Tutorial/1:
Installieren und Härten

Pinguin-Patrol

Lukas Grunwald

Nur wer seine Systeme im Griff hat und gleichzeitig auf Details achtet, kann sie gegen Manipulation und unbefugten Zugriff sichern. Das fängt bei der Installation an und hört aber nach der Konfiguration längst nicht auf.

Kaum eine Woche vergeht, da nicht die Kunde neu entdeckter Sicherheitslücken die Runde macht – die auch für Linux-Rechner gelten. Den größten Schaden verursachen unzureichende Konfiguration der Systeme und mangelnde Organisation. Dementsprechend beschäftigt sich der erste Teil dieses Tutorials mit dem generischen Absichern eines Linux-Systems am Beispiel eines dedizierten Webservers.

Um ein System zu härten, das heißt sein Sicherheitsniveau zu erhöhen, empfiehlt es sich, es innerhalb eines separaten Netzes ohne Verbindung zum internen Firmen-Intranet oder zu externen Netzen wie dem Internet neu aufzusetzen. Dafür verbindet man einen zweiten Rechner, der als Auditsystem dient, über ein Crossover-Kabel oder einen Hub mit dem künftigen Server. Ein weiterer Rechner mit Internet-Anschluss dient der Suche nach Patches und Updates, mit Wechselmedien lassen sich diese einspielen.

Bei der Installation einer Linux-Distributionen ist darauf zu achten, dass man die minimale Konfiguration wählt, die dem Zweck entspricht, den der Server, die Firewall oder das sicherheitskritische System erfüllen soll. Nichtsdestotrotz bringen die meisten Installationstools eine Menge überflüssiger und sicherheitskritischer Komponenten auf das System, die man im Nachhinein manuell entfernen oder härten muss.

Deshalb ist eine automatische Installation auf jeden Fall zu vermeiden. Stattdessen empfiehlt es sich, über die manuelle Auswahl das System so klein wie möglich halten. Vor allem sollte man sämtliche Entwicklertools und Programmiersprachen aus der Auswahl entfernen, die ein Angreifer für seine Zwecke, etwa das Kompilieren von Angriffswerkzeugen oder Root-Kids, benutzen könnte. Auch sollte man nach dem Aufspielen des Basissystems – oder des zu härtenden Zielsystems –, weder *taskdirect* (Debian) noch *yast* (Suse) ausführen, da sonst die Flut der Pakete das Härten und Abspecken des Systems unweigerlich erschweren würde.

Auf der Suche nach Schwachstellen

Während der Grundkonfiguration bieten ein Debian- ebenso wie ein RedHat-Linux als Erstes die Option, statt der klassischen Unix-Crypt-Passwörter MD5-Passwörter zu verwenden, was sich dringend empfiehlt, da sie einen besseren Schutz bieten. Wer die MD5-Passwörter nachträglich aktiviert – etwa bei einem Suse-Linux in den Dateien */etc/security/pam_unix2.conf* und */etc/security/pam_pwcheck.conf* die Option *password: md5* setzt –, sollte danach das *root*-Passwort neu vergeben, um den MD5-Hash-Wert zu bilden.

Die Aktivierung der Shadow-Passwörter stellt außerdem sicher, dass Angreifer bei einer Schwachstelle des Webservers die verschlüsselten Passwörter nicht ohne „Root Exploit“ auslesen können.

Nach der Installation ist über das Internet zu überprüfen, ob weitere Hotfixes oder Updates beziehungsweise Upgrades verfügbar sind. Außerdem empfiehlt es sich, über die bekannten Vulnerability Databases (Schwachstellen-Datenbanken) wie Bugtraq, Security Focus oder Open Source Vulnerability Database (s. Kasten „Wichtige URLs“) zu überprüfen, ob bekannte Schwachstellen in den verwendeten Versionen vorhanden sind.

Ist der Upgrade-Zyklus durchlaufen, kann das Härten des Systems beginnen. Zuerst ist vom Auditsystem aus mit dem Portscanner *nmap* zu überprüfen, welche Rechnerports in der Defaultkonfiguration geöffnet sind. Das Ergebnis des Portscans fließt nach der Interpretation selbstredend in die Dokumentation für Auditkreislauf und Security-Change-Management über künftige Änderungen und Updates ein.

