–Netzwerken ist wahrscheinlich eine der größten technologischen Herausforderungen des 21. Jahrhunderts.
Jedoch, wie in vielen Besorgnis erregenden Bereichen, jeder redet darüber, aber diejenigen, die sich am meisten betroffen fühlen sollten, scheinen den Umfang des möglichen Desasters nicht entdeckt zu haben. Die "hauptsächlich Betroffenen" sind natürlich die wichtigsten Software– oder System–Designer. Das beste Beispiel kommt erneut aus Redmond, wo Sicherheit als ein Wort erscheint, das viel weniger "unter Kontrolle" ist als z. B. Marketing.
Zum Glück haben die letzten beiden Jahrzehnte des 20. Jahrhunderts die Geburt der Freien Software und die damit verbundene Philosophie erlebt. Wenn Sie es "wünschen", die Sicherheit Ihrer Maschinen, Ihrer Systeme, Ihrer Netzwerke zu verbessern, ... dann müssen Sie danach Ausschau halten. Die Freie–Software–Gemeinschaft hat mehr für die Sicherheit getan als alle großen Software–Firmen zusammen.

D. h., die Werkzeuge allein schaffen nicht alles, und z. B. die Sicherung eines Netzwerkes ist eine ständige Aufgabe: jederzeit gibt es neue Herausforderungen!
Das bedeutet, dass Sie niemals sagen können, dass ein Netzwerk 100 Prozent sicher ist. Sie können die Risiken nur reduzieren. Was wir hier zeigen, ist nur ein kleiner Teil dessen, was Sie tun können, um diese Risiken zu reduzieren. Nach dem Lesen dieser Spezialausgabe (Hinweis des Autors: Denken Sie daran, dieser Artikel war Teil einer Spezialausgabe des französischen Linux−Magazins, dessen Schwerpunkt auf Sicherheit lag) werden Sie etwas mehr über Sicherheit wissen, aber auf keinen Fall werden Sie sagen können, dass Ihr Netzwerk sicher sei. Sie sind gewarnt.
Vor allem: solch ein Artikel kann nicht erschöpfend sein. Es gibt eine Menge Literatur zu dem Thema und das Problem ist immer noch nicht erschöpfend behandelt. Entsprechend sollten Sie von diesem Artikel nicht erwarten, alles zu erwähnen, was Betriebssysteme, Werkzeuge, Konfiguration, Benutzung ... betrifft.
Zum Schluss dieses Vorwortes sollten wir hinzufügen, dass einige Teile dieses Artikels von LinuxFocus entliehen sind, aber keine Sorge, dies geschah mit der Erlaubnis des Autors, es stellt sich heraus, dass es sich um die gleiche Person handelt!

Präsentation

Zunächst sprechen wir über die Struktur eines sehr heterogenen Netzwerks, das mehr oder wenig weit verbreitete Systeme enthält. Je mehr Betriebssysteme, umso schlimmer ist die Komplexität, da nicht alle Systeme gegenüber Bedrohungen gleich sind. Desweiteren sollten die als Server benutzten Maschinen in einem Netzwerk verschiedene Funktionen haben; wir werden ein sehr ungleichartiges Netzwerk haben.
Als nächstes werden wir einige Werkzeuge besprechen, die essentiell für die Verbesserung der Sicherheit sind. Die Auswahl ist willkürlich: es gibt einfach zu viele, um alle zu erwähnen. Natürlich werden wir erklären, wie man Maschinen und Netzwerke mit diesen Werkzeugen sichert. Der folgende Abschnitt betrachtet die Fähigkeiten verschiedener Systeme während des Sicherungsprozesses.
Der Abschluss versucht, die "Relativität" des Sicherungsprozesses zu erklären, um zu zeigen, warum dies ein langer Weg ist, ohne in zukünftige Entwicklungen "abzutauchen".

Beispiel eines heterogenen Netzwerkes

Erster Vorteil ist, dass das TCP/IP−Protokoll von jedem Betriebssystem auf Erden "gesprochen" wird. Damit sind sehr verschiedene System in der Lage, miteinander zu kommunizieren. Entsprechend wird TCP/IP in unserem Beispiel−Netzwerk immer präsent sein. Anders gesagt, werden wir keine proprietären Protokolle erwähnen, weder die weniger verbreiteten noch die überholten. Wir werden auch nicht über die physikalische Struktur sprechen, also die Verbindungsart, die Kategorie usw.
Wir werden in dieses Netzwerk etwas von allem packen. Natürlich finden wir Unix, proprietär oder frei: z. B. etwas Solaris 2.6 oder SunOS 5.6, wenn Sie möchten, etwas Irix 6.5, Linux (RH 6.2), MacOS X. Wir hätten etwas von QNX oder NeXTSTEP, NetBSD oder OpenBSD hinzufügen können. Auf der "konventionellen" Seite schliessen wir das eine und einsame Nicht Terminiert 4.0 (nein, kein anderes, die sind schlimmer) ein. Auch hier könnten wir OS/2 hinzufügen, was weniger schlimm ist. Zum Schluss fügen wir etwas "unkonventionelles" hinzu wie BeOS und AmigaOS (ja, es existiert ... nun ja, nicht sehr!)
Natürlich werden sich einige schon beschweren: was, kein AIX, kein HP−UX? Nein! Wenn wir jedes Unix erwähnen wollten, würde dies ein 10bändiger Artikel. Die grundlegenden Sicherheitsregeln sind für alle Systeme anwendbar.