Durch das Abspecken des Start- oder *init*-Systems vermeidet man das automatische Starten überflüssiger Dienste: entweder durch händisches Entrümpeln oder erneutes Aufsetzen. Wie man das gesamte Start-up-System durch ein einfaches Init-Skript ersetzen und damit den gesamten Systemboot- und Start-

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.

iX-TRACT

- Die Absicherung von Servern fängt bereits bei der Installationsmethode an.
- Je geringer die Anzahl der installierten Software-Pakete ist, desto einfacher ist ein dedizierter Webserver zu warten und desto weniger Angriffsfläche bietet er.
- Um eine mehrstufige Absicherung des Systems zu erreichen, sollte man alle vom System bereitgestellten Mechanismen nutzen: vom Kernel über den TCP-Wrapper bis zur Anwendungsebene.

prozess vereinfachen kann, beschreibt der dritte Teil des Tutorials; der erste Teil bleibt beim Aufräumen per Hand.

Vor dem Entrümpeln ist zu überprüfen, welche Dienste des Super-Dämons *inetd* laufen müssen. Überflüssige Services lassen sich durch Auskommentieren der entsprechenden Zeilen in der */etc/inetd.conf* deaktivieren. Das gilt auch, wenn der Super-Dämon keinen einzigen Dienst starten soll. Damit lässt sich ein Starten über die Kommandozeile verhindern. Ein erneuter Aufruf von *nmap* vom Auditsystem aus zeigt, ob die Dienste tatsächlich deaktiviert sind.

Sind die *inetd*-Komponenten gebändigt, überprüft das Kommando *netstat –na*, ob weitere Services und Dämonen auf TCP- und UDP-Sockets gebunden sind. Nach dem Deaktivieren sollte man die überflüssigen Dienste aus der Runlevel-Konfiguration entfernen und deinstallieren. Ein weiterer Audit mittels *nmap* schafft erneut Klarheit über den Zustand der Ports.

Der Widerspenstigen Zähmung

SSH hat sich inzwischen als Ersatz für das betagte und unsichere Telnet etabliert, das in seiner Standardeinstellung keinerlei kryptografische Maßnahmen unterstützt. Doch auch SSH bedarf einer gründlichen Konfiguration, will man das Sicherheitsniveau des Systems so hoch wie möglich halten. Zuerst ist nach dem Erzeugen des Host-Keys dessen Fingerprint zu notieren und das Schlüsselpaar zu sichern, auszudrucken und in die Security-Change-Dokumentation aufzunehmen. Dadurch kann der Admin den Fingerprint bei der ersten Verbindung von der Administrationskonsole zum Webserver händisch überprüfen, bevor er ihn akzeptiert und permanent auf der Administrationskonsole hinterlegt.

Danach sollte man auf jeden Fall auf dem Webserver für den *sshd* das SSH-1-Protokoll deaktivieren, da es eine potenzielle Schwachstelle darstellt. Zudem ist auf dem Administrationsrechner ein SSH-Schlüsselpaar zu erzeugen, etwa mit *ssh-keygen –t dsa –b 2048*, und mit einem Key-Passwort zu schützen. Denn wenn der private Teil des Schlüssels in falsche Hände gerät, muss der Angreifer erst das Passwort knacken. Der Administrator gewinnt dadurch etwas mehr Zeit, das Schlüsselpaar gänzlich auszutauschen.

Sobald das Public/Private-Key-basierte Login funktioniert, kann man das gesamte passwortbasierte Login-Subsystem abschalten. Dazu tauscht man in der Datei */etc/shadow* den MD5-Hash, der das Root-Passwort repräsentiert, respektive in */etc/passwd* das *x*, durch ein Sternchen „*“ aus.

Ist das Root-Passwort deaktiviert, kann der Admin den klassischen Unix-Passwort-basierten Login-Mechanismus im *sshd* über den Eintrag *“PasswordAuthentication no“* in der Datei */etc/ssh/sshd_config* deaktivieren. Wichtig dabei ist, dass auch die PAM-Authentifizierung mit *"PAMAuthenticationViaKbdInt no"* ausgeschaltet ist (s. Listing 1).