Nun, was verlangen wir von ihnen?
Z. B. soll Solaris als Anwendungsserver dienen. Irix wird die Sicherungen verwalten. NT wird ein weiterer

Anwendungsserver sein. Linux dient als Gateway. Ein weiterer Linuxrechner wird ein HTTP−Server oder ein Datenbank−Server sein. Alle anderen Maschinen sind Klienten. Wir überlegen, dass dieses Netzwerk ungefähr 30 Maschinen umfasst, die Authentifizierung über Passwort−Dateien benutzen. Wir hätten eine bessere Authentifizierung benutzen können: NIS (Yellow Pages) oder LDAP oder Kerberos ... Lasst uns die Dinge einfach halten! Auch werden wir kein NIS benutzen. Selbst wenn es hilfreich sein kann, wo es um Sicherheit geht, vergessen Sie es besser, trotz einiger Fortschritte. In Frankreich pflegen ältere Leute zu sagen "Lege nicht alle Deine Eier in den gleichen Korb". Ferner werden die "unsicheren", aber erforderlichen Dienste oder Protokolle nur einmal vorhanden sein, und zwar auf Maschinen, die sonst nichts anderes machen. Z. B. nur ein FTP−Server, ein HTTP−Server, vorzugsweise auf Unix−Rechnern. Einige Unix−Rechner werden SSH−Server sein und die anderen SSH−Klienten. Mehr dazu später. Wir werden statische IP−Adressen nutzen: kein DHCP. Anders ausgedrückt: wir bleiben grundlegend! Dies kann natürlich bei einem Netzwerk mit 50 Maschinen angewandt werden; mit viel mehr Maschinen kann dies zu einem Alptraum werden.

Werkzeuge und wie man sie benutzt

Wie üblich, gibt es mehr als einen Weg zum Ziel. Der Idealfall wäre bei Null anzufangen, mit neu zu installierenden Maschinen und einem aufzusetzenden Netzwerk. Aber dies ist nur in Filmen wahr! Dementsprechend sollten wir ein Netzwerk betrachten, das im Lauf der Zeit gewachsen ist, mit Maschinen, die von einer Stelle zu anderen umgesetzt wurden, neue Maschinen kommen hinzu und so weiter. Aufgrund des Mhz−"Rennens" z. B. halten Intel−Maschinen heute nicht mehr lang vor. Nach ungefähr 3 Jahren wird es schwierig, Ersatzteile zu finden. Daher recycelt man diese Maschinen für nachgeordnete Aufgaben oder entsorgt sie gleich: traurig, aber wahr! Glücklicherweise halten einige viel länger und verdienen es, verbessert zu werden. Glauben Sie nicht, dass dies nicht zum Thema gehört: ein Administrator muss bei seiner Arbeit auch hohe Verfügbarkeit im Sinn haben.

Grundlegendes

Wir könnten den ersten Schritt der Arbeit "Allgemeines" nennen. Er besteht darin, alles Nutzlose auf jeder Maschine zu entfernen: keine "einfache" Aufgabe! Jedes Betriebssystem, Unix eingeschlossen, installiert eine unglaubliche Anzahl an Diensten, Protokollen, die Sie niemals benutzen werden. Die Anweisung lautet: Schmeiss sie weg! Unter Unix ist ein einfacher ... und grober Weg, alles in inetd.conf auszukommentieren. Das bedeuet einige Dienste weniger. Natürlich ist dies etwas übertrieben, aber auf manchen Maschinen ist es akzeptabel. Es hängt von Ihren Bedürfnissen ab. Unter Linux und einigen anderen BS können Sie auch das *chkconfig*−Kommando benutzen, um einige Dienste zu deaktivieren.
Überprüfen Sie auch die SUID/SGID−Dateien und zögern Sie nicht, das "fehlerhafte" Bit zu entfernen oder überlegen Sie, das Programm zu deaktivieren. Ein Befehl wie: *find / −user root −a \(−perm −4000 −o −perm −2000 \) −print* gibt Ihnen die Liste dieser Dateien. Um das "s"−Bit zu entfernen, geben Sie *chmod a−s programname* ein (Hinweis: Natürlich verlieren Sie einige Funktionalität durch das Entfernen des "s"−Bits, aber es hat schon seinen Zweck).
Entfernen Sie "gefährliche" Programm oder die als "riskant" bekannten: Die remote−Befehle wie rsh, rlogin, rcp... z. B. SSH kann sie sehr gut ersetzen.
Überprüfen Sie die Berechtigungen für Verzeichnisse wie /etc, /var ... Je restriktiver, um so besser. So ist z. B. ein Befehl wie *chmod −R 700* auf das Verzeichnis, das die Startup−Dateien enthält (/etc/rc.d/init.d in vielen Unix−Versionen) keine schlechte Idee. Die gleiche Regel findet Anwendung auf alle Systeme, die Teil des Netzwerkes sind: entfernen Sie, was Sie nicht benutzen oder deaktivieren Sie es zumindest. Unter NT sollten Sie sich frei fühlen, ein Maximum an Diensten über das Konfigurationspanel zu stoppen. Es sind viele grundlegende "Dinge" zu tun und es gibt eine Menge Literatur zu dem Thema.

Die Werkzeuge

Lassen Sie uns mit Unix beginnen, weil dies das einzige BS ist, das Sicherheitsprobleme wirklich ernst nimmt. Desweiteren gibt es eine riesige Menge freier Werkzeuge, und die meisten von ihnen arbeiten auf (fast) allen Unix−Geschmacksrichtungen.

Zuerst werden wir auf den einzelnen Maschinen arbeiten, weil die Sicherung eines Netzwerkes zuallererst bedeutet, dessen Elemente zu sichern. Die Installation dieser Werkzeuge ist recht einfach, daher werden wir uns nicht lange damit aufhalten. Deren Parameter sind außerdem abhängig von dem System, dessen Anforderungen ... Es liegt an Ihnen, wie Sie dies für Ihre Situation anpassen. Das erste erforderliche Werkzeug nennt sich *shadow utils*. Es ist eine Möglichkeit zur Passwort−Verschlüsselung. Zum Glück ist es Teil vieler Unix−Distributionen. Die /etc/shadow−Datei wird aus der /etc/password−Datei "kreiert".