Putzkolonne im Anmarsch

Beim ersten Login auf den Rechner sollte der SSH-Dämon den Fingerprint ausgeben. Stimmt er mit dem der Dokumentation überein, kann man ihn archivieren, da das System nun vor Man-in-the-Middle-Attacken während einer SSH-Sitzung geschützt ist. Sollte die

Tutorial-Übersicht

Teil 1: Härten und Abspecken eines sicherheitsrelevanten Linux-Minimalsystems am Beispiel eines dedizierten Webservers.

Teil 2: Erstellen von Ereignismitschnitten und Auswertung von Log-Dateien.

Teil 3: Konfigurieren und Härten einer Firewall am Beispiel von Linux. Planung, Dokumentation und Aufbau von Audit-Kreisläufen.

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.

Immer schön ordentlich

Die technischen Maßnahmen sind bestenfalls die halbe Miete, wenn es um IT-Sicherheit geht. Ein sicheres System erfordert vor der Implementierung eine gründliche Planung. Wichtig für den gesamten Prozess ist vor allem die lückenlose Dokumentation aller geplanten und durchgeführten Schritte. Zudem muss nach jedem durchgeführten Schritt eine Analyse – auch Audit genannt – überprüfen, ob die Aktionen richtig und vor allem wirksam durchgeführt wurden. Konsequenterweise müssen die Ergebnisse der Analyse in der Überarbeitung des Konzepts ihren Niederschlag finden. Damit schließt sich ein Kreislauf von Planung, Implementierung, Dokumentation, Audit und Überarbeitung, der sich auf der nächsten, hoffentlich höheren, Stufe wiederholt (s. Abb. 1).

Trivial, aber oft nicht bedacht: Es ist sinnlos, nur die Sicherheit eines einzelnen Systems zu verbessern. Genauso gut könnte man bei einem Gartenzaun eine zehn Meter hohe Latte anbringen, während die übrigen nur 50 Zentimeter hoch sind.

Den Beginn der Planung bildet eine Bestands- und Bedarfsanalyse. Dabei ist durch Audit-Tools – zum Beispiel einen Portscan – zu überprüfen, welche Dienste das entsprechende System nach außen anbietet. Als nächsten Schritt sollte man alle benötigten respektive erwünschten Dienste, die man durch geschicktes Befragen der Benutzer und ähnliche Maßnahmen in Erfahrung bringen kann, zusammenstellen. In einer Tabelle lassen sich die tatsächlich auf den Systemen laufenden, die benötigten und die optionalen Dienste übersichtlich gegenüberstellen.

Anschließend sind für die Rechner die Betriebskonzepte zu schreiben. In diesen wird definiert, wie die Trägersysteme auszusehen haben, welches Betriebssystem, welche Distribution, welche Version des Kernels oder der Servicepacks eingesetzt werden sollen, und welche Software dafür zuständig sein soll.

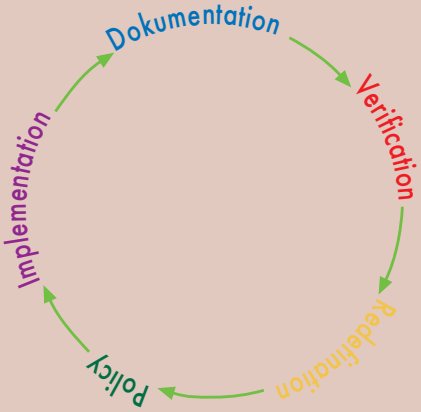
Solche Betriebskonzepte müssen flächendeckend für alle funktionellen und administrativen Einheiten erstellt werden, etwa für die Firewall, den Mail- und den Fileserver, denn das berühmte schwächste Glied bringt die Kette zum Reißen, gefährdet also das ganze Netz. Berechtigungs- und Virenschutzkonzepte sind ebenso notwendig wie ein Konzept für einen Syslog-Server, einen Konsolenserver oder das Zugangsberechtigungskonzept für den Rechnerraum.

Diese Entwürfe müssen immer in gedruckter Form vorliegen – denn was nützt ein Konzept für den Wiederanlauf nach einem totalen Stromausfall, wenn es als Onlinedokument auf einem der ausgefallenen Server liegt.

Zum Berechtigungskonzept etwa gehören sämtliche Vorgaben, wie wer und vor allem mit welchen Rechten auf welche Ressource zugreifen darf. Zudem sind in dem Betriebskonzept die verantwortlichen Personen und Abteilungen für das System niederzulegen, ebenso eine Backup- und Recovery-Strategie zu definieren und eine Methode zu beschreiben, wie nach einem Totalausfall ein Wiederanlauf erfolgen kann.

Das Berechtigungskonzept für den Serverraum beispielsweise kann besagen, dass dieser Raum immer abzuschließen sei und der Schlüssel beim Administrator beziehungsweise beim Chef zu liegen habe. Dort darf er keinesfalls offen zugänglich sein, etwa an einem Schlüsselbrett. Wenn jemand den Serverraum betreten hat, muss er dies auf einer Liste mit Namen, Zeitpunkt und Grund dokumentieren. Die Liste ist ein Geschäftsdokument und entsprechend zu archivieren.

Neben dem Betriebskonzept bedarf es eines Security-Change-Managements, das die Versionsstände und Patchlevel sämtlicher benutzter Software auf den Systemen erfasst. Ziel dieses Mechanismus ist es, einerseits den Audit zu vereinfachen und andererseits beim Auftreten von neuen Schwachstellen und Sicherheitslücken sofort überprüfen zu können, ob man davon betroffen ist. Ob die dokumentierten Softwareversionen tatsächlich im produktiven Betrieb eingesetzt sind, muss ein Audit, der regelmäßig gegenüber den Security-Changes zu fahren ist, überprüfen.



Die Absicherung eines Systems ist ein permanenter Prozess, der sich als Kreislauf darstellen lässt (Abb. 1).

Nichts mit „mal eben schnell"

Der größte Feind aller Konzepte ist bekanntlich ihre fehlende Umsetzung. Die umfangreichsten Dokumente nützen niemandem etwas, wenn sie im Schrank verschimmeln, etwa weil die IT-Abteilung hoffungslos überlastet ist, Kosten gespart werden sollen oder IT-Sicherheitsbeauftragte nicht in der Lage sind, ihre Bedenken gegenüber ihren Vorgesetzten zu artikulieren.

Häufig werden zusätzliche Sicherheitsmaßnahmen, die nicht – wie bei Banken – gesetzlich vorgeschrieben sind, auch mit realitätsfremden Sätzen wie „Warum sollte es ein Angreifer gerade auf uns abgesehen haben?" abgetan. Doch Skript-Kiddies ist es egal, wer ihr Ziel ist – und die Kosten treten erst auf, wenn die Verfügbarkeit oder Integrität nicht mehr gewährleistet ist.

Solange nicht auch IT-Leiter und Chefs bereit sind, sich einer Policy unterzuordnen, und stattdessen den „Hey-Joe"-Support mit „mach mal schnell" einer geordneten Beauftragung und einem Security-Change-Management vorziehen, hilft keine noch so durchdachte Policy, die IT-Sicherheit zu steigern.

Die Administratoren haben sich ebenfalls umzugewöhnen: Schnell ein Update via Internet auf dem Server zu installieren, ist dann nicht mehr möglich. Zumindest sollte eine Testabnahmephase sicherstellen, dass bei einem Update die MD5-Checksumme einer Datei mit der Original-Checksumme vom Hersteller übereinstimmt. Bei besonders kritischen Rechnern ist sie via Telefon oder einen anderen – gesonderten – Kommunikationsweg zu überprüfen.

Neben der Dokumentation aller sicherheitsrelevanten Änderungen am System hat auch das Mitprotokollieren und Niederlegen von sicherheitsrelevanten Ereignissen und deren Auswertung Bedeutung. Das kann beispielsweise über einen Syslog-Server geschehen, der zentral alle System- und Service-Meldungen einsammelt und die Meldungen manipulationssicher abspeichert. Details dazu lesen Sie im zweiten Teil dieses Tutorials.

Sicherheit auf dem Admin-Rechner nicht gewährleistet sein, empfiehlt es sich, nach der Sitzung die *~/.ssh/known_hosts* zu löschen und mit Hilfe einer Checkliste den Hostkey bei jeder Verbindung manuell zu verifizieren.

Nun muss der Administrator entscheiden, welche Pakete nicht für den Betrieb des Webservers notwendig sind und dem Aufräumkommando zum Opfer fallen können. Das Kommando *dpkg --list* respektive *rpm –qa* gibt eine komplette Liste der installierten Pakete aus, die ebenfalls in die Dokumentation gehört. Die überflüssigen Pakete lassen sich mit den Befehlen *apt-get remove* oder *rpm –e* auf der Konsole entfernen. Sollte letzterer auf Widerstand stoßen,

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.