Noch besser, *PAM* (Pluggable Authentication Modules) erlauben es, den Zugang pro Dienst zu beschränken. Alles wird aus dem Verzeichnis verwaltet, das die Konfigurationsdateien für jeden betroffenen Dienst enthält, normalerweise /etc/pam.d. Viele Dienste können über PAM "gesteuert" werden, wie ftp, login, xdm usw. Damit wird dem Administrator die Möglichkeit gegeben auszuwählen, wer was tun darf.

Das nächste Werkzeug muss man haben: *TCPWrapper* . Es funktioniert ebenfalls auf jedem Unix oder auf fast jedem. Kurz gesagt erlaubt es, den Zugriff auf Dienste auf einige Rechner zu beschränken. Diese Rechner werden zugelassen oder abgewiesen über zwei Dateien: /etc/hosts.allow und /etc/hosts.deny. TCPWrapper kann auf zwei Arten konfiguriert werden: entweder durch das Verschieben der Dämonprogramme oder das Ändern der /etc/inetd.conf−Datei. Später werden wir sehen, dass TCPWrapper sehr gut mit anderen Werkzeugen zusammenarbeitet. Sie finden TCPWrapper unter ftp://ftp.porcupine.org/pub/security

Ein weiteres interessantes Programm ist xinetd. Erneut, um es kurz zu machen, xinetd ist ein Ersatz für inetd mit viel mehr Fähigkeiten. Entsprechend dem, was wir oben über inetd gesagt haben, werden wir nicht darauf bestehen. Wenn Sie interessiert sind, finden Sie es unter http://www.xinetd.org.

Unter Linux gibt es ein Werkzeug, ohne das Sie nicht leben können: Es wird Bastille−Linux genannt. Sie finden es unter http://www.bastille−linux.org. Dieses in Perl geschriebene Werkzeug ist nicht nur didaktisch, sondern auch sehr effizient. Nach dem Start eines Skripts beantworten Sie viele Fragen und Bastille−Linux agiert entsprechend. Jede Frage ist erläutert und es werden Fehlwert−Antworten geboten. Sie können die Änderungen zurücknehmen, eine neue Konfiguration starten, prüfen, was angerichtet wurde ... Es ist alles vorhanden! Es bietet außerdem eine Firewall−Konfiguration: mehr dazu später. Zum Zeitpunkt dieses Artikels ist Bastille−Linux bei Version 1.1.1, aber die Version 1.2.0 ist bereits als Veröffentlichungs−Kandidat verfügbar. Es ist sehr verbessert, und bietet eine grafische Oberfläche, die auf Tk und dessen Perl−Modul basiert. (Hinweis des Autors: Dieser Artikel wurde schon vor einigen Monaten geschrieben. Tatsächlich ist die aktuelle Version von Bastille−Linux 1.3.0).

Einbruchs−Entdeckungs−Systeme sind auch wichtig. Die zwei "Schwergewichte" sind snort und portsentry. Das erste findet sich unter http://www.snort.org und das zweite unter der Abacus−Webseite, http://www.psionic.com. Diese beiden sollten nicht miteinander verglichen werden: das erste ist ein NIDS (Network Intrusion Detection System), welches hauptsächlich Informationen bereitstellt, während das zweite als mehr Rechner−orientiert und aktiver betrachtet werden kann. Snort hat eine Menge an Optionen zur Überwachung des Netzwerkverkehrs. Sie können auf alles lauschen, was Sie möchten: eingehend, ausgehend, innerhalb der Firewall, außerhalb der Firewall. Natürlich kann es dann riesige Log−Dateien erzeugen, aber Sie müssen wissen, was Sie wollen! Eine Win32−Version ist verfügbar, dies ist wichtig, wenn wir die Anzahl freier Werkzeuge auf diesen "Systemen" zählen.

Portsentry hat eine sehr interessante Fähigkeit: Es kann die überprüften Ports entsprechend Ihrer Wahl blockieren: Entweder leiten Sie den Angreifer auf eine unbenutzte Adresse oder zur Firewall um. Natürlich können Sie auswählen, wer blockiert wird und wer nicht. Nun kommen wir zurück zu TCPWrapper: portsentry kann in die Datei /etc/hosts.deny schreiben, wenn Sie dies möchten. Dadurch wird portsentry sehr

effizient. Wir werden uns nicht an der Debatte über die portsentry−Philosophie über Port−Bindungen einmischen. Es liegt an Ihnen: Treffen Sie Ihre Entscheidung, wenn Sie sich tiefer mit der Materie beschäftigt haben. Außerdem sollten Sie wissen, dass portsentry eine Maschine "unsichtbar" machen kann, was nicht schlecht ist!. Zu guter Letzt kann portsentry verschiedene Betriebsmodi nutzen, wobei die fortschrittlichsten (zumindest derzeit) für Linux "reserviert" sind.

Wir können nicht über Sicherheit reden ohne Verschlüsselung zu erwähnen. Die Gesetze hierüber sind jedoch von Land zu Land verschieden und manchmal ist es komplett verboten, Verschlüsselung zu benutzen.
Hinweis des Autors: Der folgende Abschnitt wurde aus der englischen Version dieses Artikels entfernt, da er nur das französische Recht betrifft.
Schlussfolgerung: Wenn Ihr Land Verschlüsselung erlaubt, dann sollten Sie ssh−Clients und −Server auf Ihren Unix−Rechnern installieren (entsprechend Ihren Anforderungen).

Um das Thema Unix−Werkzeuge abzuschliessen, sollten wir diejenigen erwähnen, die zu proprietären Unix−Versionen gehören. Unter Solaris gibt es ndd, aset; unter Irix können Sie ipfilterd nutzen. MacOS X bietet Ihnen einige freie Werkzeuge: ssh, ipfwadm ...
Wir kommen später darauf zurück.

Nun lassen Sie uns über das eine und (glücklicherweise) einsame Nicht Terminiert 4.0 reden. Hier können wir nicht über freie Werkzeuge reden ... der Mann aus Redmond bietet uns jedoch "freie" Sachen, um die Systemfunktionalitäten zu verbessern (es hat nichts mit Fehlerkorrekturen zu tun, da es dort keine Fehler gibt!). Was Sicherheit angeht, ist NT 4.0 ein Modell ... an Absurdität. Es ist wie ein Sieb. Trotzdem. Sie müssen nur entsprechend das letzte Servicepack (6 z. Z. dieses Artikels) herunterladen und die HotFixe ... die Sicherheitspatches sind. Als nächstes können Sie einige freie Werkzeuge (im Sinne von frei verfügbar und ohne den Quellcode) erhalten. Das ist alles.

Für andere Systeme müssen Sie suchen. Beim AmigaOS scheint die Entwicklung nicht viele Leute zu motivieren und der TCP/IP−Layer ist etwas alt. Es gibt jedoch immer noch eine aktive Public−Domain−Szene. Was BeOS angeht, sind die Dinge nicht besser: dieses großartige Betriebssystem hat eine sehr schlechte Zukunft und der Bone genannte Netzwerk−Layer ist immer noch in Entwicklung.
Hinweis des Autors: Unglücklicherweise ist BeOS nun wirklich tot. Einige wenige Personen versuchen, es als freies Software−Produkt zu erhalten ... und sie machen ihre Sache sehr gut.
Aber auch hier finden Sie einige Werkzeuge aus der Unix−Welt, um die Lage etwas zu verbessern.

Sicherung der Rechner

Nun müssen Sie das alles konfigurieren! Lassen Sie uns nun wieder annehmen, dass jede Unix−Maschine mit shadow−utils, PAM, und TCPWrapper "ausgerüstet" ist, dass jeder nutzlose Dienst gestoppt oder entfernt wurde und dass die Rechte auf "sensiblen" Verzeichnissen eingeschränkt wurden.

Auf den Linux−Maschinen ist es nun an der Zeit, Bastille−Linux zu starten. (Dieses Werkzeug sollte auf den meisten Linux−Distributionen laufen, es war jedoch ursprünglich für RedHat und Mandrake gedacht). Fühlen Sie sich frei, die Fragen sehr restriktiv zu beantworten.
Die Linux−Maschine, die als Gateway genutzt wird, sollte sehr "minimalistisch" eingerichtet sein. Sie können die meisten Server entfernen: http, ftp, usw. Löschen Sie X11, Sie brauchen es nicht! Entfernen Sie die nicht benötigte Software ... d. h. fast alles. Stoppen Sie die nutzlosen Dämonen. Sie sollten ein System erhalten, auf dem *ps ax* nicht mal eine Bildschirmseite benötigt. Wenn Sie IP−Masquerading nutzen, sollte der *lsof −i*−Befehl nur eine Zeile anzeigen: diejenige, die sich auf den lauschenden Server bezieht (wir setzen voraus, dass dies keine permanente Verbindung ist).
Wir werden portsentry auf den Linux−Maschinen installieren und es wird gleich beim Systemstart aktiv unter Benutzung des "fortgeschrittenen" Modus (reserviert für Linux, d. h. mit den −atcp− und −audp−Optionen). Dies impliziert, dass TCPWrapper und eine Firewall installiert wurden. Mehr dazu später.

Für Solaris nutzen wir die *aset*– und *ndd*–Befehle. Wir könnten IP–Filter hinzufügen und die Standard–Version von RPCbind mit der von porcupine.org erhältlichen Version 2.1 ersetzen. Für Irix wählen wir ipfilterd für die Paketfilterung, wie der Name schon sagt. Es ist Teil der Irix–Distributionen, wird aber standardmäßig nicht installiert.

Was NT angeht, werden die Dinge etwas komplizierter ... die "faschistische" Lösung besteht aus dem Blockieren der Ports 137 und 139, das berühmte NetBIOS (oder noch besser Entfernen von NetBIOS) ... aber dann ist kein Netzwerk mehr vorhanden (d. h. kein Windows–Netzwerk). Dies kann ein kleines Problem sein, wenn es um einen Anwendungsserver geht! Sie können auch snort installieren, aber es hindert diese Maschinen nicht daran, wie ein Sieb zu sein. Entsprechend müssen Sie sehr restriktiv sein über Partitionsberechtigungen, Verzeichnisberechtigungen ..., natürlich nur, wenn Sie mit NTFS–Partitionen arbeiten. Es gibt ein frei verfügbares Programm, dass das Gast–Konto entfernt, aber der Quellcode ist nicht verfügbar. Desweiteren sollten Sie alle Sicherheitspatches installieren, die Sie finden können. Zum Schluss krempeln Sie Ihre Ärmel hoch und versuchen Sie, es weniger verletzlich zu machen. Es ist, als ob man sich auf ein Schlachtfeld begibt, aber es muß sein.

Für die "exotischen" Betriebssysteme müssen Sie suchen und wählen. Wie üblich und vor allem anderen, sollten die grundlegenden Regeln angewandt werden: Je weniger aktive Dienste, umso besser.