Listing 1: /etc/ssh/sshd_config

```
alf:# This is the sshd server system-wide configuration
file.
# See the sshd(8) manpage for defails
#
# What ports, IPs and protocols we listen for
Port 22
#
# Beschränken von Interface und Port
# auf dem der Dämon horchen soll,
# wenn mehr als eine NIC vorhanden ist!
ListenAddress 192.168.181.12:22
#
# Nur SSH Version 2 verwenden
Protocol 2
#
# HostKeys für Protokoll Version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
#
#Privilegien-Trennung der Kinds-Prozesse
UsePrivilegeSeparation yes
#
#Abschalten der PAM-Authentifizierung:
PAMAuthenticationViaKbdInt no
#
# Logging:
SyslogFacility AUTH
LogLevel INFO
#
# Authentifizierung:
LoginGraceTime 600
PermitRootLogin yes
StrictModes yes
RSAAuthentication yes
PubkeyAuthentication yes
RhostsAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
#
# Abstellen des Password-basierten Logins:
PasswordAuthentication no
X11Forwarding no
PrintMotd no
KeepAlive yes
```

können die Optionen *--force* und *--nodeps* zum Erfolg verhelfen, aber auch zu Inkonsistenzen führen.

Auf keinen Fall gehören Automounter, *at*- oder *cron*-Dämon auf ein System, das als dedizierter Webserver laufen soll, ebenso wenig ein *telnet*-Client. NFS, NIS und DHCP sollten weder als Client noch als Server vertreten sein, PPP und *ping* gehören ebenfalls nicht hierher. Wer auf Systemnachrichten per Mails nicht verzichten will, sollte seinen Mailserver nur im Pipe-only-Modus betreiben und nicht auf einen von außen zugänglichen Socket binden. Dadurch vermeidet man, dass der Mail-Dämon permanent läuft und als Opfer einer Attacke die Tür zum System öffnen kann.

cpio, *dump* respektive *tar* schließen den Reigen der zu entfernenden Pakete, nachdem mit ihnen zuvor ein Backup vom System erzeugt wurde, beispielsweise mit *find / –print –depth –xdev | cpio –ov –H crc | gzip > /var/backups/mybackup.cpio.gz*, auszuführen für jedes einzelne Dateisystem.

Ein – radikales – Verfahren besteht darin, nach den überflüssigen Paketen und den *deb*- oder *rpm*-Tools die Paketmanagementdatenbanken zu löschen. Das verhindert aber weitere automatisierte Updates, etwa um Security-Patches komfortabel einzupflegen. Daher ist es ratsam, die Paketmanagementtools auf dem System zu belassen und nur die überflüssigen Pakete zu entfernen.

Eine Alternative besteht darin, die Patches über ein Schattensystem mit Paketmanagement einzuspielen: Nachdem man über ein Image ein zweites identisches System im Admin-Netz erzeugt hat, kann man auf dem Webserver – dem eigentlichen Zielsystem der Updates – die Paketmanagementtools entfernen. Auf dem Schattensystem erzeugt man nach dem Update mit *find / –print –depth > copy.txt* eine Liste des Dateisystems, aus der die Einträge der unnötigen Dateien zu löschen sind. Anschließend lässt sich mit *cat copy.txt | cpio –ov –H crc > system.new.cpio* ein Image ohne Pakettools und -datenbank erstellen und auf dem Zielsystem mit *cat system.new.cpio | cpio –idv* aufspielen.

Auf jeden Fall sollte man die Manpages und die Manual-Datenbank entfernen, um einem potenziellen Angreifer nicht auch noch die Dokumentationen und Hilfeseiten auf dem silbernen Tablett zu präsentieren.

Fingerabdrücke für die Akten

Nachdem die Zahl der Pakete auf ein Minimum reduziert ist, lässt sich mit dem Fingerprinting fortfahren. Dazu erstellt man vom Wurzelverzeichnis aus über das gesamte Dateisystem eine Liste von MD5-Checksummen für sämtliche Binaries. Ausnahmen bilden dynamisch sich verändernde Filesysteme wie */proc* und die *tmp*-Verzeichnisse, da es dort keinen Sinn ergibt, Checksummen zu bilden. Der Befehl *find / –xdev –type f –perm –700 –exec md5sum {} \;* gibt eine solche Liste aus. Sie kann später zur Dokumentation, zum Entdecken von Rootkits und anderen Manipulationen am System oder für eine spätere forensische Analyse genutzt werden.