Schutz des Netzwerkes

Wenn die Rechner entsprechend "präpariert" sind, ist die Hälfte des Weges geschafft. Aber Sie müssen noch weiter gehen. Da wir über freie Software sprechen, werden wir eine freie Firewall für den Gateway–Rechner nehmen: es ist die Maschine, die Ihnen den Zugang zur "wilden" Welt gewährt. Erneut wählen wir willkürlich einen Linux–Rechner, so dass wir die Bastille–Linux–Firewall nehmen können. Sie arbeitet entsprechend Ihrer Kernel–Version mit ipchains oder ipfwadm. Wenn Sie einen 2.4–Kernel nutzen, arbeitet er mit iptables.

Eine kleine Abweichung: es ist keine gute Idee, sich mit all den Anfangsproblemen herumzuschlagen, wenn es um die Sicherheit geht. Das "Rennen" nach der letzten Kernel–Version kann zu einer sehr negativen Situation führen. Das heißt nicht, dass die Arbeit an neuen Kernel–Versionen nicht gut ist, jedoch, die "Verbindung" mit existierenden Werkzeugen, die nicht dafür gedacht sind, kann ein großer Fehler sein. Ein Rat: Seien Sie geduldig! Das neue Firewall–Werkzeug, das Teil des 2.4–Kernels ist, ist sehr vielversprechend, aber wahrscheinlich noch etwas "jung". Nachdem wir dies gesagt haben: es liegt an Ihnen ...

Die Bastille–Linux–Firewall ist sowohl einfach als auch effizient. Es gibt jedoch ein sehr viel besseres Werkzeug, etwa wie eine "Gasfabrik", T.REX genannt. Es ist verfügbar von http://www.opensourcefirewall.com. Wenn Sie nach einem sehr raffinierten freien Werkzeug suchen, das ist es.

Es gibt andere Lösungen wie Proxies, sie sind jedoch nicht immer besser. Noch eine Abweichung. Proxies werden oft als "Firewall" bezeichnet. Es sind jedoch zwei verschiedene Dinge. Die Firewalls, über die wir reden, benutzen Paketfilterung und bieten keine Authentifizierungsmethode. Es gibt zwei Arten von Proxy–Servern: Anwendungen oder Socks. Kurz gesagt, ein Anwendungsproxy erledigt für Sie die ganze Kommunikation und gestattet eine Benutzerauthentifizierung. Deshalb benötigt er viel mehr Ressourcen als eine Firewall. Aber, immer wieder gesagt, diese Art Werkzeug schützt nur für eine kurze Zeitspanne. Eine Firewall kann in ungefähr 15 Minuten "geknackt" werden. Gut zu wissen, nicht wahr? Daher die Notwendigkeit, auch die Rechner in Ihrem Netzwerk ordentlich zu sichern: Die Entscheidung, ein Netzwerk nur durch eine Firewall oder einen Proxy zu sichern, grenzt an Ketzerei!

Eine andere Methode zur Reduzierung der Risiken in einem Netzwerk ist Verschlüsselung. Die Benutzung von Telnet ist z. B. wie eine Einladung an Cracker, auf einem roten Teppich hereinzuspazieren. Es gibt ihnen praktisch die Schlüssel zum Laden in die Hand. Sie können nicht nur die umlaufenden Daten sehen, sondern – noch besser – erhalten das Passwort im Klartext: hübsch, nicht wahr? Entsprechend sollten Sie sich frei fühlen, ssh mit den "ungewissen" Protokollen (oder stattdessen) zu verwenden. Wenn Sie Telnet benutzen MÜSSEN (?), senden Sie die Daten durch eine sichere Verbindung. In anderen Worten, leiten Sie den Telnet−Port auf einen sicheren Port um. Mehr darüber finden Sie in dem Artikel "Durch den Tunnel" (www.linuxfocus.org/Deutsch/May2001/article202.shtml). (freie Werbung)

OK, wir haben versucht, die Sicherheit zu verbessern, aber nun sollten wir unsere Arbeit prüfen. Dazu werden wir zu "Crackern", irgendwie: wir werden ihre Werkzeuge nutzen. Ist das nicht häßlich? Auch in diesem Bereich gibt es eine schöne Sammlung von Programmen, wieder wählen wir willkürlich 2 von ihnen: nmap und nessus. Das ist keine Redundanz, da z. B. das zweite das erste voraussetzt. Diese Programme sind Port−Scanner, obwohl nessus viel mehr ist als nur das. Nessus informiert Sie über Systemverwundbarkeiten, indem es die Scan−Ergebnisse mit seiner Schwachstellen−Datenbank vergleicht. Die Benutzung dieser Werkzeuge in einem Netzwerk erlaubt es Ihnen, die Schwachstellen eines jeden Rechners zu entdecken, unabhänig vom jeweiligen Betriebssystem. Die Ergebnisse sind recht aufschlußreich, daher muss man diese Programme einfach haben. Sie finden nmap bei http://www.insecure.org und nessus bei http://www.nessus.org.

Seit dem Anfang dieses Artikel reden wir über das Sichern eines lokalen Netzwerkes, in dem einige Maschinen für die externe Welt geöffnet werden. Der Fall eines Internet−Service−Providers ist sicherlich ganz anders gelagert und wir werden nicht in die vielen Details dieses Themas einsteigen. All das, was wir erwähnt haben, ist immer noch verfügbar, aber Sie werden wesentlich kompliziertere Methoden wie VPN (Virtual Private Network) oder LDAP für Authentifizierung benutzen müssen. Es ist fast ein ganz anderes Thema, weil die Beschränkungen in diesem Fall viel zahlreicher sind. Lassen Sie uns nicht über E−Business−Rechner reden, wo die Dinge ganz sorglos gehandhabt werden. Sie reden von sicheren Rechnern! Erzählen Sie mir nichts ... Senden Sie Ihre Kreditkarten−Nummer über das Internet? Falls ja, dann sind Sie sehr mutig. Vorschlag: Wenn Sie Französisch lesen können, schauen Sie sich einmal diese Webseite an: http://www.kitetoa.com, es lohnt sich.

System−Besonderheiten

Wie bereits erwähnt, sind die Systeme nicht gleichwertig, wenn sie dem Feind gegenüber stehen. Einige haben sehr gute Fähigkeiten, während andere wie Siebe sind. Paradoxerweise (nun, nicht wirklich) sind freie Betriebssysteme unter den besseren. Die verschiedenen BSD−Varianten (OpenBSD, NetBSD, FreeBSD ...), die verschiedenen Linux−Versionen sind ganz vorn, wenn Sicherheit eine Rolle spielt. Auch dies ist das Ergebnis der großartigen Arbeit der Freien−Software−Gemeinschaft. Die anderen, auch wenn sie ein Unix−Label tragen, sind etwas weniger fortgeschritten. Wenn es sich nicht um Unix handelt, ist es noch viel schlimmer!

Alle in diesem Artikel erwähnten Werkzeuge wurden für freie Betriebssysteme entwickelt. Die meisten der proprietären Unix−Systeme können von ihnen profitieren. Diese proprietären Betriebssystem haben jedoch oft ihre eigenen Werkzeuge. Z. B. erwähnten wir in Bezug auf Solaris *ndd* and *aset*. Entgegen einer weitverbreiteten Meinung sind Sun−Systeme keine Modelle für Sicherheit. Ein Werkzeug wie aset erlaubt eine Verbesserung hinsichtlich der Zugriffsrechte. Aset bietet drei Schutzebenen: niedrig, mittel und hoch. Sie können es aus der Shell starten oder als cron−Aufgabe. In einem aktiven Netzwerk ändert sich die Situation ständig, was um 5.00 Uhr richtig war, kann um 5.30 Uhr schon falsch sein. Daher das Interesse, Programme periodisch zu starten, um Homogenität aufrecht zu erhalten. Deshalb hat aset die Möglichkeit, per cron gestartet zu werden. Dann wird es alle 30 Minuten, oder wann immer Sie es für erforderlich halten, die

Berechtigungen von Verzeichnissen und Dateien prüfen ...
Ndd erlaubt es, IP−Stack−Parameter zu ändern. Es erlaubt z. B., den sog. Fingerabdruck eines Systems zu verstecken. Ein erkanntes System ist verwundbarer, weil die Cracker wesentlich besser wissen, wo sie "zuschlagen" können. Mit ndd könen Sie die TCP−Maximum−Segment−Size ändern. Fehlwertmäßig ist diese 536 unter Solaris 2.6. Der Befehl *ndd −set /dev/tcp tcp_mss_def 546* ändert diese auf 546. Je größer die MSS ist, umso besser (aber nicht zuviel). Nmap z. B. findet diese Schwachstelle. Mittels ndd nehmen Sie ihm den Boden unter den Füssen weg. Wenn Sie Maschinen mit Solaris haben, sollten Sie ndd nutzen. Es gibt viele Optionen: lesen Sie die Handbuchseite.
Sie können auch IP−Filter nutzen, ein Werkzeug für Paketfilterung. Es ist verfügbar unter ftp://coombs.anu.edu/pub/net/ip−filter.

Was Irix angeht, ist die Situation wieder etwas anders. SGI (ehemals Silicon Graphics) hat − wie der Name sagt − seine Systeme auf Grafikverarbeitung ausgerichtet. Sicherheit war kein wichtiger Design−Punkt. Notwendigkeit kennt keine Gesetze, es wurde verpflichtend, Wege zur Reduzierung der Risiken zu finden. Dazu wurde in Irix ipfilterd bereitgestellt, es wird jedoch nicht standardmäßig installiert: Sie müssen danach suchen! Ipfilterd wird natürlich für Paketfilterung benutzt; es erlaubt Ihnen, denjenigen Zugang zu verweigern, bei denen Sie es wollen. Es verlässt sich auf eine Konfigurationsdatei namens ipfilterd.conf und hier werden die Dinge etwas trickreich. Die Syntax dieser Datei ist etwas eigentümlich und mag keine unerwarteten Leerzeichen und Leerzeilen. Um der Maschine "mars" Zugang zu der Maschine namens "jupiter" (der SGI−Rechner) zu gestatten, müssen Sie eine Zeile wie
accept −i ec0 between jupiter mars
eingeben. Die in dieser Datei nicht aufgeführten Maschinen können jupiter nicht kontaktieren. Noch schlimmer: Wenn Sie den *ipfilterd_inactive_behavior*−Parameter nicht mittels systune ändern, kann niemand auf die Maschine zugreifen. Das ist doch effizient, oder? Der Fehlwert hierfür ist 1, und Sie müssen ihn mittels *systune −i ipfilterd_inactive_behavior 0* auf 0 setzen.
Eine andere bekannte Sache, die man nicht vergessen sollte: Irix hat eine große Schwachstelle namens fam (File Alteration Monitor). Dieses Programm steuert eine sehr nette Fähigkeit, die Kommunikation zwischen verschiedenen Dämonen. Z. B. erlaubt es dieses Programm, die schönen Icons im Programm−Manager zu erhalten. Nichtsdestoweniger, es gibt nur eins zu tun: Deaktivieren Sie es! Traurig, aber so ist es.

Um mit den Unix−Systemen zu Ende zu kommen, sollten wir erwähnen, das QNX sehr verwundbar ist, aber es kann natürlich auch von den freien Werkzeugen profitieren. MacOS X bietet bereits einige dieser Werkzeuge.