Zusätzlich kann man mit *find /etc –type f –exec md5sum {} \;* eine weitere Liste von Fingerprints über wichtige Dateien im */etc*-Verzeichnis erstellen. Auch müssen die Libraries, die nicht immer als ausführbar gekennzeichnet sind, über das Erstellen von MD5-Checksummen gesichert werden, da sich in ihnen ausführbarer Code befinden kann. Glücklicherweise ist auf

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.

Listing 2: /etc/nsswitch.conf (Debian)

```
@lf:# /etc/nsswitch.conf --
# Example configuration of
# GNU Name Service Switch functionality.
#
passwd:         files
group:          files
shadow:         files

hosts:          files dns
networks:       files dns

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       db files
```

einem abgespeckten System die Zahl der Bibliotheksverzeichnisse nicht allzu groß.

Danach beginnt das Netz-Setup. Grundsätzlich sollte man alle vom System bereitgestellten Mechanismen nutzen, um ein dreistufiges Sicherheitskonzept zu bedienen: Via *iptables* lassen sich nicht benötigte Ports blockieren, über den TCP-Wrapper Dienste unterbinden, die Adressräume von Clientsystemen auf Anwendungsebene beschränken und – so es die Software zulässt – einzelne Dienste auf bestimmte Interfaces binden. Ein gutes Beispiel dafür bietet der *sshd* (s. Listing 1).

Zuerst sollte man sämtliche relevanten Netzwerk- und Interface-Adressen als IP-Adresse ohne DNS-Namensauflösung manuell in die Datei */etc/hosts* eintragen. Dorthin gehören neben Hostnamen und den eigenen Adressen mindestens die von Time- und Syslog-Server, sowie Gateways und Default-Route, damit das System nicht für eine DNS-Spoof-Attacke anfällig ist. Des Weiteren muss in der Datei */etc/nsswitch.conf* die Resolve-Order für Hosts von *„dns files“* auf *„files dns“* umgestellt werden, damit der Rechner die lokale Hostliste den DNS-Daten vorzieht (s. Listing 2).

Ist ein DHCP-Client vorhanden, sollte man ihn entfernen und in den Dateien */etc/network/interfaces* (Debian) respektive */etc/sysconfig/network/ifcfg-** (Suse) die IP-Adresse, das Gateway und die Broadcast-Adresse statisch eintragen (siehe Listing 3). Außerdem empfiehlt es sich, die MAC-Adresse des Routers und der benachbarten Systeme im Intranet beziehungsweise in der DMZ (Demilitarisierte Zone) per *arp –s* ebenfalls statisch einzutragen, um ein Arp-Spoofing zu verhindern, mit dem sich Man-in-the-Middle-Attacken auf OSI-Layer 2 gegen das System durchführen lassen.

Danach ist über den TCP-Portwrapper der Zugriff auf der Zwischenebene ebenfalls zu sichern. Der Eintrag *ALL: PARANOID* in der Datei */etc/hosts.deny* blockiert generell den Zugriff für alle Programme, die den TCP-Wrapper benutzen und dokumentiert zudem die abgewiesenen Kommunikationsversuche.

Identitätswechsel unterbinden

Der Beschränkung von Remote-Zugriffen folgt die Suche nach Binaries mit SUID-Flag (Set User ID), die gute Angriffpunkte bilden, da der normale Benutzer sie mit *root*-Rechten ausführt. In der Regel gehören dazu *login*, *su*, *mount* und *umount*, allesamt im Verzeichnis */bin* zu finden. Sie lassen sich mit dem Befehl *find /bin –perm –4000 –type f* aufspüren, weitere mit *find / –perm –4000 –type f*.