Wir müssen noch etwas über die absolute Referenz unter den Netzwerksystemen sprechen: das einzige und einsame NT 4.0. Dessen Sicherung ist eine utopische Aufgabe, einerlei was der König von Redmond (und viele andere) sagen. Eine Angriffssimulation mit nessus ist z. B. ein Alptraum. Soweit NetBIOS aktiv ist, zeigt Ihnen nessus den Namen jeder Maschine in der Domain mit ihren entsprechenden Benutzern, einschließlich der Administratoren. Die Antwort: Trennen Sie sich von NetBIOS! Richtig, wie bereits erwähnt, kein NetBIOS, kein Netzwerk ... Hier müssen Sie wählen, auf welche Seite Sie sich stellen.
Nessus wird Sie freundlich darüber informieren, dass Sie sich als Gast−Benutzerin mit einer NULL−Sitzung (d. h. OHNE Benutzername und OHNE Passwort) anmelden können. Entfernen Sie es! Ja, aber wie? Und so ähnlich ist alles!
Reduzieren Sie den Zugriff auf Partitionen (NTFS), auf Verzeichnisse. Für FAT−Partitionen ... keine Lösung. Evtl. müssen Sie aufgrund der von Ihnen benutzten Software jedoch FAT−Partitionen nutzen; einige Software arbeitet nicht auf NTFS. Zum Schluß, vermeiden Sie den großartigen IIS, speziell als FTP−Server. Sie sollten ihn einfach nicht installieren. Viele ISPs sind verrückt genug, ihn zu benutzen, wir können ihnen vorschlagen, stattdessen Apache zu nutzen, aber ... Wir sollten nicht zuviel Zeit auf IIS verwenden, es gibt eine Menge an Literatur zu dem Thema.
Tatsächlich gibt es einen Weg, aus dem Sieb einen Filter zu machen (die Löcher sind schmaler). Das Problem ist, dass dies ein recht langer Weg ist und das ganze Magazin wäre nicht genug. Lassen Sie uns nur das wichtigste erwähnen. Offensichtlich können wir die Sicherung nicht mit freier Software durchführen: wir

reden über die Microsoft−Welt! Der erste Vorschlag ist, den MCSE (Microsoft Security Configuration Editor) aus dem ServicePack 4 zusammen mit der MMC (Microsoft Management Console) zu nutzen. Seien Sie jedoch außerordentlich vorsichtig! Wenn Sie einen Fehler machen, haben Sie gewonnen. Natürlich ist diese Software eine englische Version. Wenn Sie eine fremde (nicht englische) Version des Systems nutzen: lassen Sie sich gesagt sein, dass die Mischung von Sprachen in der Redmond−Welt noch nie gute Ergebnisse gezeitigt hat. Sie sind gewarnt. Als nächstes unter den erforderlichen Maßnahmen müssen Sie das Administrator−Konto "sichern" oder sogar deaktivieren. Schauen Sie sich *passprop* aus dem SP 3 an. Sie können auch die Passwörter mittels der *passfilt* dll durch die Registriy verschärfen (Ich habe immer gedacht, dieses Ding sei unter LSD−Einfluß erfunden worden ...). Deaktivieren Sie das berühmte Gast−Konto. Es ist nicht besonders nützlich (s. o.), aber es kann die Dinge weniger schlimm machen. Sie können seinen Zugriff auf Logdateien durch die Registry beschränken. In "HKEY_LOCAL_MACHINE" erstellen Sie die Schlüssel *System\CurrentControlSet\Services\EventLog\Application*, Security und System (diese beiden letzten sollten Application ersetzen). Ihr Name ist "RestrictGuestAccess", der Typ ist REG_SZ und der Wert ist 1. Sie können die Passwörter mit *syskey* verschlüsseln. Vorsicht, dies ist eine Operation, die nicht rückgängig gemacht werden kann! Zum Schluss eine gute Nachricht: sie können den Gastzugang beschränken. Erneut spielen wir mit der Registry, immer noch in "HKEY_LOCAL_MACHINE". Jetzt geht es um den Schlüssel *System\CurrentControlSet\Control\Lsa*. Der Name ist "RestrictAnonymous", der Typ ist "REG_DWORD" und der Wert ist 1. Aber die Microsoft−Welt ist eine harte Nuss: Es sei Ihnen gesagt, dass diese Änderung einige Netzwerkdienste ändern kann ... Unter den wichtigen Dingen können Sie den Zugriff auf einige Ports sperren, über die Netzwerkanwendung im Konfigurations−Panel. Von den TCP/IP−Eigenschaften wählen Sie "Erweitert" und kreuzen Sie das "Sicherheit aktivieren"−Feld an (ich glaube, das ist der Name, aber ich habe es nicht zu Hause und kann es nicht überprüfen). Aus dem "Sicherheits"−Fenster wählen Sie "Nur erlaubt" und wählen Sie die Ports, die Sie aktivieren möchten. Auch hier sollten Sie vorsichtig sein. Sie sollten wissen, was Sie tun, es kann sein, dass einige Dienste nicht mehr funktionieren.
Es kann noch viel mehr getan werden, aber diese sind wesentlich. Um mehr zu lernen, können Sie sans.org besuchen: Eine Menge an Dokumenten ist verfügbar.