Da das passwordbasierte Login deaktiviert ist, lassen sich die SUID-Bits von *login* und *su* ebenso wie die von *gpasswd*, *passwd* oder *unix_chkpwd* – neben anderen unter */usr/bin* zu finden – ohne Bedenken entfernen. Deshalb genügt die Eingabe von *find / –perm –4000 –type f –exec chmod 755 {} \;* auf der Konsole, um gleich alle SUID-Bits zu entfernen, statt für jedes einzelne gefundene Binary ein *chmod 755* auszuführen.

Zwar ist das System jetzt frei von SUID-Programmen, aber nicht von solchen mit SGID-Bit (Set Group ID). Die lassen sich mit *find / –perm –2000 –type f* ermitteln. Auf dem Beispielrechner *ixdemo* listet der Befehl *wall*, *chage*, *expiry* und *write* auf. Vor dem Ausführen von *find / –perm –2000 –type f –exec chmod 755 {} \;* sollte man sich vergewissern, dass sich darunter keine Log-Dateien befinden.

Auch die Dateisysteme *ext2*- und *ext3* erlauben es, mit Attributen seine Dateien vor dem *root*-User zu schützen. Dazu kann man mit dem Befehl *chattr* die *ext2/ext3*-Attribute so modifizieren, dass man an Dateien nur etwas anhängen respektive eine Datei weder löschen noch modifizieren kann. Das erreichen die Optionen *+a* (append-only) oder *+i* (immutable). Beispielsweise kann der Befehl *chattr +i /etc/ssh/sshd_config /root/.ssh/authorized_keys2 /etc/passwd /etc/shadow* die wichtigen Dateien für das Login schützen.

Will ein Angreifer durch einen Fehler in einem CGI-Skript (Common Gateway Interface) oder mit einem Rootkit die Datei */etc/passwd* um einen weiteren Benutzer – etwa durch *echo roor::0:0:root:/root:/bin/bash >> /etc/passwd* ohne Password erweitern, schlägt sein Versuch fehl. Echte Angreifer, die Zugriff auf eine *root*-Shell auf dem System erlangen, lassen sich jedoch von solch simplen Sicherheitsmerkmalen nicht abhalten.

Zentrale Übersetzungsarbeiten

Um die Angriffsfläche weiter zu verringern, ist der Bau eines neuen Systemkerns notwendig. Erstens ist grundsätzlich darauf zu achten, dass der neue Kernel nur die wirklich benötigten Teile von Treibern und Kommunikationskomponenten enthält. Als Dateisysteme genügen *ext2*/*ext3* zusammen mit dem *proc*- und *shm*-Filesystem. Die Unterstützung von Floppys, VFAT- oder CD-ROM-Dateisystemen ist überflüssig. Zudem sollten nur Treiber für tatsächlich vor-

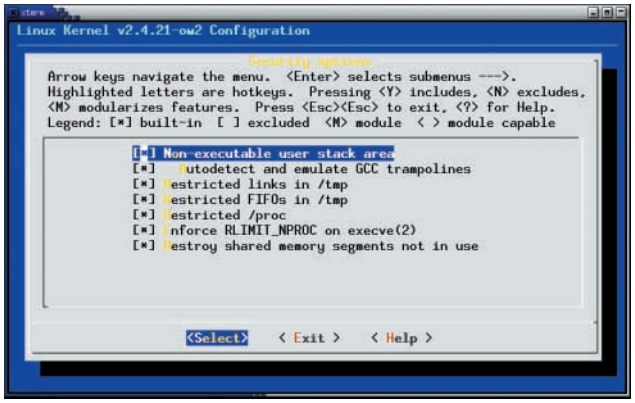
Listing 3: /etc/network/interfaces

```
@lf:# /etc/network/interfaces --
# configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# The first network card
# iface eth0 inet dhcp
auto eth0
iface eth0 inet static
    address 192.168.181.12
    netmask 255.255.255.0
    broadcast 192.168.181.255
    gateway 192.168.181.1
```

WICHTIGE URLS

www.vulnwatch.org	Vulnerability Disclosure List
packetstormsecurity.nl	Security Tools Resource
www.osvdb.org	Open Source Vulnerability Database
www.securityfocus.com	Security Focus Vulnerability Database von Symantec
lists.insecure.org/bugtraq	Bugtraq, Internet Security List
www.secunia.com	Secunia Advisory And Vulnerability Database
www.kernel.org	Quelle des „Vanilla-Kernels“
www.nsa.gov/selinux	SE-Linux-Kernel, -Userland und Patches der NSA
acl.bestbits.at	ACL-Unterstützung für Linux
www.openwall.com/linux	Openwall Patches

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.



Nach dem Einspielen der Openwall-Patches existiert ein zusätzliches Menü in der Kernel-Konfiguration (Abb. 2).

handene und benötigte Hardware eingebaut sein: Bei einem Webserver kann man getrost auf den Support von Multimedia-Hardware wie Soundkarten und des USB-OHCI verzichten.

Zum Zweiten sollte der neue Kernel auf jeden Fall ein statischer sein, denn ein modularer birgt die Gefahr, dass ein Angreifer Rootkits über den Module Loader ausführen kann. Deshalb müssen alle benötigten Treiber als „build-in“ ausgewählt sein, nicht als ladbare Module. Um sicherzugehen, dass der Kernel keine, vor allem fremde Module lädt, muss die Modul-Unterstützung im Kernel deaktiviert sein. Außerdem sollten die *modutils* wie *modprobe* oder *insmod* nicht installiert sein.

Zudem kann man den Systemkern nicht auf dem zu härtenden System bauen, da dort weder Entwicklungsumgebungen wie der *gcc* noch Linker installiert sein dürfen: Sie befinden sich auf einem separaten Rechner im Labor. Dort kann man den Kernel ohne Bedenken kompilieren und mit dem Befehl *scp* (secure copy) über SSH unter Verwendung des bereits hinterlegten Root-Keys auf den neuen Server bringen. Nachdem der Kern darauf installiert ist, muss man gegebenenfalls noch den Bootloader anpassen und das System mit dem gehärteten Kern neu booten.

Für ein sicherheitsrelevantes System wie den Webserver bieten sich mehrere Kernelvarianten beziehungsweise Patches an. Grundlage sollte jedoch immer der „Vanilla Kernel“, das heißt der ungepatchte Kernel von ftp.kernel.org sein: Beispielsweise lässt sich das System mit den von der NSA stammenden Security Enhanced (SE) Linux Kernel Patches und Tools durch fein graduierte Rechte genau an ein eigenes Berechtigungskonzept anpassen. Allerdings ist das nur auf einem Mehrbenutzersystem sinnvoll, zumal es einen immensen Audit-Aufwand mit sich bringt. *iX* wird das SE-Linux in der nächsten Ausgabe ausführlich behandeln.

Ein anderer Weg besteht darin, die Openwall-Patches auf einen Vanilla-Kernel anzuwenden. Sie bringen spezielle Erweiterungen und Sicherheitsupdates mit und ändern das Verhalten des Systems: Etwa ist die „user stack area“ nicht mehr ausführbar (s. Abb. 1). Das entschärft manche Buffer Overruns und Exploits. Außerdem entsorgt der Kernel Shared Memory, das kein Prozess mehr nutzt.

Damit der Admin überprüfen kann, dass der Stack nicht mehr ausführbar ist, liefern die Openwall-Patches das Programm *stacktest* mit. Nach der statischen Übersetzung kann man es auf dem Zielsystem zur Kontrolle der Openwall-Patches einsetzen: Warnungen des Systemkerns, die unter *kern.alert* gespeichert sind, gehen an den Syslog-Server.

Zudem sollte die Systemkonsole über den seriellen Port sowie die Ausgabe von der Konsole auf einen Printerport aktiviert sein, damit man im Zweifelsfall einen Log-Printer anschließen kann. Beim Netz-Setup ist es ratsam, die *iptables*-Paketfilter zu aktivieren, um gegebenenfalls schon auf Kernelebene einen Schutz herbeizuführen. Auch ist der Enhanced Realtime Clock Support zu aktivieren, den der nächste Teil des Tutorials unter den Aspekten des Sysloggings und Timekeepings genauer betrachten wird. (sun)

LUKAS GRUNWALD

arbeitet als Consultant bei der DN Systems GmbH in Hildesheim und ist in diverse freie Softwareprojekte involviert.

Literatur

[1] Jörg Fritsch; IT-Sicherheit; Security-Checks vernetzter Systeme; Fangfragen; *iX* 11/2002, S. 48

© Copyright by Heise Zeitschriften Verlag GmbH & Co. KG. Veröffentlichung und Vervielfältigung nur mit Genehmigung des Heise Zeitschriften Verlags.