Die untragbare Leichtigkeit der Dinge

Nun haben Sie all das getan. Sie starten nessus zum Scannen des gesamten Netzwerkes und Sie finden immer noch Sicherheitslücken. Wir sagen Ihnen nicht, wo die herkommen ... wir wissen es bereits. Vesuchen Sie, diesen System−Ersatz zu täuschen. Es wird die von NetBIOS "bereitgestellten" Löcher nicht entfernen, aber den Schaden begrenzen. Erstellen Sie Unter−Domainen. Melden Sie sich nicht als Administrator an. Wenden Sie Patches an. Zum Schluss, versuchen Sie dies alles hinter Unix−Rechnern als Gateways zu verstecken. Unglücklicherweise kommt die relative Sicherheit nicht nur von Produkten aus Redmond. Ein Netzwerk ist lebendig: es ist immer etwas los. Ein guter Administrator ist immer etwas "paranoid", prüfen Sie entsprechend oft Ihr "festes Inventar". Schreiben Sie Skripte, um die Tests zu automatisieren, um z. B. regelmäßig SUID/SGID−Programme zu kontrollieren, die kritischen Dateien, die Logdateien ... Wenn Sie sich Freunde machen wollen, sperren Sie die Floppy− und CD−Laufwerke Ihrer Benutzerinnen. Akzeptieren Sie es nicht, dass Ihre Benutzer Software ohne Ihre Zustimmung abrufen, ganz besonders, wenn diese Software wie in der Microsoft−Welt üblich ausführbar ist. Halten Sie Ihre Benutzer durch ein Mail−Filtersytem davon ab, angehängte Dokumente im Word− oder Excel−Format zu öffen. Ja, ich weiß, es ist wie Faschismus, aber was können Sie gegen Makro−Viren tun? Benutzen Sie keine Programme wie Outlook. Erneut: Sie müssen wissen, was Sie wollen! Ich weiß, was ich sage, ist nutzlos, aber können Sie bei solchen Produkten von Sicherheit reden? Das berühmte "I love You" hat den Leuten nichts beigebracht.
Was Unix angeht, müssen Dateiabrufe genauso kontrolliert werden. Prüfsummen wurden nicht zufälligerweise erfunden.

Gewöhnen Sie sich an, Ihr Netzwerk regelmäßig mit Logs, Skripten und Scans zu prüfen ... Sie werden bemerken: die Dinge ändern sich recht schnell und nicht nur in einer guten Richtung.

Zum Schluß, wir haben noch nichts darüber gesagt, aber vergessen Sie die Sicherungen nicht. Die Strategie ändert sich nicht: täglich, wöchentlich und monatlich. Auch eine Unix−Maschine kann Probleme haben, obwohl dies ungewöhnlich ist. Und manchmal machen auch die Benutzer Fehler ... aber nicht sehr oft. Es ist bekannt, dass die Probleme immer von der Maschine oder von der verantwortlichen Abteilung kommen:−(

Endlich, es ist vorbei!

Wenn Sie diesen Abschnitt erreicht haben, dann sind Sie sehr tapfer. Das Problem ist, dass wir das Thema nur angekratzt haben! Sicherheit hat kein Ende und betrifft nicht nur Netzwerke. Verwundbare Anwendungen können ein Netzwerk komprimittieren. Eine schlecht konfigurierte Firewall ist viel gefährlicher als überhaupt keine Firewall. Ein Unix−Rechner enthält oft Tausende von Dateien. Wer kann sicher sein, dass keine von ihnen verletzlich ist? Wer glaubt, dass ein Cracker einen 128−Bit−Schlüssel knacken will? Lassen Sie sich nicht täuschen: Er wird versuchen, eine Tür hinter dem Haus zu finden. Erneut: Sie können alle verfügbaren Sicherheitswerkzeuge installieren, wenn Sie ein sehr kleines Loch lassen, da wird das "Böse" durchkommen.

Sicherheit ist auch eine Verhaltensweise: folgen Sie dem aktuellen Geschehen. Besuchen Sie z. B. regelmäßig die Sicherheitswebseiten, ebenso die für Ihr Betriebssystem. So publiziert Sun jeden Monat empfohlene Patches. SGI veröffentlicht alle 3 Monate eine neue Irix−Version. Microsoft veröffentlicht regelmäßig Service−Packs oder HotFixes. Linux−Distributoren veröffentlichen Fehlermeldungen für jede neu entdeckte Verwundbarkeit. Das gleiche gilt für die verschiedenen BSDs. Wenn Sie die Produkte zu einem Patch nicht nutzen, entfernen Sie diese von Ihrer Festplatte. Und so weiter: die Liste, der zu erledigenden Sachen ist eine sehr, sehr lange. Kurz gesagt, dieser Job sollte keine Arbeitslosigkeit kennen.

Zum Schluß will ich es nochmals sagen, all dieses wird nur dazu beitragen, Ihr Netzwerk etwas weniger verletzlich zu machen. Erwarten Sie nicht, dass Sie ein zu 100 Prozent sicheres Netzwerk erhalten werden, auch nicht zu einer bestimmten Zeit (nun ja, vielleicht wenn alle Maschinen gestoppt sind). Dies gesagt, ist es keine Voraussetzung für diesen Job, paranoid zu sein ... aber es hilft! Aber seien Sie in Ihrem täglichen Leben nicht so, es wird für die Menschen um Sie herum viel angenehmer sein ...

Referenzen

- http://www.linuxsecurity.com
- http://www.sans.org
- http://www.infosyssec.org
- http://www.securityfocus.com
- http://www.cs.purdue.edu/coast/hotlist/

Das Leben ist traurig: Lasst uns etwas Spass haben!

Auch eine Möglichkeit, den Job zu erledigen ;−)

Translation information:
fr −−> −− : Georges Tarbouriech <georges.t(at)linuxfocus.org>
fr −−> en: Georges Tarbouriech <georges.t(at)linuxfocus.org>
en −−> de: Hermann J. Beckers <beckerst/at/lst−oneline.de>

2005−01−11, generated by lfparser_pdf version 2.51